



Towards Efficient Generative AI: Software/Hardware Co-Design for the Next Generation of Intelligent Systems

Yiran Chen

*Department of Electrical and Computer Engineering, Duke University
Duke University Center for Computational Evolutionary Intelligence (CEI)
NSF IUCRC For Alternative Sustainable and Intelligence Computing (ASIC)
NSF AI Institute for Edge Computing Leveraging Next-generation Networks (Athena)*

Duke

Generative AI is Powerful

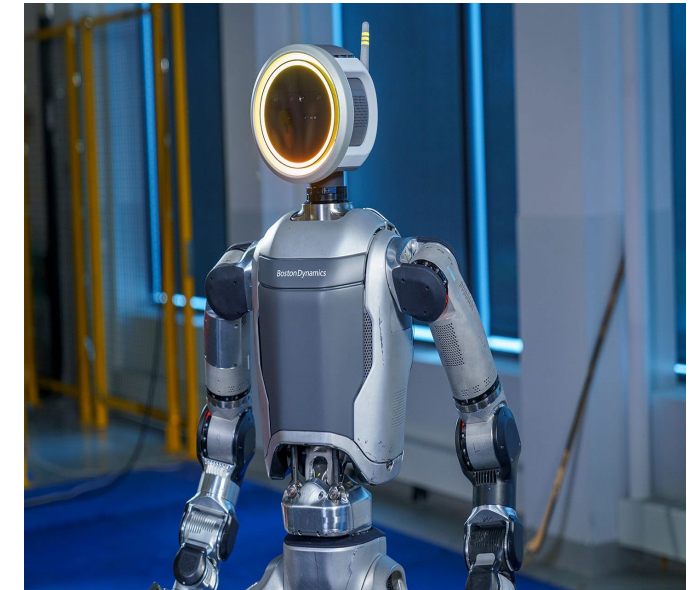
- We are living in an era of rapid progress in Generative AI



Text Generation



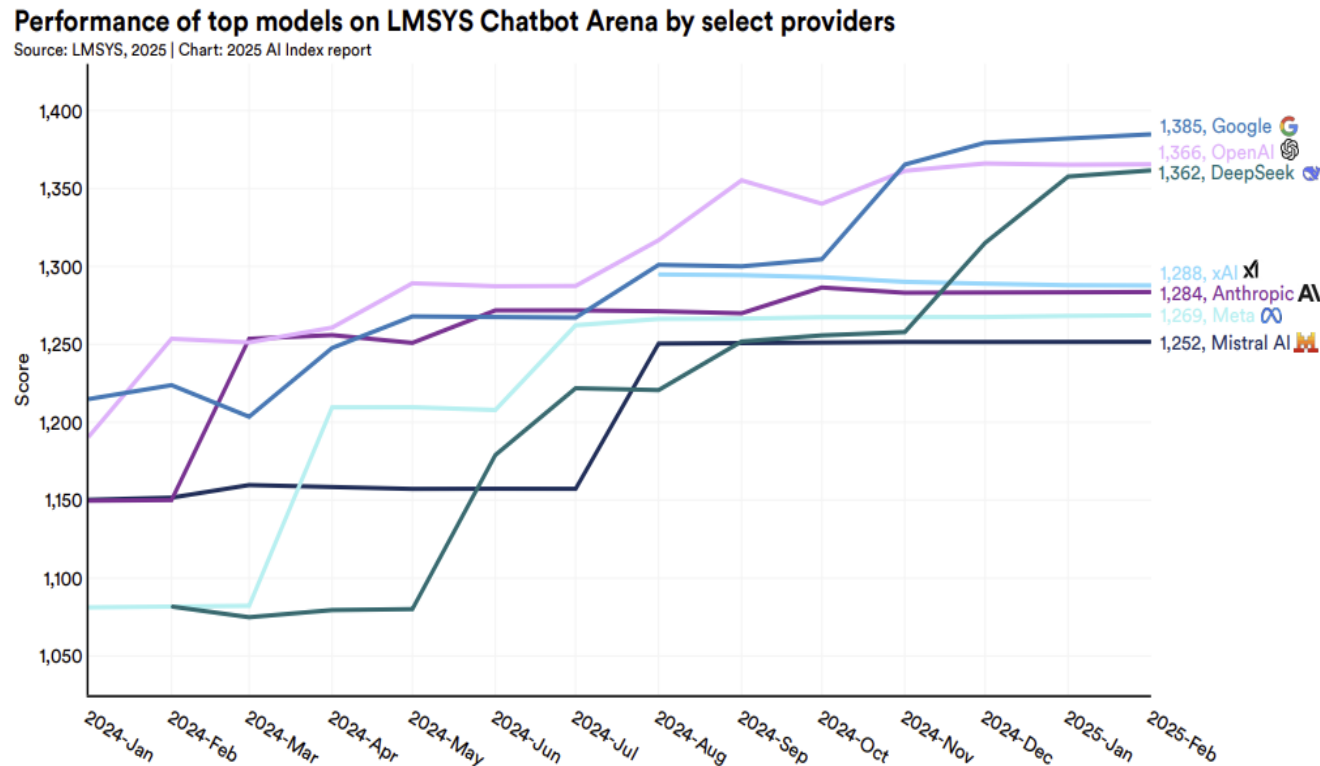
Image/Video Generation



Action Generation

Generative AI is Powerful

- We are living in an era of rapid progress in Generative AI



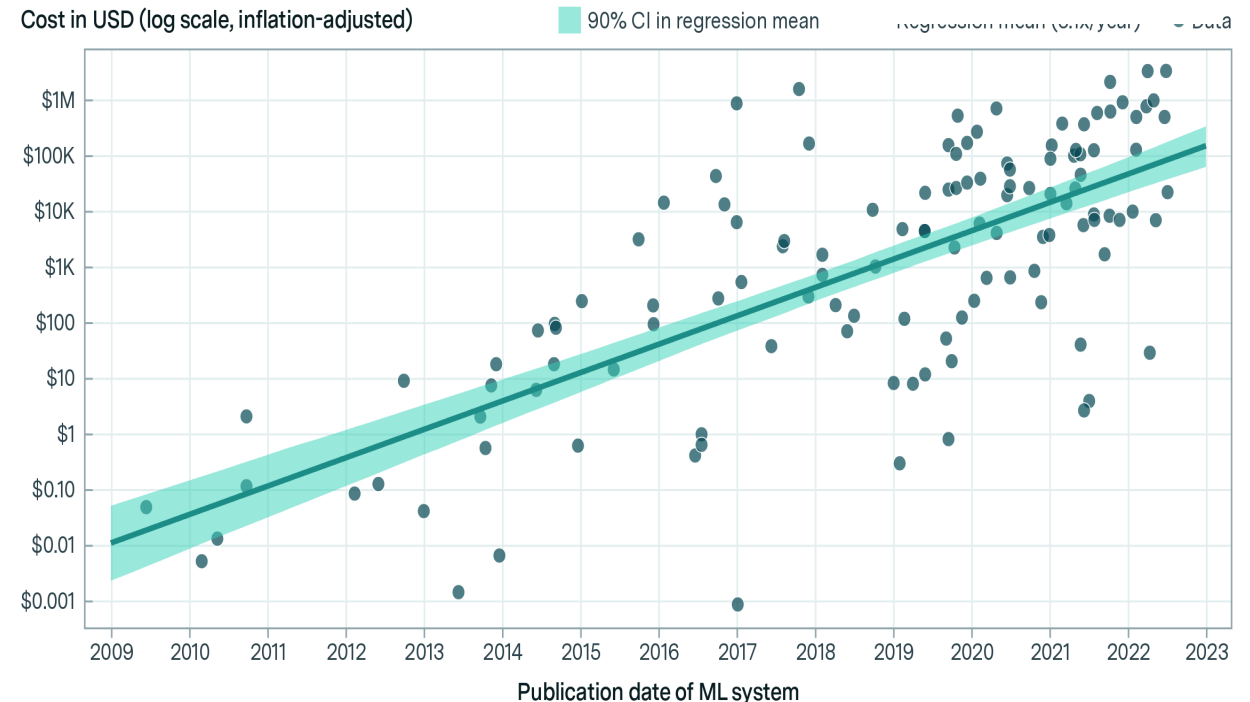
Index, Artificial Intelligence. "Artificial intelligence index report 2025." (2025).

But they are also Expensive

Expensive Training

- **BERT-Base (110 M)** can be pre-trained on your laptop for hours.
- **LLaMA-7B (7 B)** can be trained with a small multi-GPU rig in a couple of days.
- **Turing NLG (17 B)**; you will need a mid-scale GPU pod for several days.
- **GPT-3 (175 B)**; you will need thousands of GPUs and weeks of wall-clock time.

