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

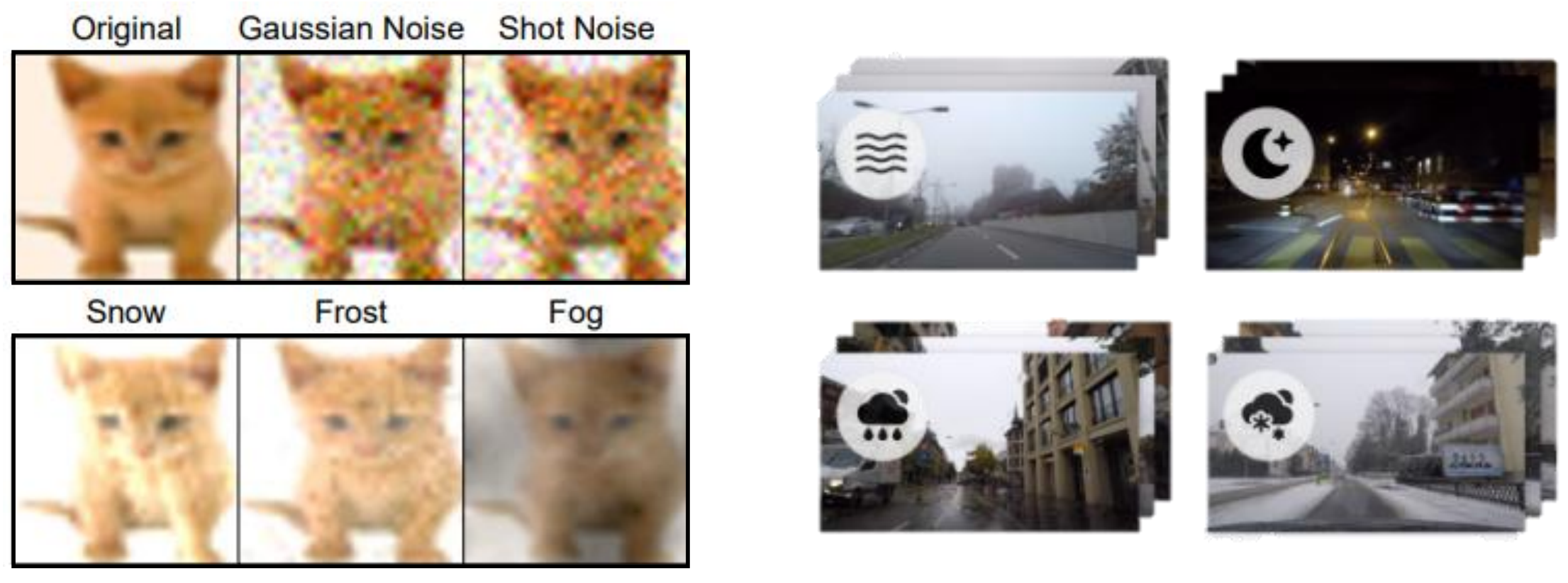
Hyeongheon Cha[1], Taesik Gong[2], Sung-Ju Lee[1]

