Low-discrepancy sequences (aka Quasi Monte Carlo methods)

UCSD CSE 272
Advanced Image Synthesis

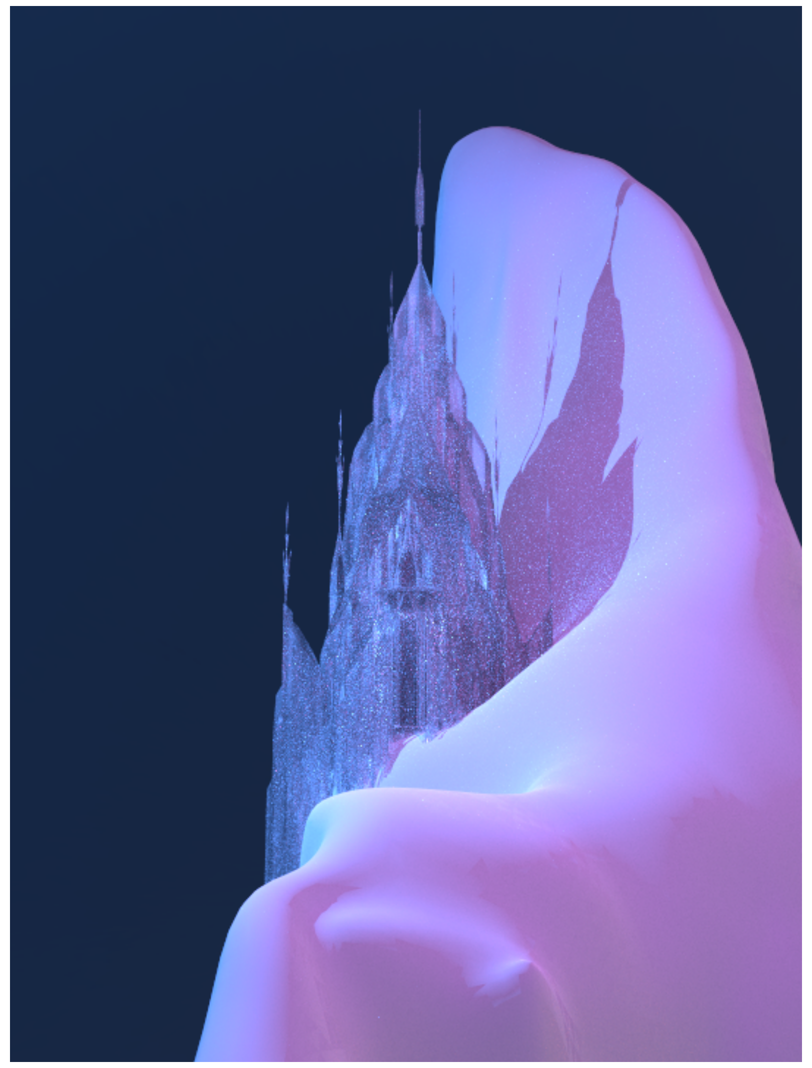
Tzu-Mao Li

slides are largely inspired by Wojciech Jarosz and Lois Paulin

HWI graded

Haolin Lu

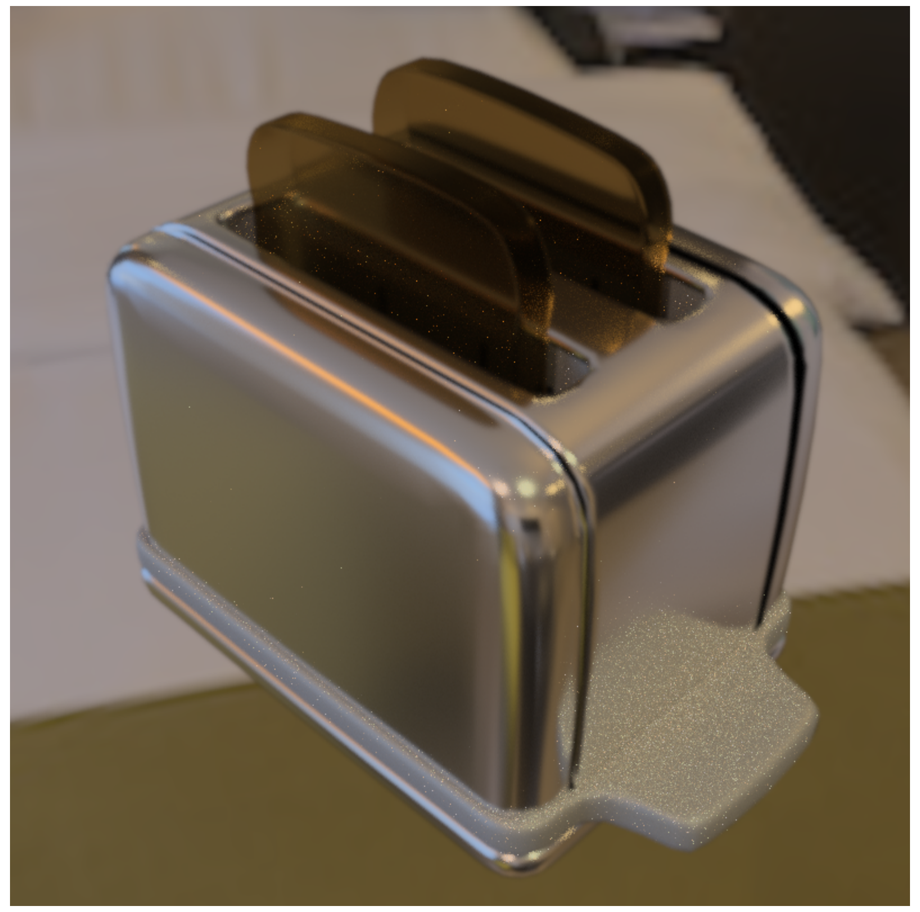
some cool images



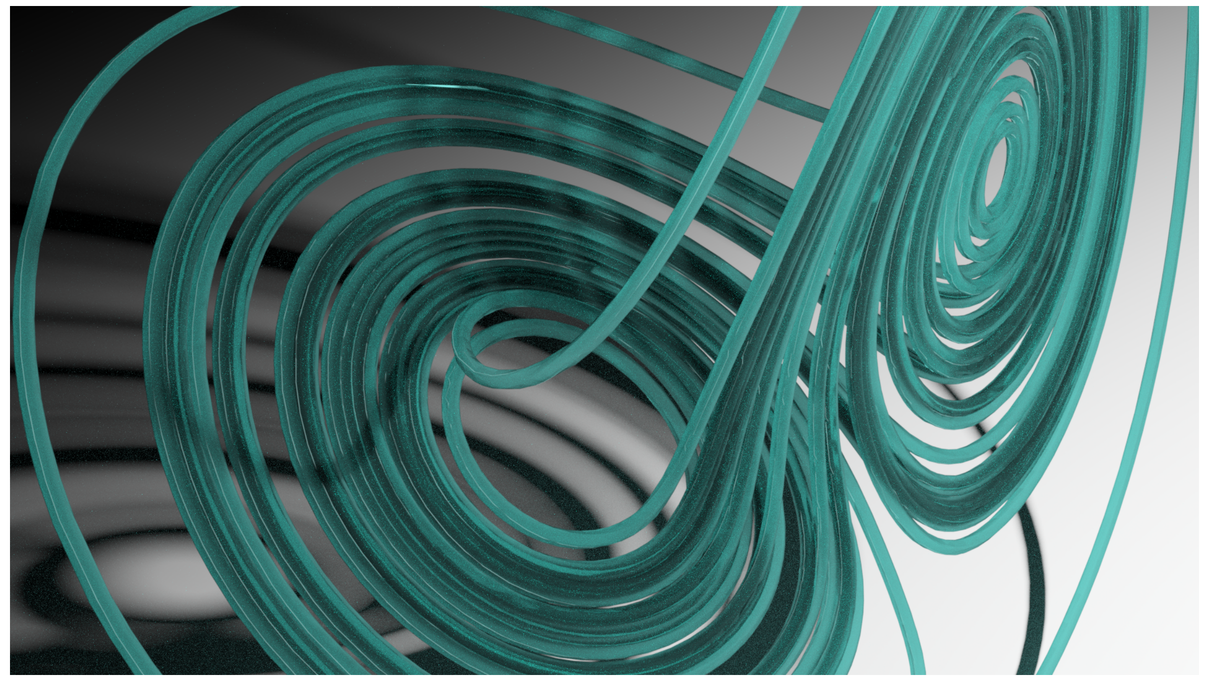
Peter Wu

Keli Wang

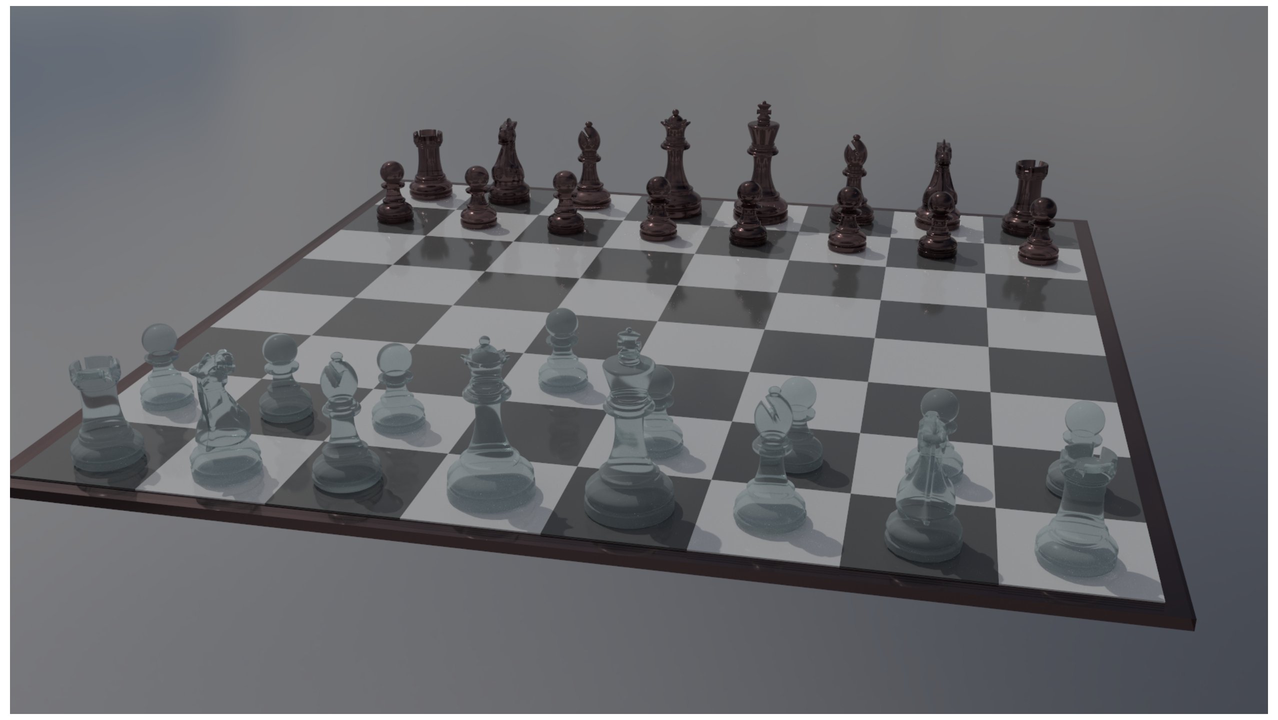
Issac Nealey



Alexander Mai



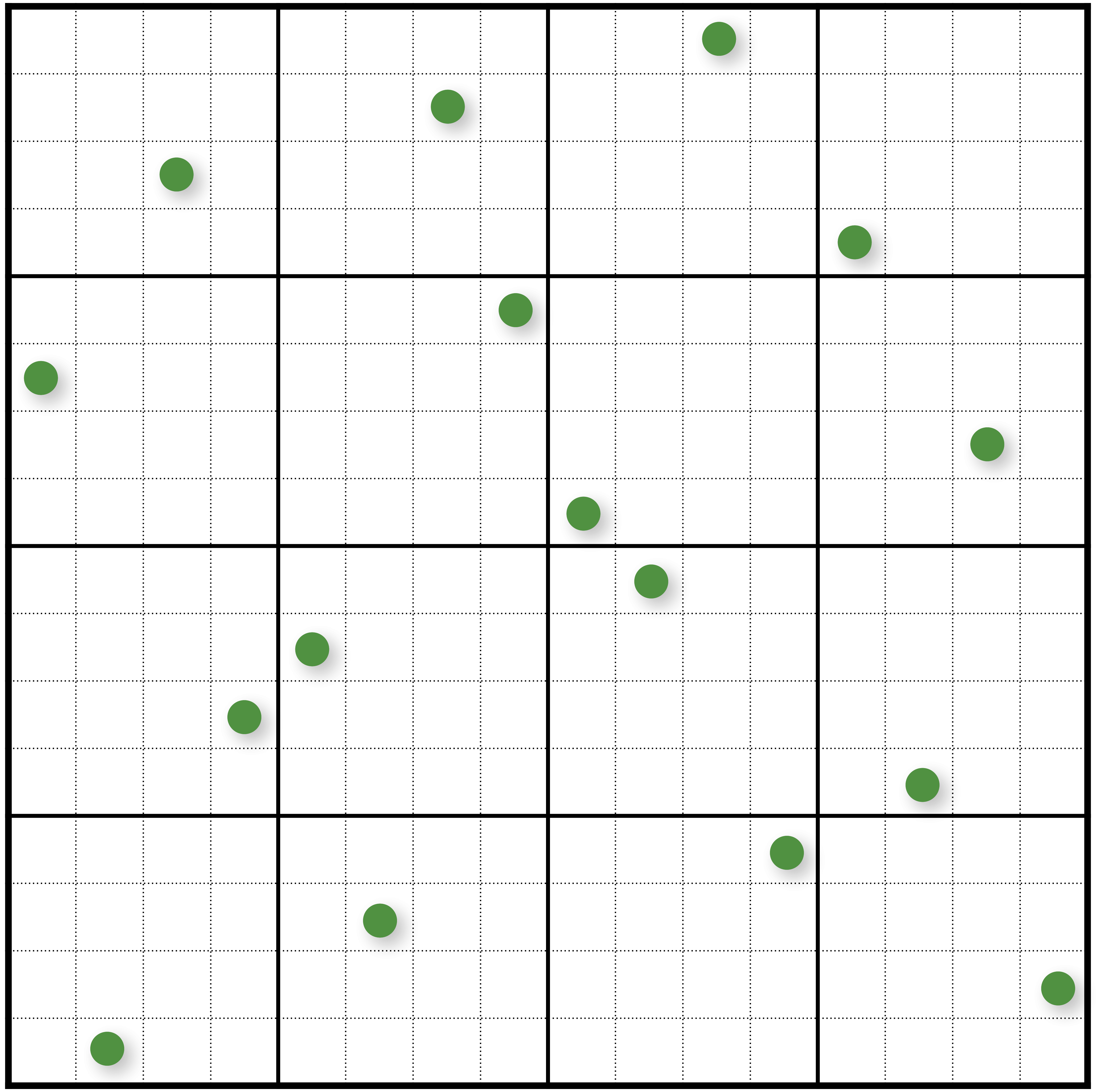
Zimu Guan



Pasha Bouzarjomehri

Multi-jittered sampling

- one sample in each square
- one sample in each row
- one sample in each column

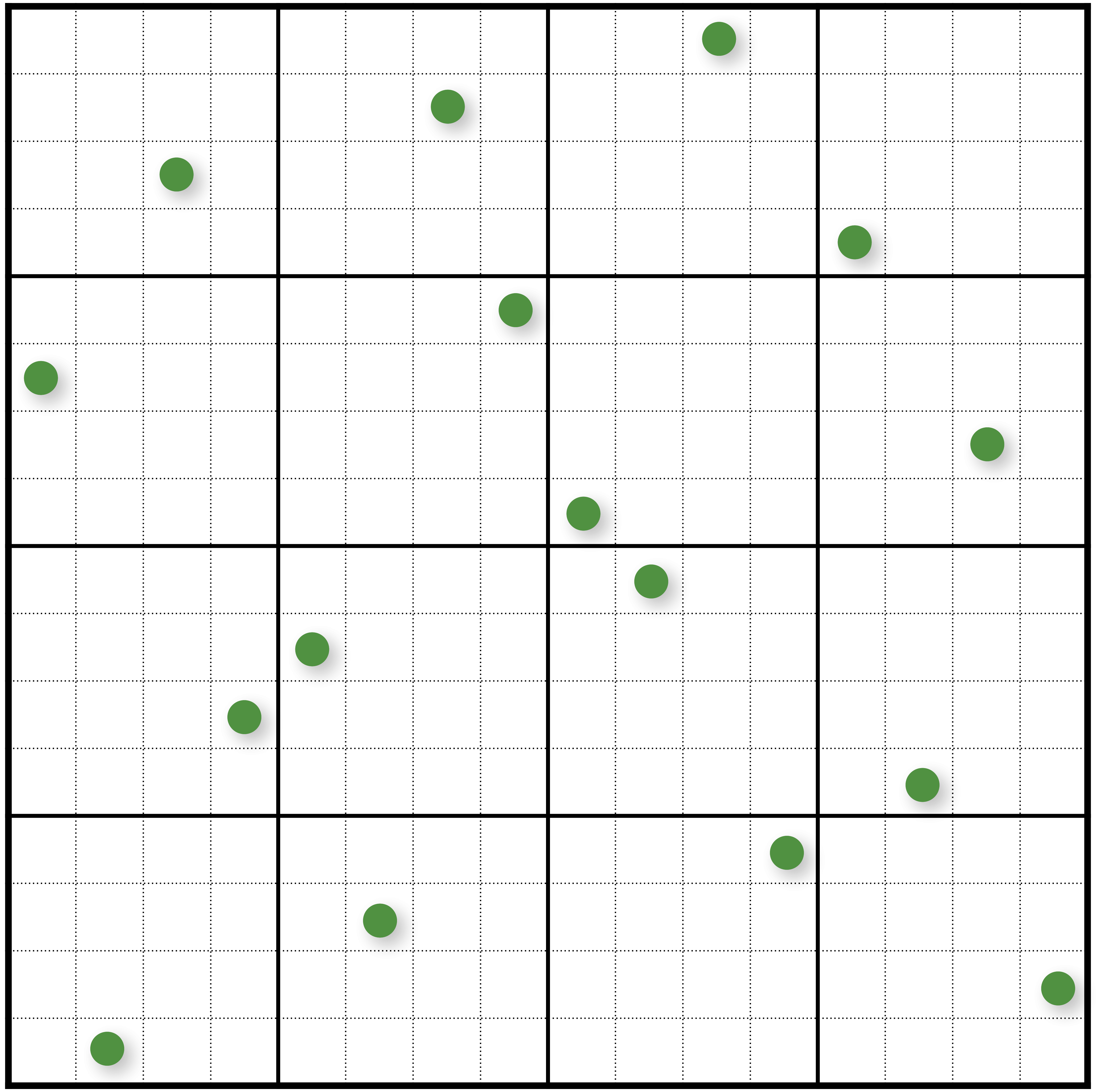


https://cs.dartmouth.edu/~wjarosz/publications/subr16fourier-slides-2-patterns.pdf

