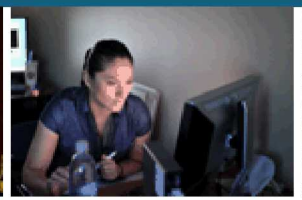
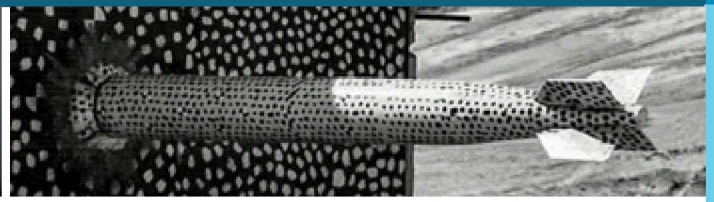
This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

SAND2020-6987C

ALO-NMF: Accelerated Locality-Optimized Non-negative Matrix Factorization





Presented by:

Gordon E. Moon[*], J. Austin Ellis[*], Aravind Sukumaran-Rajam[#], Srinivasan Parthasarathy[†], P. Sadayappan[‡]

Sandia National Laboratories[*], Washington State University[#], The Ohio State University[†], University of Utah[‡]

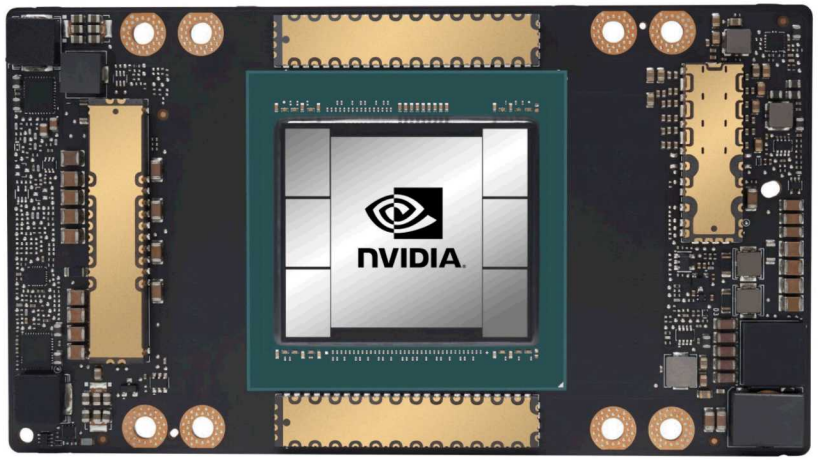
Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

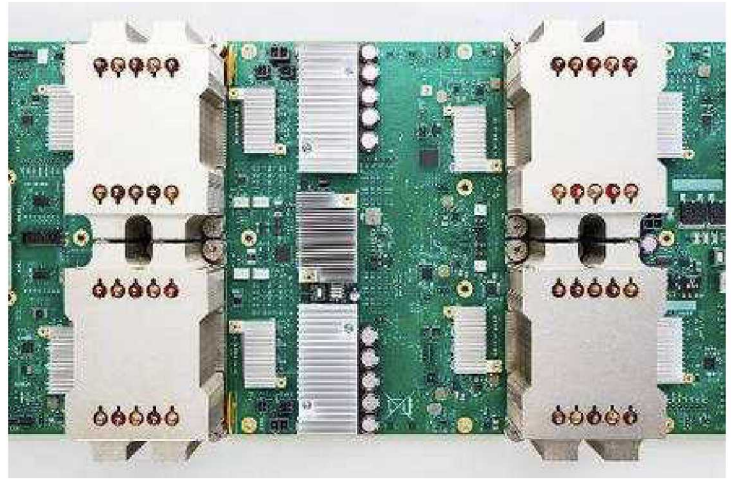
Machine Learning Everywhere

Machine Learning is becoming an integral part of everyday life

Top-Trending Machine Learning Architecture

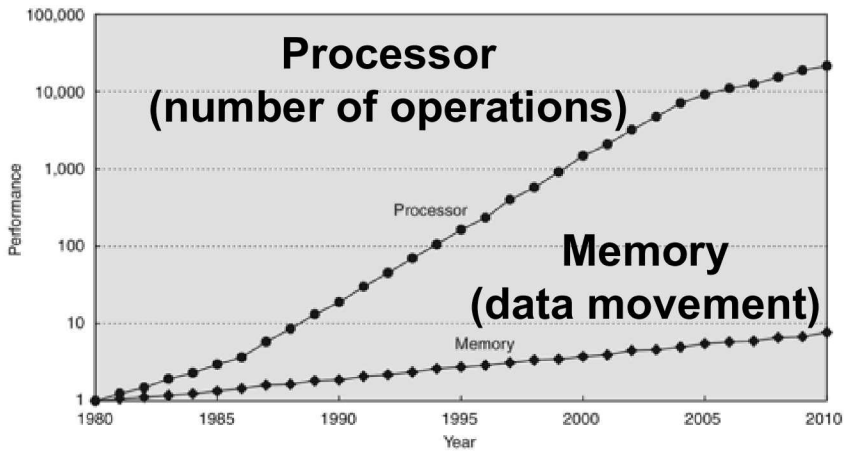
How to achieve good performance on specialized architectures?



