



# $\mathcal{H}$ -Matrix Formats with Floating Point Compression

Storage and Arithmetic

**Ronald Kriemann**

Max-Planck-Institute MiS Leipzig

**SIAM PP26**

**ZIB / FU Berlin**

**MAX PLANCK INSTITUTE**  
FOR MATHEMATICS IN THE SCIENCES



# Hierarchical Matrices

---

# Hierarchical Matrices

Approximate dense data  $M_{\tau,\sigma} \in \mathbb{C}^{\#\tau \times \#\sigma}$  of  $M \in \mathbb{C}^{n \times n}$  by

$$U_{\tau,\sigma} \cdot V_{\tau,\sigma}^H$$

with  $U_{\tau,\sigma} \in \mathbb{C}^{\#\tau \times k}$ ,  $V_{\tau,\sigma} \in \mathbb{C}^{\#\sigma \times k}$  and

$$k \ll \min(\#\tau, \#\sigma)$$

such that

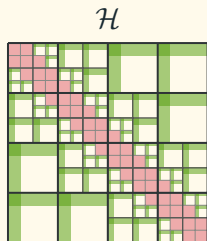
$$\|M_{\tau,\sigma} - U_{\tau,\sigma} V_{\tau,\sigma}^H\| \leq \delta \quad \text{or}$$

$$\|M_{\tau,\sigma} - U_{\tau,\sigma} V_{\tau,\sigma}^H\| \leq \varepsilon \|M_{\tau,\sigma}\|$$

yielding an approximation  $\widetilde{M}$  of  $M$  with  $\mathcal{O}(n \log^\alpha n)$  storage.

In the literature, many different formats of (hierarchical) lowrank matrices exist.

# Hierarchical Matrices



$$M_{\tau,\sigma} = U_{\tau,\sigma} \cdot V_{\tau,\sigma}^H$$

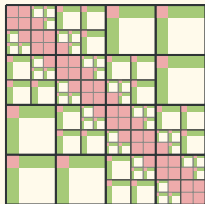
with

$$U_{\tau,\sigma} \in \mathbb{R}^{\#\tau \times k}, V_{\tau,\sigma} \in \mathbb{R}^{\#\sigma \times k}$$

$$\mathcal{O}(n \log n)$$

# Hierarchical Matrices

$\mathcal{H}$



$$M_{\tau,\sigma} = W_{\tau,\sigma} \cdot S_{\tau,\sigma} \cdot X_{\tau,\sigma}^H$$

with

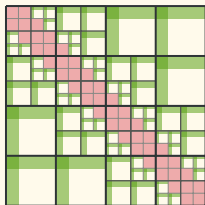
$$W_{\tau,\sigma} \in \mathbb{R}^{\#\tau \times k}, X_{\tau,\sigma} \in \mathbb{R}^{\#\sigma \times k}$$

$$S_{\tau,\sigma} \in \mathbb{R}^{k \times k},$$

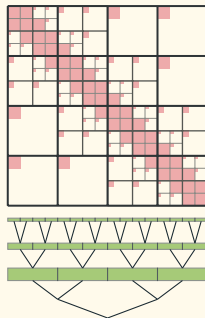
$$\mathcal{O}(n \log n)$$

# Hierarchical Matrices

$\mathcal{H}$



Uniform- $\mathcal{H}$



$$M_{\tau,\sigma} = W_{\tau,\sigma} \cdot S_{\tau,\sigma} \cdot X_{\tau,\sigma}^H$$

with

$$W_{\tau,\sigma} \in \mathbb{R}^{\#\tau \times k}, X_{\tau,\sigma} \in \mathbb{R}^{\#\sigma \times k}, \\ S_{\tau,\sigma} \in \mathbb{R}^{k \times k},$$

$$\mathcal{O}(n \log n)$$

$$M_{\tau,\sigma} = \mathcal{W}_{\tau} \cdot S_{\tau,\sigma} \cdot \mathcal{X}_{\sigma}^H$$