Multi-jittered sampling

- one sample in each square
- one sample in each row
- one sample in each column

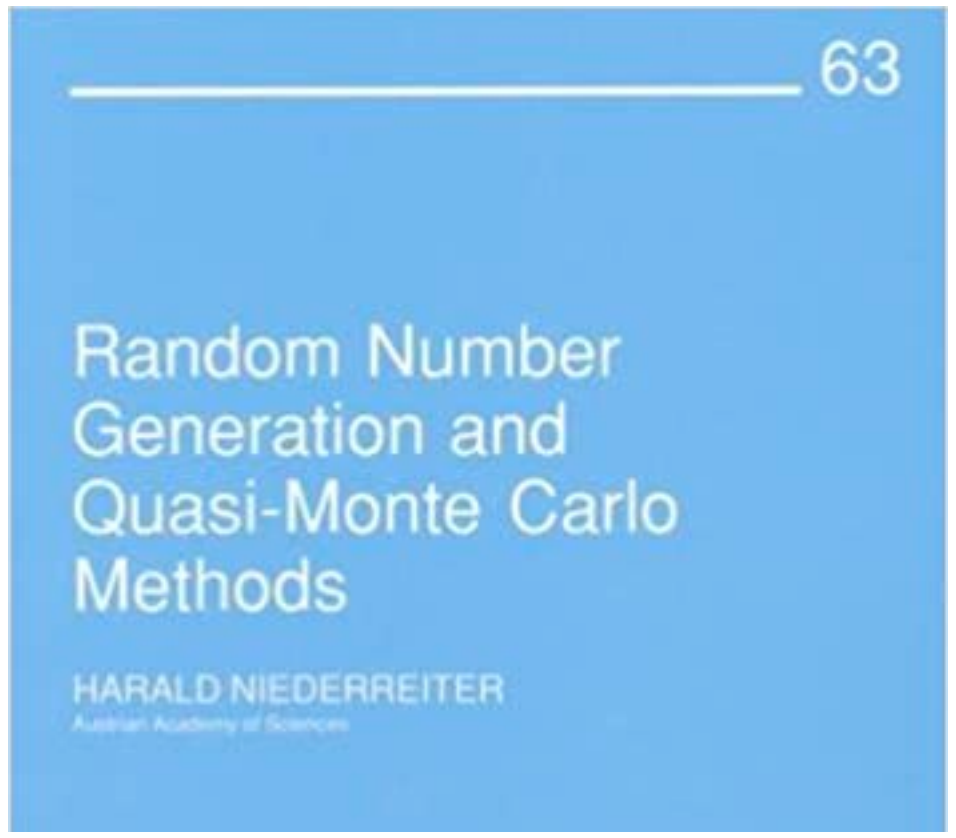
quiz: can we add more constraints to the point set?

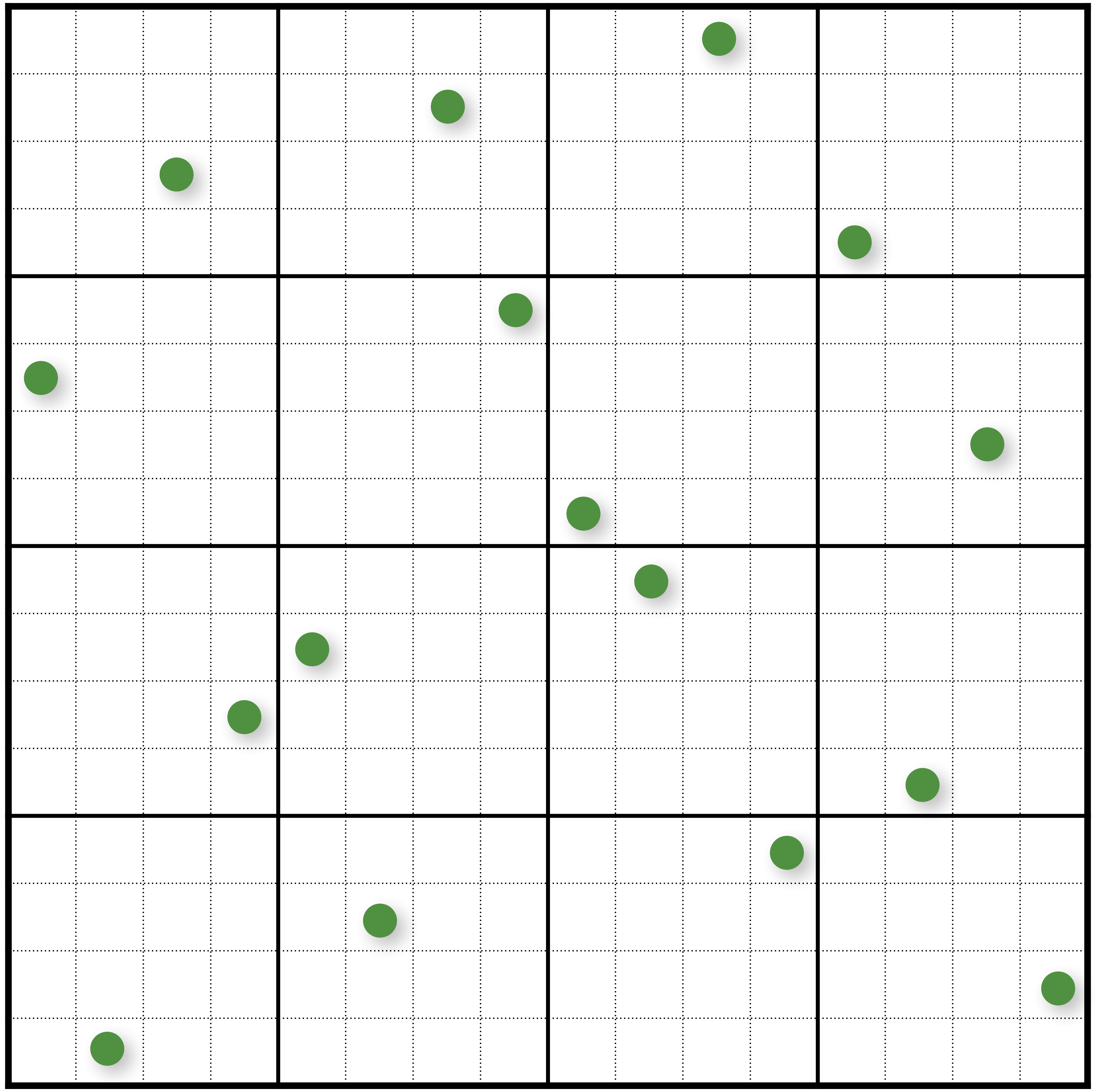


https://cs.dartmouth.edu/~wjarosz/publications/subr16fourier-slides-2-patterns.pdf

Digital nets (aka (t,m,s) nets)

- one sample in all rectangular partition of the space



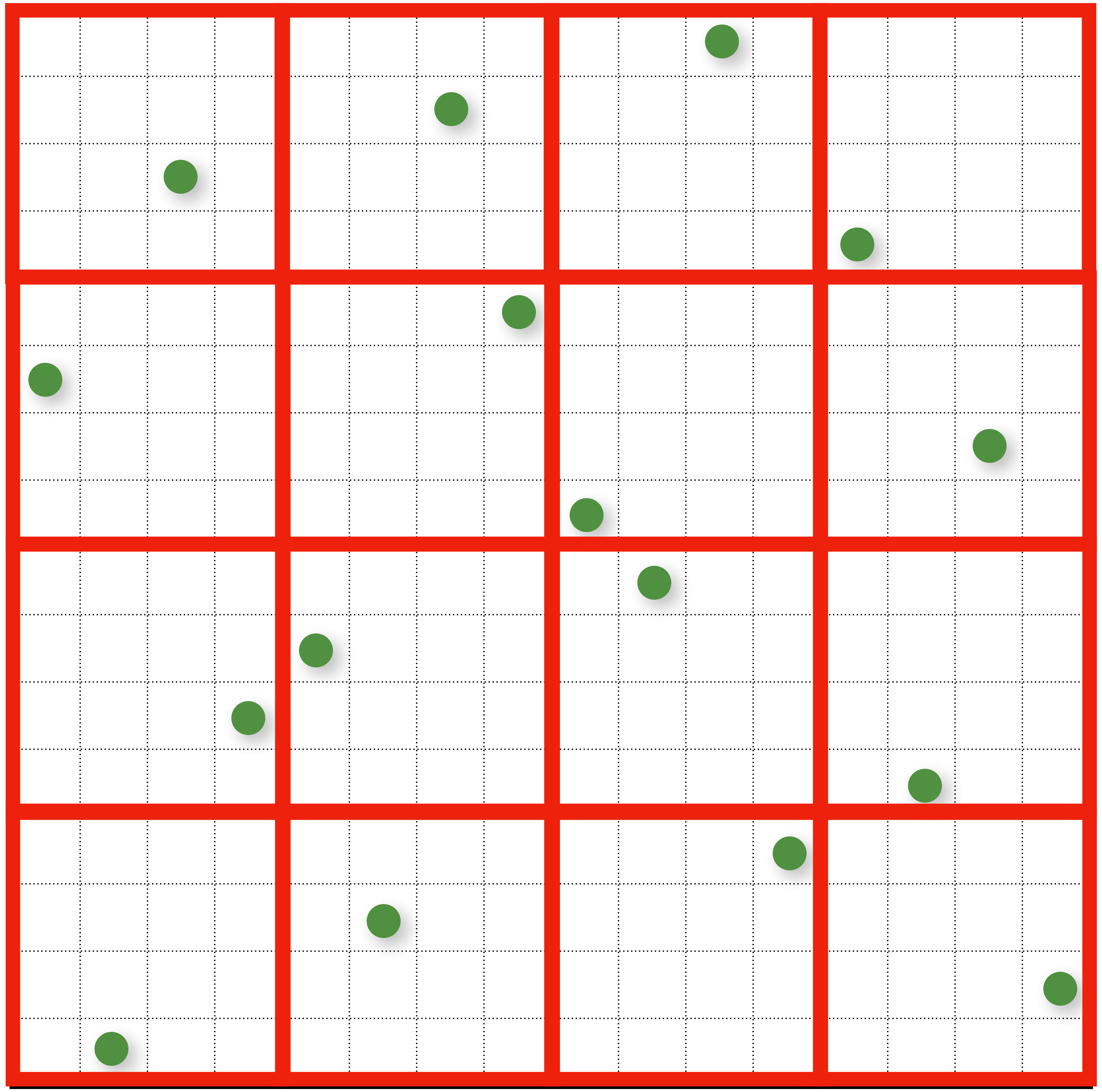


https://cs.dartmouth.edu/~wjarosz/publications/subr16fourier-slides-2-patterns.pdf

Digital nets (aka (t,m,s) nets)

- one sample in all rectangular partition of the space

partition of 16 4x4 rectangles

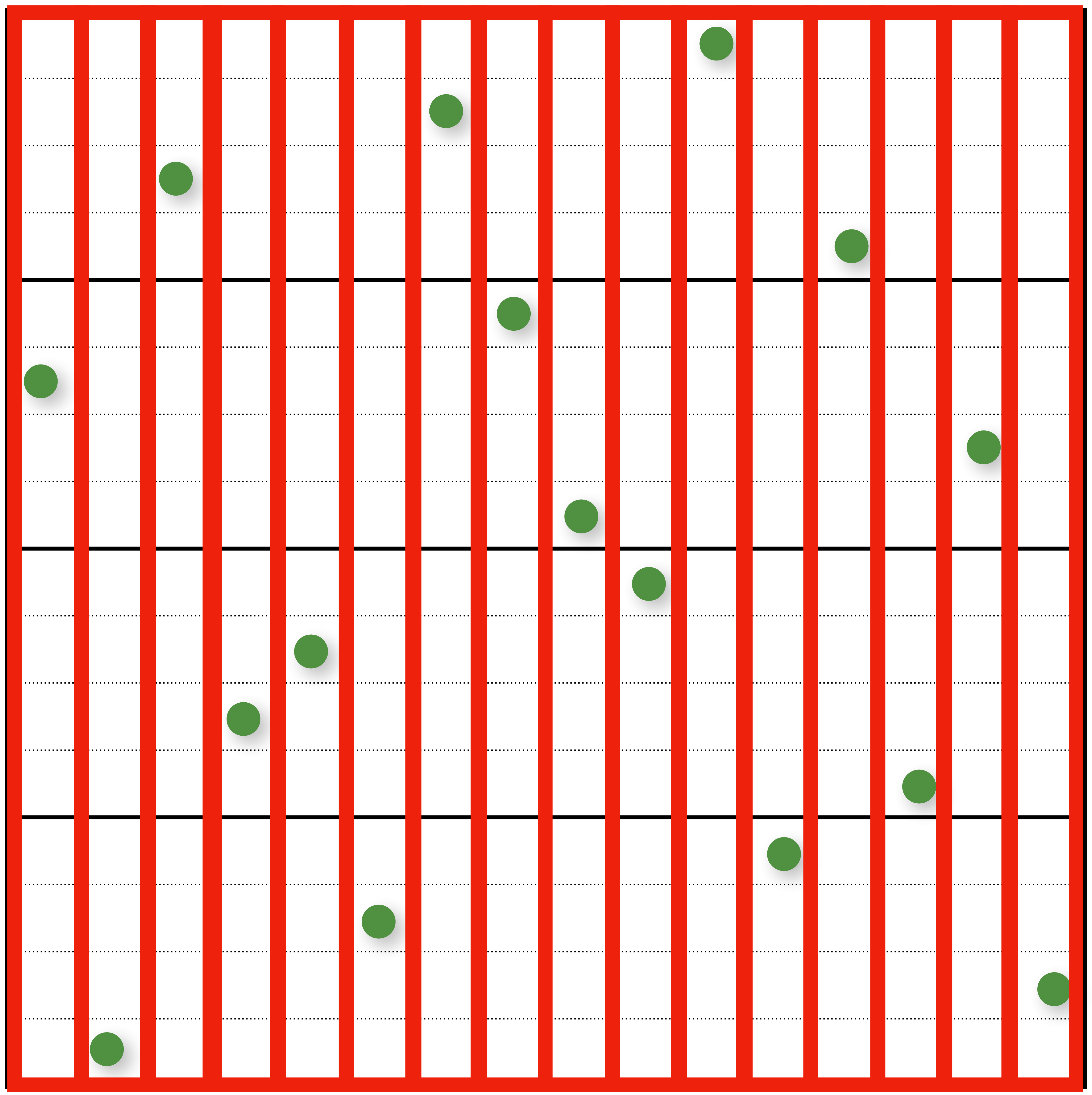


https://cs.dartmouth.edu/~wjarosz/publications/subr16fourier-slides-2-patterns.pdf

Digital nets (aka (t,m,s) nets)

- one sample in all rectangular partition of the space

partition of 16 16x1 rectangles

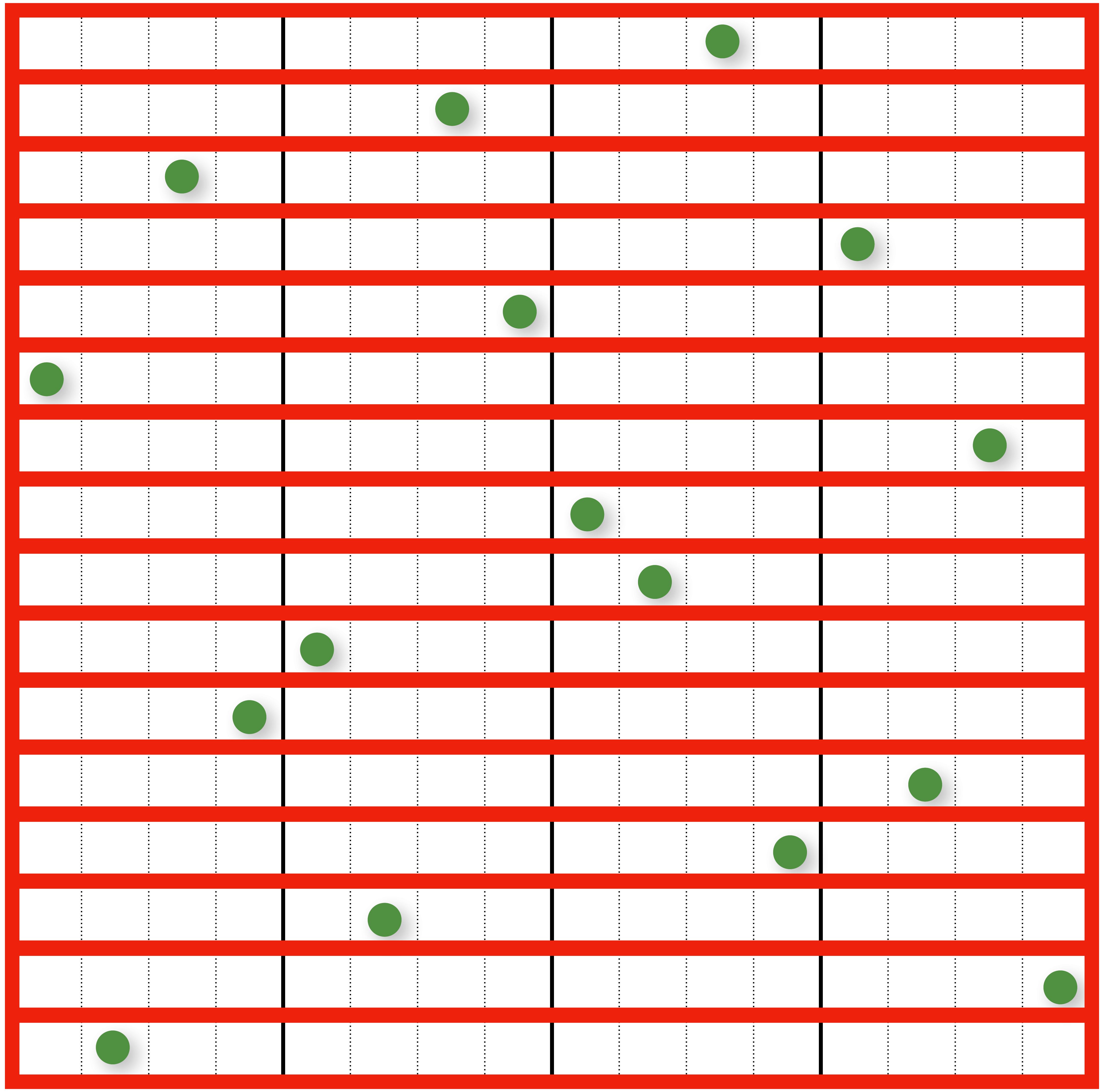


https://cs.dartmouth.edu/~wjarosz/publications/subr16fourier-slides-2-patterns.pdf

