Data-Driven Sampling Matrix Boolean Optimization for Energy-Efficient Biomedical Signal Acquisition by Compressive Sensing

Yuhao Wang, Xin Li, *Senior Member, IEEE*, Kai Xu, Fengbo Ren, *Member, IEEE*, and Hao Yu, *Senior Member, IEEE*

***Abstract*—Compressive sensing is widely used in biomedical applications, and the sampling matrix plays a critical role on both quality and power consumption of signal acquisition. It projects a high-dimensional vector of data into a low-dimensional subspace by matrix-vector multiplication. An optimal sampling matrix can ensure accurate data reconstruction and/or high compression ratio. Most existing optimization methods can only produce real-valued embedding matrices that result in large energy consumption during data acquisition. In this paper, we propose an efficient method that finds an optimal Boolean sampling matrix in order to reduce the energy consumption. Compared to random Boolean embedding, our data-driven Boolean sampling matrix can improve the image recovery quality by 9 dB. Moreover, in terms of sampling hardware complexity, it reduces the energy consumption by 4.6× and the silicon area by 1.9× over the data-driven real-valued embedding.**

***Index Terms*—Compressive sensing, low power sensor, quantization, resistive random-access memory (RRAM), sampling matrix optimization.**

I. Introduction

BIOMEDICAL wireless circuits for applications such as health telemonitoring [1], [2] and implantable biosensors [3], [4] are energy sensitive. To prolong the life-time of their services, it is essential to perform the dimension reduction while acquiring original data. The compressive sensing [5] is a signal processing technique that exploits signal sparsity so that signal can be reconstructed under lower sampling rate than that of Nyquist sampling theorem. The existing works that apply compressive sensing technique on biomedical hardware focus on the efficient signal reconstruction by either dictionary learning [4], [6] or more efficient algorithms of finding the sparsest coefficients [1], [2], [7]–[9]. However, these works, by improving the reconstruction on mobile/server nodes instead of data acquisition on sensor nodes, can only indirectly reduce the number of samples for wireless transmission with lower energy. In this work, we aim to achieve both high performance signal acquisition and low sampling hardware cost at sensor nodes directly.

In compressive sensing, the sampling is performed by multiplying the original signal vector with a linear embedding matrix, which projects the high-dimensional data vector into a low-dimensional subspace with preserved intrinsic information. The concise representation is called a low-dimensional embedding. The sampling matrix can be either random, generally optimized or optimized towards specific dataset. The random sampling matrices, Bernoulli or Gaussian, though easier to construct, have two major limitations. Firstly, the guarantee on signal recoverability using random sampling matrices is only probabilistic and therefore large recovery error may be incurred. Secondly, its construction is independent on the data under investigation, and therefore the geometric information of dataset cannot be exploited. The generally optimized sampling matrices, such as Reed-Muller code [10], [11] or Puffer transformation [4], have deterministic recoverability by constructing matrices with minimized mutual coherence. However, they are still independent on data type of interest so that the performance cannot be maximized. The data-driven optimized embedding, on the other hand, can leverage geometric structure of dataset in particular application with additional learning phase, which is especially beneficial for biomedical sampling hardware where the target data type is predetermined. Signal acquisition by a data-driven optimized sampling matrix can preserve more intrinsic information of original signal, and therefore ensure more accurate signal reconstruction and/or higher compression ratio.

Most existing data-driven optimization methods only produce real-valued embedding matrices [12]. However, the hardware mapping of real-valued sampling matrix is much more power consuming than that of Boolean matrix. The reason is that for real-valued embedding operation, the required hardware resources, primarily full-adders, are quadratically depending on the precision required, while only linearly for Boolean embedding mapping. Therefore, a Boolean sampling matrix is preferred as the significantly reduced hardware resources can contribute to higher energy- and area-efficiency. In fact, the random Bernoulli matrix is the most widely used sampling matrix in existing CMOS based implementations for low power

Manuscript received February 6, 2016; revised July 5, 2016; accepted July 14, 2016. Date of publication November 14, 2016; date of current version March 22, 2017. This work is sponsored by grants from Singapore MOE Tier-2 (MOE2015-T2-2-013), NRF-ENIC-SERTD-SMES-NTUJTCI3 C-2016 (WP4) and NRF-ENIC-SERTD-SMES-NTUJTCI3 C-2016 (WP5). This paper was recommended by Associate Editor A. Bermak.

Y. Wang and H. Yu are with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798 (e-mail: ywang29@e.ntu.edu.sg; haoyu@ntu.edu.sg).

X. Li is with Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: xinli@cmu.edu).

K. Xu and F. Ren are with the School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, AZ 85281 USA (e-mail: kaixu@asu.edurenfengbo@asu.edu; renfengbo@asu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TBCAS.2016.2597310

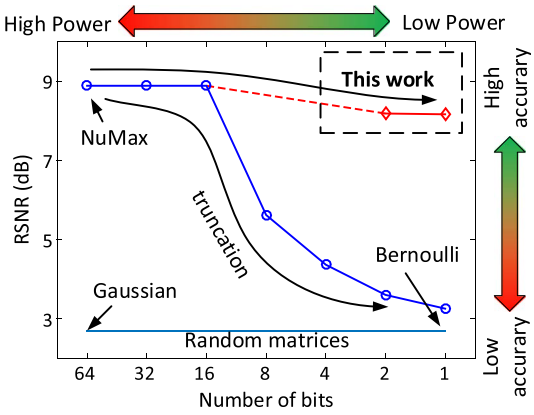


Fig. 1. The comparison of information loss of NuMax through truncation quantization versus proposed quantization. This work achieves both high signal recovery performance as well as low sampling hardware cost by proposed quantization algorithm.

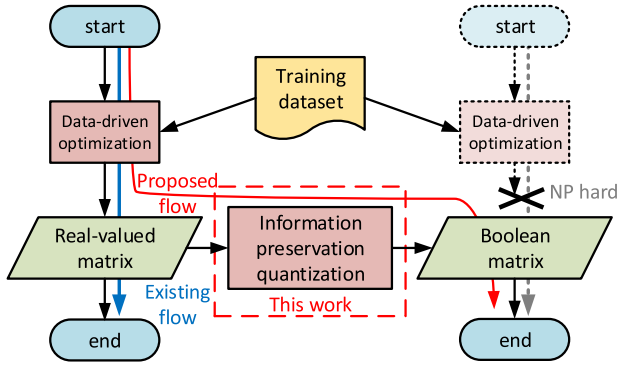


Fig. 2. The proposed flow for data-driven Boolean sampling matrix optimization.

consumption [4], [7], [13]. In addition, the recent emerging resistive random-access memory (RRAM) [14]–[16] in crossbar (or cross-point) structure [14] can provide intrinsic fabric for matrix-vector multiplication, which potentially enables both energy- and area-efficient hardware implementation of linear embedding. However, the limited RRAM programming resolution also favors only Boolean embedding matrices to be mapped to RRAM crossbar structure.

Therefore, the hardware realization of sampling matrix faces a dilemma. On one hand, if data-driven optimized real-valued embedding, such as NuMax [12], is mapped for better recovery quality, large power overhead will be expected. On the other hand, if non-data-driven random Boolean embedding or Boolean Reed-Muller [10], [11] embedding is mapped for better hardware energy-efficiency, high signal recovery accuracy cannot be accomplished. Such trade-off is illustrated in Fig. 1. Fig. 1 also reveals that quantization of NuMax by straightforward truncation works only above precision of 16-bit, below which it will incur significant performance degradation. Therefore, without data-driven optimized Boolean embedding, the advantages of both sides cannot be achieved simultaneously. The challenge to perform data-driven Boolean embedding optimization is that, with large amount of dataset involved in the sampling matrix optimization, only convex methods with real-valued optimized matrices are feasible.

In this paper, towards high performance (data-driven) and low power (Boolean) sampling, instead of optimizing Boolean embedding on original dataset, we propose an optimizing algorithm that transforms a data-driven optimized real-valued sampling matrix to a Boolean sampling matrix. The proposed optimization flow is illustrated in Fig. 2. As the input optimized real-valued embedding matrix is optimized towards the specific dataset, and the proposed algorithm seeks least intrinsic information loss, so the resulting Boolean embedding matrix is still optimized towards the same training dataset. In addition, we have discussed the corresponding hardware implementations based on both CMOS technology and emerging non-volatile resistive random-access-memory (RRAM) technology for obtained optimized Boolean embedding. Such capability was first exploited in our preliminary work [17]. The numerical experiments demonstrate that the proposed data-driven Boolean embedding can combine both high signal quality and also low sampling power. Specifically, it can improve image recovery quality (RSNR) by 9 dB compared to the non-data-driven Bernoulli embedding, and improve energy efficiency by 4.6× than that of data-driven real-valued sampling circuit.

The contributions of this paper are summarized as below:

- To our best knowledge, the data-driven optimized Boolean sampling matrix is constructed for the first time. Being Boolean and optimized towards dataset, we achieve both highest signal recovery quality and best hardware energy efficiency among all existing schemes.
- We formulate the problem of finding best transformation that quantizes real-valued sampling matrix into Boolean matrix with minimal information loss.

The rest of this paper is organized as follows. Section II introduces the background of compressive sensing and near-isometric embedding. Section III presents the sampling hardware for Boolean embedding with the corresponding optimization problem formulated. Sections IV and V detail the two proposed Boolean embedding optimization algorithms. Numerical results are presented in Section VI with conclusion in Section VII.