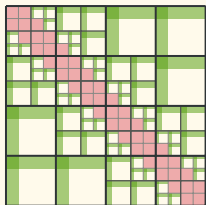
with

$$\mathcal{W}_{\tau} \in \mathbb{R}^{\#\tau \times k}, \mathcal{X}_{\sigma} \in \mathbb{R}^{\#\sigma \times k}, \\ S_{\tau,\sigma} \in \mathbb{R}^{k \times k}$$

$$\underline{\mathcal{O}(n)} + \mathcal{O}(n \log n)$$

# Hierarchical Matrices

$\mathcal{H}$



$$M_{\tau,\sigma} = W_{\tau,\sigma} \cdot S_{\tau,\sigma} \cdot X_{\tau,\sigma}^H$$

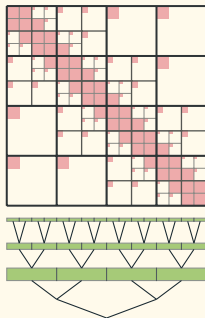
with

$$W_{\tau,\sigma} \in \mathbb{R}^{\#\tau \times k}, X_{\tau,\sigma} \in \mathbb{R}^{\#\sigma \times k}$$

$$S_{\tau,\sigma} \in \mathbb{R}^{k \times k},$$

$$\mathcal{O}(n \log n)$$

Uniform- $\mathcal{H}$



$$M_{\tau,\sigma} = \mathcal{W}_{\tau} \cdot S_{\tau,\sigma} \cdot \mathcal{X}_{\sigma}^H$$

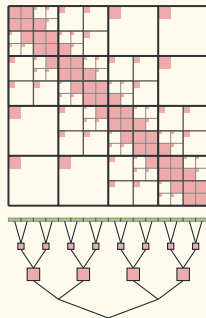
with

$$\mathcal{W}_{\tau} \in \mathbb{R}^{\#\tau \times k}, \mathcal{X}_{\sigma} \in \mathbb{R}^{\#\sigma \times k},$$

$$S_{\tau,\sigma} \in \mathbb{R}^{k \times k}$$

$$\underline{\mathcal{O}(n)} + \mathcal{O}(n \log n)$$

$\mathcal{H}^2$



$$M_{\tau,\sigma} = \widetilde{\mathcal{W}}_{\tau} \cdot S_{\tau,\sigma} \cdot \widetilde{\mathcal{X}}_{\sigma}^H$$

with

*implicit*  $\widetilde{\mathcal{W}}_{\tau}, \widetilde{\mathcal{X}}_{\sigma}$

$$\mathcal{O}(n)$$

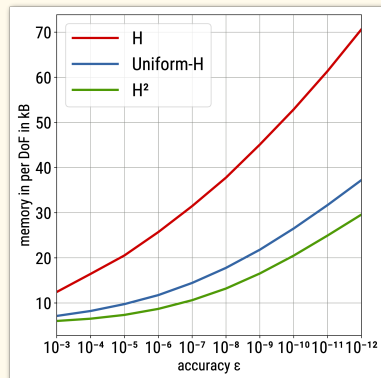
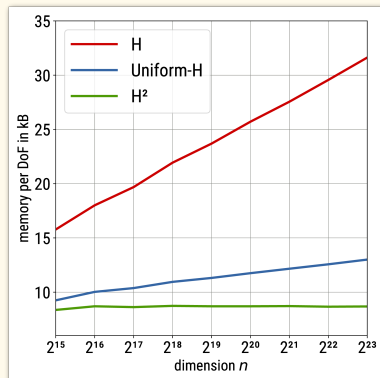
# Hierarchical Matrices

## Memory Complexity

Model problem is Laplace SLP over  $\Gamma = \{x \in \mathbb{R}^3 : \|x\|_2 = 1\}$ :

$$\int_{\Gamma} \frac{1}{\|x - y\|} u(x) dy = f(x), \quad x \in \Gamma$$

with unknown  $u$  and right-hand side  $f$ .



# Compressed Storage

---

# Compressed Storage

## Idea

Replace FP64/FP32 storage for lowrank/dense blocks by a *lossy* compression scheme but keep FP64/FP32 as *compute* precision.

# Compressed Storage

## Idea

Replace FP64/FP32 storage for lowrank/dense blocks by a *lossy* compression scheme but keep FP64/FP32 as *compute* precision.