Digital nets (aka (t,m,s) nets)

- one sample in all rectangular partition of the space

partition of 16 1x16 rectangles

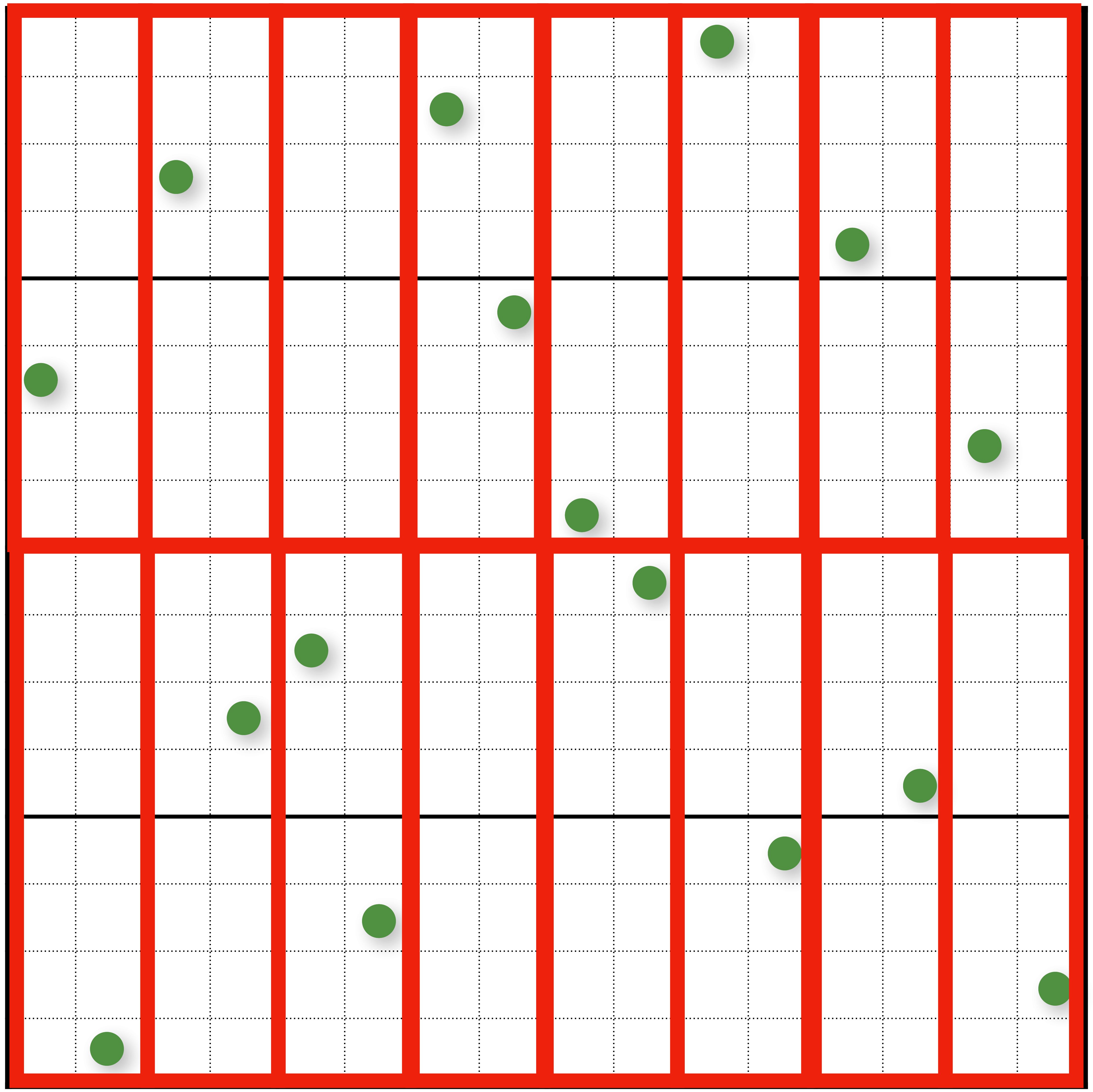


https://cs.dartmouth.edu/~wjarosz/publications/subr16fourier-slides-2-patterns.pdf

Digital nets (aka (t,m,s) nets)

- one sample in all rectangular partition of the space

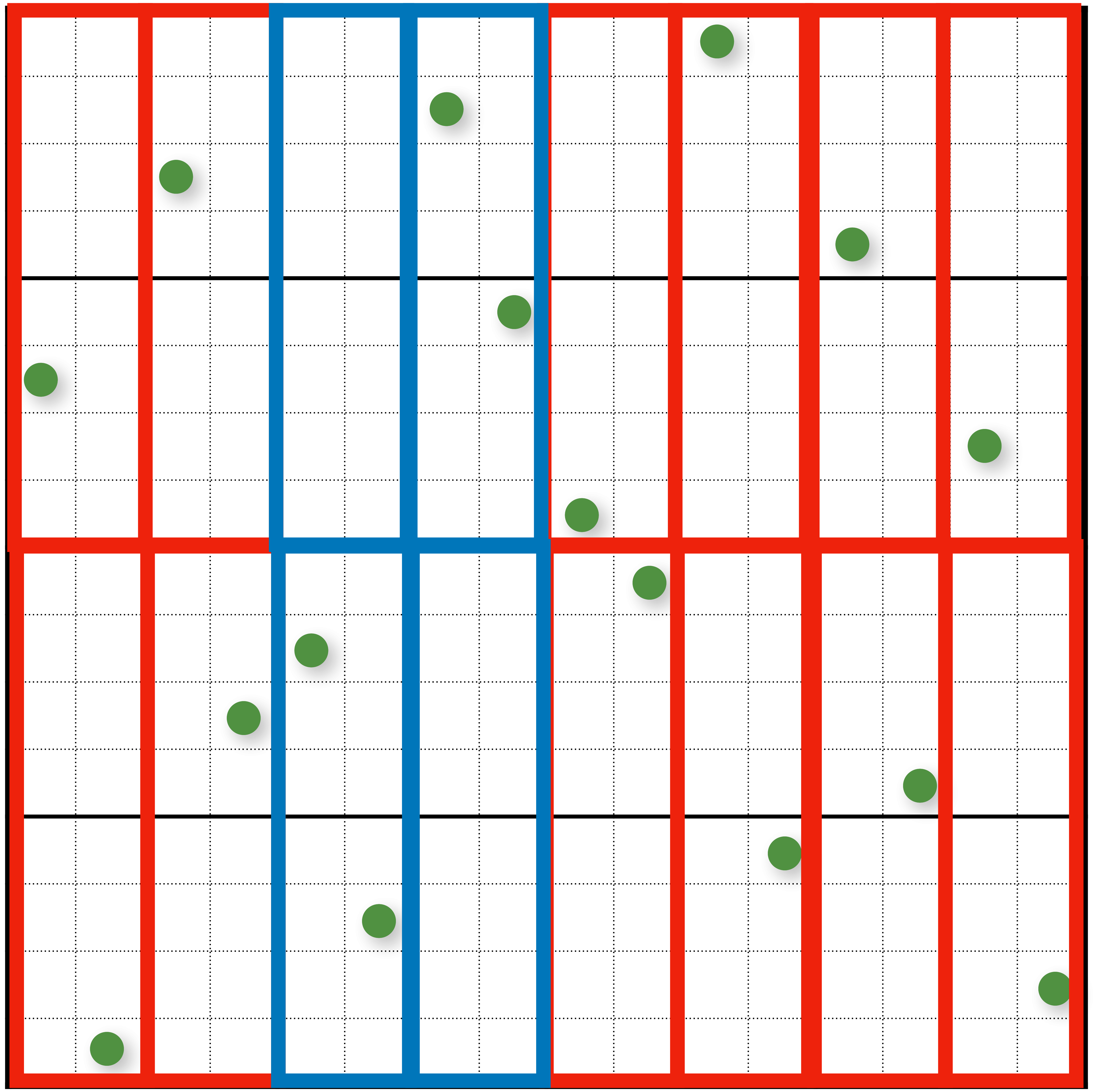
partition of 16 8x2 rectangles



https://cs.dartmouth.edu/~wjarosz/publications/subr16fourier-slides-2-patterns.pdf

Digital nets (aka (t,m,s) nets)

- one sample in all rectangular partition of the space



partition of 16 8x2 rectangles

multi-jittered sampling fails to satisfy the digital nets property!

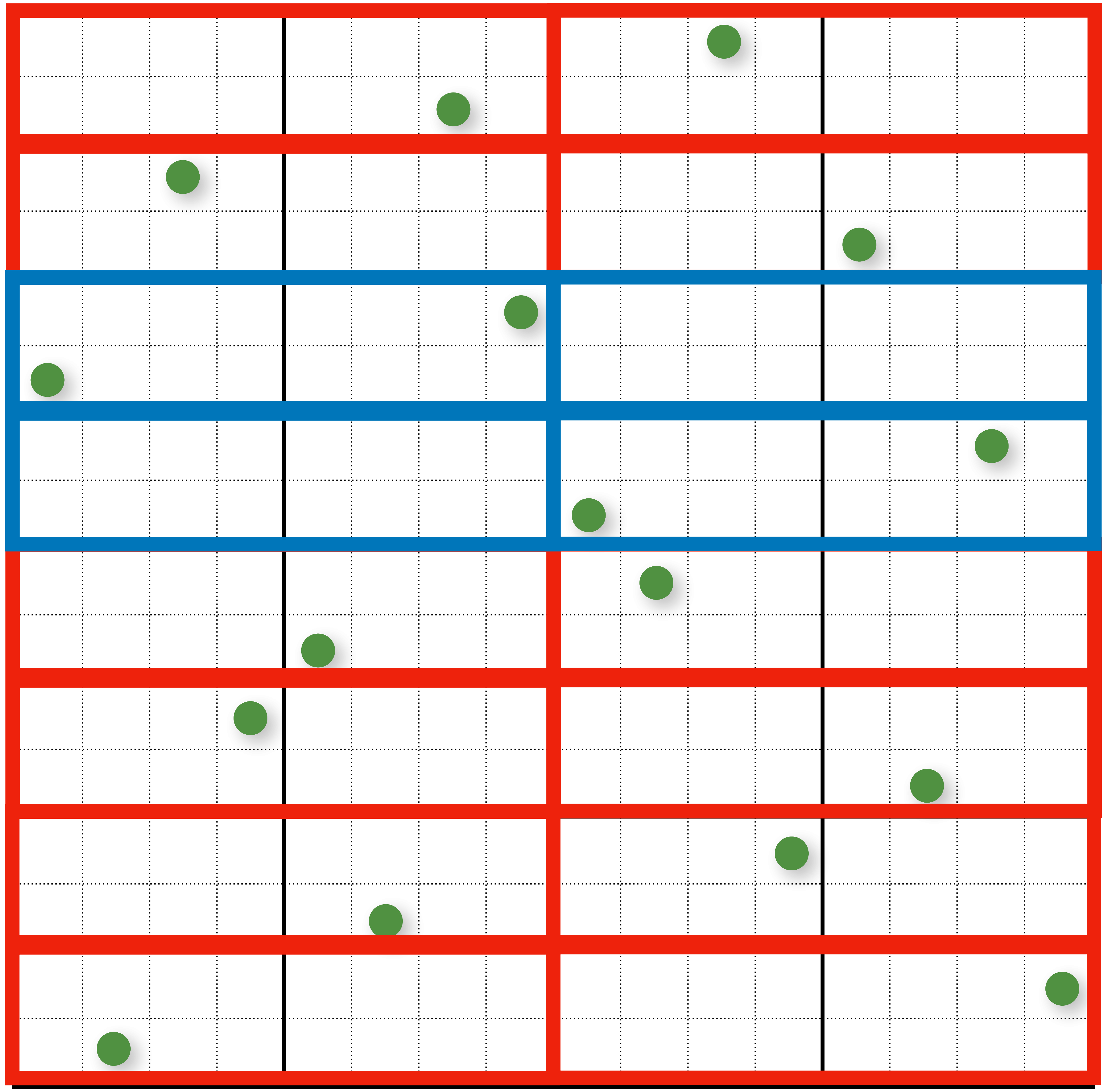
https://cs.dartmouth.edu/~wjarosz/publications/subr16fourier-slides-2-patterns.pdf

Digital nets (aka (t,m,s) nets)

- one sample in all rectangular partition of the space

partition of 16 2x8 rectangles

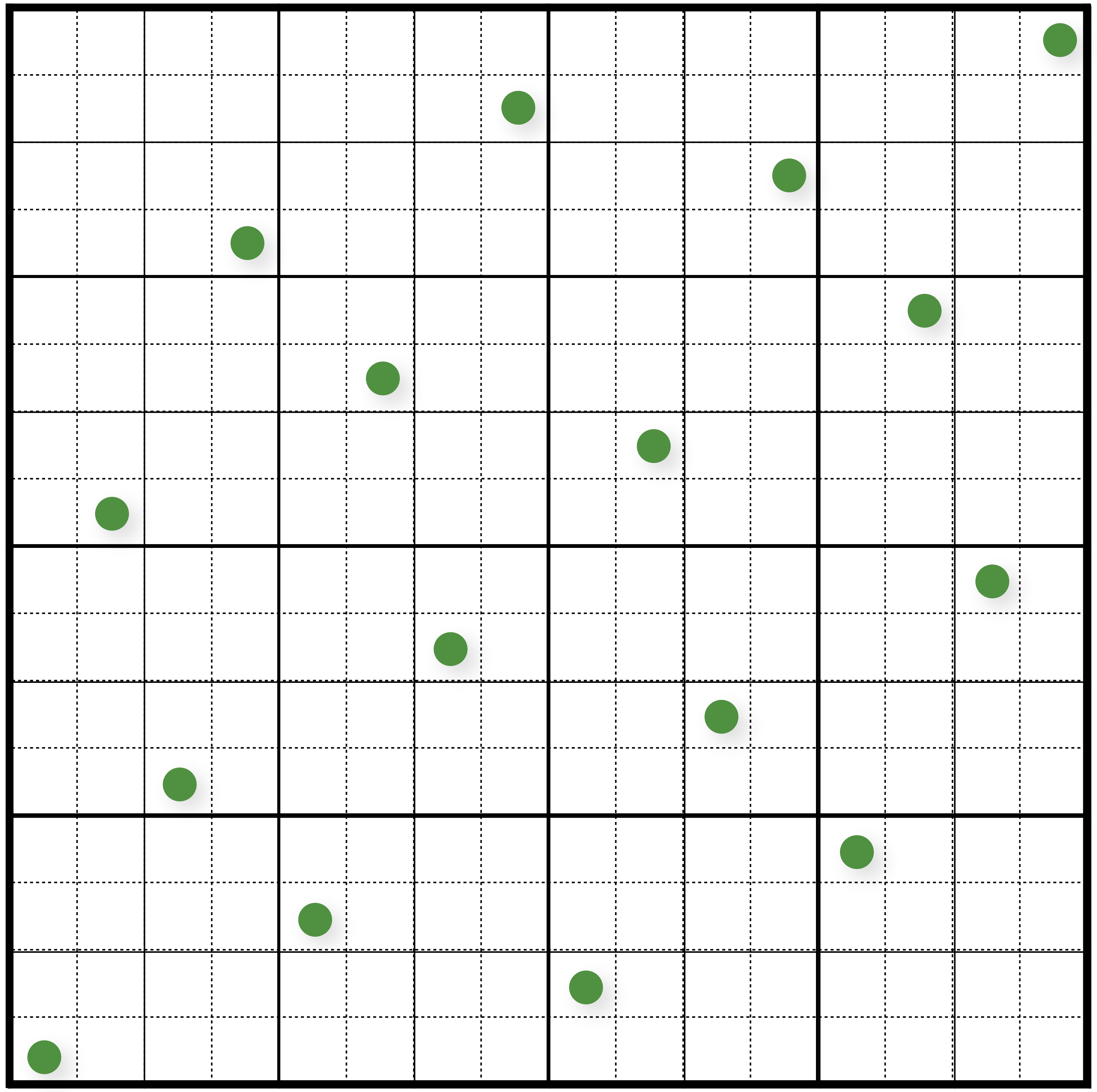
multi-jittered sampling fails to satisfy the digital nets property!



https://cs.dartmouth.edu/~wjarosz/publications/subr16fourier-slides-2-patterns.pdf