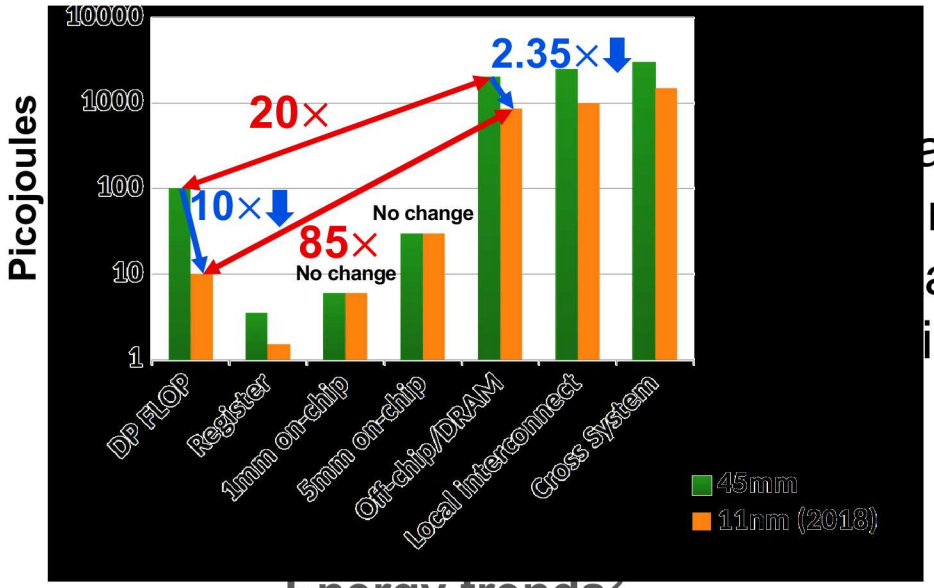


Architecture-aware Machine Learning

- Aspects of Performance
  - Processor (number of operations)
  - Memory (data movement)



FLOPs vs. Data movement[1]



Energy trends[2]

ative energy cost
movement vs.
ation keeps
ing (20× → 85×)

- FLOPs are free, but data movement is expensive
- Minimization of data movement overheads is increasingly critical
- Architecture-aware algorithm design

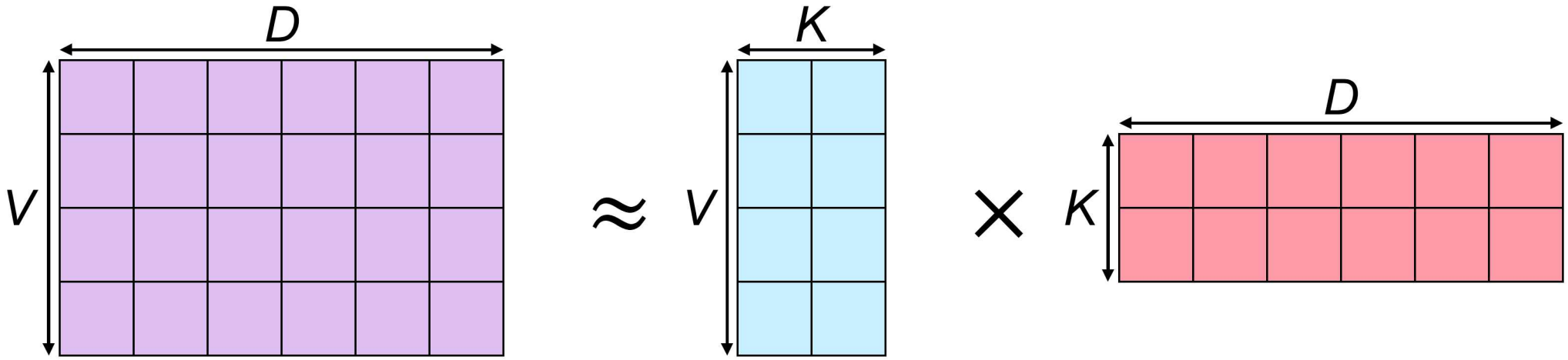
[1]Source: John L. Hennessy (Stanford) and David A. Patterson (UC Berkeley)
[2]Source: Jim Demmel (UC Berkeley) and John Shalf (LBL)

Non-negative Matrix Factorization (NMF)

Given a matrix $\mathbf{A} \in \mathbb{R}_{+}^{V \times D}$ and latent variable $K \ll \min(V, D)$,

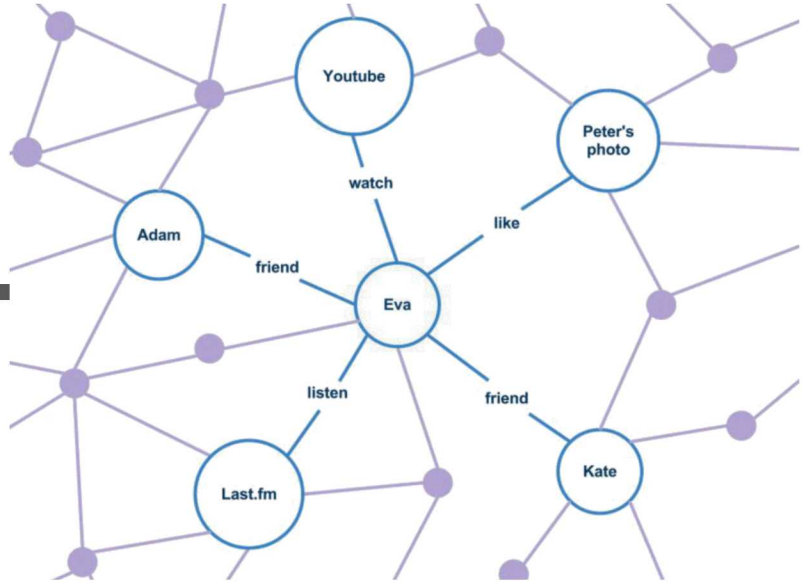
NMF estimates two rank-K matrices $\mathbf{W} \in \mathbb{R}_{+}^{V \times K}$ and $\mathbf{H} \in \mathbb{R}_{+}^{K \times D}$ such that,

