ASPLOS 2025

清華大學 Tsinghua University

Neuralink: Fast LLM Inference on Smartphones with Neuron Co-Activation Linking

Tuowei Wang*, Ruwen Fan*, Minxing Huang, Zixu Hao, Kun Li, Ting Cao, Youyou Lu, Ju Ren

Tsinghua University

ASPLOS 2025

清華大學
Tsinghua University

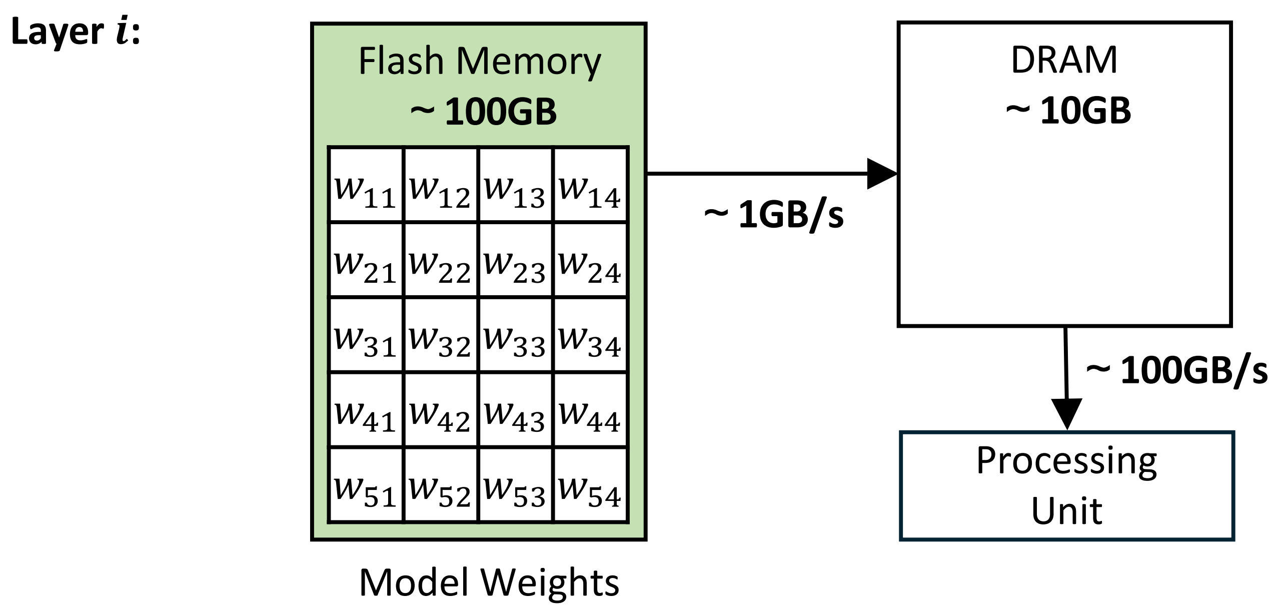
Neuralink: Fast LLM Inference on Smartphones with Neuron Co-Activation Linking

Tuowei Wang*, Ruwen Fan*, Minxing Huang, Zixu Hao, Kun Li, Ting Cao, Youyou Lu, Ju Ren

Tsinghua University

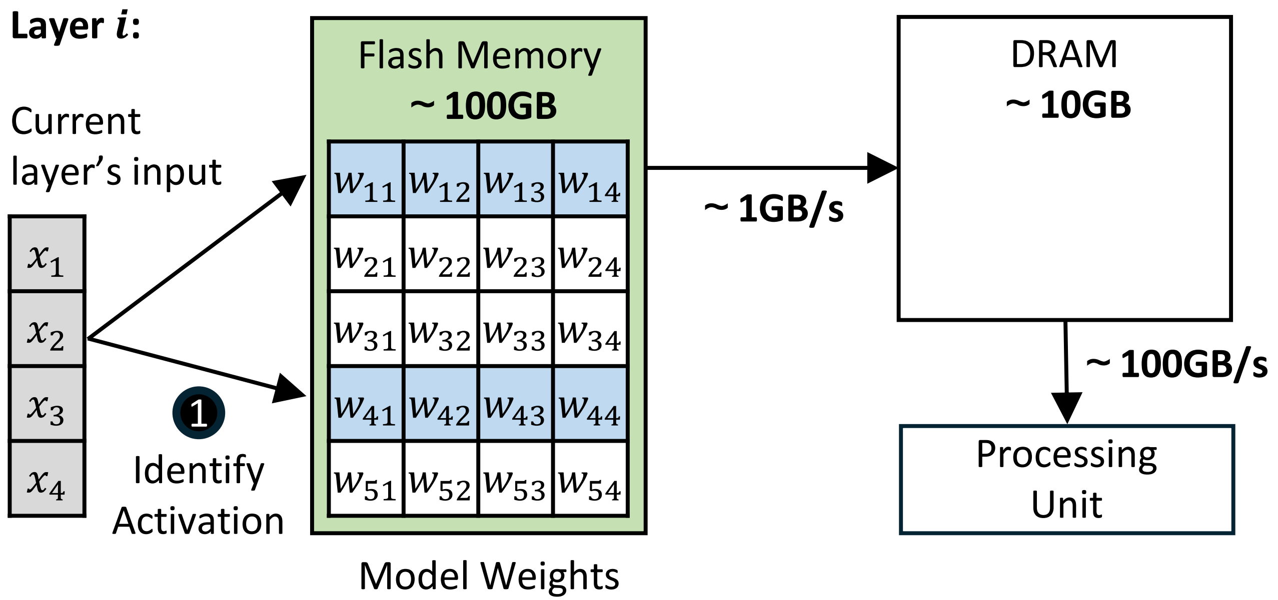
Background: Activation Sparsity in on-Device LLM Inference

- There is a growing demand for deploying LLMs on **mobile devices**, such as smartphones.
- Given the limited DRAM capacity, **activation sparsity** is widely used to support on-device LLM inference.
  - **The full model parameters are stored in the much larger flash memory.**



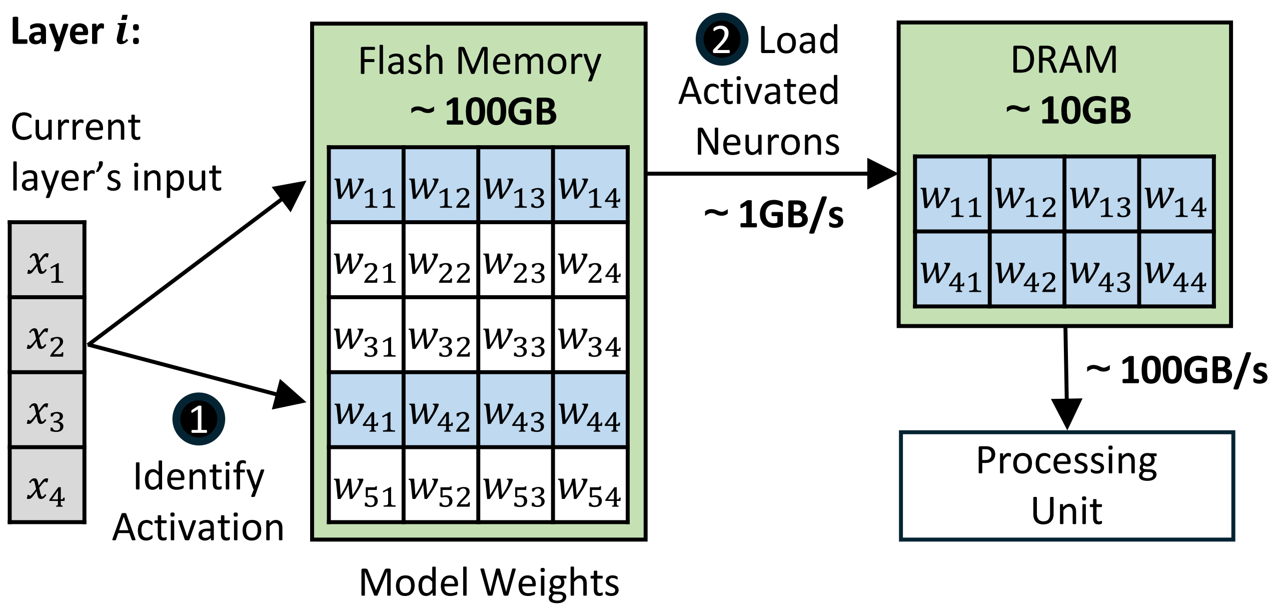
Background: Activation Sparsity in on-Device LLM Inference