A point set that satisfies the digital net property

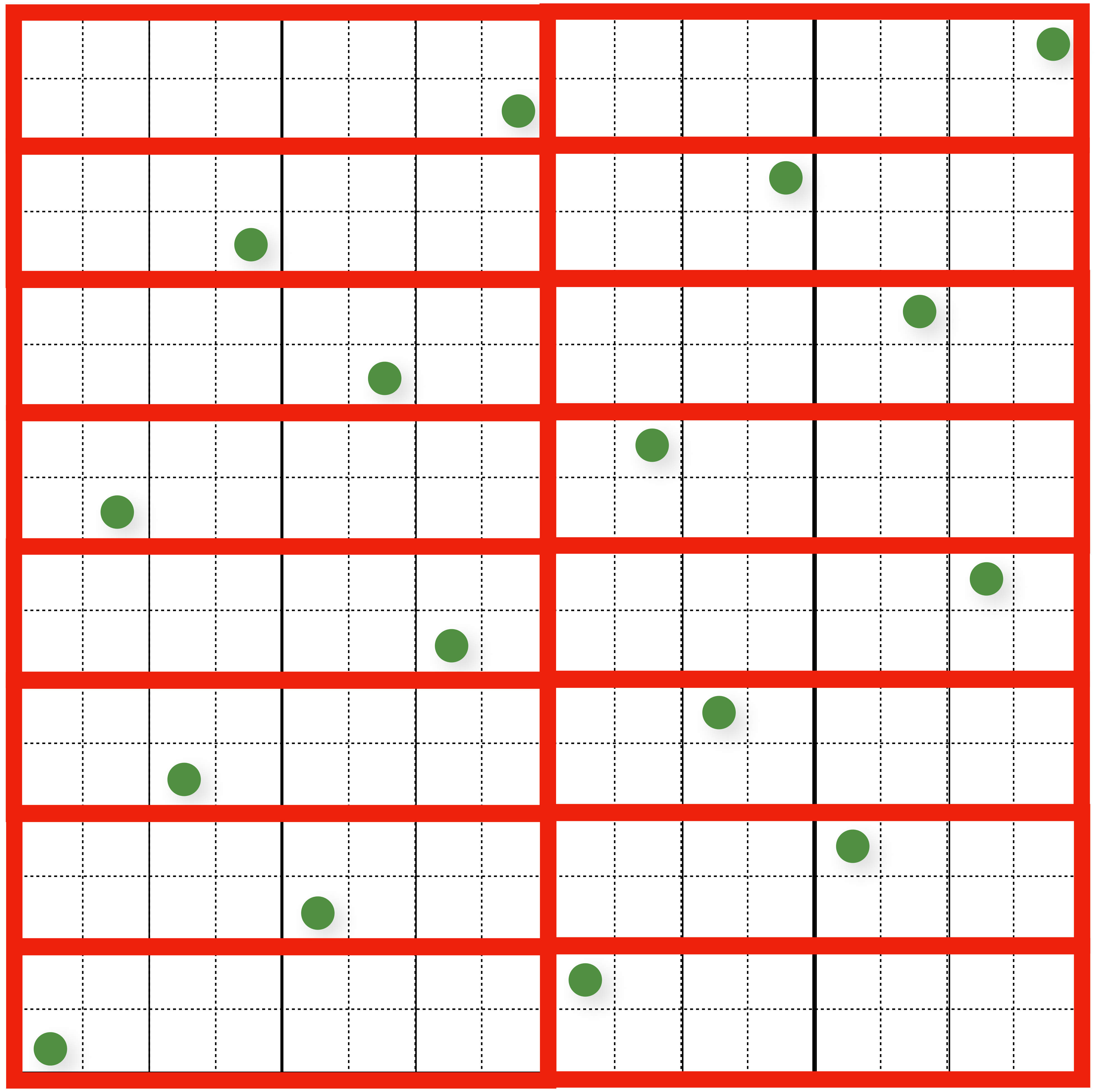
- one sample in all rectangular partition of the space



https://cs.dartmouth.edu/~wjarosz/publications/subr16fourier-slides-2-patterns.pdf

A point set that satisfies the digital net property

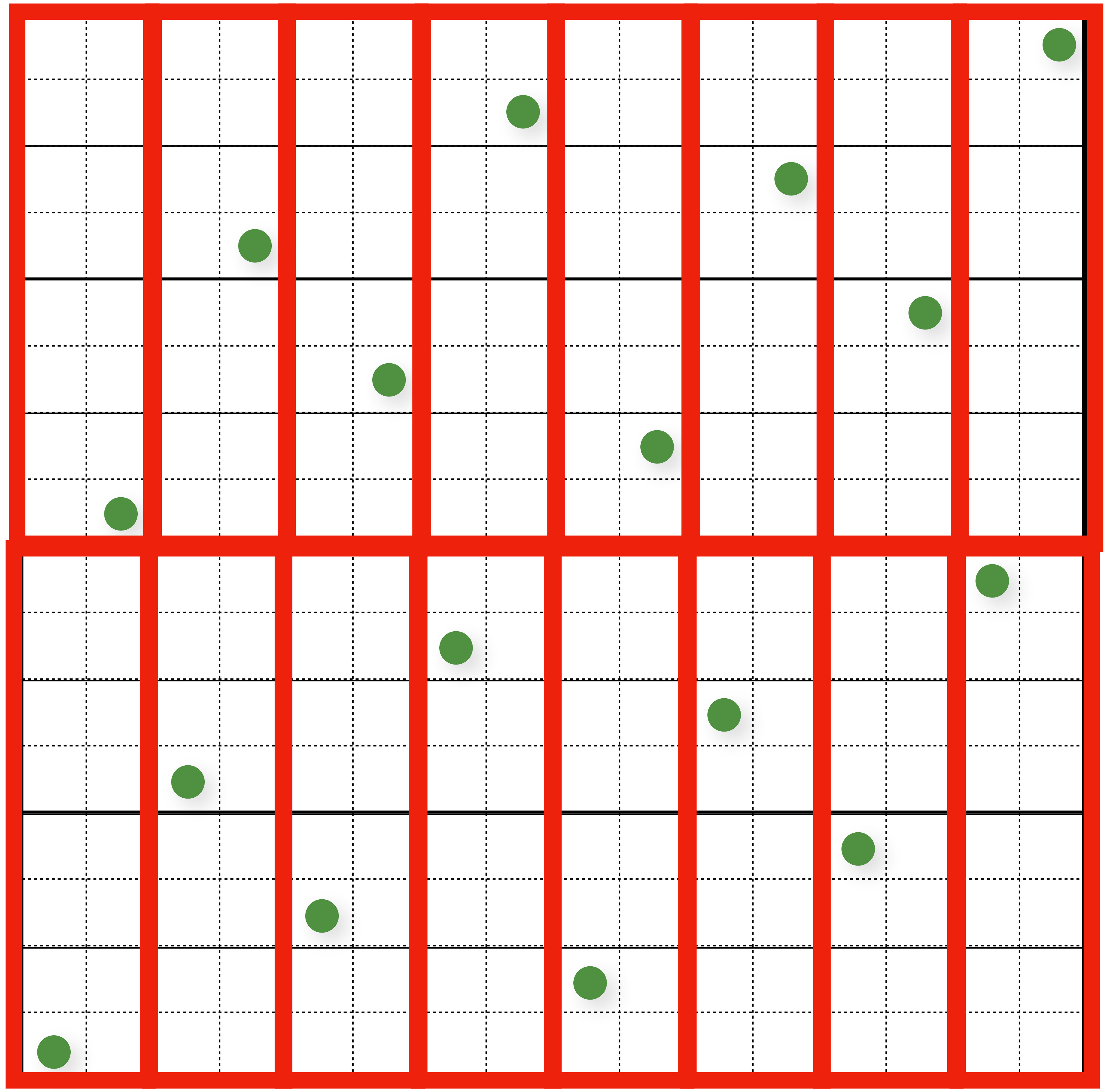
- one sample in all rectangular partition of the space



https://cs.dartmouth.edu/~wjarosz/publications/subr16fourier-slides-2-patterns.pdf

A point set that satisfies the digital net property

- one sample in all rectangular partition of the space



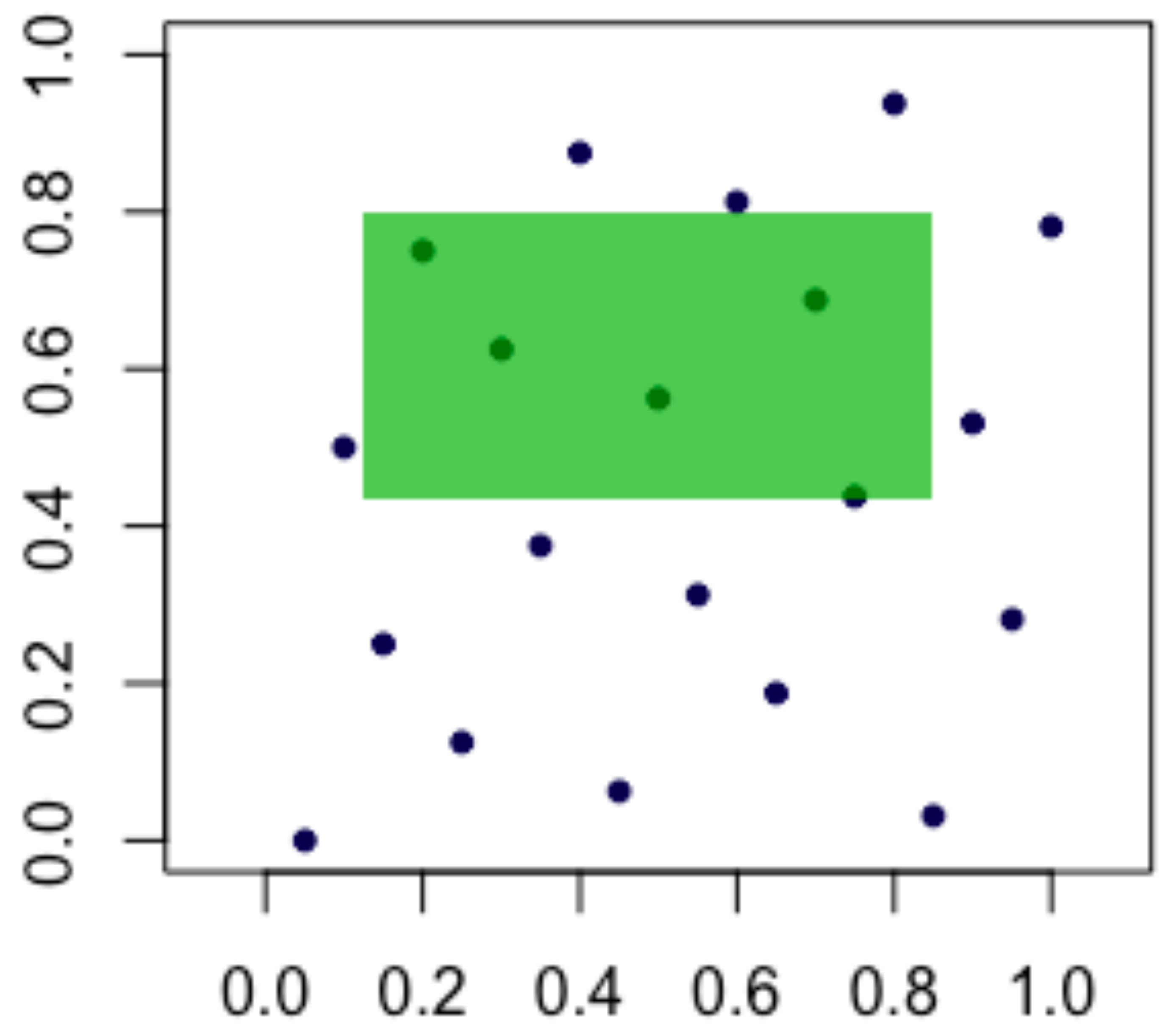
https://cs.dartmouth.edu/~wjarosz/publications/subr16fourier-slides-2-patterns.pdf

Low-discrepancy sequences

- deterministic & progressive Latin hypercube samples based on the minimization of **discrepancy**

- entire field of study called “Quasi-Monte-Carlo”

$$D_n = \max_{\text{all rectangles}} \left| \frac{\text{no. of points in the rectangle}}{n} - \text{area(rectangle)} \right|$$



https://stats.stackexchange.com/questions/40384/fake-uniform-random-numbers-more-evenly-distributed-than-true-uniform-data

Koksma-Hlawka inequality

- discrepancy is the upper bound of the absolute estimation error!

V: the "total variation" of f

$$\left| \frac{1}{n} \sum_{i=0}^{n} f(x_i) - \int f(x) \mathrm{d}x \right| \leq V(f) D_n^*(x_1, x_2, \ldots, x_n)$$

star discrepancy: only consider rectangles with one vertex at the origin

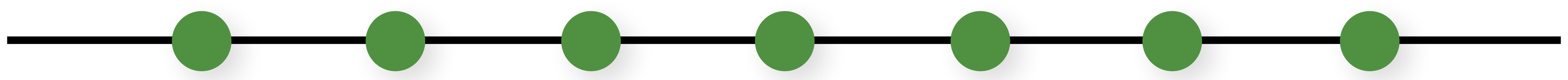
$$D_n = \max_{\text{all rectangles}} \left| \frac{\text{no. of points in the rectangle}}{n} - \text{area(rectangle)} \right|$$

The van der Corput sequence Φ_b

the simplest low-discrepancy sequence in 1D

define a sequence for a base b

k	Base 2	Φ_b
1	1	$.1_2 = 1/2$
2	10	$.01_2 = 1/4$
3	11	$.11_2 = 3/4$
4	100	$.001_2 = 1/8$
5	101	$.101_2 = 5/8$
6	110	$.011_2 = 3/8$
7	111	$.111_2 = 7/8$
...		



https://cs.dartmouth.edu/~wjarosz/publications/subr16fourier-slides-2-patterns.pdf

The van der Corput sequence Φ_b

the simplest low-discrepancy sequence in 1D

define a sequence for a base b

k	Base 3	Φ_b
1	1	$.1_3 = 1/3$
2	2	$.2_3 = 2/3$
3	10	$.01_3 = 1/9$
4	11	$.11_3 = 4/9$
5	12	$.21_3 = 7/9$
6	20	$.02_3 = 2/9$
7	21	$.12_3 = 5/9$
...		

The van der Corput sequence Φ_b

subdivide the 1D space
into b regions

k	Base 3	Φ_b
1	1	$.1_3 = 1/3$
2	2	$.2_3 = 2/3$
3	10	$.01_3 = 1/9$
4	11	$.11_3 = 4/9$
5	12	$.21_3 = 7/9$
6	20	$.02_3 = 2/9$
7	21	$.12_3 = 5/9$
...		

The van der Corput sequence Φ_b

subdivide the 1D space into b regions

sample the boundaries

k	Base 3	Φ_b
1	1	$.1_3 = 1/3$
2	2	$.2_3 = 2/3$
3	10	$.01_3 = 1/9$
4	11	$.11_3 = 4/9$
5	12	$.21_3 = 7/9$
6	20	$.02_3 = 2/9$
7	21	$.12_3 = 5/9$
...		

The van der Corput sequence Φ_b

subdivide the 1D space into b regions

sample the boundaries

recurse into each region

k	Base 3	Φ_b
1	1	$.1_3 = 1/3$
2	2	$.2_3 = 2/3$
3	10	$.01_3 = 1/9$
4	11	$.11_3 = 4/9$
5	12	$.21_3 = 7/9$
6	20	$.02_3 = 2/9$
7	21	$.12_3 = 5/9$
...		

The van der Corput sequence Φ_b

subdivide the 1D space into b regions

sample the boundaries

recurse into each region

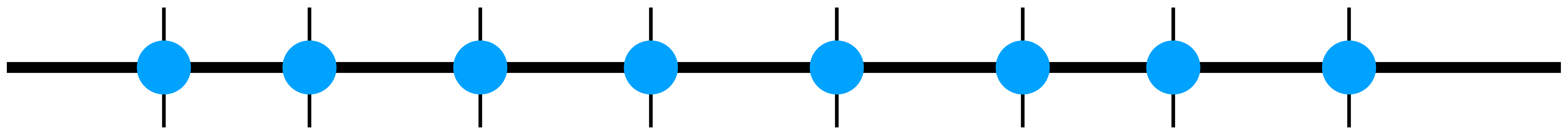
k	Base 3	Φ_b
1	1	$.1_3 = 1/3$
2	2	$.2_3 = 2/3$
3	10	$.01_3 = 1/9$
4	11	$.11_3 = 4/9$
5	12	$.21_3 = 7/9$
6	20	$.02_3 = 2/9$
7	21	$.12_3 = 5/9$
…		

The van der Corput sequence Φ_b

subdivide the 1D space into b regions

sample the boundaries

recurse into each region



k	Base 3	Φ_b
1	1	$.1_3 = 1/3$
2	2	$.2_3 = 2/3$
3	10	$.01_3 = 1/9$
4	11	$.11_3 = 4/9$
5	12	$.21_3 = 7/9$
6	20	$.02_3 = 2/9$
7	21	$.12_3 = 5/9$
...		

The van der Corput sequence Φ_b

subdivide the 1D space into b regions

sample the boundaries

recurse into each region

k	Base 10	Φ_b
1	1	$.1_{10} = 1/10$
5	5	$.5_{10} = 5/10$
9	9	$.9_{10} = 9/10$
10	10	$.01_{10} = 1/100$
11	11	$.11_{10} = 11/100$
12	12	$.21_{10} = 21/100$
21	21	$.12_{10} = 12/100$
...		

