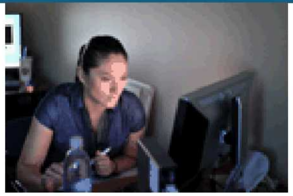
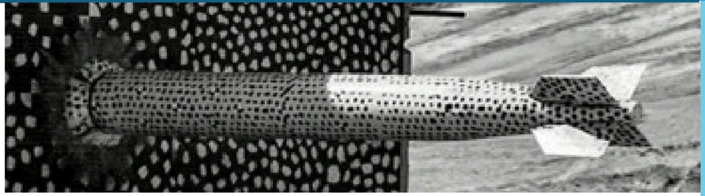
SAND2020-5964C

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





Presented by:

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

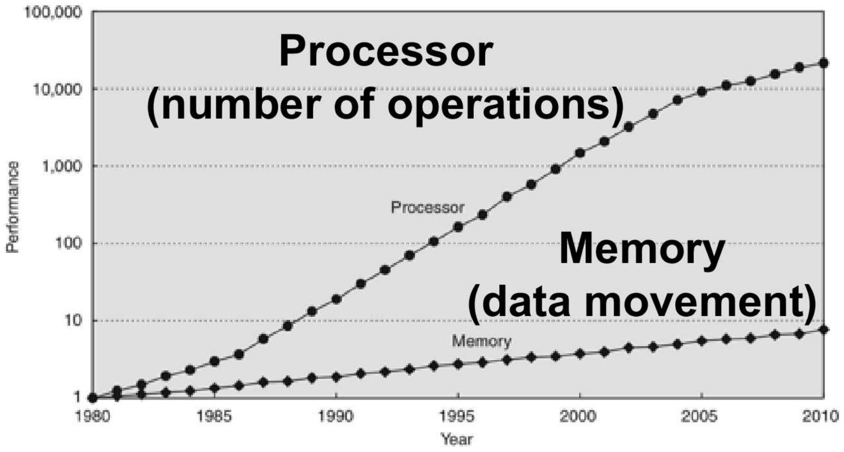
Sandia National Laboratories[*], Washington State University[#], The Ohio State University[†], University of Utah[‡]

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.

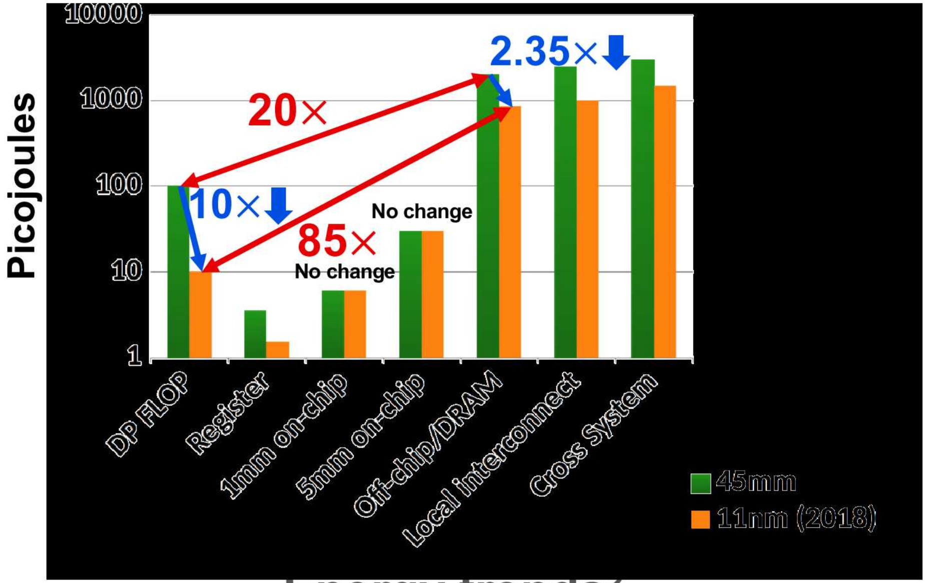
Architecture-aware Machine Learning

Machine Learning is becoming an integral part of everyday life

- How to achieve good performance on specialized architectures?