- There is a growing demand for deploying LLMs on **mobile devices**, such as smartphones.
- Given the limited DRAM capacity, **activation sparsity** is widely used to support on-device LLM inference.
  - The full model parameters are stored in the much larger flash memory.
  - **For a given input, only a subset of relevant model parameters is identified.**



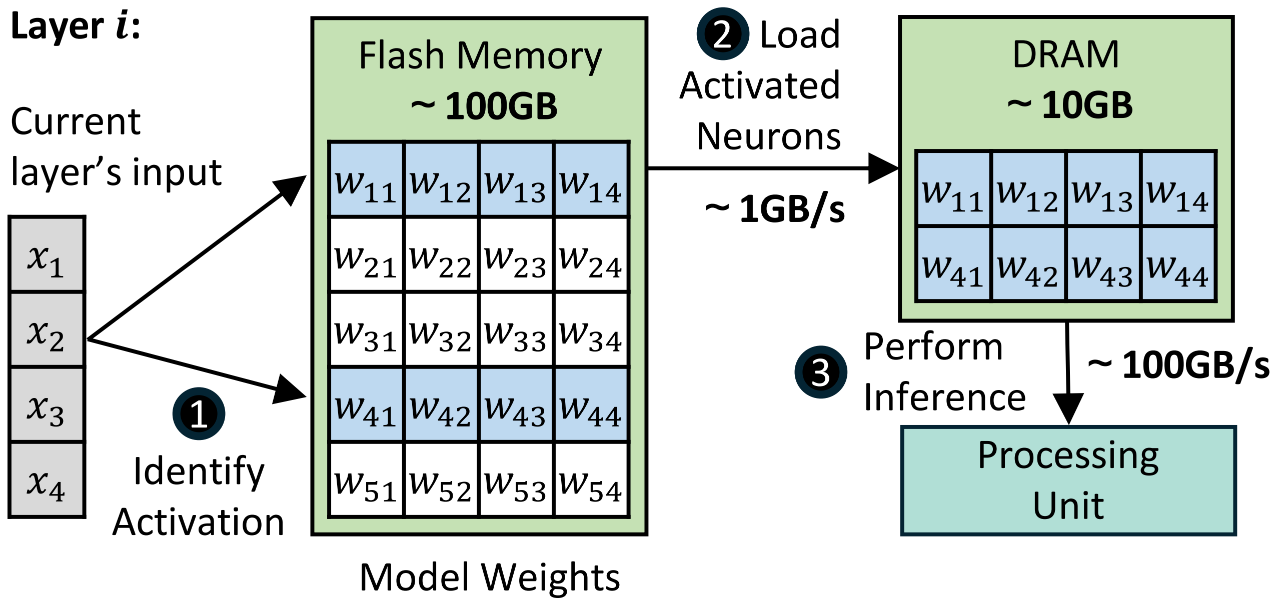
Background: Activation Sparsity in on-Device LLM Inference

- There is a growing demand for deploying LLMs on **mobile devices**, such as smartphones.
- Given the limited DRAM capacity, **activation sparsity** is widely used to support on-device LLM inference.
  - The full model parameters are stored in the much larger flash memory.
  - For a given input, only a subset of relevant model parameters is identified.
  - **The corresponding activated neurons are loaded from flash memory into DRAM.**



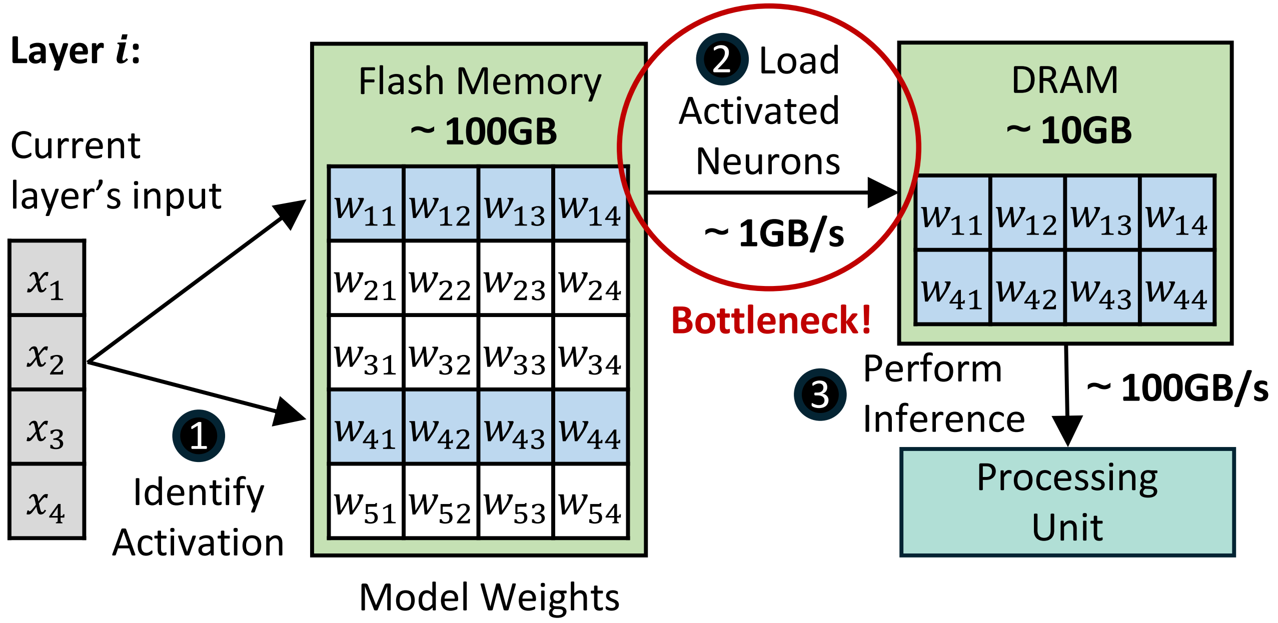
Background: Activation Sparsity in on-Device LLM Inference

- There is a growing demand for deploying LLMs on **mobile devices**, such as smartphones.
- Given the limited DRAM capacity, **activation sparsity** is widely used to support on-device LLM inference.
  - The full model parameters are stored in the much larger flash memory.
  - For a given input, only a subset of relevant model parameters is identified.
  - The corresponding activated neurons are loaded from flash memory into DRAM.
  - **Sparse computation is then performed, producing outputs that are nearly the same as those of dense models.**



