IndexCache: Accelerating Sparse Attention via **Cross-Layer Index Reuse**

Yushi Bai[1†], Qian Dong[1†], Ting Jiang[2], Xin Lv[2]

Zhengxiao Du[2], Aohan Zeng[12], Jie Tang[1], Juanzi Li[1]

[1]Tsinghua University [2]Z.ai

Presenter : Ruibo Liu, Ouxiang Zhou @ USTC

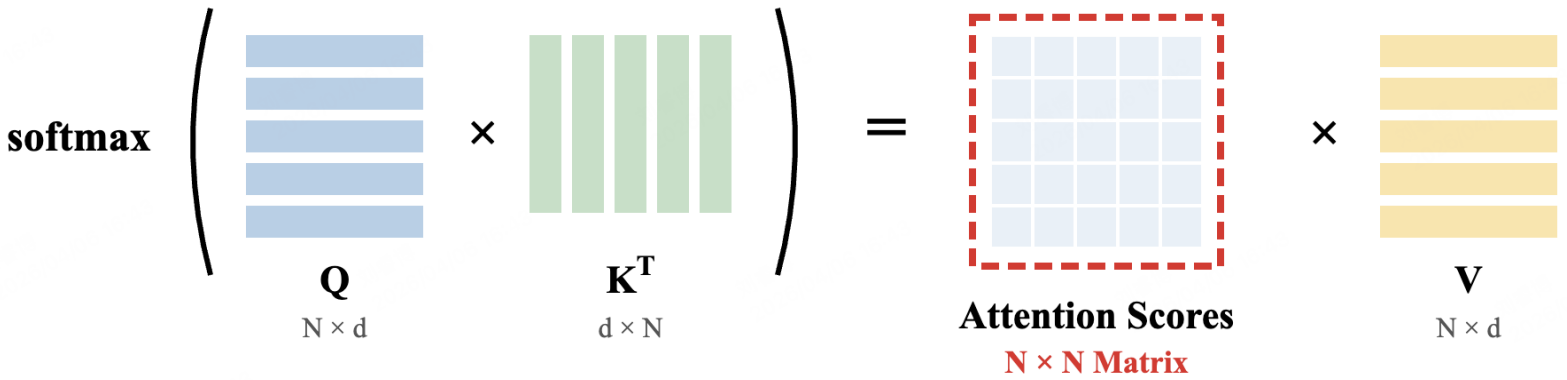
Outline

- **Background & Motivation**
  - DeepSeek Sparse Attention (DSA) & Lightning Indexer
  - High similarity of Cross-Layer Top-k Index
- **Design**
  - Main idea : Full & Shared Layer
  - Training-Free & Training-Aware IndexCache
- **Evaluation**
- **Discussion**

Background – DeepSeek Sparse Attention

❑Self Attention

❖Every Q attends to all previous tokens → $O(N^2)$ compute & memory



❑Sparse Attention

❖Each query selects only the most relevant subset instead of attending to all preceding tokens

❖DeepSeek Sparse Attention, or **DSA**

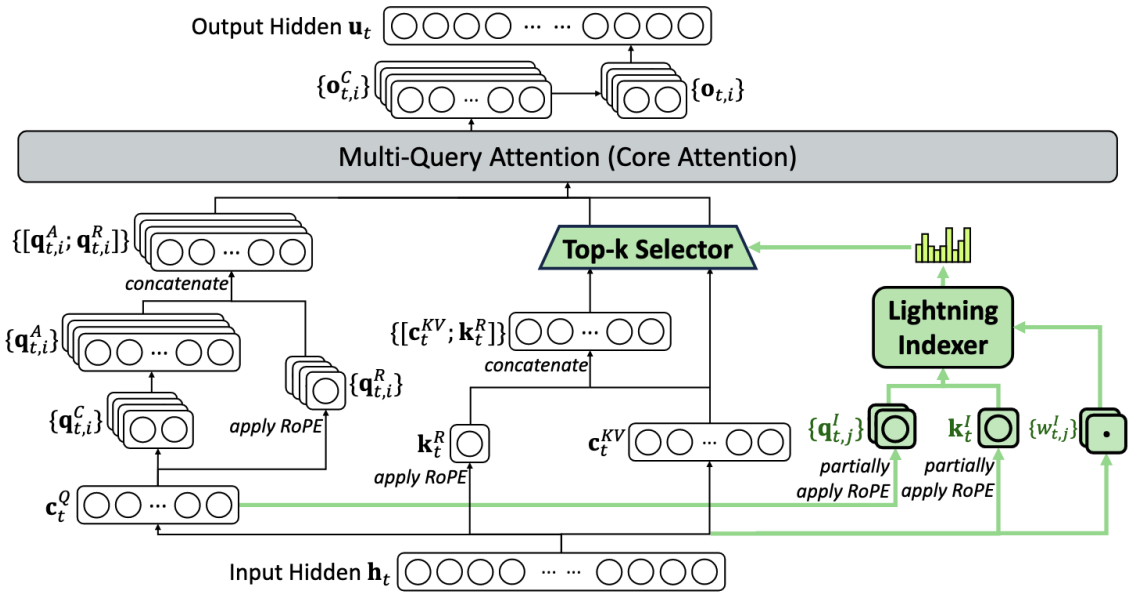
Background – DSA & Lightning Indexer

❑DeepSeek Sparse Attention

- ❖Pick k most important tokens based on index score approximation

❑Lightning Indexer

- ❖a lightweight **auxiliary model**
  - ➢ **FP8 Quantization**
  - ➢ **ReLU** Activation Function
  - ➢ **Low-Rank projections**
- ❖Attn: $O(L^2)$->$O(kL)$, $k << L$
- ❖Indexer: $O(\alpha L^2)$ $\alpha << 1$

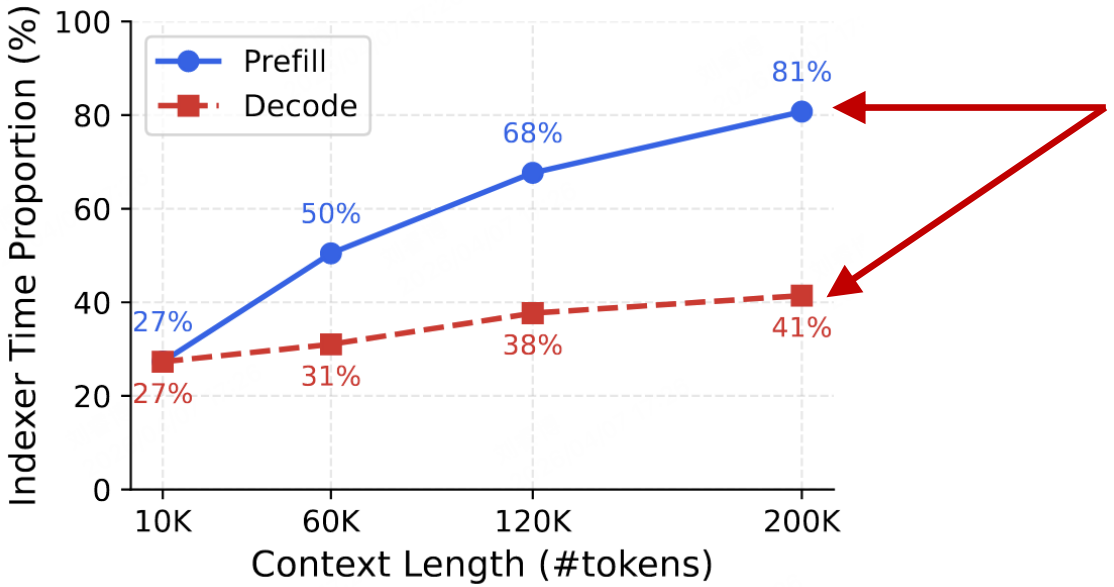


Motivation & Challenge

❑Motivation:

❖Necessity of all N per-layer indexer computations

- ➢ Although cheaper per-FLOP than the main attention computation, its total cost across N layers is O(NL^2)
- ➢ Becoming a significant fraction of the total attention budget at long context