## Options

- compression libraries: ZFP<sup>1,2</sup>, BLOSC, SZ (?), ...

---

<sup>1</sup>K, Ltaief, Luong, Pérez, Im, Keyes: "High-Performance Spatial Data Compression for Scientific Applications", Euro-Par 2022

<sup>2</sup>Claus, Ghysels, Liu, Nhan, Thirumalaisamy, Bhalla, Li: "Sparse Approx. Multifrontal Factorization with Composite Compr. Methods", ACM TOMS, 2023

# Compressed Storage

## Idea

Replace FP64/FP32 storage for lowrank/dense blocks by a *lossy* compression scheme but keep FP64/FP32 as *compute* precision.

## Options

- compression libraries: ZFP<sup>1,2</sup>, BLOSC, SZ (?), ...
- IEEE-754 related<sup>3,4</sup>,

---

<sup>1</sup>K, Ltaief, Luong, Pérez, Im, Keyes: "High-Performance Spatial Data Compression for Scientific Applications", Euro-Par 2022

<sup>2</sup>Claus, Ghysels, Liu, Nhan, Thirumalaisamy, Bhalla, Li: "Sparse Approx. Multifrontal Factorization with Composite Compr. Methods", ACM TOMS, 2023

<sup>3</sup>K.: "Hierarchical Low-Rank Arithmetic with Floating Point Compression", SIAM SISC, 2025

<sup>4</sup>Carson, Chen, Liu: "Mixed Precision HODLR Matrices", SIAM SISC, 2025

# Compressed Storage

## Idea

Replace FP64/FP32 storage for lowrank/dense blocks by a *lossy* compression scheme but keep FP64/FP32 as *compute* precision.

## Options

- compression libraries: ZFP<sup>1,2</sup>, BLOSC, SZ (?), ...
- IEEE-754 related<sup>3,4</sup>,
- Posits<sup>3</sup>,
- ...

---

<sup>1</sup>K, Ltaief, Luong, Pérez, Im, Keyes: "High-Performance Spatial Data Compression for Scientific Applications", Euro-Par 2022

<sup>2</sup>Claus, Ghysels, Liu, Nhan, Thirumalaisamy, Bhalla, Li: "Sparse Approx. Multifrontal Factorization with Composite Compr. Methods", ACM TOMS, 2023

<sup>3</sup>K: "Hierarchical Low-Rank Arithmetic with Floating Point Compression", SIAM SISC, 2025

<sup>4</sup>Carson, Chen, Liu: "Mixed Precision HODLR Matrices", SIAM SISC, 2025

# Compressed Storage

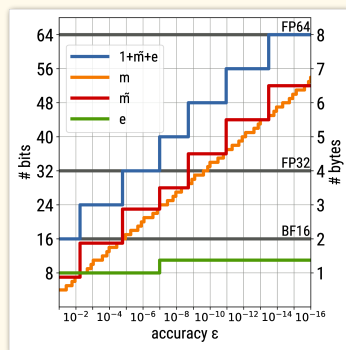
## Floating Point eXtended (FPX)

Use *truncated* FP64/FP32 format for error adaptive storage with  $m_\epsilon := \lceil -\log_2 \epsilon \rceil$  mantissa bits.

Keep  $e$  exponent bits from FP64/FP32 format.

Round up  $m_\epsilon$  to  $\tilde{m}_\epsilon$  such that  $1 + e + \tilde{m}_\epsilon$  is multiple of 8 (permits AVX-512<sup>1</sup>).

$\lceil -\log_2 \epsilon \rceil$	$e$	$m_\epsilon$	Format
0...7	8	7	BF16
8...15	8	15	FP24
16...23	8	23	FP32
24...28	11	28	FP40
29...36	11	36	FP48
37...44	11	44	FP56
45...52	11	52	FP64



<sup>1</sup>Amestoy, Jego, L'Excellent, Mary, Pichon: "BLAS-based Block Memory Accessor with Applications to Mixed Precision Sparse Direct Solvers", Preprint, 2025

# Compressed Storage

## Block Floating Point (BFP)<sup>1</sup>

Let  $v \in \mathbb{R}^n$ . Rescale each  $v_i$ ,  $1 \leq i \leq n$ , into *integer* (fixed-point) range