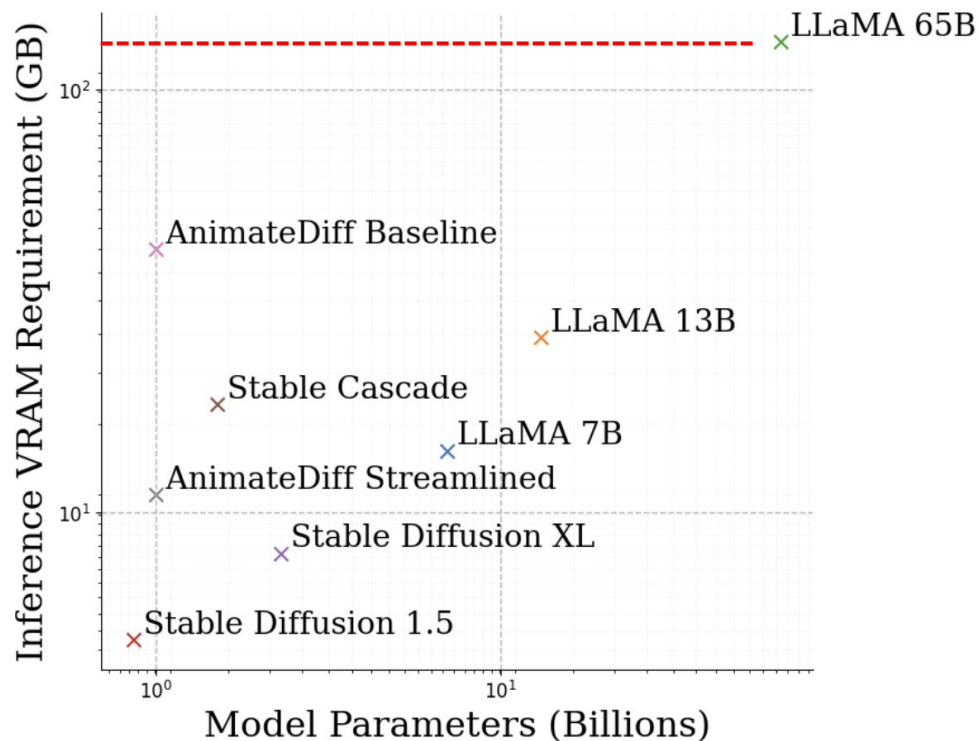
Cost of training compute for notable ML systems



