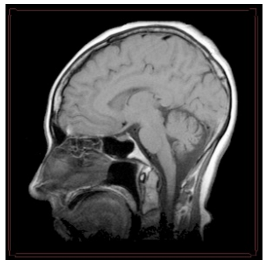
SSketch: An Automated Framework for Streaming Sketch-based Analysis of Big Data on FPGA*

Bita Darvish Rouhani, Ebrahim Songhori, Azalia Mirhoseini, and Farinaz Koushanfar
Department of ECE, Rice University
Houston Texas, USA

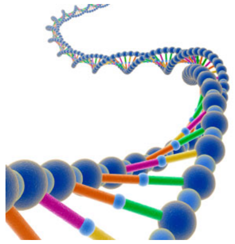
* This work was supported in parts by the Office of Naval Research grant (ONR- N00014-11-1-0885)

Introduction

- Era of big data
  - Decision making
  - Find patterns
  - Prevent failures



- Applications
  - Medical
  - Cybersecurity
  - Media/social networks
  - Finance



- Machine learning and statistical optimizations are main enablers

Efficient Data Transformation

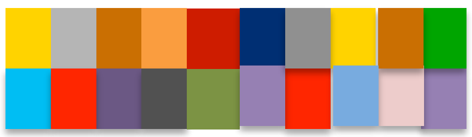
- Data transformation
  - Compact representation of the data collection
  - Exploiting the redundancy present in the dataset

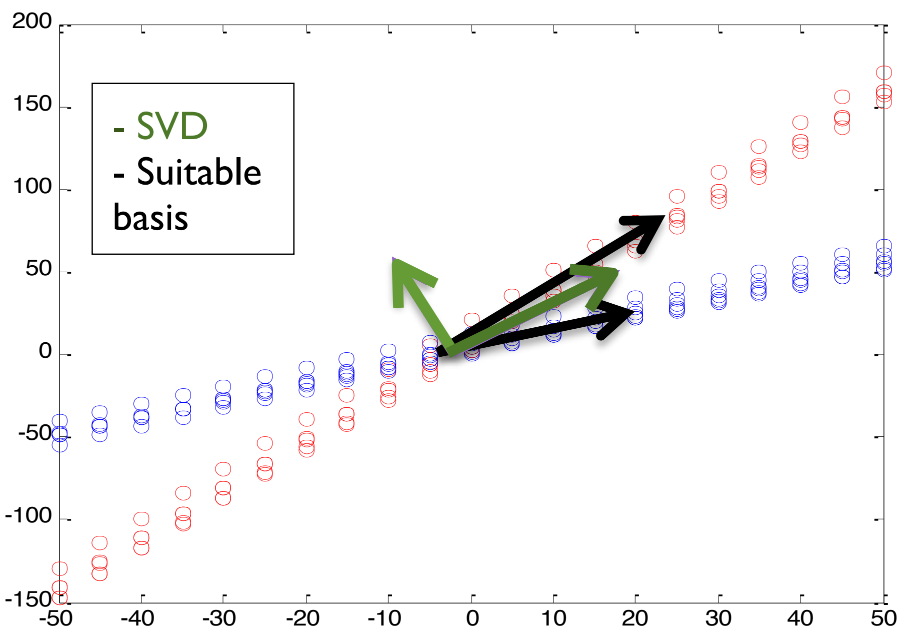
- An efficient data transformation should simultaneously consider the:
  - Scalability
  - Application
  - Underlying platform constraints

Ensemble of Lower Dimensional Structures

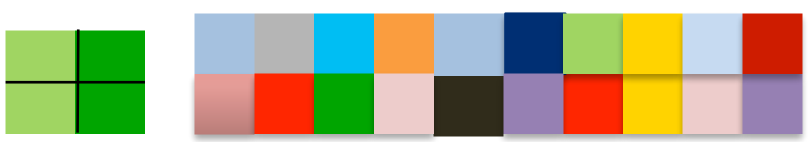
- Many modern massive datasets are either low-rank or lie on a union of lower dimensional subspaces [1]