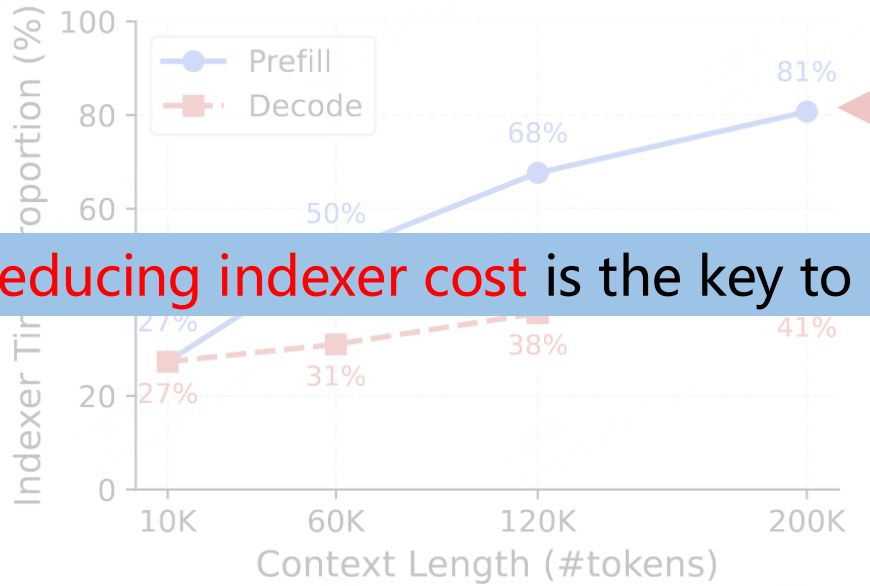
The indexer's share of total latency rises sharply with context length, particularly during the prefill stage

Motivation & Challenge

❑Motivation:

❖Necessity of all N per-layer indexer computations

- ➢Although cheaper per-FLOP than the main attention computation, its total cost across N layers is O(αNL^2)
- ➢Becoming a significant fraction of the total attention budget at long context



The indexer's share of total latency rises sharply with context length, particularly during the prefill stage

Reducing indexer cost is the key to accelerating long-context DSA inference

Motivation & Challenge

- **Goals:**
  - Remove the majority of indexers in DSA
  - Let most layers reuse top-k indices from retained indexer layers
  - Retain quality

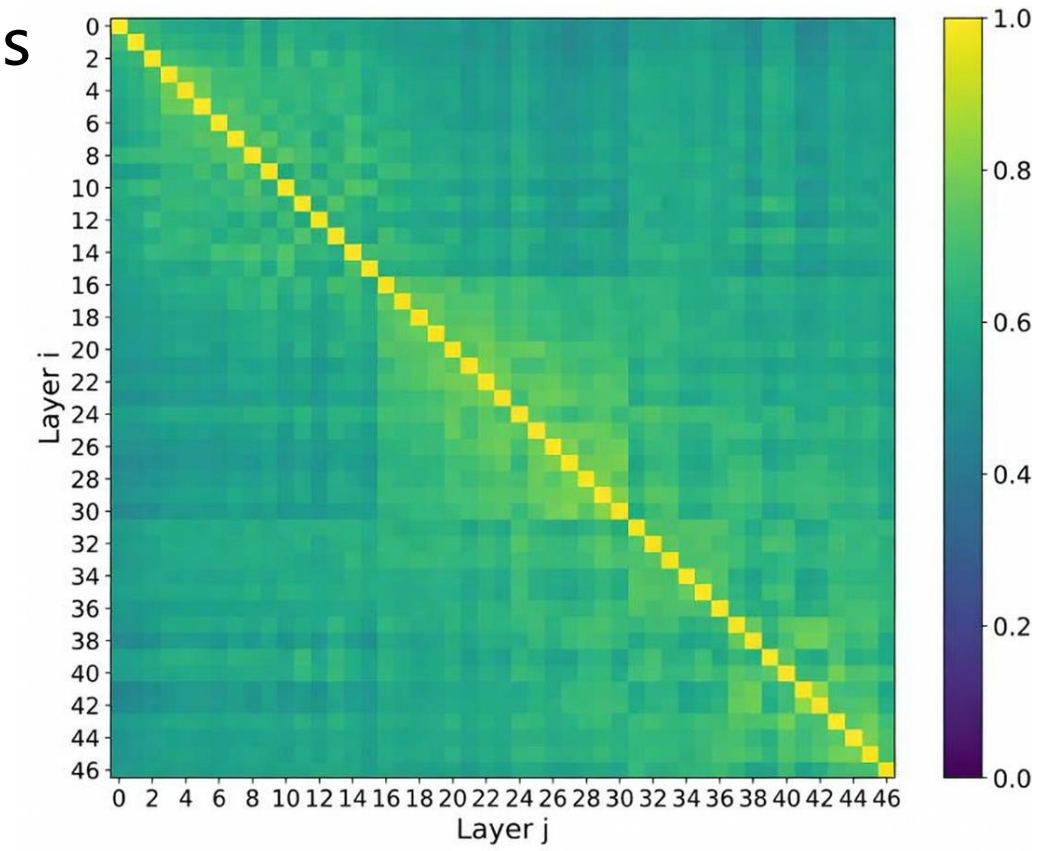
Motivation & Challenge

- **Goals:**
  - Remove the majority of indexers in DSA
  - Let most layers reuse top-k indices from retained indexer layers
  - Retain quality
- **Challenges:**
  - Stability across consecutive transformer layers in DSA

Motivation & Challenge

❑Insight:

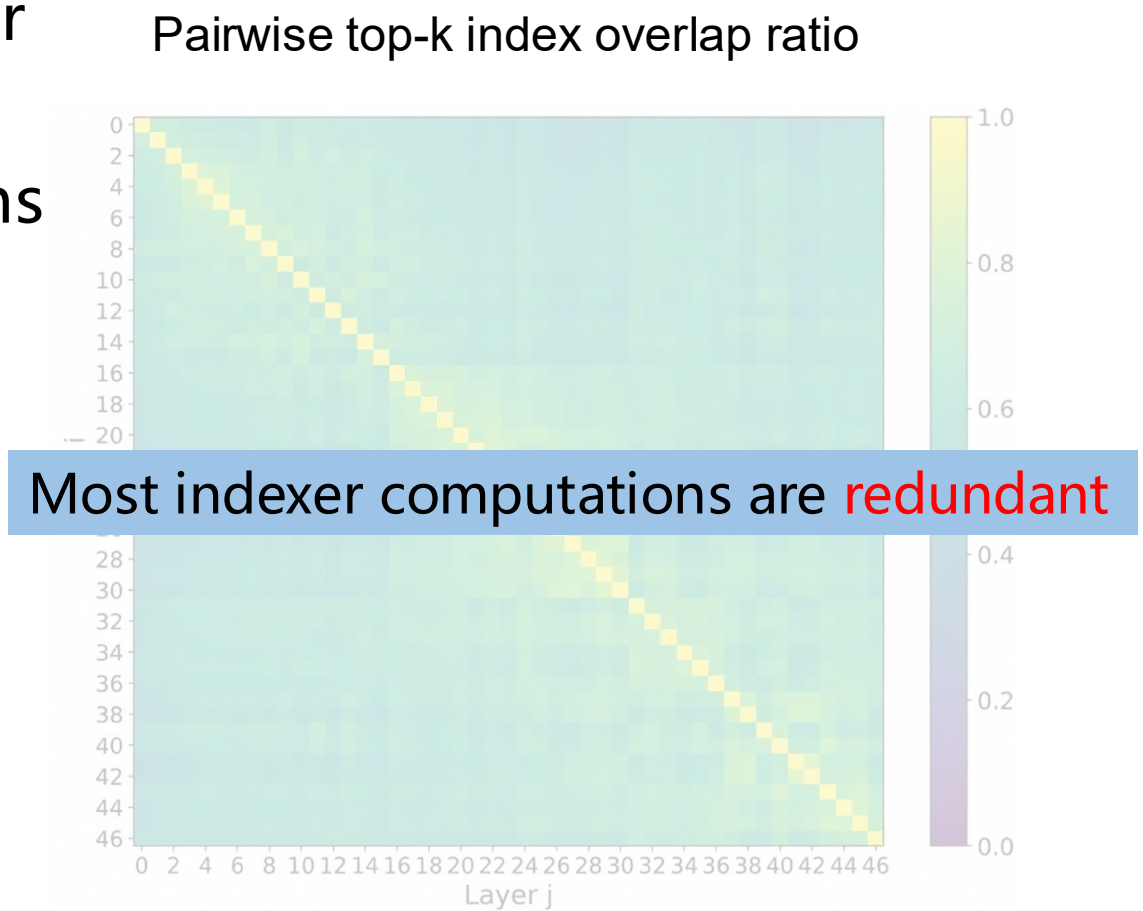
- ❖Model：GLM-4.7-30B
- ❖High correlation of top-k selections across consecutive layers
  - High overlap near the diagonal
  - Block structure
  - Uneven decay
  - Early-late distinction