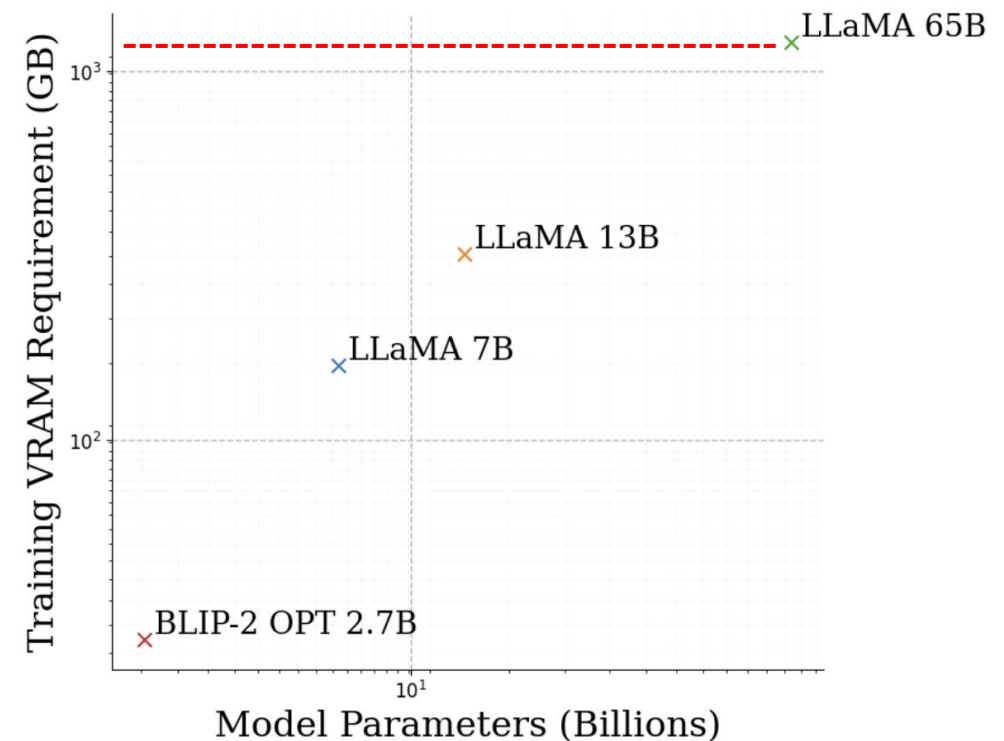
Challenge 1: Memory Gap

- **Model Size and Memory Grow Rapidly According to Scaling Laws.**

Inference VRAM vs. Model Size



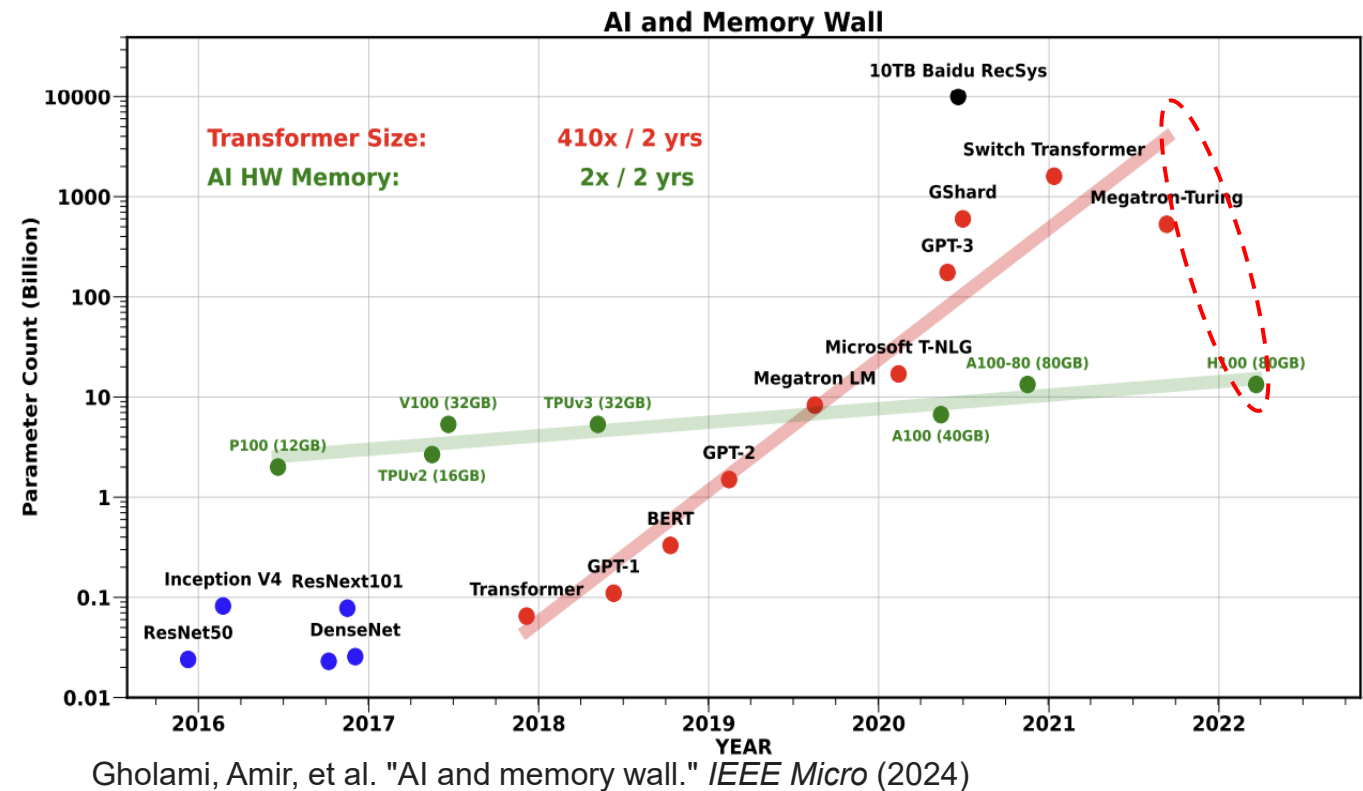
Training VRAM vs. Model Size



Challenge 1: Memory Gap

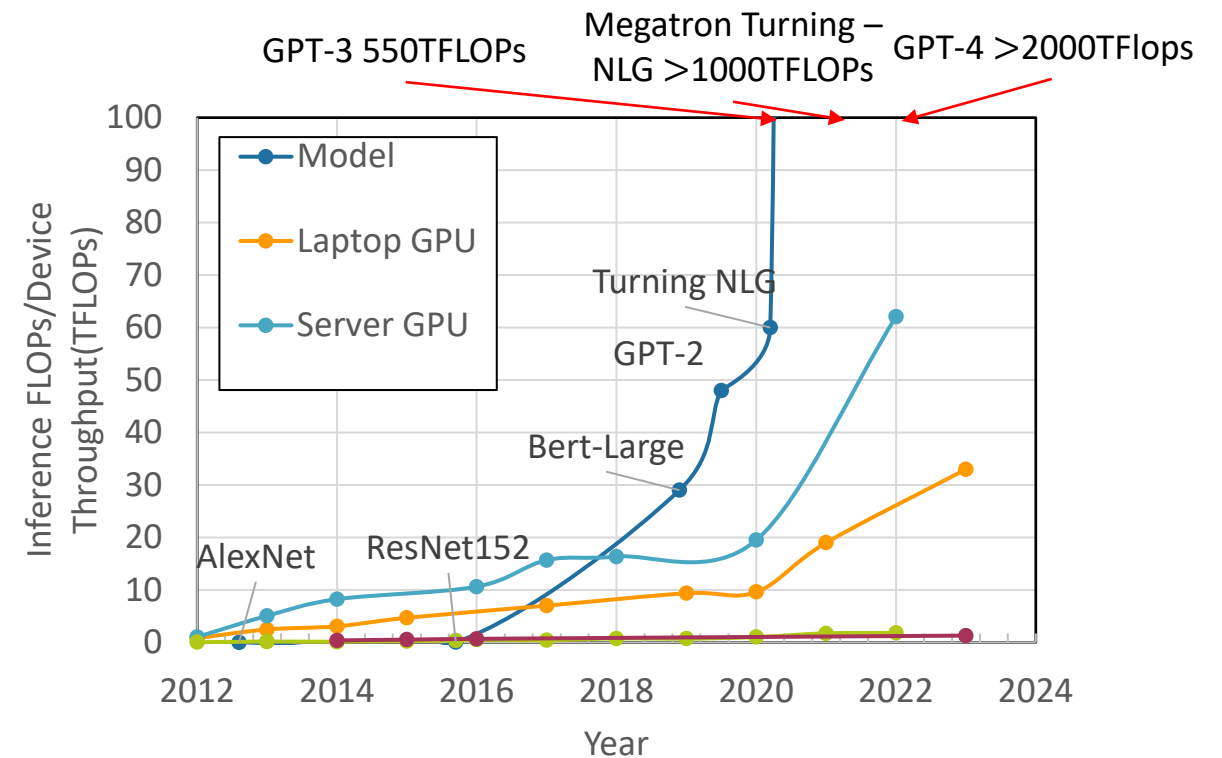
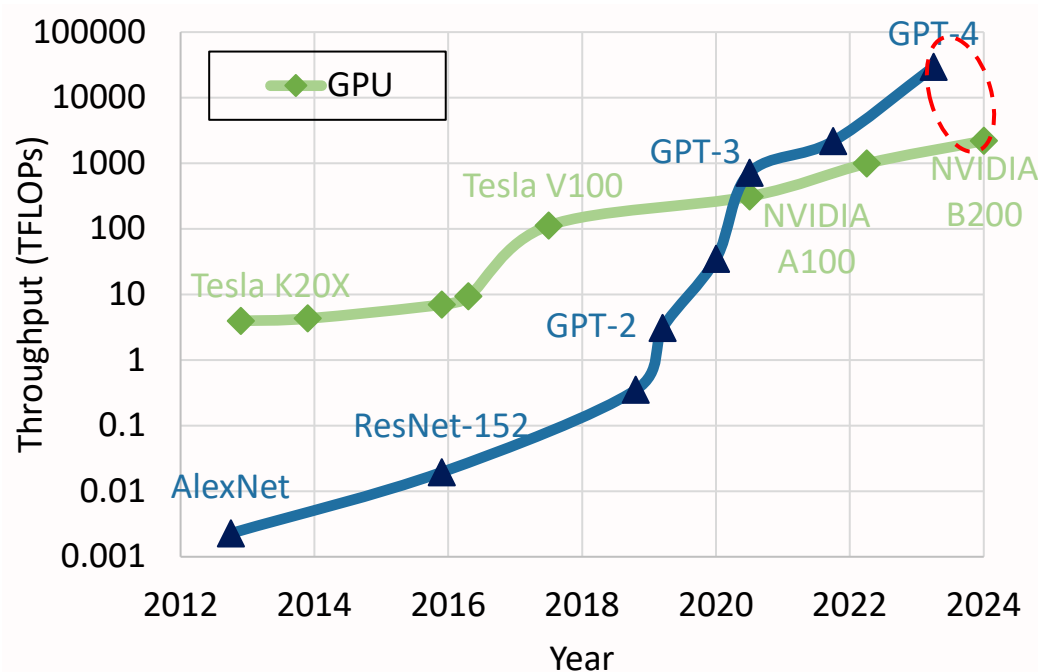
- Hardware memory has struggled to catch up with modern AI models

- The number of parameters in large Transformer models has been growing exponentially by a factor of **410×** every two years
- The single GPU memory has only been scaled at a rate of **2×** every 2 years.



Challenge 2: Computation Gap

- Advanced hardware is not able to catch up fast enough.





Synapses vs. Silicon

- **Biological Brain:** A ~20-watt system of $\sim 10^{15}$ synapses enabling continuous, associative, and efficient lifelong learning.
- **Silicon Brain:** AI progress is throttled by the Memory Wall, where data movement energy cost now dwarfs that of computation.
- The next architectural leap requires learning from biology's algorithmic primitives to re-architect AI computing and memory systems.

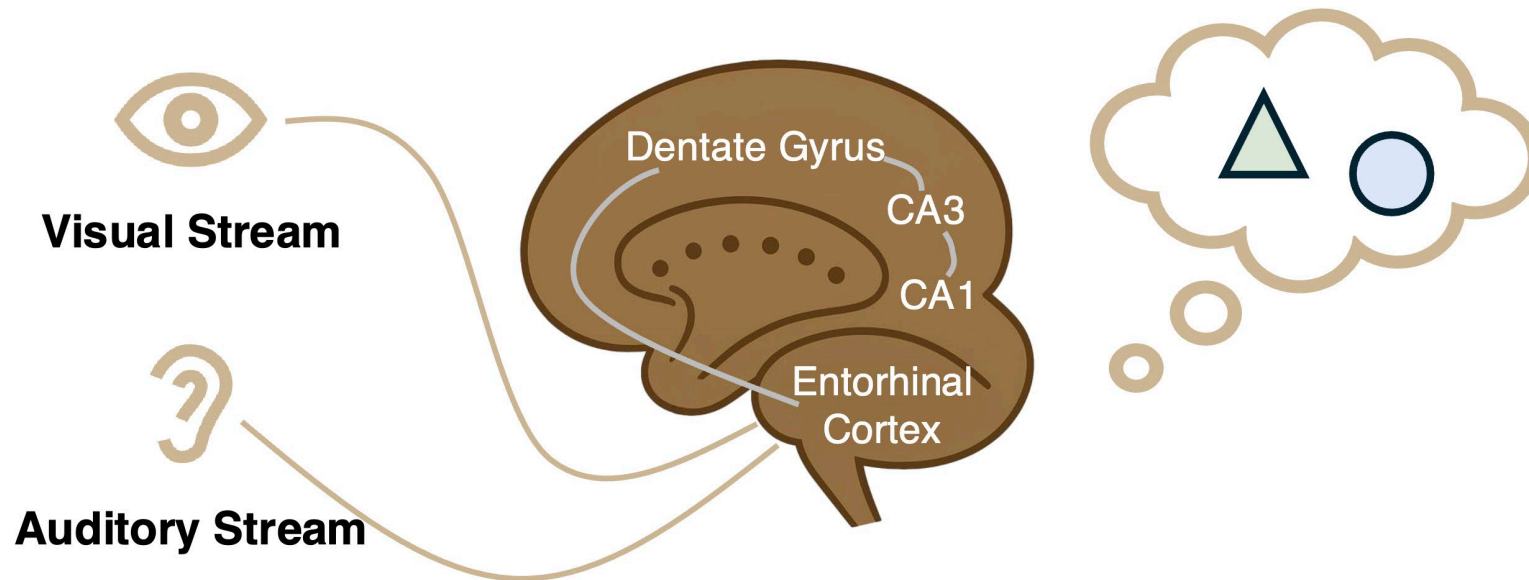
Data from Drachman, D. A. (2005). Do we have brain to spare? *Neurology*, 64(12). / Sterling, P., & Laughlin, S. (2015). *Principles of Neural Design*. MIT Press.

Engineering Episodic Intelligence (HippoMM)

How can we build systems that don't just process data, but remember experiences?

The Challenge of Understanding Long Multimodal Events

- **Humans:** Effortlessly segment and form associative memories from continuous audiovisual experiences.
- **State-of-the-Art AI:** Struggles with temporal integration and cross-modal recall from long-form events.

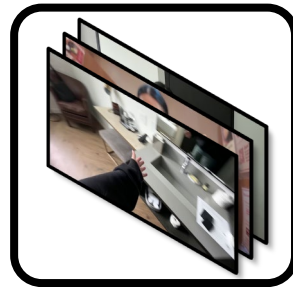


From Messy Streams to Coherent Episodes

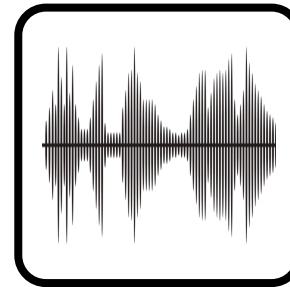
- **Temporal Pattern Separation**

- Mimic the hippocampus's ability to segment continuous experience into discrete events.
- Employs content-adaptive segmentation, which is more semantically coherent than fixed-duration chunking. It uses SSIM to detect visual changes and audio energy to detect silence.

