# PaSTRI: Error-Bounded Lossy Compression for Two-Electron Integrals in Quantum Chemistry


Ali Murat Gok (Northwestern University, USA)

Sheng Di (Argonne National Laboratory, USA)

Yuri Alexeev (Argonne National Laboratory, USA)

**Dingwen Tao (The University of Alabama, USA)**

Vladimir Mironov (Lomonosov Moscow State University, Russia)

Xin Liang (University of California, Riverside, USA )

Franck Cappello (Argonne National Laboratory, USA)


September 2018

# ANL SZ Lossy compressor (ECP EZ project)

Sheng Di (ANL), Xin Liang (U. C. Riverside), Dingwen Tao (U. Alabama), Franck Cappello (Lead)

**Key features:**

- **Production quality** lossy compressor for scientific data respecting user set error bounds
- **For 1D, 2D, 3D structured and unstructured datasets**. E.g. 3D simulation fields, instruments data, time series.
- For **floating point** and **integer** data
- **Strict error controls** (absolute error, relative error, PSNR, *error distribution*)
- **Thorough testing procedures**, bug tracking, tests on the CORAL systems
- **Integrated in the ADIOS, HDF5 and PnetCDF I/O libraries**.
- Reader/Writer for ADIOS, HDF5, netCDF
- **Optimized compression ratios** (transforms, decompositions, multiple predictors, compression in time, lossless compression, etc.)
- **High compression/decompression speed** (*MPI + OpenMP*)

By the way, compression is also an art

*César BALDACCINI dit CÉSAR (1921 - 1998)*
*Compression de couverts en métal argenté.*

SZ is supported by the DOE/NNSA **ECP EZ and ExaSky** Projects and the **NSF Aletheia** Project.

# Outline

- Introduction
- Background
  - Electron Repulsion Integrals (ERIs)
  - ERI Data Representation
- Patterns in ERIs
- PaSTRI Compression
- Optimizations of Quantization & Encoding
- Experimental Evaluation
- Conclusion

# Outline

- Introduction
- Background
  - Electron Repulsion Integrals (ERIs)
  - ERI Data Representation
- Patterns in ERIs
- PaSTRI Compression
- Optimizations of Quantization & Encoding
- Experimental Evaluation
- Conclusion

# Introduction

- HPC applications work with extremely large data (Petabytes!)
- Large data → System bottlenecks (Memory, Storage, Bandwidth)

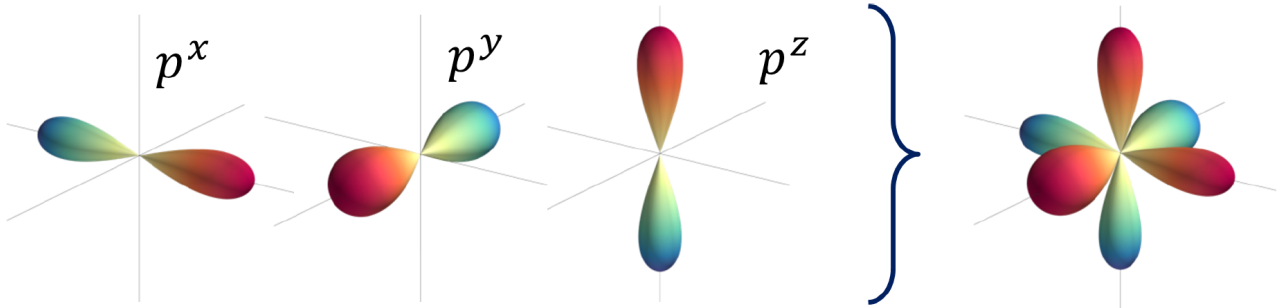
# Introduction

- HPC applications work with extremely large data (Petabytes!)
- Large data → System bottlenecks (Memory, Storage, Bandwidth)
- Electron Repulsion Integrals (ERIs):
  - Large data size: Petabytes
  - Costly computations: $O(N^4)$
  - Data reuse: ~10-30 times
- PaSTRI: **Pa**ttern **S**caling for **T**wo-Electron **R**epulsion **I**ntegrals
  - Calculate and compress once
  - Decompress whenever needed

# Outline

- Introduction
- Background
  - Electron Repulsion Integrals (ERIs)
  - ERI Data Representation
- Patterns in ERIs
- PaSTRI Compression
- Optimizations of Quantization & Encoding
- Experimental Evaluation
- Conclusion

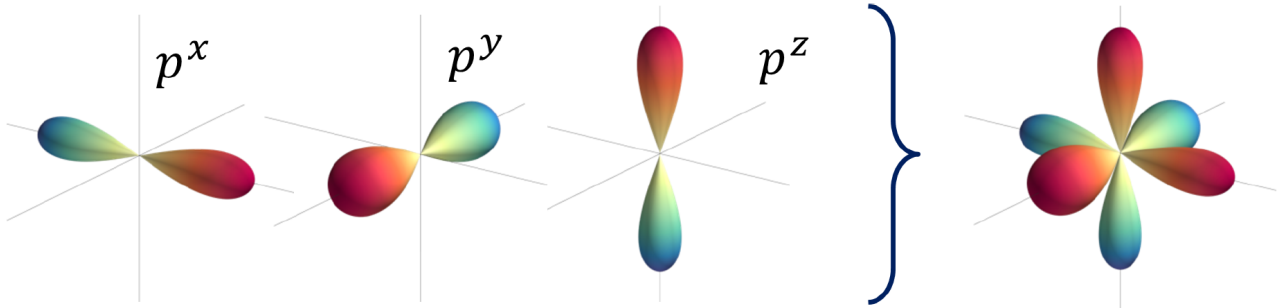
# Electron Repulsion Integrals (ERIs)




$p$-type basis functions

$p$-shell

# Electron Repulsion Integrals (ERIs)

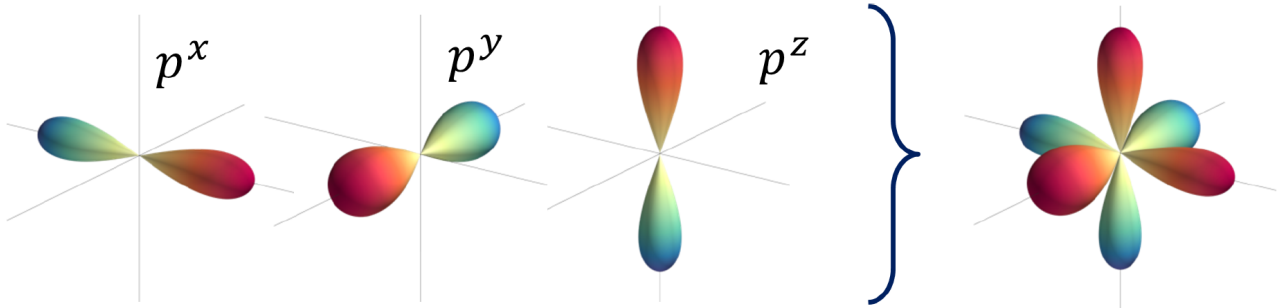



$p$-type basis functions

$p$-shell

| Orbital | # of BFs |
|---|---|
| s | 1 |
| p | 3 |
| d | 6 |
| f | 10 |
| g | 15 |
| … | … |

# Electron Repulsion Integrals (ERIs)




$p$-type basis functions

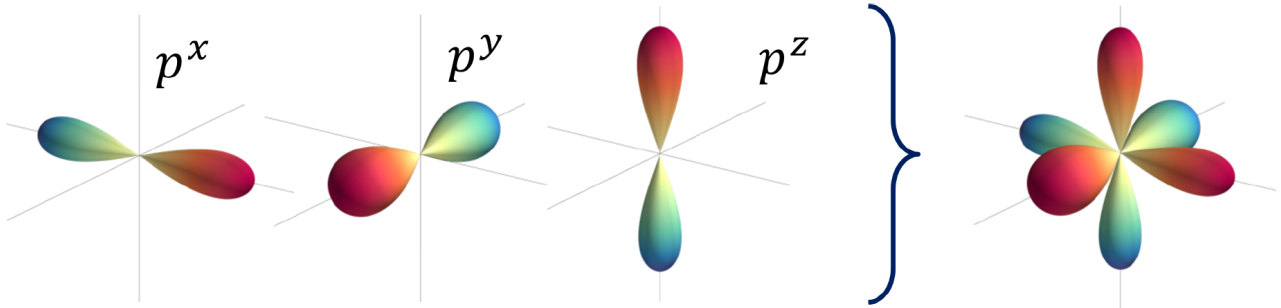
$p$-shell

| Orbital | # of BFs |
|---|---|
| s | 1 |
| p | 3 |
| d | 6 |
| f | 10 |
| g | 15 |
| … | … |

- ERIs are a part of solving the Schrödinger equation:

$$(i,j|k,l) = \int_{3D} d\mathbf{r}_1 \int_{3D} d\mathbf{r}_2 \frac{\phi_i(\mathbf{r}_1)\phi_j(\mathbf{r}_1)\phi_k(\mathbf{r}_2)\phi_l(\mathbf{r}_2)}{|\mathbf{r}_1 - \mathbf{r}_2|},$$

# Electron Repulsion Integrals (ERIs)




$p$-type basis functions

$p$-shell

| Orbital | # of BFs |
|---|---|
| s | 1 |
| p | 3 |
| d | 6 |
| f | 10 |
| g | 15 |
| … | … |

- ERIs are a part of solving the Schrödinger equation:

$$(i,j|k,l) = \int_{3D} d\mathbf{r}_1 \int_{3D} d\mathbf{r}_2 \frac{\phi_i(\mathbf{r}_1)\phi_j(\mathbf{r}_1)\phi_k(\mathbf{r}_2)\phi_l(\mathbf{r}_2)}{|\mathbf{r}_1 - \mathbf{r}_2|}, \longrightarrow \text{scale as } O(N^4)$$

# Outline

- Introduction
- Background
  - Electron Repulsion Integrals (ERIs)
  - ERI Data Representation
- Patterns in ERIs
- PaSTRI Compression
- Optimizations of Quantization & Encoding
- Experimental Evaluation
- Conclusion

# ERI Data Representation

- (ij|kl) representation examples: (dd|dd), (dp|ff), (ps|df), …

# ERI Data Representation

- (ij|kl) representation examples: (dd|dd), (dp|ff), (ps|df), …

(dd|dd) block

| | |
|---|---|
| 0,0,0,0 | 1234E-6 |
| 0,0,0,1 | 2345E-7 |
| ⋮ | ⋮ |
| 0,0,0,5 | 3456E-6 |
| 0,0,1,0 | 4567E-8 |
| ⋮ | ⋮ |
| 5,5,5,5 | 6789E-5 |

6*6*6*6 = 1296 pts

| Orbital | # of BFs |
|---|---|
| s | 1 |
| p | 3 |
| d | 6 |
| f | 10 |
| g | 15 |
| … | … |