II. Background

A. Compressive Sensing and Isometric Distortion

Recently, the emerging theory of compressive sensing has enabled the recovery of undersampled signal, if the signal is sparse or has sparse representation on certain basis, such as wavelet transformation and Fourier transformation. And the recovery can be achieved by solving

$$\begin{aligned} \underset{x\in\mathbb{R}^N}{\text{minimize}} \quad & \|x\|_1 \\ \text{subject to} \quad & y = \Psi\Omega x \end{aligned} \tag{1}$$

where $x \in \mathbb{R}^N$ is the sparse coefficients vector and $\Omega \in \mathbb{R}^{N\times N}$ is the basis on which the original signal is sparse; $\Psi \in \mathbb{R}^{M\times N}$ is the sensing matrix and $y \in \mathbb{R}^M$ $(M \ll N)$ the undersampled data in low dimension. To ensure a successful recovery, the

TABLE I
NOTATION TABLE OF USED MATHEMATICAL SYMBOLS

Symbols	Descriptions
δ	the isometric distortion of the restricted isometry property (RIP)
x	original high-dimensional sparse signal to be sampled by compressive sensing
y	sampled low-dimensional signal by compressive sensing
Ω	sparse basis/dictionary
$\hat{x}$	reconstructed/recovered original signal
Ψ	sampling matrix
$\hat{\Psi}$	Boolean sampling matrix losslessly quantized from real-valued Ψ
T	orthogonal RIP preserving transformation matrix for quantization
t_i	i_{th} row of T obtained during row-generation algorithm
$\hat{\psi}_i$	i_{th} row of $\hat{\Psi}$ obtained during row-generation algorithm
χ	training dataset $\chi = \{x_1, x_2, ..., x_i\}$ for sampling matrix optimization
$S(\chi)$	set of all pairwise distances among every two x in χ as input for NuMax
χ'	testing dataset that has no overlap with training dataset χ
σ_{LRS}	standard deviation of RRAM resistance in low resistance state (LRS)
σ_{HRS}	standard deviation of RRAM resistance in high resistance state (HRS)

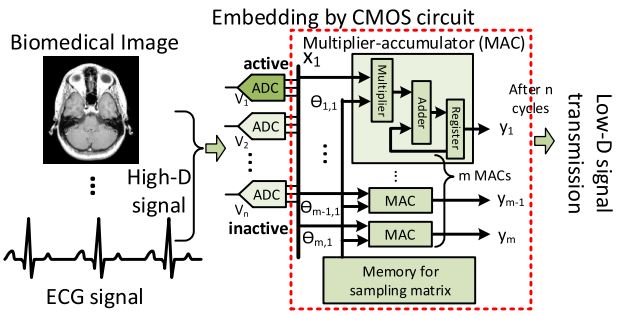


Fig. 3. The embedding circuit by CMOS matrix-vector multiplier.

sensing matrix (Ψ) must meet the restricted isometry property (RIP), which is defined as: if there exists a $\delta \in (0, 1)$ such that the following equation is valid for every vector $v \in \mathbb{R}^N$

$$(1-\delta)\|v\|_2^2 \leq \|\Psi v\|_2^2 \leq (1+\delta)\|v\|_2^2 \tag{2}$$

then Ψ has the RIP with isometric distortion constant δ. The notations of all used symbols are summarized in Table I.

B. Optimized Near-Isometric Embedding

The easiest way to construct a matrix with RIP is to generate a random matrix. The work [18] proves that random matrix is of a very high possibility to satisfy RIP, yet not deterministic. Different from the random Bernoulli sampling matrix that the RIP is probabilistic, a data-driven sampling matrix can ensure the RIP of the finite given data points. One recent work in [12] proposed the **NuMax** framework to construct a near-isometric embedding matrix with deterministic RIP. Given a dataset $\chi = \{x_1, x_2, \ldots, x_i\} \in \mathbb{R}^N$, the NuMax produces an optimized continuous-valued embedding matrix Ψ so that every pairwise distance vector v for χ can preserve its norm after embedding up to a given distortion tolerance $\delta_{\max}$.

Once the optimized NuMax sampling matrix Ψ is obtained, the signal acquisition $y = \Psi\Omega x$ can be performed by multiplying the embedding matrix Ψ with signal vector Ωx. The conventional CMOS circuit based data acquisition front-end that performs real-valued embedding is shown in Fig. 3. The sampling circuit has two major components, the SRAM memory that stores the embedding matrix, and the multiplier-accumulators (MAC) that perform multiplication and addition. For an embedding matrix $\Psi \in \mathbb{R}^{m\times n}$ $(m \ll n)$, in each cycle,

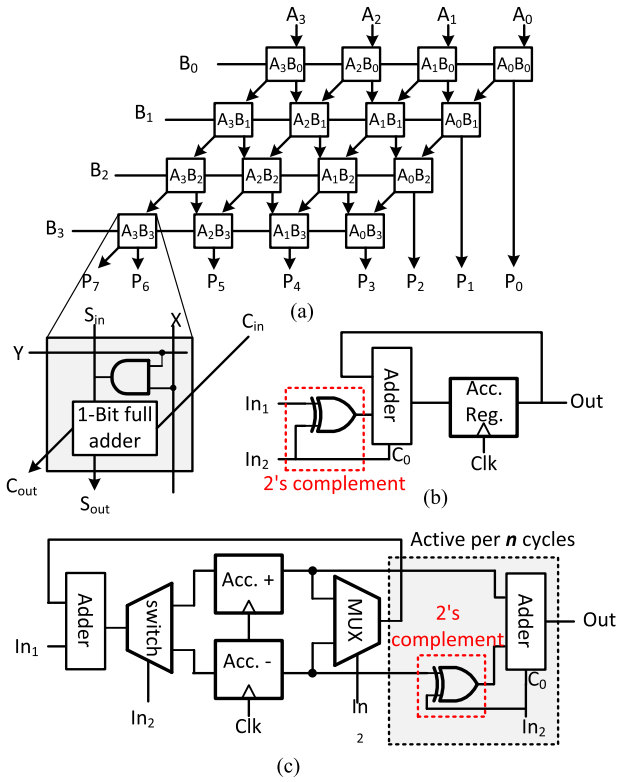


Fig. 4. The implementation of (a) MAC with multiplier in 4-bit resolution for real-valued embedding matrix and (b) MAC for $\{-1, 1\}^{m\times n}$ embedding matrix (c) MAC for $\{-1, 1\}^{m\times n}$ embedding matrix with power optimization.

m MACs multiply one element of input vector Ωx with one column of Ψ, and then add with previously accumulated results. Therefore, it requires n cycles to obtain the embedded signal y.

As NuMax produces real-valued Ψ, the precision of Ψ substantially determines the hardware complexity. For example, a multiplier in MAC with 4-bit resolution, shown in Fig. 4(a), requires 16 full-adders. In fact, the number of required full-adders generally depends quadratically on the precision of both Ωx and Ψ. For the typical precision of real-valued elements in sampling matrix, 16-bit (Fig. 1) resolution may lead to as many as hundreds of full-adders for each MAC, which makes the real-valued NuMax embedding less appealing for signal acquisition hardware mapping.

III. BOOLEAN EMBEDDING FOR SIGNAL ACQUISITION FRONT-END

A. CMOS-Based Boolean Embedding Circuit

The mapping of a Boolean embedding matrix can eliminate the usage of multipliers. For a $\{0, 1\}^{m\times n}$ Boolean embedding matrix, the MAC only accumulate signal data when Boolean multiplicand is 1. For a more general $\{-1, 1\}^{m\times n}$ Boolean matrix, the Boolean multiplicand indicates addition or subtraction for the signal data. That is to say, the required resources of full-adders are only linearly depending on the precision of signal Ωx. As such, the hardware resource can be significantly reduced.

Specifically for the $\{-1, 1\}^{m\times n}$ embedding matrix mapping, the multiplication by -1 requires the calculation of 2's complement. The intuitive approach is illustrated in Fig. 4(b). The

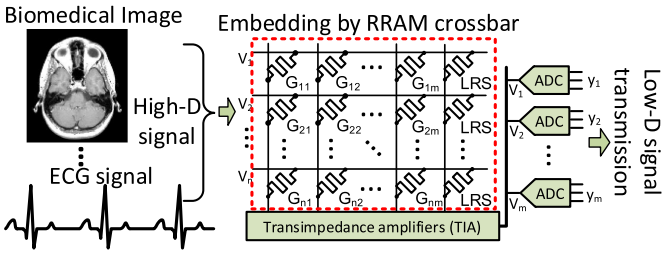


Fig. 5. The embedding circuit by emerging non-volatile RRAM crossbar.

In_2 signal is 1 for multiplying -1 and 0 for multiplying 1. To obtain the 2's complement, the XOR logic is used to get the complement of each bit and the C_0 (carry 0) is applied as well. However, the 2's complement calculation close to input will incur substantial dynamic power for the combinational logic. Instead, the circuit diagram in Fig. 4(c) first accumulates all data to be multiplied by 1 and -1 separately, and the subtraction is performed at the very last cycle. Therefore, the 2's complement circuit is only active every n cycles, which can greatly improve the power efficiency.

