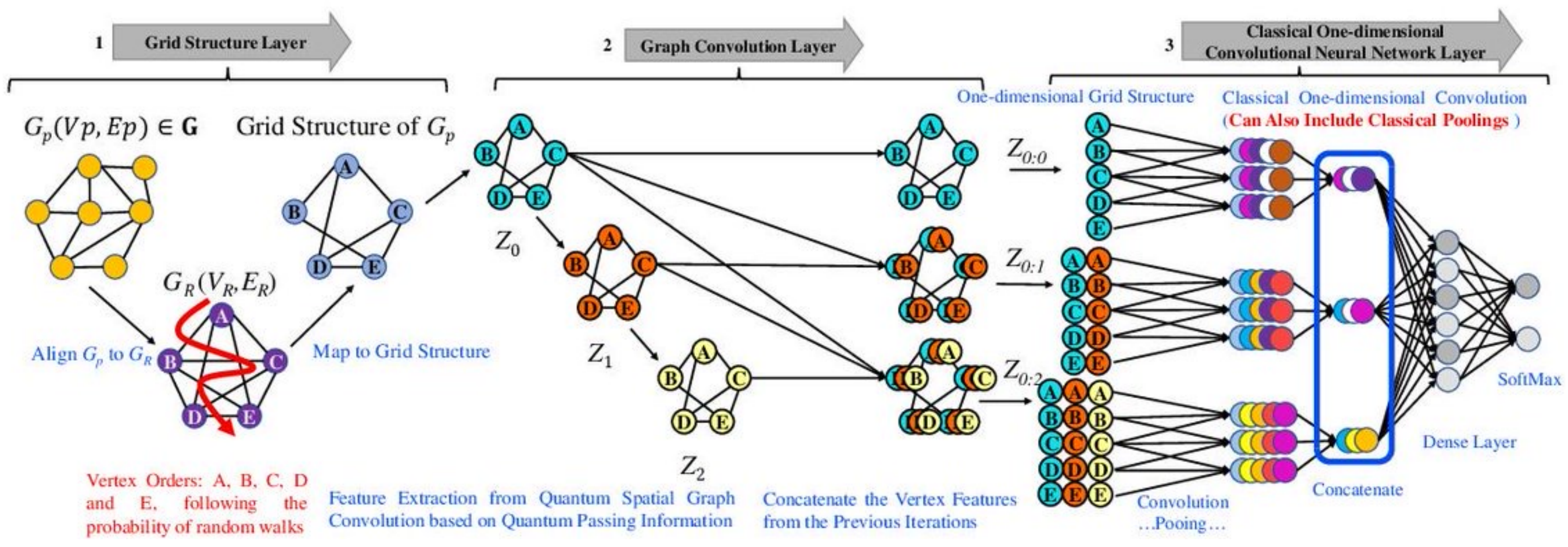
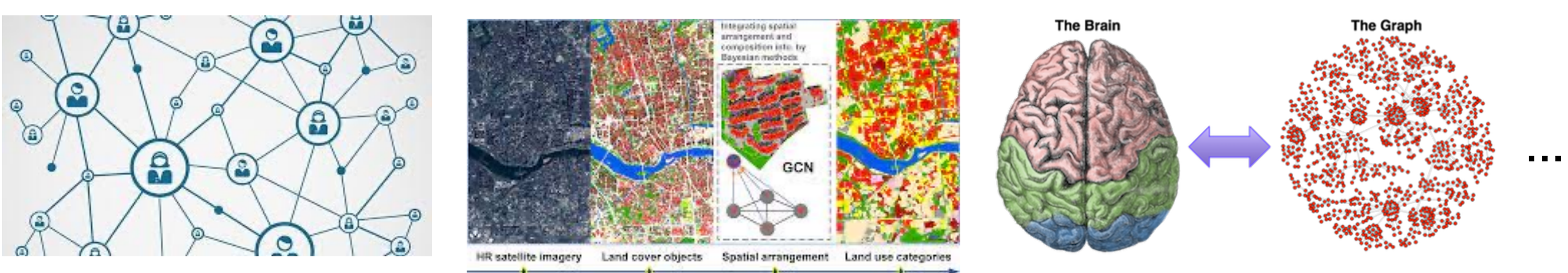
GraphGem: Optimized Scalable System for Graph Convolutional Networks

Advitya Gemawat
Halıcıoğlu Data Science Institute, University of California San Diego

1. Background & Motivation

GCNs *expand* CNN's convolution operation to:
- work with data in non-Euclidean spaces,
- model complex 'long-range' dependencies and network embeddings,
- offer endless real-world applications with promising results!