[1]KAIST [2]Nokia Bell Labs

hyeongheon@kaist.ac.kr

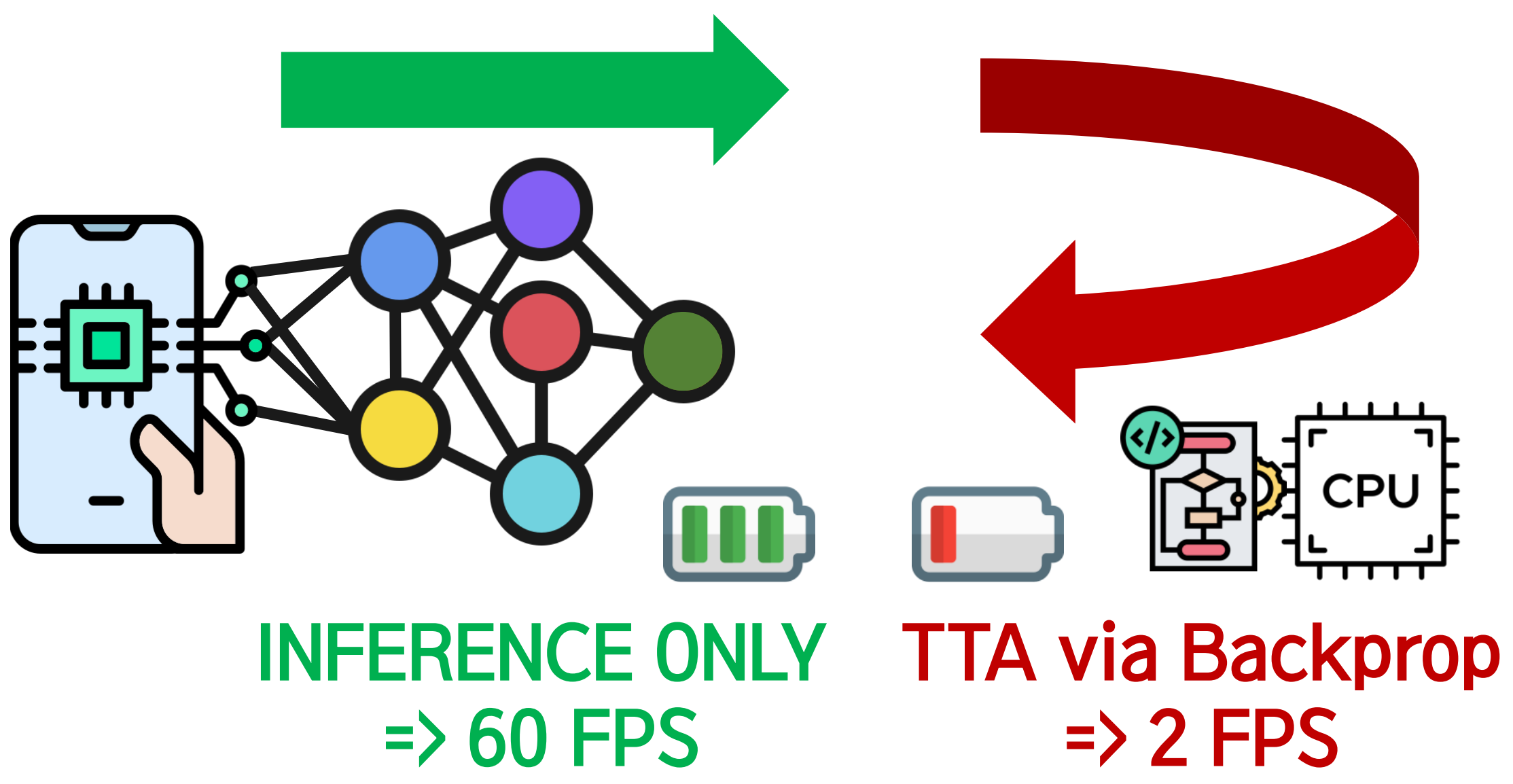
Background

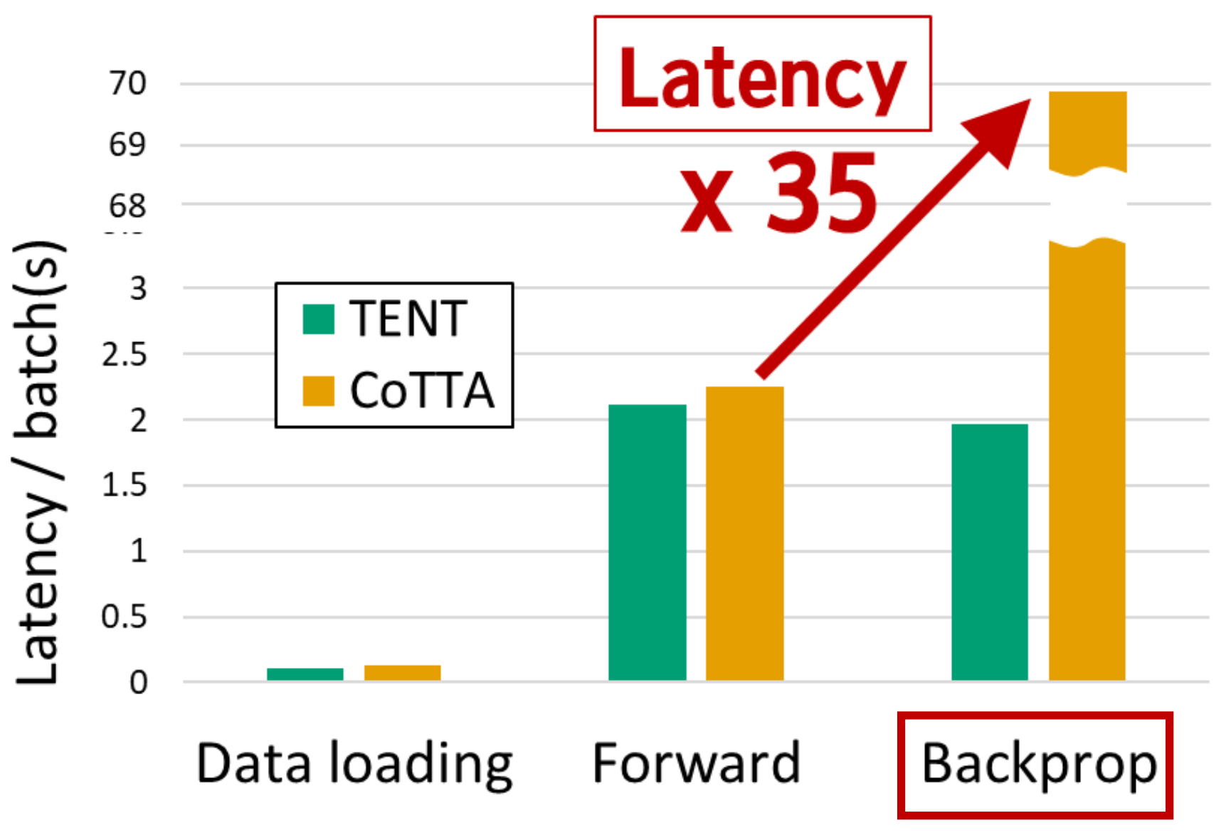
- Deep learning models on **mobile applications** often suffer from **domain shifts**
  - *Lighting changes / Sensor noises* resulting from different weather conditions or time



