LANCE: Low Rank Activation Compression for Efficient On-Device Continual Learning

Marco Apolinario[1,2] and Kaushik Roy[2]
[1]Delft University of Technology (TU Delft), [2]Purdue University

ECCV 2026 | MALMÖ | SEPT 8–12

Motivation

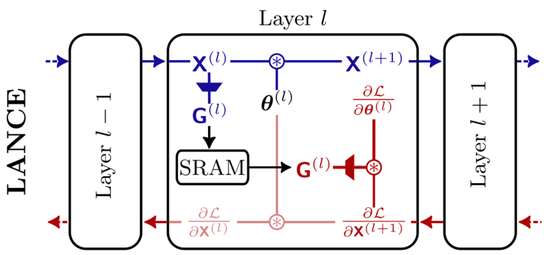
Personalization, privacy, and long-term adaptation all require models that keep learning after deployment — fine-tuning to new data and acquiring new tasks without forgetting old ones.

- Storing activations for backpropagation dominates memory — **5–10× more than parameters** — yet microcontrollers offer only a few hundred KB of SRAM.
- Existing low-rank methods **recompute an SVD at every step**: large compute overhead, and no path to continual learning.

Can one fixed activation subspace serve all of training — and all tasks?

Method: LANCE

LANCE calibrates a fixed low-rank activation subspace once, then reuses it for the rest of training — eliminating repeated decompositions.

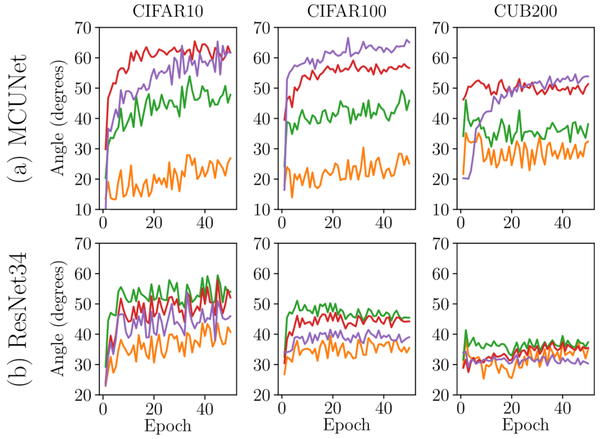


- **One-shot HOSVD.** Factor matrices $\{U_i\}$ are computed once from N calibration batches (N = 2 already suffices), then frozen.
- **Store the core, not the tensor.** Only the compact core G is kept in SRAM; the frozen factors are reused in the backward pass.
- **No repeated decomposition** — memory and FLOPs both fall, with no specialised hardware.
- **Continual learning by construction.** For each new task, mode-d factors are drawn from the **null space** of the stored subspace memory M — orthogonal to past tasks, with no replay buffer and no large task-specific matrices.

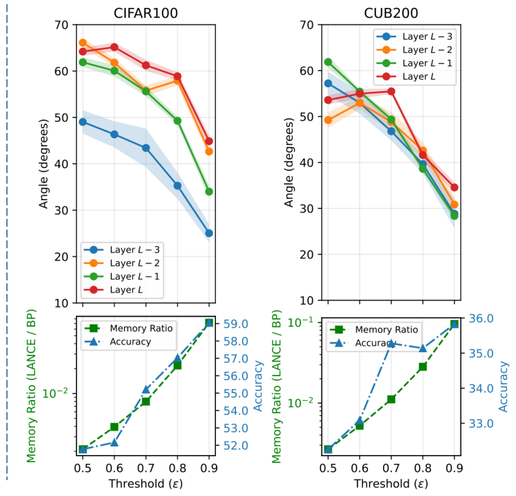
Results — and the Questions They Answer

Q1 · Does one-shot calibration preserve gradient fidelity?

Yes — LANCE gradients stay within ~70° of full backpropagation throughout training, and stabilise as training progresses, even under two orders of magnitude of compression.



Across multiple datasets and architectures, LANCE gradients stay within ~70° of full BP and stabilize as training progresses.



As ε drops, alignment degrades smoothly — even at two orders of magnitude less memory, LANCE stays within ~70° of full BP.

Q2 · How much memory does it save?

Activation storage drops **25× to 380×** versus BP depending on architecture, while accuracy stays within ~1–2% of full BP on residual networks.

Fine-tuning MCUNet and ResNet34 pretrained on ImageNet-1k

		CIFAR10		CIFAR100	
Model	Method	Acc	MB	Acc	MB
MCUNet	BP	82.91	17.57	58.38	17.57
	LANCE	76.23	0.09	52.59	0.09
ResNet34	BP	94.37	49.00	77.50	49.00
	LANCE	93.56	1.01	75.94	1.02

380×
peak activation memory reduction (MCUNet / MobileNetV2)

≤2%
accuracy gap vs. full BP on standard residual networks

Q3 · Is it competitive for continual learning?

Yes — on par with full-rank gradient-projection methods at up to **10× lower memory**, with near-zero forgetting.

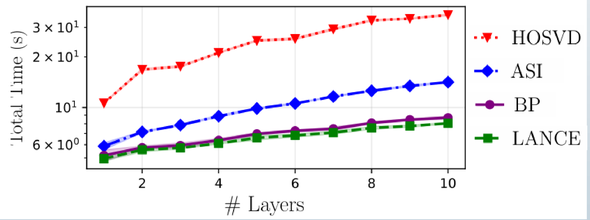
10×+
less activation memory than full-rank gradient-projection CL baselines (GPM, CODE-CL)

Continual Learning on 10-task Split CIFAR-100 and 20-task Split MiniImageNet

	Split CIFAR-100			Split MiniImageNet		
Method	ACC%	BWT	MB	ACC%	BWT	MB
Multitask	79.58	–	–	69.46	–	–
GPM	72.48	−0.9	22.27	60.41	−0.9	127.37
CODE-CL	77.21	−1.1	33.80	71.16	−1.1	283.82
LANCE	71.52	−0.17	2.32	59.68	−0.99	32.96

Edge runtime — Raspberry Pi 3B+ (MCUNet, CIFAR-10, batch 128)

LANCE achieves the **lowest forward, backward and total training time** across 1–10 trainable layers. Relative to LANCE: BP is **3–9% slower**, ASI **19–76% slower**, iterative HOSVD **2.1×–4.4× slower** — one-shot calibration removes the decomposition overhead entirely.



Conclusions

- A **single fixed low-rank subspace** is sufficient for both fine-tuning and continual learning.
- **25×–380×** less activation memory, no repeated decompositions, no extra hardware cost.
- Accuracy within **~2%** of full BP on ResNet-18/34, and the result scales to ImageNet-1k.
- Enable CL by learning new tasks in the null space of previous ones.

Future Directions

- Deployment on true microcontroller-class targets (≤ 256 KB SRAM) with quantised core tensors.
- Adaptive, per-layer energy thresholds and re-calibration under distribution shift.
- Beyond convolutional vision: transformers, sequence models and multi-modal backbones.

Acknowledgements

Supported by the Center for Co-design of Cognitive Systems (CoCoSys), a JUMP 2.0 center (SRC), and the U.S. Department of Energy.

CODE