$$\mathbf{A} \approx \mathbf{W}\mathbf{H}$$

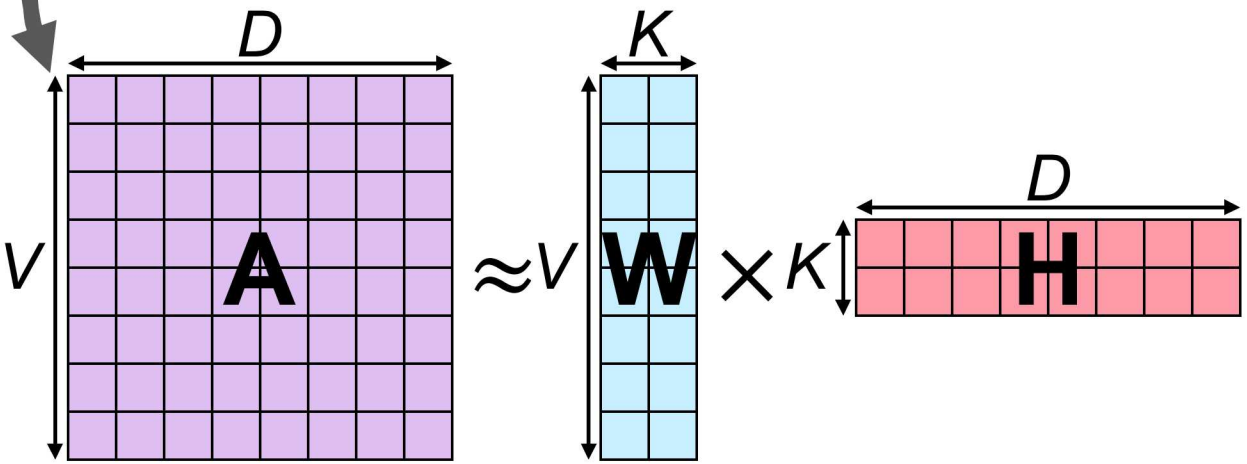


Matrix **A** Matrix **W** Matrix **H**

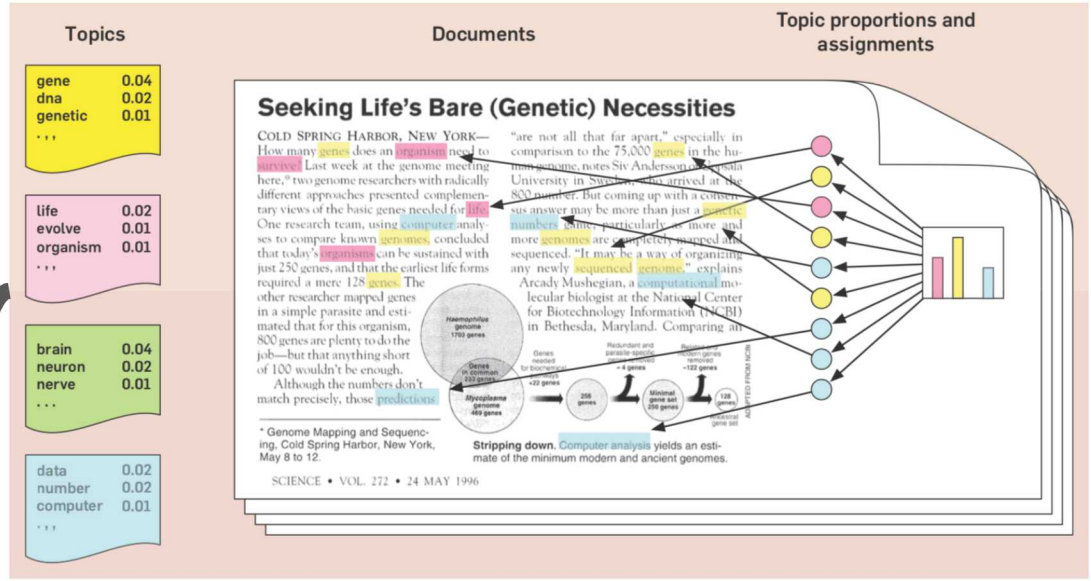
NMF Applications



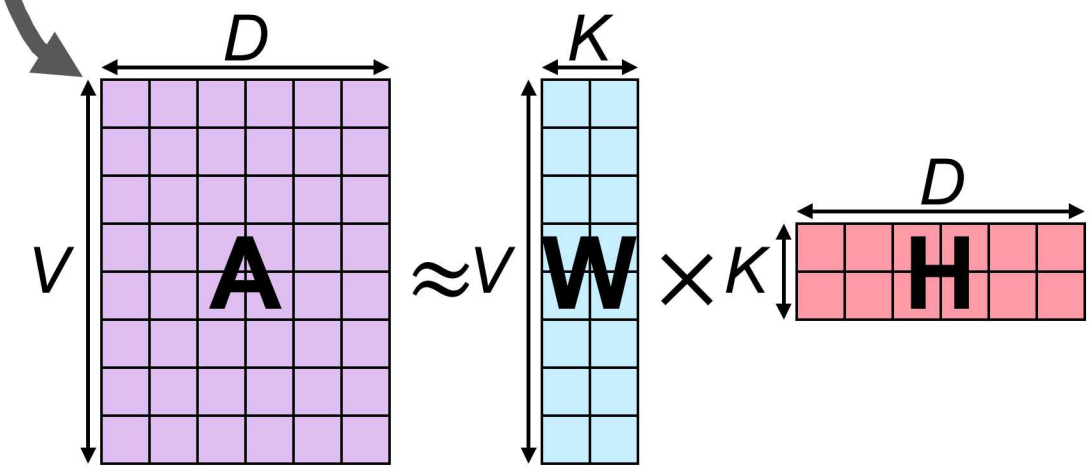
Node Embedding for Graph Mining



V: number of unique nodes
D: number of unique nodes



Topic Modeling for Text Mining*



V: vocabulary size
D: number of documents

*Source: David Blei. "Probabilistic Topic Models". (2012)

NMF Algorithms

- Objective function

$$\mathrm{D_F(A||WH)} = \frac{1}{2}||\mathrm{A} - \mathrm{WH}||_{\mathrm{F}}^2 = \frac{1}{2}\sum_{vd}(\mathrm{A}_{vd} - (\mathrm{WH})_{vd})^2$$

- Variants of NMF
  - Multiplicative Update (MU)
  - Additive Update (AU)
  - Alternating Non-negative Least Squares (ANLS)
  - Hierarchical Alternating Least Squares (HALS)