Visual Input



Auditory Input



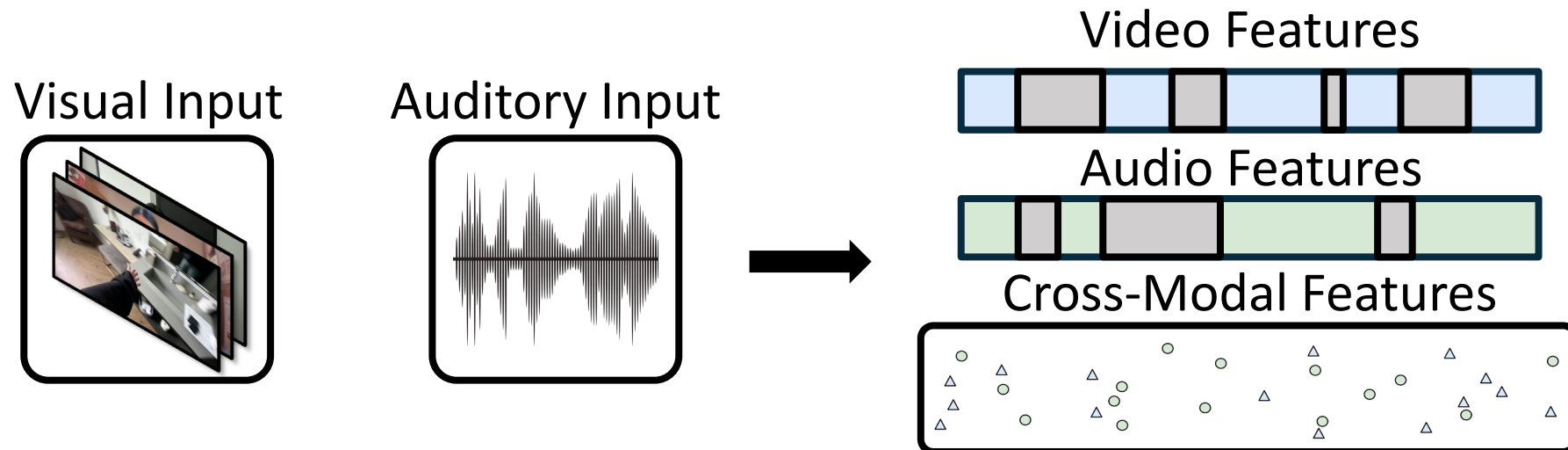
$$\mathcal{S}_t = \bigvee_{m \in \{v, a\}} 1 [d_m(\text{Input}_m(t), \text{Input}_m(t - 1)) > \tau_m]$$

Consolidation & Semantic Replay

- Temporal Pattern Separation

- Perceptual Encoding

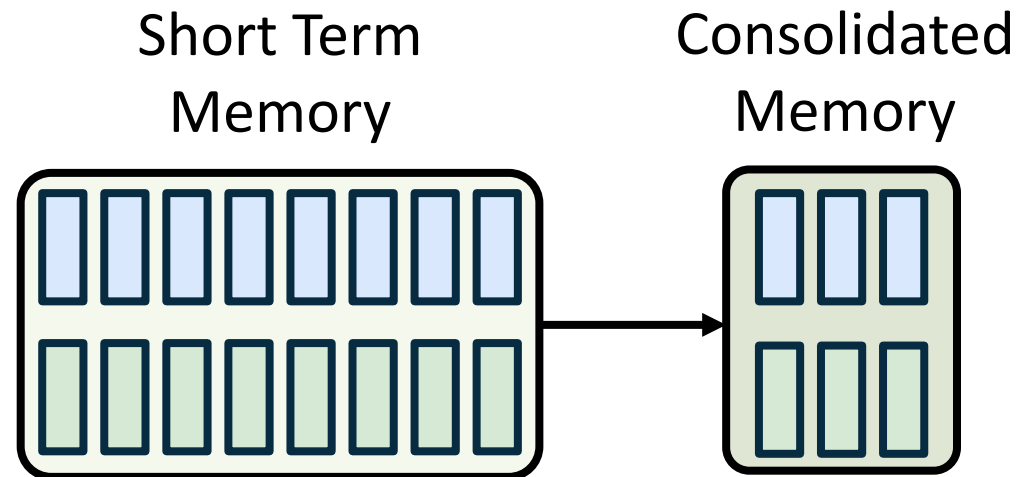
- Transform raw input into rich, multimodal representations, mirroring the entorhinal cortex's role.
- A tripartite strategy processes visual, auditory, and cross-modal features. We use pre-trained models like ImageBind for embeddings and Whisper for transcriptions to form a detailed “**Short Term Memory**” object for each segment.



From Detailed Traces to Semantic Gist

- **Memory Consolidation**

- To reduce redundancy and interference by filtering similar consecutive memory segments, inspired by biological memory stabilization.
- A filtering mechanism based on semantic similarity is used.

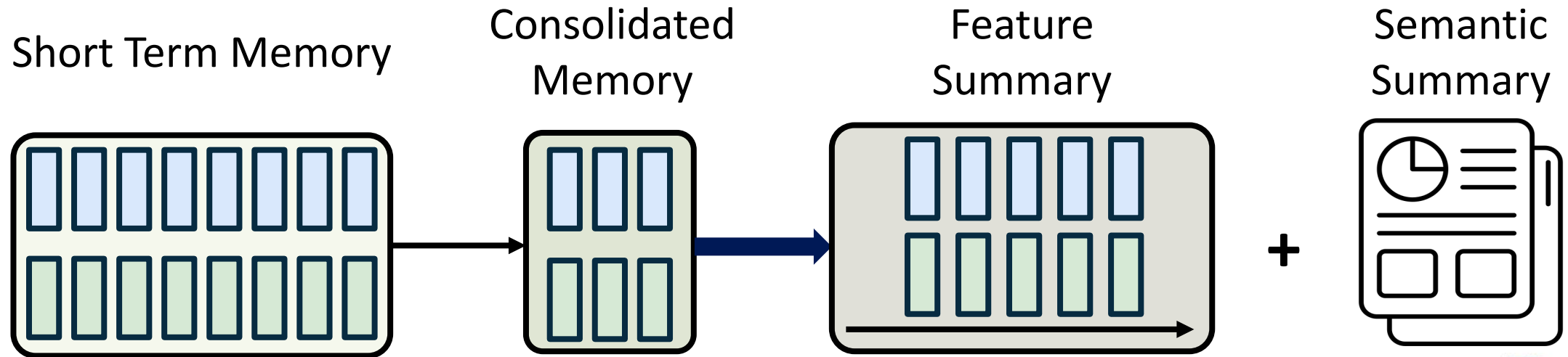


From Detailed Traces to Semantic Gist

- **Memory Consolidation**

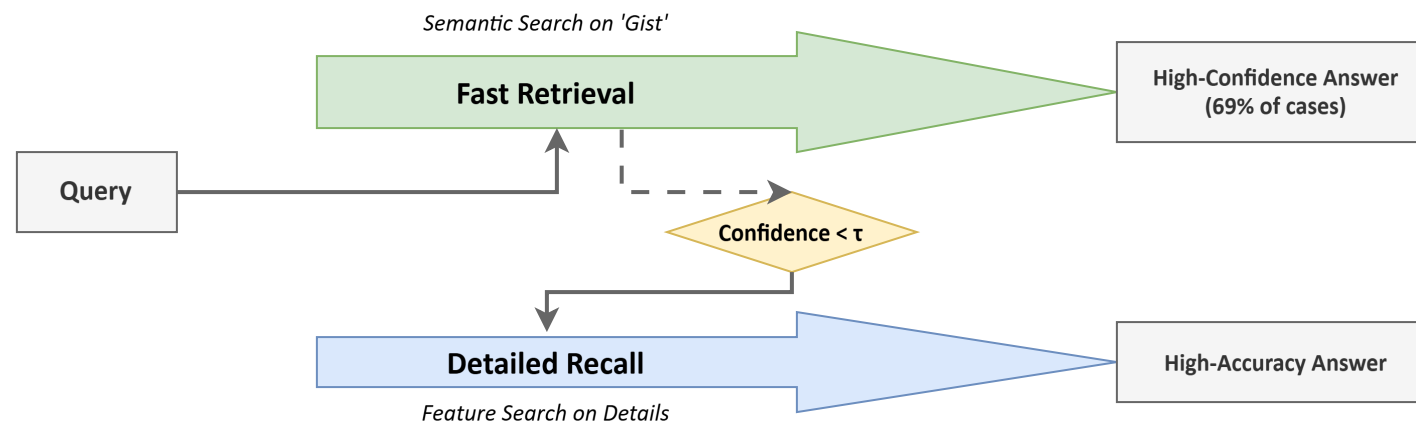
- **Semantic Replay**

- Transform detailed, consolidated short-term memories into efficient long-term representations.
- An LLM “replays” the essence of each consolidated memory segment to generate a high-level textual summary or “gist”. This summary becomes the core of a **ThetaEvent** object, the system's abstract long-term memory format.



Query and Cross-Modal Recall

- Retrieval begins by classifying the query's modality focus (e.g., Visual, Auditory, Semantic) to select the optimal retrieval pathway.
- The system first attempts a highly efficient **Fast Retrieval** on the abstract, semantic “ThetaEvent” summaries.
- Only if confidence from the fast path is below a threshold does the system invoke a **Detailed Recall** on the richer, consolidated Short-Term Memory objects.
- A final LLM-based module synthesizes the retrieved evidence into a coherent answer.



Performance and Impact of HippoMM

- HippoMM achieves 14% higher accuracy while delivering responses 5× faster than state-of-the-art method. PT: Processing Time; ART: Average Response Time; A+V: Cross-modal (Audio+Visual) accuracy; A: Audio-only accuracy; V: Visual-only accuracy; S: Semantic understanding accuracy, DR: Detailed Recall, FR: Fast Retrieval; AR: Adaptive Reasoning.

Method	PT ↓	ART ↓	Modality Performance				Avg. Acc. ↑
			A+V ↑	A ↑	V ↑	S ↑	
<i>Prior Methods</i>							
NotebookLM	–	–	28.40%	23.20%	28.00%	26.80%	26.60%
Video RAG	9.46h	112.5s	63.6%	67.2%	41.2%	84.8%	64.2%
<i>Ablation Studies</i>							
HippoMM w/o DR, AR	5.09h	4.14s	66.8%	73.2%	60.4%	90.0%	72.6%
HippoMM w/o FR, AR	5.09h	27.3s	72.0%	80.0%	66.8%	83.2%	75.5%
HippoMM w/o AR	5.09h	<u>11.2s</u>	68.8%	<u>80.8%</u>	<u>65.6%</u>	<u>92.0%</u>	<u>76.8%</u>
HippoMM (Ours)	5.09h	20.4s	<u>70.8%</u>	81.6%	66.8%	93.6%	78.2%

Lessons Learned and The Path Forward

Key Findings

- **Validated the Approach:** Proved that translating neuroscientific memory principles into computational architectures is a viable and effective strategy.
- **Set a New State-of-the-Art:** Significantly outperformed existing methods in both accuracy and speed on a challenging audio-visual retrieval task.