High-dimensional generalization of van der Corput sequence: Halton sequence

$$\mathrm{Halton}(k) = \left(\Phi_2(k), \Phi_3(k), \Phi_5(k), \cdots\right)$$

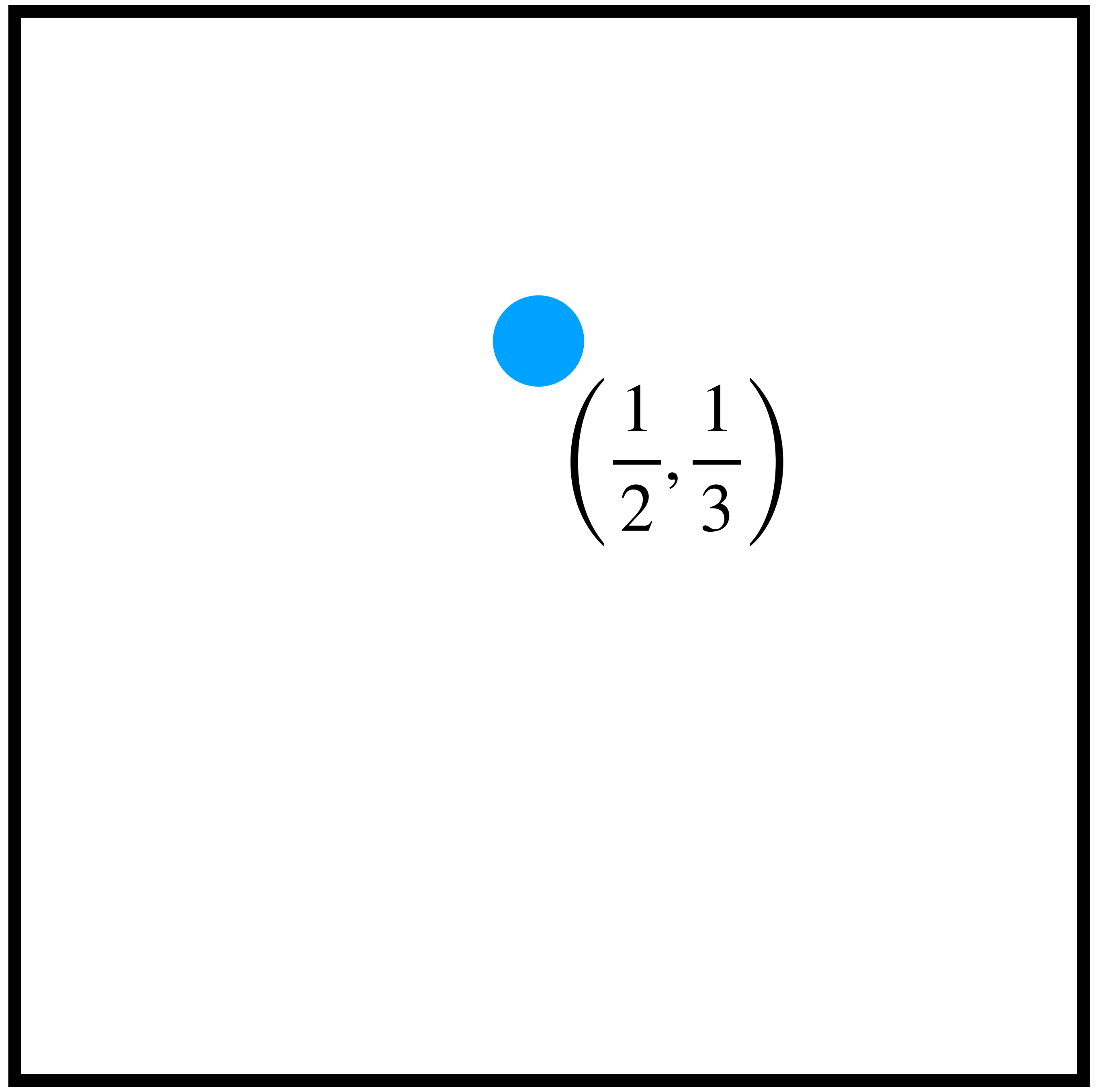
concatenate van der Corput sequences with co-prime bases into a vector

ALGORITHM 247
RADICAL-INVERSE QUASI-RANDOM POINT SEQUENCE [G5]
J. H. HALTON AND G. B. SMITH (Recd. 24 Jan. 1964 and 21 July 1964)
Brookhaven National Laboratory, Upton, N. Y., and University of Colorado, Boulder, Colo.

High-dimensional generalization of van der Corput sequence: Halton sequence

$$\text{Halton}(k) = \left(\Phi_2(k), \Phi_3(k), \Phi_5(k), \cdots\right)$$

concatenate van der Corput sequences with co-prime bases into a vector

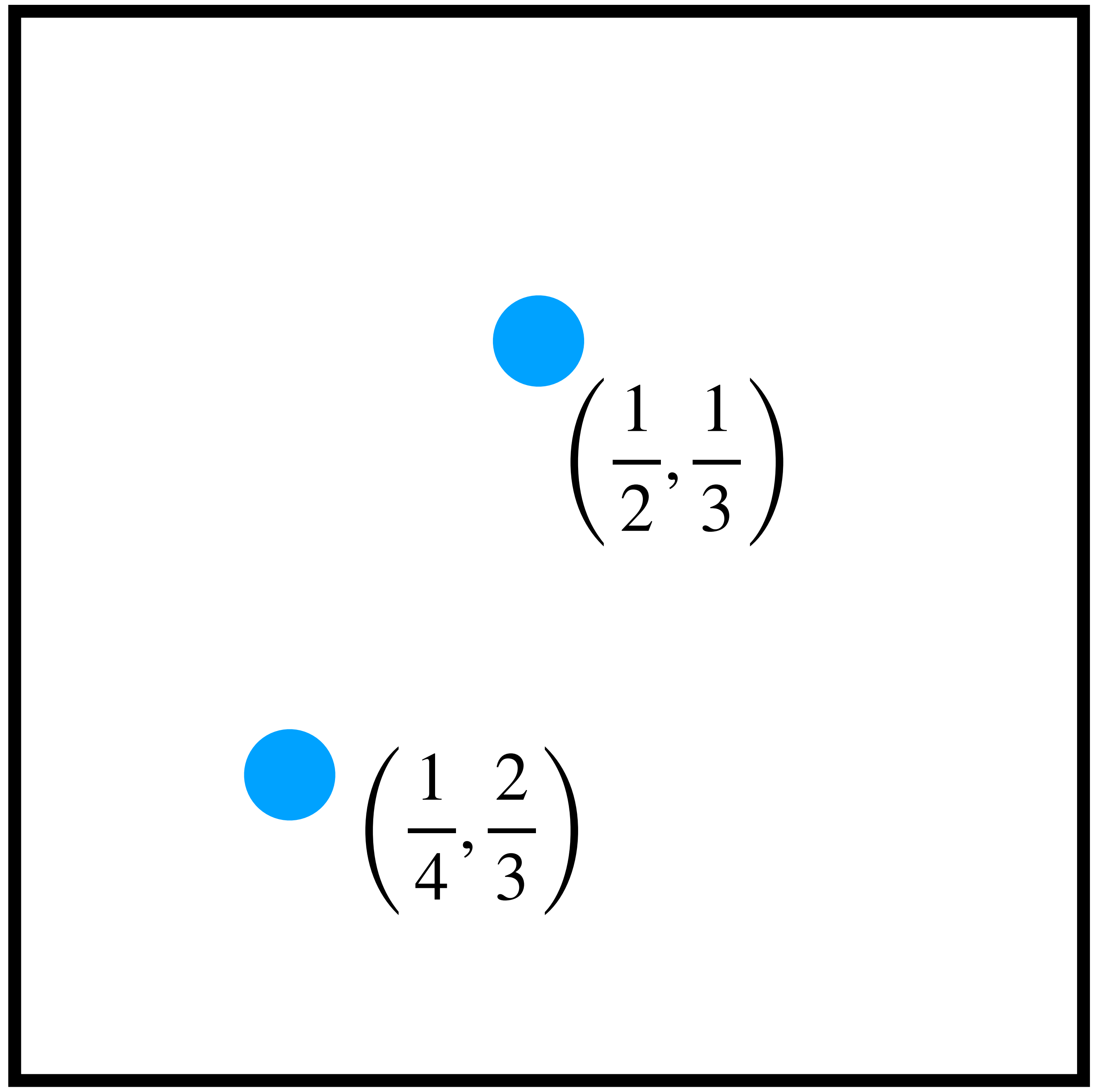


ALGORITHM 247
RADICAL-INVERSE QUASI-RANDOM POINT SEQUENCE [G5]
J. H. Halton and G. B. Smith (Recd. 24 Jan. 1964 and 21 July 1964)
Brookhaven National Laboratory, Upton, N. Y., and University of Colorado, Boulder, Colo.

High-dimensional generalization of van der Corput sequence: Halton sequence

$$\text{Halton}(k) = \left(\Phi_2(k), \Phi_3(k), \Phi_5(k), \cdots\right)$$

concatenate van der Corput sequences with co-prime bases into a vector

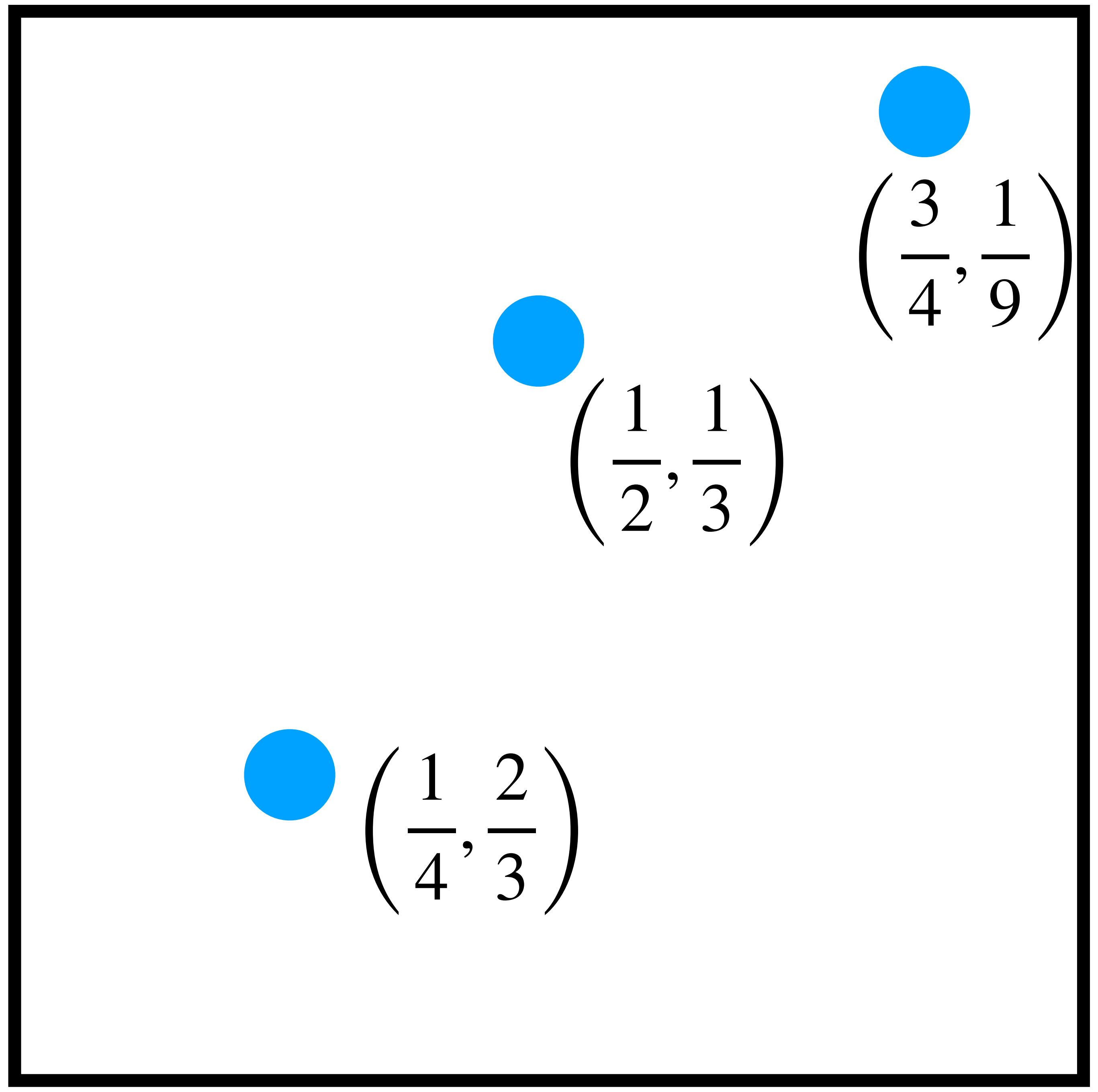


ALGORITHM 247
RADICAL-INVERSE QUASI-RANDOM POINT SEQUENCE [G5]
J. H. Halton and G. B. Smith (Recd. 24 Jan. 1964 and 21 July 1964)
Brookhaven National Laboratory, Upton, N. Y., and University of Colorado, Boulder, Colo.

High-dimensional generalization of van der Corput sequence: Halton sequence

$$\text{Halton}(k) = \left(\Phi_2(k), \Phi_3(k), \Phi_5(k), \cdots\right)$$

concatenate van der Corput sequences with co-prime bases into a vector

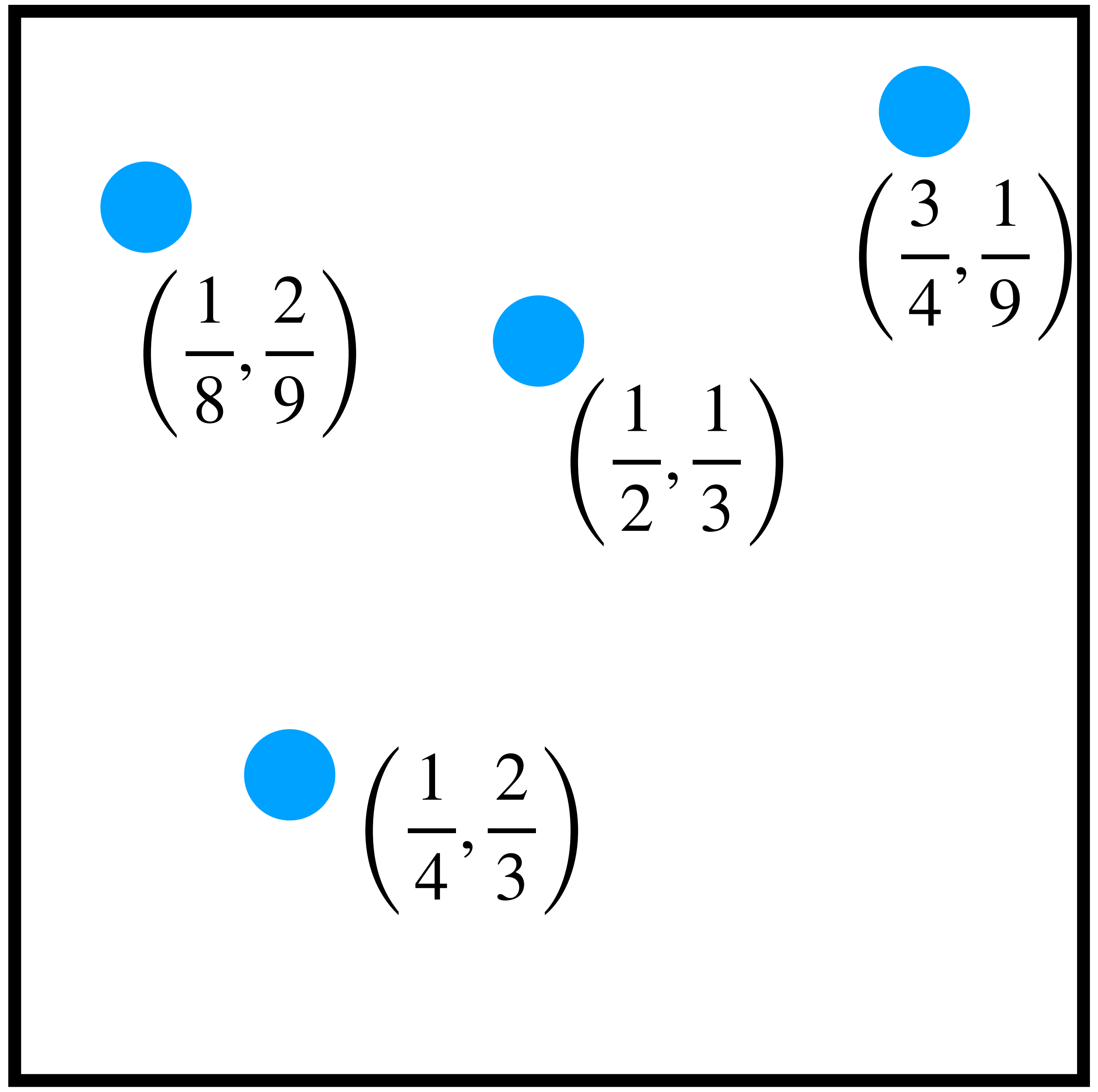


ALGORITHM 247
RADICAL-INVERSE QUASI-RANDOM POINT SEQUENCE [G5]
J. H. Halton and G. B. Smith (Recd. 24 Jan. 1964 and 21 July 1964)
Brookhaven National Laboratory, Upton, N. Y., and University of Colorado, Boulder, Colo.

High-dimensional generalization of van der Corput sequence: Halton sequence

$$\text{Halton}(k) = \left(\Phi_2(k), \Phi_3(k), \Phi_5(k), \cdots\right)$$

concatenate van der Corput sequences with co-prime bases into a vector



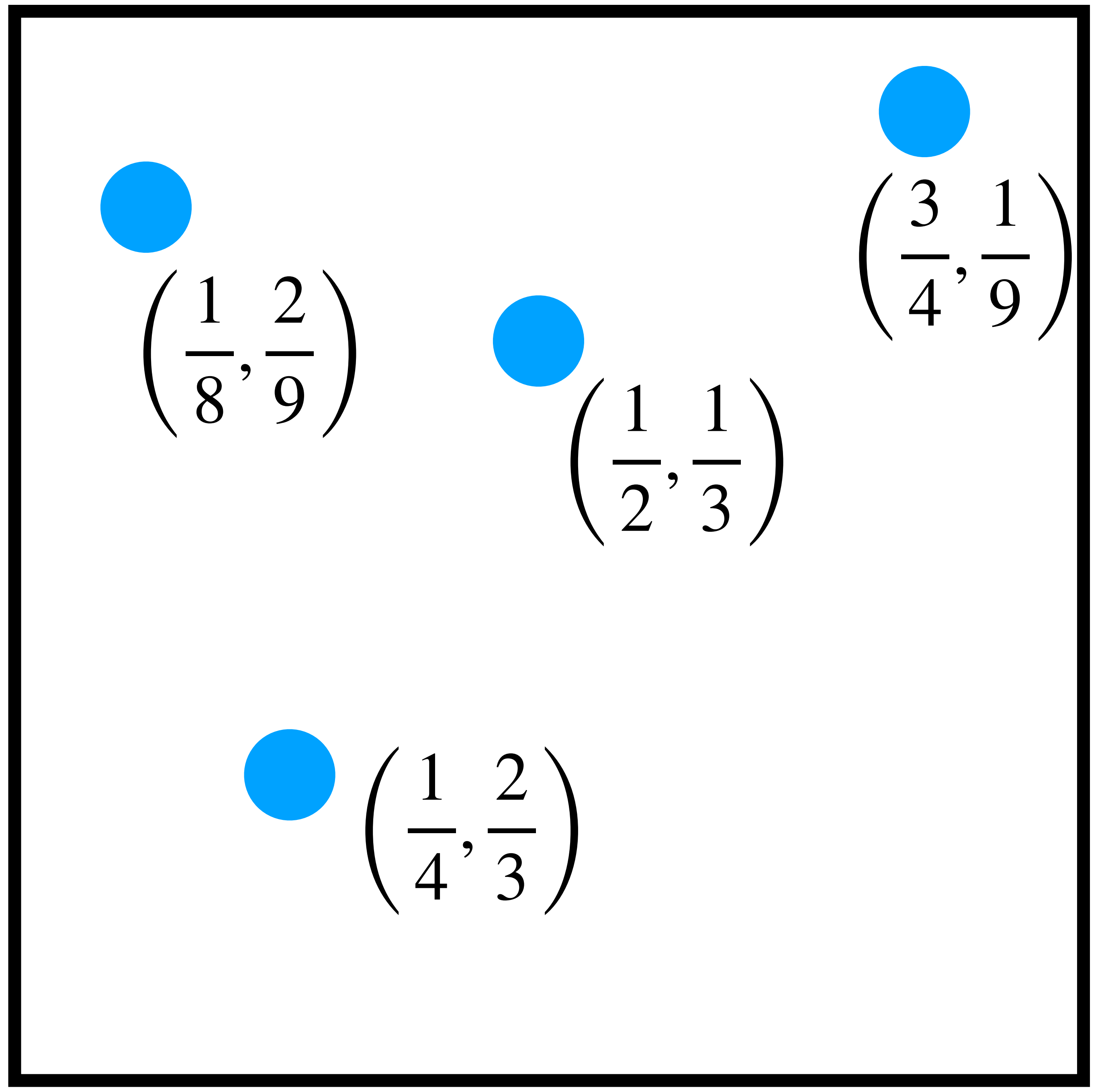
ALGORITHM 247
RADICAL-INVERSE QUASI-RANDOM POINT SEQUENCE [G5]
J. H. Halton and G. B. Smith (Recd. 24 Jan. 1964 and 21 July 1964)
Brookhaven National Laboratory, Upton, N. Y., and University of Colorado, Boulder, Colo.