Motivation & Challenge

❑Motivation:

- ❖Necessity of all N per-layer indexer computations
- ❖High correlation of top-k selections across consecutive layers
  - High overlap near the diagonal
  - Block structure
  - Uneven decay
  - Early-late distinction



Most indexer computations are redundant

Motivation & Challenge

❑Goals:

- ❖Remove the majority of indexers in DSA
- ❖Let most layers reuse top-k indices from retained indexer layers
- ❖Retain quality

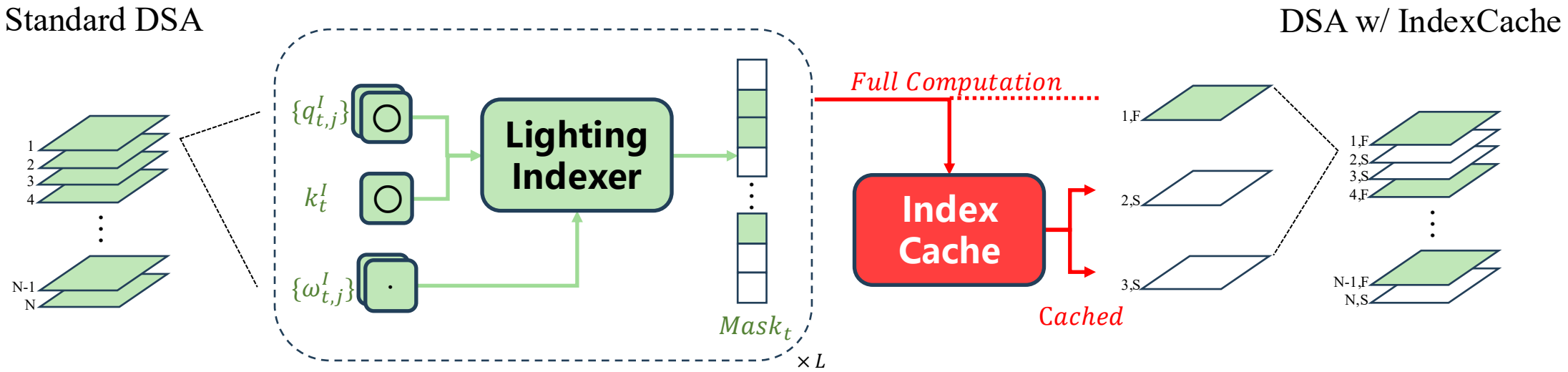
❑Challenges:

- ~~❖Stability across consecutive transformer layers in DSA~~
- ❖Algorithm to distinguish whether to reuse
- ❖Maximum reuse ratio achievable before quality degrades
- ❖Model adaptation to close the performance gap introduced by aggressive index reuse

Design: IndexCache

❑IndexCache: Reuse top-K masks across layers

- ❖F layers perform the full Indexer topK score computation
- ❖S layers inherit the index set from the nearest preceding F layer



Design: IndexCache

❑DSA Inference

- ❖Introduce only one branch statement

❑Complexity

- ❖$O(NL^2)$ total indexer cost eliminated
- ❖O(NLk) core attention unchanged

❑Key point

- ❖Achieve the best pattern

$$\mathbf{c}=\{c_l \in \{F,S\} \mid l = 1,2,\ldots,N\}$$

(a) Standard DSA Inference

Require: Input $\mathbf{X}$, layers $1 \ldots N$

for $\ell = 1$ **to** N **do**
  $\mathbf{I}^{(\ell)} \leftarrow \text{INDEXER}_\ell(\mathbf{X})$
  $\mathcal{T}^{(\ell)} \leftarrow \text{Top-k}(\mathbf{I}^{(\ell)})$
  $\mathbf{X} \leftarrow \text{SPARSEATTN}_\ell(\mathbf{X}, \mathcal{T}^{(\ell)})$
  $\mathbf{X} \leftarrow \text{FFN}_\ell(\mathbf{X})$ ▷ + norm, residual, etc.
end for

(b) IndexCache Inference

Require: Input $\mathbf{X}$, layers $1 \ldots N$, pattern $\mathbf{c}$

for $\ell = 1$ **to** N **do**
  if $c_\ell = \text{F}$ **then**
    $\mathbf{I}^{(\ell)} \leftarrow \text{INDEXER}_\ell(\mathbf{X})$
    $\mathcal{T}^{(\ell)} \leftarrow \text{Top-k}(\mathbf{I}^{(\ell)})$
    $\mathcal{T}_{\text{cache}} \leftarrow \mathcal{T}^{(\ell)}$
  else $\{c_\ell = \text{S}\}$
    $\mathcal{T}^{(\ell)} \leftarrow \mathcal{T}_{\text{cache}}$ ▷ reuse
  end if
  $\mathbf{X} \leftarrow \text{SPARSEATTN}_\ell(\mathbf{X}, \mathcal{T}^{(\ell)})$
  $\mathbf{X} \leftarrow \text{FFN}_\ell(\mathbf{X})$ ▷ + norm, residual, etc.
end for

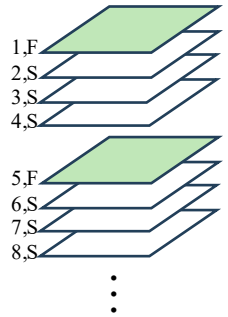
IndexCache: Training-Free

❑Notions:

- ❖Input: Pretrained DSA model
- ❖Output: Pattern **c** with a given ratio of S layers
- ❖Goal: Minimize $LM\ loss$

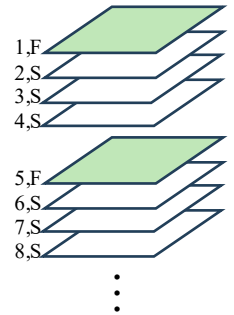
❑Uniform interleaving strategy

- ❖Retain every r-th layer's indexer and skip the rest
- ❖e.g. **c** = { FSSSFSSS. . . } for r = 4



IndexCache: Training-Free

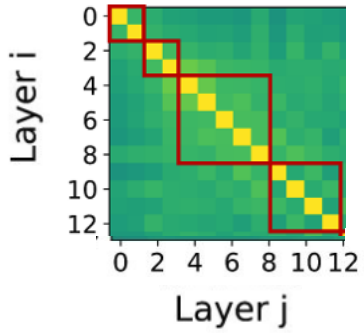
❑Uniform interleaving strategy

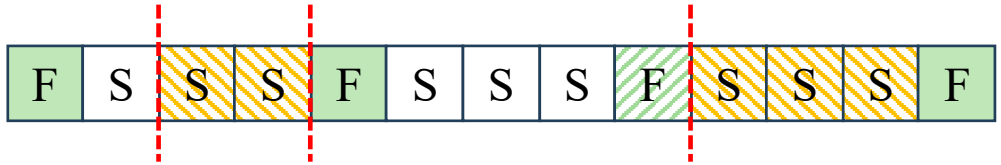


- ❖Retain every r-th layer's indexer and skip the rest
- ❖e.g. **c** = { FSSSFSSS… } for r = 4