- **Test-Time Adaptation (TTA)** rapidly adapts models without **any source** or **labeled data**

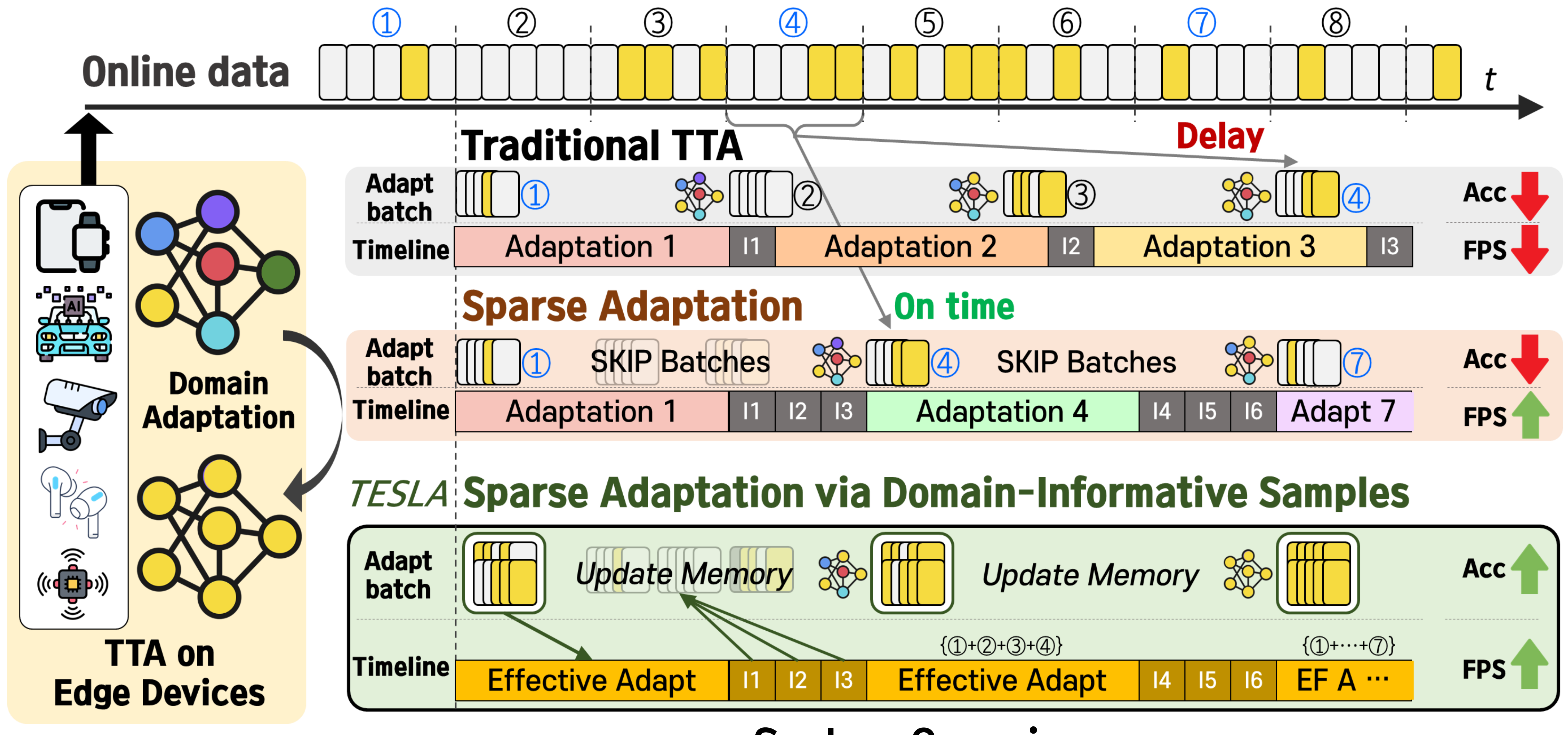
Time-Efficient TTA Suitable for Mobile App is Needed