To be compatible with optimized sampling matrices such as Reed-Muller code [10], [11] and NuMax [12], SRAM block is required to store the matrix and provide reconfigurability. For non-optimized sampling matrices such as random Gaussian and Bernoulli, apart from storing the matrix in SRAM, the matrix can also be generated at runtime, which can improve hardware efficiency. In practice, the pseudo-random number generator (PRNG) is used [7], [13], which avoids the power-consuming SRAM arrays. As PRNG produces 0/1 sequences with a pre-determined pattern, one of the issue for PRNG is the self-coherence. For example, a 8-bit per cycle PRNG has a period of 256 (2^8), and when filling a $n \times 256$ sampling matrix by rows with such 0/1 sequences, all rows of the matrix will be identical. This can be overcome by increasing the number of bits the PRNG produces per cycle at a cost of higher hardware complexity. With limited hardware resources, the pseudo-random numbers generated by PRNG usually lead to performance degradation of signal acquisition compared to stored random sampling matrix.

B. RRAM Crossbar Based Boolean Embedding Circuit

The emerging resistive random-access-memory (RRAM) crossbar [14], [15] provides an intrinsic in-memory fabric of matrix-vector multiplication, which is proposed in Fig. 5. Compared to CMOS embedding circuit, RRAM crossbar based approach can provide three major advantages: 1) embed the sensing matrix [Ψ in (1)] in-memory without the need for loading it externally each cycle, 2) perform the matrix-vector multiplication in single cycle, and 3) minimize the leakage power due to its non-volatility. A RRAM crossbar structure is composed of three layers: horizontal wires at top layer, vertical wires at bottom layer and RRAM devices in the middle layer at each cross-point. For a $m \times n$ RRAM crossbar, assume the input signal of i_{th} row is V_I^i and the conductance of RRAM device on i_{th} row j_{th} column is G_{ij}, then the output current flowing down j_{th} column $I_j = \Sigma_{i=1}^{m} V_I^i G_{ij}$. In other words, crossbar structure intrinsically supports in-memory embedding operation

$$\begin{bmatrix} V_O^1 \\ V_O^2 \\ \vdots \\ V_O^m \end{bmatrix} = Z \begin{bmatrix} G_{11} & G_{12} & \cdots & G_{1n} \\ G_{21} & G_{22} & \cdots & G_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ G_{m1} & G_{m2} & \cdots & G_{mn} \end{bmatrix} \begin{bmatrix} V_I^1 \\ V_I^2 \\ \vdots \\ V_I^n \end{bmatrix} \quad (3)$$

where Z is the transimpedance of the transimpedance amplifier (TIA) and V_O^i the output voltage of i_{th} column. It must be ensured that input $||V_I||_\infty \ll V_{th}$ to avoid accidental value changes of G, in which the V_{th} is the device programming threshold voltage.

The mapping of embedding matrix is accomplished by the resistance programming of RRAM crossbar according to Ψ. Intended for memory usage, RRAM devices are commonly bistable with on-resistance and off-resistance ratio as high as $10^3 \sim 10^4$ [15], [19]. Resistance programming with higher resolution has been demonstrated in 4 or 5 levels at most [19], [20]. Therefore, resistance programming in continuous (or close-continuous) value resolution is practically challenging due to large process variation under current manufacture technology. In other words, the real-valued sampling matrix does not comply with RRAM crossbar and Boolean sampling matrix is preferred.

As the RRAM crossbar is essentially (0,1) binary in terms of conductance, the mapping of (0,1) Boolean matrix follows: 0 corresponds to high resistance state (HRS) and 1 maps to low resistance state (LRS). To map $\hat{\Psi} \in \{-1, 1\}^{m \times n}$, simple linear transformation needs to be considered: $\hat{\Psi}x = (2\Theta - J)x = 2\Theta x - Jx$, where $\Theta \in \{0, 1\}^{m \times n}$, J all-ones matrix and x input vector. The Jx is implemented by an additional all-LRS column that generates Σx as current offset for other columns, as shown in Fig. 5. The sampling matrices that each type of hardware supports are illustrated in Table II.

TABLE II
COMPATIBILITY OF DIFFERENT SAMPLING MATRICES ON VARIOUS HARDWARE PLATFORMS

Hardware	MAC/MEM	RRAM crossbar	MAC/PRNG
NuMax	✓	×	×
this work	✓	✓	×
Reed-Muller	✓	✓	×
Bernoulli	✓	✓	✓
Gaussian	✓	×	✓

C. Problem Formulation

For an sampling matrix Ψ that satisfies RIP with distortion of δ_Ψ, the following equation will also hold true:

$$(1 - \delta_\Psi)\|x\|_2^2 \leq \|T\Psi x\|_2^2 \leq (1 + \delta_\Psi)\|x\|_2^2 \quad (4)$$

if T is an orthonormal rotation matrix. In other words, if we can find an orthonormal rotation matrix that transforms real-valued NuMax embedding matrix Ψ into a matrix that is close enough to a Boolean matrix $\hat{\Psi}$, then the Boolean embedding of $\hat{\Psi}$ can preserve original distortion δ_Ψ. In other words, the resulting $\hat{\Psi}$ is still optimized towards the same training dataset

as NuMax embedding Ψ, and meanwhile it can be efficiently mapped to circuits in Figs. 4(c) and 5 with greatly reduced power consumption.

The Boolean sampling matrix optimization can be then formulated as the following optimization problem:

$$\begin{aligned} \underset{T,\hat{\Psi}}{\text{minimize}} \quad & \|T\Psi - \hat{\Psi}\|_F^2 \\ \text{subject to} \quad & T^T \cdot T = I \\ & \hat{\Psi} \in \{-1, 1\}^{m\times n} \end{aligned} \tag{5}$$

where $\Psi \in \mathbb{R}^{m\times n}$ $(m < n)$ is the optimized real-valued sampling matrix learned from dataset, that projects data from high n-dimension to low m-dimension; $T \in \mathbb{R}^{m\times m}$ is an orthonormal rotation matrix that attempts to transform Ψ to a Boolean matrix. $\hat{\Psi} \in \mathbb{R}^{m\times m}$ is the closest Boolean matrix solution where closeness is defined by the Frobenius norm.

Ideally, if an orthonormal transformation matrix T can rotate Ψ to an exact Boolean matrix, i.e., the optimal value of (5) is zero, then the distortion δ of optimized Boolean embedding will be exactly the same as the NuMax real-valued embedding. In practice, with a non-zero optimal value, the closeness of $T\Psi$ to $\hat{\Psi}$ indicates information loss degree from Ψ to $\hat{\Psi}$. Alternatively, it can be interpreted as an equivalent near-orthogonal rotation T' transforming the real-valued Ψ to an exact Boolean $\hat{\Psi}$. The degree of orthogonality implies the information loss of Ψ.

IV. Iterative Heuristic Algorithm

It is intractable to solve the problem formulated in (5) considering the orthogonal constraint $T^T \cdot T = I$ and the integer constraint $\hat{\Psi} \in \{-1, 1\}$ simultaneously, as both constraints are non-convex. When one constraint is considered at one time, (5) can be split into two manageable problems: if the orthogonal constraint is considered for T, and $\hat{\Psi}$ a given Boolean matrix, the problem becomes the search of an *orthogonal rotation* matrix for maximal matrix agreement; if the integer constraint is considered for $\hat{\Psi}$, and T a given orthogonal matrix, the problem turns to a *Boolean quantization* problem. In this section, a heuristic approach is proposed that iteratively solves *orthogonal rotation* problem and *Boolean quantization* problem, and gradually approximates the optimal solution of $\hat{\Psi}$ in each round.

A. Orthogonal Rotation

The problem of finding an orthogonal transformation matrix T that can rotate a given real-valued projection matrix Ψ to another given Boolean matrix $\hat{\Psi}$ can be formulated as

$$\begin{aligned} \underset{T,k}{\text{minimize}} \quad & ||kT\Psi - \hat{\Psi}||_F^2 \\ \text{subject to} \quad & T^T \cdot T = I. \end{aligned} \tag{6}$$

The cost function can be represented by trace function as

$$||kT\Psi - \hat{\Psi}||_F^2 = k^2\text{Tr}(\Psi^T\Psi) + \text{Tr}(\hat{\Psi}^T\hat{\Psi}) - 2k\text{Tr}(T^T\hat{\Psi}\Psi^T). \tag{7}$$

As Ψ and $\hat{\Psi}$ are given matrices, $\text{Tr}(\Psi^T\Psi)$ and $\text{Tr}(\hat{\Psi}^T\hat{\Psi})$ are therefore two constants. Consider k as constant first, the formulated optimization problem in (6) can be rewritten as

$$\begin{aligned} \underset{T}{\text{maximize}} \quad & \text{Tr}(T^T\hat{\Psi}\Psi^T) \\ \text{subject to} \quad & T^T \cdot T = I \end{aligned} \tag{8}$$

and with the singular value decomposition $\hat{\Psi}\Psi^T = U\Sigma V^T$ where $\Sigma = \text{diag}(\sigma_1, \ldots, \sigma_n)$, the cost function of (8) can be rewritten as

$$\begin{aligned} \text{Tr}(T^T\hat{\Psi}\Psi^T) &= \text{Tr}(T^TU\Sigma V^T) \\ &= \text{Tr}(V^TT^TU\Sigma) \le \sum_{i=1}^{n}\sigma_i. \end{aligned} \tag{9}$$