Background: Activation Sparsity in on-Device LLM Inference

- There is a growing demand for deploying LLMs on **mobile devices**, such as smartphones.
- Given the limited DRAM capacity, **activation sparsity** is widely used to support on-device LLM inference.
  - The full model parameters are stored in the much larger flash memory.
  - For a given input, only a subset of relevant model parameters is identified.
  - The corresponding activated neurons are loaded from flash memory into DRAM.
  - Sparse computation is then performed, producing outputs that are nearly the same as those of dense models.
- **The I/O overhead between flash memory and DRAM becomes the primary bottleneck.**

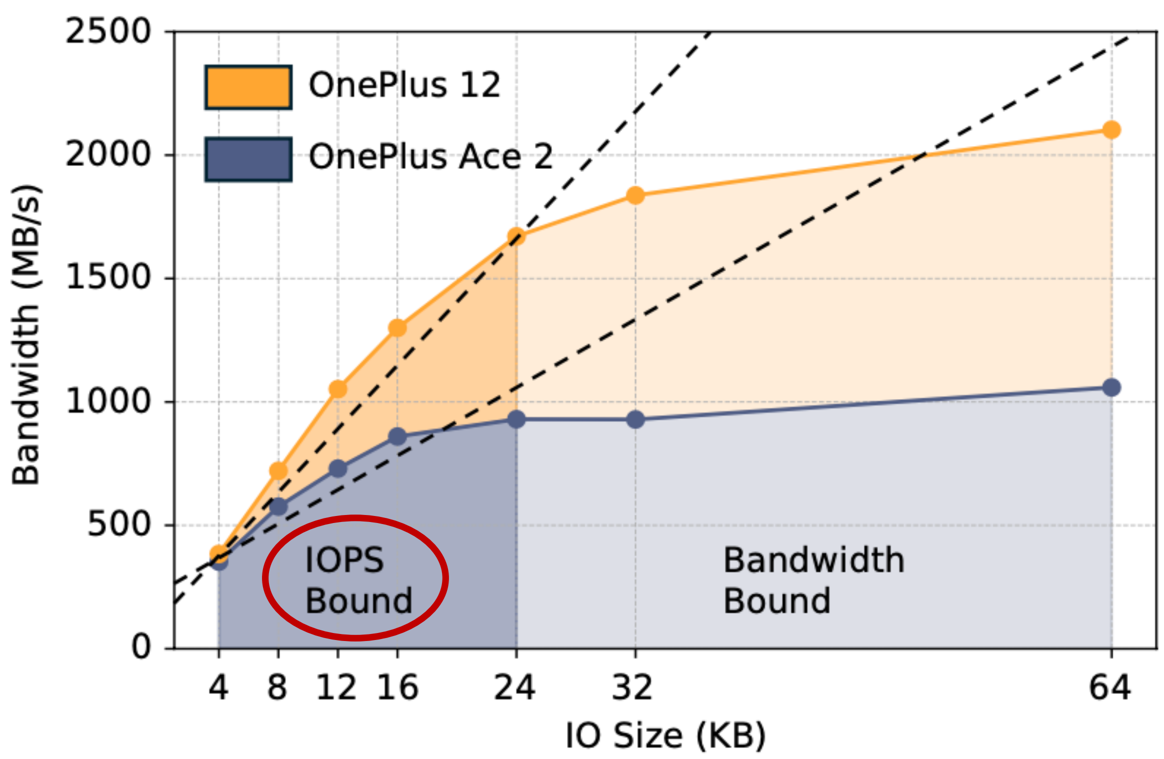


Breakdown of inference latency per token when offloading FFN blocks to flash memory on OnePlus Ace2.

Model	Compute	I/O	Total	I/O Ratio
OPT-350M	82 ms	776 ms	858 ms	90.4%
OPT-1.3B	202 ms	988 ms	1,190 ms	83.0%
OPT-6.7B	804 ms	2,224 ms	3,028 ms	73.4%
Llama-2-7B	609 ms	10,388 ms	10,997 ms	94.5%
Mistral-7B	540 ms	12,220 ms	12,760 ms	95.8%
MobiLlama-1B	230 ms	1,909 ms	2,139 ms	89.2%
Phi-2-2.7B	461 ms	1,976 ms	2,437 ms	81.1%

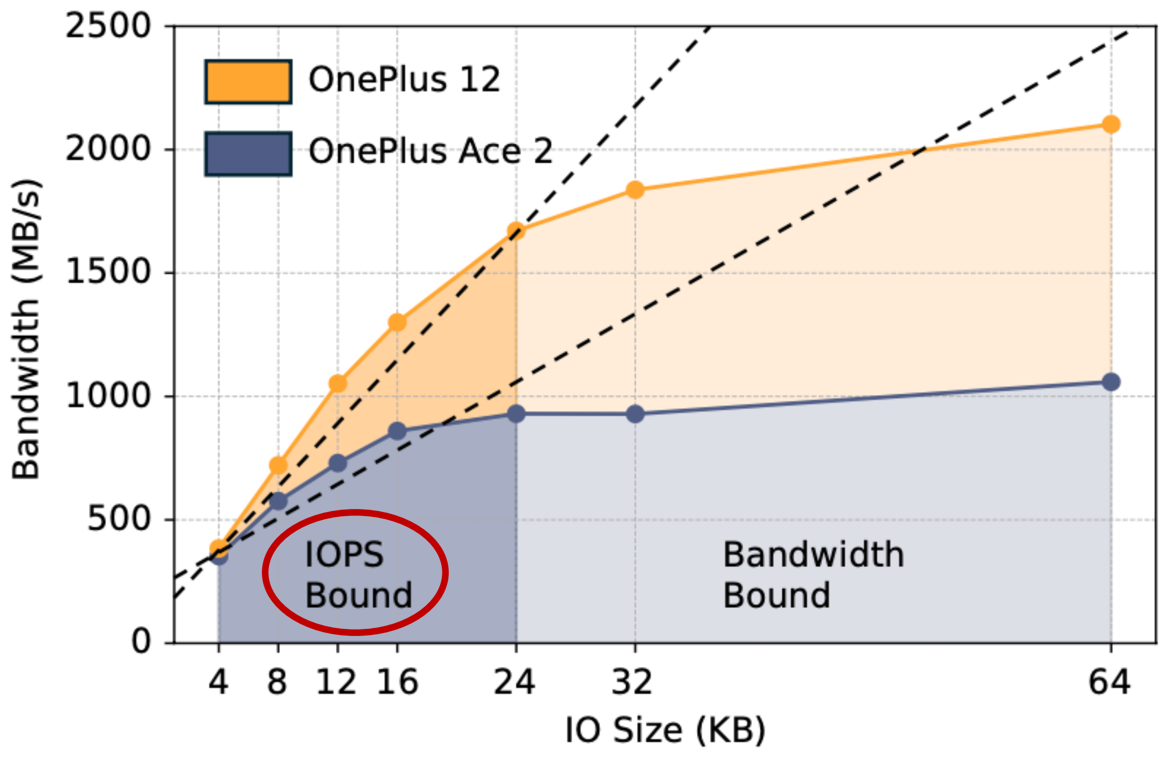
Motivation: IOPS as the Performance Bottleneck

- Mobile devices utilize **Universal Flash Storage (USF)** as the storage protocol.
- Compared to server-side NVMe, UFS provides a much **shallower command queue** (e.g., 32 entries in UFS 4.0).
- This limitation restricts flash read **I/O Operations Per Second (IOPS)** and prevents full utilization of the bandwidth.