Original dense dataset $A_{m \times n}$



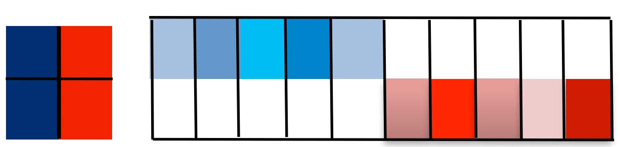


Singular value decomposition on A



$U_{m \times l}$ $V_{l \times n}$

Ensemble of lower dimensional structures



$D_{m \times l}$ $V_{l \times n}$

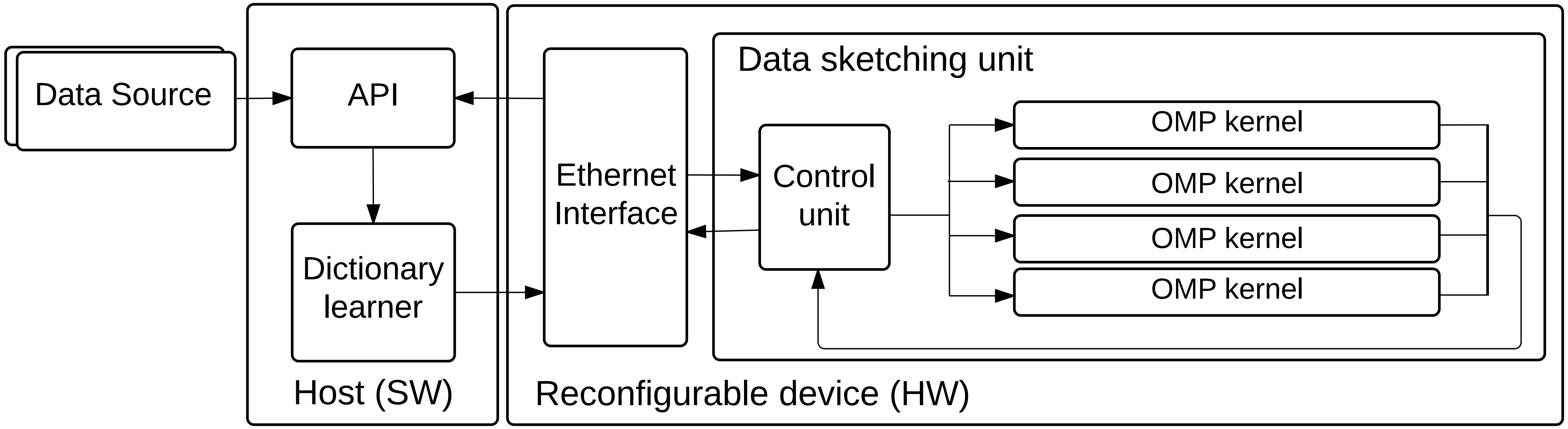
[1] Mirhoseini et. al, “Rankmap: A Platform-aware Framework for Distributed Learning from Dense Datasets”, arXiv:1503.08169, 2015

SSketch Framework

- Streaming Sketch-based analysis of big data using FPGA

- Transforming the big data with dense correlations to an ensemble of lower dimensional subspaces

- Streaming applications:
  - Limited storage
  - Single pass access to data



SSketch Framework

- The transformed data is applicable to a broad set of matrix-based data analysis algorithms:
    - Regularized loss function optimization
    - Power method
    - Image processing
        - De-noising
        - Super-resolution
        - Classification

SSketch Methodology

- Data transformation:

$$\underset{\mathbf{D}\varepsilon R^{m\times l},\mathbf{V}\varepsilon R^{l\times n}}{\text{minimize}} \quad \|\mathbf{A}-\mathbf{DV}\|_F \quad \text{subject to} \quad \|\mathbf{V}\|_0 \leq kn$$

- Adaptive error-based dictionary learning
- Single pass access to data

Algorithm 1 SSketch algorithm

Inputs: Measurement matrix A, projection threshold α, sparsity level k, error threshold ϵ, and dictionary size l.

Output: Matrix D, and coefficient matrix V.

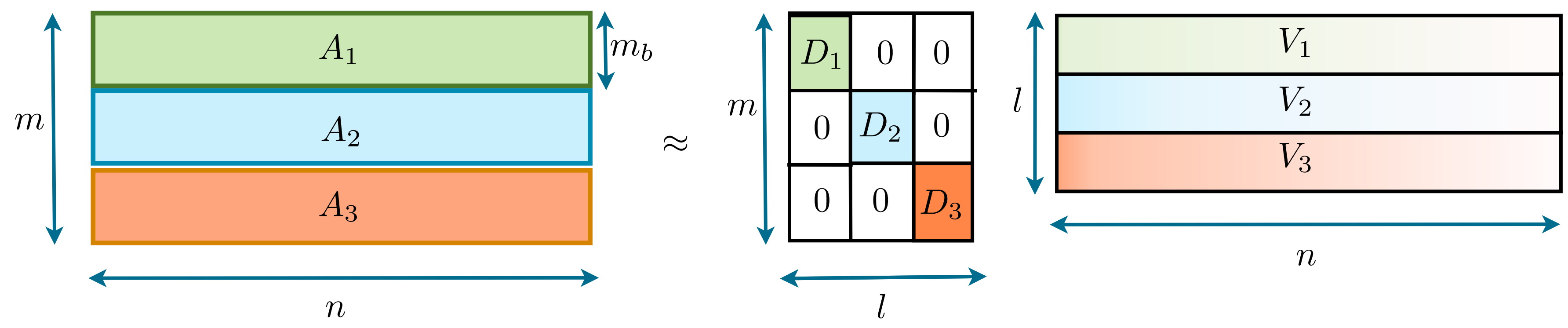
$D \leftarrow$ **empty**
$j \leftarrow 0$
for i = 1,...,n **do**
  $W(\mathbf{A}_i) = \frac{\|\mathbf{D}(\mathbf{D}^t\mathbf{D})^{-1}\mathbf{D}^t\mathbf{A}_i-\mathbf{A}_i\|_2}{\|\mathbf{A}_i\|_2}$
  if $W(\mathbf{A}_i) > \alpha$ **and** $j < l$ **then**
    $\mathbf{D}_j = \mathbf{A}_i/\sqrt{\|\mathbf{A}_i\|_2}$
    $\mathbf{V}_{ij} = \sqrt{\|\mathbf{A}_i\|_2}$
    $j \leftarrow j+1$
  else
    $\mathbf{V}_i \leftarrow OMP(\mathbf{D},\mathbf{A}_i,k,\epsilon)$
  end if
end for

SSketch Methodology

- Data transformation:

$$\underset{\mathbf{D}\varepsilon R^{m\times l},\mathbf{V}\varepsilon R^{l\times n}}{\text{minimize}} \quad \|\mathbf{A}-\mathbf{DV}\|_F \ \text{ subject to } \ \|\mathbf{V}\|_0 \leq kn$$

