THE KERNEL REALITY CHECK

Benchmarking and Distilling Efficient Attention at Scale

Fırat Öncel · Cem Subakan · Mirco Ravanelli · Çağatay Yıldız
Concordia University / Mila – Quebec AI Institute / Université Laval / University of Tübingen

TL;DR:
Near-softmax quality is achievable. Beating FlashAttention's efficiency *is* the challenge.

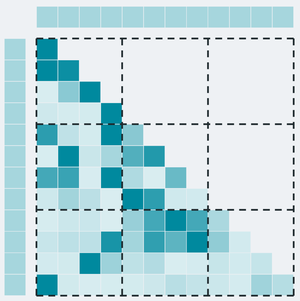
THE QUESTION

Does kernelized attention really help?

$O(N)$ on paper. But does it beat Flash-Attention in time, memory and accuracy?

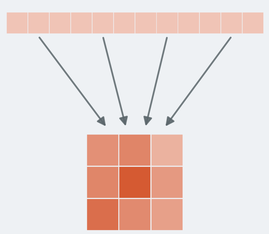
BACKGROUND

Two ways to compute attention



Softmax / FlashAttention

✗ $N \times N$ scores
✗ grow with N^2
✓ FlashAttention: computed in tiles, never stored, $O(N^2)$ compute



$$S_i = S_{i-1} + \phi(k_i)\, v_i^{\top}$$

Kernelized (linear)

✓ $d \times d$ state
✓ fixed size
✓ updated once per token [3]

OUR EXPERIMENT SETUP

A fair comparison

- ✓ Models: **ViT-S/B/L/G, GPT2-S/M/L**
- ✓ Measured: **time + peak GPU memory**
- ✓ Same **dtype, batch, GPU** (A100)
- ✓ Tuned kernels: **FlashAttention-2 [2], FLA [7]**
- ✓ Same **distillation protocol** for all

01 / EFFICIENCY

FlashAttention is more efficient in training and non-causal inference.

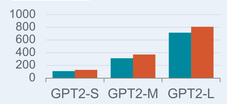
Linear attention's clearest win: causal LLM inference memory.

	ViT	GPT2
Training forward + backward	Flash faster + leaner	Flash faster + leaner
Prefill whole input at once	Flash faster + leaner	time: Flash memory: Linear
Decode one token per step	n/a	Linear 5–6× less memory

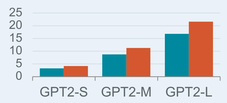
Exact numbers (GPT2)

Flash · Best kernelized

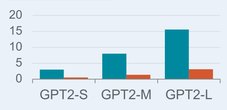
Train time — Flash 1.1–1.2×



Train mem. — Flash 1.3×



Decode mem. — Linear 5.0–5.8×



REPORTED GAINS DO NOT PRODUCE

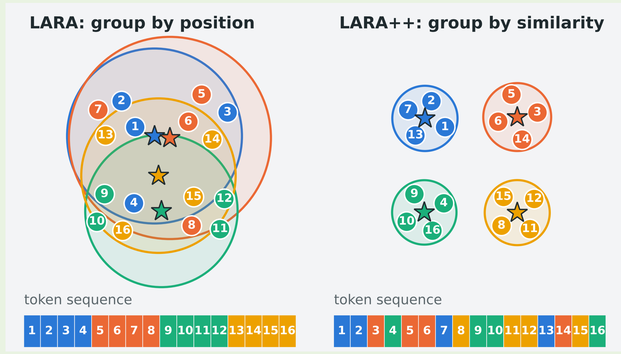
Against FlashAttention-2, Hedgehog prefill is **3.5× slower**, not the reported 5× faster. Cached Flash decoding throughput is as flat as DiJiang's.

02 / OUR METHOD

LARA++

LARA [6] writes softmax attention as an **expectation** and estimates it with **importance sampling**:

$$\mathrm{Attn}(\mathbf{q}_n) \approx \frac{\sum_{s=1}^{S} w_s\, f_n(\omega_s)}{\sum_{s=1}^{S} w_s}, \qquad \omega_s \sim q(\omega)$$



Our improvements:

1. **Clustered proposals:** group queries by similarity, not position (greedy k-center [10]).
2. **Deterministic features:** use proposal means directly; no sampling noise.

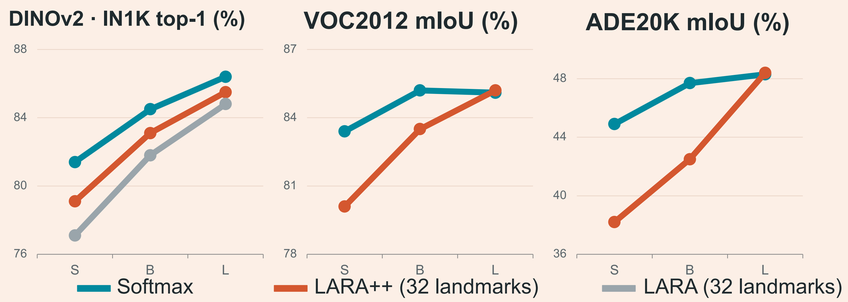
03 / APPROXIMATING SOFTMAX BY KERNELIZED ATTENTION

The quality gap is not intrinsic to kernelization.

LARA++ beats every kernelized baseline, and recovers up to 99% of softmax accuracy; but at a compute cost.

DeiT distillation, ImageNet top-1

Method	S/224	B/224	B/384
Softmax	79.8	81.8	82.9
Hedgehog [4]	−2.1	−2.1	−2.5
DiJiang [5]	−7.9	−7.3	−10.1
LARA [6]	−2.4	−1.8	−1.9
LARA++ (ours)	**−1.3**	**−1.0**	**−0.8**



BUT: THE COST REMAINS

Slowest kernelized method: 1.2–1.3× training step time.

Clustering takes 43–60% of the forward pass.

C = N/k landmarks: $O(N^2/k)$ clustering cost.

References
[1] Dao et al. FlashAttention. NeurIPS 2022.
[2] Dao. FlashAttention-2. ICLR 2024.
[3] Katharopoulos et al. Transformers are RNNs. ICML 2020.
[4] Zhang et al. The Hedgehog & the Porcupine. ICLR 2024.
[5] Chen et al. DiJiang: compact kernelization. ICML 2024.
[6] Zheng et al. LARA: linear randomized attention. ICML 2022.
[7] Yang & Zhang. FLA: flash-linear-attention. 2024.
[8] Touvron et al. DeiT. ICML 2021.
[9] Oquab et al. DINOv2. TMLR 2024.
[10] Sener & Savarese. Core-set (k-center). ICLR 2018.