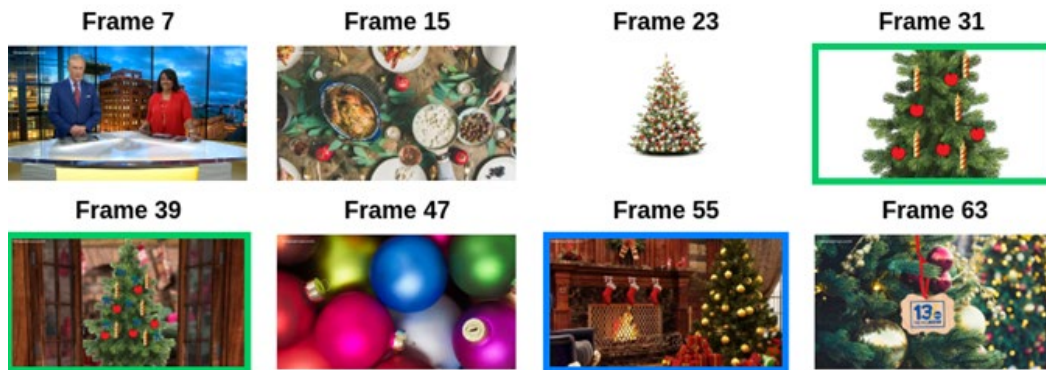
The Path Forward


- **Encoding Efficiency:** How can we more intelligently select what information is encoded to begin with?
- **Memory Scalability:** How do we consolidate and compress memories to manage massive, long-term knowledge bases?
- **Retrieval Speed:** How can we accelerate search to find the right memory in near-instant time?



Encode: Sculpting What Enters the Memory

How can we more intelligently select what information is encoded to begin with?

KVTP: Keyframe Oriented Video Token Pruning



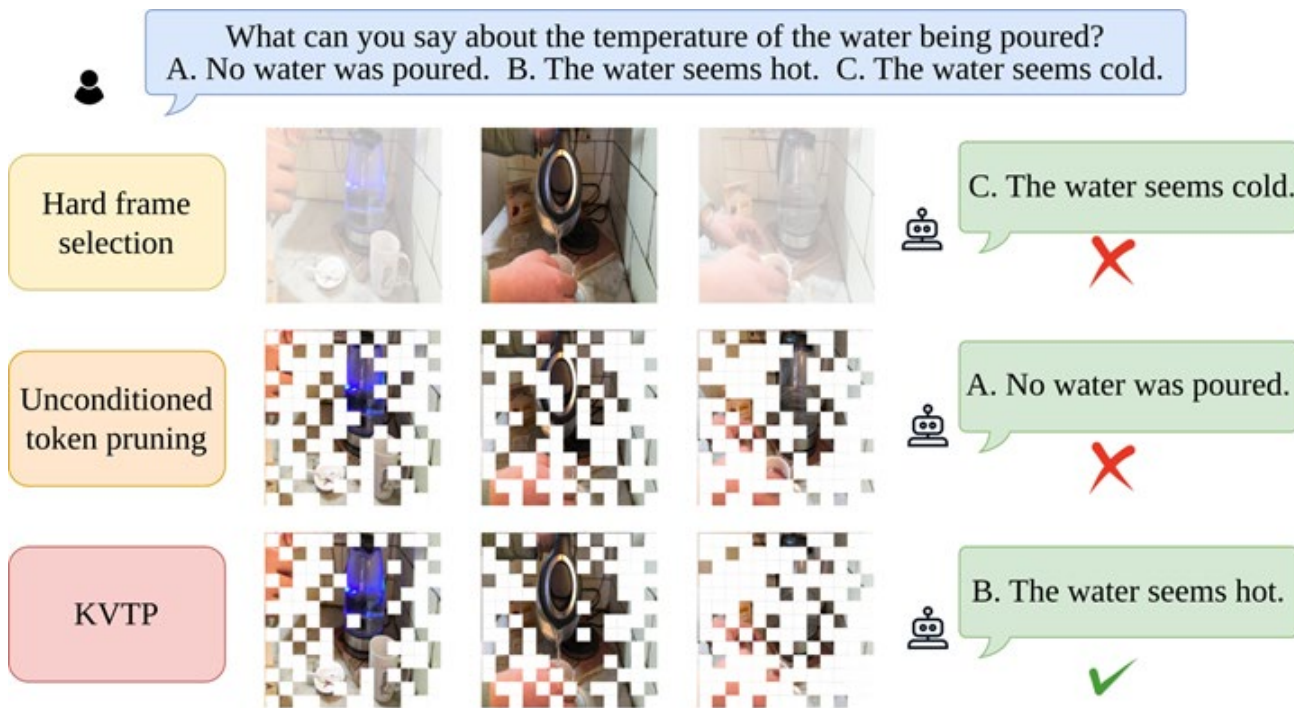
All frame required	
	<div>What is the genre of this video?</div>

Keyframe required	
	<div>When demonstrating the Germany modern Christmas tree is initially decorated with apples, candles and berries, which kind of the decoration has the largest number?</div>
	<div>How many red socks are there on the fireplace?</div>

VLM: Vision Language Model

- Large number of vision tokens is the main bottleneck for efficient VLM, resulting in large memory overhead which cannot be afforded by edge devices
- High spatial and temporal redundancy exist in long-form videos, vision tokens can be pruned without sacrificing performance
- Too many frames, but not all frames are equally important or relevant for answering the queries in one video

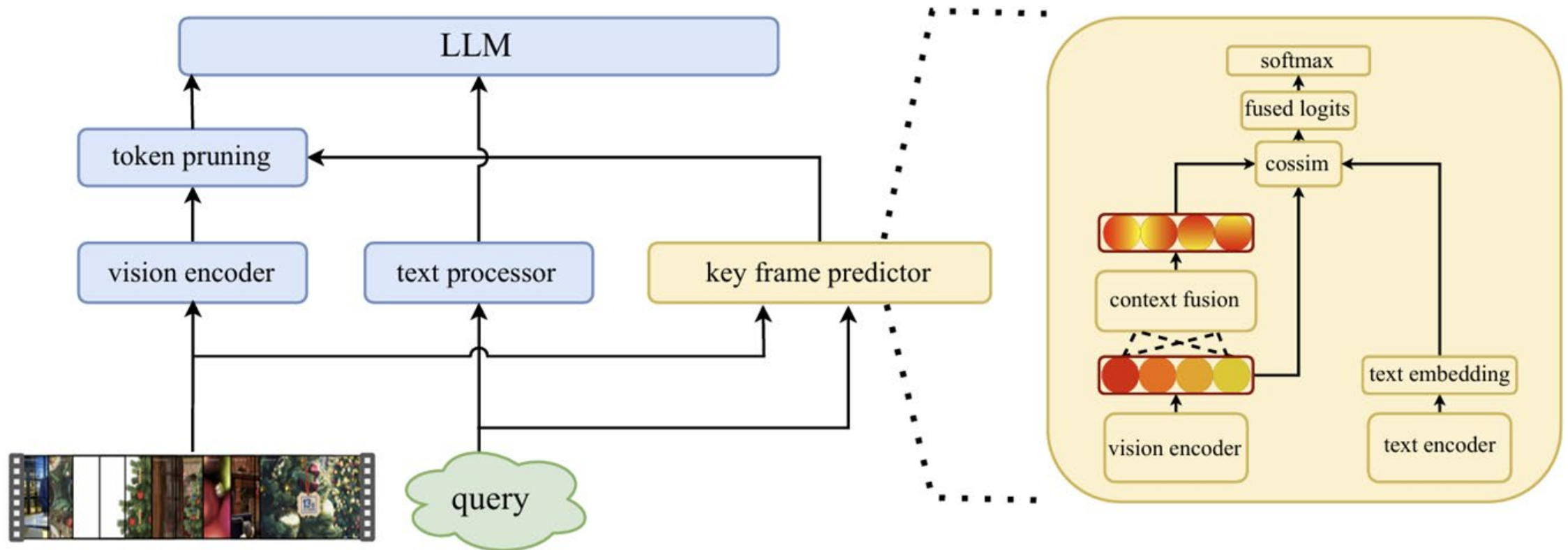
Comparing with existing works



A qualitative result showing the difference between our approach and existing approaches (Hard frame selection and Unconditioned token pruning)

- **Hard frame selection:** only preserve keyframes and discard all other frames, breaking the temporal and context connections
- **Unconditioned token pruning:** Treat every frame equally and assign same pruning rate, causing significant information loss in keyframe.
- **KVTP:** Introducing adaptive pruning rates oriented by frame importance, keeping context connections while preserving more information from keyframes.

A learnable plug-in-and-play keyframe predictor module



Experiment results

- KVTP reduces visual token usage by 80% for long-video processing with no compromise in model performance.

Table 2. Performance comparison of all pipelines on LLaVA-Video-7B.