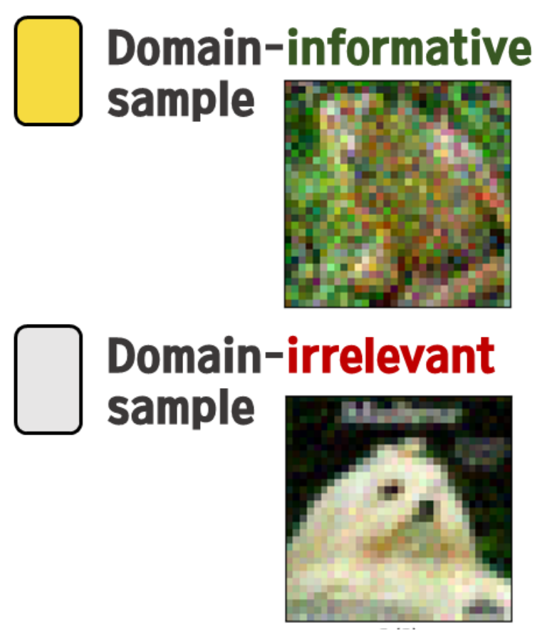


- **High latency**: Bottleneck in applying TTA to mobile device/scenario
  - Backpropagation, Augmentations, Teacher-Student Models …
- State-of-the-art TTA algorithms have been designed and evaluated mainly on **GPU** servers, focusing on **improving accuracy**

Sparse Adaptation Framework : Strategically Skip Batches and Effectively Update Model

