Samsung Research | Università di Pisa | Università di Padova | ACL 2026 San Diego July 2–7

K-Merge: Online Continual Merging of Adapters for On-device Large Language Models

Donald Shenaj[1,2,3*] Ondrej Bohdal[1] Taha Ceritli[1] Mete Ozay[1] Pietro Zanuttigh[3] Umberto Michieli[1]

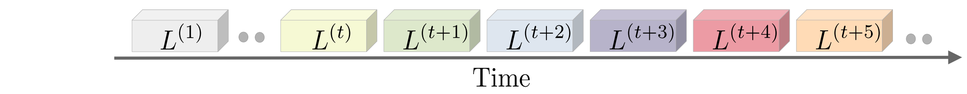
[1]Samsung R&D Institute UK [2]University of Pisa [3]University of Padova

*Research completed during internship at Samsung R&D Institute UK

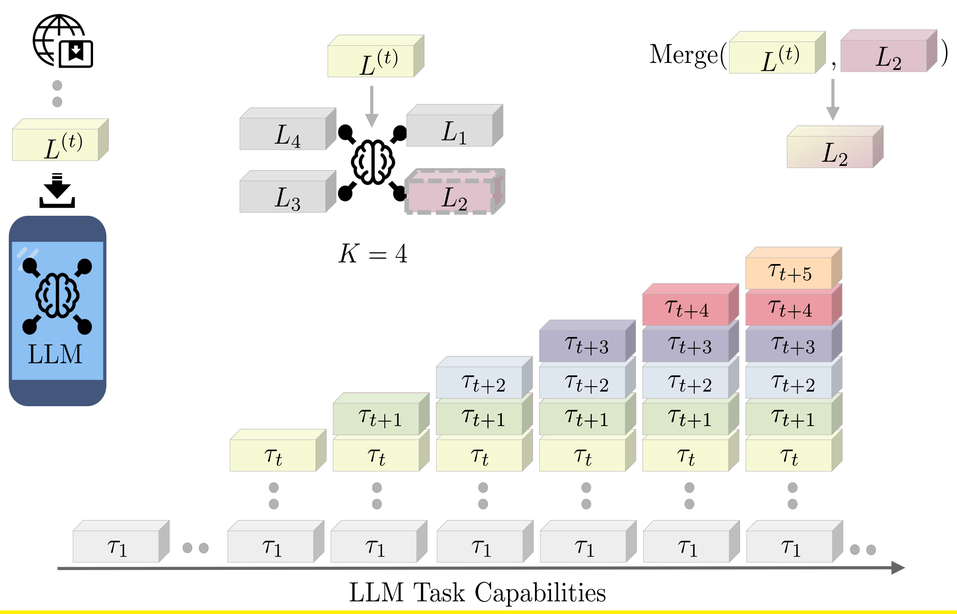
Motivation

- On-device LLMs rely on **Low-Rank Adapters (LoRAs)** to support diverse tasks under tight resource constraints.
- Mobile storage is limited - storing one LoRA per task (e.g. 40 tasks $\approx 1\,\text{GB}$) is infeasible.
- Existing **model merging** methods (Linear, DARE, TIES) assume *all* LoRAs are available simultaneously, but in practice, LoRAs arrive *incrementally* as users request new functionality.
- We formalise **online continual merging**: incorporate each new LoRA into a fixed-size collection $\mathcal{L}^{(t)}$ of size K, without access to previous training data.
- We propose a lightweight, data-free merging strategy that identifies adapters for merging using information from the history of merges and employs dynamic weighting to balance old and new capabilities.

Online Continual Merging



1. Downloads new adapters over time.
2. Selects the most similar stored adapter to the new one.
3. Update the selected stored adapter by merging it with the new adapter.