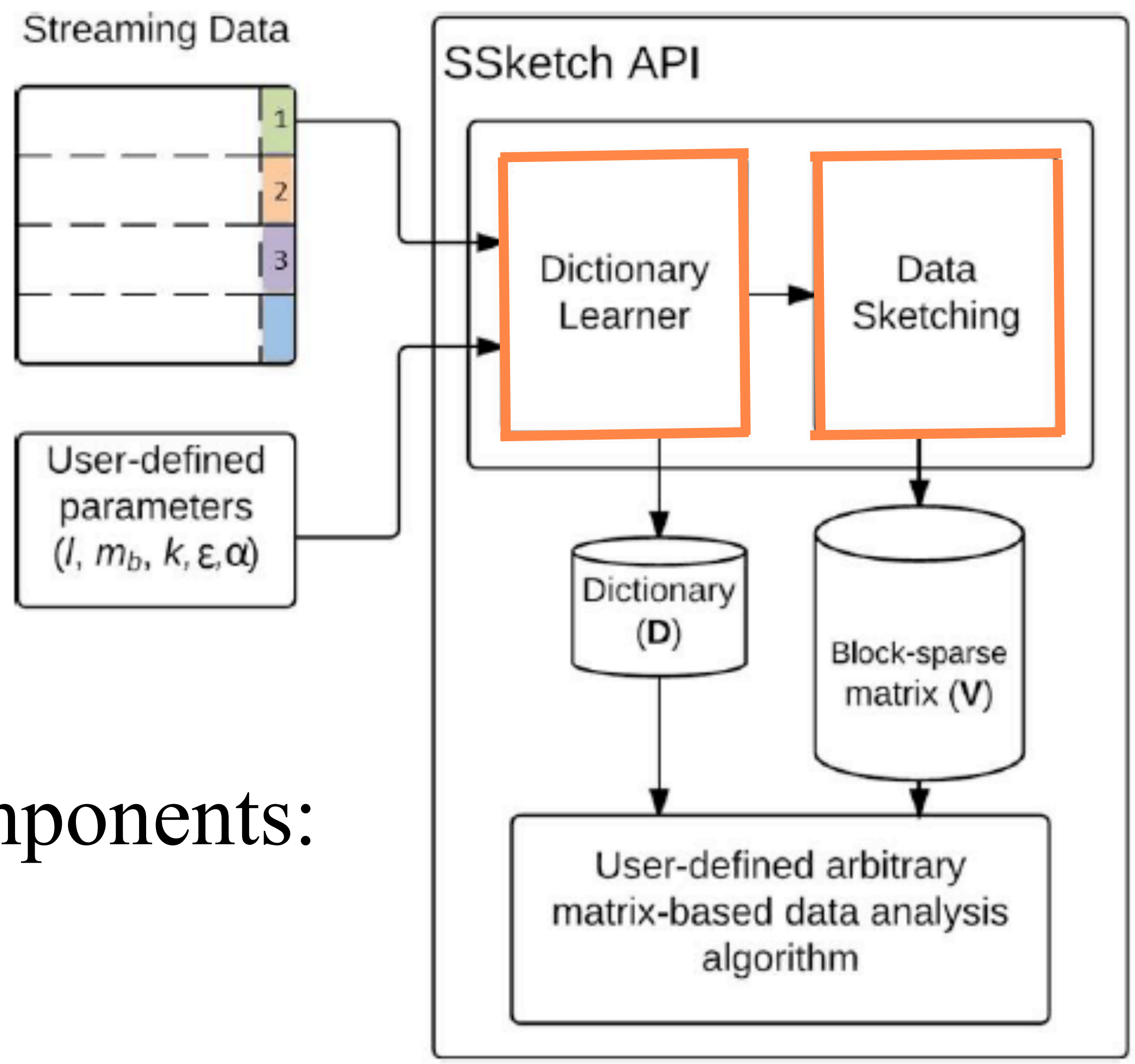
- Block splitting
- Amenable to FPGA accelerator