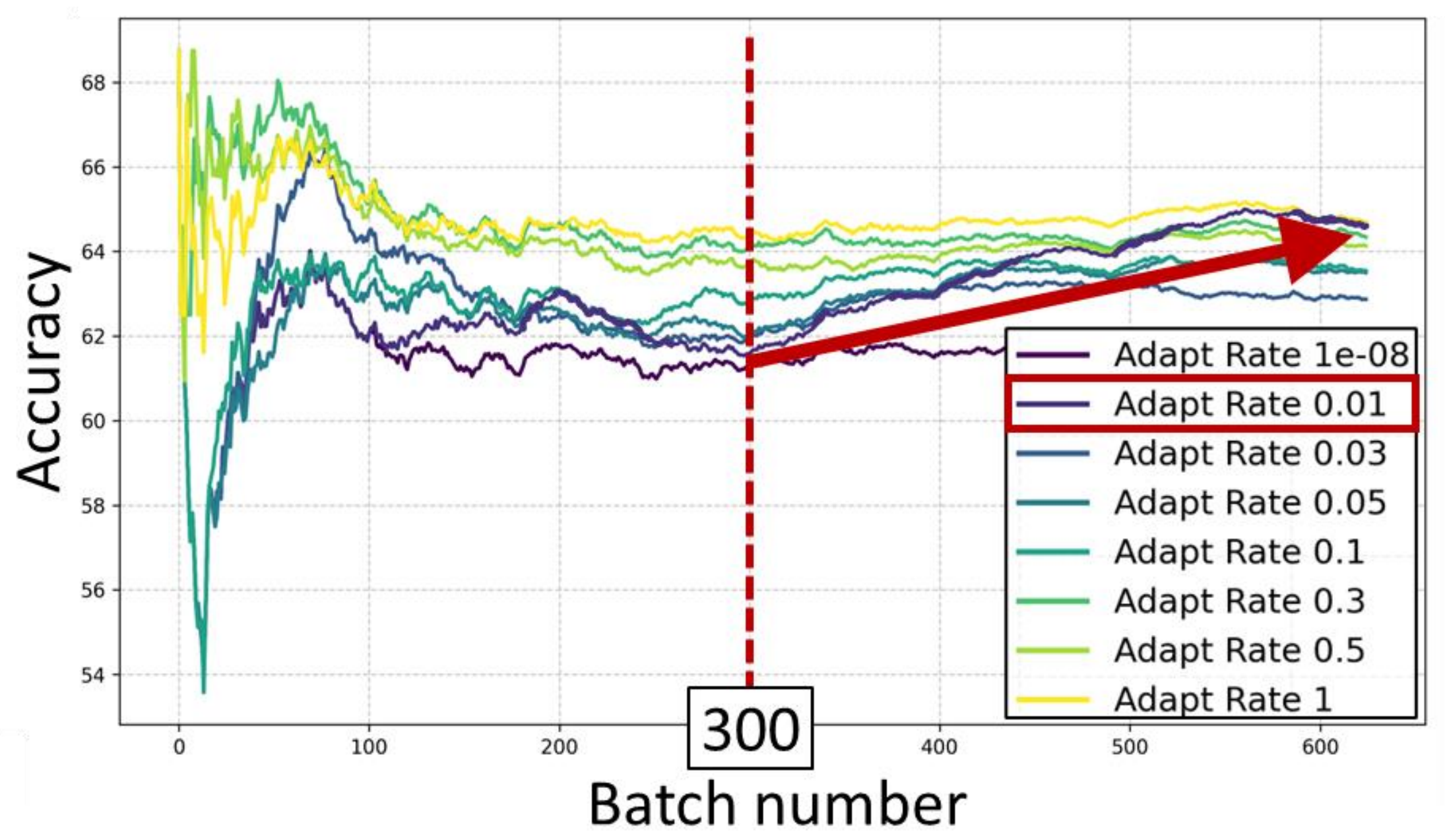


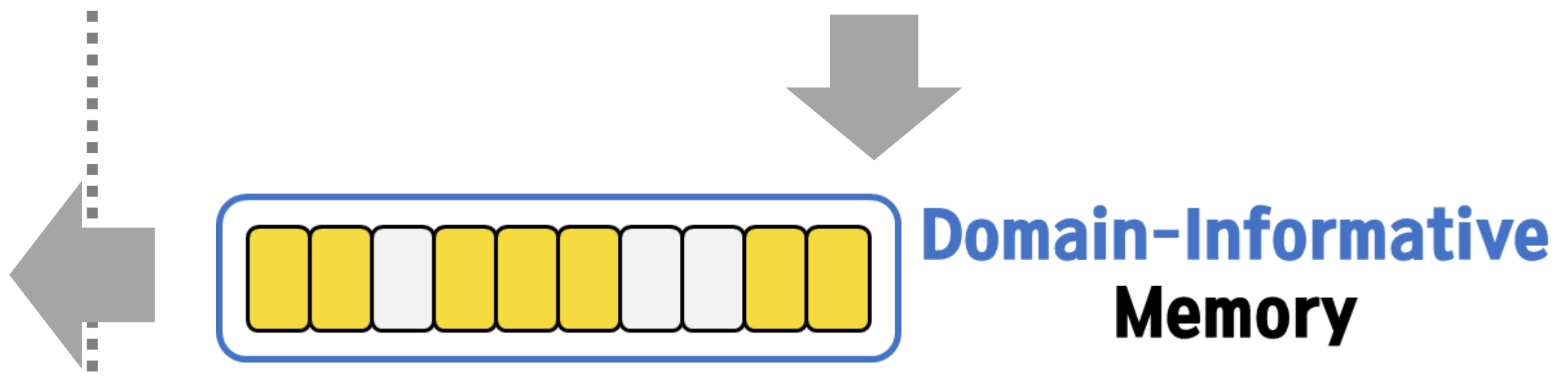
System Overview



- Sparse adaptation: **Skip adaptation** boldly and compensate for it by strong update via **domain-informative samples**
- By strategically controlling the **AD**aptation **R**ate (**ADR**), our system balances **inference fps** and **model accuracy**