# ERI Data Representation

- (ij|kl) representation examples: (dd|dd), (dp|ff), (ps|df), …

(dd|dd) block

| | |
|---|---|
| 0,0,0,0 | 1234E-6 |
| 0,0,0,1 | 2345E-7 |
| ⋮ | ⋮ |
| 0,0,0,5 | 3456E-6 |
| 0,0,1,0 | 4567E-8 |
| ⋮ | ⋮ |
| 5,5,5,5 | 6789E-5 |

6*6*6*6 = 1296 pts

(dp|ff) block

| | |
|---|---|
| 0,0,0,0 | 1234E-6 |
| 0,0,0,1 | 2345E-7 |
| ⋮ | ⋮ |
| 0,0,0,9 | 3456E-6 |
| 0,0,1,0 | 4567E-8 |
| ⋮ | ⋮ |
| 5,2,9,9 | 6789E-5 |

6*3*10*10 = 1800 pts

| Orbital | # of BFs |
|---|---|
| s | 1 |
| p | 3 |
| d | 6 |
| f | 10 |
| g | 15 |
| … | … |

# ERI Data Representation

- (ij|kl) representation examples: (dd|dd), (dp|ff), (ps|df), …

(ff|ff) block

| | |
|---|---|
| 0,0,0,0 | 1234E-6 |
| 0,0,0,1 | 2345E-7 |
| ⋮ | ⋮ |
| 0,0,0,9 | 3456E-6 |
| 0,0,1,0 | 4567E-8 |
| ⋮ | ⋮ |
| 9,9,9,9 | 6789E-5 |

10*10*10*10 = 10000 pts

| Orbital | # of BFs |
|---|---|
| s | 1 |
| p | 3 |
| d | 6 |
| f | 10 |
| g | 15 |
| … | … |

# ERI Data Representation

- (ij|kl) representation examples: (dd|dd), (dp|ff), (ps|df), …

1D Index

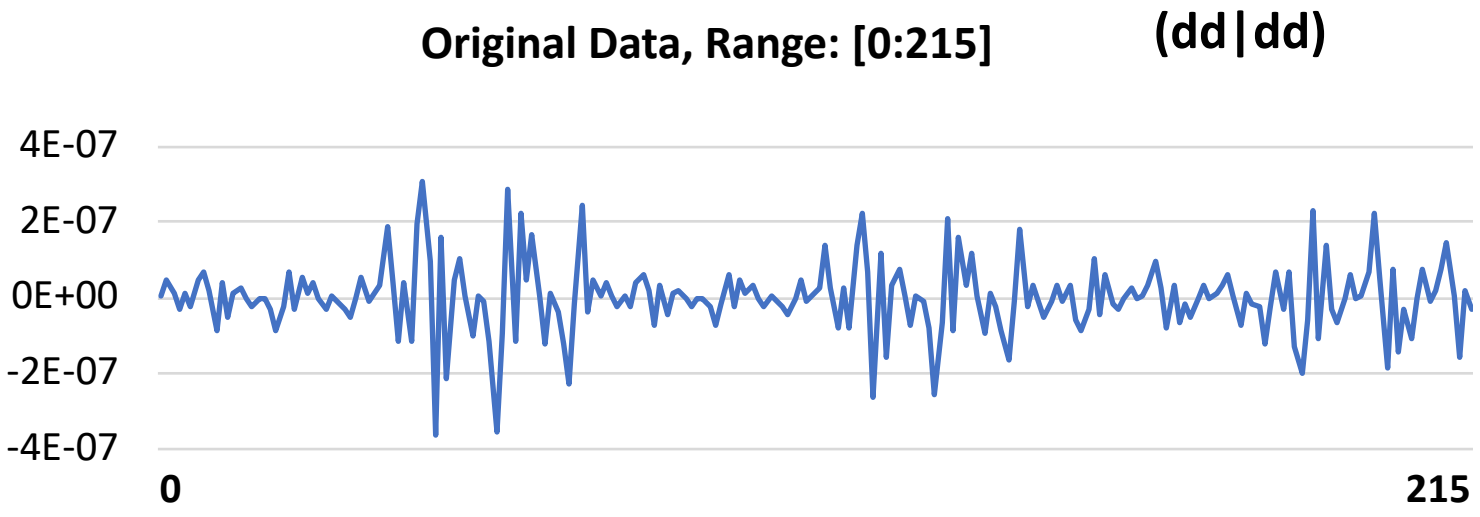
| (ff\|ff) | | (dd\|dd) | | (fd\|ps) | |
|---|---|---|---|---|---|
| 1D | 4D | 1D | 4D | 1D | 4D |
| 0 | 0,0,0,0 | 0 | 0,0,0,0 | 0 | 0,0,0,0 |
| 1 | 0,0,0,1 | 1 | 0,0,0,1 | 1 | 0,0,1,0 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 9 | 0,0,0,9 | 6 | 0,0,0,6 | 19 | 1,0,1,0 |
| 10 | 0,0,1,0 | 7 | 0,0,1,0 | 20 | 1,0,2,0 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 9999 | 9,9,9,9 | 1295 | 5,5,5,5 | 179 | 9,5,2,0 |

| Orbital | # of BFs |
|---|---|
| s | 1 |
| p | 3 |
| d | 6 |
| f | 10 |
| g | 15 |
| … | … |

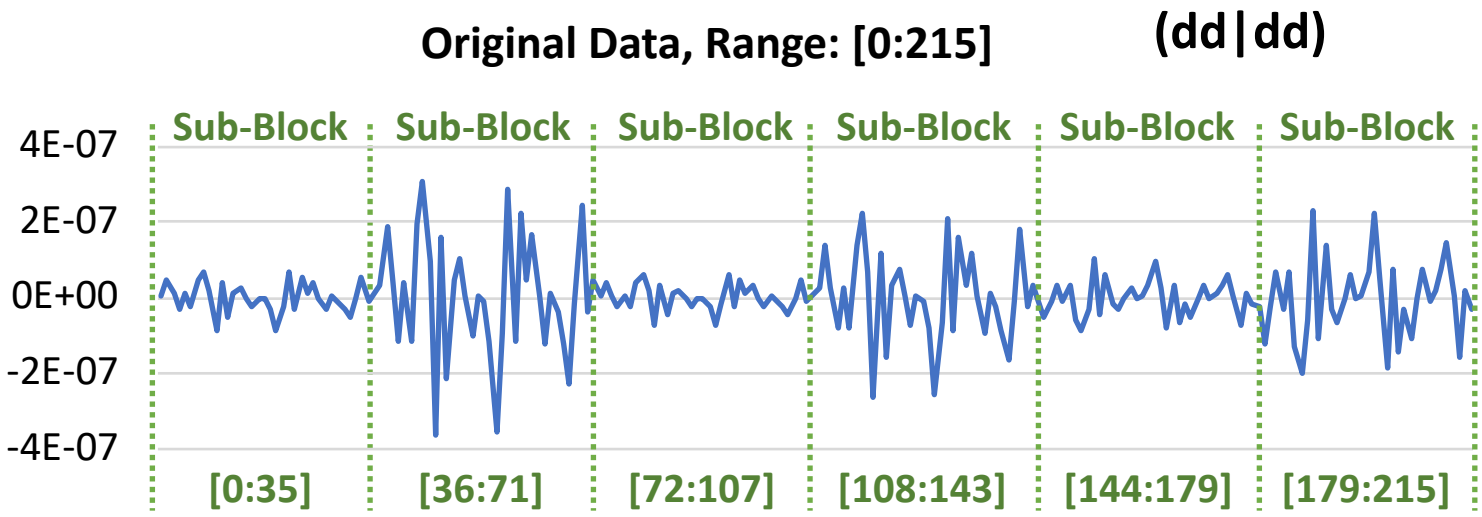
# Outline

- Introduction
- Background
  - Electron Repulsion Integrals (ERIs)
  - ERI Data Representation
- Patterns in ERIs
- PaSTRI Compression
- Optimizations of Quantization & Encoding
- Experimental Evaluation
- Conclusion