Schematic depiction of blocking SSketch

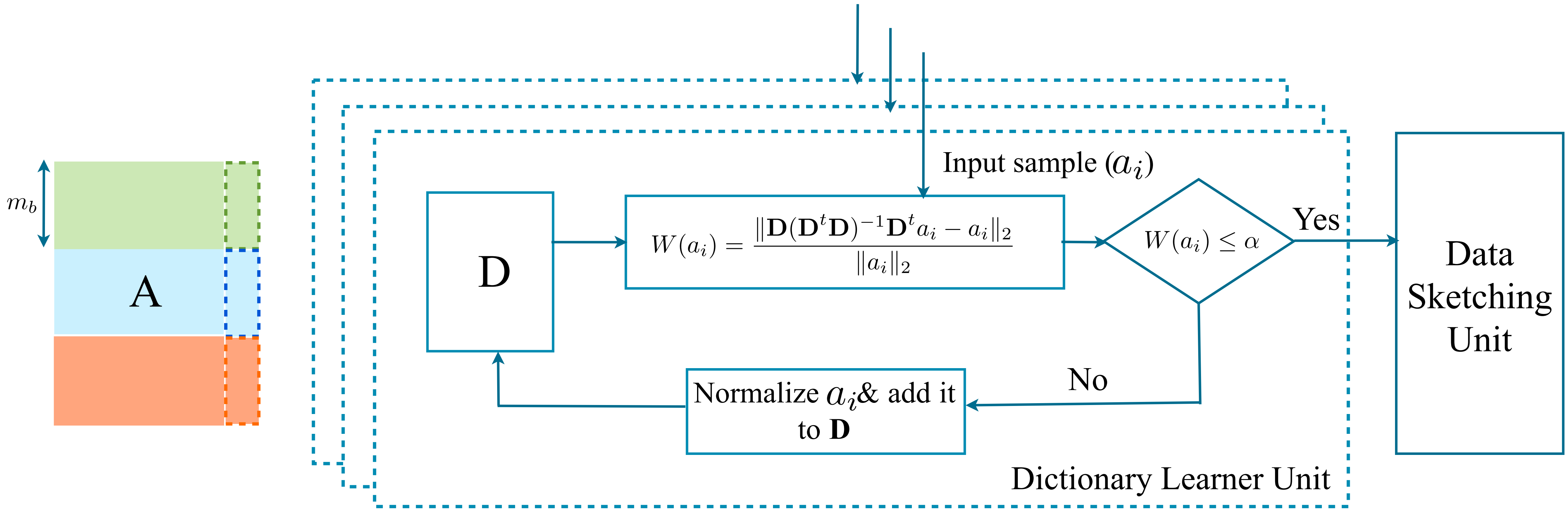
SSketch API

- Inputs:
  - Stream of input data
  - SSketch Algorithmic parameters
- Outputs:
  - Dictionary matrix D
  - Block-sparse matrix V
- SSketch consists of two main components:
  - Dictionary learner unit
  - Data sketching unit



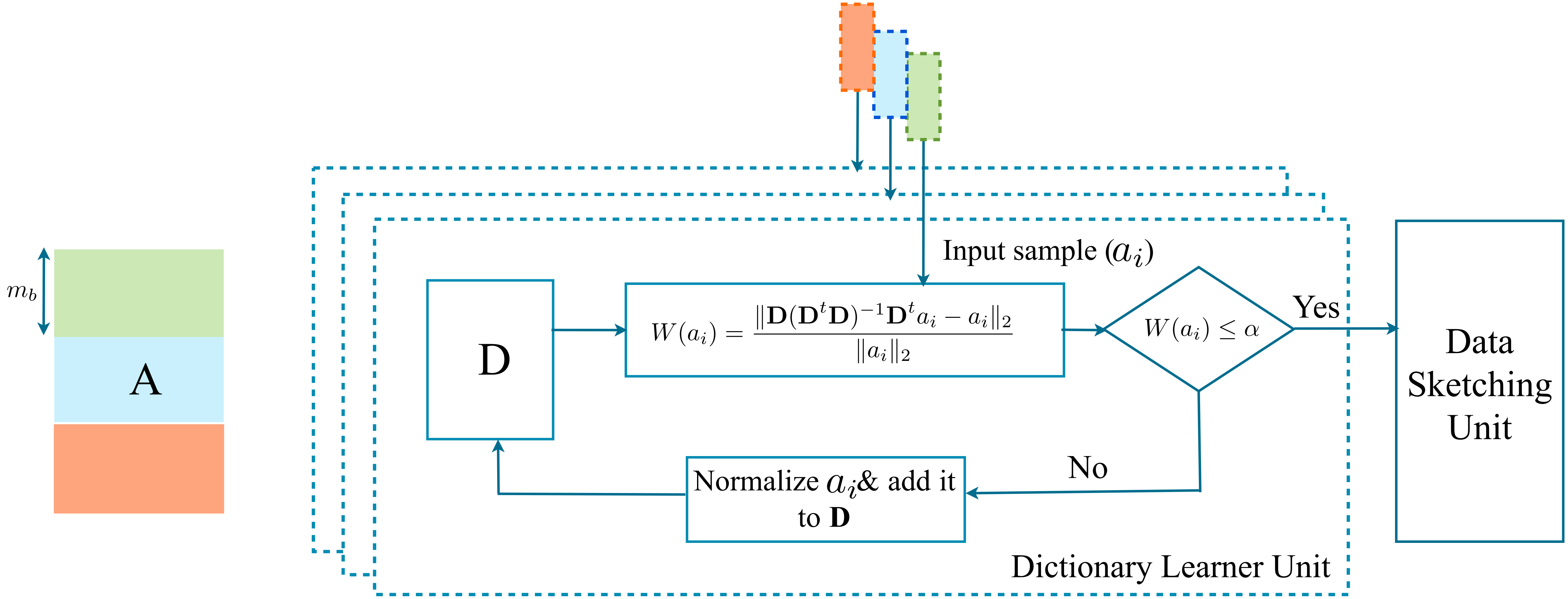
Adaptive Dictionary Learning

- By Streaming the input data, SSketch adaptively:
  - Learns/updates the corresponding dictionary of each block
  - Computes the data sketch