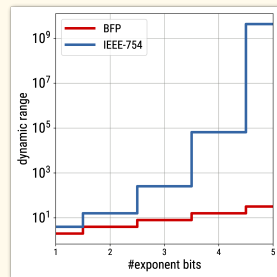
$$[-s_{\text{int}}, s_{\text{int}}]$$

with

$$s_{\text{int}} := 2^{\tilde{e} + m_\varepsilon} - 1.$$

and

$$\tilde{e} := \left\lceil \log_2 \frac{\max_{i \leq n} |v_i|}{\min_{i \leq n} |v_i|} \right\rceil$$



**Problem:**  $\tilde{e}$  is small compared to IEEE-754 related formats.

**Related:** micro scaling formats MXFP?/MXINT?.

<sup>1</sup>Wilkinson: "Rounding Errors in Algebraic Processes", 1963

# Variable Accuracy for Low-Rank (VALR)<sup>1</sup>

Given  $\|M_{\tau,\sigma} - W_{\tau,\sigma}\Sigma_{\tau,\sigma}X_{\tau,\sigma}^H\| \leq \delta$  with singular values  $\sigma_i$  in  $\Sigma_{\tau,\sigma}$ . Let  $\widetilde{W}_{\tau,\sigma}, \widetilde{X}_{\tau,\sigma}$  be the representations of  $W_{\tau,\sigma}, X_{\tau,\sigma}$  with the  $i$ 'th column of  $W_{\tau,\sigma}, X_{\tau,\sigma}$  stored with accuracy

$$\delta_i = \frac{\delta}{\sigma_i}$$

Then the representation error is bounded by<sup>1,2</sup>

$$\|W_{\tau,\sigma}\Sigma_{\tau,\sigma}X_{\tau,\sigma}^H - \widetilde{W}_{\tau,\sigma}\Sigma_{\tau,\sigma}\widetilde{X}_{\tau,\sigma}^H\| \leq \delta \left( 2k + \delta \sum_{i=1}^k \frac{1}{\sigma_i} \right)$$

If the  $i$ 'th vector of a cluster basis  $\mathcal{W}_\tau$  is stored with accuracy  $\delta_i$ , then<sup>3</sup>:

$$\|\mathcal{W}_\tau - \widetilde{\mathcal{W}}_\tau\| \leq \delta \sqrt{\sum_{i=1}^k \frac{1}{\sigma_i^2}}.$$

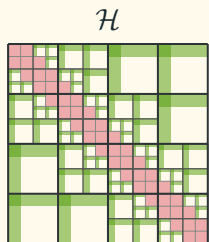
<sup>1</sup>Amestoy, Boiteau, Buttari, Gerest, Jézéquel, L'Excellent, Mary: "Mixed precision low-rank approximations and their application to block low-rank LU factorization", IMA J. of Num. Analysis, 2022

<sup>2</sup>K: "Hierarchical Low-Rank Arithmetic with Floating Point Compression", SIAM SISC, 2025

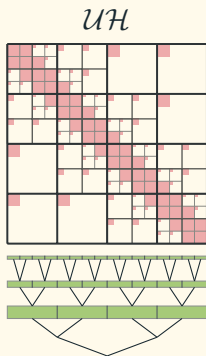
<sup>3</sup>K: "Floating Point Compression of Hierarchical Matrix Formats and its Impact on Matrix-Vector Multiplication", Preprint, 2026

# Variable Accuracy for Low-Rank (VALR)<sup>1</sup>

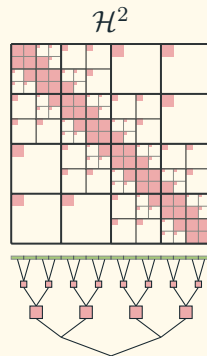
Depending on the matrix format, VALR can be applied to different data.



all *lowrank blocks*



all *cluster bases*



all *leaf cluster bases*

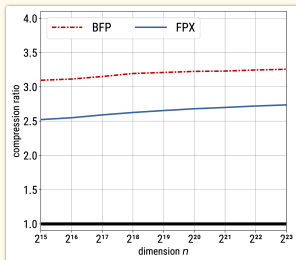
For *everything else* the compression scheme is applied directly!