❑Sub-optimality

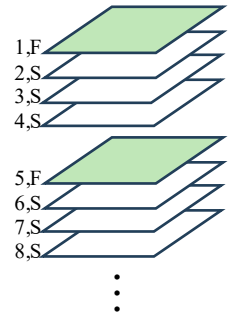
- ❖Indexer importance varies significantly across layers
- ❖Remove a critical indexer while retaining a redundant one





IndexCache: Training-Free

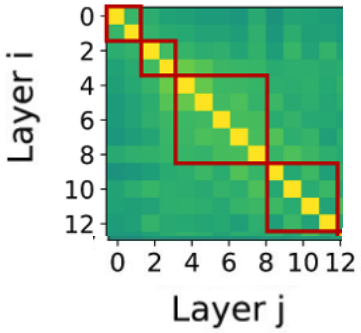
❑Uniform interleaving strategy

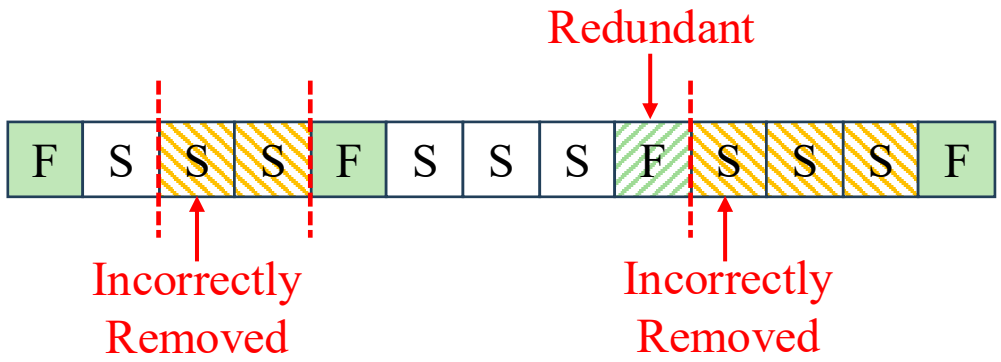


- ❖Retain every r-th layer's indexer and skip the rest
- ❖e.g. **c** = { FSSSFSSS… } for r = 4

❑Sub-optimality

- ❖Indexer importance varies significantly across layers
- ❖Remove a critical indexer while retaining a redundant one

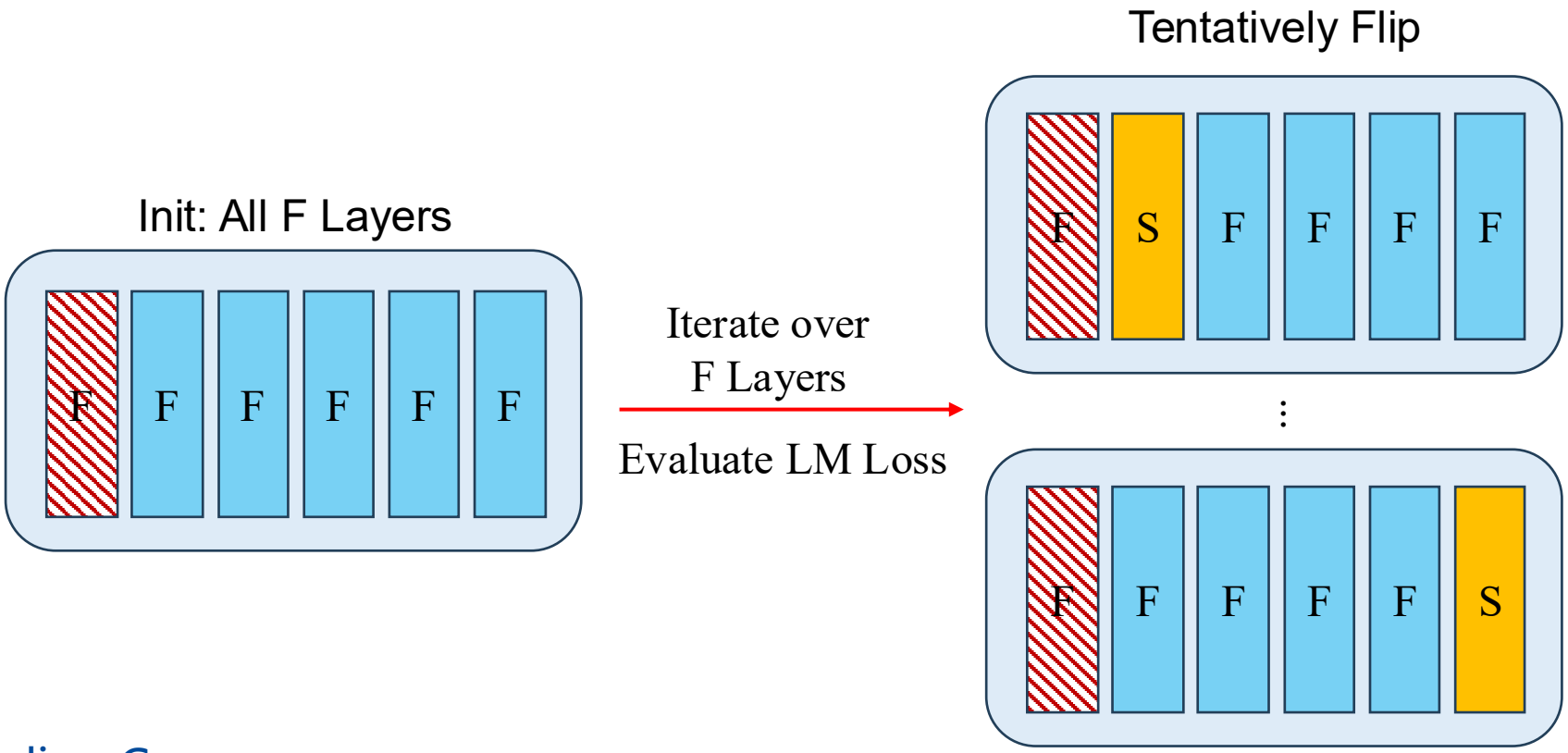




IndexCache: Training-Free

❑Layer Selection Algorithm

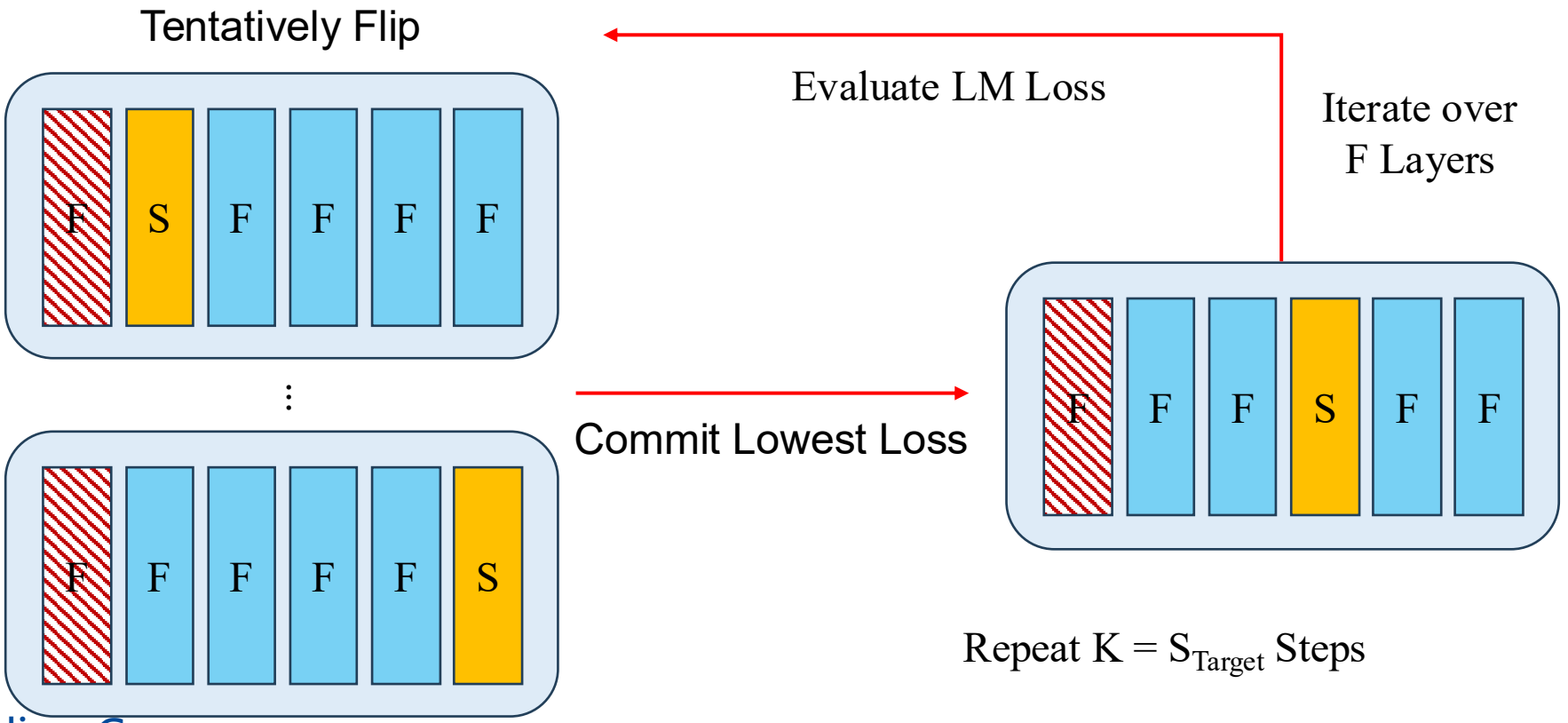
- ❖Greedy search to convert F Layers to S ones
- ❖Using B mini-batches cached from the training data as calibration set