High-dimensional generalization of van der Corput sequence: Halton sequence

$$\text{Halton}(k) = \left(\Phi_2(k), \Phi_3(k), \Phi_5(k), \cdots\right)$$

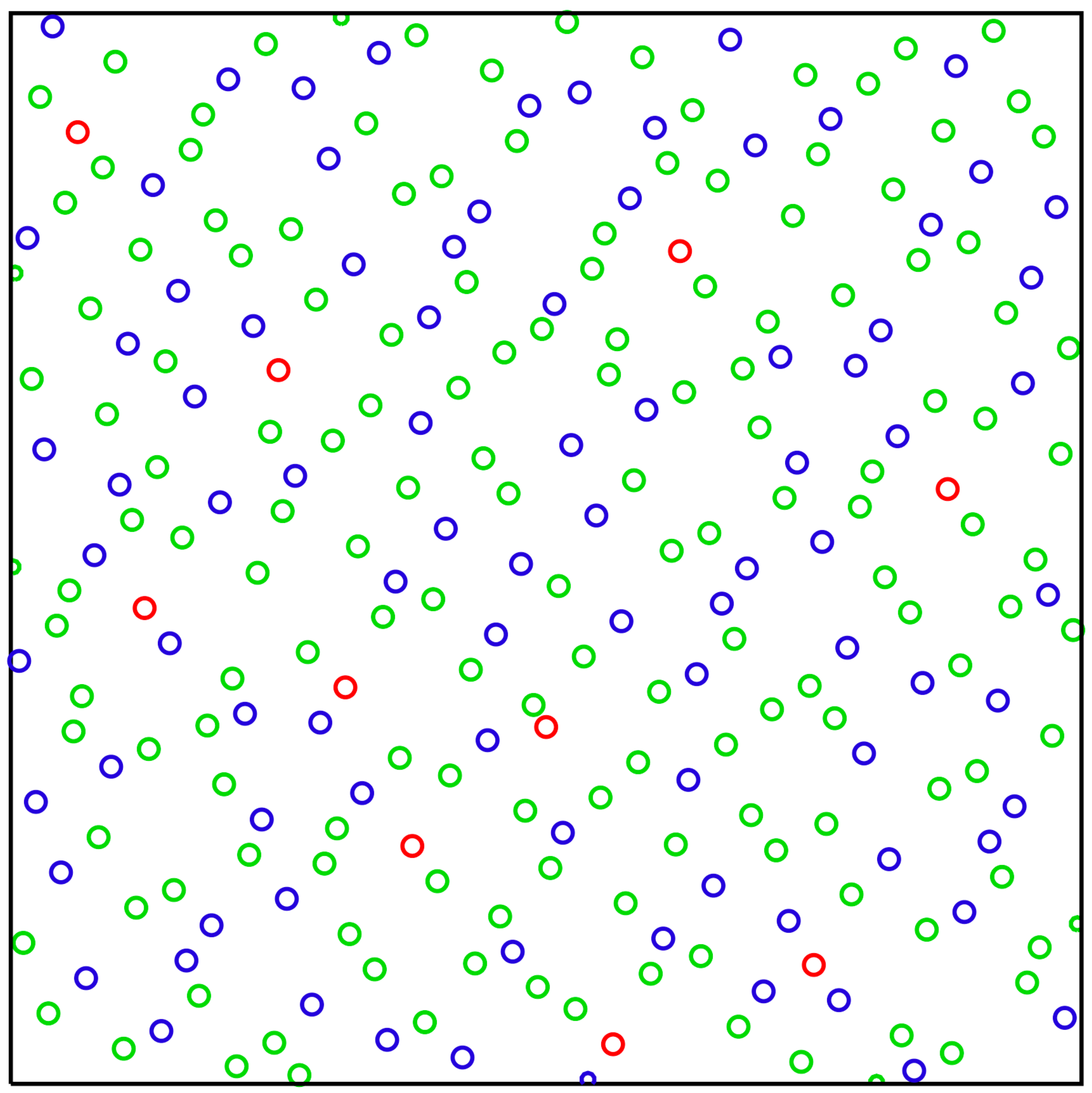
concatenate van der Corput sequences with co-prime bases into a vector

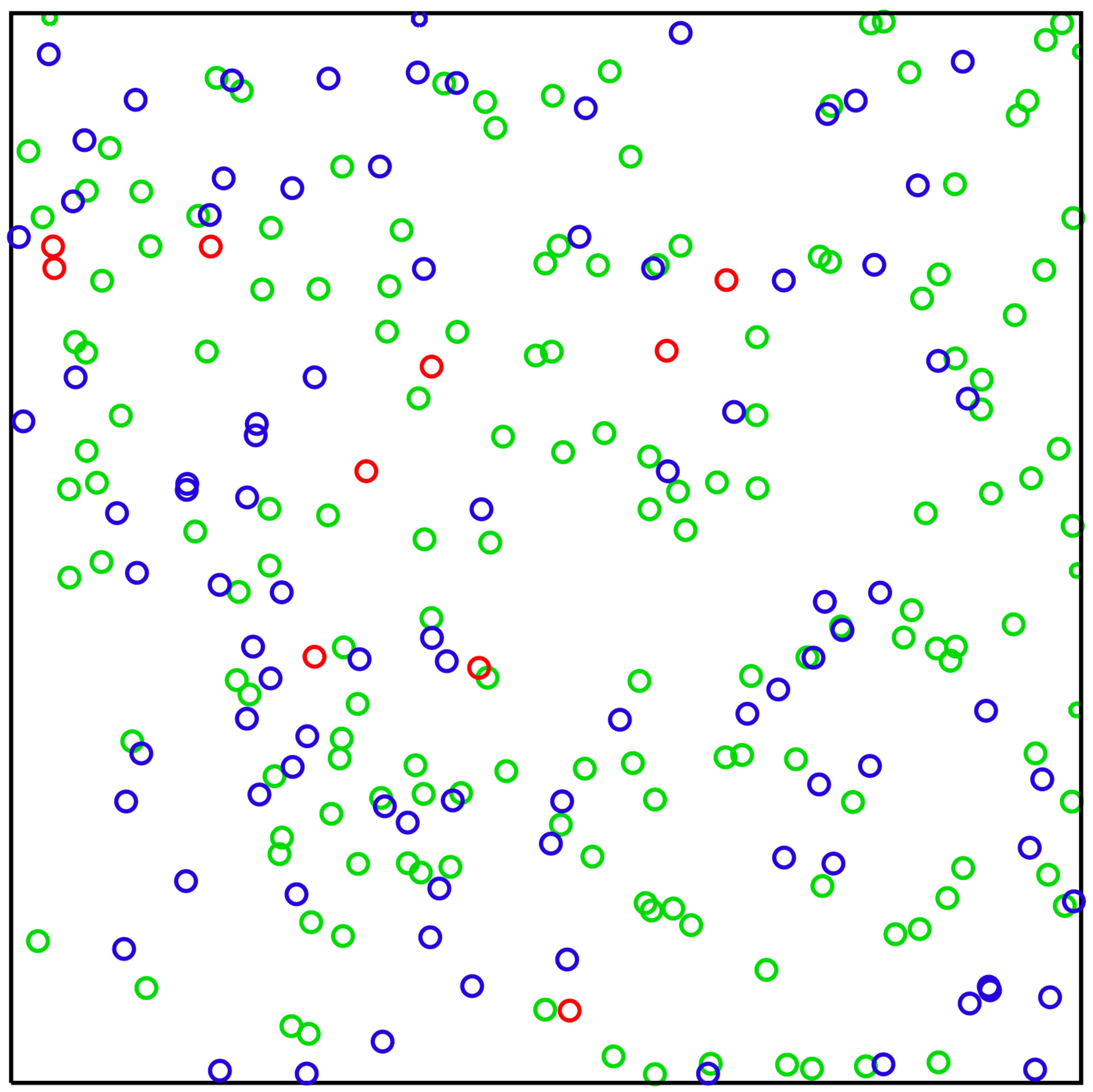
progressive &
naturally generalize to
higher dimension!



ALGORITHM 247
RADICAL-INVERSE QUASI-RANDOM POINT SEQUENCE [G5]
J. H. Halton and G. B. Smith (Recd. 24 Jan. 1964 and 21 July 1964)
Brookhaven National Laboratory, Upton, N. Y., and University of Colorado, Boulder, Colo.

Halton sequence vs independent noise





https://en.wikipedia.org/wiki/Halton_sequence

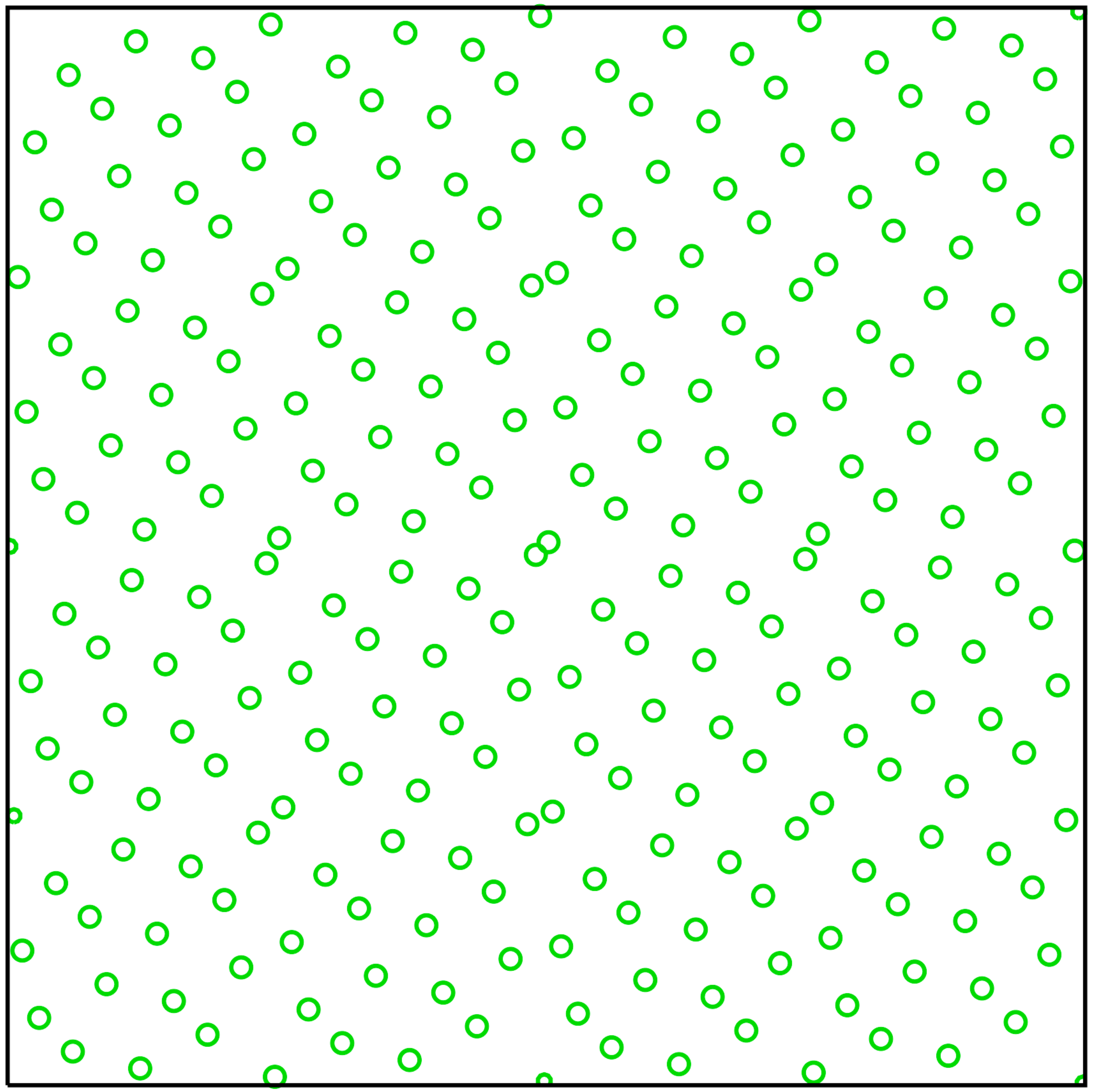
Hammersley sequence

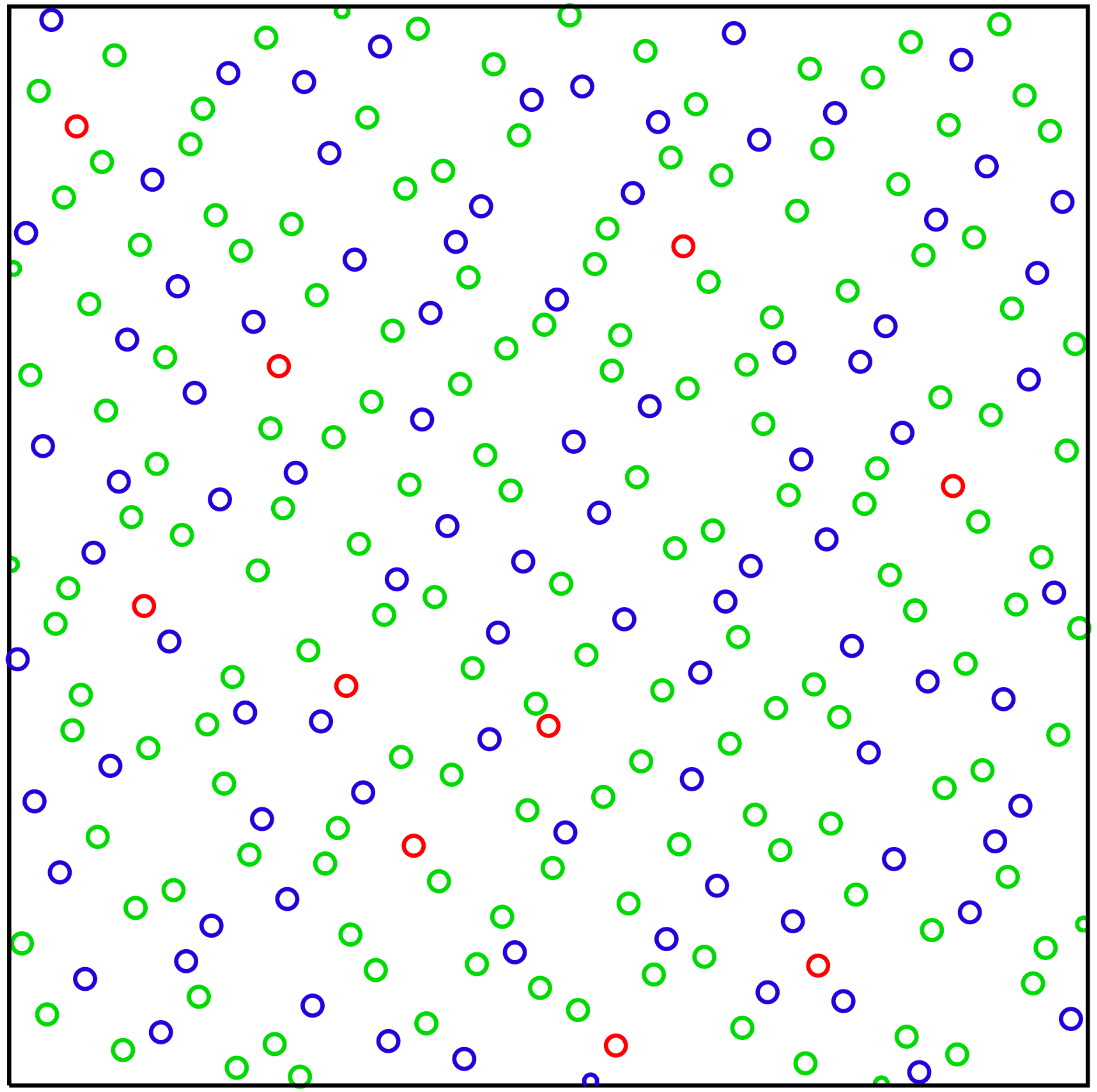
- append Halton sequence with $\frac{k}{N}$, N is the total number of samples
  - not progressive anymore, but more evenly distributed

$$\text{Halton}(k) = \left(\Phi_2(k), \Phi_3(k), \Phi_5(k), \cdots\right)$$

$$\text{Hammersley}(k) = \left(\frac{k}{N}, \Phi_2(k), \Phi_3(k), \Phi_5(k), \cdots\right)$$

Hammersley vs Halton sequences





https://en.wikipedia.org/wiki/Low-discrepancy_sequence

Discrepancies of Halton / Hammersley sequence

$$D_n^* = O\left(\frac{\left(\log N\right)^d}{N}\right)$$

Koksma-Hlawka inequality

$$\left|\frac{1}{n}\sum_{i=0}^{n} f(x_i) - \int f(x)\mathrm{d}x\right| \leq V(f)D_n^*(x_1, x_2, \ldots, x_n)$$

Convergence rates compared (2D)

