Communication Avoiding Parallel Sketching via Sparse Subspace Embeddings

Hussam Al Daas[1], Grey Ballard[2], Laura Grigori[3,4], Md Taufique Hussain[2]
Suraj Kumar[5], Johannes Laute[4], Kathryn Rouse[6]

[1]Rutherford Appleton Laboratory [2]Wake Forest University [3]EPFL
[4]Paul Scherrer Institute [5]Inria Lyon [6]Inmar Intelligence

SC26

Randomized sketching

- Compress a matrix: $A \in \mathbb{R}^{n_1 \times n_2} \rightarrow B = \Omega A \in \mathbb{R}^{\ell \times n_2}, \quad \ell \ll n_1$.
- Used in least squares, low-rank approximation, Krylov methods.
- Ω is drawn independently of A (oblivious subspace embedding).

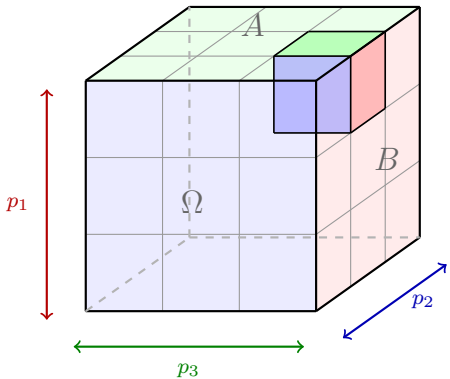
For a k-dimensional subspace $\mathcal{S} \subset \mathbb{R}^{n_1}$, with probability at least $1 - \delta$:

$$(1 - \varepsilon)\|x\|_2 \leq \|\Omega x\|_2 \leq (1 + \varepsilon)\|x\|_2 \qquad \forall x \in \mathcal{S}.$$

The required ℓ depends on k, ε, δ and on the distribution of Ω.

This talk: the communication cost of computing $B = \Omega A$ in parallel.

Parallel matrix multiplication



$$\Omega A = B$$

1. All-Gather Ω and A along grid fibers.
2. Multiply local blocks.
3. Reduce-Scatter B.

Each processor owns a block of the iteration space; it communicates the parts of Ω, A, B on the faces of its block.

Agarwal et al., *A three-dimensional approach to parallel matrix multiplication*, IBM J. Res. Dev., 1995.

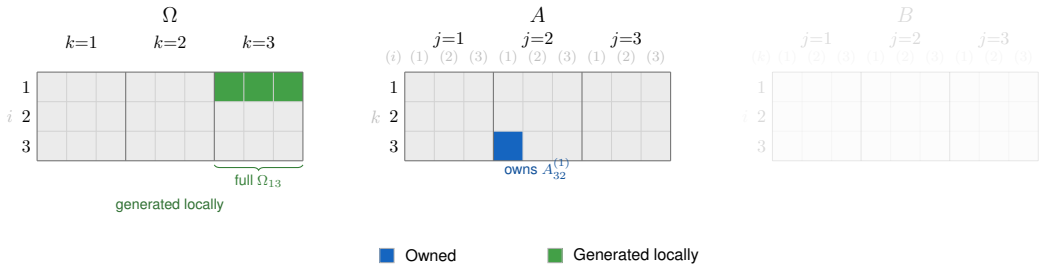
Sparse sketches and local generation of Ω

$$B = \Omega A, \qquad \Omega \in \mathbb{R}^{\ell \times n_1}, \quad A \in \mathbb{R}^{n_1 \times n_2}, \quad \ell \ll n_1.$$

- Each column of Ω has exactly ζ nonzeros, at uniformly random rows, with values $\pm 1/\sqrt{\zeta}$.
- $\zeta = 1$: CountSketch (Charikar et al., ICALP 2002);
  $\zeta \geq 2$: Sparse Signs (Martinsson and Tropp, Acta Numerica 2020, §9.2).
- Ω is generated from a shared seed with a counter-based RNG: every processor can generate any block of Ω locally, and Ω is the same for every processor grid.
- Ω is not communicated; generation costs computation only.