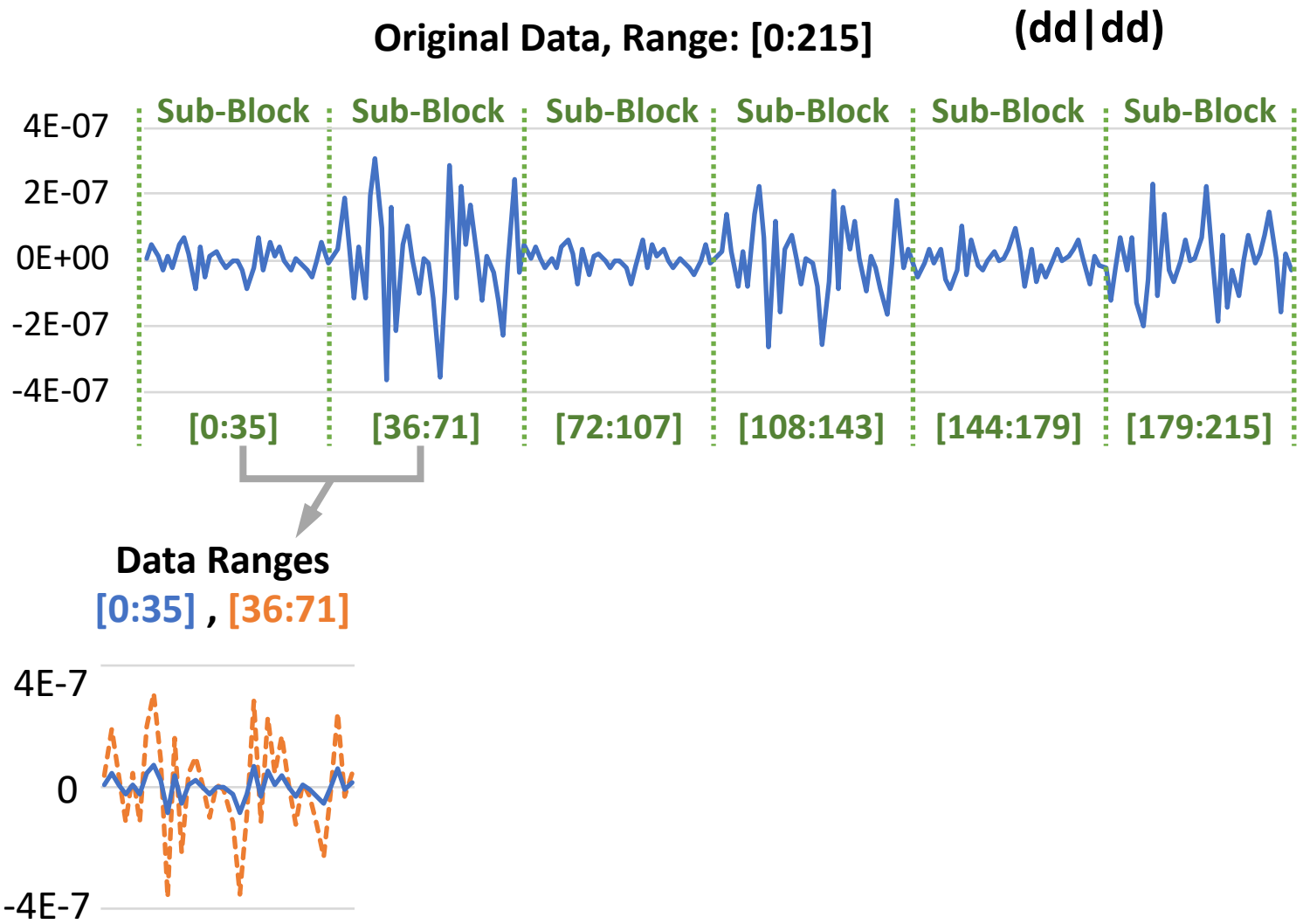
# Patterns in ERIs



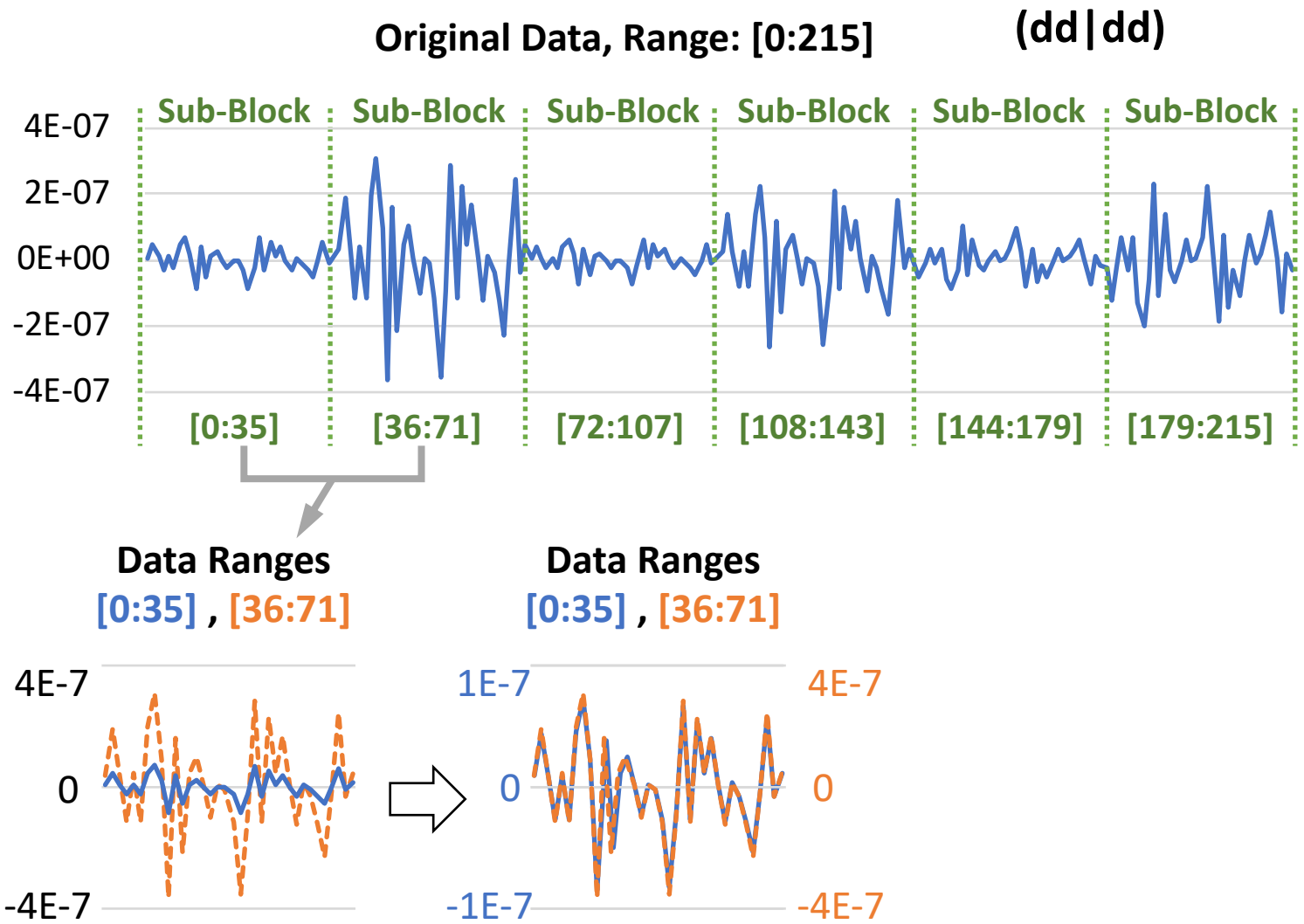
# Patterns in ERIs



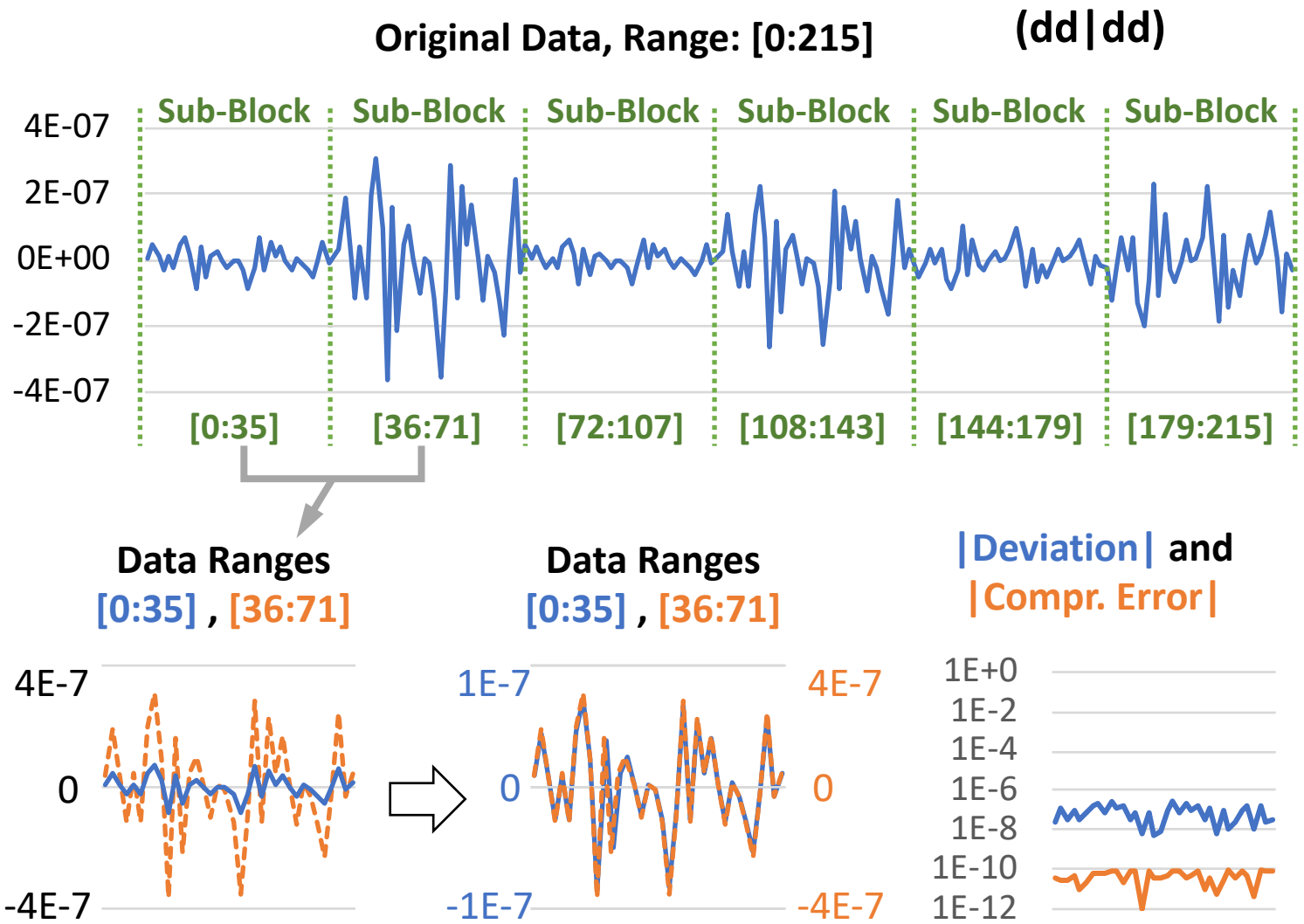
# Patterns in ERIs



# Patterns in ERIs



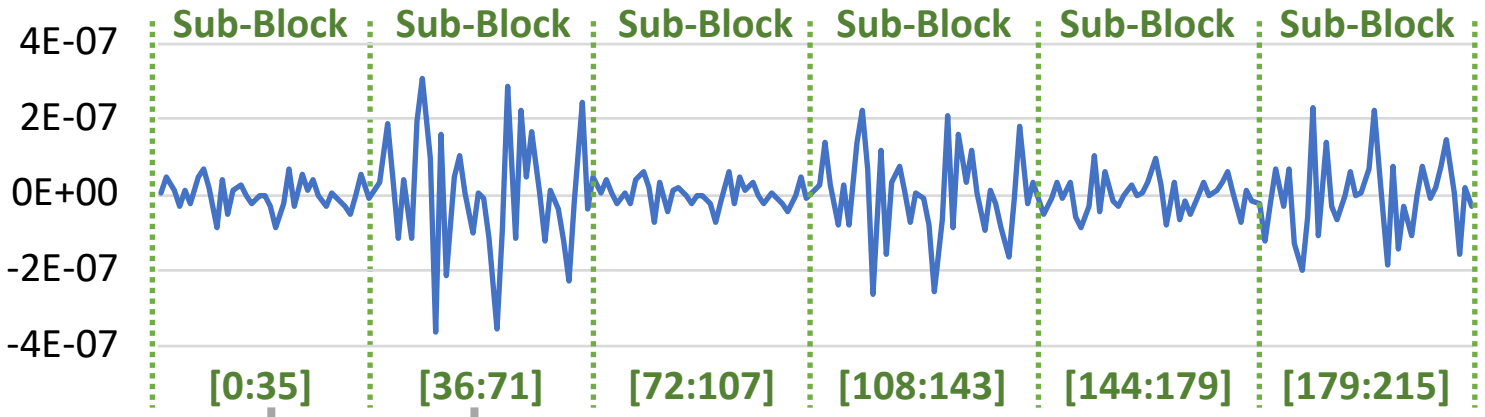
# Patterns in ERIs




**Reasonable Absolute Error Bound: $10^{-10}$**

# Patterns in ERIs

**Original Data, Range: [0:215]**

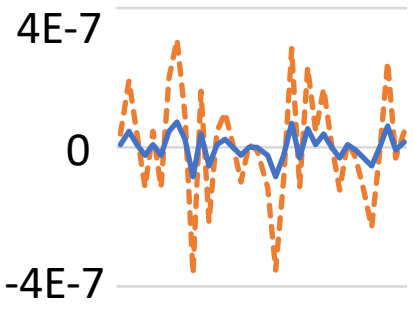



**(dd|dd) → (6*6 | 6*6) → (36 | 36) → (1296)**

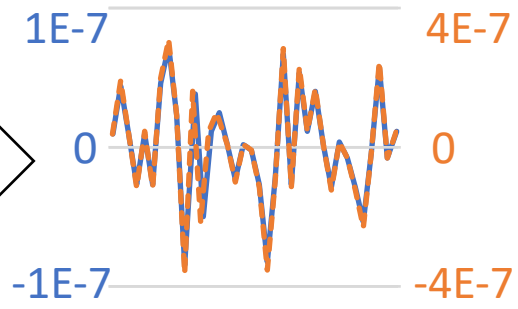
**# of SBs** **Period (SB Size)** **Block Size**

| Orbital | # of BFs |
|---|---|
| s | 1 |
| p | 3 |
| d | 6 |
| f | 10 |
| g | 15 |
| … | … |