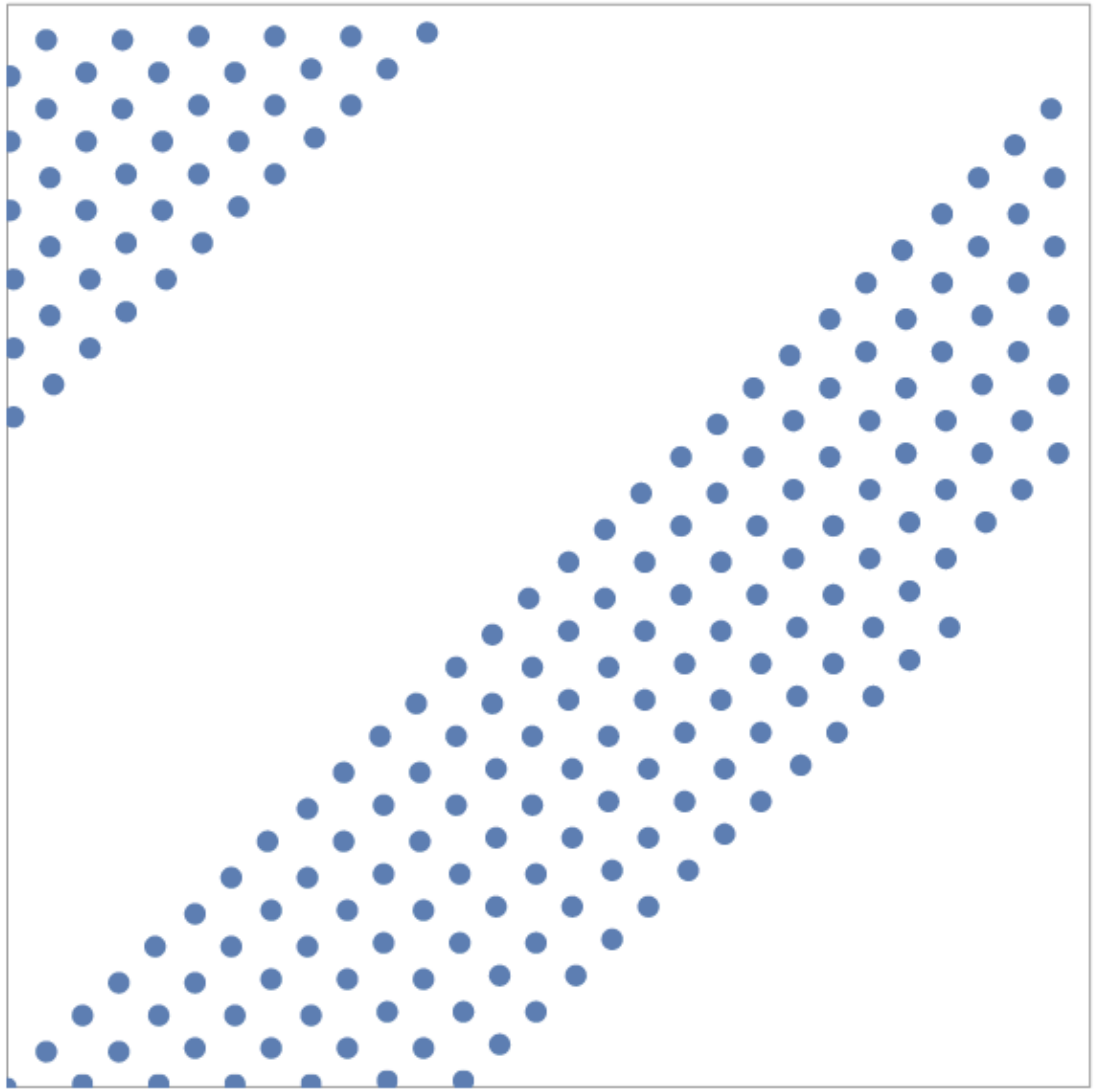
Samplers	Worst Case	Best Case
Random	$\mathcal{O}(N^{-1})$	$\mathcal{O}(N^{-1})$
Jitter	$\mathcal{O}(N^{-1.5})$	$\mathcal{O}(N^{-2})$
Poisson Disk	$\mathcal{O}(N^{-1})$	$\mathcal{O}(N^{-1})$
CCVT	$\mathcal{O}(N^{-1.5})$	$\mathcal{O}(N^{-3})$

$$O\left(\frac{(\log N)^2}{N}\right)$$

for large N, low-discrepancy sequences win

(though note that V(f) is often unbounded in rendering)

Issues of Halton/Hammersley sequences: correlated pattern in high dimension



$\left(\Phi_{29}, \Phi_{31}\right)$

https://www.pbr-book.org/3ed-2018/Sampling_and_Reconstruction/The_Halton_Sampler

A solution: scramble the digits of each coordinate

scrambling preserves (often improves) discrepancy!

$$\Phi_{29}(k) = 0.abcdefg_{29}$$

↓ scramble

$$\Phi_{29}(k) = 0.cdabgfe_{29}$$

apply the same scramble to all k!

1992

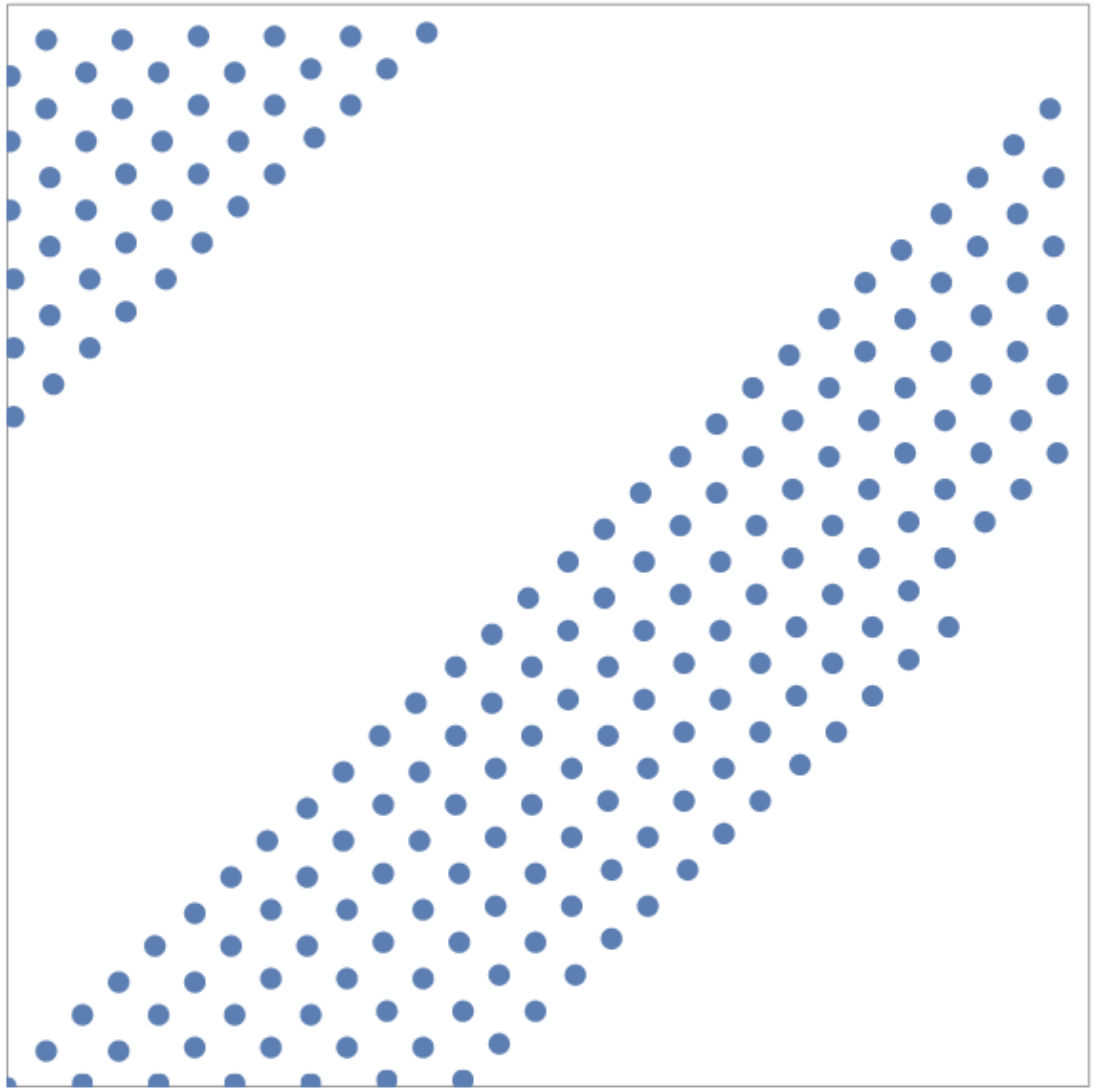
Good Permutations for Extreme Discrepancy

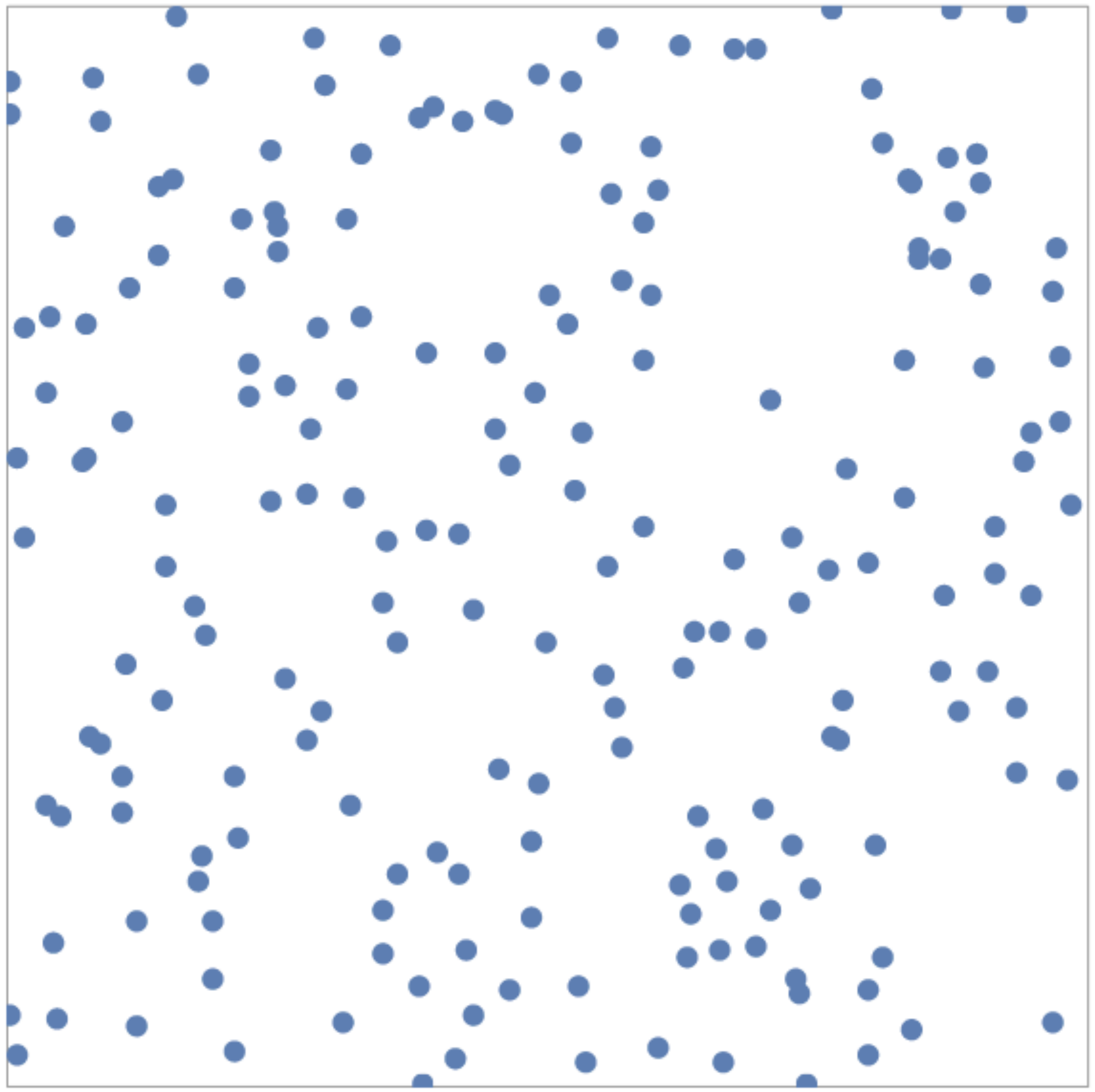
HENRI FAURE

Mathématiques Informatique, Université de Provence, 3, Place Victor-Hugo, 13331 Marseille, Cedex 3, France

Scrambling fixes the regularity issue

(the clumping might look bad, but note that this is a projection of a high-dimensional point set)

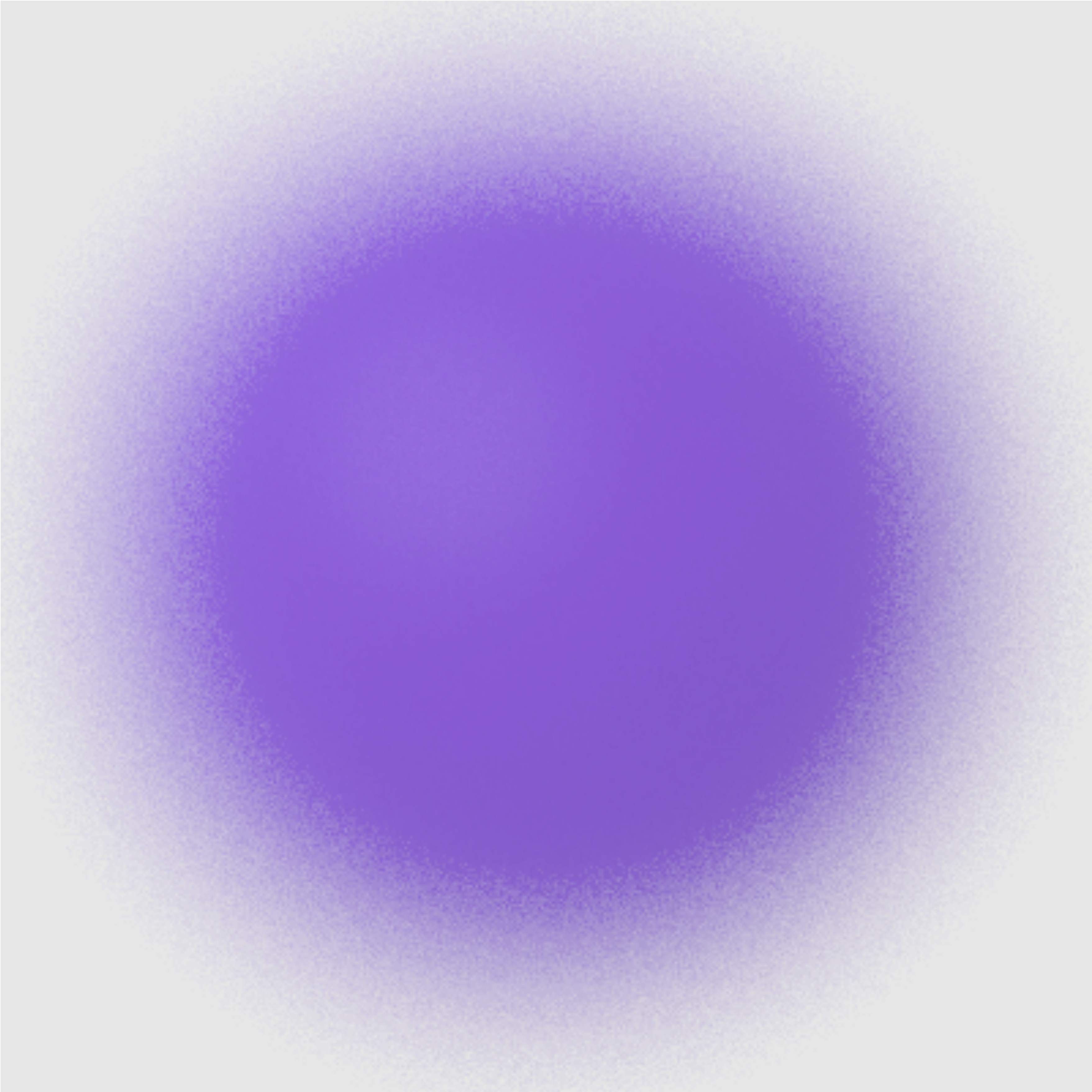




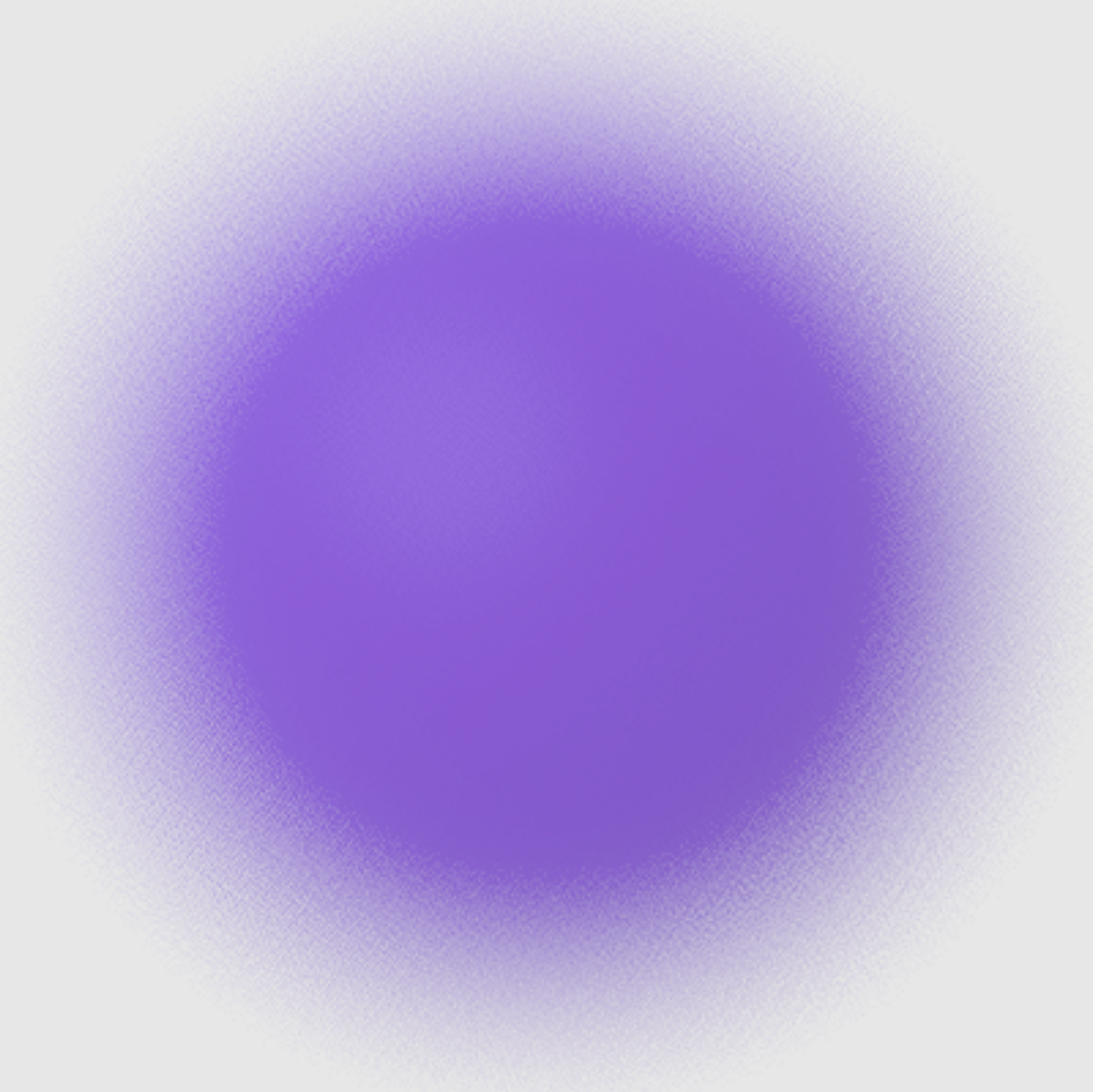
(Φ_{29}, Φ_{31})

Jittered vs Halton

4D integral



jittered



Halton

https://www.pbr-book.org/3ed-2018/Sampling_and_Reconstruction/Sobol_Sampler

Another solution to the correlation problem: don’t use high bases!