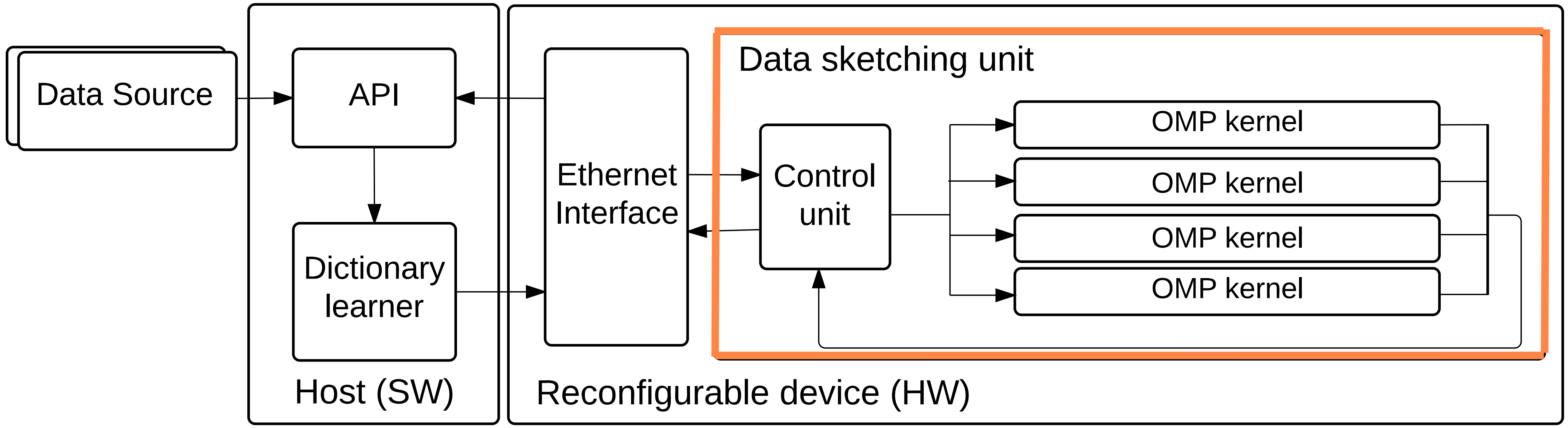
Adaptive Dictionary Learning

- By Streaming the input data, SSketch adaptively:
  - Learns/updates the corresponding dictionary of each block
  - Computes the data sketch



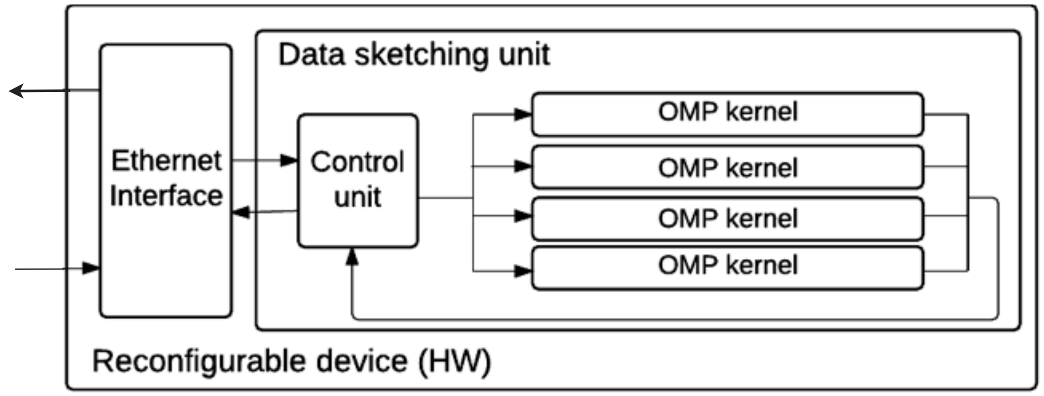
Data Sketching

- Computing the block-sparse matrix $\boldsymbol{V}$
  - Applying the efficient, and greedy Orthogonal Matching Pursuit (OMP) routine
- Asynchronous parallel approach via a control unit



Hardware Implementation

- Xilinx Virtex-6 FPGA ML605 Evaluation
- IEEE 754 single precision floating point format
- Resource utilization



Virtex-6 resource utilization

	Used	Available	Utilization
Slice Registers	50888	301440	16%
Slice LUTs	81585	150720	54%
RAM B36E1	382	416	91%
DSP 48E1s	356	768	46%

OMP Routine

- OMP is mainly consists of three steps:
  - Find best fitting column
  - LS optimization
  - Residual update
- Use QR decomposition to address the LS optimization

Algorithm 2 OMP algorithm

Inputs: Matrix D, measurement $\mathbf{A}_i$, sparsity level k, threshold error ϵ.
Output: Support set Λ and k-dimensional coefficient vector v.

```
r ← A_i
Λ^0 ← ∅
for i = 1,...,k do
    Λ ← Λ ∪ argmax_j| < r^{i-1}, D_j > |  Find best fitting column
    v^i ← argmin_v ||r^{i-1} − D_{Λ^i} v||_2^2  LS Optimization
    r^i ← r^{i-1} − D_{Λ^i} v^i  Residual Update
end for
```

$r \leftarrow \mathbf{A}_i$

$\Lambda^0 \leftarrow \emptyset$

for i = $1,...,k$ **do**

  $\Lambda \leftarrow \Lambda \cup argmax_j | < r^{i-1}, \mathbf{D}_j > |$ **Find best fitting column**

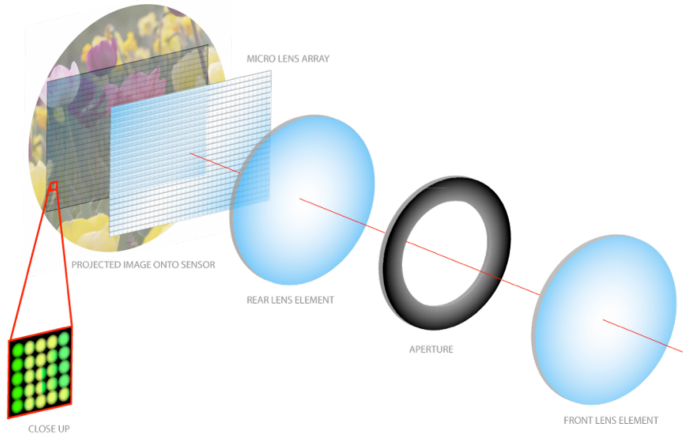
  $v^i \leftarrow argmin_v \|r^{i-1} - \mathbf{D}_{\Lambda^i} v\|_2{}^2$ **LS Optimization**