FLOPs vs. Data movement[1]



Energy trends[2]

- FLOPs are free, but data movement is expensive
- Minimization of data movement overheads is increasingly critical

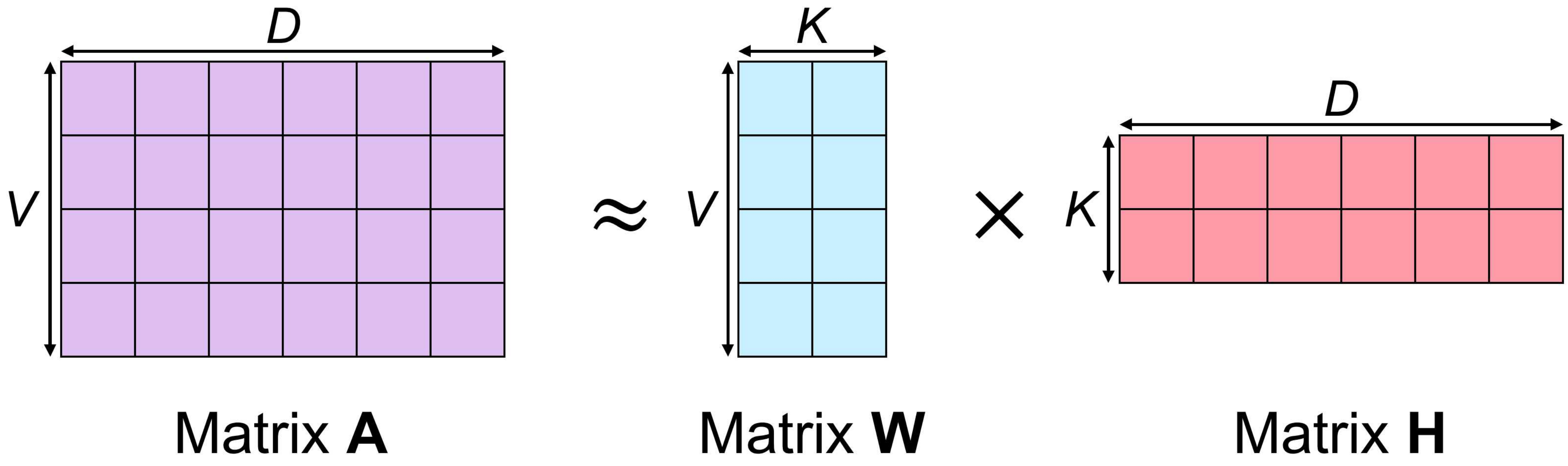
[1]Source: John L. Hennessy (Stanford) and David A. Patterson (UC Berkeley)
[2]Source: Jim Demmel (UC Berkeley) and John Shalf (LBL)

Non-negative Matrix Factorization (NMF)

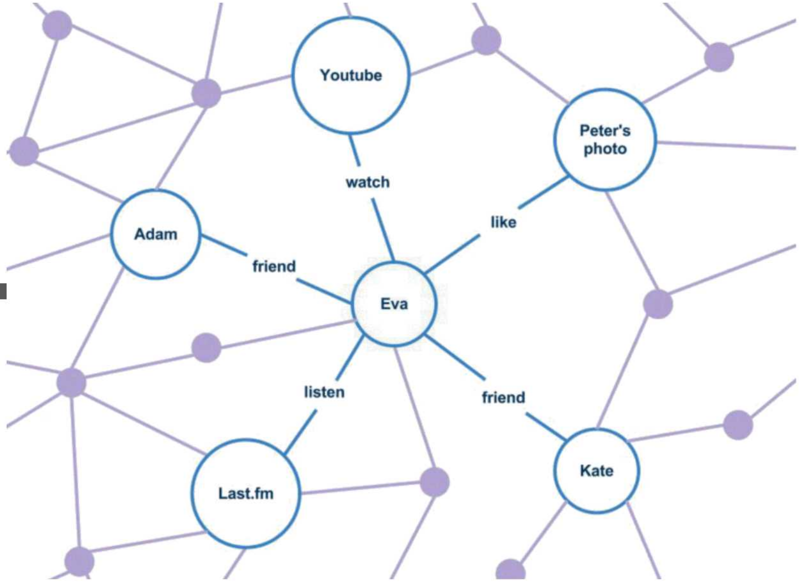
Given a matrix **A** $\in \mathbb{R}_+^{V \times D}$ and latent variable $K \ll \min(V, D)$,