Performance Challenges in HALS-based NMF

Input: $A \in \mathbb{R}_{+}^{V \times D}$: non-negative input matrix, ε: machine epsilon

Initialize $W \in \mathbb{R}_{+}^{V \times K}$ and $H \in \mathbb{R}_{+}^{K \times D}$ with random non-negative numbers

repeat

Updating H

$R = A^T W$

$S = W^T W$

for $k = 0$ **to** $K - 1$ **do**

$H_k = \max(\varepsilon, H_k + R_k - H^T S_k)$

Updating W

$P = AH^T$

$Q = HH^T$

for $k = 0$ **to** $K - 1$ **do**

$W_k = \max(\varepsilon, W_k Q_{kk} + P_k - WQ_k)$

$W_k = \frac{W_k}{||W_k||_2}$

until convergence

The main data movement overhead is associated with these k loops

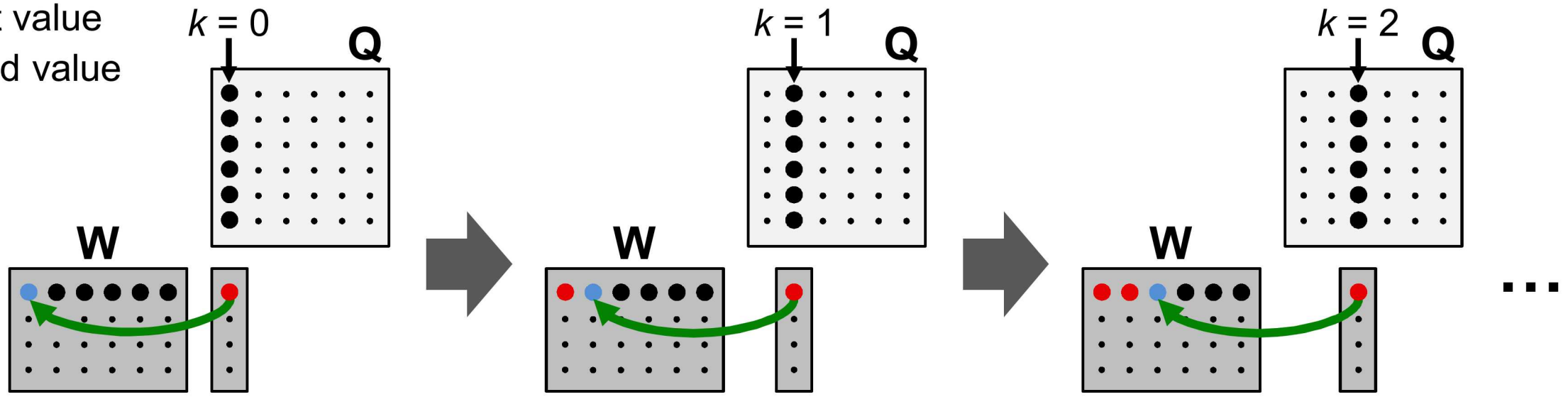
➤ **91% of the combined fractional data movement overhead**

How to reduce data movement cost of these k loops?