  $r^i \leftarrow r^{i-1} - \mathbf{D}_{\Lambda^i} v^i$ **Residual Update**

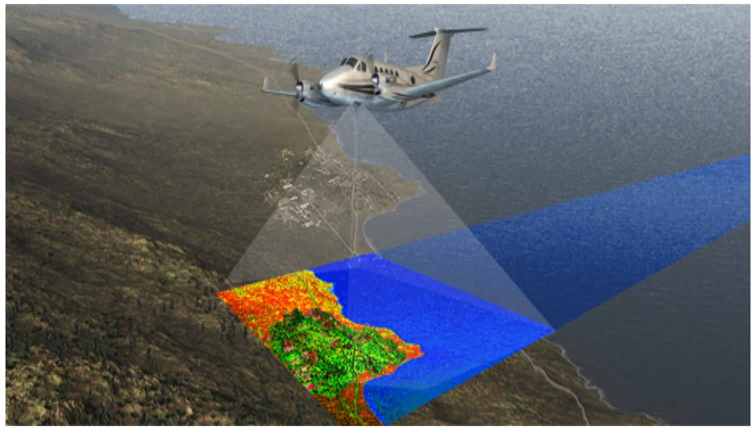
end for

Benchmark Datasets

- Three different datasets:
  - Light Field imaging
    - A sequence of multi-dimensional array of images that are simultaneously captured from slightly different viewpoints



  - Hyper-Spectral imaging
    - A sequence of images generated by hundreds of detectors that capture the information from across the electromagnetic spectrum
  - Synthetic data