Method	FLOPs	VideoMME	EgoSchema	NextQA
LLaVA-Video-7B	x100%	62.63	54.17	78.51
Random Sampling	x28%	58.28	50.69	75.20
ToMe	x30%	58.90	51.45	72.19
PruMerge	x28%	59.77	52.49	75.58
KeyVideoLLM	x36%	51.32	46.78	64.33
FastV	x64%	61.79	52.42	77.89
Random Sampling + Ours	x36%	60.16	52.73	76.50
ToMe + Ours	x38%	<u>62.36</u>	<u>53.24</u>	75.88
PruMerge + Ours	x36%	63.29	54.71	<u>76.76</u>

Table 3. Performance comparison of all pipelines on LLaVA-Video-72B.

Method	FLOPs	VideoMME	EgoSchema	NextQA
LLaVA-Video 72B	x100%	69.52	65.76	83.20
Random Sampling	x21%	62.37	60.47	78.93
ToMe	x21%	62.89	61.22	76.45
PruMerge	x21%	64.52	63.11	80.74
KeyVideoLLM	x23%	60.49	55.23	76.45
FastV	x56%	<u>66.25</u>	63.56	<u>80.34</u>
Random Sampling + Ours	x23%	64.32	62.19	80.12
ToMe + Ours	x23%	65.77	<u>63.61</u>	79.51
PruMerge + Ours	x23%	67.34	64.12	81.21

Table 4. Comparison of Different Methods for Assigning Adaptive Pruning Rates.

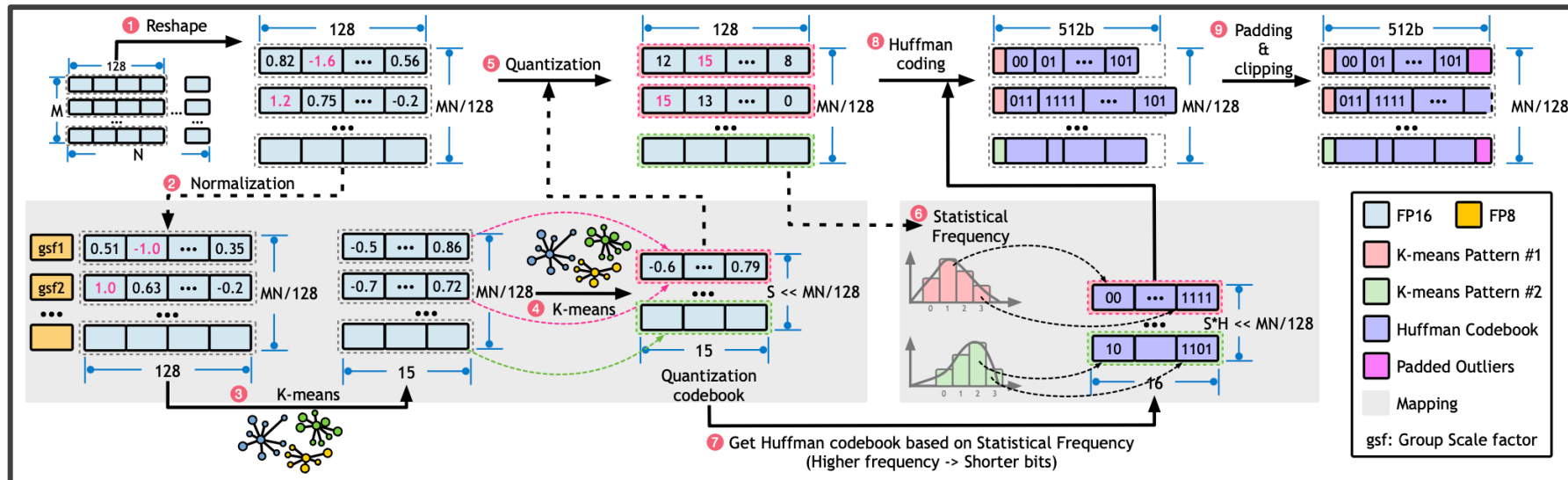
Method	# of Trained Parameters	VideoMME	NextQA	EgoSchema
GPT Assigned	0	62.83	75.96	53.33
KeyVideoLLM	7.88B	61.23	75.80	51.91
KVTP	0.88B	63.29	76.76	54.71

Consolidate: Entropy-Driven Compression

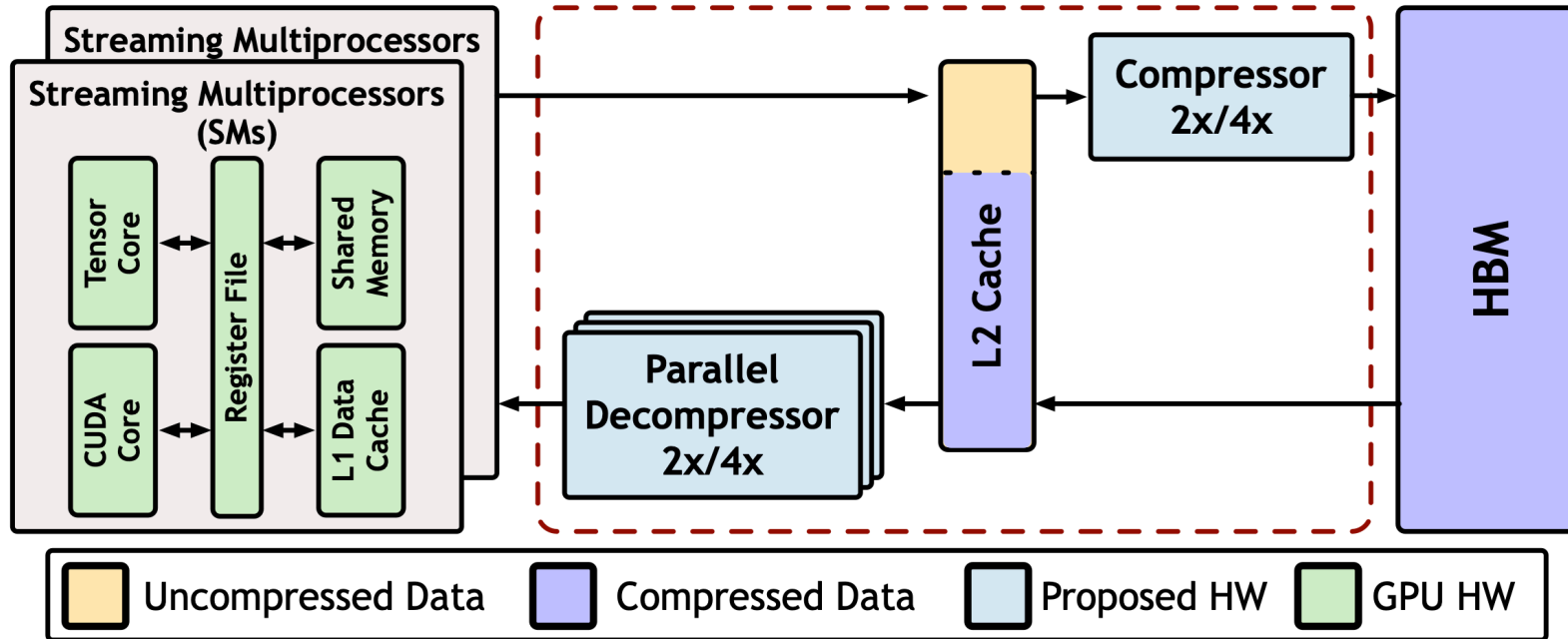
How do we consolidate memories to manage massive, long-term knowledge bases?

Ecco Algorithm Design

- Stage 1 (Steps 1,2,3,4) – Pattern Discovery (Hierarchical K-Means)
 - K-Means per 128-weight chunk \Rightarrow local centroids
 - Cluster those centroid vectors again \Rightarrow S global K-means patterns \rightarrow pattern map
- Stage 2 (Steps 5,6,7,8,9)– Pattern-Aware Compression
 - Quantize each chunk with suitable shared K-means patterns
 - Entropy-code centroid indices using pattern-specific frequencies (H Huffman codebooks)



Ecco System Design



- Decompressor is placed between the SMs and the L2 cache
- Compressor is positioned between the L2 cache and the HBM

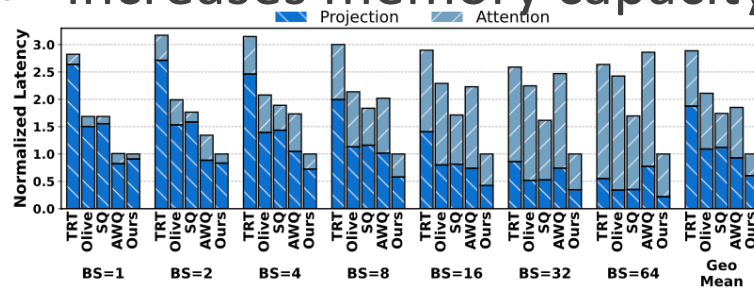
- Mixed SW & HW control
 - Explicitly declare compression properties in *CUmemAllocationProp*
 - TLBs are augmented with additional bits indicating
 - Compressed/Uncompressed
 - Compression ratio (2x/4x)

Evaluation

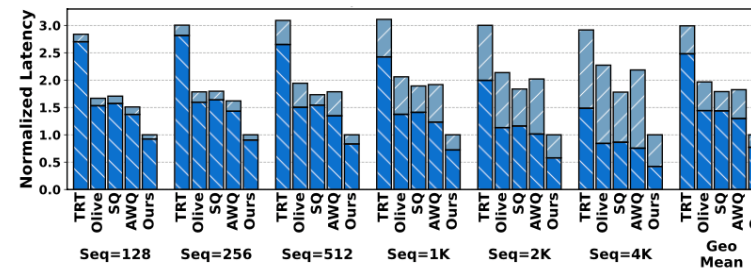
- Accuracy (Perplexity)
- Normalized latency
 - (a) Across batch sizes: 1, 2, 4, 8, 16, 32, 64
 - (b) Across sequence lengths: 128, 256, 512, 1k, 2k, 4k
 - (c) Across various models: 7B, 13B, 30B, 65B, 70B
- Achieves an up to $2.9\times$ and $1.9\times$ speedup over the SOTA AWQ and SmoothQuant framework, $2.4\times$ over the Olive accelerator
- Increases memory capacity by nearly $4\times$ and maintaining SOTA LLM accuracy.