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

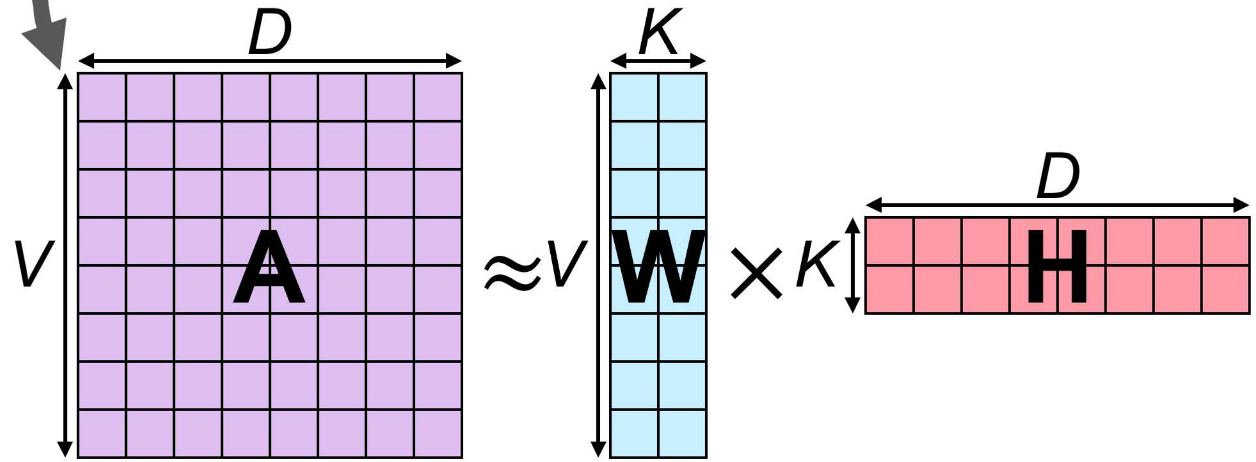
$$\mathbf{A} \approx \mathbf{WH}$$



NMF Applications

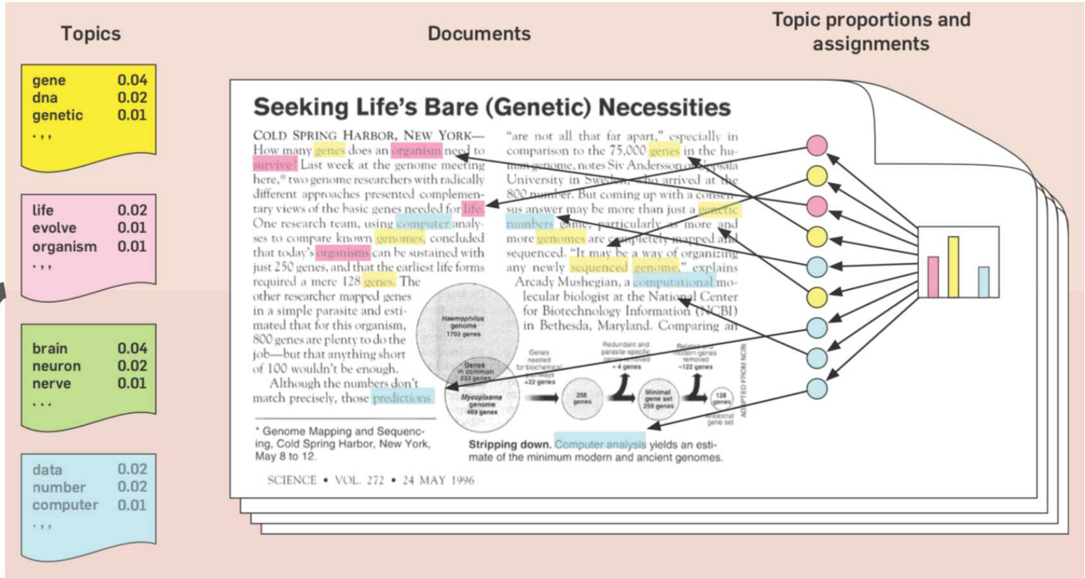


Node Embedding for Graph Mining

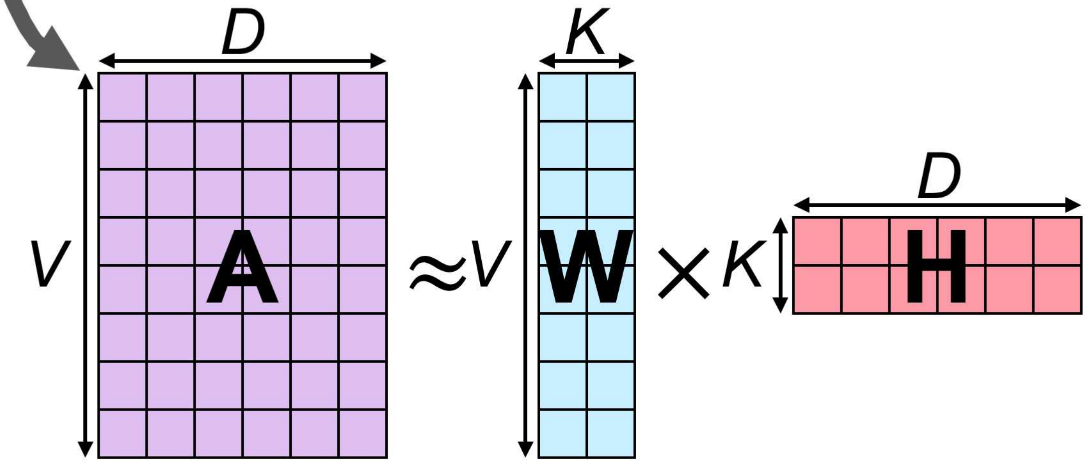


V: number of unique nodes

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

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

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**