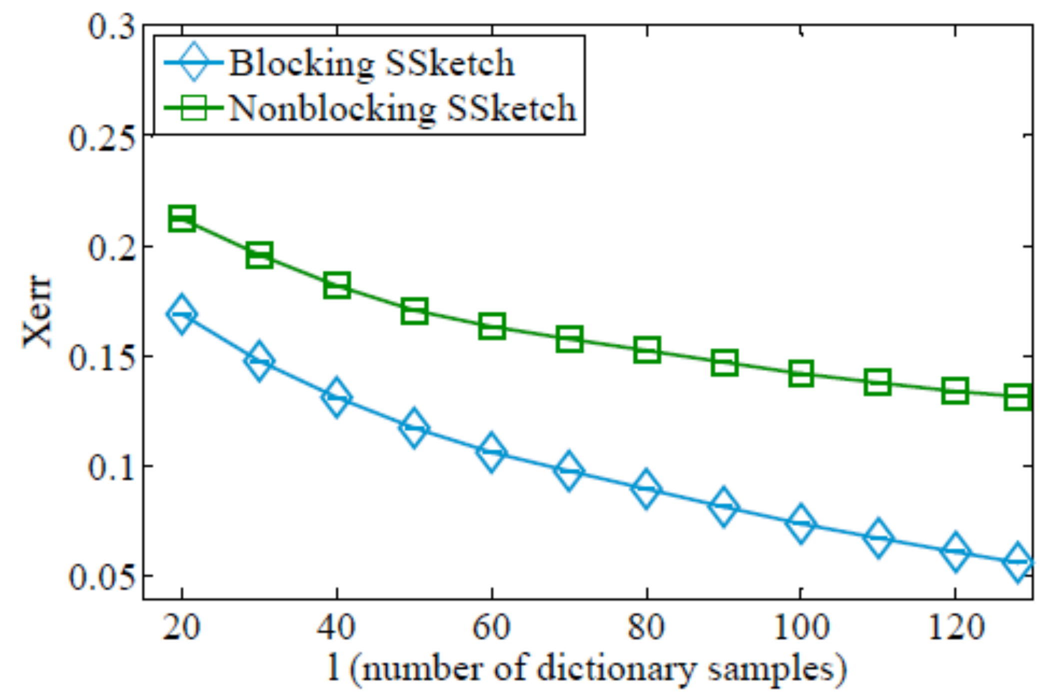
Evaluation Results

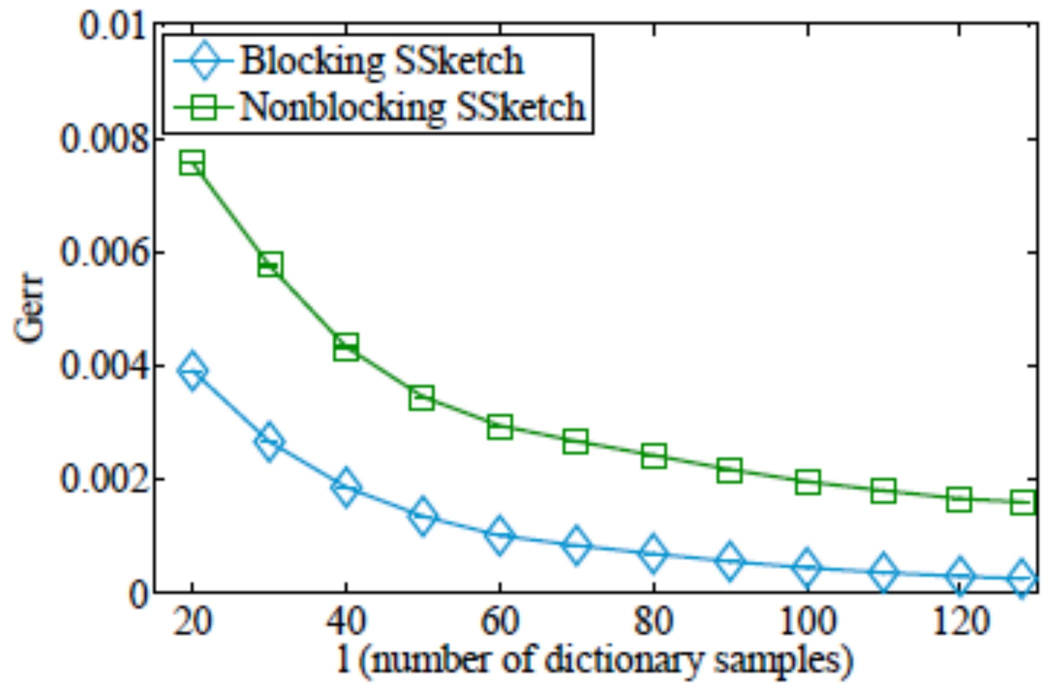
$$Xerr = \frac{\|\mathbf{A} - \tilde{\mathbf{A}}\|_F}{\|\mathbf{A}\|_F}, \quad \tilde{\mathbf{A}} = \mathbf{DV}$$

$$Gerr = \frac{\|\mathbf{A}^t\mathbf{A} - \tilde{\mathbf{A}}^T\tilde{\mathbf{A}}\|_F}{\|\mathbf{A}^t\mathbf{A}\|_F}$$

$$Compression\_rate = \frac{nnz(\mathbf{D}) + nnz(\mathbf{V})}{nnz(\mathbf{A})}$$

- Data sketching error decreases as the dictionary size increases



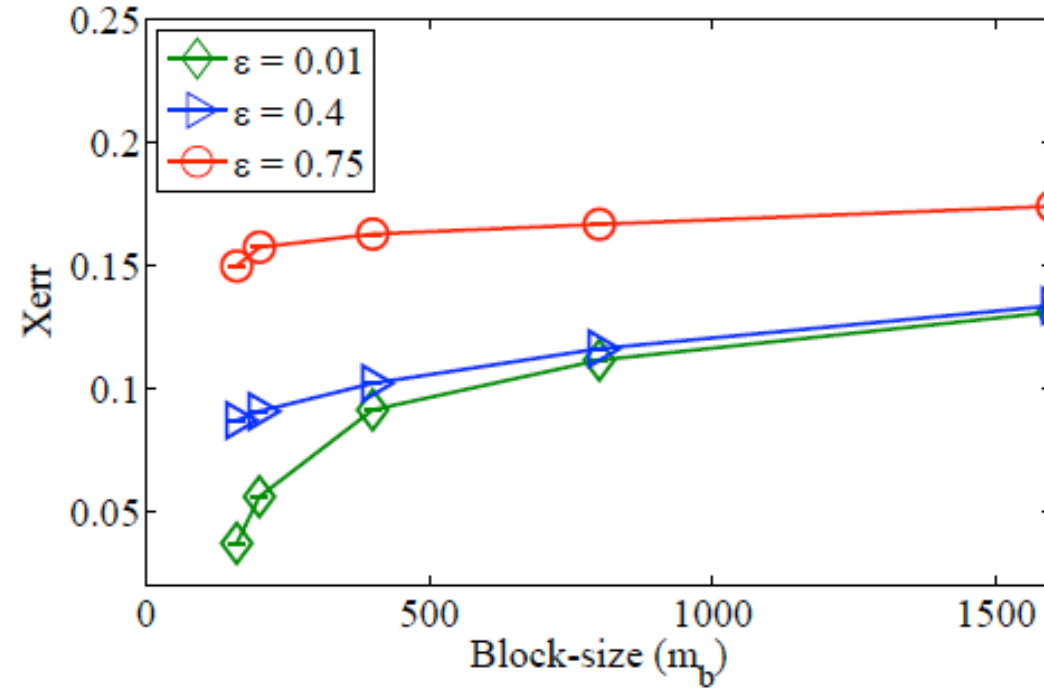


Evaluation Results

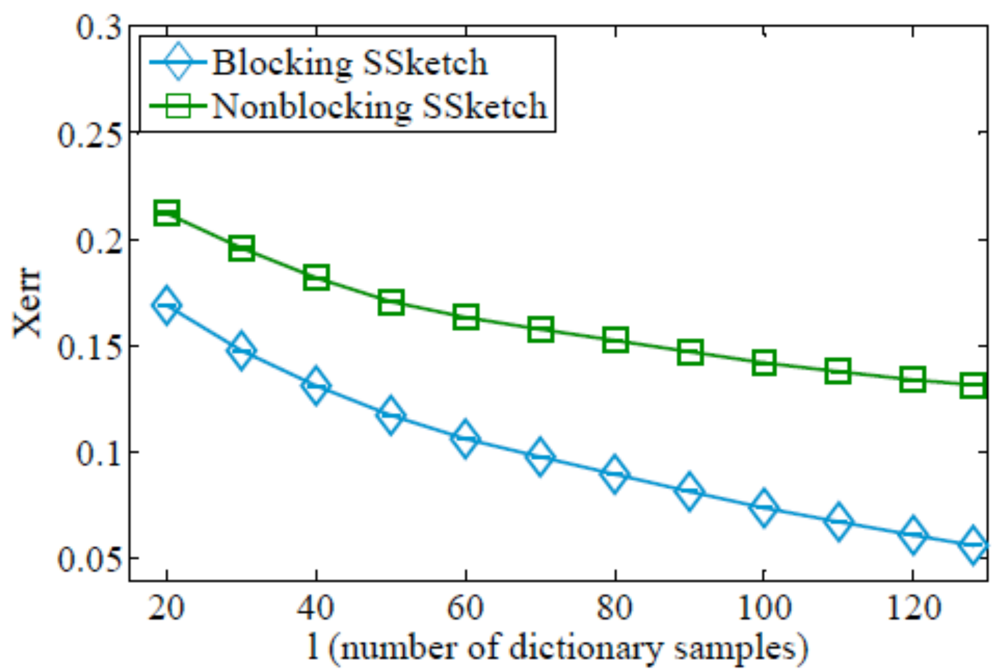
$$Xerr = \frac{\|\mathbf{A} - \tilde{\mathbf{A}}\|_F}{\|\mathbf{A}\|_F}, \quad \tilde{\mathbf{A}} = \mathbf{DV}$$