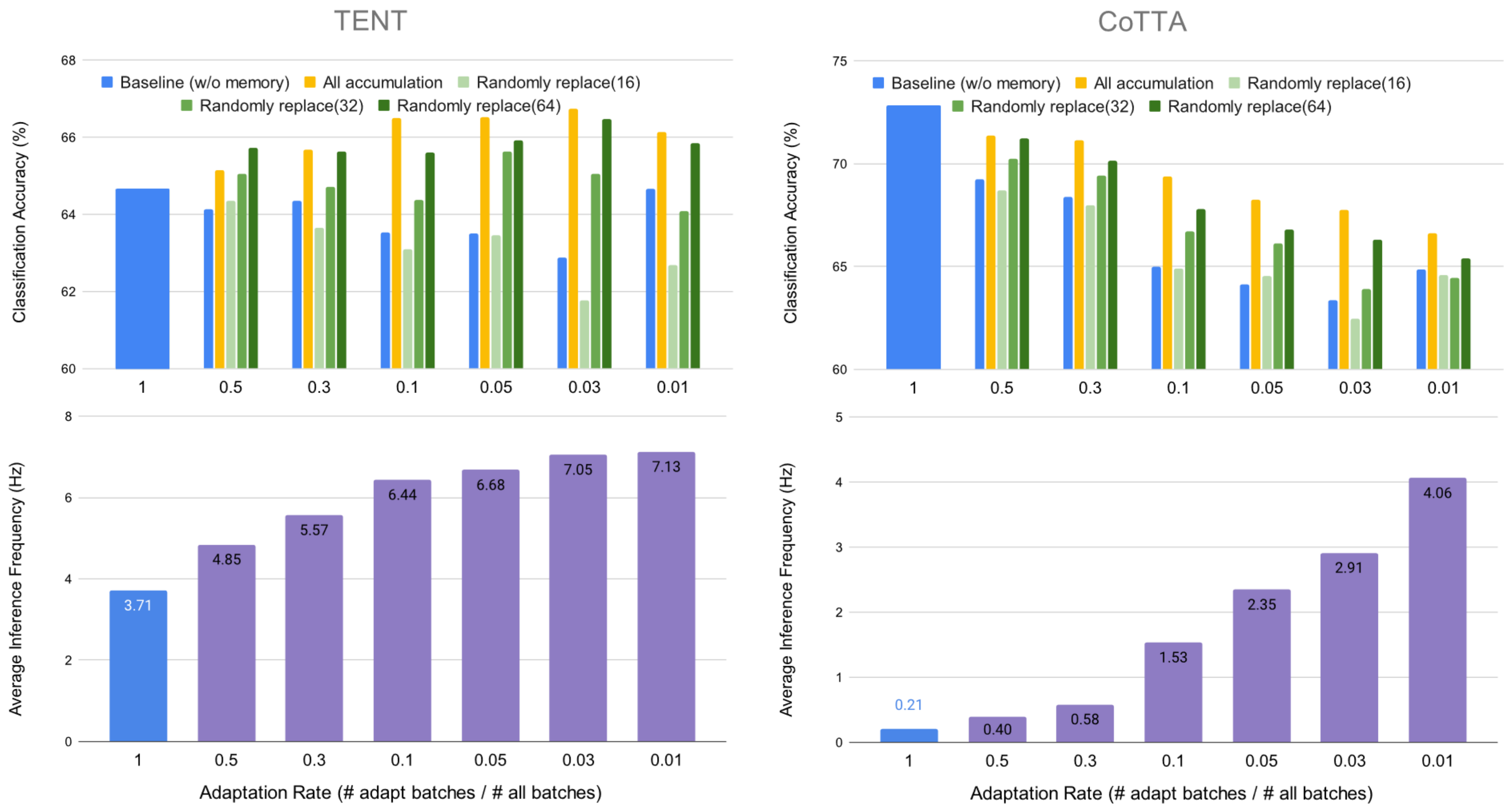
Motivation





- Extremely sparse **ADR of 0.01** can achieve competitive accuracy **even against 1**.
  ⇒ *Some samples can **greatly contribute** to domain adaptation loss.*

Experimental Results & Discussions



Performance (average accuracy and inference rate) variation across diverse *adaptation rates* – *sampling methods*

- [FPS] Average **inference speed** improves up to **20x**
- [Acc] Sparse adaptation with memory can achieve **even higher** accuracy than adapting every batch (baseline)
- **Seamlessly integrate** with existing lightweight adaptation and optimization algorithms, **further accelerating** inference across diverse mobile systems

TESLA: enabling efficient and effective TTA for resource-constrained real-time mobile applications

Future works

- **Memory** optimization: maintaining a **large buffer** is impractical
- **Balancing** FPS: still suffers from the periodical **bottleneck**
- Room for **Acc improvement**: sparse update-**aware** inference