$\quad H_K = \max(\varepsilon, H_K + R_K - H^T S_K)$

Updating W:

$P = AH^T$

$Q = HH^T$

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

$\quad W_K = \max(\varepsilon, W_K Q_{KK} + P_K - W Q_K)$

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

until convergence

The main data movement overhead is associated with these k loops

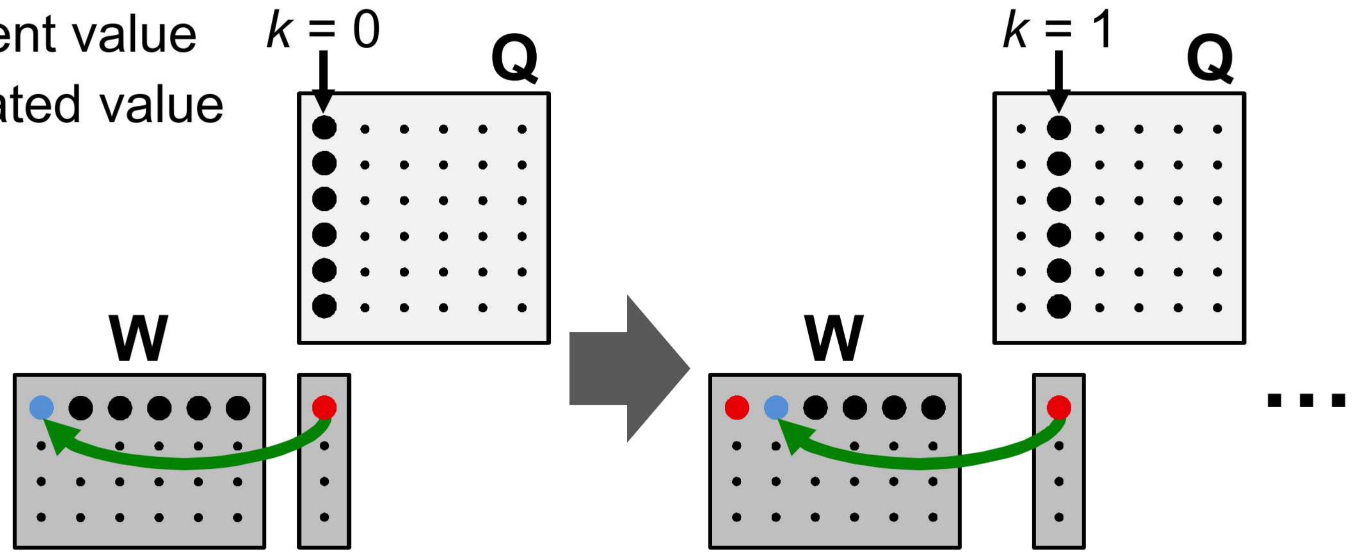
➤ 91% of the combined fractional data movement overhead

How to reduce data movement cost of these k loops?