IndexCache: Training-Free

❑Layer Selection Algorithm

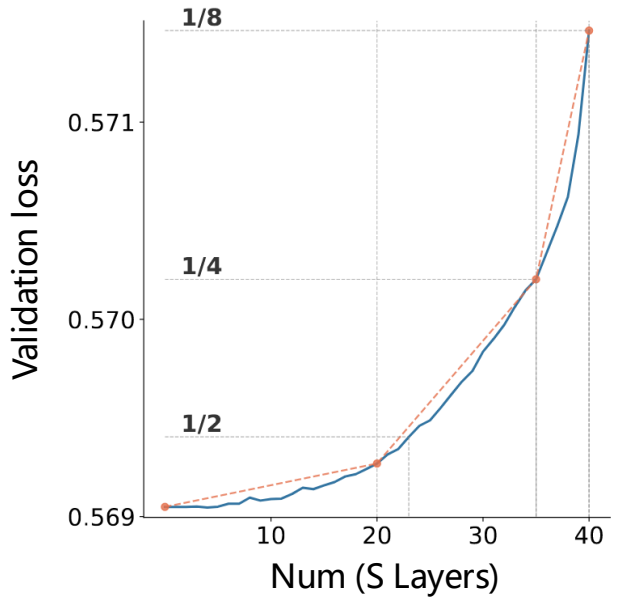
- ❖Greedy search to convert F Layers to S ones
- ❖Using B mini-batches cached from the training data as calibration set



IndexCache: Training-Free

❑Layer Selection Algorithm

- ❖Complexity
  - ➢ $O(N^2/P)$ forward passes with **P** stage Pipeline Parallelism
- ❖Properties
  - ➢ Outperforming uniform interleaving at the same retention ratio
  - ➢ A natural ordering of indexer importance
  - ➢ Stable results across different calibration sets



IndexCache: Training-Aware

❑Standard DSA Training

❖Distilled via KL divergence against aggregated attention distribution

$$\mathcal{L}^{\mathrm{I}} = \sum_t D_{\mathrm{KL}}\left(\mathbf{p}_t^{(\ell)} \,\|\, \mathbf{q}_t^{(\ell)}\right)$$

➢ $p_t^{(l)} = \frac{1}{H}\Sigma_{h=1}^{H} a_{t,h}^{(l)}$: Aggregated attention distribution

➢ $q_t^{(l)} = Softmax(I_t^{(l)})$: Indexer's output distribution

❖Each indexer was originally trained to serve only its own layer

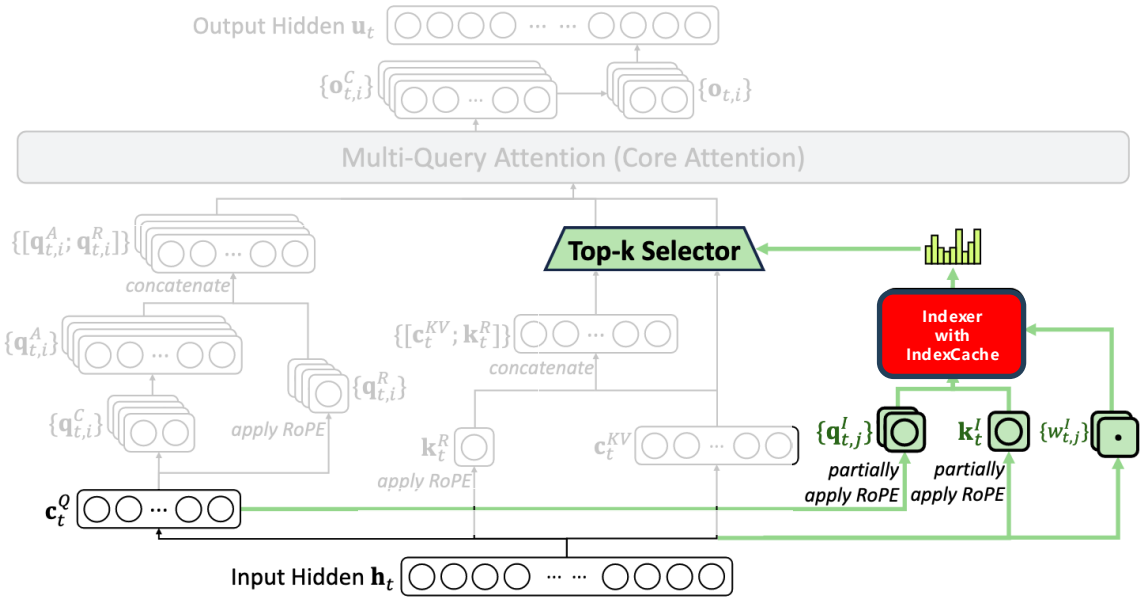
❑Serve multiple layers simultaneously

IndexCache: Training-Aware

❑Multi-layer distillation

$$\mathcal{L}^{\mathrm{I}}_{\mathrm{multi}} = \sum_{j=0}^{m} \frac{1}{m+1} \sum_{t} D_{\mathrm{KL}}\left(\mathbf{p}_t^{(\ell+j)} \,\|\, \mathbf{q}_t^{(\ell)}\right)$$