HALS-based NMF

Interaction between different columns of **W** with iterative matrix-vector multiplications

- original value
- current value
- updated value



Replace W[0][0] with an updated value

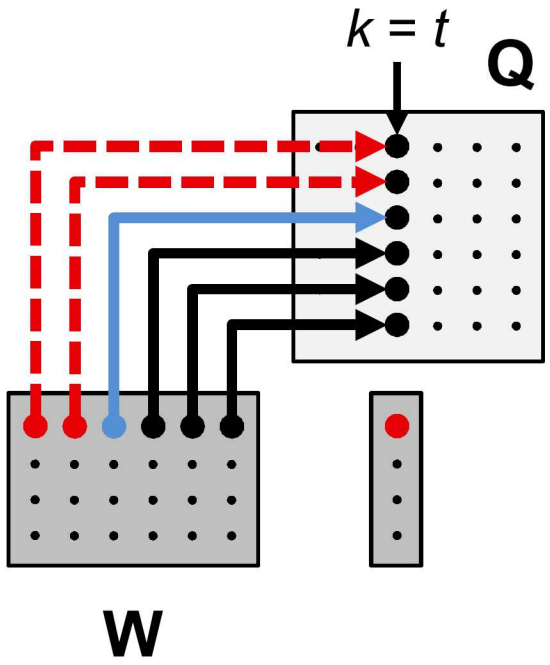
Replace W[0][1] with an updated value

Replace W[0][2] with an updated value

HALS-based NMF

Updating **W** with iterative matrix-vector multiplications

- original value
- current value
- updated value

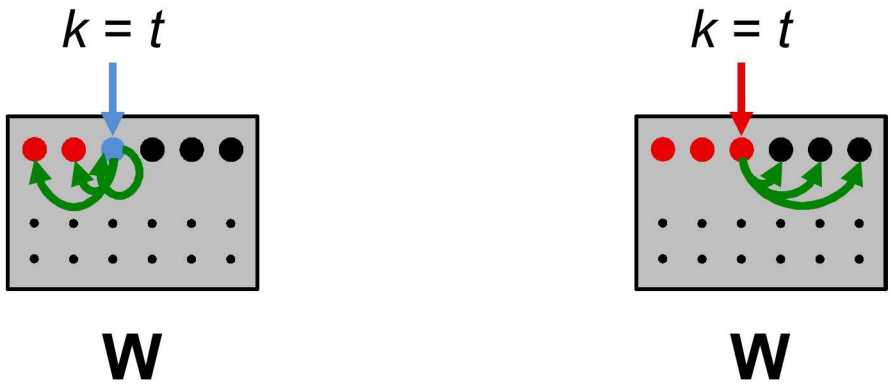


$$W\_new_{0,0}Q_{0,t} + W\_new_{0,1}Q_{1,t} + W\_old_{0,2}Q_{2,t} + W\_old_{0,3}Q_{3,t} + W\_old_{0,4}Q_{4,t} + W\_old_{0,5}Q_{5,t}$$

HALS-based NMF

Updating **W** with iterative matrix-vector multiplications

- original value
- current value
- updated value



Overview of Our Approach

Our goal is to minimize data movement cost

Original HALS-based NMF

➤ **The lower bound of data movement for iterative matrix-vector multiplications**

$$= VK^2 + K^2 + VK$$

where V: # rows in W
K: # columns in W, # rows and # columns in Q

Our ALO-NMF

➤ **The lower bound of data movement for efficient tiled matrix-matrix multiplication***