Original HALS-based NMF

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

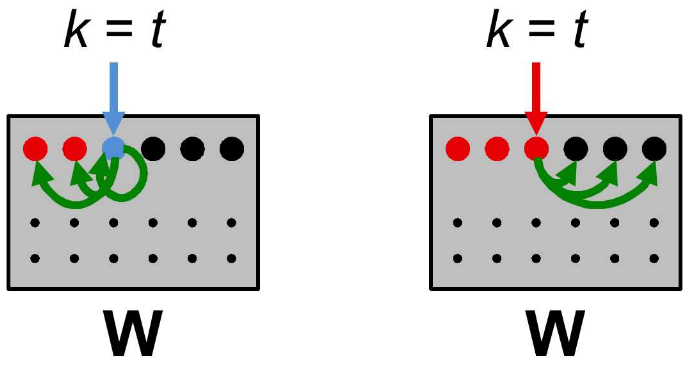
- original value
- current value
- updated value



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

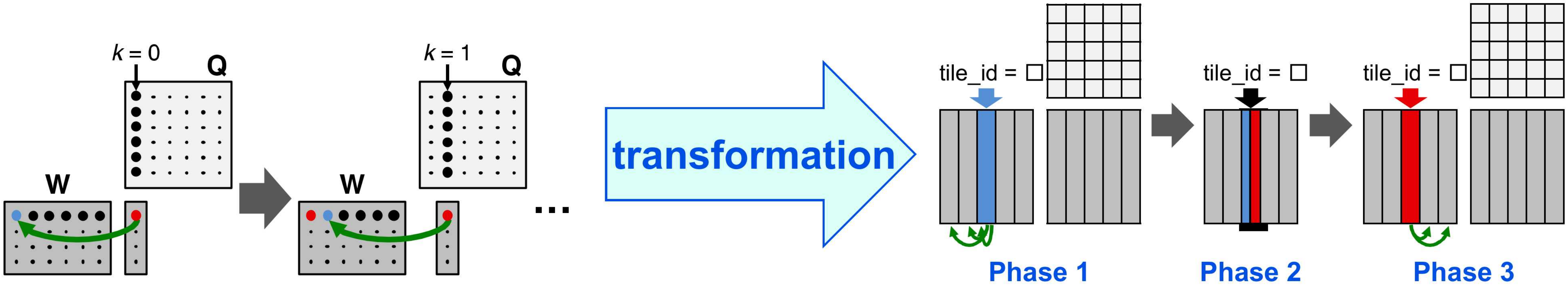
Replace W[0][1] with an updated value

- original value
- current value
- updated value



Overview of Our Approach

Our goal is to minimize data movement cost



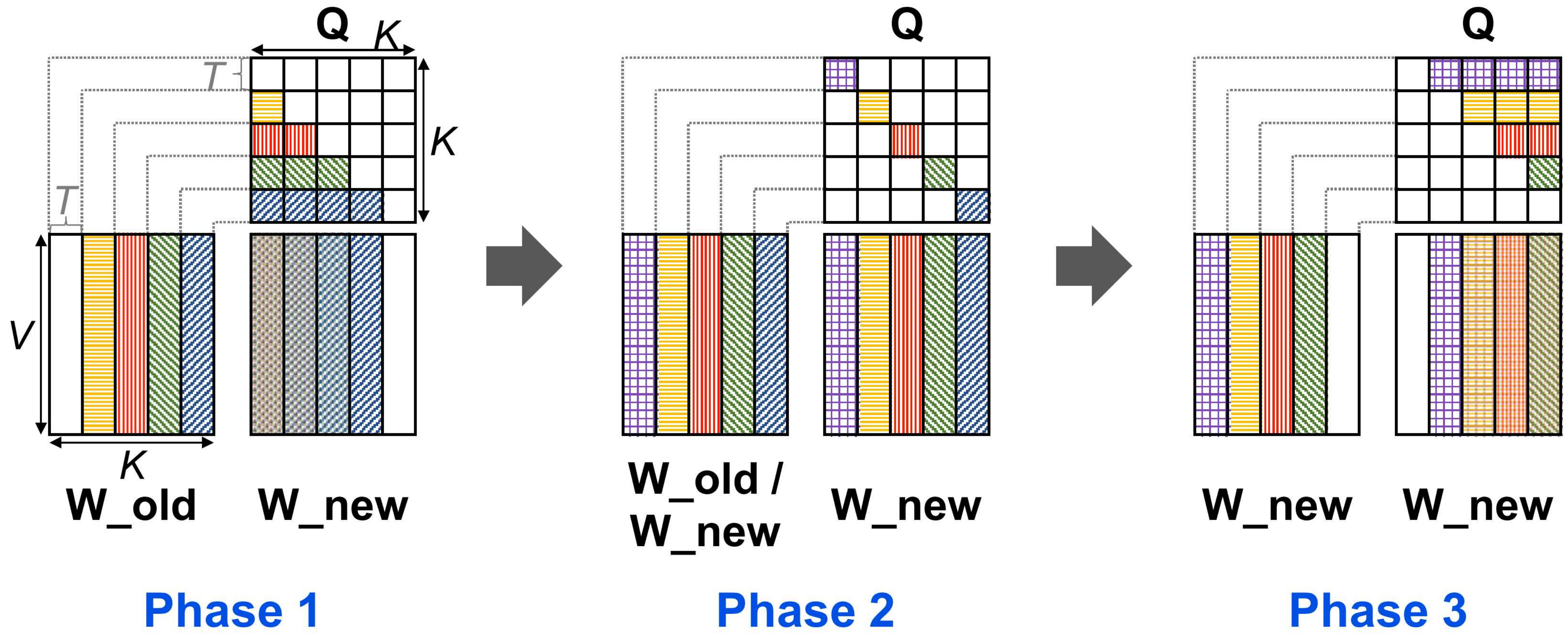
Original HALS-based NMF

Our ALO-NMF

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