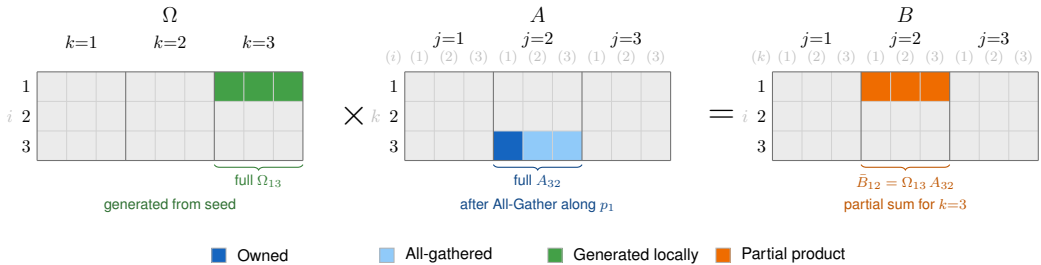
Same approach for dense Ω: Al Daas et al., SPAA 2026. This work: sparse Ω.

CA-SparseSketch: data distribution



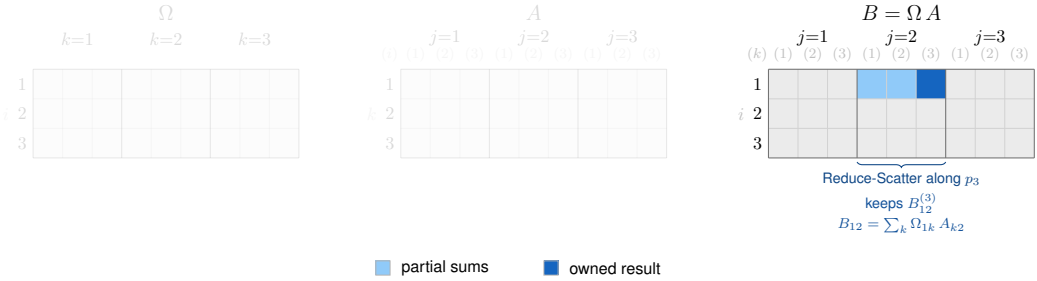
Processor $(i, j, k) = (1, 2, 3)$ on a $3 \times 3 \times 3$ grid. Block A_{kj} is split over the p_1 processors of a fiber; processor (i, j, k) owns $A_{kj}^{(i)}$. Block Ω_{ik} is generated on every processor that needs it.

CA-SparseSketch: local computation



All-Gather A_{kj} along p_1. Generate Ω_{ik}. Local sparse-dense product (SpMM) $\bar{B}_{ij} = \Omega_{ik} A_{kj}$.

CA-SparseSketch: output reduction



Reduce-Scatter $\bar{B}_{ij}$ along p_3; processor (i, j, k) keeps $B_{ij}^{(k)}$. Only A and B are communicated.

CA-SparseSketch: communication cost

For a $p_1 \times p_2 \times p_3$ grid, $P = p_1 p_2 p_3$:

$$W_{\mathrm{CA}} = \underbrace{(p_1 - 1)\frac{n_1 n_2}{P}}_{\text{All-Gather } A} + \underbrace{(p_3 - 1)\frac{\ell n_2}{P}}_{\text{Reduce-Scatter } B} .$$

- $P \le n_2$: grid $(1, P, 1)$ gives $W_{\mathrm{CA}} = 0$.
- The communication pattern does not depend on ζ; ζ only changes the local work.

W: words sent and received along the critical path. A and B are each stored once, in the layout of the previous slides.

Communication lower bound

$\Omega \in \mathbb{R}^{\ell \times n_1}$ with exactly ζ nonzeros per column, $A \in \mathbb{R}^{n_1 \times n_2}$, P processors.

Processors	Lower bound on W
$1 \le P \le n_2$	0
$n_2 < P \le \dfrac{n_1 n_2}{\zeta}$	$\max\left(0,\ \zeta - \dfrac{n_2 \ell}{P}\right)$
$P > \dfrac{n_1 n_2}{\zeta}$	$2\sqrt{\dfrac{n_1 n_2 \zeta}{P}} - \dfrac{n_1 n_2 + n_2 \ell}{P}$

Assumes computation or data balanced; one copy of A at the start and of B at the end. For $\zeta = \ell$ this is the dense bound.

- CA-SparseSketch attains the bound for $P \le n_2$.
- For $n_2 < P \le n_2 \ell / \zeta$ the bound is 0; CA-SparseSketch communicates up to ℓ words.
- Beyond that, its leading term exceeds the bound by at most ℓ/ζ, resp. $\sqrt{\ell/\zeta}$.
- A restricted block-diagonal Ω attains the bound for all P (backup).

Lower bound: proof sketch

Task (i,j,k): $B(i,j) \mathrel{+}= \Omega(i,k)\,A(k,j)$; $\zeta n_1 n_2$ tasks in total. Consider a processor with task set V, $|V| \geq \zeta n_1 n_2/P$, that owns at most $(n_1 n_2 + \ell n_2)/P$ words.