Motivation: IOPS as the Performance Bottleneck

- Mobile devices utilize **Universal Flash Storage (USF)** as the storage protocol.
- Compared to server-side NVMe, UFS provides a much **shallower command queue** (e.g., 32 entries in UFS 4.0).
- This limitation restricts flash read **I/O Operations Per Second (IOPS)** and prevents full utilization of the bandwidth.
- The scattered activation of parameters induces numerous **small-grained read accesses**, further intensifying the constraint.



Latency (ms) and bandwidth (MB/s) across different ratios of non-activated neurons in OPT-350M on OnePlus 12.

Ratio	dense	10%	20%	30%	40%
Bandwidth	1637.61	1355.35	1089.24	904.69	746.03
Latency	234.49	254.96	281.96	297.10	308.76
Speedup	-	0.92	0.83	0.79	0.76
Ratio	50%	60%	70%	80%	90%
Bandwidth	598.82	524.50	441.33	396.43	368.05
Latency	320.63	292.78	260.86	193.68	104.18
Speedup	0.73	0.80	0.90	1.21	2.25

Observation: Neuron Co-Activation Phenomenon

- Conventional **model-structure-based placement** scatters activated parameters across flash memory.
- **Neuron Co-Activation:**
    - Some neurons tend to be activated together with a relatively fixed set of other neurons.
    - Prevalent across different model structures and datasets.
- **Insight: Co-locating frequently co-activated neurons in flash memory** to enable more continuous reads.

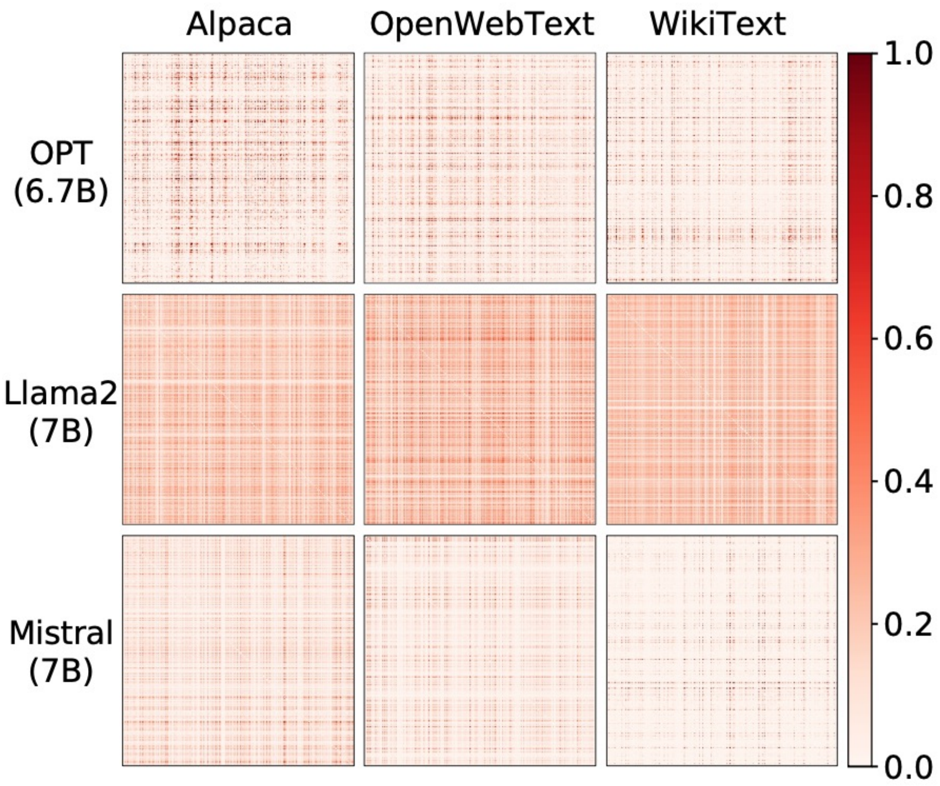
Observation: Neuron Co-Activation Phenomenon

- Conventional **model-structure-based placement** scatters activated parameters across flash memory.
- **Neuron Co-Activation:**
  - Some neurons tend to be activated together with a relatively fixed set of other neurons.
  - Prevalent across different model structures and datasets.
- **Insight: Co-locating frequently co-activated neurons in flash memory** to enable more continuous reads.

Challenge 1: Extensive Search Space
The immense number of neurons in LLM results in an exponentially large space of possible placement combinations.

Observation: Neuron Co-Activation Phenomenon

- Conventional **model-structure-based placement** scatters activated parameters across flash memory.
- **Neuron Co-Activation:**
  - Some neurons tend to be activated together with a relatively fixed set of other neurons.
  - Prevalent across different model structures and datasets.
- **Insight: Co-locating frequently co-activated neurons in flash memory** to enable more continuous reads.



Challenge 1: Extensive Search Space
The immense number of neurons in LLM results in an exponentially large space of possible placement combinations.

Challenge 2: Inherent Activation Dynamics
The activation patterns of parameters inherently exhibit dynamics across varying inputs, causing unexpected discontinuities.

Observation: Neuron Co-Activation Phenomenon

- Conventional **model-structure-based placement** scatters activated parameters across flash memory.
- **Neuron Co-Activation:**
  - Some neurons tend to be activated together with a relatively fixed set of other neurons.
  - Prevalent across different model structures and datasets.
- **Insight: Co-locating frequently co-activated neurons in flash memory** to enable more continuous reads.

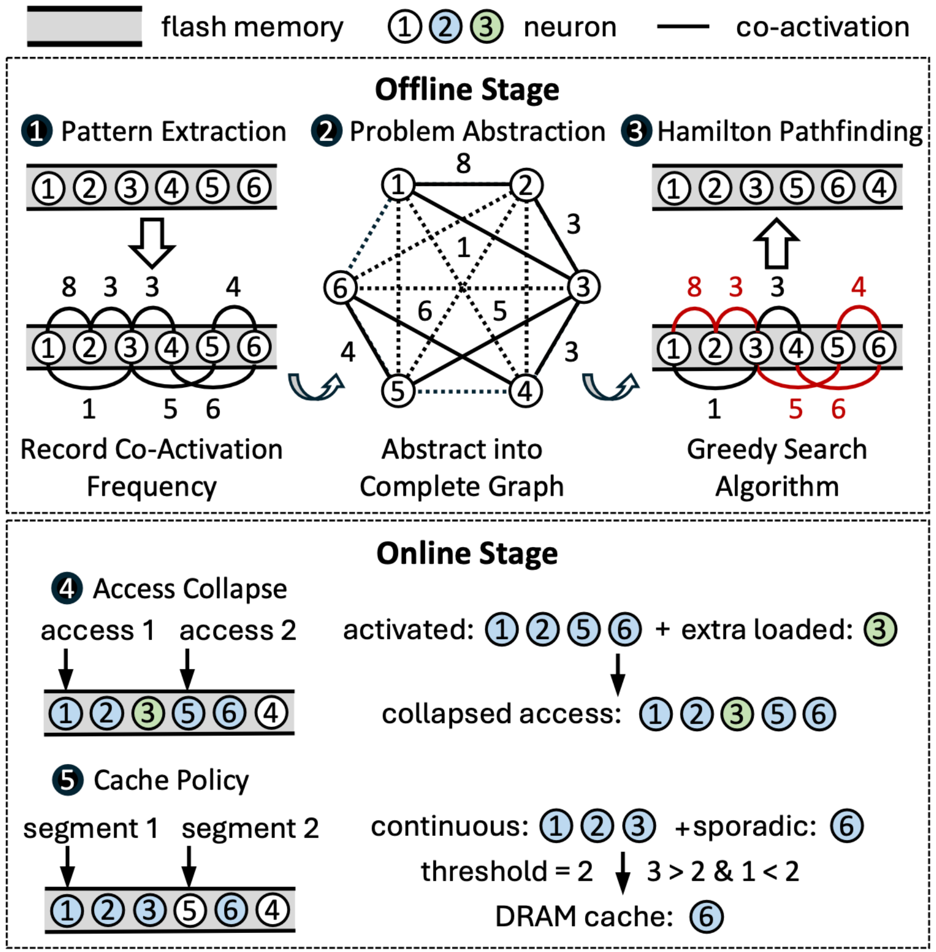
Challenge 1: Extensive Search Space
The immense number of neurons in LLM results in an exponentially large space of possible placement combinations.