The inequality holds as V, T, and U are all orthonormal matrices. As such, the trace is maximized when $V^TT^TU = I$, which leads to

$$T = UV^T. \tag{10}$$

To optimize k, let $\partial f/\partial k = 0$ in which f is the cost function of (7), and the best scaling factor can be obtained by

$$k = \frac{\text{Tr}(T^T\hat{\Psi}\Psi^T)}{\text{Tr}(\Psi^T\Psi)}. \tag{11}$$

B. Quantization

T is a known orthogonal transformation matrix, and Ψ is a given real-valued optimized projection matrix, the problem to find its closest Boolean matrix can be formulated as

$$\begin{aligned} \underset{\hat{\Psi}}{\text{minimize}} \quad & ||kT\Psi - \hat{\Psi}||_F^2 \\ \text{subject to} \quad & \hat{\Psi} \in \{-1, 1\}. \end{aligned} \tag{12}$$

It is obvious that the solution for (12) is

$$\hat{\Psi}_{ij} = \begin{cases} 1, & (kT\Psi)_{ij} \ge 0 \\ -1, & (kT\Psi)_{ij} < 0. \end{cases} \tag{13}$$

This can be seen as Boolean quantization. The quantization error can be defined as

$$e = ||kT\Psi - \hat{\Psi}||_F^2. \tag{14}$$

In ideal case, the error would be zero which means an orthogonal transformation T on optimized real-valued projection matrix Ψ finds an exact Boolean matrix $\hat{\Psi}$. Therefore, the distortion $\delta_{\hat{\Psi}}$ caused by $\hat{\Psi}$ will be the same as δ_Ψ. With $e \neq 0$, it can be inferred that $\delta_{\hat{\Psi}} > \delta_\Psi$. To reduce the quantization error, it is an intrinsic idea to increase the level of quantization.

Consider a modified problem formulation

$$\tilde{\Psi}_{ij} = \begin{cases} 1, & (kT\Psi)_{ij} \ge 1/2 \\ 0, & -1/2 \le (kT\Psi)_{ij} < 1/2 \\ -1, & (kT\Psi)_{ij} < -1/2 \end{cases} \tag{15}$$

with each element of the matrix Ψ normalized within the interval of $[-1, 1]$. It is important to keep matrix Boolean so that it can be mapped to RRAM crossbar structure efficiently, thus it requires that the matrix $\tilde{\Psi}$ can be split into two Boolean matrices $\tilde{\Psi} = (1/2)(\hat{\Psi}^1 + \hat{\Psi}^2)$ where $\tilde{\Psi} \in \{-1, 0, 1\}$ and $\hat{\Psi}^1, \hat{\Psi}^2 \in \{-1, 1\}$. With Boolean quantization, only one projection RRAM crossbar is needed. Two RRAM crossbars are needed for the three-level quantization case, as a result of trade-off between error and hardware complexity.

C. Overall Optimization Algorithm

The heuristic optimization process is summarized in Algorithm 1. Given some initial guess of $\hat{\Psi}$, the inner loop of Algorithm 1 tries to find the local close-optimal solution by improving $\hat{\Psi}$ through iterations. Within each iteration, (6) and (12) are solved by singular vector decomposition and quantization as concluded in (10) and (15), respectively. The iterations terminate when the $\hat{\Psi}$ stops improving and converges.

Algorithm 1: Iterative heuristic Boolean sampling matrix optimization algorithm

input : real-valued embedding matrix Ψ, search width, and quantization level
output: optimized Boolean embedding matrix $\hat{\Psi}_{opt}$

initialize $\hat{\Psi}_{opt} \leftarrow$ random $m \times n$ Bernoulli matrix;
while *not reach search width limit* **do**
  seed $\leftarrow$ random $m \times m$ matrix;
  U, S, V $\leftarrow$ SVD of seed;
  T $\leftarrow$ U;
  while *not converged* **do**
    $\hat{\Psi} \leftarrow$ quantization of $T\Psi$;
    U, S, V $\leftarrow$ SVD of $\hat{\Psi}\,\Psi^T$;
    T $\leftarrow$ UV;
    $k \leftarrow \mathrm{Tr}\,(T^T\,\hat{\Psi}\,\Psi^T)/\,\mathrm{Tr}\,(\Psi^T\Psi)$;
    if $||kT\Psi - \hat{\Psi}\,||_F^2 < ||k_{opt}T_{opt}\Psi - \hat{\Psi}_{opt}\,||_F^2$ **then**
      $\hat{\Psi}_{opt} \leftarrow \hat{\Psi}$;

As both integer constraint and orthogonal constraint are non-convex, the local optimum in most cases is not optimal globally. In other words, the solution strongly depends on the initial guess that leads to the local close-optimum. Therefore, the outer loop of Algorithm 1 increases the search width by generating numerous initial guesses that are scattered within orthogonal matrices space. For each initial guess it will gradually converge to a local optimum, thus the increase of search width will compare numerous local optimal solutions and approximate the global optimum.

V. Row Generation Algorithm

The formulated problem in (5) is a mixed-integer non-linear programming (MINLP) problem, as it has both nonlinear orthogonal constraint $T^T \cdot T = I$ and the integer constraint $\hat{\Psi} \in \{-1, 1\}^{m \times n}$. Although such mixed-integer non-linear programming (MINLP) problem can be solved by existing algorithms such as genetic algorithm [21], it lacks efficiency and only problem in small size can be managed. For the embedding matrix in compressive sensing, the transformation matrix T could have dozens of rows while matrix $\hat{\Psi}$ may have thousands of Boolean variables, so current solvers may fail in such scale. In this section, we proposed a row generation algorithm that also can efficiently tackle the problem.

A. Elimination of Norm Equality Constraint

The orthonormality of T in (5) implies two specific constraints, the orthogonality of rows of T that

$$t_i^T \cdot t_j = 0 \quad \forall\, i, j \text{ that } i \neq j \tag{16}$$

and the norm equality that

$$\|t_i\|_2^2 = 1 \quad \forall\, i \tag{17}$$

where t_i is the i_{th} row of T. Both imply numerous quadratic equality constraints (non-convex) and therefore hard to manage simultaneously. The non-convex quadratic norm equality constraint of rows of T indicates the normalization of rows after orthogonality is satisfied. In the following, we show how the norm equality constraint can be eliminated without affecting the solution accuracy of problem in (5).

Assume we only impose orthogonal constraint on T rather than more strict orthonormal constraint, the original problem can be then relaxed to

$$\begin{aligned} \underset{T,\hat{\Psi}}{\text{minimize}} \quad & \|T\Psi - \hat{\Psi}\|_F^2 \\ \text{subject to} \quad & T^T \cdot T = D^2 \\ & \hat{\Psi} \in \{-1, 1\}^{m \times n} \end{aligned} \tag{18}$$

where $D = \mathrm{diag}(d_1, d_2, \ldots, d_m)$ is a diagonal matrix, and d_i is the norm of i_{th} row of T. That is to say, an additional row scaling operation is introduced during the sensing stage

$$y = D^{-1}\hat{\Psi}\Omega x \tag{19}$$

where $\hat{\Psi} \approxeq T\Psi$ is the optimized Boolean embedding matrix that can be efficiently realized in hardware, Ω is the orthonormal sparse basis of original signal, and x is the sparse coefficients.

In fact, the row scaling operation during signal acquisition is unnecessary and can be transferred to recovery stage if an implicit sensing is performed

$$\hat{y} = \hat{\Psi}\Omega x \tag{20}$$

with corresponding signal reconstruction by

$$\begin{aligned} \underset{x \in \mathbb{R}^N}{\text{minimize}} \quad & \|x\|_1 \\ \text{subject to} \quad & |D^{-1}\hat{y} - D^{-1}T\Psi\Omega x| \leq \epsilon \end{aligned} \tag{21}$$

where ϵ is the tolerance for noise on sampled signal data $\hat{y}$. As such, the norm equality constraint is eliminated while the compressive sensing signal acquisition front-end hardware complexity stays the same and recovery quality is not affected.

B. Convex Relaxation of Orthogonal Constraint

To construct a transformation matrix T with orthogonal rows and minimize the cost function at the same time is challenging. In the following, we propose a convex row generation algorithm that seeks local optimal solution. The idea is to construct each row of T at one time while minimize the cost function. Assume $t_1, t_2, \ldots, t_{i-1}$ are first $i-1$ rows that are already built with orthogonality, to construct the i_{th} row t_i

$$\begin{aligned} \underset{t_i, \hat{\psi}_i}{\text{minimize}} \quad & \|t_i \Psi - \hat{\psi}_i\|_2^2 \\ \text{subject to} \quad & \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_{i-1} \end{bmatrix} \cdot t_i^T = \mathbf{0} \\ & \hat{\psi}_i \in \{-1, 1\}^n. \end{aligned} \tag{22}$$

In other words, each time to construct a new row t_i, it has to be orthogonal with previously built $t_1, t_2, \ldots, t_{i-1}$. The iterative row generation algorithm is shown in Algorithm 2. From the geometric perspective, Algorithm 2 seeks to find an orthogonal basis in the m-dimensional space iteratively. Initially T is empty so the first direction vector has the greatest freedom to minimize the cost function. After the first basis vector is chosen, the algorithm finds the best basis vector in the left (m-1)-dimensional subspace that best minimizes the target function. This is iteratively performed until the last direction is selected in only 1-dimensional subspace with freedom for length only.