$$= \frac{2VK^2}{\sqrt{C}}$$

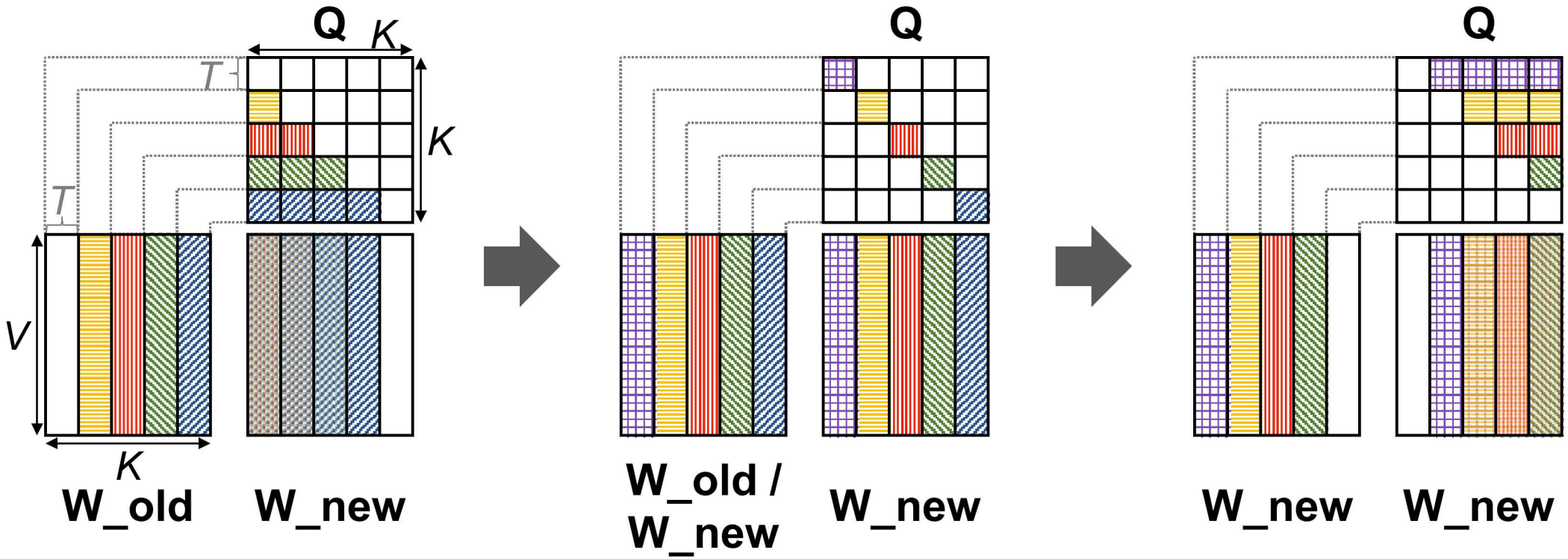
where C: cache size

*Source: Julien Langou. "Communication Lower Bounds for Matrix-Matrix Multiplication". (2015)

How to reformulate the original iterative matrix-vector multiplications to matrix-matrix multiplication?

Brand New ALO-NMF (Accelerated Locality-Optimized NMF)

Updating **W** with tiled matrix-matrix multiplications



Phase 1

W_new[: , 0 : τ ×T-1] -=
W_old[: , τ ×T : (τ+1)×T-1]
×

Phase 2

W_new[: , τ ×T : (τ+1)×T-1] -=
W[: , τ ×T : (τ+1)×T-1] ×
Q[τ ×T : (τ+1)×T-1, τ ×T : (τ+1)×T-1]

Phase 3

W_new[: , (τ+1)×T : K-1] -=
W_new[: , τ ×T : (τ+1)×T-1] ×
Q[τ ×T : (τ+1)×T-1, (τ+1)×T : K-

Data Movement Comparison

Running on a PIE **dense** image dataset

V (# rows in W)	K (low rank)	T (tile size)	C (cache size)
11,554	256	16	33MB

Comparison of data movement cost for updating **W**

Original HALS-based NMF (byte)	Our ALO-NMF (byte)
$K(VK + K + 6V + 1)$ = 775,015,680	$V\left(\frac{1}{T} + \frac{2}{\sqrt{C}}\right)(K^2 - KT) + KVT$ = 338,840,256

2.29× reduced

Determination of the Optimal Tile Size

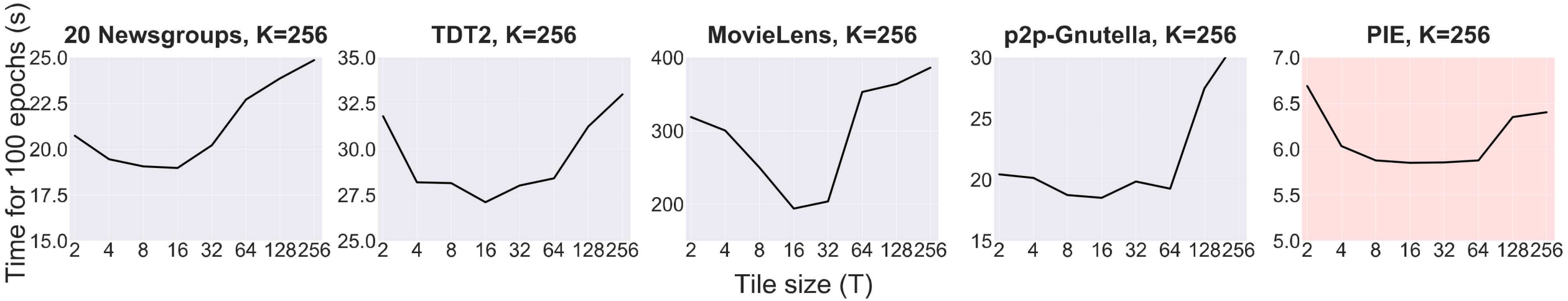
Total data movement of phases 1, 2 and 3 =

$$vol(T) = V\left(\frac{1}{T} + \frac{2}{\sqrt{C}}\right)(K^2 - KT) + KVT$$

$$\frac{d(vol(T))}{dT} = T^2\left(\frac{2}{\sqrt{C}} - 1\right) + K = 0$$

Optimal tile size *T* ➡ $$\boldsymbol{T = \sqrt{\frac{K\sqrt{C}}{\sqrt{C} - 2}}}$$

For a machine with *C* = 33 MB and *K* = 256, optimal tile size *T* = 19.82



Tile size *T* selected by a function are optimal!