Challenge 2: Inherent Activation Dynamics
The activation patterns of parameters inherently exhibit dynamics across varying inputs, causing unexpected discontinuities.

Challenge 3: Misaligned Cache Strategy
Existing cache strategies treat neurons individually, leading to fragmentation in their placement in flash memory.

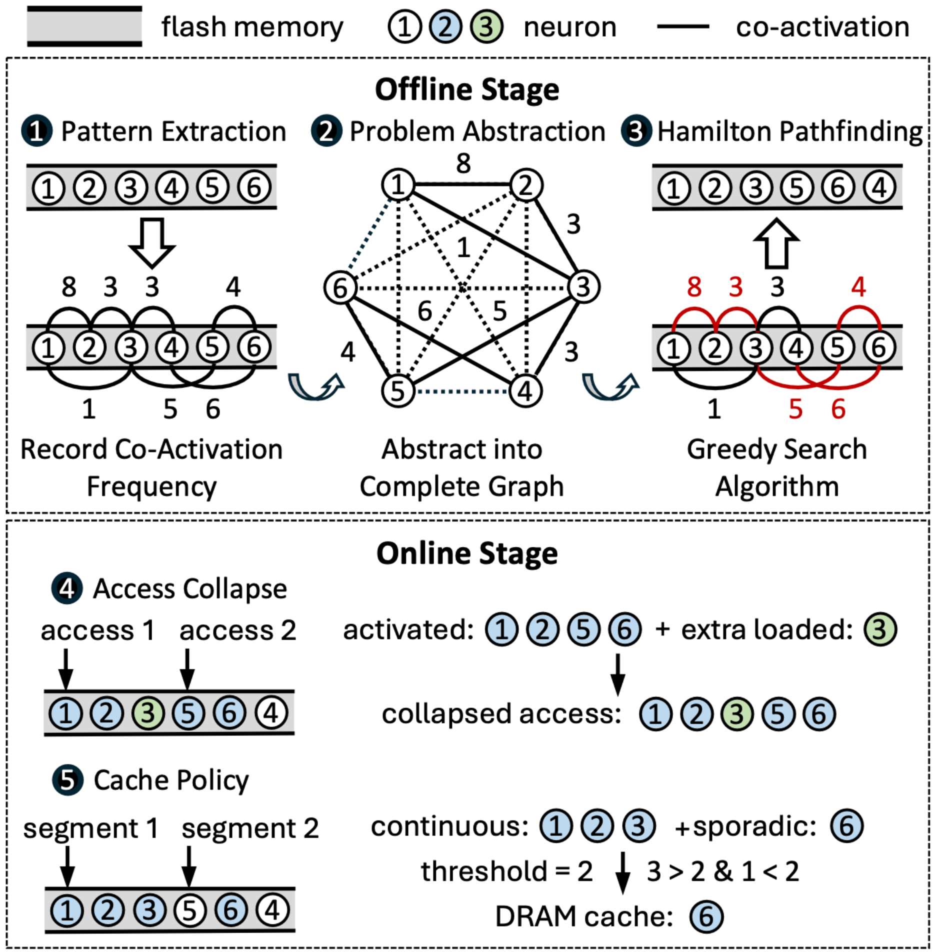
Design: Neuralink Overview

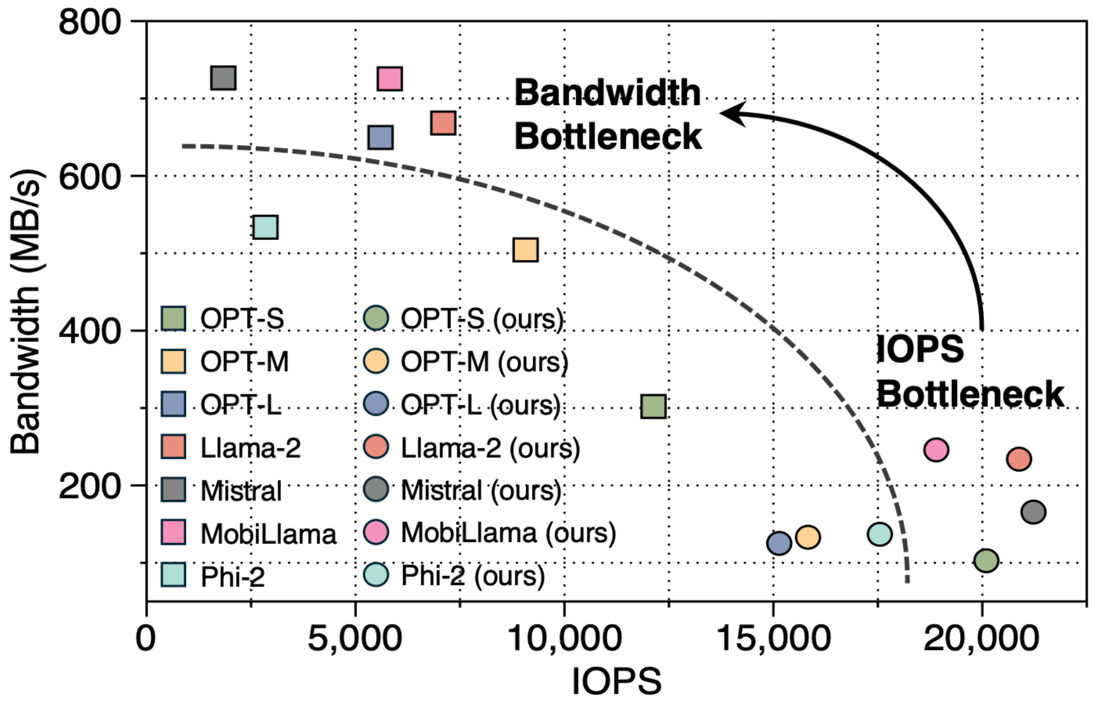
- We propose **Neuralink**, an approach to accelerating LLM inference on smartphones through optimized I/O access.
- **Offline Correlation-Aware Clustering:** Identifies an optimized neuron placement in flash memory.
- **Online Continuity-Centric Processing:** Employs customized data access and DRAM management at runtime.



Design: Neuralink Overview

- We propose **Neuralink**, an approach to accelerating LLM inference on smartphones through optimized I/O access.
- **Offline Correlation-Aware Clustering:** Identifies an optimized neuron placement in flash memory.
- **Online Continuity-Centric Processing:** Employs customized data access and DRAM management at runtime.





Neuralink shifts the I/O bottleneck from IOPS to Bandwidth!