As T is a square matrix, there always exists a solution for Algorithm 2. The MINLP problem with $m \times n$ integer variables in (18) is therefore relaxed to m MINLP sub-problems each with only n integer variables.

Algorithm 2: Iterative row generation algorithm

input : real-valued embedding matrix Ψ
output: orthogonal transformation matrix $\mathcal{T}$, optimized Boolean embedding matrix $\hat{\Psi}$

1 initialize $\mathcal{T} = \emptyset$, $\hat{\Psi} = \emptyset$;
2 **for** $i \leftarrow 1$ **to** m **do**
3 get t_i by solving problem in Eq. 22 ;
4 update $\mathcal{T} = \begin{bmatrix} \mathcal{T} \\ t_i \end{bmatrix}$, $\hat{\Psi} = \begin{bmatrix} \hat{\Psi} \\ \hat{\psi}_i \end{bmatrix}$;

C. Overall Optimization Algorithm

The overall algorithm to solve (18) is illustrated in Algorithm 2. The 0–1 programming problem in (22) within the loop can be readily solved by branch-and-cut method, under the condition that the number of Boolean variable is kept small. The branch-and-cut method is widely implemented in solvers such as MOSEK [22] and BARON [23].

Without the linearization by row generation, the branch-and-cut method cannot be applied as the orthogonal constraint is strongly nonlinear thus evaluation of lower and upper bounds for each sub-problem will be extremely complicated. In addition, the linearization by row generation significantly reduces the number of Boolean variables thus reduces the worst-case complexity from $2^{m \times n}$ to $m \cdot 2^n$. As such, the row generation together with widely available integer programming solvers can find solution for problem formulated in (18).

VI. Numerical Results

A. Experiment Setup

In this part, we evaluate different compressive sensing sampling matrices from both software and hardware perspectives. The numerical experiments are performed within Matlab on a desktop with 3.6 GHz Intel i7 processor and 16 GB memory. The software performance of sampling matrices is mainly characterized by the signal recovery quality of sampling matrices. For this purpose, both LFW image data [24] and biomedical ECG data [25] are used. For both types of data, the NuMax [12] optimization is first applied with varied training parameter δ values ($[0.05, 0.1, \ldots, 0.35]$), NuMax produces optimized real-valued sampling matrices with different ranks. As depicted in the flow chart in Fig. 2, the proposed algorithms are then applied to Booleanize NuMax sampling matrices. Apart from above data-driven sampling matrices, random Gaussian, Bernoulli, and Reed-Muller [10], [11] (non-data-driven optimization) sampling are also compared. The **reconstructed signal-to-noise ratio (RSNR)** is used as signal recovery quality metric, which is defined as

$$\text{RSNR} = 20 \log_{10} \left(\frac{\|x\|_2}{\|x - \hat{x}\|_2} \right) \tag{23}$$

where x is the original signal and $\hat{x}$ is the reconstructed signal.

With respect to hardware cost consideration, above all sampling matrices can be mapped to three different sampling hardware configurations. Specifically, the MAC/SRAM, MAC/PRNG and RRAM crossbar configurations with their variations are evaluated to examine the hardware friendliness of all the sampling schemes. For real-valued MAC, 16-bit resolution is used as we find that resolution higher than 16-bit will not improve accuracy, as shown in Fig. 1. For RRAM crossbar, the resistance of 1 KΩ and 1 MΩ are used for RRAM on-state resistance and off-state resistance according to [19]. The area of the RRAM crossbar is evaluated by multiplying the cell area ($4F^2$) with sampling matrix size plus one additional column to calculate current offset as discussed in Section III-B. Dynamic power of the RRAM crossbar is evaluated statistically under 1000 random input patterns following an uniform distribution with voltage ranging from -0.5 V to 0.5 V ($|V| < |V_{\text{set}}| = 0.8$ V and $|V| < |V_{\text{reset}}| = 0.6$ V [19]) and the duration of operation is 5 ns [19]. Both the real-valued and Boolean digital CMOS matrix multiplier designs are implemented in Verilog and synthesized with GlobalFoundries 65 nm low power PDK. A pseudo-random number generator design [7] is also implemented and synthesized. The SRAM that stores sampling matrix is evaluated by CACTI [26] memory modeling tool with the setting of 65 nm low standby power fabrication process.

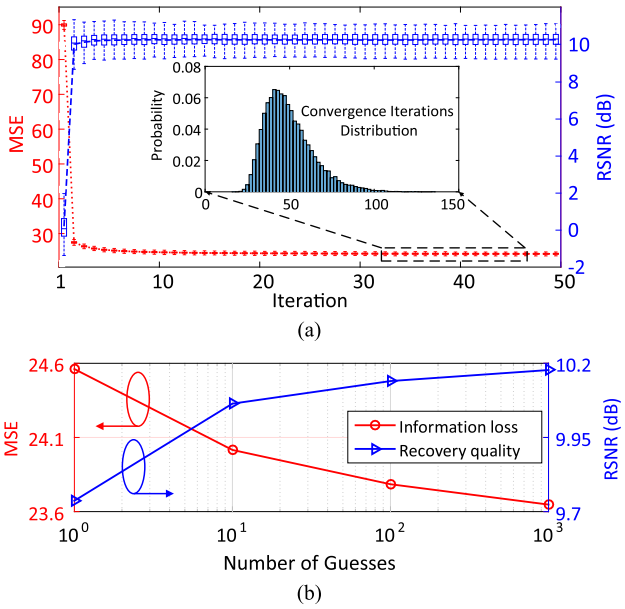


Fig. 6. The algorithm efficiency for (a) local search convergence and (b) global search convergence.

Table II shows all valid combinations of sampling matrix and hardware configuration that will be compared in the section. Among all the combinations, we will show in this section that our proposed sampling matrix can achieve both best signal recovery quality and hardware efficiency.

B. Iterative Heuristic Algorithm on High-D ECG Signals

For training stage (NuMax), 1,000 ECG periods in dimension of 256 are randomly picked from database [25] as the dataset χ, which leads to around 1 million of pairwise distance vectors in set $S(\chi)$. For testing phase, another 1,000 ECG periods are selected as dataset χ', which have no overlap with learning dataset χ. The ECG signal reconstruction is performed on unseen data set χ' by solving (1) with Battle-Lemarie wavelet bases used.

a) Algorithm convergence and effectiveness: The efficiency of Algorithm 1 can be examined from two aspects, finding both local and global optima. The efficiency of finding local optimum is assessed by convergence rate. The local search terminates when the approximation error $||T\Psi - \hat{\Psi}||_F^2$ stops improving.

Given specific RIP upper-bounds, NuMax [12] provides Ψ with different ranks. With RIP constraint of 0.1, the NuMax produces a $\Psi \in R^{19\times 256}$ sampling matrix. Algorithm 1 is applied to Ψ with total 10 000 repeated local search and the convergence is illustrated in Fig. 6(a). It can be observed that the relative error reduces dramatically within the first few iterations. The zoomed sub-figure shows that local search on average converges within 50 iterations, where convergence is defined as less than 1e-6 error reduction in two consecutive iterations. Generally, the local optimum can be considered found in less than 100 iterations.

The global search is achieved by scattering many initial guesses in the orthogonal matrices space for T, and comparing the corresponding local optima. The errors under varying number of initial guesses are shown in Fig. 6(b). Considering the Boolean constraint and the orthogonal constraint, the problem formulated in (5) is generally NP-hard. Therefore, the relative error can be improved by scattering exponentially more initial guesses, yet no convergence is observed. Hence an efficient search policy should be designed in a way scattering as many initial points as possible and limiting the local search for each initial guess within 100 iterations.

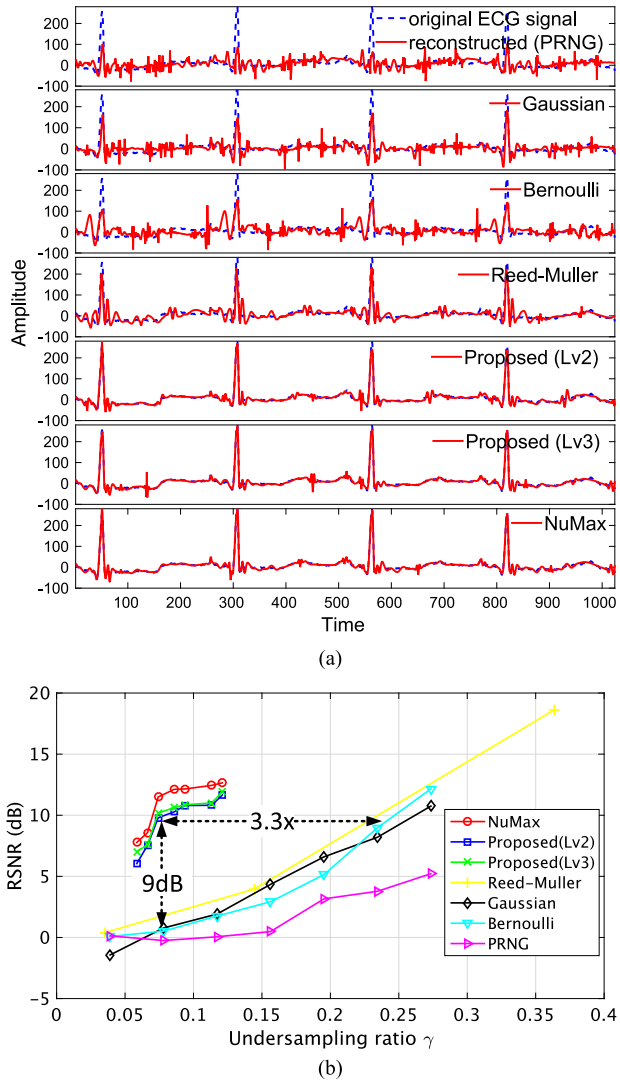


Fig. 7. The recovery quality comparison among different sampling matrices. (a) Examples of recovered ECG signals at $\gamma = 19/256$ and (b) RSNR for 1000 ECG periods.

b) ECG recovery quality comparison: The ECG signal recovery examples at the undersampling ratio $\gamma = 19/256$ are shown in Fig. 7. For non-data-driven sampling matrices, both the random Bernoulli and Gaussian show similar reconstruction quality. In other words, the increase of bits in random numbers will not improve recovery quality. This is because, the increase of bits of random number will not gain any additional information.