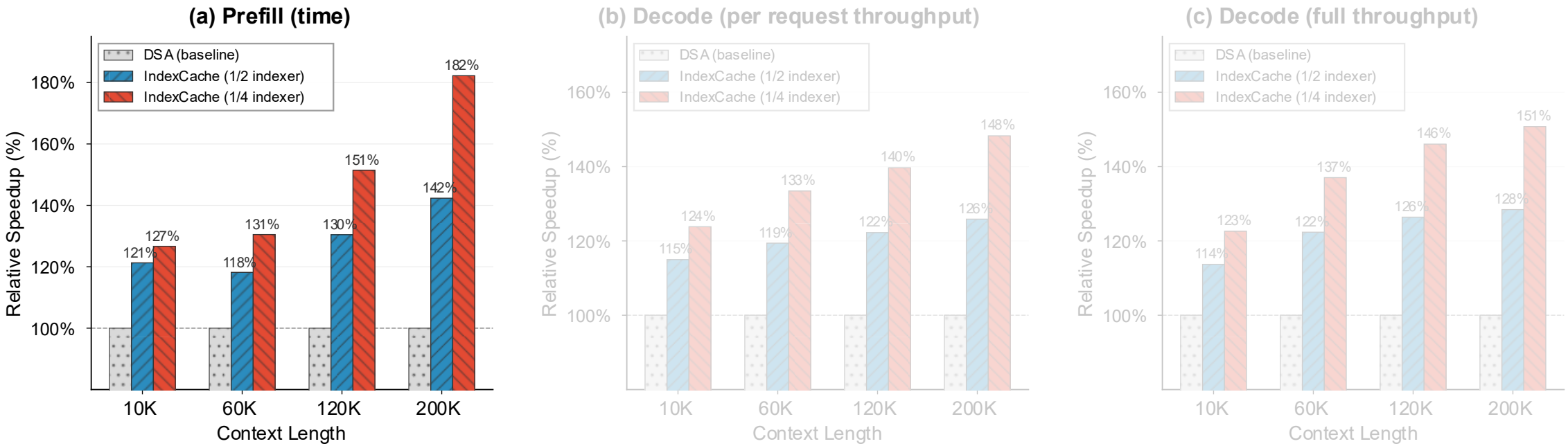
- ❖Predict a top-k set that is jointly useful for all layers it serves
  - ➢ ℓ: a retained F layer with m following S layers reuse its index set
- ❖Training: Identical to standard DSA
  - ➢Warm-up phase
  - ➢Sparse-training phase



Evaluation: Experimental Setup

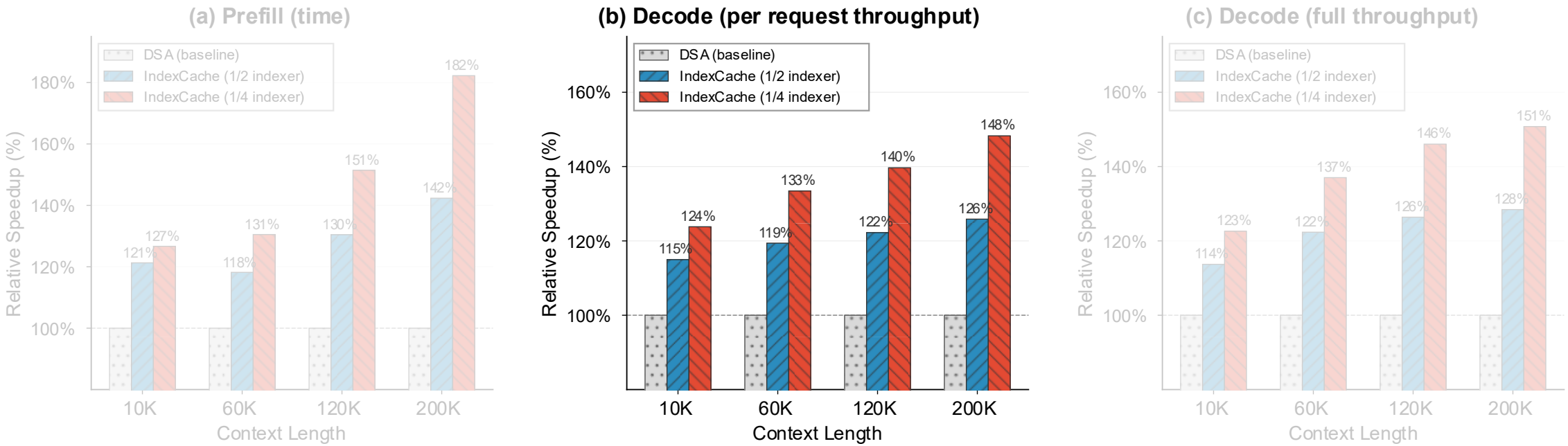
- **Base Model:** GLM-4.7-Flash MoE (30B-A3B) with DSA
- **Context Length:** 200k max
- **Benchmarks:**
  - 5 Long-Context Tasks, e.g., MRCR v2, RULER, LongBench v2
  - 4 General & Reasoning Tasks, e.g., AIME 2025, LiveCodeBench
- **Hardware:** NVIDIA H100 nodes

Evaluation: Inference Speedup



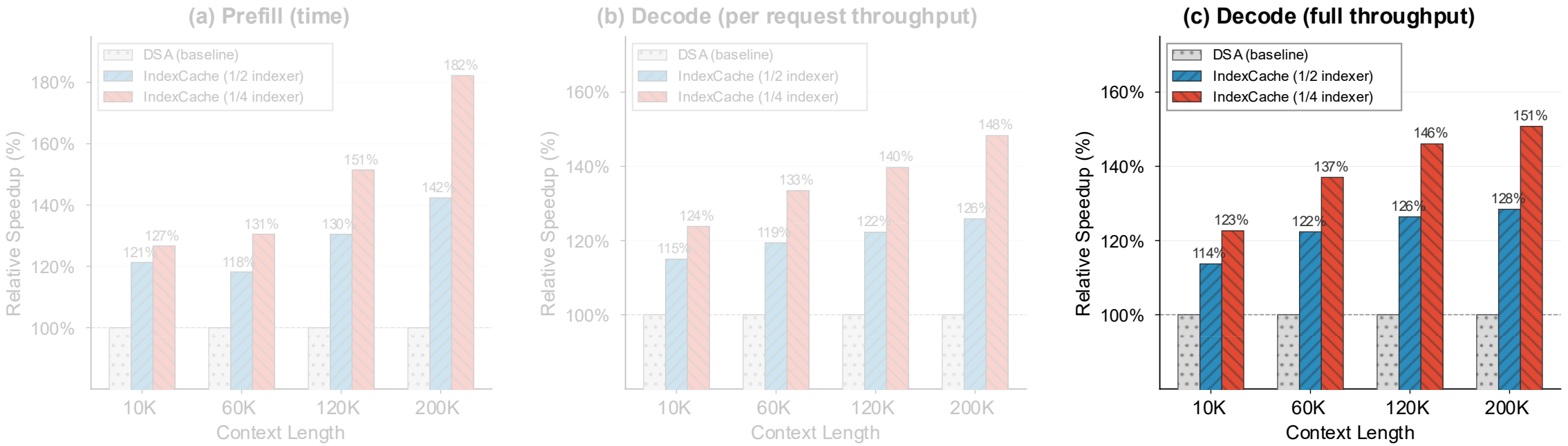
Prefill : At 200K tokens, IndexCache (1/4) achieves a 1.82× speedup

Evaluation: Inference Speedup



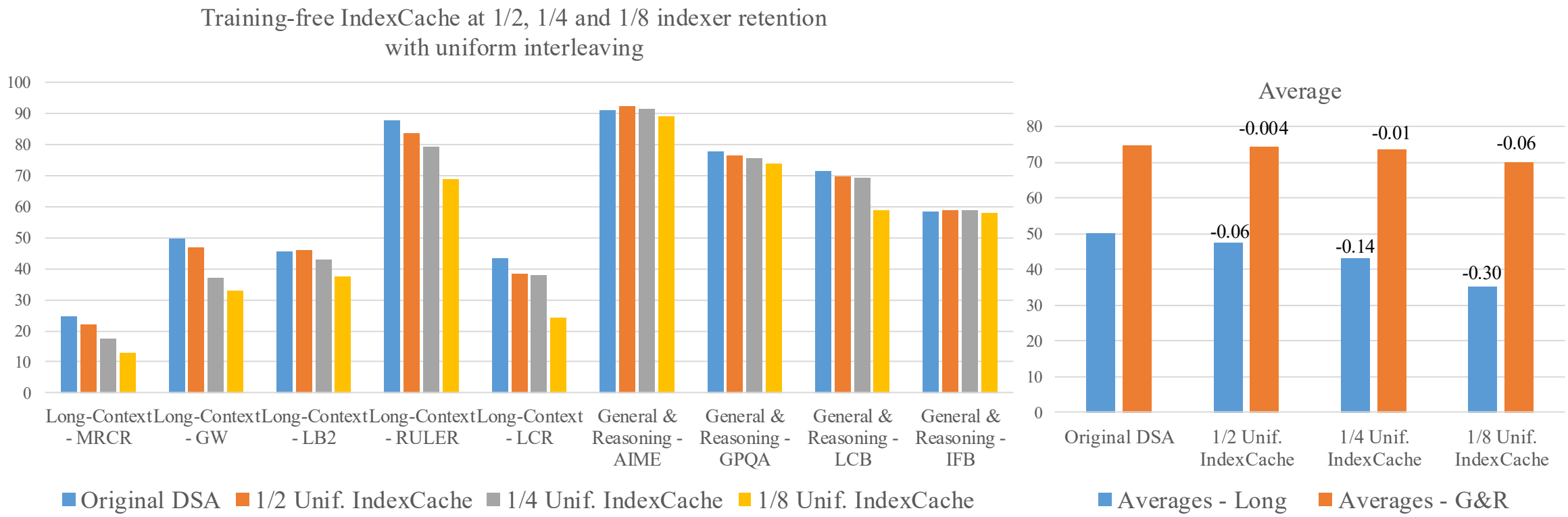
Decode : At 200K, DSA’ s decode speed is 58 tok/s, while IndexCache (1/4) achieves 86 tok/s, a 1.48× speedup