Offline Design: Correlation-Aware Clustering

- **Step 1: Extract neuron co-activation patterns from profiling results.**

Co-activation probability of neuron n_i and neuron n_j: $P(ij) = \dfrac{f(n_i, n_j)}{\sum_{k=1}^{N} \sum_{l=1}^{N} f(n_k, n_l)}$

	1	2	3	4	5	6
1	-					
2	8	-				
3	15	7	-			
4	7	37	6	-		
5	5	29	21	3	-	
6	2	1	2	10	18	-

Neuron Co-Activation Pattern

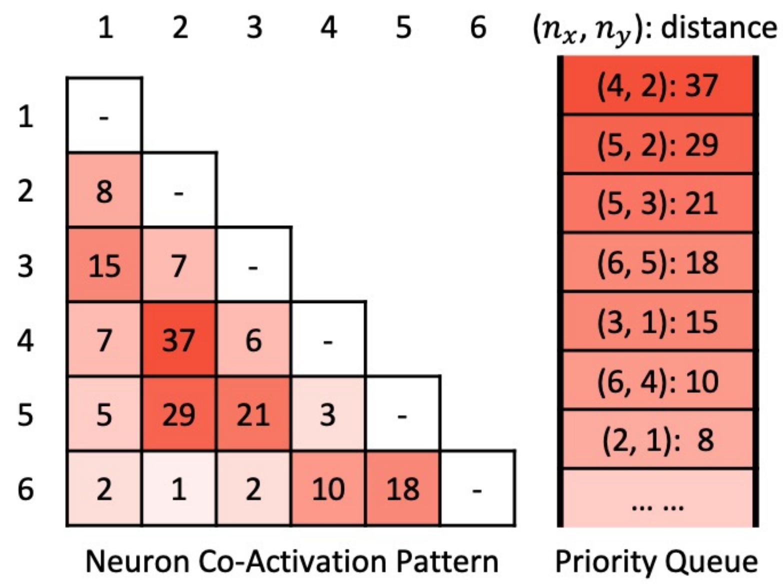
Offline Design: Correlation-Aware Clustering

- **Step 1: Extract neuron co-activation patterns from profiling results.**

  Co-activation probability of neuron n_i and neuron n_j: $P(ij) = \dfrac{f(n_i, n_j)}{\sum_{k=1}^{N} \sum_{l=1}^{N} f(n_k, n_l)}$

- **Step 2: Model neuron placement in flash memory using a graph-based representation.**

  Distance between two neurons: $\mathrm{dist}(n_i, n_j) := 1 - P(ij)$



Offline Design: Correlation-Aware Clustering

- **Step 1: Extract neuron co-activation patterns from profiling results.**

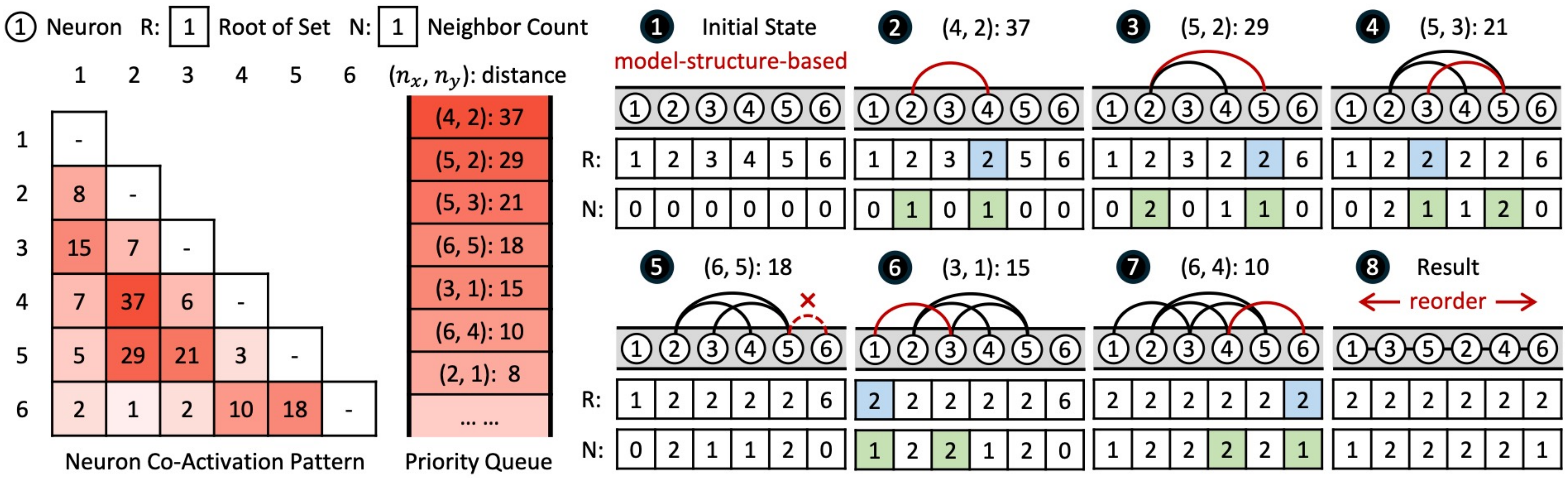
  Co-activation probability of neuron n_i and neuron n_j: $P(ij) = \frac{f(n_i, n_j)}{\sum_{k=1}^{N} \sum_{l=1}^{N} f(n_k, n_l)}$

- **Step 2: Model neuron placement in flash memory using a graph-based representation.**

  Distance between two neurons: $\mathrm{dist}(n_i, n_j) := 1 - P(ij)$

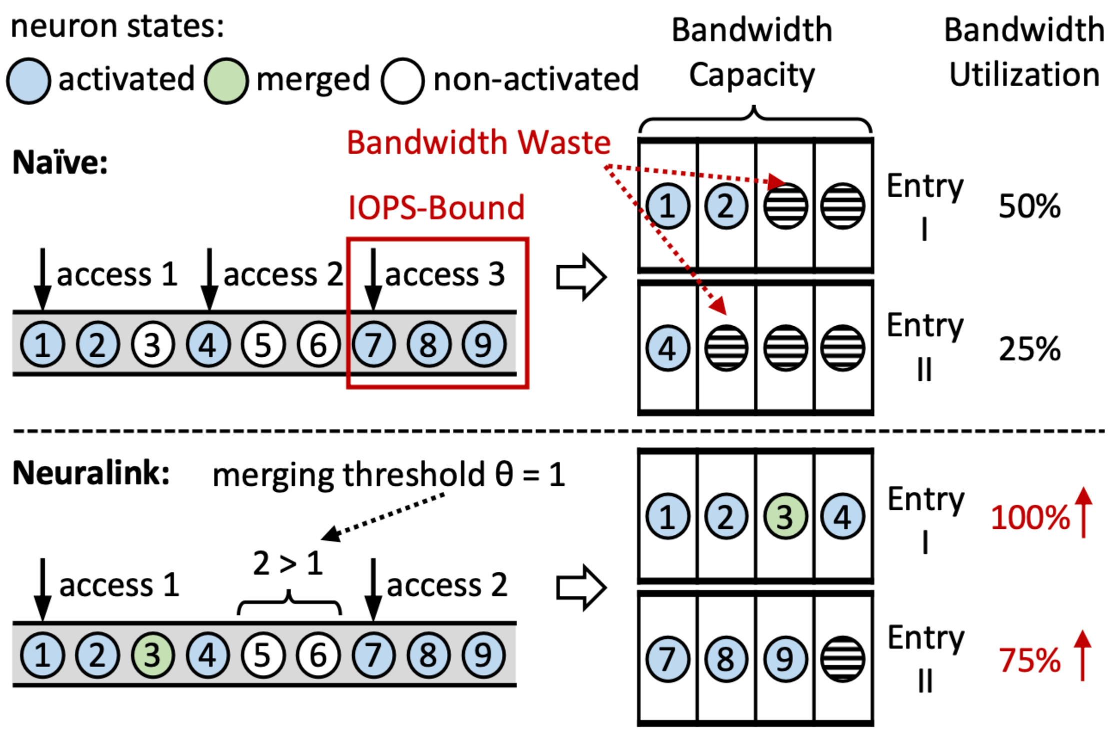
- **Step 3: Design a polynomial-time heuristic algorithm to search for an optimized neuron placement.**