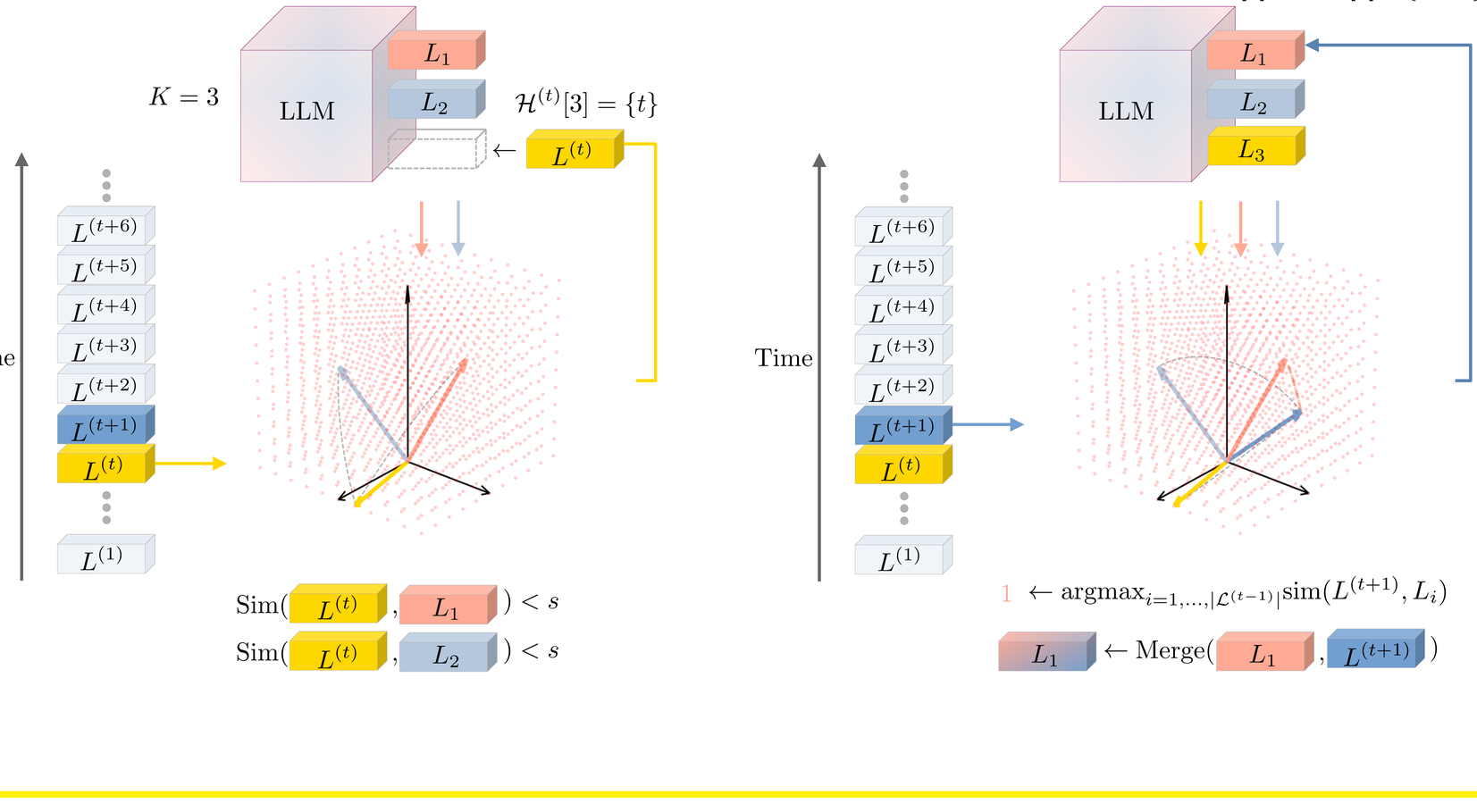


K-Merge and K-Merge^{++}

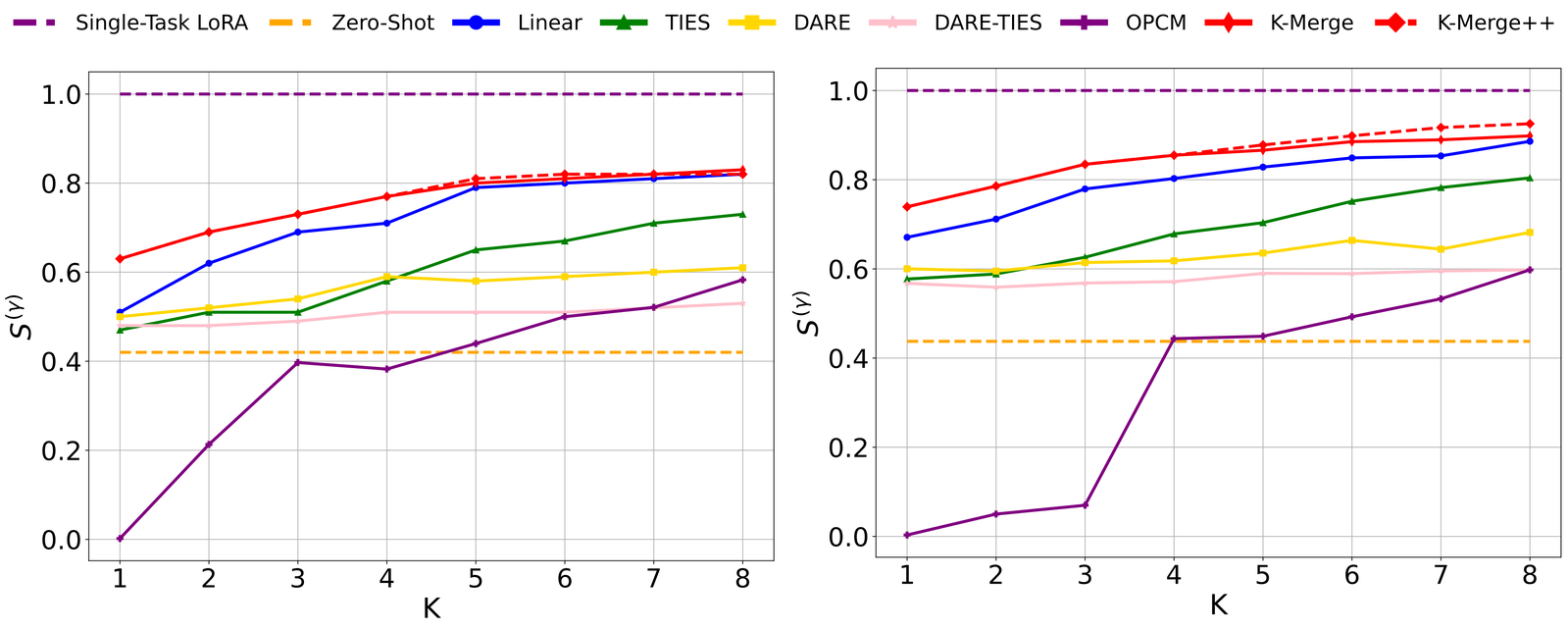
Parameters: budget K , threshold s

Initialize on-device collection by $\mathcal{L}^{(0)} = \emptyset$.
Initialize history by $\mathcal{H}^{(0)} = \emptyset$.
while new LoRA $L^{(t)}$ received, $t > 0$ **do**
  if $|\mathcal{L}^{(t-1)}| > 0$ **then**
    $c \leftarrow \text{argmax}_i \, \text{sim}(L^{(t)}, L_i)$
  end if
  if $|\mathcal{L}^{(t-1)}| = K$
  or $(|\mathcal{L}^{(t-1)}| > 0$ **and** $\text{sim}(L^{(t)}, L_c) \geq s)$
  then
    # merge branch
    $\mathcal{L}^{(t)} \leftarrow \{\mathcal{L}^{(t-1)} \setminus L_c\} \cup \{\text{merge}(L_c, L^{(t)})\}$
    $\text{merge}(L_c, L^{(t)}) = \frac{L^{(t)} + |\mathcal{H}^{(t-1)}[c]| \cdot L_c}{|\mathcal{H}^{(t-1)}[c]| + 1}$
    $\mathcal{H}^{(t)}[c] \leftarrow \mathcal{H}^{(t-1)}[c] \cup \{t\}$
  else
    # store branch $(|\mathcal{L}^{(t-1)}| < K)$
    $\mathcal{L}^{(t)} \leftarrow \mathcal{L}^{(t-1)} \cup \{L^{(t)}\}$
    $\mathcal{H}^{(t)}[|\mathcal{L}^{(t-1)}| + 1] \leftarrow \{t\}$
  end if
end while
return $\mathcal{L}^{(t)}$

K-Merge^{++} only unshaded = both variants



Quantitative Results



References

[1] Hu et al., "LoRA: Low-Rank Adaptation of Large Language Models", *ICLR* 2022.

[2] Yadav et al., "Ties-merging: Resolving interference when merging models", *NeurIPS* 2024.

[3] Yu et al., "Language models are super mario: Absorbing abilities from homologous models as a free lunch" *ICML* 2024.

[4] Tang et al., "Merging on the fly without retraining: A sequential approach to scalable continual model merging", *NeurIPS* 2025.

Paper and Code