Poster: Time-Efficient Sparse and Lightweight Adaptation for Real-Time Mobile Applications

Hyeongheon Cha
KAIST
hyeongheon@kaist.ac.kr

Taesik Gong
Nokia Bell Labs
taesik.gong@nokia-bell-labs.com

Sung-Ju Lee
KAIST
profsj@kaist.ac.kr

ABSTRACT

When deployed in mobile scenarios, deep learning models often suffer from performance degradation due to domain shifts. Test-Time Adaptation (TTA) offers a viable solution, but current approaches face latency issues on resource-constrained mobile devices. We propose ***TESLA***: **T**ime-**E**fficient **S**parse and **L**ightweight **A**daptation strategy for real-time mobile applications, which skips adaptation for specific batches to increase the inference sample rate. Our method balances model accuracy and inference speed by accumulating domain-informative samples from non-adapted batches and sparsely adapting them. Experiments on edge devices demonstrate competitive accuracy even with sparse adaptation rates, highlighting the effectiveness of our approach in real-time mobile applications. Our strategy can seamlessly integrate with existing lightweight adaptation and optimization algorithms, further accelerating inference across diverse mobile systems.

CCS CONCEPTS

• **Computing methodologies** → **Learning under covariate shift**; **Online learning settings**.

KEYWORDS

Test-Time Adaptation, Domain Adaptation, Efficient Learning

ACM Reference Format:
Hyeongheon Cha, Taesik Gong, and Sung-Ju Lee. 2024. Poster: Time-Efficient Sparse and Lightweight Adaptation for Real-Time Mobile Applications. In *The 22nd Annual International Conference on Mobile Systems, Applications and Services (MOBISYS '24), June 3–7, 2024, Minato-ku, Tokyo, Japan.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3643832.3661442

1 INTRODUCTION

While deep learning has revolutionized AI, its performance degrades significantly under domain shifts caused by environmental changes or noise [1]. To mitigate this, Test-Time Adaptation (TTA) is a promising solution utilizing only test samples without source or labeled data. Most TTA algorithms have been designed and evaluated on GPU-accelerated scenarios, focusing on improving classification accuracy. In contrast, many real-world applications necessitate deploying TTA on resource-constrained edge devices with low computational power, such as mobile devices, CCTVs, or sensors. Existing TTA methods face significant latency bottlenecks on these devices, hampering their applicability for real-time applications such as autonomous driving, in which rapidly varying conditions demand efficient domain adaptation.

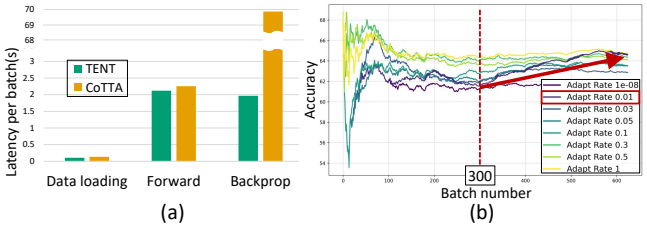

Figure 1: TTA run on the mobile edge device: (a) Average latency per batch. (b) Continuous adaptation performance by TENT with diverse adaptation rates.

The latency issue in TTA algorithms stems from computationally intensive operations such as backpropagation, augmentations, teacher-student model structures, and model ensembling. These operations demand substantial time and memory resources, which can be problematic on mobile edge devices. Figure 1(a) shows that even lightweight algorithms such as TENT [2] demand almost four seconds per batch (16) adaptation cycle on real edge devices.

Furthermore, computational requirements escalate significantly for algorithms such as CoTTA [3] that update the entire parameters, making them impractical for real-time mobile applications. Another critical challenge arises from addressing the continuous adaptation requirements of rapid data streams from mobile devices. Existing TTA algorithms adapt to every input batch as illustrated in Figure 2, leading to a growing mismatch between adaptation speed and data stream rate. Consequently, the adapted model becomes increasingly outdated, resulting in accuracy degradation or failure to meet the desired real-time inference sample rate.

Our study reveals conventional adaptation strategies are ill-suited for real-time mobile applications due to significant latency overhead. To overcome this, we propose ***TESLA***: **T**ime-**E**fficient **S**parse and **L**ightweight **A**daptation strategy for real-time mobile applications, which carefully skips adaptation for specific batches. This counterintuitive yet effective approach strategically rests the adaptation process, significantly boosting the average inference speed. Additionally, by utilizing a lightweight memory to accumulate informative samples from non-adapted batches and update the model through them, our method can potentially maintain or even improve the model's accuracy in realistic settings. The combination of sparse adaptation and selective aggregation of informative samples strikes a balance between model accuracy and inference

MOBISYS '24, June 3–7, 2024, Minato-ku, Tokyo, Japan

ACM ISBN 979-8-4007-0581-6/24/06.
https://doi.org/10.1145/3643832.3661442

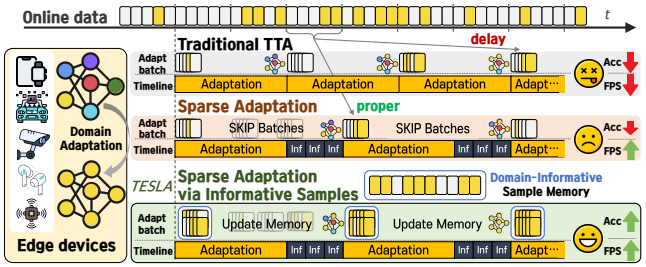

Figure 2: Design overview of *TESLA* (green) and comparison with the baselines. Yellow samples are the domain-informative ones that can effectively adapt the model.