- Ω entries are generated, so only accesses to A and B count. (k,j) and (i,j) determine (i,j,k):

$$|V| \leq |\phi_A(V)|\,|\phi_B(V)|$$

(Al Daas et al., SPAA 2026). $\phi_M(V)$: entries of M accessed by V.

- Each $A(k,j)$ is used by exactly ζ tasks $\Rightarrow |\phi_A(V)| \geq n_1 n_2/P$.
- Each $B(i,j)$ is a sum of at most n_1 tasks $\Rightarrow |\phi_B(V)| \geq \zeta n_2/P$.

With $x = |\phi_A(V)|$, $y = |\phi_B(V)|$:

$$W \geq \max\left(0,\ \min_{x,y}(x+y) - \frac{n_1 n_2 + \ell n_2}{P}\right),$$

$$\text{subject to} \quad xy \geq \frac{\zeta n_1 n_2}{P}, \qquad x \geq \frac{n_1 n_2}{P}, \qquad y \geq \frac{\zeta n_2}{P}.$$

The active constraints change with P; this gives the three ranges.

Experimental setup

GPU: Alps (CSCS)

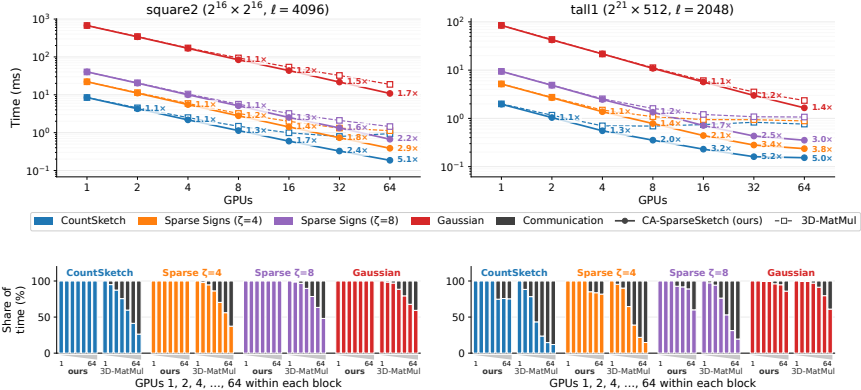
1–64 NVIDIA GH200
cuSPARSE / cuBLAS, NCCL
Square: $2^{16} \times 2^{16}$, $\ell = 4096$
Tall: $2^{21} \times 512$, $\ell = 2048$

CPU: AMD EPYC 7742

1–64 MPI ranks, 16 threads each
AOCL-Sparse / BLIS, Cray MPICH
Square: $2^{15} \times 2^{15}$, $\ell = 1024$
Tall: $2^{21} \times 512$, $\ell = 2048$

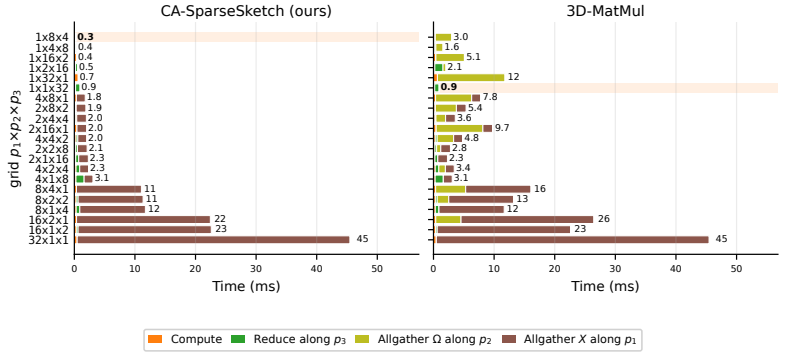
- Single precision; $\zeta = 1, 4, 8$; Gaussian Ω for reference.
- Baseline **3D MatMul**: same local libraries, Ω stored once and All-Gathered along p_2.
- All grids (p_1, p_2, p_3) tested; best grid of each method; median of 5 runs.
- In all runs $P \leq n_2$: the lower bound is 0, and the communication-optimal grid is $(1, P, 1)$.