The pseudo-random number generator (PRNG) based Bernoulli exhibits the lowest reconstructed signal quality. This

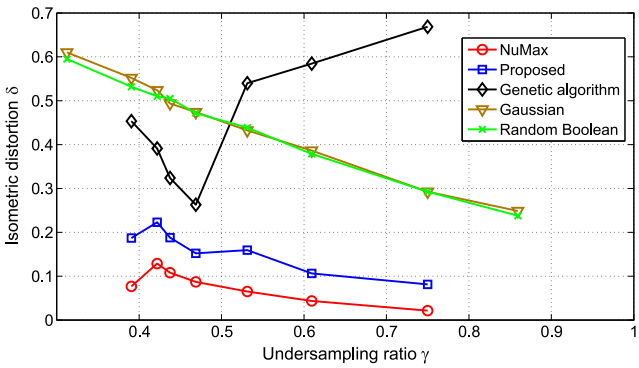


Fig. 8. The isometric distortion on the unseen dataset χ' for different embeddings.

is because, as PRNG produces 0/1 sequences with a predetermined pattern, it has self-coherence issue. For example, a 8-bit per cycle PRNG have a period of 256 (2^8), and when filling a sampling matrix by rows with such 0/1 sequences, all rows of the matrix will be identical.

The Reed-Muller code optimizes the sampling matrix by minimizing the correlations among different rows/columns, which helps to improve sampling performance. Being a generic sampling matrix that works with all data types, it cannot exploit the isometric property of ECG signal, which limits its performance in particular applications. Specifically, it only shows 1 dB improvement compared to random Bernoulli sampling.

The data-driven NuMax real-valued sampling exhibits the best recovery quality (highest RSNR) as shown in Fig. 7(b). The proposed iterative heuristic (IH) algorithm quantizes the real-valued NuMax sampling with slight quality loss. Specifically, at the undersampling ratio $\gamma = 19/256$, the IH (lv2) exhibits 8 dB, 9 dB, and 10 dB higher RSNR than that of Reed-Muller, Bernoulli, and pseudoBernoulli samplings, respectively. Also, the level-3 quantization through (15) can preserve more information than level-2 quantization through (13). The RSNR of IH (lv3) shows marginal 0.48 dB higher RSNR than IH (lv2). The IH (lv3) sampling matrix $\Psi \in \{-1, 0, 1\}^{m\times n}$ will incur additional hardware overhead compared to IH (lv2) $\Psi \in \{-1, 1\}^{m\times n}$.

Fig. 7(a) gives a visual effect of quality of recovered ECG signal segments with different sampling matrices. The data-driven sampling matrices, i.e., NuMax, IH (lv2), and IH (lv3), can recover signals that tightly coincide with original signals.

C. Row Generation Algorithm on Low-D Image Patches

For training stage (NuMax), 6,000 patches with size of 8×8 are randomly picked throughout all images as the dataset χ, which leads to around 18 millions of pairwise distance vectors in set $S(\chi)$. For testing phase, another 6,000 patches with size of 8×8 are selected as dataset χ' with no overlap with learning dataset χ. The image reconstruction is performed on unseen data set χ' by solving (1) with 2D DCT bases.

The genetic algorithm [21] is adopted as the baseline solver for the mixed-integer nonlinear programming (MINLP) in (5), which is compared with the proposed algorithm in Algorithm 2. Both algorithms are run given same amount of time, i.e., $m \times 500$ seconds where m is the rank of Ψ that indicates the size of the problem.

a) Algorithm effectiveness: The idea behind the proposed real-valued matrix Booleanization is to preserve the RIP of NuMax sampling matrix, which differs from the truncation based quantization. The information loss during the quantization is directly related to the RIP preservation. The algorithm effectiveness in this part will be examined by the isometric distortion δ, defined in (2).

The distortions of the all embeddings are tested on unseen dataset χ'. The isometric distortions of both random embeddings are almost invariant. Being optimized on image dataset χ, both the NuMax and proposed (quantized NuMax) are significantly better than random embeddings. With focus on the Boolean sampling matrices that are hardware friendly, the isometric distortion of optimized Boolean embedding is $3.0\times$ better than random Boolean embedding on average.

Due to the near-orthogonal rotation, the optimized Boolean embedding experiences some penalty on isometric distortion δ compared to NuMax approach. For genetic algorithm, as it experiences higher distortion than that of the proposed algorithm, it can be inferred that Algorithm 2 can find a more precise solution. In addition, it can be observed that the genetic algorithm fails when undersampling ratio m/n increases, and this is because the proposed row generation based algorithm requires linearly more time when the number of row m increases, while the genetic algorithm needs exponentially more time. Moreover, the solution provided by genetic algorithm is stochastic, which has no guarantee on its effectiveness while the proposed algorithm is deterministic.

b) Image recovery quality comparison: The recovery examples under $\gamma = 25/64$ are shown in Fig. 9(a). The reconstructed images in blue box correspond to Boolean embeddings that have low power hardware implementations, and images in red box are from optimization based approaches which show lower recovery errors. The genetic algorithm is also optimization based, but the effectiveness is inconsiderable. Therefore, only the proposed can achieve both low power and high recovery performance. The numerical image reconstruction quality is shown in Fig. 9(b). The two random embeddings show similar reconstruction RSNR, which is averagely 8.3 dB lower than that of the proposed optimized Boolean sampling matrix. The RSNR of optimized Boolean embedding is close to that of NuMax embedding, which is 2.5 dB lower as a result of information loss by near-orthogonal rotation.

On the other hand, the genetic algorithm shows no obvious effectiveness of improving recovery quality even though it optimizes a Boolean embedding matrix. The main reason is that during the conversion from Ψ to $\hat{\Psi}$ too much information loss leads $\hat{\Psi}$ to be close to a random Boolean matrix. In other words, the genetic algorithm is ineffective to solve the problem in (5). In addition, the stochastic nature of genetic algorithm makes it necessary to perform the algorithm considerably many times. The proposed algorithm, on the contrary, guarantees to produce a Boolean matrix with high performance with single execution.

The proposed iterative heuristic algorithm is also compared with the row generation (RG) algorithm. The signal recovery quality of RG algorithm outperforms heuristic algorithm by

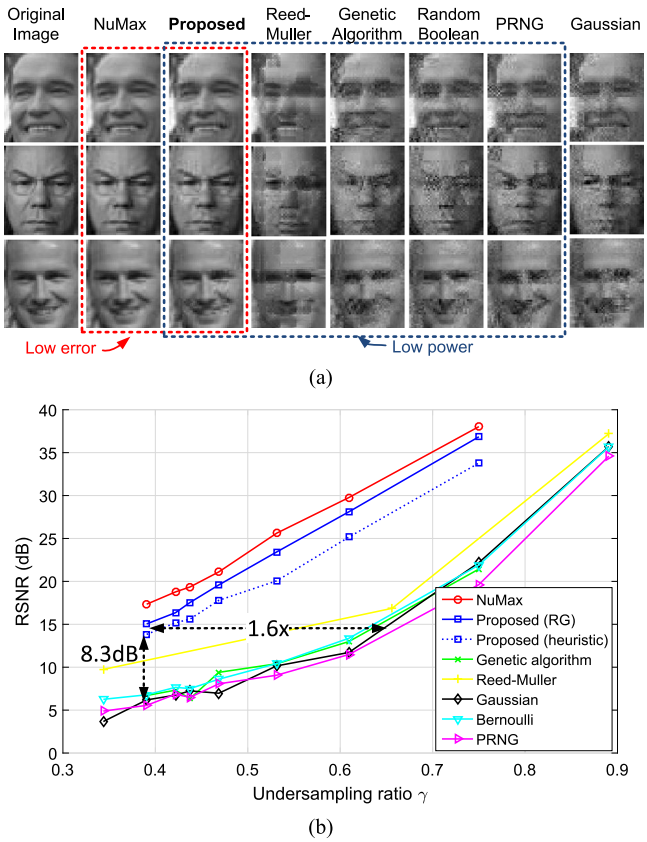


Fig. 9. The recovery quality comparison among different embedding matrices. (a) Examples of recovered images under $\gamma = 25/64$ and (b) RSNR on 6000 8×8 image patches.

2.1 dB on average. Though RG algorithm is more effective, it involves binary programming (22), and is unscalable to applications with high-dimensional signal. To conclude, heuristic algorithm is efficient for applications of both low/high-dimensional signals, and RG algorithm provides the best performance for low-dimension signal applications.