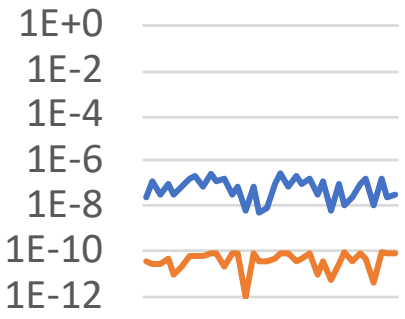
**Data Ranges**
**[0:35] , [36:71]**




**Data Ranges**
**[0:35] , [36:71]**

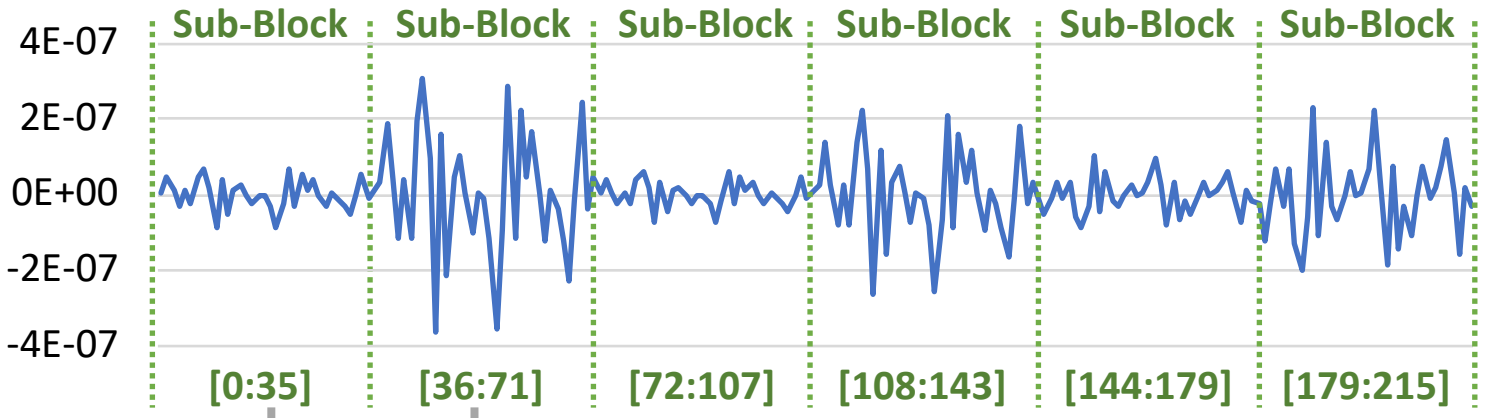



**|Deviation| and |Compr. Error|**



# Patterns in ERIs

**Original Data, Range: [0:215]**



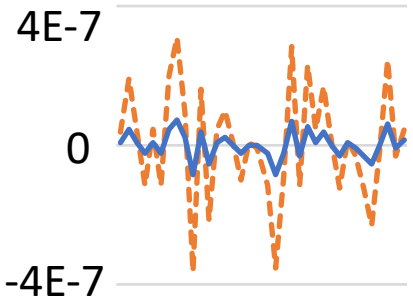

**(dd|dd) → (6*6 | 6*6) → (36 | 36) → (1296)**

**# of SBs** | **Period (SB Size)** | **Block Size**

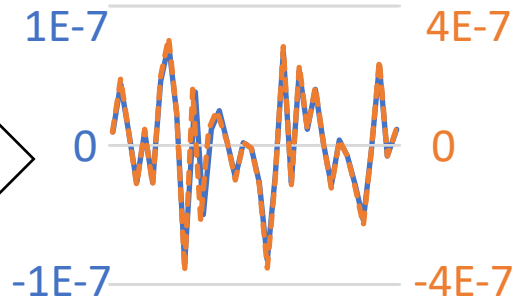
**Original Data:**
**Full Block: 1296 (64-bit)**

**Compressed:**
**Pattern: 36 (<64-bit)**
**Scale: 36 (<64-bit)**
**Error Correction: ? bits**

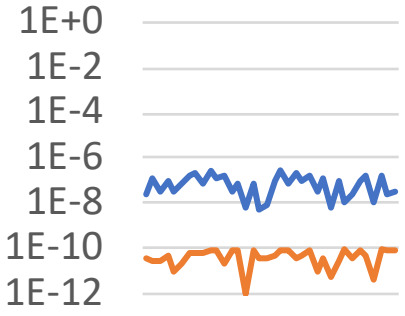
**Data Ranges**
**[0:35] , [36:71]**




**Data Ranges**
**[0:35] , [36:71]**




**|Deviation| and |Compr. Error|**

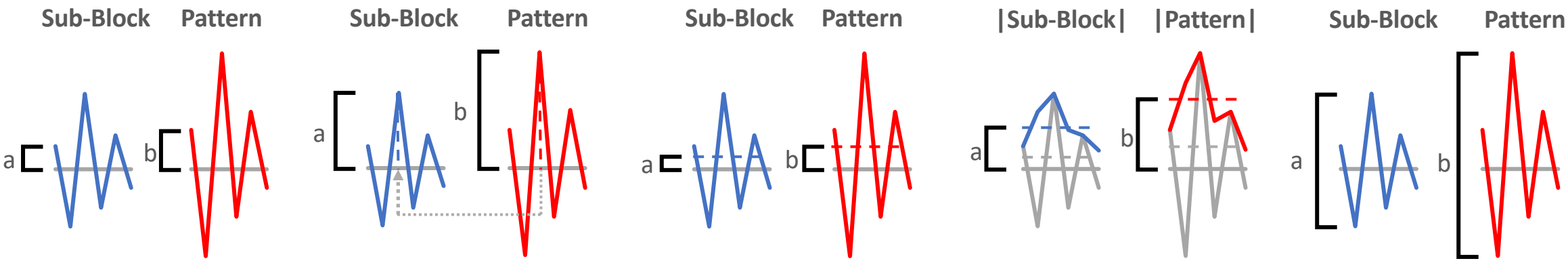


# Why are there patterns in ERIs?

$$(i,j|k,l) = \int_{3D} d\mathbf{r}_1 \int_{3D} d\mathbf{r}_2 \frac{\phi_i(\mathbf{r}_1)\phi_j(\mathbf{r}_1)\phi_k(\mathbf{r}_2)\phi_l(\mathbf{r}_2)}{|\mathbf{r}_1 - \mathbf{r}_2|},$$

- ERI values are calculated in ordered loops
- ERI values depend on both the shape and the distance of electron clouds
  - For distant clouds, the shape loses its importance, distance dominates
  - Most of the electron clouds are distant from each other

# Generating Pattern and Scaling Coefficients




FR
(Ratio of Firsts)

ER
(Ratio of Extremums)

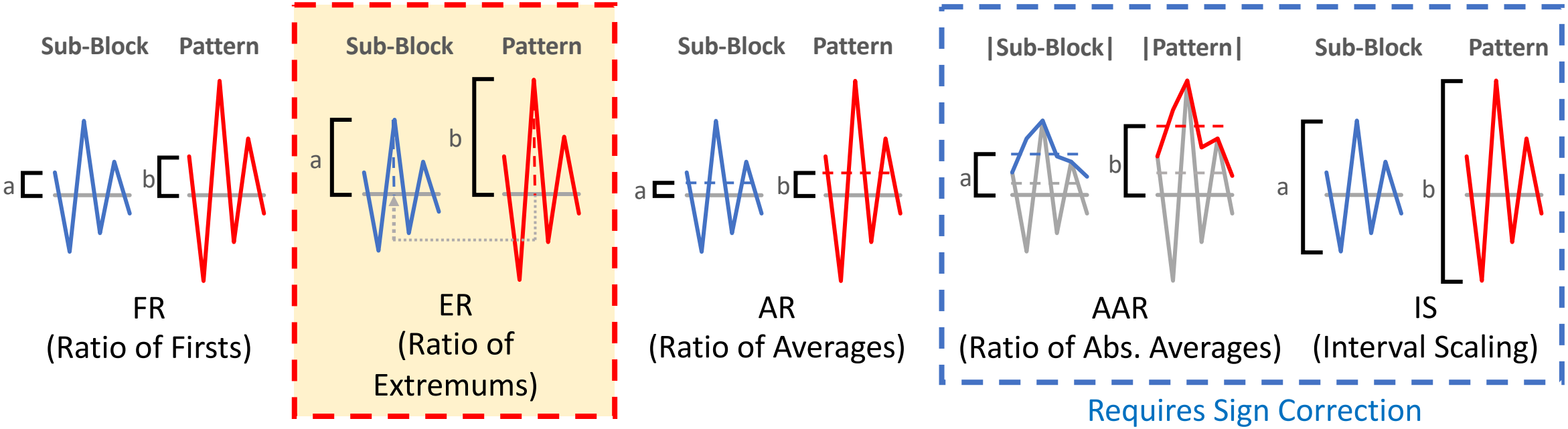
AR
(Ratio of Averages)

AAR
(Ratio of Abs. Averages)

IS
(Interval Scaling)

Scaling coefficient = a / b (Note: |b| ≥ |a|)

# Generating Pattern and Scaling Coefficients




Best Compression, Fast

Requires Sign Correction

Scaling coefficient = a / b (Note: |b| ≥ |a|)

**"a"** or **"b"** can be close to zero !

# Outline

- Introduction
- Background
  - Electron Repulsion Integrals (ERIs)
  - ERI Data Representation
- Patterns in ERIs
- PaSTRI Compression
- Optimizations of Quantization & Encoding
- Experimental Evaluation
- Conclusion

# PaSTRI Compression

- Calculate period (based on the last two BFs)
- Determine Pattern (P), then quantize P to PQ
- Calculate Scaling coefficients (S), then quantize S to SQ
  - # of elements in PQ and SQ depend on block type (s, p, d, f, g,…)
- Calculate Error Correction (EC), then quantize EC to ECQ
  - EC = Original data - PQ * P_binsize * SQ * S_binsize
  - # of elements in ECQ depends on deviation (atoms are distant or not)
- Decide encoding mode
  - Sparse or Non-sparse
- Encode PQ, SQ, and ECQ and write to output file