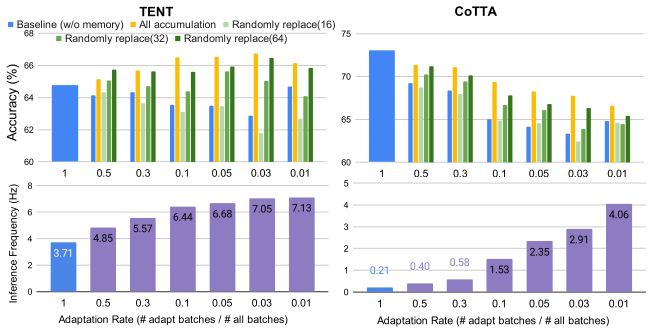

Figure 3: Performance (average accuracy and inference rate) variation across diverse adaptation rates. Additionally, it compares scenarios without (blue, baseline) and with (the others) memory.

speed, enabling efficient and effective TTA for resource-constrained real-time mobile applications.

2 DESIGN OF *TESLA*

As a practical scheme for real-time applications on edge devices, our proposed strategy seamlessly integrates with TTA algorithms that adapt to input data streams. The key idea of ***TESLA*** is strategically skipping adaptation for over 50% of the batches while maintaining competitive performance. This sparse adaptation approach significantly increases the inference sample rate, addressing the latency problem faced by existing methods that adapt to every batch.

Our intermittent adaptation procedure selectively adapts to batches based on predetermined criteria and policies. It sparsely adapts to meet real-time inference sample rate requirements while collecting domain-wise informative samples for effective adaptation. This strategy reduces overall latency and mitigates catastrophic forgetting by avoiding constant updates with potentially harmful samples. To leverage informative samples from non-adapted batches, we employ a memory-efficient mechanism to temporarily store and collectively adapt these samples when resources are available. This method benefits from valuable information without increasing adaptation frequency.

By strategically controlling the adaptation rate, our approach enables existing TTA methods to maintain competitive performance while significantly improving the inference sample rate, making TTA efficient and effective for latency-critical real-time applications. It complements lightweight, optimization-free algorithms and model compression techniques, further accelerating inference on edge devices.

3 EVALUATION

Experiments are conducted on a Raspberry Pi 4 device (Quad-core Cortex-A72 64-bit SoC @ 1.8GHz, 4GB RAM) using ResNet-18 with batch size 16 on CIFAR-10C dataset (Gaussian noise, severity level 5). Following Section 2, we employ a sparse adaptation scheme, updating the model with only a subset of batches based on a predefined adaptation rate (ADR) while freezing the model for the remaining batches.

Varying Adaptation Rates. We analyze the sparse adaptation performance, including average accuracy and inference frequency across different ADRs. As expected, higher ADRs increase accuracy but lower inference frequency, and vice versa (Figure 3). The remarkable point is the following: an extremely sparse ADR of 0.01 (updating once per 100 batches) achieves comparable accuracy to adapting 10 times more frequently, as seen in Figure 1(b). This plot demonstrates that from around the 300th batch onwards, ADR 0.01 consistently maintains competitive performance even against 1. These results suggest that intelligently skipping batches and selectively adapting to informative samples can effectively balance inference rate and accuracy.

Domain-Informative Sample Accumulation. Instead of adapting to entire batches or simply skipping, we explored accumulating samples from non-adapted batches in a memory buffer and collectively adapting the model to these stored samples. Figure 3 yellow bars show that adapting to all non-adapted batch data without specific sampling led to significant accuracy improvements over the baseline across ADRs. However, maintaining a large buffer is impractical for edge devices and risks including non-informative samples. We investigated memory sizes of 16, 32, and 64 samples with random replacement. Figure 3 green bars show employing modest memory sizes consistently outperformed no-memory, with larger sizes yielding more significant gains. Notably, for TENT, sparse adaptation with memory achieved even higher accuracy than adapting every batch (baseline), suggesting skipping updates can efficiently and effectively learn domains while mitigating overfitting to indifferent samples.

REFERENCES

[1] Quiñonero-Candela, Sugiyama, Schwaighofer, and Lawrence (Eds.). 2008. *Dataset Shift in Machine Learning*. The MIT Press. https://doi.org/10.7551/mitpress/9780262170055.001.0001

[2] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. 2020. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726* (2020).

[3] Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. 2022. Continual test-time domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7201–7211.