D. Hardware Performance Evaluation

In this part, the hardware performance benefits of Boolean embedding will be investigated in details. The evaluation only focuses on the embedding hardware as indicated by red dash-lined boxes in Figs. 3 and 5.

a) Hardware comparison: The matrix-vector multiplier is composed of multiple multiplier-accumulator (MAC) in parallel. To multiply the signal vector with a 19×256 sampling matrix, 19 MACs are needed and each MAC requires 256 cycles to perform the inner-product with each cycle (1 ns). To store the NuMax real-valued sampling matrix, 16 kB SRAM with 64-bit I/O bus-width is used.

The proposed Boolean optimization quantizes NuMax sampling matrix into a $\{-1, 1\}^{m \times n}$ Boolean matrix. The size of SRAM to store sampling matrix is therefore reduced from 16 kB to 1 kB. Compared to a Bernoulli $\{0, 1\}^{m \times n}$ matrix, a $\{-1, 1\}^{m \times n}$ multiplication requires calculations of 2's complement of input signal vector, which incurs additional hardware cost for MACs. To minimize the overhead of 2's complement, the MAC design in Fig. 4(c) is used, which calculates 2's complement only once every 256 cycles.

TABLE III
HARDWARE PERFORMANCE COMPARISON AMONG DIFFERENT SAMPLING MATRICES (19×256) ON VARIED HARDWARE CONFIGURATIONS

Matrix type	Hardware configuration	Energy (nJ)	Leakage power (μW)	Area (μm^2)	Cycle
Real-valued	MAC (16-bit) MEM (16kB)	116.38 8.08	119.63 4.66	127984 31550	256
-1/1 Boolean	MAC (1-bit) MEM (1kB)	24.81 2.43	69.22 0.29	73207 9800	256
0/1 Bernoulli	MAC (1-bit) PRNG	21.87 8.26e-2	30.40 0.04	29165 32	$\sim 512^{\dagger}$
Boolean	RRAM crossbar	1.06	-	173	1

†Pseudo-random number generator (PRNG) used produces 10 bits per cycle.

RRAM crossbar supports both $\{0, 1\}^{m \times n}$ and $\{-1, 1\}^{m \times n}$ Boolean matrices. As the sampling matrix is embedded into the RRAM crossbar which also performs the matrix multiplication, no separate memory is required.

The performance of four hardware schemes that support different types of sampling matrices is compared in Table III. Compared to the NuMax real-valued embedding on 16-bit MAC and 16 kB SRAM hardware, the proposed quantized $-1/1$ Boolean embedding on 1-bit MAC and 1 kB SRAM consumes $4.6\times$ less operation energy per embedding, $1.8\times$ smaller leakage power, and $1.9\times$ smaller area. This is because, as mentioned in Section III, the real-valued multiplier generally requires quadratically increasing number of full-adders when resolution increases, while Boolean multiplier only needs linearly more full-adders.

When the proposed quantized $-1/1$ Boolean embedding is performed on RRAM crossbar, it further improves the hardware performance significantly. Specifically, for the operation energy per embedding, the RRAM crossbar based embedding outperforms the CMOS circuit based real-valued embedding by $117\times$. The area of the RRAM crossbar based embedding is nearly $1000\times$ better than that of CMOS circuit based real-valued embedding. In addition, the RRAM crossbar will not experience the leakage power which is at the scale of hundreds of μW for the CMOS circuit based approach. For the operation speed, the RRAM crossbar embedding executes in single cycle while the CMOS circuit requires 256 cycles due to the reuse of hardware. The overall performance for different sampling matrices on varied hardware platforms is summarized in Table IV.

b) Impact of RRAM variation: One non-negligible issue of mapping Boolean embedding matrix to RRAM crossbar is the RRAM RHS and LHS variations. With high resistance variation, the embedding matrix will deviate from expected values to be represented by RRAM resistance, and hence the recovery quality may degrade. The sensitivity study of recovery quality on the resistance variations of RRAM is shown in Fig. 10. The resistance of RRAM is assumed to follow log-normal distribution with the mean to be R_{LRS} and R_{HRS}, and standard deviation σ_{LRS} and σ_{HRS} for LRS and HRS cells respectively.

With varied σ_{LRS} and σ_{HRS}, it can be observed from Fig. 10 that the performance degradation is more susceptible to resistance variation of LRS, while less sensitive on variation of RHS.

TABLE IV
COMPARISON OF ALL VALID SAMPLING MATRIX AND HARDWARE COMBINATIONS

Platform		Boolean?	Optimized?	Data driven?	Construction		Recovery quality‡	Energy consumed⋆
Sampling matrix	Sampling hardware				Stage	Effort		
Gaussian	MAC(16-bit) + MEM(16kB)	×	×	×	off-line	Immediate	1.12	117.4
Bernoulli	MAC(1-bit) + MEM(1kB)	✓	×	×	off-line	Immediate	1.09	22.9
Bernoulli	RRAM crossbar	✓	×	×	off-line	Immediate	1.09	**1.0**
Pseudo-Bernoulli	MAC(1-bit) + PRNG	✓	×	×	runtime	-	1.00	**20.7**
NuMax [12]	MAC(16-bit) + MEM(16kB)	×	✓	✓	off-line	∼100s	**3.86**	117.4
Reed-Muller [11][10]	MAC(1-bit) + MEM(1kB)	✓	✓	×	off-line	Fast	1.25	22.9
Reed-Muller [11][10]	RRAM crossbar	✓	✓	×	off-line	Fast	1.25	22.9
Proposed†	MAC(1-bit) + MEM(1kB)	✓	✓	✓	off-line	∼100s	**3.18**	25.7
Proposed	RRAM crossbar	✓	✓	✓	off-line	∼100s	**3.18**	**1.0**

‡ the recovery quality depicts the quality performance ratio of all sampling matrices over baseline pseudo-Bernoulli. The RSNR dB for ECG is converted to mean-squared error. Numbers in bold are ones with good performance.
⋆the energy consumption is shown as ratio of used energy of all sampling matrices over that of RRAM crossbar. Numbers in bold are ones with good performance.
†the **proposed** denotes the Booleanized NuMax sampling matrix by proposed Algorithms.

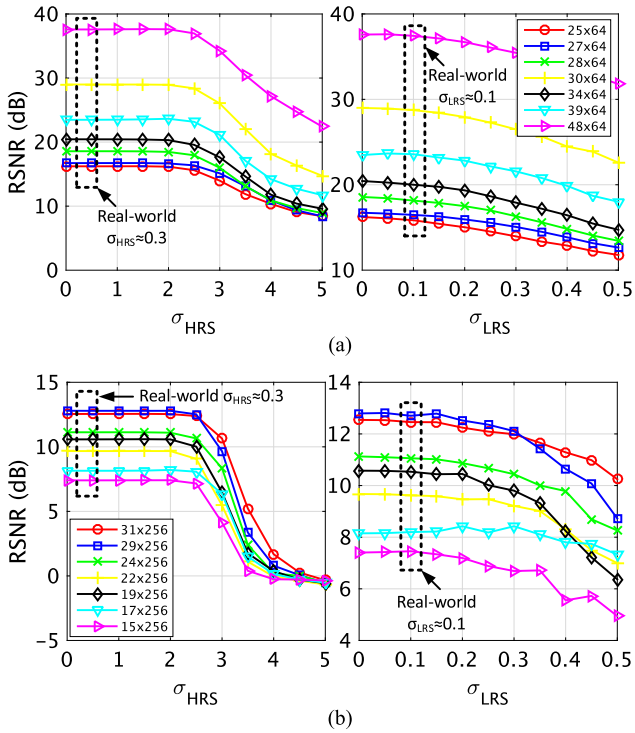


Fig. 10. The sensitivity of recovery quality of (a) image signal and (b) ECG signal on the resistance standard deviation σ of RRAM for both low resistance state (LRS) and high resistance state (HRS) following Log-normal distribution.

In practice, the RHS variation σ_{HRS} is approximately 0.3 [27], and LHS variation σ_{LRS} roughly 0.1 [27]. The real-world σ is annotated in Fig. 10 and it can be concluded that the proposed Boolean embedding on RRAM crossbar is robust against RRAM device variations when on/off ratio is high ($G_{\mathrm{LRS}} \gg G_{\mathrm{HRS}} \approx 0$). To further suppress the performance degradation, material engineering [28] and verification programming method [27] can help achieve higher LHS uniformity.

VII. CONCLUSION

In this work, towards energy efficient and high performance hardware implementation of data acquisition by compressive sensing, a novel embedding algorithm is proposed to transform a given optimized real-valued embedding matrix into an optimized Boolean embedding matrix under (near-) orthogonal rotations. As such, the embedding not only can be effectively mapped to both CMOS circuit and RRAM crossbar with much lower power consumption, also high performance of optimized real-valued embedding can be well preserved. Numerical results show that, in terms of signal acquisition quality, the proposed data-driven optimized Boolean sampling matrix outperforms the random Bernoulli matrix by 2.9× and 3.2× with RRAM crossbar and CMOS MAC circuits with similar energy efficiency, respectively. Compared to real-valued data-driven sampling matrix, the proposed Boolean sampling can achieve 117× and 4.6× better energy efficiency on RRAM crossbar and CMOS MAC implementations with similar signal quality, respectively. Overall, the proposed data-driven Boolean sampling matrix combines both the high performance advantage of real-valued sampling and low power advantage of Boolean sampling.