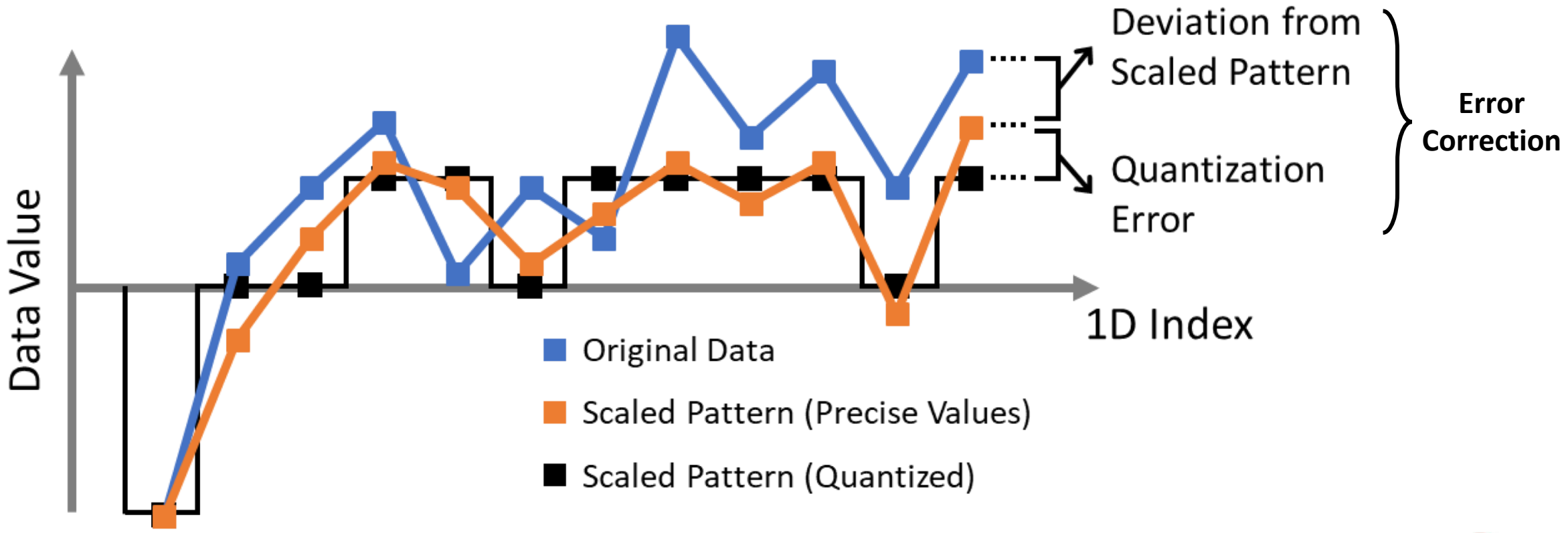
# PaSTRI Decompression

- Read encoding mode, error bound
- Calculate period
- Read PQ and reconstruct Pattern
- Read SQ and reconstruct Scaling coefficients
- Read ECQ and reconstruct Error Correction
- Reconstruct data values:
  - Decompressed Data = $\underbrace{\text{Pattern\_DQ} * \text{Scale\_DQ}}_{\text{Scaled Pattern}}$ + ErrorCorrection_DQ

# Outline

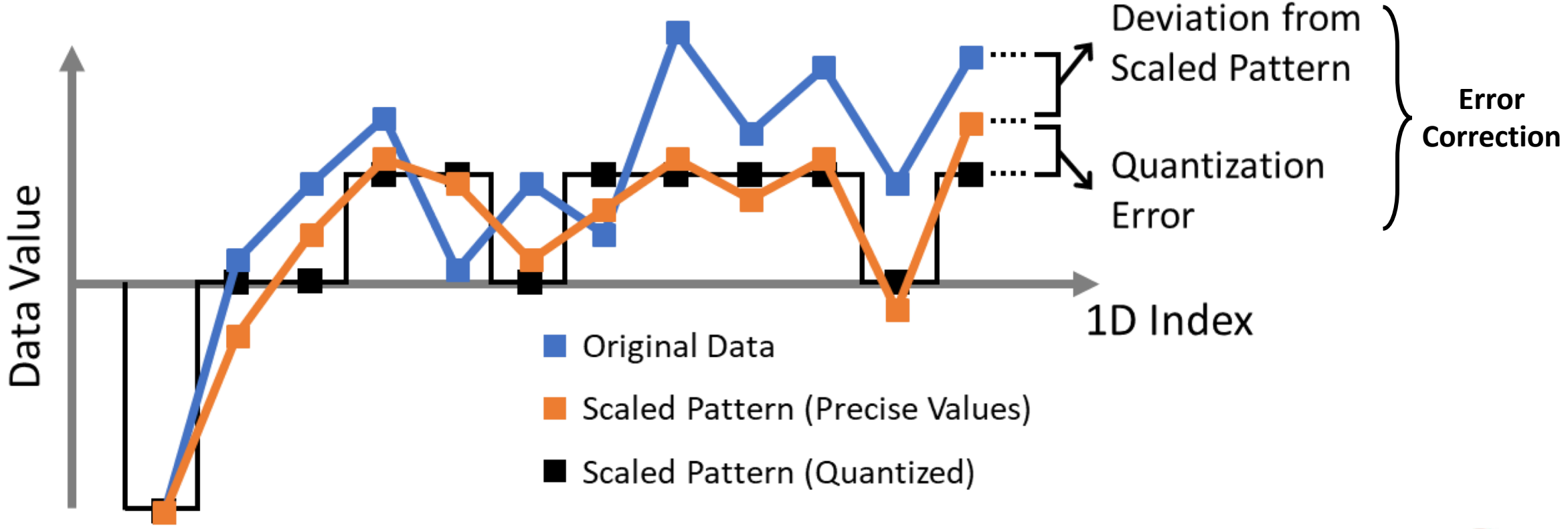
- Introduction
- Background
  - Electron Repulsion Integrals (ERIs)
  - ERI Data Representation
- Patterns in ERIs
- PaSTRI Compression
- Optimizations of Quantization & Encoding
- Experimental Evaluation
- Conclusion

# Optimizations



# Optimizations

How many bits per element for Pattern (PQ), Scaling coefficients (SQ) and Error Correction (ECQ)?



# Optimizations

- EB: $10^{-10}$
- Ranges:
  - PQ: [BlockMin, BlockMax], e.g., [$-10^{-6}$; $10^{-6}$] → 10 bits
  - SQ: [-1,1] HUGE RANGE! → 33 bits
  - ECQ: Smaller range than PQ, e.g., [$-10^{-8}$; $10^{-8}$] → 7 bits

# Optimizations

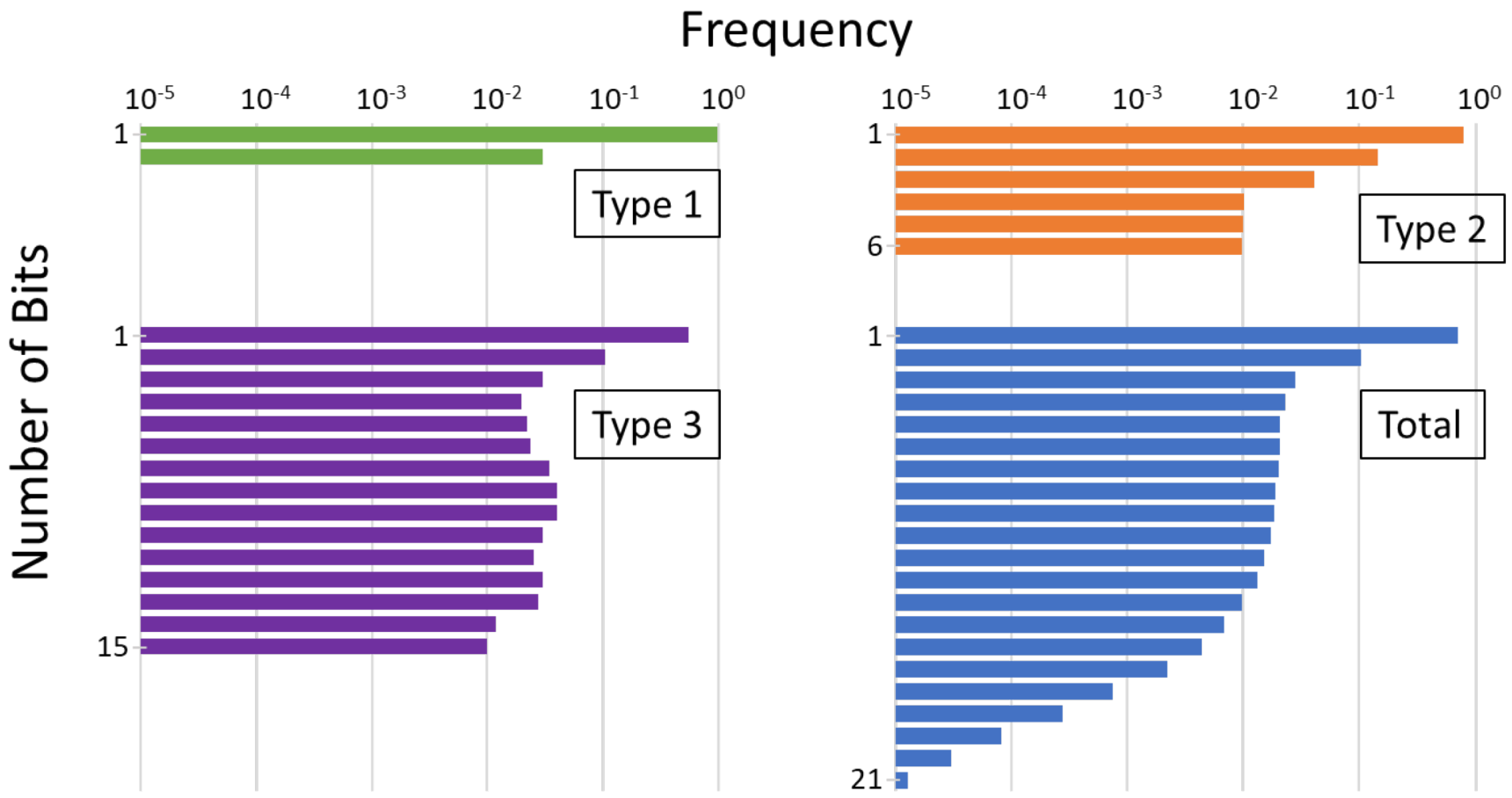
- EB: $10^{-10}$
- Ranges:
  - PQ: [BlockMin, BlockMax], e.g., [$-10^{-6}$; $10^{-6}$] → 10 bits
  - SQ: [-1,1] HUGE RANGE! → 33 bits
  - ECQ: Smaller range than PQ, e.g., [$-10^{-8}$; $10^{-8}$] → 7 bits
- Cannot enforce 2*EB quantization bin size on everyone!
- Cannot find exact optimal solution. Why?
- This is a nonlinear optimization problem (or even harder).