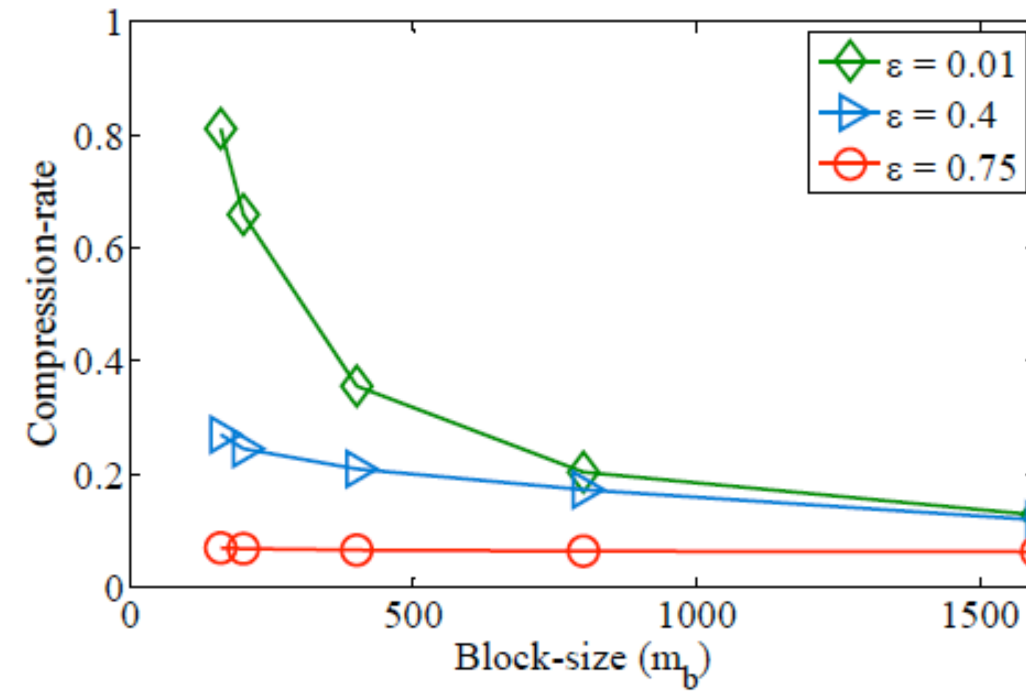
$$Gerr = \frac{\|\mathbf{A}^t\mathbf{A} - \tilde{\mathbf{A}}^T\tilde{\mathbf{A}}\|_F}{\|\mathbf{A}^t\mathbf{A}\|_F}$$

$$Compression\_rate = \frac{nnz(\mathbf{D}) + nnz(\mathbf{V})}{nnz(\mathbf{A})}$$



- There is a trade-off between the sketch accuracy and the number of non-zeros in the block-sparse matrix ***V***





Evaluation Results

- SSketch total processing time is linear in terms of the number of processed samples
- Runtime: $T_{SSketch} \approx T_{\substack{dictionary \\ learning}} + T_{\substack{Communication \\ Overhead}} + T_{\substack{FPGA \\ Computation}}$
    - SSketch runtime is dominated by $T_{\substack{FPGA \\ Computation}}$

$Size\ of\ n$	$T_{SSketch}$ $(l = 128)$	$T_{SSketch}$ $(l = 64)$
$1k$	3.635s	2.31s
$5k$	21.029s	12.01s
$10k$	43.446s	24.32s
$20k$	90.761s	48.52s

$$m_b = 256, \epsilon = 0.01, and\ \alpha = 0.1$$

Conclusion

- Adaptive hardware-accelerated streaming-based data transformation
- Scalable streaming-based sketching methodology
  - Amenable to FPGA acceleration
  - Fixed, and low memory footprint
- User-friendly API
  - Rapid prototyping of an arbitrary matrix-based data analysis
- Scalable, floating-point implementation of OMP on FPGA
- Up to 200 folds speed up compared to the software-only realization
- Less than 4% message passing delay for communication between the processor and accelerator

Acknowledgment




- From left to right:

Bita Darvish Rouhani, PhD. student, Rice University (bita@rice.edu)
Ebrahim Songhori, PhD. student, Rice University (ebrahim@rice.edu)
Azalia Mirhoseini, PhD. student, Rice University (azalia@rice.edu)
Farinaz Koushanfar, Associate Professor, Rice University (farinaz@rice.edu)

- This work was supported in parts by the Office of Naval Research grant (ONR- N00014-11-1-0885)