Best compression is expected for  $\mathcal{H}$  and least for  $\mathcal{H}^2$ .

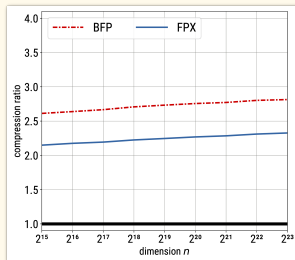
# Compression Results

## Compression Ratio for FPX/BFP

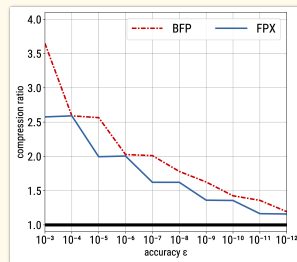
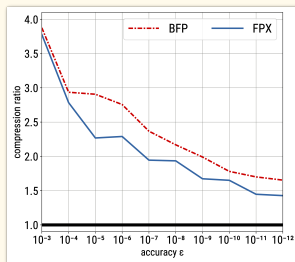
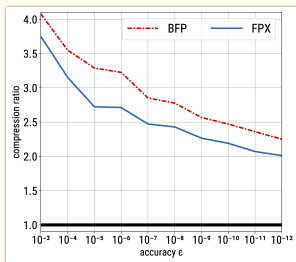
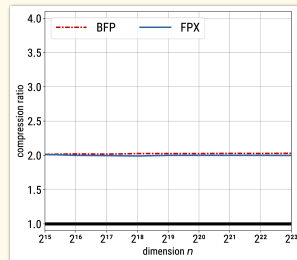
$\mathcal{H}$



$\mathcal{UH}$

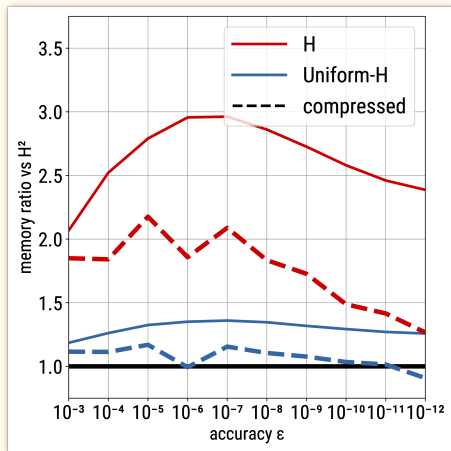
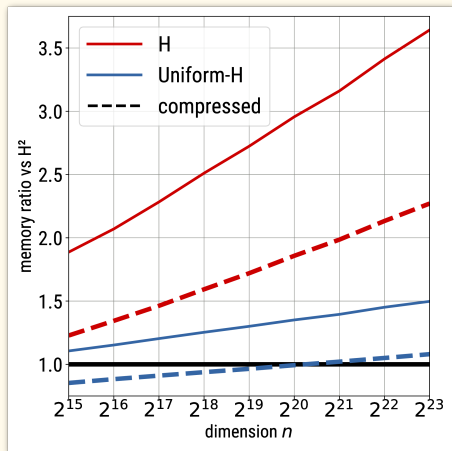


$\mathcal{H}^2$



# Compression Results

Ratio of  $\mathcal{H}/U\mathcal{H}$  vs.  $\mathcal{H}^2$



Compression reduced the gap between  $\mathcal{H}/U\mathcal{H}$  and  $\mathcal{H}^2$ .

# Matrix-Vector Multiplication

---

# Matrix-Vector Multiplication

Low-rank arithmetic normally relies on BLAS (and LAPACK) for arithmetic kernels.

Such kernels can not be used directly with compressed storage.