  Distance between two neurons links: $\mathrm{dist}(l_i, l_j) := \min\{\mathrm{dist}(l_i(h), l_j(h)), \mathrm{dist}(l_i(h), l_j(t)), \mathrm{dist}(l_i(t), l_j(h)), \mathrm{dist}(l_i(t), l_j(t))\}$



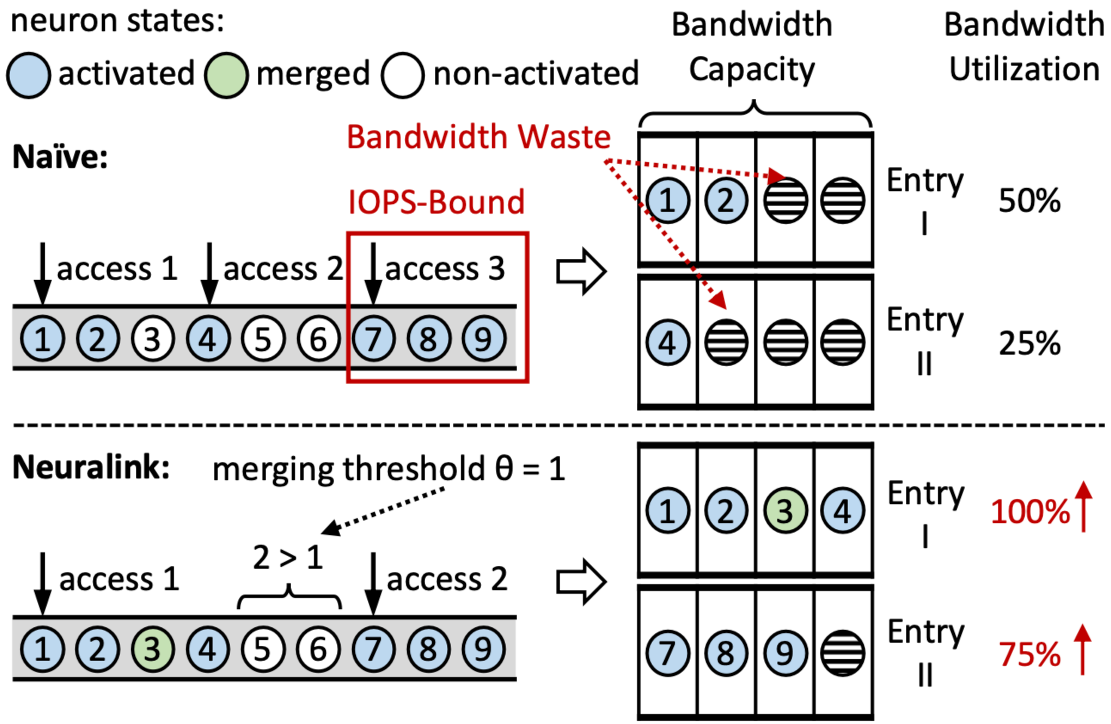
Online Design: Continuity-Centric Processing

- **IOPS-Friendly Access Collapse:** Strategically merges nearby read accesses to reduce I/O operations.
  - If the number of neurons between two neuron links falls below a threshold, access collapse is applied.
  - When bandwidth is fully utilized, the system reverts to the original access strategy.



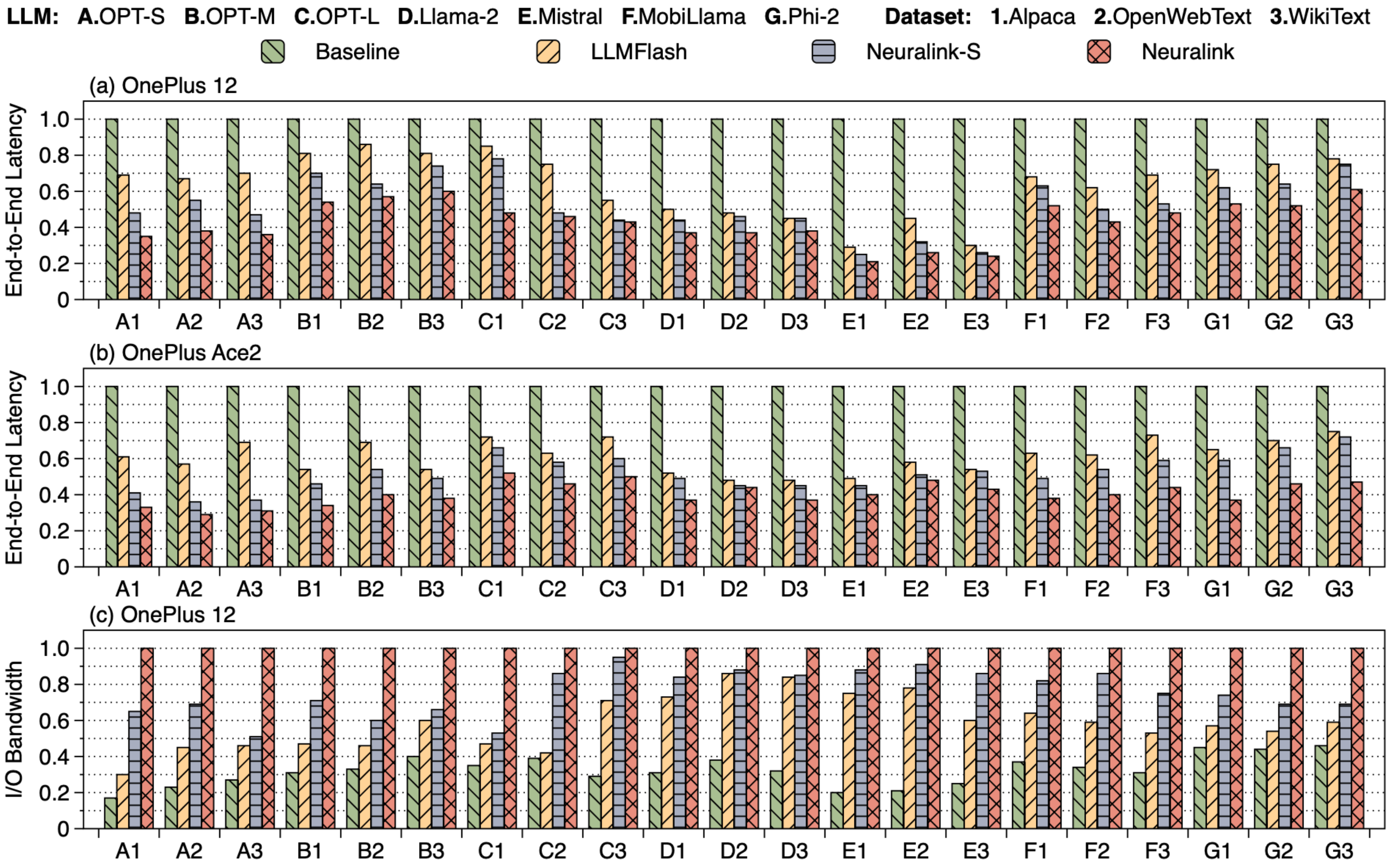
Online Design: Continuity-Centric Processing

- **IOPS-Friendly Access Collapse:** Strategically merges nearby read accesses to reduce I/O operations.
  - If the number of neurons between two neuron links falls below a threshold, access collapse is applied.
  - When bandwidth is fully utilized, the system reverts to the original access strategy.
- **Linking-Aligned Cache Policy:** Caches neurons in DRAM at the granularity of neuron segments.
  - Prioritizes caching outlier neurons that are co-activated with only a small number of surrounding neurons.
  - Caches continuous segments with lower priority than outlier neurons.