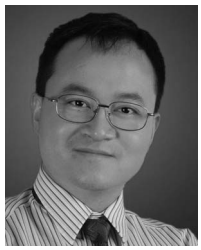
REFERENCES

[1] F. Ren and D. Markovic, "A configurable 12-to-237 ks/s 12.8 mw sparse-approximation engine for mobile ExG data aggregation," in *Proc. IEEE Int. Solid-State Circuits Conf.*, San Francisco, CA, USA, Feb. 2015, pp. 1–3.
[2] Z. Zhang *et al.*, "Compressed sensing of eeg for wireless telemonitoring with low energy consumption and inexpensive hardware," *IEEE Trans. Biomed. Eng.*, vol. 60, no. 1, pp. 221–224, Jan. 2013.
[3] A. Dixon *et al.*, "Compressed sensing system considerations for ecg and emg wireless biosensors," *IEEE Trans. Biomed. Circuits Syst.*, vol. 6, no. 2, pp. 156–166, Apr. 2012.
[4] Y. Suo *et al.*, "Energy-efficient multi-mode compressed sensing system for implantable neural recordings," *IEEE Trans. Biomed. Circuits Syst.*, vol. 8, no. 5, pp. 648–659, Oct. 2014.
[5] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.
[6] J. Pant and S. Krishnan, "Compressive sensing of electrocardiogram signals by promoting sparsity on the second-order difference and by using dictionary learning," *IEEE Trans. Biomed. Circuits Syst.*, vol. 8, no. 2, pp. 293–302, Apr. 2014.
[7] F. Ren and D. Markovic, "A configurable 12–237 ks/s 12.8 mw sparse-approximation engine for mobile data aggregation of compressively sampled physiological signals," *IEEE J. Solid-State Circuits*, vol. 51, no. 1, pp. 68–78, Jan. 2016.
[8] D. Malioutov and M. Malyutov, "Boolean compressed sensing: LP relaxation for group testing," in *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, Kyoto, Japan, Mar. 2012, pp. 3305–3308.
[9] M. Fatemi and M. Vetterli, "Randomized recovery for boolean compressed sensing," in *Proc. IEEE Int. Symp. Inform. Theory*, Istanbul, Turkey, Jul. 2013, pp. 469–473.
[10] R. Calderbank and S. Jafarpour, "Reed Muller sensing matrices and the lasso," in *Sequences and Their Applications SETA 2010*, C. Carlet and A. Pott, Eds. Berlin, Germany: Springer, 2010, pp. 442–463.

[11] S. Jafarpour, "Deterministic Compressed Sensing," dissertation, Dept. Comput. Sci., Princeton Univ., Princeton, NJ, USA, Aug. 2011.
[12] C. Hegde *et al.*, "Numax: A convex approach for learning near-isometric linear embeddings," *IEEE Trans. Signal Process.*, vol. 63, no. 22, pp. 6109–6121, Nov. 2015.
[13] F. Chen, A. Chandrakasan, and V. Stojanovic, "Design and analysis of a hardware-efficient compressed sensing architecture for data compression in wireless sensors," *IEEE J. Solid-State Circuits*, vol. 47, no. 3, pp. 744–756, Mar. 2012.
[14] D. B. Strukov *et al.*, "The missing memristor found," *Nature*, vol. 453, no. 7191, pp. 80–83, May 2008.
[15] S. Kim and Y.-K. Choi, "Resistive switching of aluminum oxide for flexible memory," *Appl. Phys. Lett.*, vol. 92, no. 22, pp. 223 508.1–223 508.3, Jun. 2008.
[16] Y. Wang, H. Yu, and W. Zhang, "Nonvolatile cbram-crossbar-based 3-D-integrated hybrid memory for data retention," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 22, no. 5, pp. 957–970, May 2014.
[17] Y. Wang *et al.*, "Optimizing boolean embedding matrix for compressive sensing in RRAM crossbar," in *Proc. ACM/IEEE Int. Symp. Low Power Electron. Design*, Rome, Italy, Jul. 2015, pp. 13–18.
[18] R. Baraniuk *et al.*, "A simple proof of the restricted isometry property for random matrices," *Constr. Approx.*, vol. 28, no. 3, pp. 253–263, Dec. 2008.
[19] H. Lee *et al.*, "Low power and high speed bipolar switching with a thin reactive ti buffer layer in robust HfO2 based RRAM," in *Proc. IEEE Int. Electron Devices Meeting*, San Francisco, CA, USA, Dec. 2008, pp. 1–4.
[20] S.-S. Sheu *et al.*, "A 5 ns fast write multi-level non-volatile 1 k bits RRAM memory with advance write scheme," in *Proc. Symp. VLSI Circuits*, Kyoto, Japan, 2009. pp. 82–83.
[21] L. Costa and P. Oliveira, "Evolutionary algorithms approach to the solution of mixed integer non-linear programming problems," *Comput. Chem. Eng.*, vol. 25, no. 2, pp. 257–266, Mar. 2001.
[22] M. ApS, The MOSEK optimization toolbox for MATLAB manual. Version 7.1 (Revision 28), 2015. [Online]. Available: http://docs.mosek.com/7.1/toolbox/index.html
[23] N. Sahinidis, "Baron: A general purpose global optimization software package," *J. Global Optim.*, vol. 8, no. 2, pp. 201–205, Mar. 1996.
[24] G. B. Huang *et al.*, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," Univ. Massachusetts, Amherst, MA, Tech. Rep. 07-49, Oct. 2007.
[25] G. B. Moody and R. G. Mark, "The impact of the mit-bih arrhythmia database," *IEEE Eng. Med. Biol.*, vol. 20, no. 3, pp. 45–50, May–Jun. 2001.
[26] S. J. Wilton and N. P. Jouppi, "Cacti: An enhanced cache access and cycle time model," *IEEE J. Solid-State Circuits*, vol. 31, no. 5, pp. 677–688, May 1996.
[27] Y. Chen *et al.*, "Highly scalable hafnium oxide memory with improvements of resistive distribution and read disturb immunity," in *Proc. IEEE Int. Electron Devices Meeting*, Baltimore, MD, USA, Dec. 2009, pp. 1–4.
[28] H.-S. Wong *et al.*, "Metal-oxide RRAM," *Proc. IEEE*, vol. 100, no. 6, pp. 1951–1970, May 2012.

Yuhao Wang received the B.S. degree in microelectronics engineering from Xi'an Jiao Tong University, Xi'an, China, and the Ph.D. degree from the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, in 2011 and 2015, respectively.

Currently, he is a Senior R&D Engineer at Synopsys, Mountain View, CA, USA. His research interests include EDA topics related to emerging nonvolatile memory design flow and hardware optimization with emphasis on energy efficiency.



Xin Li (S'01–M'06–SM'10) received the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, USA, in 2005.

Currently, he is an Associate Professor in the Department of Electrical and Computer Engineering, Carnegie Mellon University. His research interests include integrated circuit and signal processing.

Dr. Li was a recipient of the National Science Foundation Faculty Early Career Development Award in 2012, the IEEE Donald O. Pederson Best Paper Award in 2013, the Best Paper Award from Design Automation Conference in 2010, two IEEE/ACM William J. McCalla International Conference on Computer-Aided Design Best Paper Awards in 2004 and 2011, and the Best Paper Award from the International Symposium on Integrated Circuits in 2014.

Kai Xu was born in Taian, China, in 1990. He received the B.S. degree in electrical engineering from Shandong University, Shandong, China, in 2011, and the M.S. degree in electrical engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2014.

Currently, he is working toward the Ph.D. degree in computer engineering at Arizona State University, Tempe, AZ, USA. His research interests include machine learning and, convex optimization for designing low power framework in Internet of Things (IoTs).

Fengbo Ren (S'10–M'15) received the B.Eng. degree from Zhejiang University, Hangzhou, China, in 2008, and the M.S. and Ph.D. degrees from the University of California, Los Angeles, Los Angeles, CA, USA, in 2010 and 2014, respectively, all in electrical engineering.

In 2015, he joined the faculty of the School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, AZ, USA. His doctoral research involved designing energy-efficient VLSI systems, accelerating compressive sensing signal reconstruction, and developing emerging memory technology. His research interests are focused on hardware acceleration and parallel computing solutions for data analytics and information processing, with emphasis on compressive sensing, sparse coding, and deep learning frameworks.

Dr. Ren received the 2012–2013 Broadcom Fellowship.

Hao Yu (M'06–SM'14) received the B.S. degree from Fudan University, Shanghai, China, in 1999, and the M.S. and Ph. D degrees in integrated circuit and embedded computing from the Electrical Engineering Department, University of California, Los Angeles, Los Angeles, CA, USA, in 2007.

He was a Senior Research Staff at Berkeley Design Automation. Since October 2009, he has been an Assistant Professor in the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. His primary research interest is in emerging CMOS technologies such as 3DIC and RFIC designs at nanotera scale. He has 195 top-tier peer-reviewed publications, five books, and six book chapters.

Dr. Yu received the Best Paper Award from the ACM Transactions on Design Automation of Electronic Systems in 2010, Best Paper Award nominations at DAC'06, ICCAD'06, and ASP-DAC'12, and the Inventor Award from the Semiconductor Research Cooperation. He is an associate editor and technical program committee member of several journals and conferences.