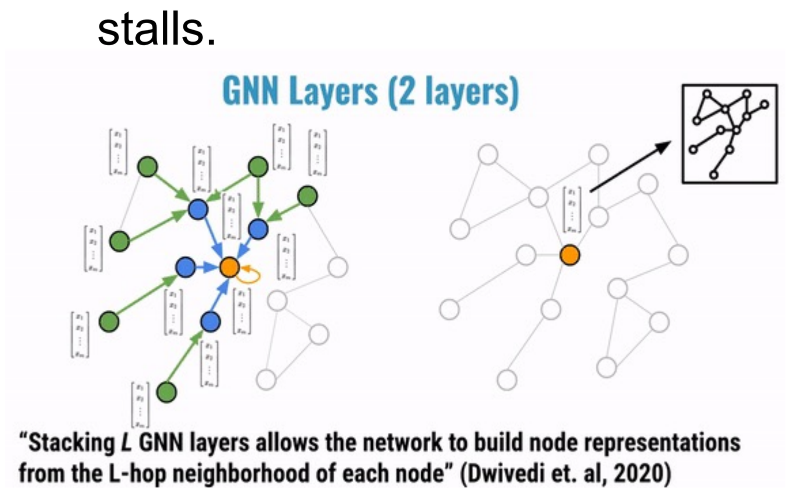


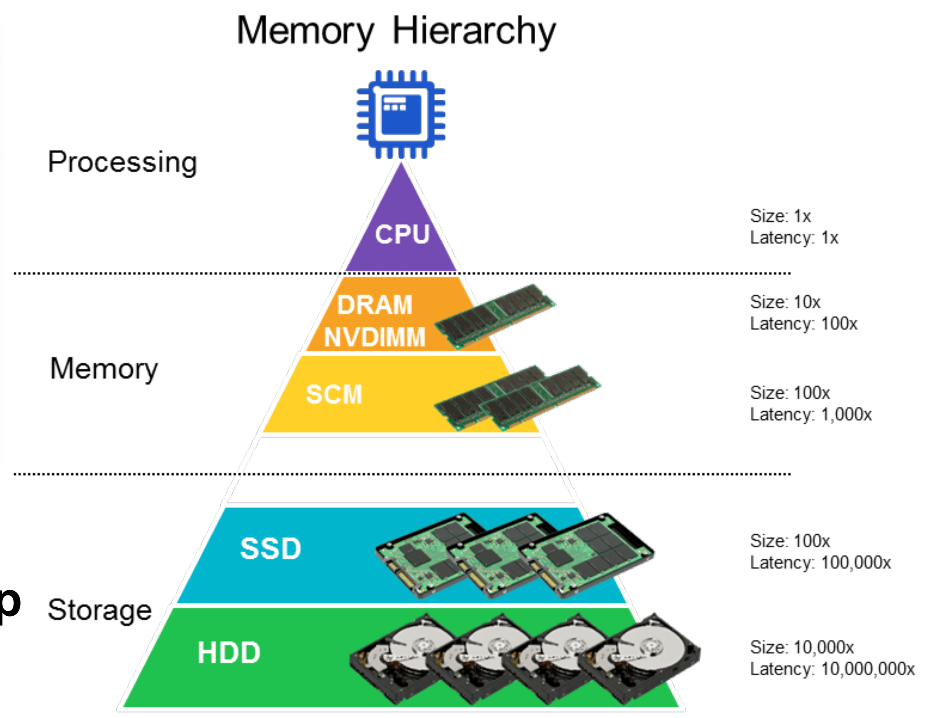
Social Networks | Geospatial Analysis | Cognitive Science

2. Challenges

- Updating a node's value involves I/O cost to read its neighbors (and more I/O for their neighbors..) and write the updated value.

- Large graph and feature matrix forces data reads to go to a lower memory level, resulting in large *random access*, increasing I/O costs and memory stalls.