Evaluation: Inference Speedup



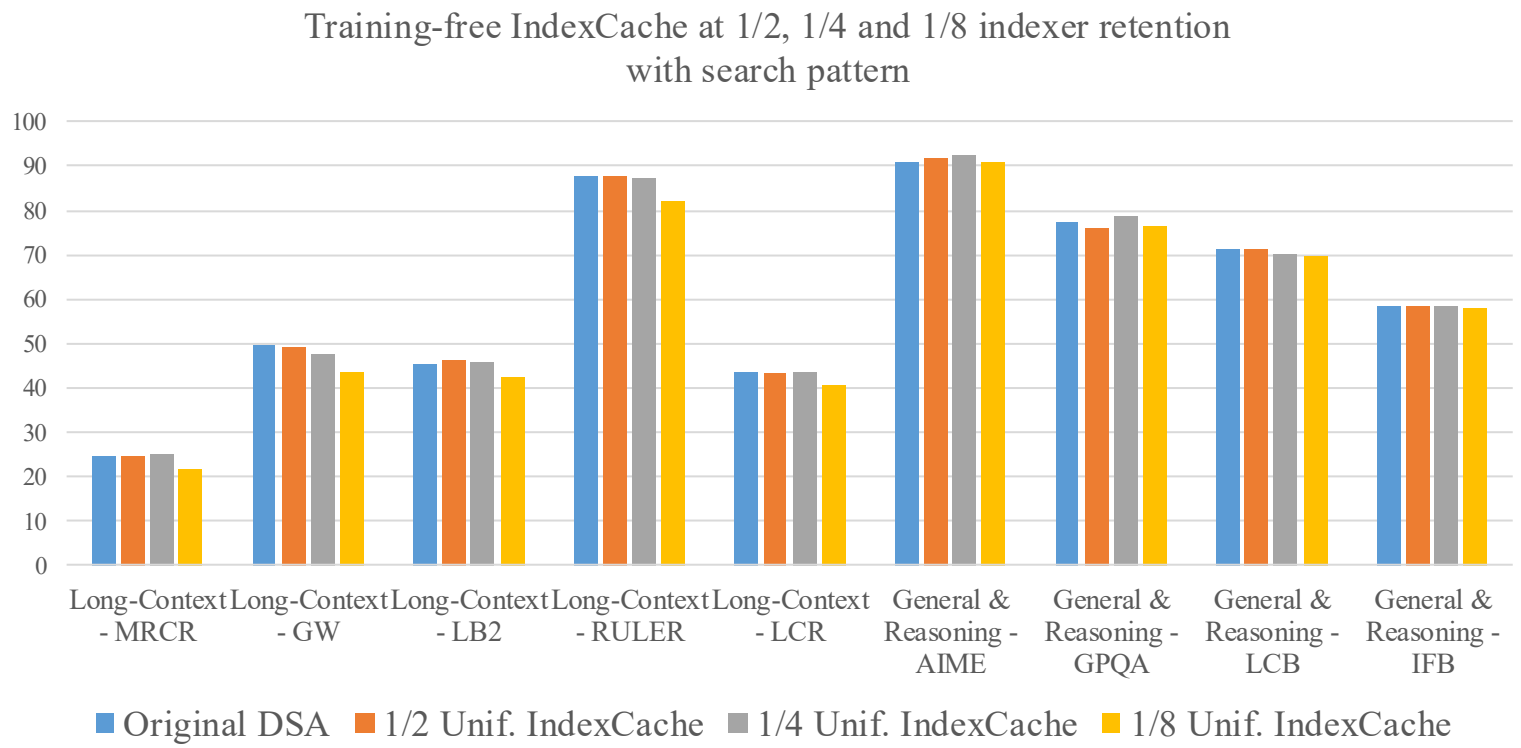
When KVCache is fully saturated, IndexCache brings a 1.51× increase