Table 1: Perplexity Comparison of Models Under Different Configurations on WikiText-2 with 2048 Sequence Length.

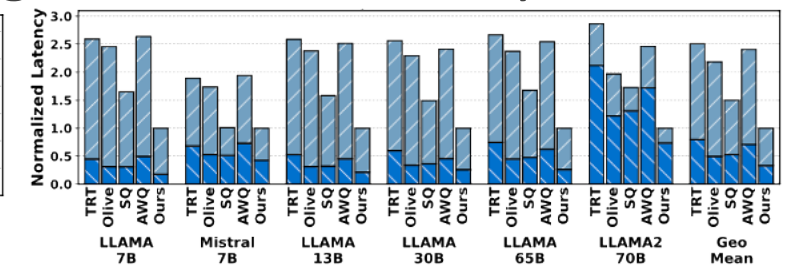
Perplexity ↓		LLaMA			LLaMA-2			Mistral
Bits	Method	LLaMA-7B	LLaMA-13B	LLaMA-30B	LLaMA-2-7B	LLaMA-2-13B	LLaMA-2-70B	Mistral-7B
FP16	-	5.68	5.09	4.10	5.47	4.88	3.32	5.25
W4A16 g128	GPTQ-R	5.83	5.20	4.22	5.63	4.99	3.43	5.39
	Olive	6.04	5.38	4.32	5.81	5.10	3.43	5.51
	AWQ	5.78	5.19	4.21	5.60	4.97	3.41	5.37
	Ecco	5.80	5.17	4.20	5.58	4.97	3.40	5.36
W4A8KV4 g128	RTN	6.23	5.46	4.56	5.99	5.19	3.70	5.59
	AWQ	5.93	5.36	4.39	5.83	5.12	3.51	5.50
	QuaRot	5.91	5.26	4.30	5.71	5.06	3.45	5.39
	QoQ	5.89	5.25	4.28	5.70	5.08	3.47	5.42
	Ecco	5.87	5.22	4.24	5.65	5.03	3.44	5.41



(a) Normalized latency vs. batch sizes on LLaMA-13B.



(b) Normalized latency vs. sequence lengths on LLaMA-13B.



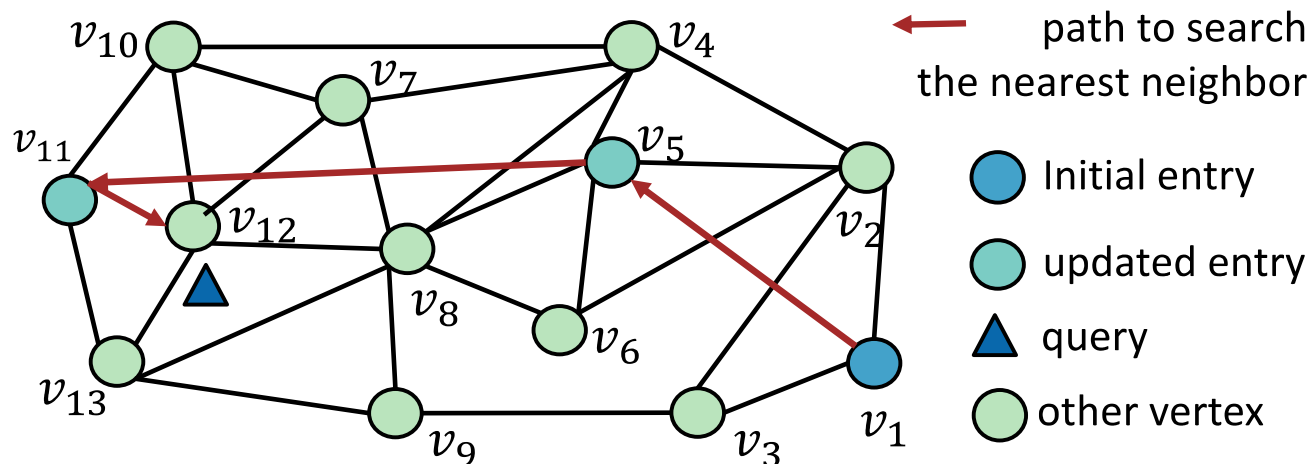
(c) Normalized latency vs. various models.

Recall: In-Memory Search at Fast Speed

How can we accelerate search to find the right memory in near-instant time?

Background - ANNS

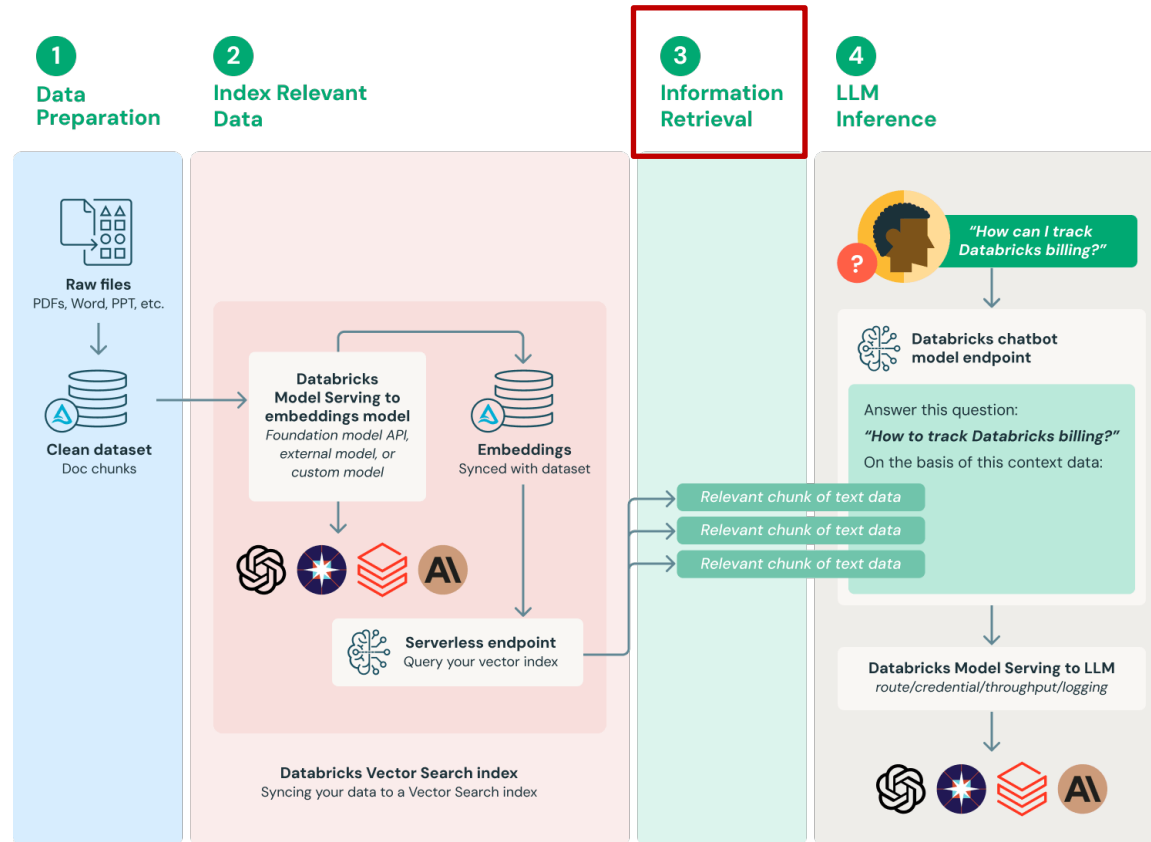
- Graph-traversal-based ANNS
 - Construction phase: building the graph
 - *Search phase*: search targets on the constructed graph
 - Breadth-first traversal search + specialized conditions



- Three basic kernels
 - Graph-traversal
 - $v_1 \rightarrow (v_2, v_3, v_5) \rightarrow (v_4, v_6, v_8, v_{11}) \rightarrow \dots$
 - Termination condition
- Distance Computation
 - Angular/Euclidean Distance
 - Candidate list
- Sorting
 - Top-k nearest neighbors

Background – ANNS Applications

- Retrieval augmented generation (for LLM)



Definition:

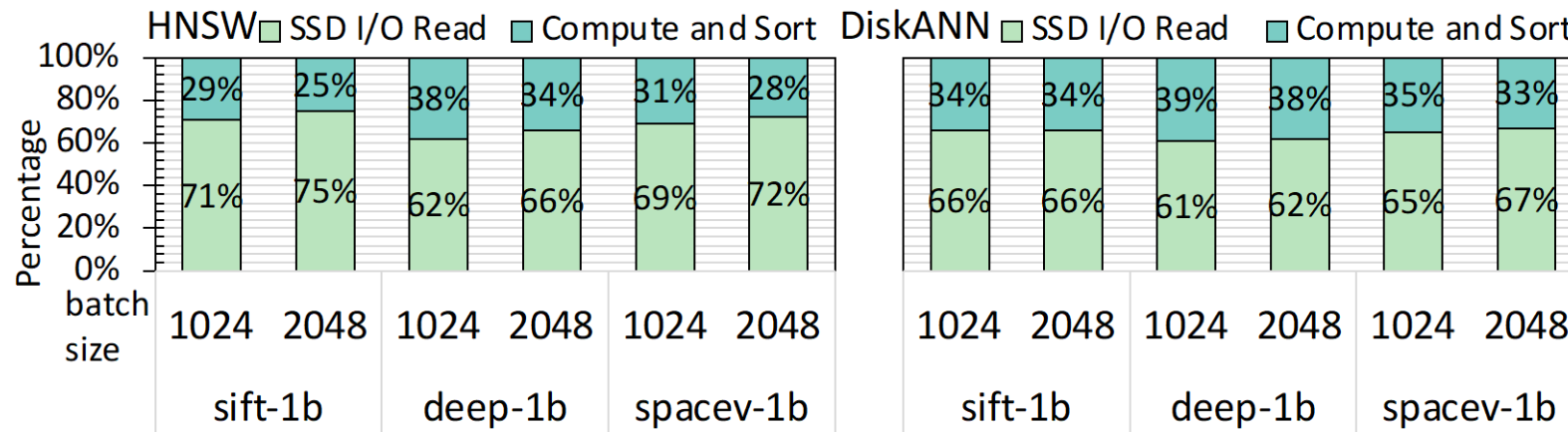
RAG takes input and retrieves a set of relevant/supporting documents given a source (e.g., Wikipedia). The documents are concatenated as context with the original input prompt and fed to the text generator which produces the final output.

Figure source: <https://www.databricks.com/glossary/retrieval-augmented-generation-rag>

Motivation – Some Profiling Results

- SSD I/O overhead

- Running HNSW and DiskANN on the graphs constructed on 3 datasets

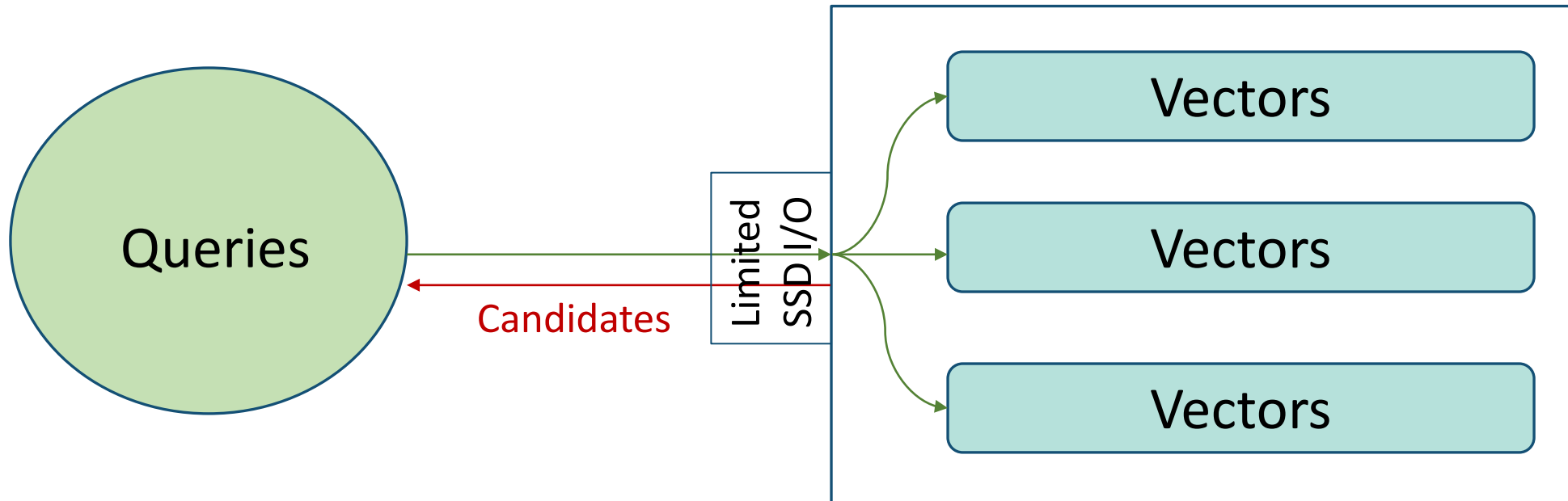


- Why?

- Graph too large (> 500GB) → redundantly/frequently fetch partitioned graphs from SSD
- Limited SSD I/O bandwidth

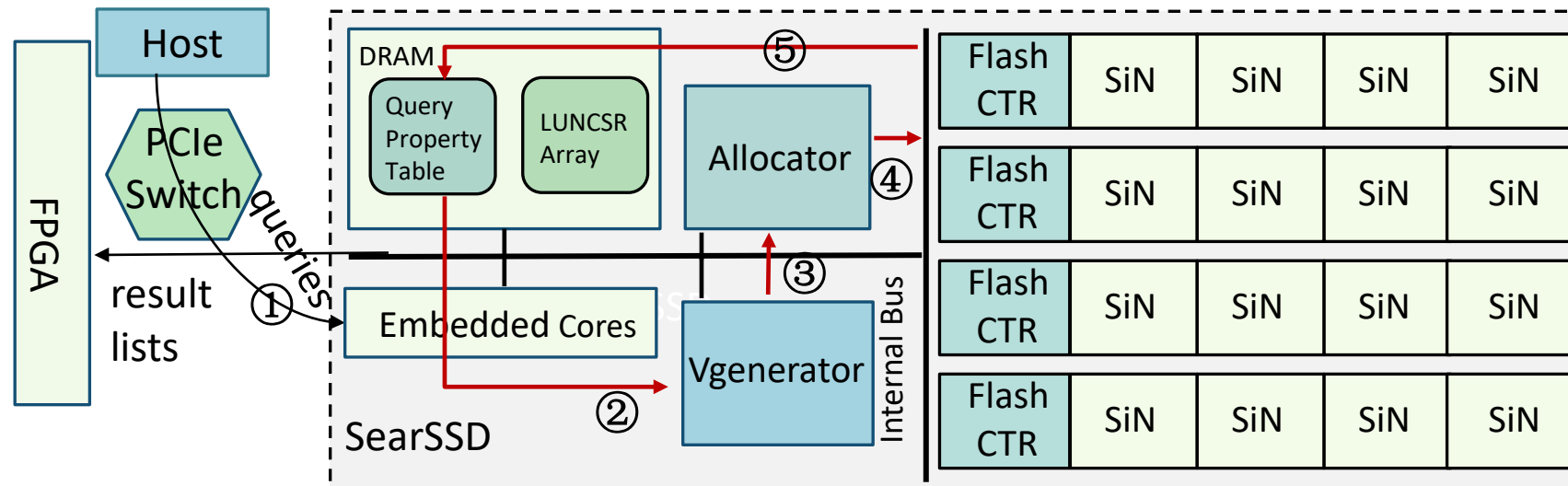
Design – In-Storage Computing Principle

- Make queries swim upstream and filter the stream from storage



Design – Hardware Architecture

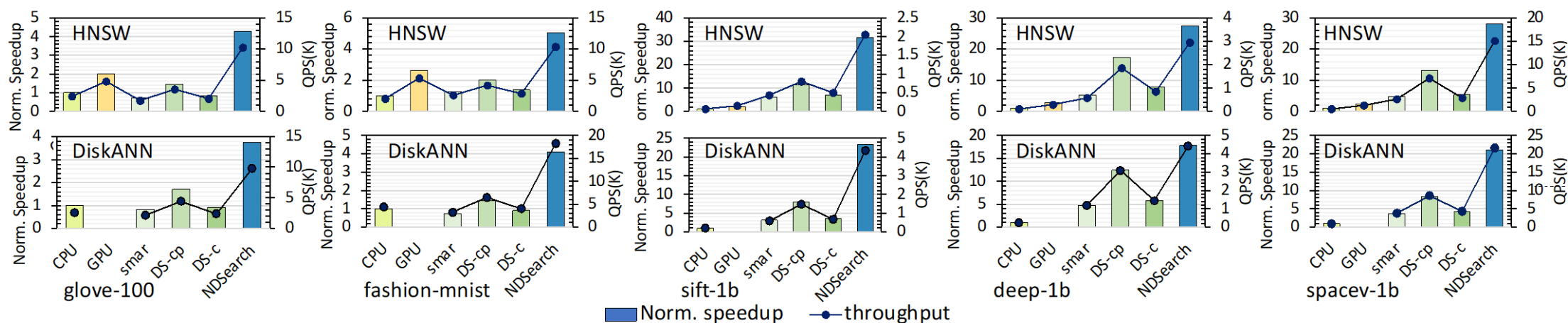
- Overall NDSearch architecture



- SearchSSD: Graph-traversal + Distance Computation
- FPGA: Sorting (Bitonic Sorting)

Evaluation

● Main Results



Speedup normalized to CPU (shown in the histogram) and throughput (shown in the line chart) comparison on various platforms. We measure the throughput by processing a batch (2048) of queries with the same memory trace on each benchmark.

NDSearch can improve the throughput by up to $31.7\times$, $14.6\times$, $7.4\times$, $2.9\times$ over CPU, GPU, a state-of-the-art SmartSSD design and DeepStore, respectively.

Our Takeaways

- AI will proactively retrieve information using contextual cues, before being queried.
- AI systems will strategically “forget” redundant data, keeping only essential patterns like biological memory.
- New architectures will merge memory and processing, eliminating many exiting computing bottlenecks.



Q&A

Duke