GNNs Beyond HBM: Compressed, Bit-Exact Feature Streaming for Out-of-Core Training

Rajni Parshuram Pawar Luke Logan Xian-He Sun Anthony Kougkas
rpawar4@hawk.illinoistech.edu, llogan@illinoistech.edu, sun@illinoistech.edu, and akougkas@illinoistech.edu

Gnosis Research Center @ ILLINOIS TECH

Motivation — compress the out-of-core feature stream

Training GNNs whose feature matrix exceeds GPU **high-bandwidth memory (HBM)** — `ogbn-papers100M` is $\sim$57 GB, beyond a commodity 8–40 GB card — is already done by **out-of-core feature streaming** from host memory across the PCIe bus (GIDS, Ginex, MariusGNN, GraphBolt, HMGraph). All stream features **uncompressed**. Error-bounded scientific compressors (SZ, ZFP; GPU-native cuSZp) and GPU-side decompression are routine in HPC checkpoint/I/O — but unused on the GNN feature-streaming path.

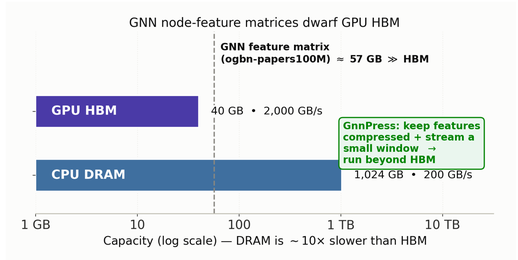


Figure 1: A GNN feature matrix is far larger than GPU HBM; host DRAM is bigger but $\sim$10$\times$ slower.

- **Opportunity:** *compress* the streamed feature store — fewer bytes over PCIe, smaller host footprint — *if* it stays exact and accuracy-neutral.
- **GnnPress:** a bounded, **compressed** feature window — **bit-exact** lossless (`zstd`) or **3.13**$\times$ lossy (cuSZp) — at **no accuracy cost**, on a real A100.
- Peak GPU is bounded by the window, not the matrix, so training runs on feature matrices far larger than HBM.

Approach — GnnPress compressed feature store

GnnPress stores the feature matrix **compressed** and streams only a small window through HBM, so the **feature** working set is bounded by the window, **not by HBM capacity**. The GNN sees an ordinary feature array.

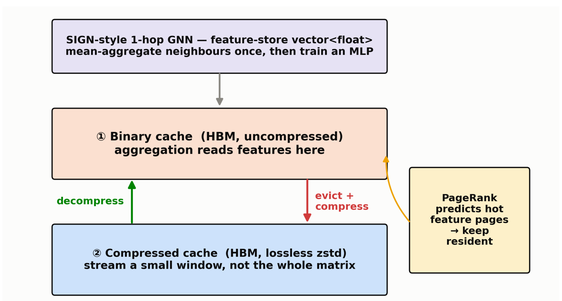


Figure 2: A two-level HBM cache for GNN features: aggregation reads the uncompressed binary cache; the compressed cache holds more in the same HBM. PageRank keeps the hot feature pages resident.

- **Compressed cache.** Level moves are (de)compress, not copies. Lossless `zstd` is modest on dense floats (**1.08**$\times$); GPU lossy **cuSZp reaches 3.13**$\times$ at no training cost (Result 1).
- **One lossless codec (`zstd`).** Features are **bit-identical** on read $\Rightarrow$ GNN output provably unchanged.
- **Async streaming window.** Prefetch/evict double-buffer hides decompression behind compute; the compressed matrix spills to host DRAM, so the **resident window is** O(**window**) (64 MiB), independent of matrix size.

Result 1 — The novelty: compressed feature streaming

Feature-store compression on the real **52.9 GiB** (56.8 GB) ogbn-papers100M matrix (A100-SXM4-40GB). Same seed, same data, same 30 epochs.