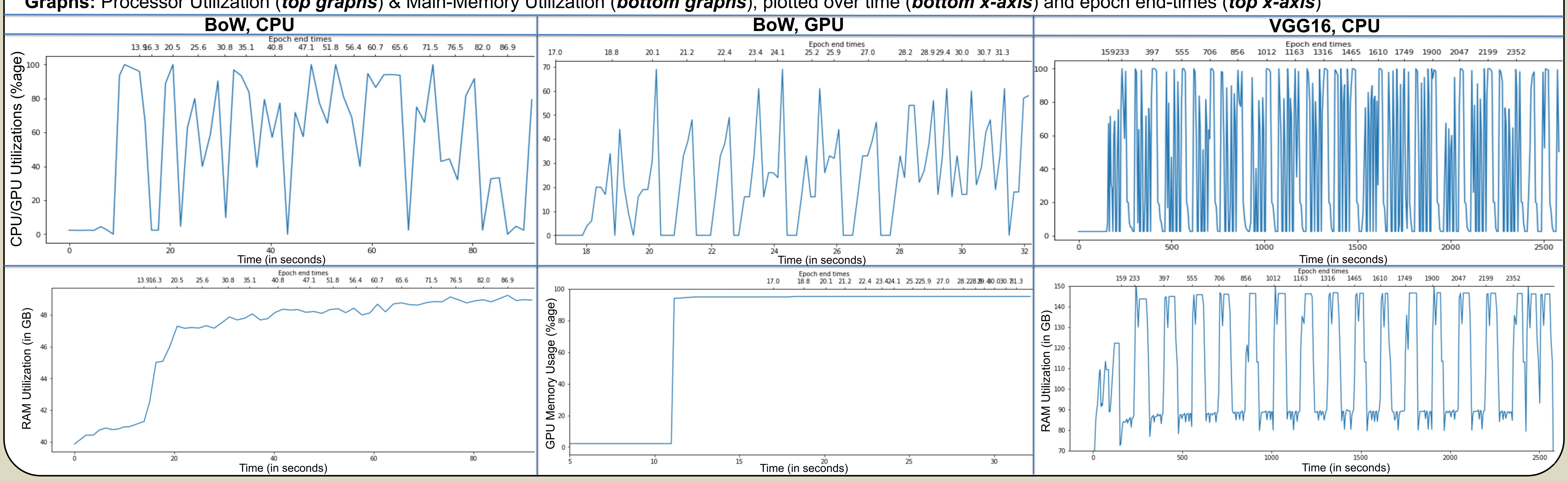
Data access pattern unlike almost all other forms of Deep Learning!

- Most GCN literature utilizes data featurizations which can be easily stored in DRAM / GPU memory!