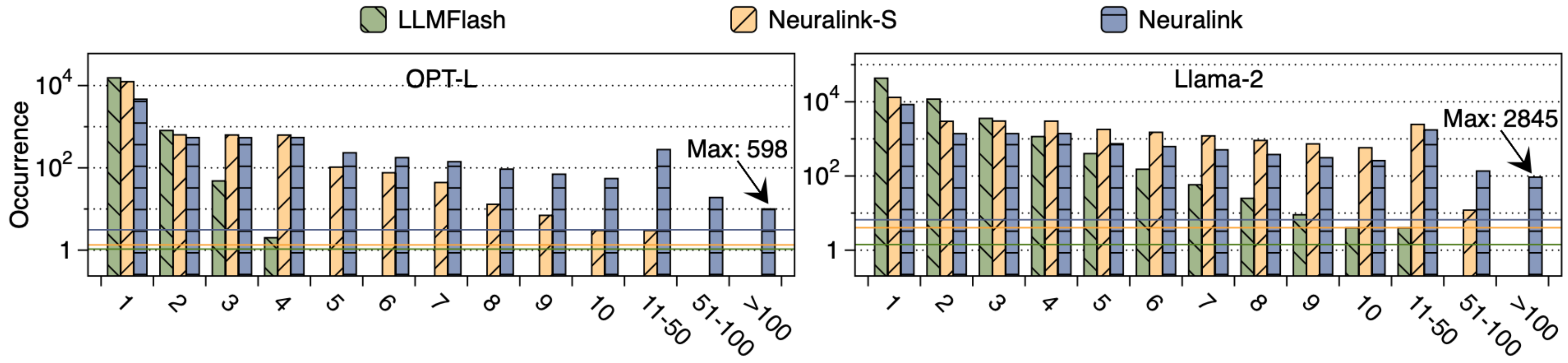
Evaluation – Overall Performance

- **7** Models × **8** Datasets × **3** Hardware × **3** Baselines (llama.cpp, LLMFlash, Neuralink-S)
- **End-to-end Latency:** Achieves average speedups of 2.37×, 1.48×, and 1.25× over the three baselines.
- **Effective Bandwidth:** Achieves average improvements of 3.28×, 1.80×, and 1.36× over the three baselines.



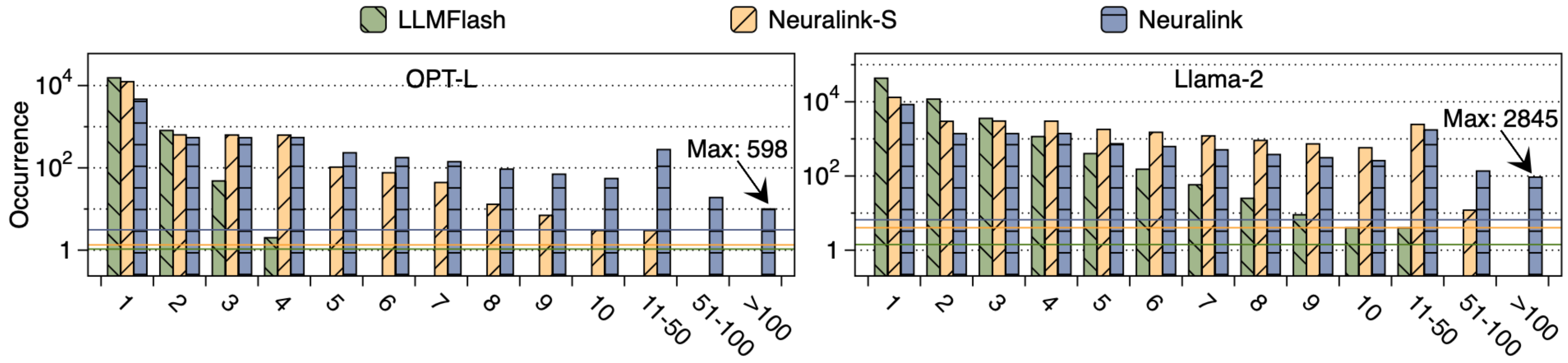
Evaluation – Ablation and Sensitivity Analysis

- **Statistical information on read access lengths per token.**
  - Neuralink increases the average read access lengths to 3.12 (from 1.06) and 6.62 (from 1.42) across OPT-L and Llama-2.



Evaluation – Ablation and Sensitivity Analysis

- **Statistical information on read access lengths per token.**
  - Neuralink increases the average read access lengths to 3.12 (from 1.06) and 6.62 (from 1.42) across OPT-L and Llama-2.



- **Performance across different combinations of profiling (rows) and testing (columns) datasets.**
  - Neuron co-activation patterns are largely intrinsic to the LLM and are minimally affected by input variations.

Model	*Profiling* Dataset	*Testing* Dataset		
		Alpaca	OpenWebText	WikiText
OPT-L	Alpaca	711.09 ms	809.80 ms	800.46 ms
	Speedup	1.86×	1.65×	1.59×
	OpenWebText	856.48 ms	802.01 ms	800.26 ms
	Speedup	1.54×	1.67×	1.59×
	WikiText	823.93 ms	1031.41 ms	784.19 ms
	Speedup	1.60×	1.30×	1.63×

Model	*Profiling* Dataset	*Testing* Dataset		
		Alpaca	OpenWebText	WikiText
Llama-2	Alpaca	4405.76 ms	4537.75 ms	3747.40 ms
	Speedup	1.27×	1.17×	1.45×
	OpenWebText	3546.12 ms	4118.59 ms	4318.28 ms
	Speedup	1.57×	1.29×	1.26×
	WikiText	4769.36 ms	4535.48 ms	4578.87 ms
	Speedup	1.17×	1.17×	1.18×

Related Works and Implementations

- **I/O + Model Weight:**
  - *Neuralink: Fast LLM Inference on Smartphones with Neuron Co-Activation Linking*
- **I/O + KV Cache:**
  - *DynaKV: Enabling Accurate and Efficient Long-Sequence LLM Decoding on Smartphones*
- **Compute + Test-Time Scaling:**
  - *Scaling LLM Test-Time Compute with Mobile NPU on Smartphones*

- **On-Device NPU Operator Library:**
  - *https://github.com/omnimind-ai/OmniOp-NPU*
- **On-Device LLM Inference Framework:**
  - *https://github.com/omnimind-ai/OmniInfer-LLM*
- **On-Device VLM Inference Framework:**
  - *https://github.com/omnimind-ai/OmniInfer-VLM*

Thank you.

wtw23@mails.tsinghua.edu.cn