Too costly to solve!

# Practical solution

- PQ: 2*EB quantization bin size
- SQ: Same number of bits as PQ
- ECQ: 2*EB quantization bin size
  - Uses a special encoding tree

# ECQ Characteristics

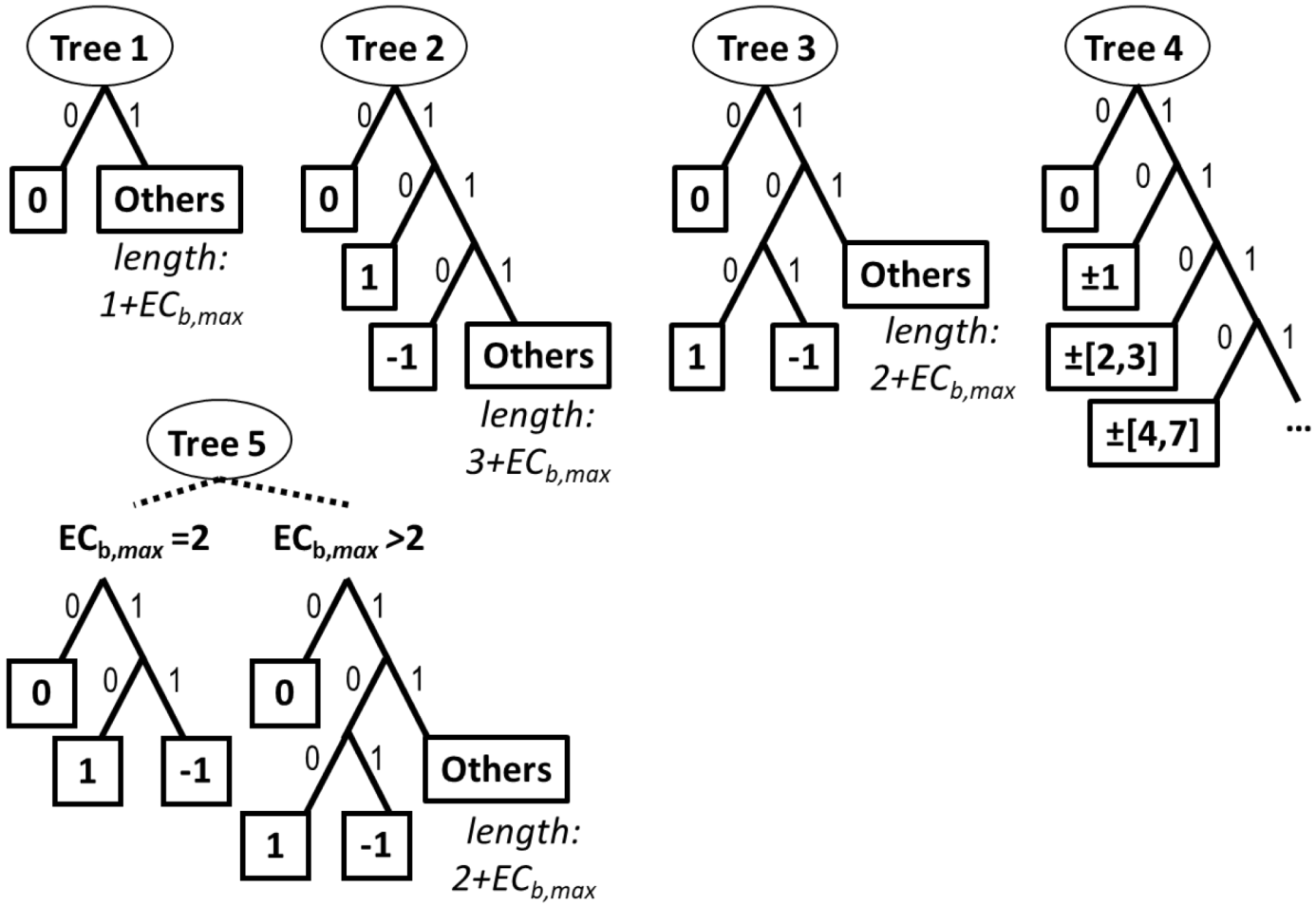



**$EC_{b,max}$ : Max bits needed for ECQ**

1 bit : 0
2 bits: 1,-1
3 bits: {2, 3, -2, -3}
4 bits: {4, 5, 6, 7, -4, -5, -6, -7}
5 bits: {8, 9, … , 14, 15, -8, -9, …, -14, -15}

⋮

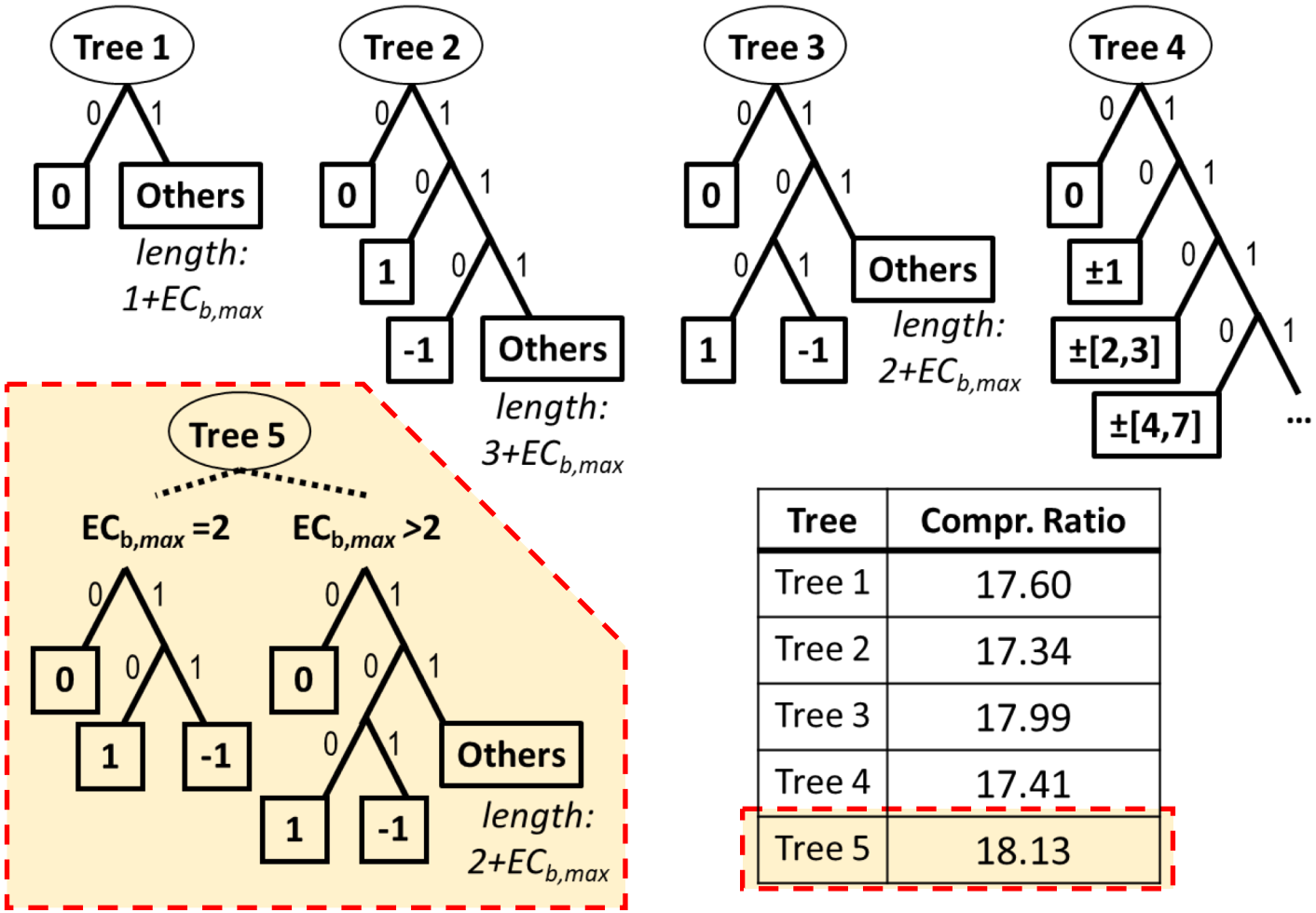
# ECQ Encoding Trees



# ECQ Encoding Trees

***Why better than Huffman?***

1) NO require generating a dictionary
2) NO require storing a dictionary
3) Single occurrences HURT Huffman compression ratio




| Tree | Compr. Ratio |
|---|---|
| Tree 1 | 17.60 |
| Tree 2 | 17.34 |
| Tree 3 | 17.99 |
| Tree 4 | 17.41 |
| Tree 5 | 18.13 |

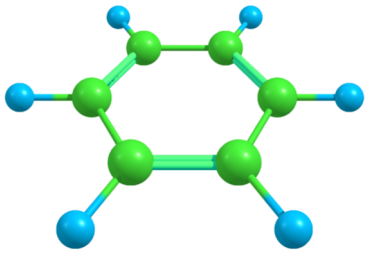
# Outline

- Introduction
- Background
  - Electron Repulsion Integrals (ERIs)
  - ERI Data Representation
- Patterns in ERIs
- PaSTRI Compression
- Optimizations of Quantization & Encoding
- Experimental Evaluation
- Conclusion

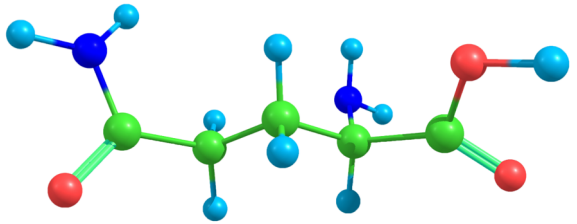
# Evaluation

- Bebop supercomputer at Argonne
  - 64 nodes (two Intel Xeon E5-2695 v4 and 128 GB memory) with 2048 cores
  - General Parallel File Systems (GPFS)
- Experimental Data
  - Tri-Alanine, Benzene, Glutamine molecules
  - (dd|dd) and (ff|ff) BF configurations
    - s and p are too small
    - g and above are not common