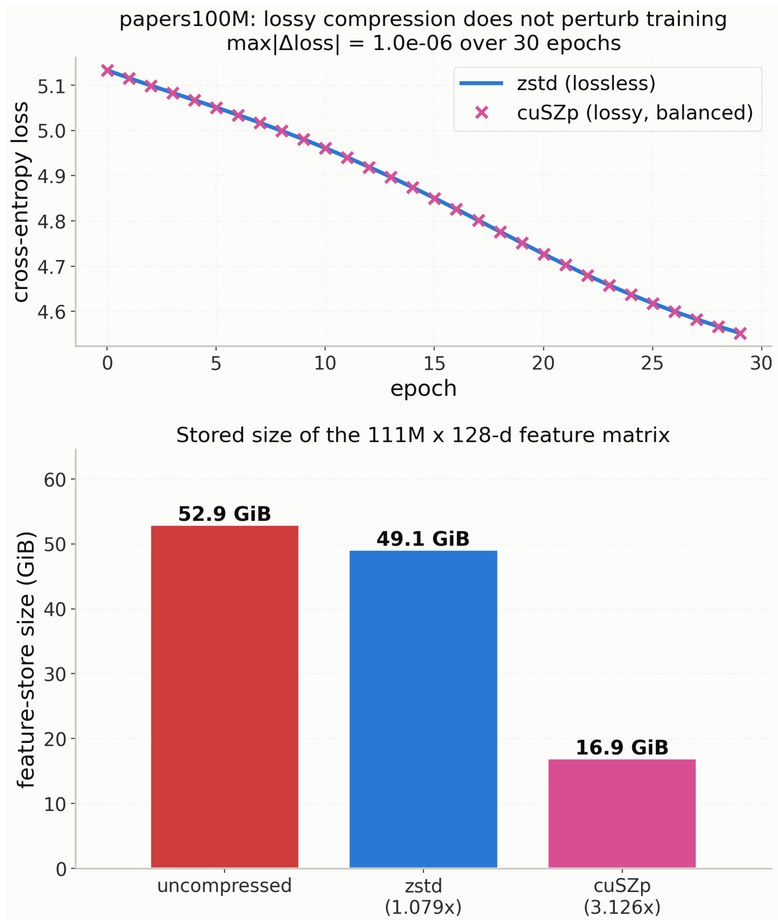


Figure 3: *Full-batch GD, loss-only comparison.* Top: lossy cuSZp gives the **same** training loss as lossless zstd (max|Δloss|=1e-6). Bottom: the 52.9 GiB matrix stores at 49.1 GiB (zstd, 1.08$\times$) vs **16.9 GiB** (cuSZp, **3.13**$\times$).

- **Lossless zstd $\rightarrow$ bit-exact.** On ogbn-arxiv the streamed run is **bit-identical** to in-core every epoch (max|Δloss/acc/val|=0) — and holds under **mini-batch** SGD too (4 configs, max|Δ|=0). Ratio is only **1.08**$\times$ on dense floats, though.
- **Lossy cuSZp $\rightarrow$ 3.13**$\times$, GPU-native (stores *faster*), with the **same training loss** (max|Δloss|=1e-6). Compressing the feature stream is the contribution vs. uncompressed out-of-core GNNs.
- **More graph per host byte.** At equal host budget, cuSZp's **3.13**$\times$ holds $\sim$3$\times$ the feature matrix and moves a third of the bytes over PCIe.

Result 2 — It learns, at bounded GPU memory

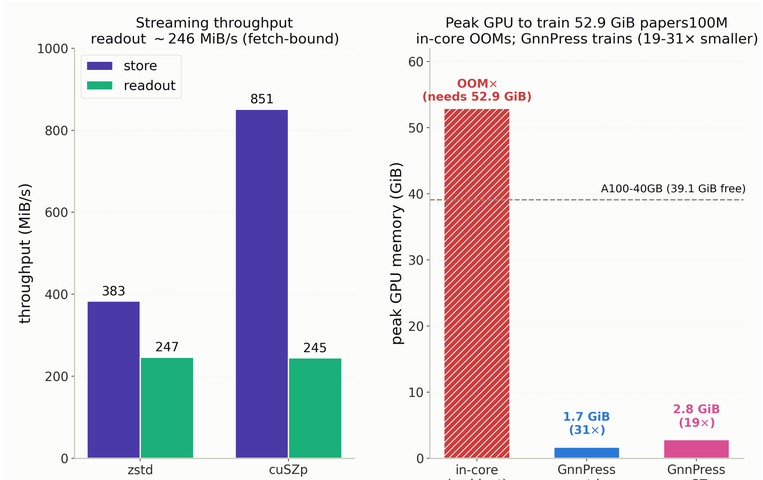


Figure 4: Left: streaming throughput — readout pinned at $\sim$246 MiB/s (fetch-path bound, *not* codec-bound). Right: **measured** peak GPU — in-core needs the whole 52.9 GiB resident (**OOMs** the 40 GB A100); GnnPress trains in **1.7 GiB** (zstd) / **2.8 GiB** (cuSZp).

- **Learns. Mini-batch** SGD trains papers100M to **61.1% / 61.2%** train/val (172 classes) — **9.8**$\times$ the full-batch floor (more updates); mini-batching also **shrinks** the window (32$\rightarrow$2 MiB, arxiv). (Index-split val, *not* OGB's temporal split $\Rightarrow$ not leaderboard-comparable.)
- **Bounded GPU, not speed.** Measured peak **1.7 GiB** (zstd) / **2.8 GiB** (cuSZp) — **31**$\times$ / **19**$\times$ $<$ 52.9 GiB (the GPU codec costs $\sim$1.1 GiB more), dominated by **O(N) labels (847 MiB)**, not the bounded 64 MiB window. Readout $\sim$246 MiB/s ($\sim$100$\times$ below PCIe) $\Rightarrow$ *no* performance claim; no UVA baseline yet.

Conclusion & Future Work

- **Compressed, bit-exact feature streaming** — the novelty vs. uncompressed out-of-core GNN training: lossless `zstd` (bit-exact) or lossy cuSZp (**3.13**$\times$) at **no accuracy cost**.
- **Learns beyond HBM:** mini-batch SGD reaches **61%** on `ogbn-papers100M` (52.9 GiB) at a measured **1.7 GiB** peak GPU (**31**$\times$), where in-core OOMs.
- **Next:** a UVA/zero-copy baseline; multi-layer GNNs (GCN/GAT/GraphSAGE); a compressed *label* store.

References

- GIDS, VLDB'24; Ginex, VLDB'22; MariusGNN, EuroSys'23 — out-of-core feature streaming.
- GraphBolt (DGL'23); HMGraph, ICPP'25 — hierarchical-memory GNN caches.
- OGB, NeurIPS'20 (datasets); SIGN, arXiv:2004.11198 (model).
- `zstd`, RFC 8878 (lossless); SZ, SC'16 & ZFP, LDAV'14, cuSZp, SC'23 (error-bounded lossy); BaM, ASPLOS'23 (GPU-initiated storage).