## Options

- block-wise decompression<sup>1,2</sup>
  - ↪ reuse BLAS/LAPACK
- on-the-fly decompression (*memory accessor*<sup>3</sup>)
  - ↪ reimplement kernels

```
function hmvm( $\alpha, M, x, y$ )  
  for  $(\tau, \sigma) \in \mathcal{L}(M)$  do  
    if  $(\tau, \sigma)$  is admissible then  
      gemv( $V_{\tau, \sigma}^H, x|_{\sigma}, t$ );  
      gemv( $U_{\tau, \sigma}, t, y'$ );  
    else  
      gemv( $D_{\tau, \sigma}, x|_{\sigma}, y'$ );  
      axpy( $\alpha, y', y|_{\tau}$ );
```

<sup>1</sup>K: "Hierarchical Low-Rank Arithmetic with Floating Point Compression", SIAM SISC, 2025

<sup>2</sup>Amestoy, Jeco, L'Excellent, Mary, Pichon: "BLAS-based Block Memory Accessor with Applications to Mixed Precision Sparse Direct Solvers", Preprint, 2025

<sup>3</sup>Anzt, Flegar, Grützmacher, Quintana-Orti: "Toward a modular precision ecosystem for high-performance computing", Int. J. of HPC Appl, 2019.

# Matrix-Vector Multiplication

Low-rank arithmetic normally relies on BLAS (and LAPACK) for arithmetic kernels.

Such kernels can not be used directly with compressed storage.

## Options

- block-wise decompression<sup>1,2</sup>
  - ↪ reuse BLAS/LAPACK
- on-the-fly decompression (*memory accessor*<sup>3</sup>)
  - ↪ reimplement kernels

In the following, block-wise (FPX) and on-the-fly (BFP) is used **without** BLAS.

**Remark:**  $\mathcal{H}$ -MVM is memory bandwidth bound.

```
function hmvm( $\alpha, M, x, y$ )
  for  $(\tau, \sigma) \in \mathcal{L}(M)$  do
    if  $(\tau, \sigma)$  is admissible then
      zgemv( $V_{\tau, \sigma}^H, x|_{\sigma}, t$ );
      zgemv( $U_{\tau, \sigma}, t, y'$ );
    else
      zgemv( $D_{\tau, \sigma}, x|_{\sigma}, y'$ );
      axpy( $\alpha, y', y|_{\tau}$ );
```

```
function zgemv( $D, x, y$ )
  for  $0 \leq j < m$  do
    for  $0 \leq i < n$  do
       $y_i += \text{decomp}(D_{ij})x_j$ ;
```

<sup>1</sup>K: "Hierarchical Low-Rank Arithmetic with Floating Point Compression", SIAM SISC, 2025

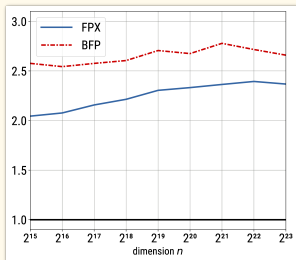
<sup>2</sup>Amestoy, Jeco, L'Excellent, Mary, Pichon: "BLAS-based Block Memory Accessor with Applications to Mixed Precision Sparse Direct Solvers", Preprint, 2025

<sup>3</sup>Anzt, Flegar, Grützmacher, Quintana-Orti: "Toward a modular precision ecosystem for high-performance computing", Int. J. of HPC Appl, 2019.

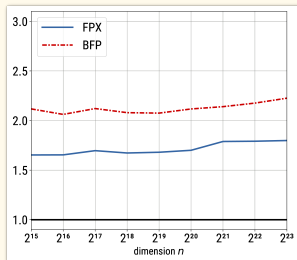
# Matrix-Vector Multiplication

## Speedup vs. uncompressed MVM

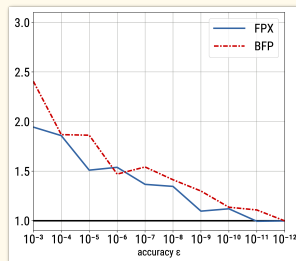
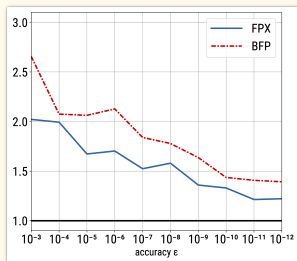
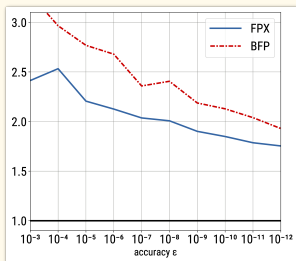
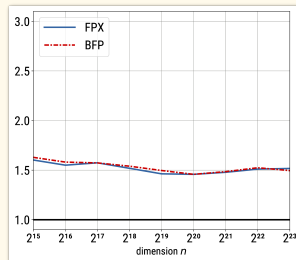
$\mathcal{H}$