Performance Evaluation: Setup

- We used four sparse matrices and a dense matrix for evaluation

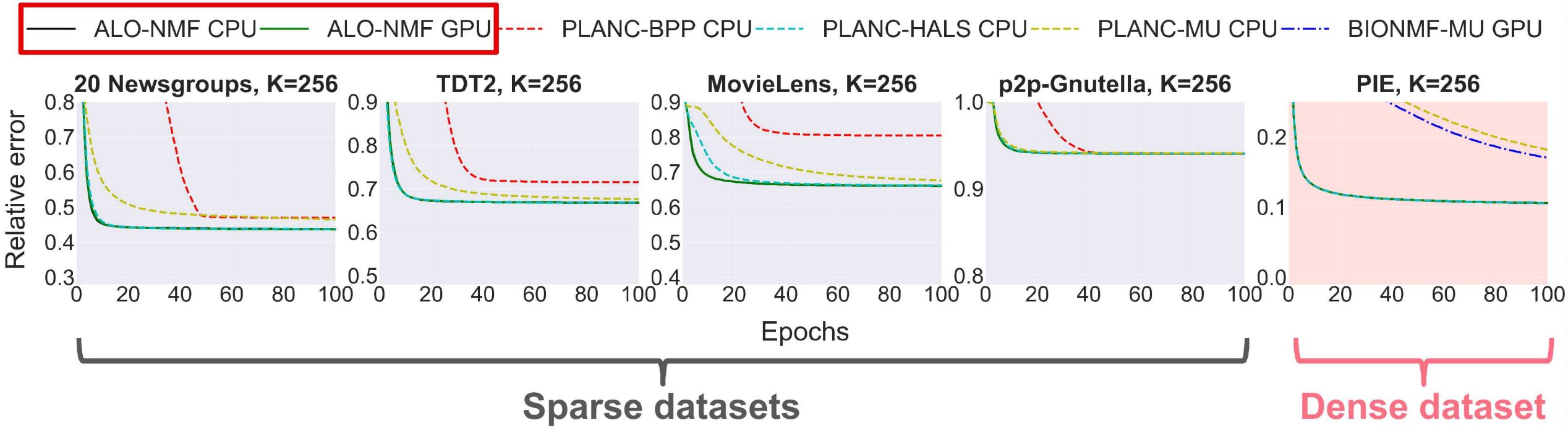
	Dataset	*V*	*D*	NNZ
Sparse datasets	**20 Newsgroups**	26,214	11,314	1,018,191
	TDT2	36,771	10,212	1,323,869
	MovieLens	18,933	8,293	389,455
	p2p	400	10,304	4,121,478
Dense dataset	**PIE**	11,554	4,096	47,321,408

- We used the relative objective function to measure the accuracy of NMF variants

$$\text{Relative error} = \sqrt{\frac{\sum_{vd}(A_{vd}-(WH)_{vd})^2}{\sum_{vd}(A_{vd})^2}}$$

Performance Comparison: Quality

HALS-based NMF implementations (ALO-NMF CPU/GPU and PLANC-HALS CPU) produced a better convergence rate than other NMF variants

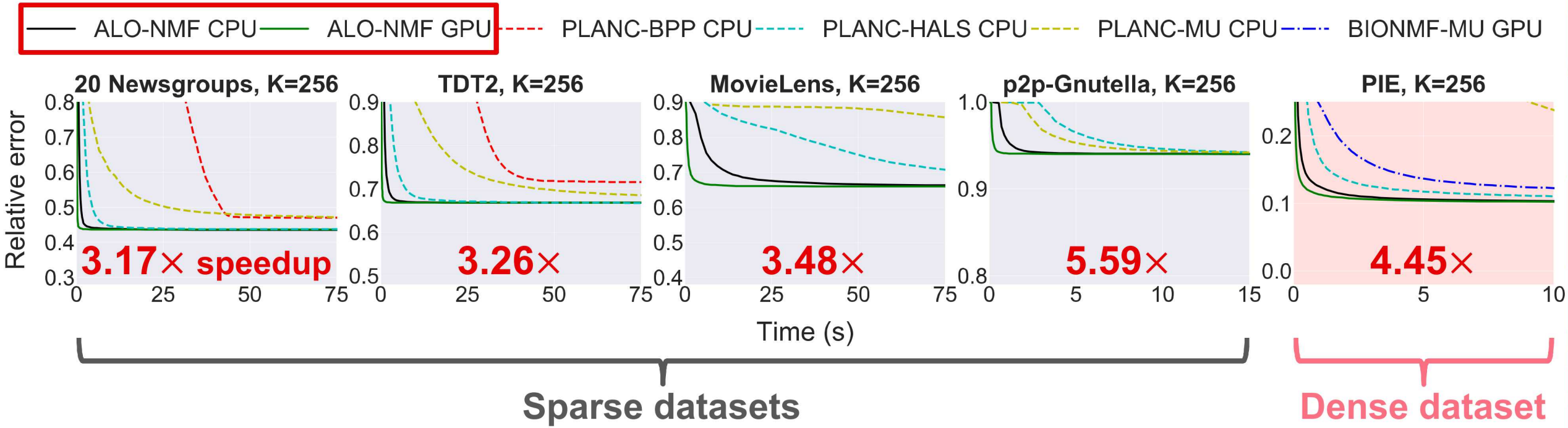


Relative error vs. Number of epochs

The lower the better

Performance Comparison: Speedup

ALO-NMF CPU/GPU achieved significant performance improvement over the existing state-of-the-art parallel NMF implementations



Relative error vs. Training time (s)

The lower the better

Performance Comparison: Speedup

- Original HALS-based NMF vs. ALO-NMF
  - Breakdown of elapsed time in the process of updating W