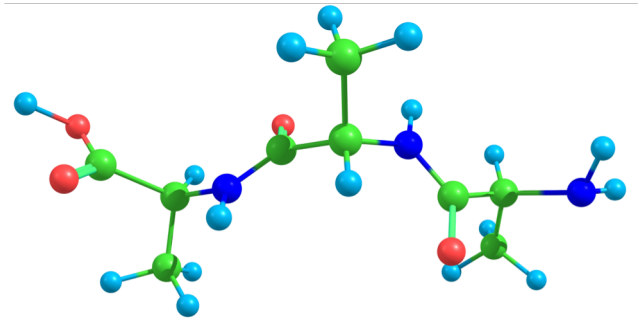
Benzene



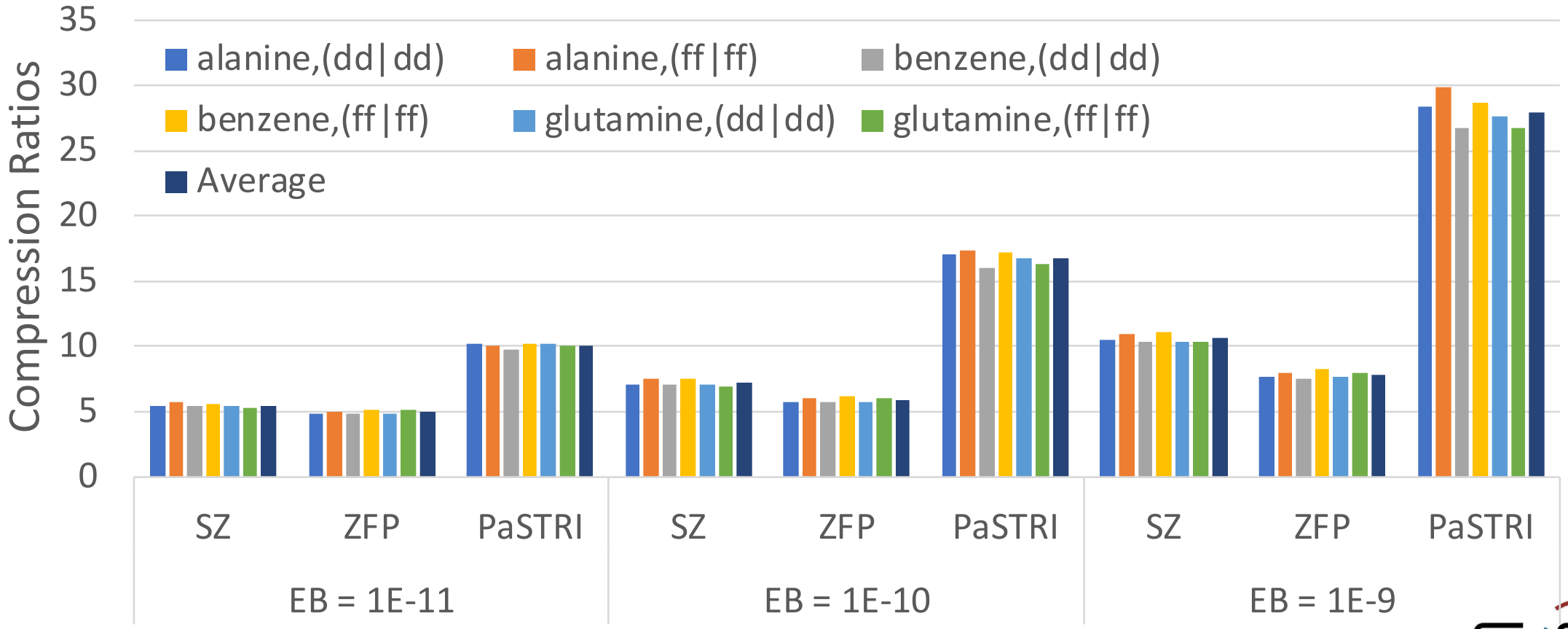
Glutamine



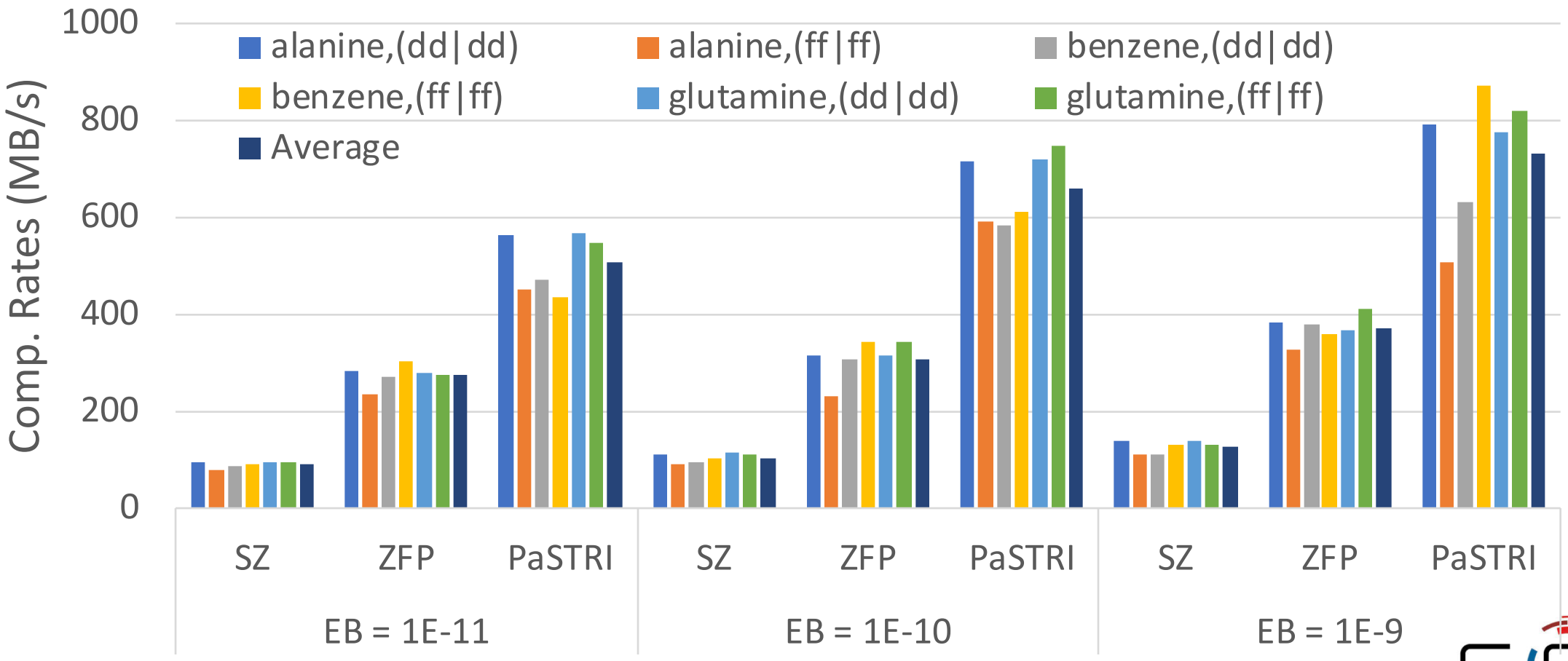
tri-Alanine



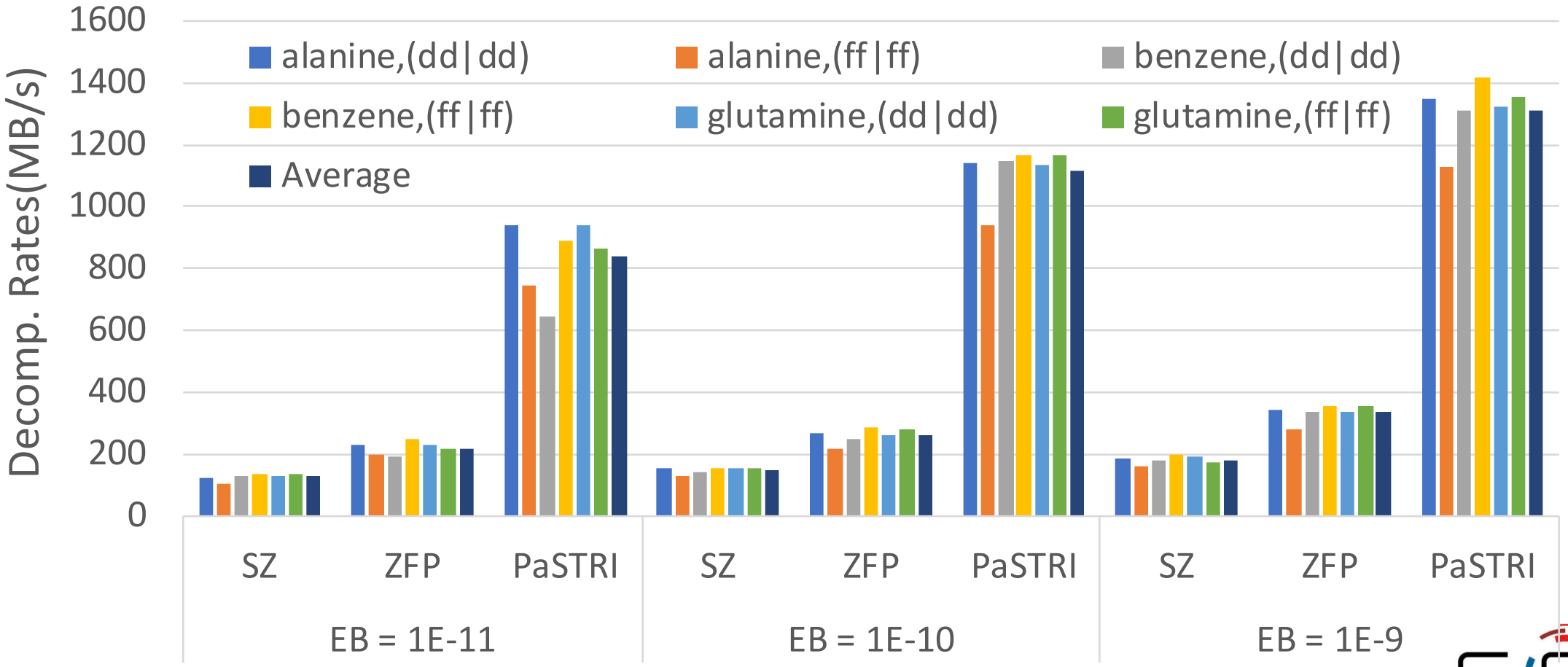
# Compression Ratios



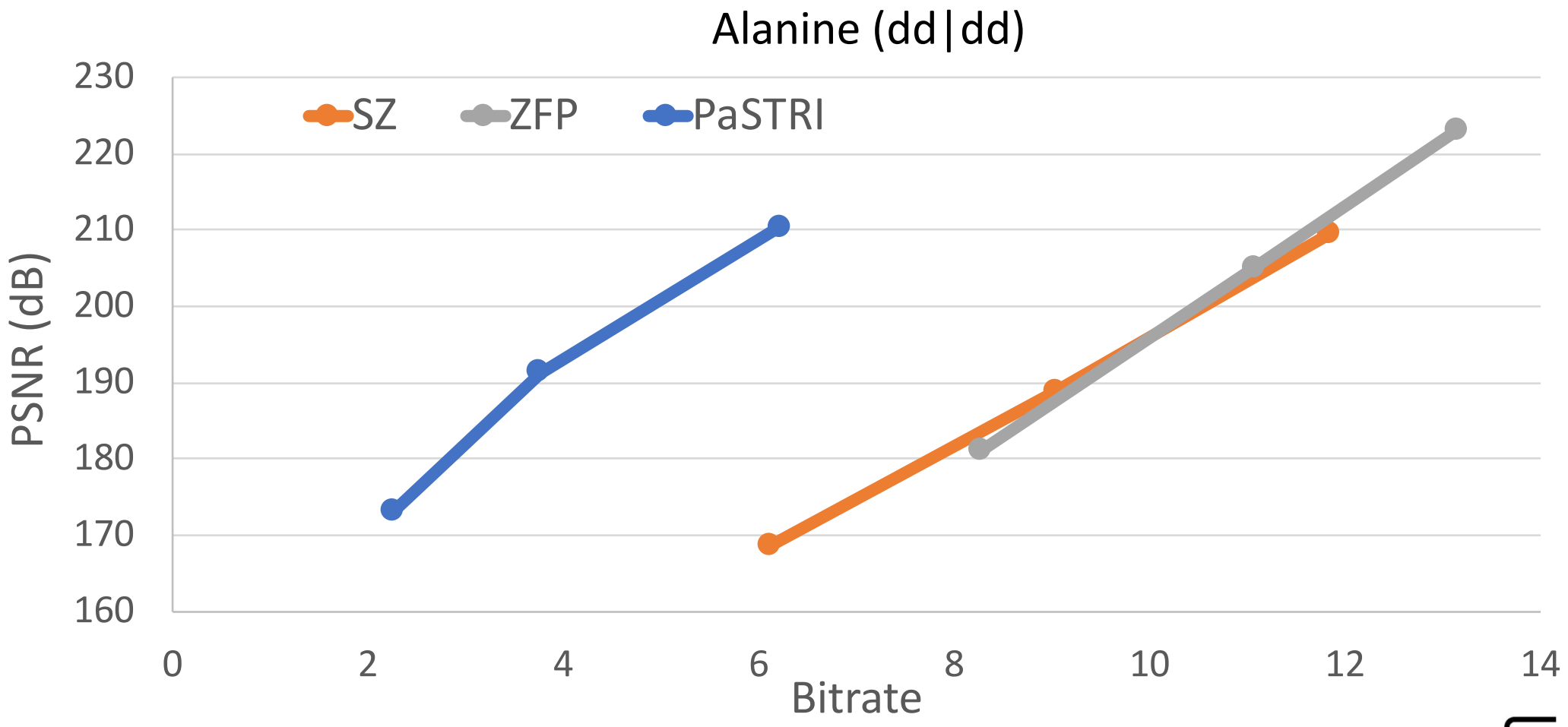
# Compression Rates



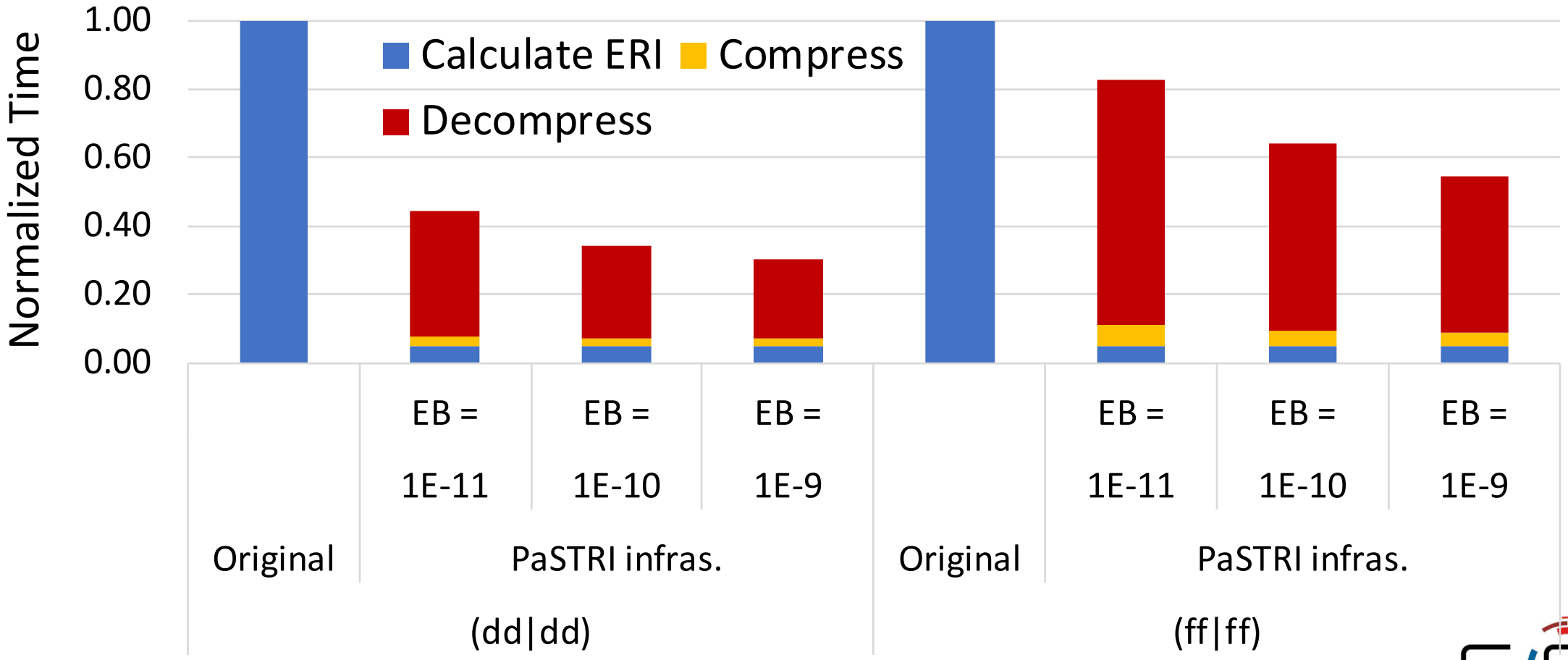
# Decompression Rates



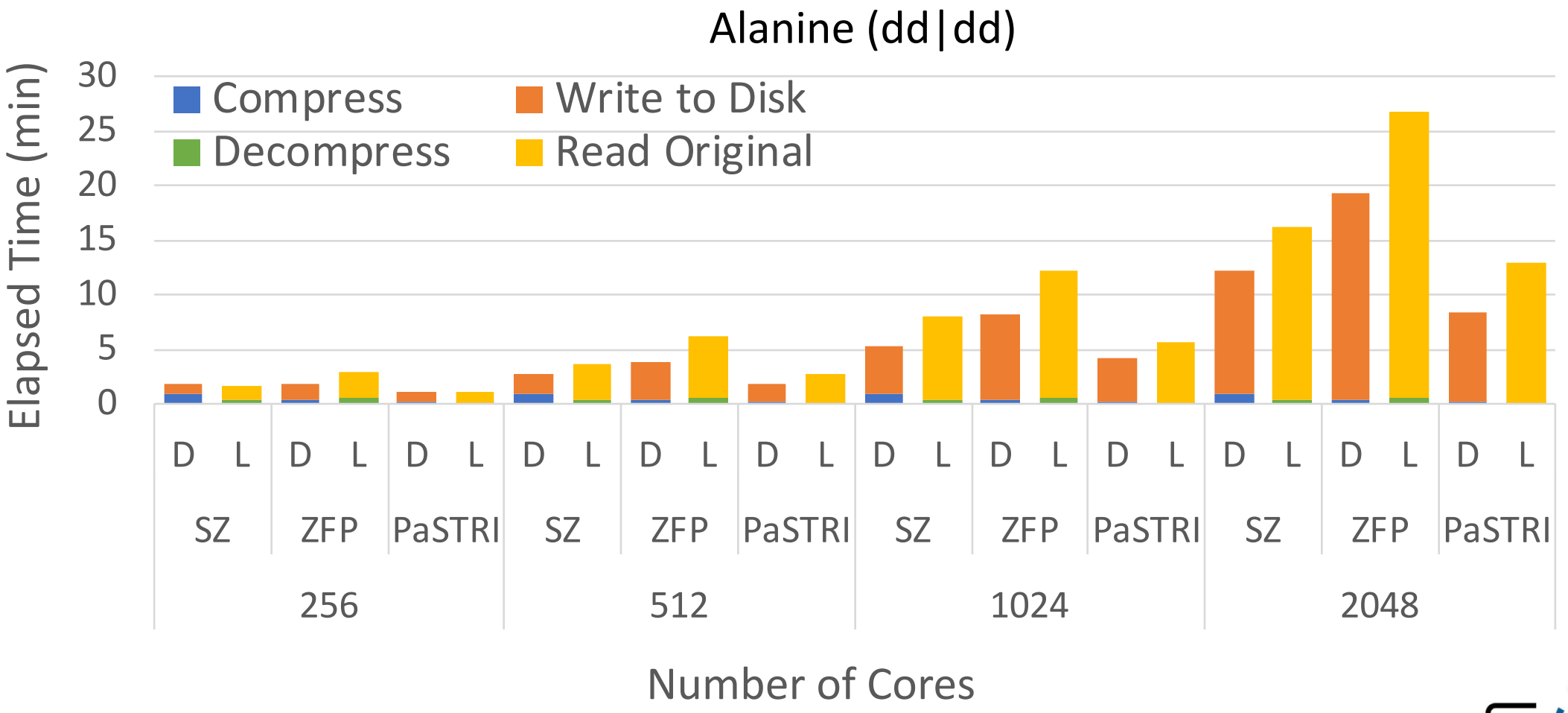
# Rate-distortion



# Baseline Comparison



# Parallel Performance (MPI)



# Conclusions

- 16.8X compression ratio, over 2.3X better
- 661 MB/s compression rate, over 2.1X better
- 1.1 GB/s decompression rate, over 4.3X better
- Significantly better rate-distortion curve
- More than 73% energy saving compared to original GAMESS infrastructure

# PaSTRI's Contributions

- Developed a model to understand ERI data
  - Data=Pattern*Scale + Deviation
- Exploited the inherent pattern features in the ERIs
- Proposed an effective lossy compression algorithm for ERIs
- PaSTRI is implemented and released as open source
- PaSTRI is evaluated by using original GAMESS data, and compared to other state-of-the-art lossy compressors

# Acknowledge

*This research was supported by the Exascale Computing Project (17-SC-20-SC), a joint project of the U.S. Department of Energy's Office of Science and National Nuclear Security Administration, responsible for delivering a capable exascale ecosystem, including software, applications, and hardware technology, to support the nation's exascale computing imperative. The material was also supported by and supported by the National Science Foundation under Grant No. 1305624, No. 1513201, and No. 1619253.*

National Science Foundation
Directorate for Computer & Information Science & Engineering (CISE)

Argonne NATIONAL LABORATORY

# PaSTRI was integrated into SZ already!

SZ is available at:
https://github.com/disheng222/SZ

Feel free to Download and Test!

Contact:

Sheng Di (sdi01@anl.gov)

Franck Cappello (cappello@mcs.anl.gov)

ECP EXASCALE COMPUTING PROJECT

# Thank you!

**Any questions are welcome!**

# Supporting Slides

| Method | Comp. Ratio |
|---|---|
| FR | N/A |
| ER | 17.46 |
| AR | 16.92 |
| AAR | 17.44 |
| IS | 17.20 |