$UH$

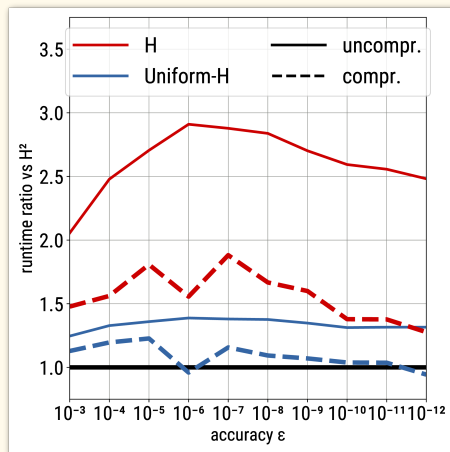
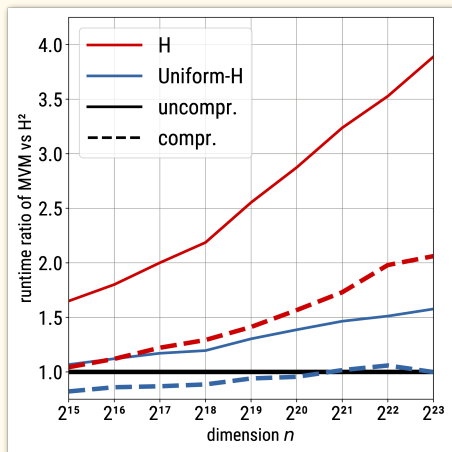


$\mathcal{H}^2$



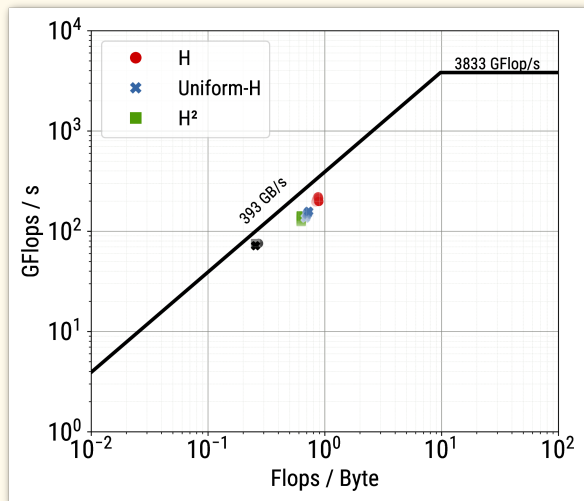
# Matrix-Vector Multiplication

## Ratio of $\mathcal{H}/U\mathcal{H}$ vs. $\mathcal{H}^2$



# Matrix-Vector Multiplication

## Roofline Plot



AMD Epyc 9554, 64 cores, 12x DDR5-4800

## Percentage of Peak Perf.

	uncompr.	compr.
$\mathcal{H}$	78%	64%
$\mathcal{UH}$	72%	56%
$\mathcal{H}^2$	75%	55%

# Conclusion

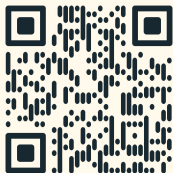
---

# Conclusion

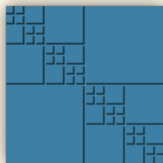
Accuracy adaptive floating point storage

- *reduces* storage demands,
- *regains* some of the  $\mathcal{H}^2$ -compression advantage and
- *accelerates* matrix-vector multiplication.

Compression ratio *dominates* decompression speed.



[doi.org/10.1137/24M1649009](https://doi.org/10.1137/24M1649009)



[libHLR.org](https://libHLR.org)



[arxiv.org/abs/2405.03456](https://arxiv.org/abs/2405.03456)



# Thank You

**MAX PLANCK INSTITUTE**  
FOR MATHEMATICS IN THE SCIENCES