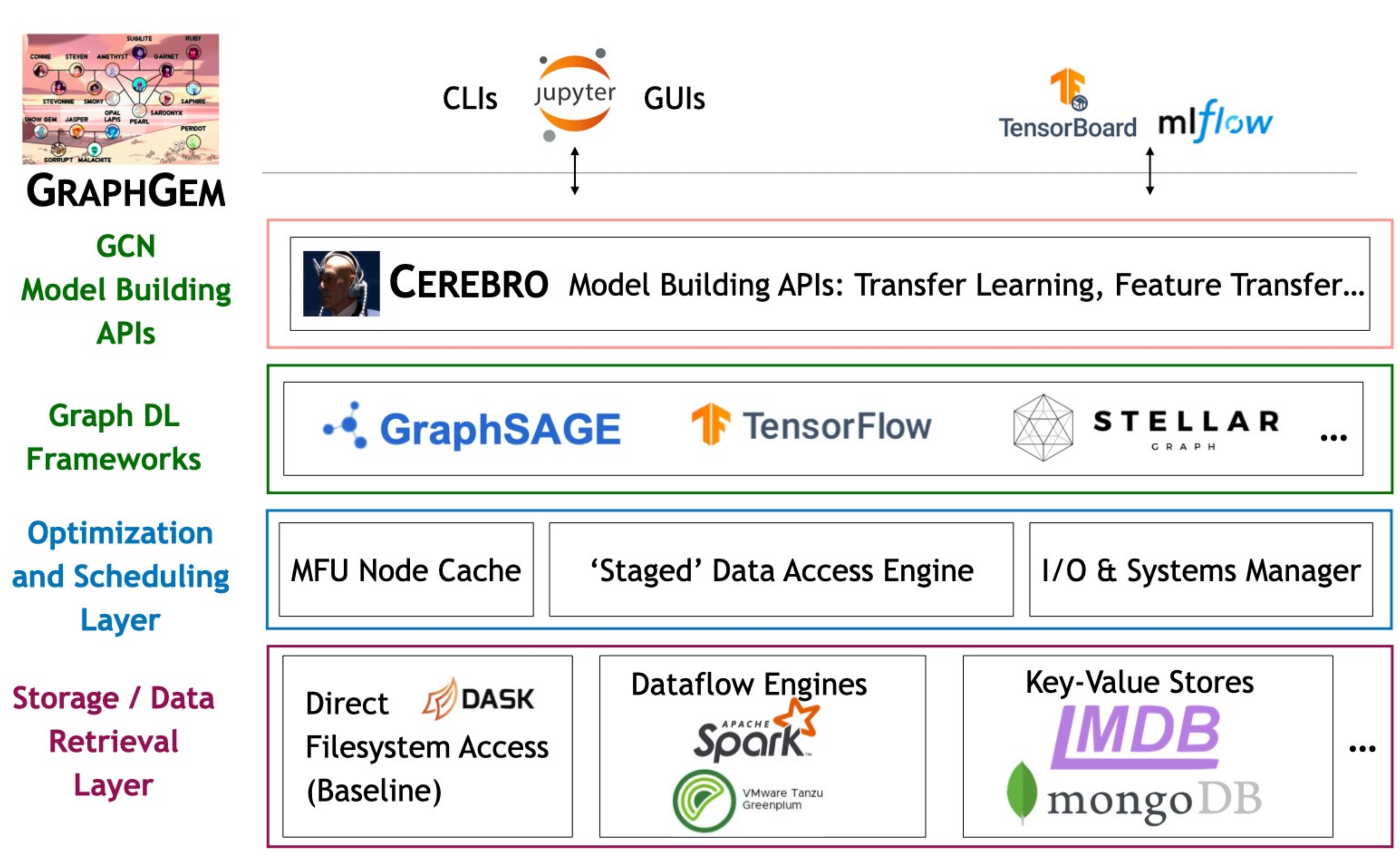
3. Experiments

Dataset: 89,250 Flickr images | **Task:** Multi-class image (node) classification **| GCN Framework:** 2-layer GraphSAINT with Random Walk sampler using TF
Featurizations: ***BoW*** **–** 500-dim bag of words of textual captions [341 MB], ***VGG16*** **–** 100,352-dim *block_5_conv_3* layer of image features [34 GB]
Machine Hardware: 1 NVIDIA *12GB* PCI P100 GPU, 2 Intel Xeon Silver 4114 10-core CPUs, 192 GB RAM
Graphs: Processor Utilization (***top graphs***) & Main-Memory Utilization (***bottom graphs***), plotted over time (***bottom x-axis***) and epoch end-times (***top x-axis***)

VGG16, GPU: OOM!



4. Proposed Architectural Stack

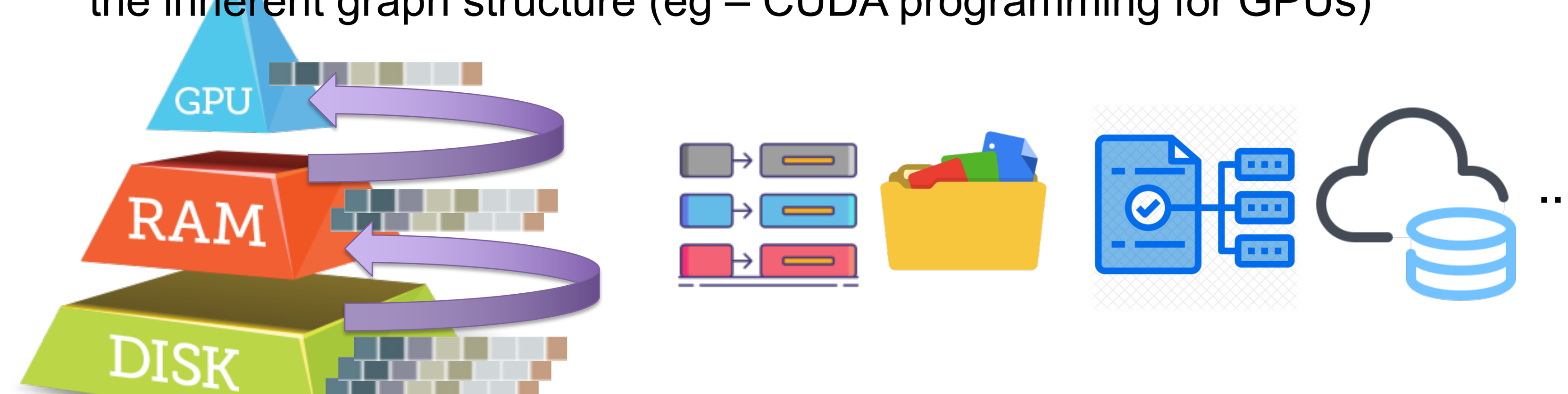


5. Future Work

- **Baseline:** Make the VGG16 feature variant work on GPU by *spilling* data to DRAM and reading it to GPU as needed

- **Additional Metrics:** *DRAM-to-GPU* network and I/O *traffic*

- **Optimization Layer:** optimally 'stage' mini-batches to processor based on the inherent graph structure (eg – CUDA programming for GPUs)



- **System Extension:** extract features that exceed RAM, spill data to disk, and monitor performance across *multiple memory levels*

- **Storage Extensions:** *Diversify experimentation* with key-value stores, multiple GPUs, distributed set-ups etc.

References: Arun Kumar et al. 2021. "Cerebro: A Layered Data Platform for Scalable Deep Learning." In CIDR.;
Zeng, Hanqing et al. "GraphSAINT: Graph Sampling Based Inductive Learning Method." ArXiv abs/1907.04931 (2020): n. pag.