Original HALS-based NMF	Elapsed time (s) per epoch
Sparse Matrix-Dense Matrix Multiplication (SpMM)	0.046
Dense Matrix-Dense Matrix Multiplication (DMM)	0.001
Iterative Dense Matrix-Dense Vector Multiplication	0.210
Total elapsed time (s)	0.257

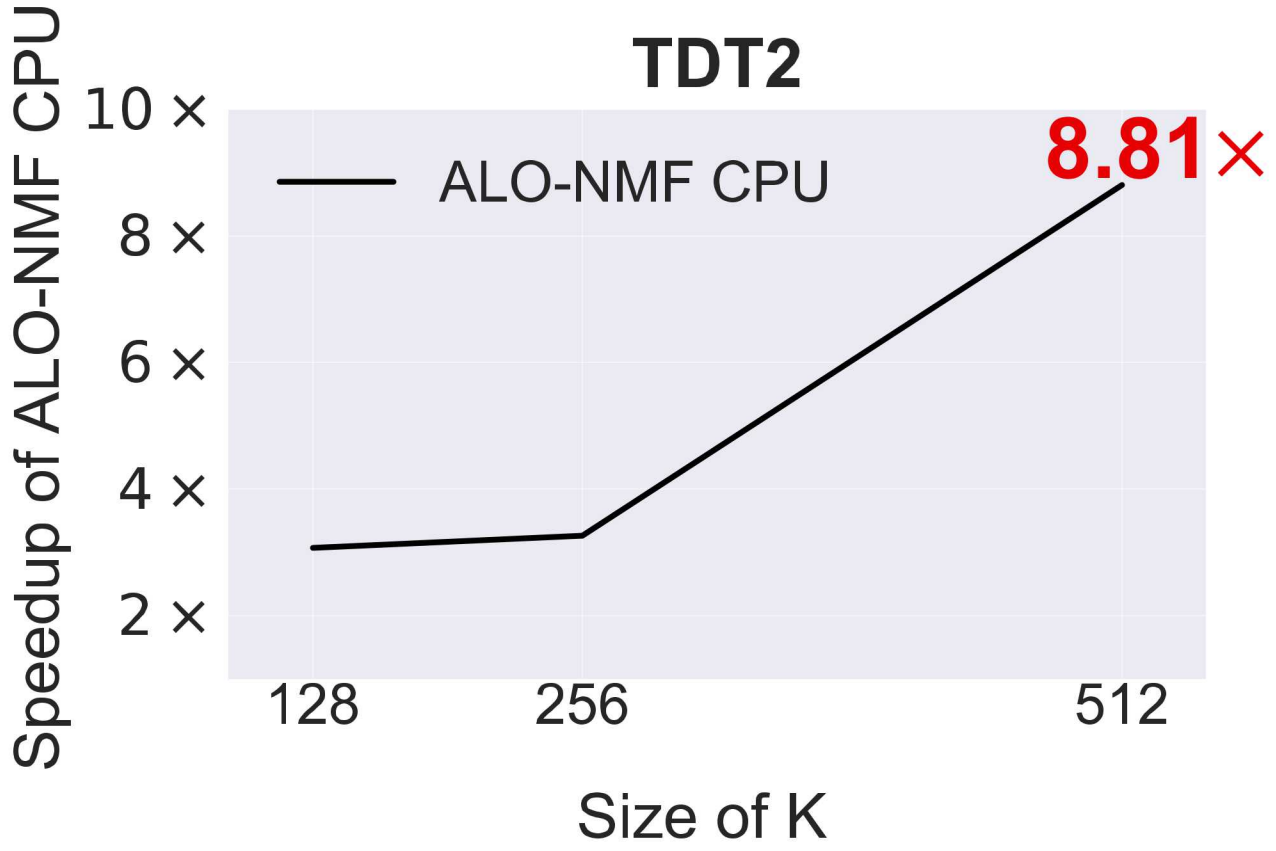
Our ALO-NMF	Elapsed time (s) per epoch
Sparse Matrix-Dense Matrix Multiplication (SpMM)	0.046
Dense Matrix-Dense Matrix Multiplication (DMM)	0.001
Phase 1	0.002
Phase 2 & 3	0.031
Total elapsed time (s)	0.080

68.87% reduced

The lower the better

Performance Comparison: Speedup

Scalability with the large rank-K



The higher the better

Summary and Conclusions

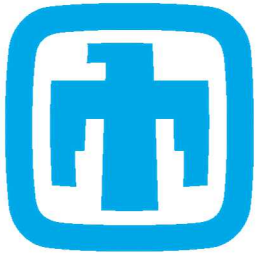
- Architecture-aware machine learning algorithm design is critical
- We focused on data locality optimizations for NMF
  - The associativity of addition is utilized to reorder additive contributions in updating elements of matrices **W** and **H**
  - We developed a function of tile size for selection of effective tile size T
- Our ALO-NMF achieved **2.29**× lower data movement and ~**4.45**× speedup compared to the existing state-of-the-art parallel NMF implementations

Acknowledgements

THE OHIO STATE UNIVERSITY

THE UNIVERSITY OF UTAH®

WASHINGTON STATE UNIVERSITY



Sandia National Laboratories

- Layer-parallel
- ARIAA

Thank you.