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

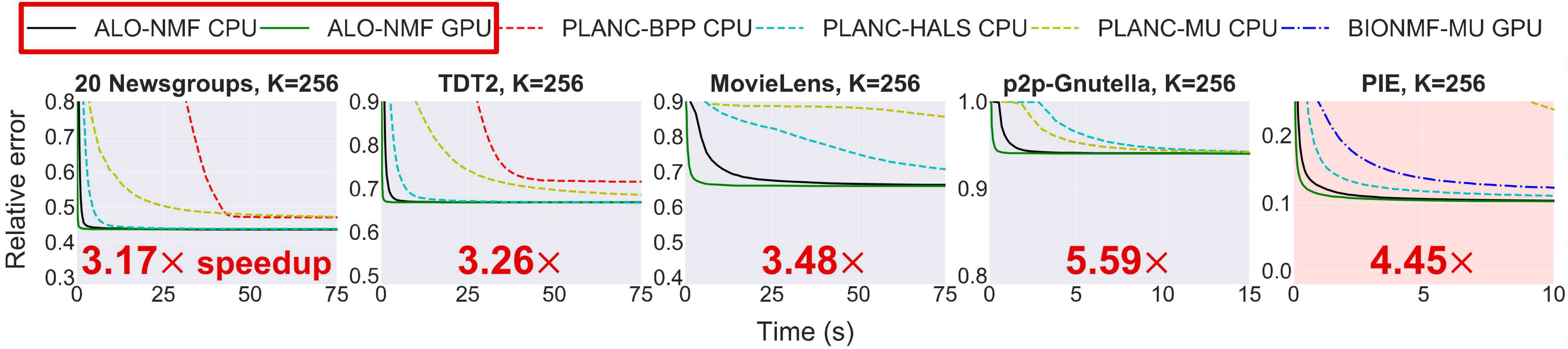
Data movement cost for updating **W**

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

2.29× reduced

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

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**
- Our ALO-NMF achieved **2.29**× lower data movement and ~**4.45**× speedup compared to the existing state-of-the-art parallel NMF implementations

Please check out our paper to learn more about this work.

Thank you. ☺