GPU strong scaling



$\zeta = 1, 4, 8$	square $2^{16} \times 2^{16}$	tall $2^{21} \times 512$
CA-SparseSketch vs. 3D MatMul, 64 GPUs	5.1×, 2.9×, 2.2×	5.0×, 3.8×, 3.0×
Speedup 1 → 64 GPUs (1 GPU: cuSPARSE SpMM)	45×, 57×, 60×	13×, 22×, 27×

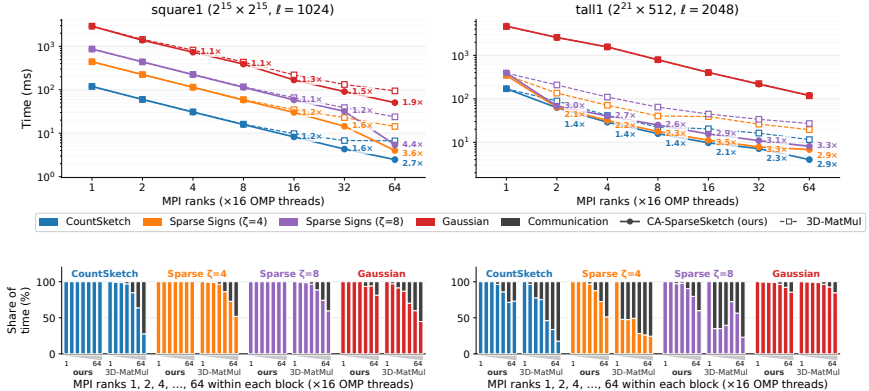
Processor grid



All grids on 32 GH200, tall $2^{21} \times 512$, $\zeta = 4$.

- Best grid $(1, 8, 4)$: 0.3 ms. Communication-free grid $(1, 32, 1)$: 0.7 ms. The best grid adds a small Reduce-Scatter of B and gives wider local SpMMs (64 instead of 16 columns of A).
- The slowest grids have large p_1 and are dominated by the All-Gather of A: worst/best $> 100\times$ (CA-SparseSketch), $50\times$ (3D MatMul).

CPU strong scaling



$\zeta = 1, 4, 8$	square $2^{15} \times 2^{15}$	tall $2^{21} \times 512$
CA-SparseSketch vs. 3D MatMul, 64 ranks	2.7×, 3.6×, 4.4×	2.9×, 2.9×, 3.3×
Speedup 1 → 64 ranks (16 → 1024 cores)	48×, 110×, 160×	43×, 51×, 48×

Summary

1. **CA-SparseSketch.** Ω is generated from a seed on every processor that needs it; only A and B are communicated.

2. **Lower bound** for Ω with exactly ζ nonzeros per column. CA-SparseSketch attains it for $P \leq n_2$; a restricted block-diagonal Ω attains it for all P.

3. **Experiments** ($P \leq n_2$). At 64 GH200: 2.2–5.1$\times$ faster than 3D MatMul. At 1024 CPU cores: 2.7–4.4$\times$. The best grid depends on local SpMM efficiency, not only on communication.

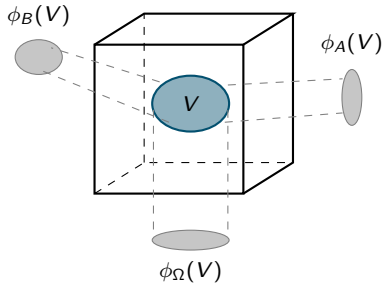
Thank you for your attention!

Questions?

Parallel computation model

- P processors with private memory, fully connected network.
- W: words sent and received along the critical path (bandwidth cost). Collectives are bandwidth-optimal.
- Either the computation or the data of A and B is load balanced.
- One copy of A at the start, one copy of B at the end.
- No assumption on the size of local memory.
- Ω is generated from a shared seed: this costs computation, not communication.

Dense matrix multiplication: lower bound



3D attains this square-case bound for compatible processor grids.

Tight rectangular bounds:
Al Daas et al., SPAA 2022.

- V: a processor's tasks, out of n^3 (i,j,k): $B(i,j) \mathrel{+}= \Omega(i,k)\,A(k,j)$
- $\phi_M(V)$: set of entries of matrix M accessed by the tasks in V (its projection).
- Loomis–Whitney:

$$|V|^2 \le |\phi_\Omega(V)| \cdot |\phi_A(V)| \cdot |\phi_B(V)|$$

- Load balance: $|V| \ge \frac{n^3}{P} \Rightarrow$ sum of projection sizes $\ge 3\left(\frac{n^3}{P}\right)^{2/3}$
- Subtract the owned data:

$$W \;\ge\; 3\left(\frac{n^3}{P}\right)^{2/3} - 3\,\frac{n^2}{P}$$

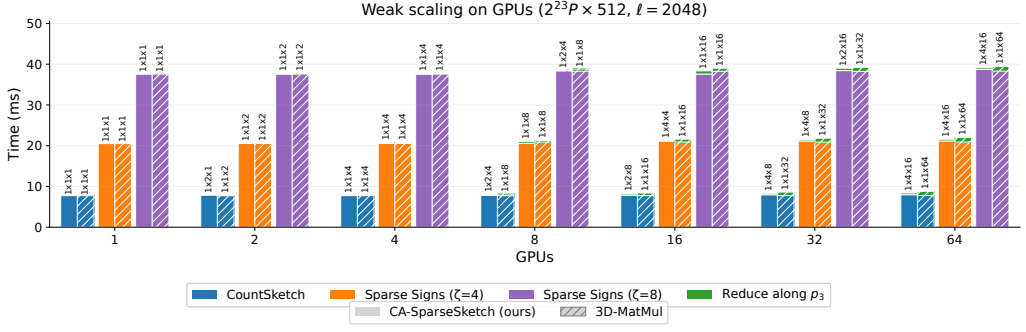
Tightness of the lower bound

- The communication of CA-SparseSketch does not depend on ζ; for $\zeta = \ell$ it matches the dense bound.
- Gap of the leading terms, CA-SparseSketch vs. lower bound:

$$\begin{cases} \text{bound } 0, \text{ algorithm} \le \ell, & n_2 < P \le n_2\ell/\zeta, \\ \text{factor} \le \ell/\zeta, & n_2\ell/\zeta < P \le n_1 n_2/\zeta, \\ \text{factor} \le \sqrt{\ell/\zeta}, & P > n_1 n_2/\zeta. \end{cases}$$

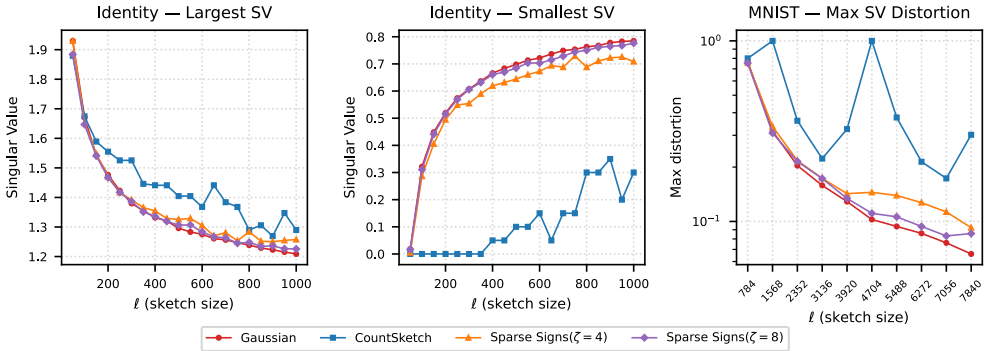
- If $\zeta \mid \ell$ and Ω is block diagonal with ℓ/ζ dense blocks, an algorithm with a matching data distribution attains the bound for all P.
- For $\zeta > 1$ such an Ω is not a Sparse Signs matrix and does not have its embedding guarantees.
- A CountSketch can be permuted into this form with block sizes balanced in expectation; this requires redistributing the rows of A.

GPU weak scaling



A has $2^{23}P$ rows and 512 columns, $\ell = 2048$; $P = 1, \ldots, 64$ GPUs. B is small, so the Reduce-Scatter along p_3 is cheap.

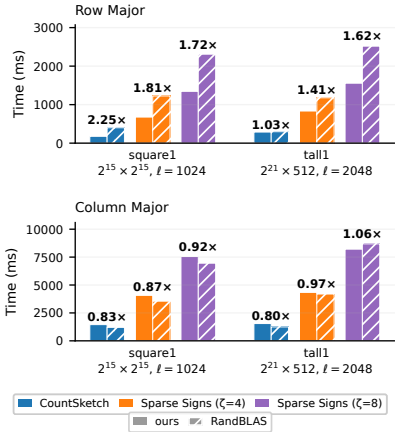
Sketch quality at fixed ℓ



Identity: 50 columns of I_{1000}; mean of the largest and smallest singular value over 10 trials.
MNIST: $70{,}000 \times 784$; maximum relative singular-value distortion over 10 trials.

CountSketch has larger distortion in both examples. Sparse Signs with $\zeta = 8$ is close to Gaussian in these examples.

Local kernel: comparison with RandBLAS



Single thread. Number above each pair: RandBLAS time / our time. Row-major A: 1.03–2.25×; column-major A: 0.80–1.06×.

Murray et al., *Randomized Numerical Linear Algebra: A Perspective on the Field With an Eye to Software*, 2023. arXiv:2302.11474.