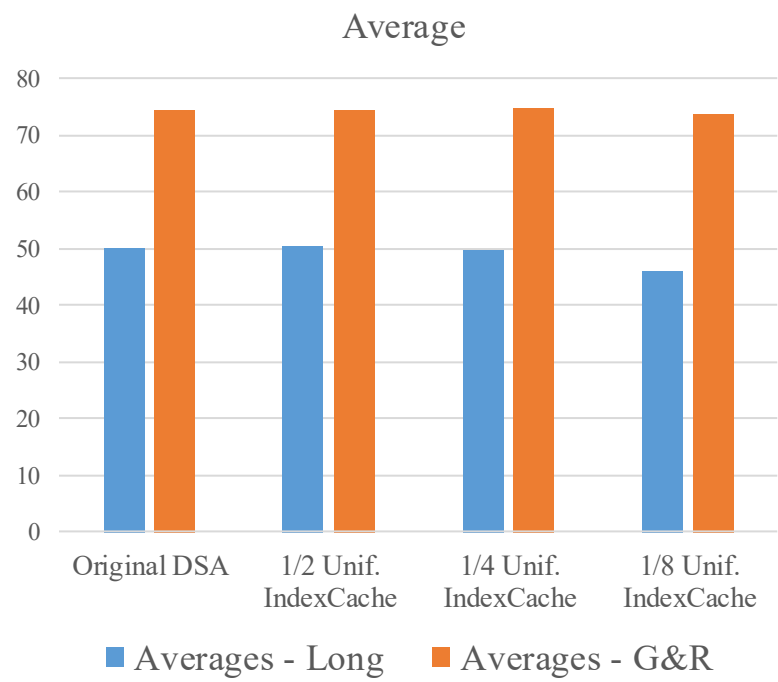
Evaluation: Training-Free Accuracy



Uniform interleaving strategy results in maximum 0.3× precision losses

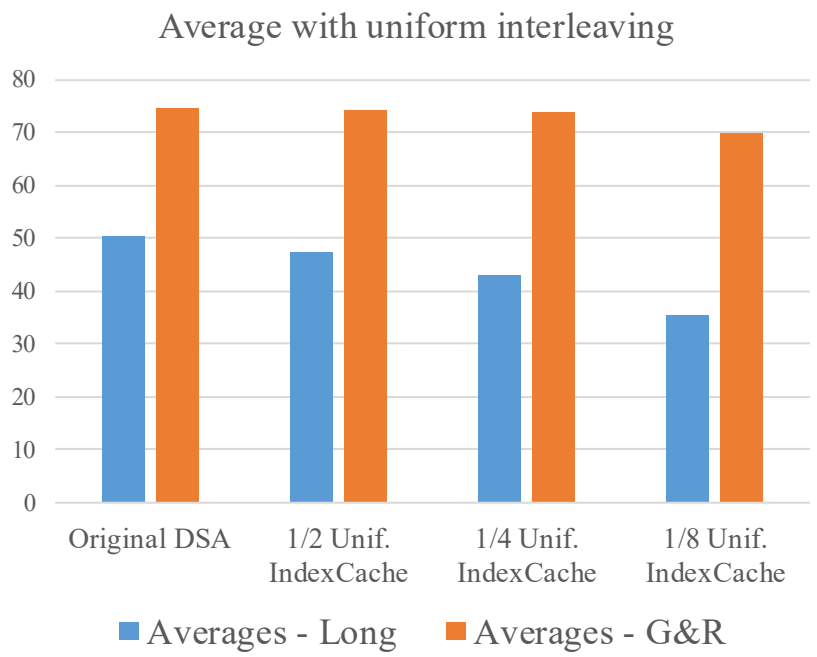
Evaluation: Training-Free Accuracy

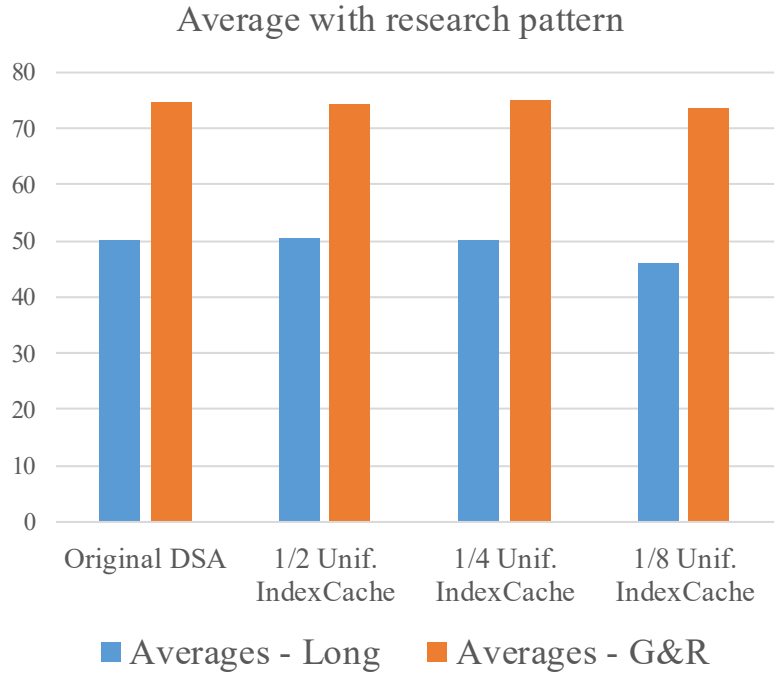




Search pattern strategy can maintain accuracy at 1/4 indexer retention

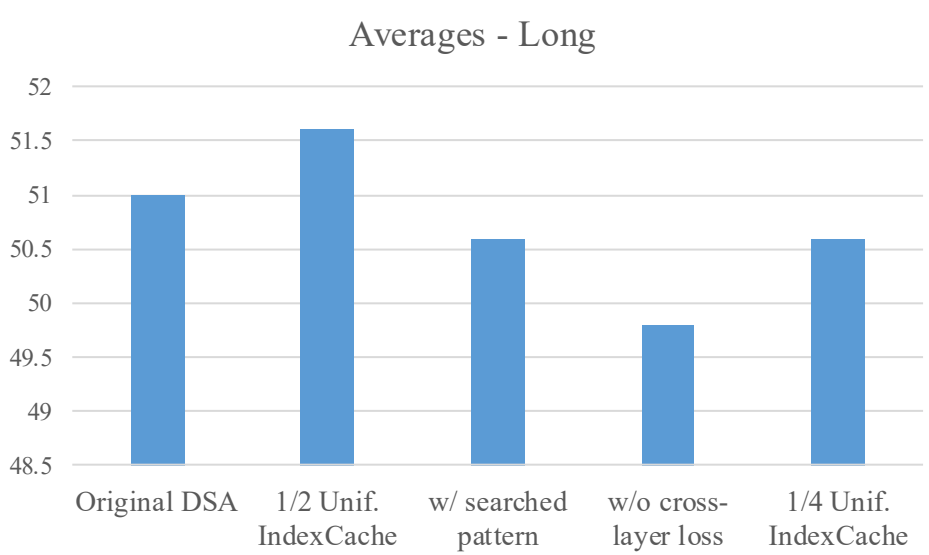
Evaluation: Training-Free Accuracy

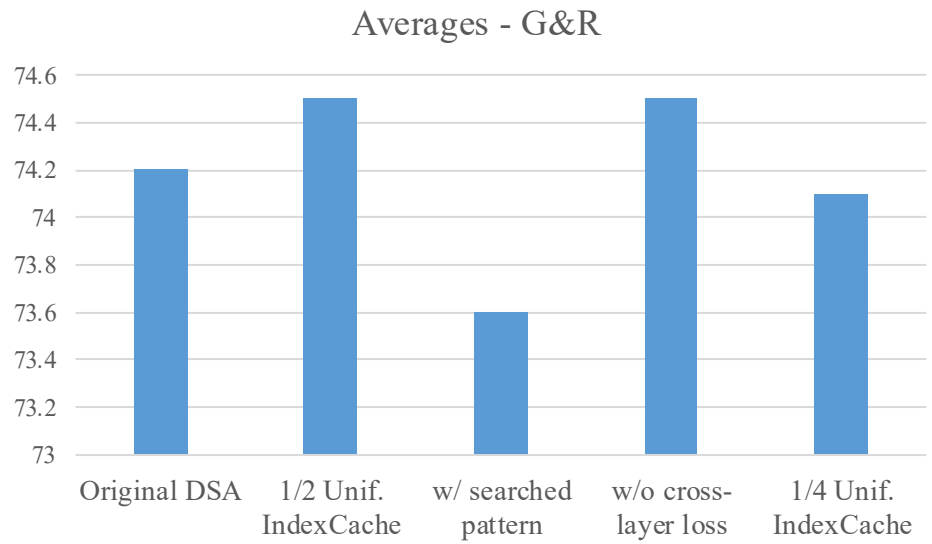




Long chain-of-thought reasoning capabilities are preserved

Evaluation: Training-Aware Accuracy





- **Pattern search is redundant:** Joint training makes simple uniform interleaving sufficient.
- **Cross-layer loss is critical:** Removing it degrades performance, proving the necessity of multi-layer optimization.

Evaluation: Scaling Up

	Long Avg	MRCR v2	GraphWalks	LongBench v2	RULER	AA-LCR
Original DSA	78.4	71.1	**92.7**	64.5	**97.7**	66.2
1/2 Unif. IndexCache	78.1	**72.8**	90.2	65.1	97.6	64.6
+Searched pattern	**78.7**	72.3	90.8	**66.0**	97.3	67.2
1/4 Unif. IndexCache	72.7	65.8	74.9	62.2	96.2	64.6
+Searched pattern	78.0	70.8	90.3	63.7	97.6	**67.6**

- **Scaling Up:** Evaluated on the production-scale GLM-5 (744B total, 40B active parameters).
- **Performance:** Long-context average remains virtually unchanged
- **Speed:** Deliver **~1.2×** end-to-end speedup at 1/2 retention.

Discussion

- **Given a DSA model with N layers compute the cosine similarity between (i > j):**
  - The core attention output at layer i when using its own indexer
  - The core attention output at layer i when reusing layer j’ s indexer
- **Seek the pattern c∗ that maximizes the total similarity**

$$\mathbf{c}^* = \arg \max_{\mathbf{c}:\,|\{i:c_i=\mathsf{F}\}|=M} \sum_{\ell:\,c_\ell=\mathsf{S}} S_{\ell,\mathrm{src}(\ell)}$$

	Avg	MRCR v2	GraphWalks	RULER
Original DSA	**54.0**	**24.5**	**49.6**	**87.9**
1/2 Unif. IndexCache	50.7	22.0	46.6	83.6
+Searched pattern	49.8	22.9	43.5	82.9

Doubts

- ❑Greedy search suboptimality
- ❑Forward-only index reuse
- ❑Weights in multi-layer distillation

$$\mathcal{L}_{\text{multi}}^{\text{I}} = \sum_{j=0}^{m} \frac{1}{m+1} \sum_{t} D_{\text{KL}}\left(\mathbf{p}_t^{(\ell+j)} \,\|\, \mathbf{q}_t^{(\ell)}\right)$$