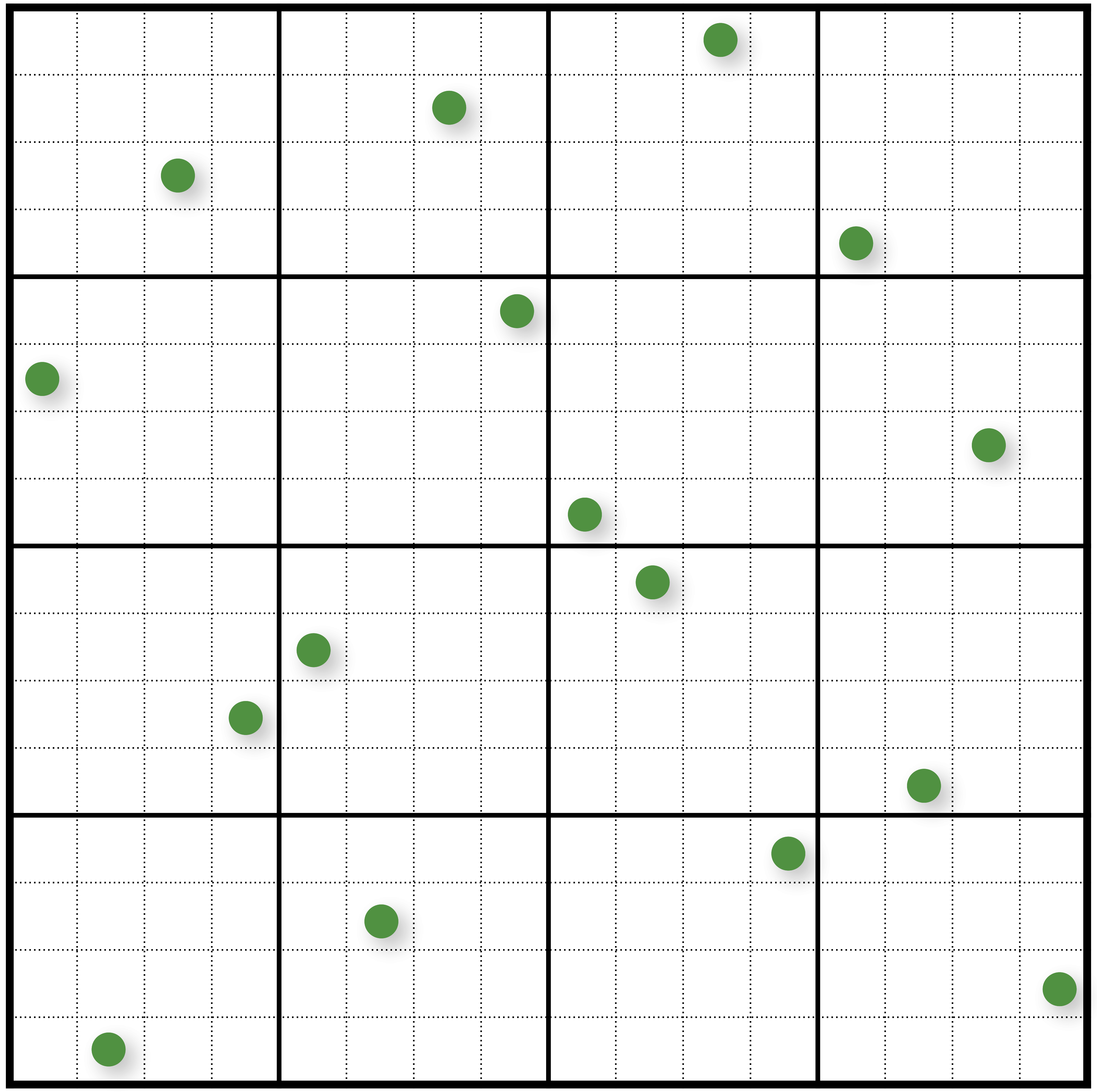
digital nets: low discrepancy sequences constructed only using low number bases

$\Phi_2(k)$ ✓

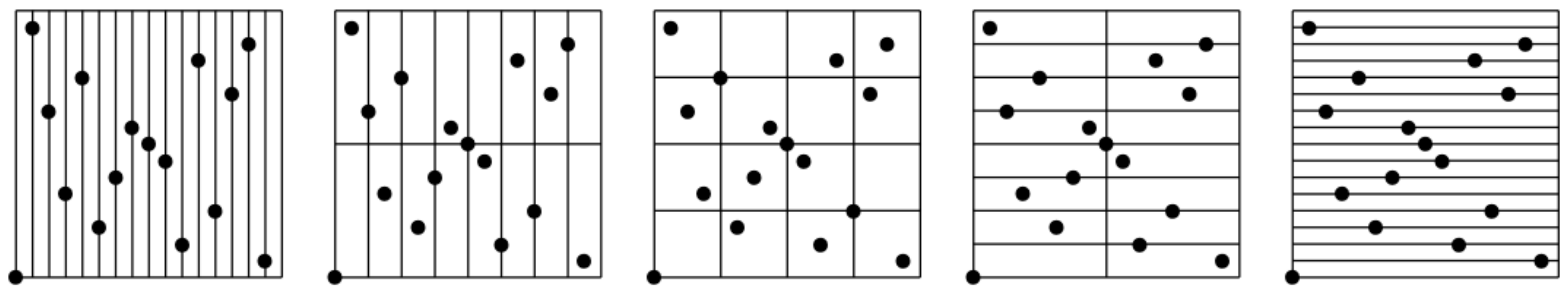
$\Phi_{29}(k)$ ✗

The first two dimensions of Hammersley sequence follows the digital nets property

higher dimensions do not follow the same property
due to the large base b



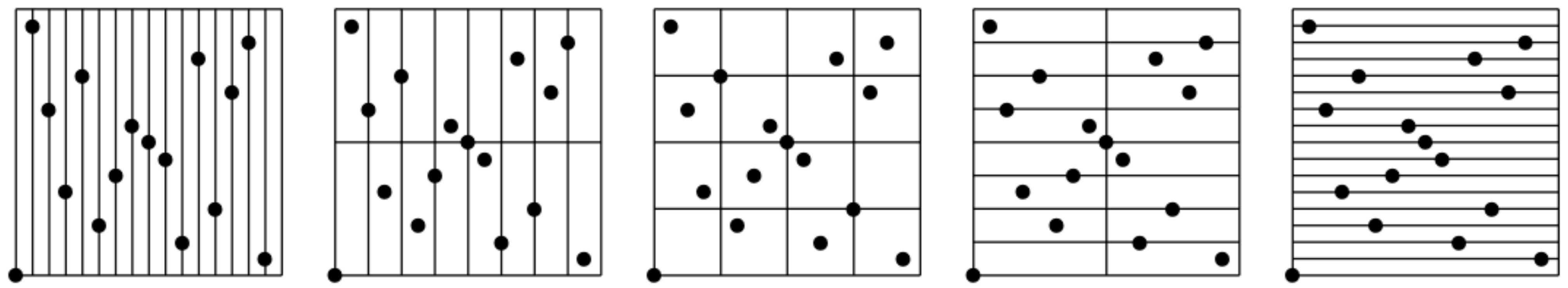
Definition of digital nets

elementary interval: partition of space into equal-size rectangles 

https://gruenschloss.org/sample-enum/sample-enum.pdf

Definition of digital nets

elementary interval: partition of space into equal-size rectangles

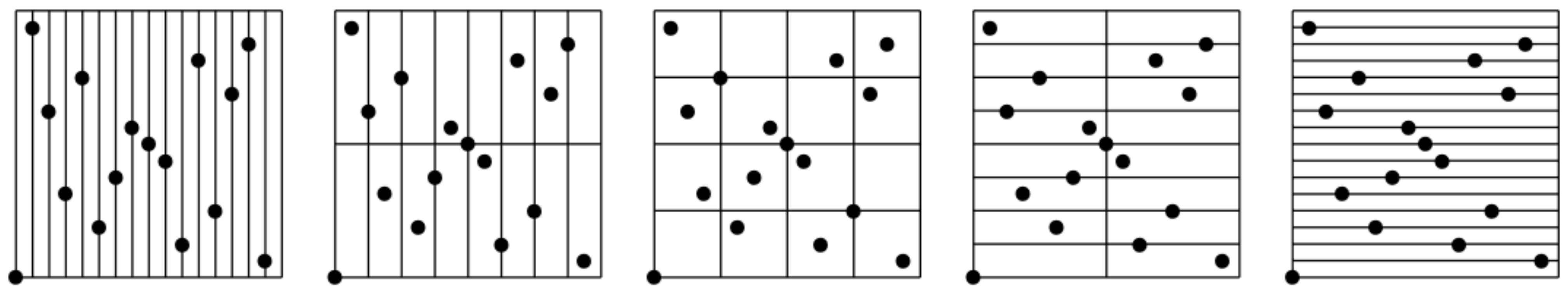


digital nets: for two non-negative integers $t \leq m$, a (t, m, s)-net in base b is a finite point set with b^m points with s dimensions where each elementary interval of volume b^{t-m} contains exactly b^t points

https://gruenschloss.org/sample-enum/sample-enum.pdf

Definition of digital nets

elementary interval: partition of space into equal-size rectangles



digital nets: for two non-negative integers $t \leq m$, a (t, m, s)-net in base b is a finite point set with b^m points with s dimensions where each elementary interval of volume b^{t-m} contains exactly b^t points

a **(t, s)-sequence** is an infinite point sequence whose subsequences form a digital net

https://gruenschloss.org/sample-enum/sample-enum.pdf

Sobol’ sequence satisfies digital nets property

aka Faure or Niederreiter or digital sequence

want to find a function $y_d = f_d(k)$ that will output a number y_d in base b for each dimension d

Sobol' sequence satisfies digital nets property

aka Faure or Niederreiter or digital sequence

want to find a function $y_d = f_d(k)$ that will output a number y_d in base b for each dimension d

represents k and y in terms of their digits

$$k = k_1 k_2 k_3 \ldots k_{m_b} \qquad\qquad y = 0.y_1 y_2 y_3 \ldots y_{m_b}$$

Sobol' sequence satisfies digital nets property

aka Faure or Niederreiter or digital sequence

want to find a function $y_d = f_d(k)$ that will output a number y_d in base b for each dimension d

$$y = 0.y_1 y_2 y_3 \ldots y_{m_b}$$

turn the digits into vectors

$$\begin{bmatrix} k_1 \\ k_2 \\ \vdots \\ k_m \end{bmatrix} \qquad \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

Sobol' sequence satisfies digital nets property

aka Faure or Niederreiter or digital sequence

want to find a function $y_d = f_d(k)$ that will output a number y_d in base b for each dimension d

let f be a linear function (!?) applied to the k vector

$$y = 0.y_1 y_2 y_3 \ldots y_{m_b}$$

$$\begin{bmatrix} c_{1,1} \cdots c_{1,m} \\ c_{2,1} \cdots c_{2,m} \\ \vdots \\ c_{m,1} \cdots c_{m,m} \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \\ \vdots \\ k_m \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

(need to take modulo of b after multiplication)

Sobol' sequence satisfies digital nets property

aka Faure or Niederreiter or digital sequence

want to find a function $y_d = f_d(k)$ that will output a number y_d in base b for each dimension d

let f be a linear function (!?) applied to the k vector

$$y = 0.y_1 y_2 y_3 \ldots y_{m_b}$$

"generator matrix"

$$\begin{bmatrix} c_{1,1} \cdots c_{1,m} \\ c_{2,1} \cdots c_{2,m} \\ \vdots \\ c_{m,1} \cdots c_{m,m} \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \\ \vdots \\ k_m \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

(need to take modulo of b after multiplication)

Intuition of the generator matrix

generator matrix can be seen as a generalization of “scrambling”
before we apply the van der Corput sequence transformation (as opposed to post scrambling)

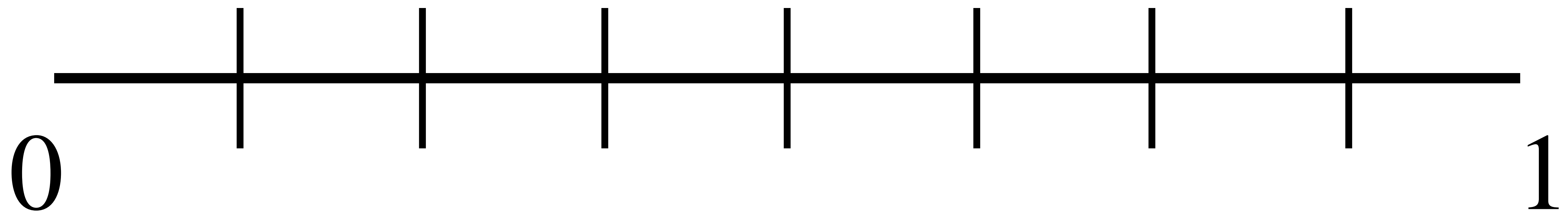
$$y = 0.y_1 y_2 y_3 \ldots y_{m_b}$$

can be a permutation matrix!

$$\begin{bmatrix} c_{1,1} \cdots c_{1,m} \\ c_{2,1} \cdots c_{2,m} \\ \vdots \\ c_{m,1} \cdots c_{m,m} \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \\ \vdots \\ k_m \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

Intuition of the generator matrix

focus on base 2 for now — k_i and y_i are either 0 or 1



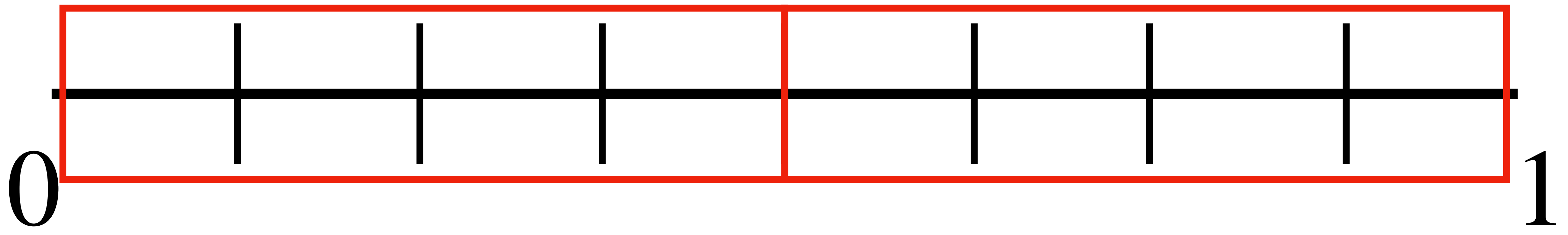
$$y = 0.y_1 y_2 y_3 \ldots y_{m_b}$$

$$\begin{bmatrix} c_{1,1} \cdots c_{1,m} \\ c_{2,1} \cdots c_{2,m} \\ \vdots \\ c_{m,1} \cdots c_{m,m} \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \\ \vdots \\ k_m \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

Intuition of the generator matrix

1D elementary intervals

focus on base 2 for now — k_i and y_i are either 0 or 1



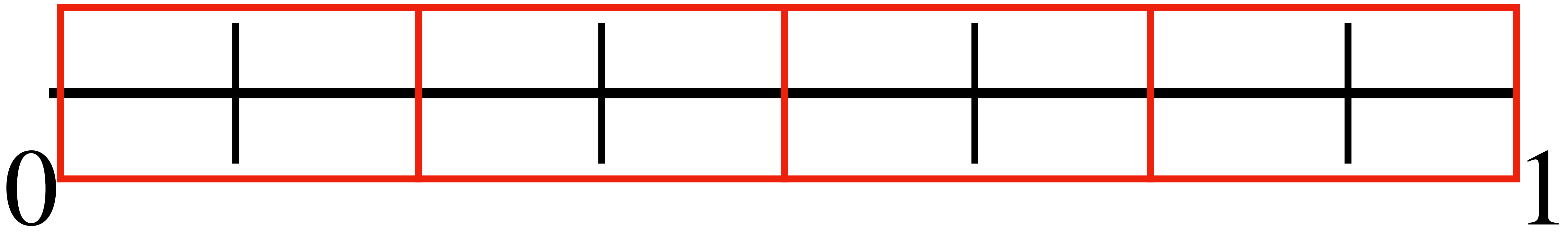
$$y = 0.y_1y_2y_3\ldots y_{m_b}$$

$$\begin{bmatrix} c_{1,1}\cdots c_{1,m} \\ c_{2,1}\cdots c_{2,m} \\ \vdots \\ c_{m,1}\cdots c_{m,m} \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \\ \vdots \\ k_m \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

first row is responsible for the first digit of y

Intuition of the generator matrix

1D elementary intervals



focus on base 2 for now — k_i and y_i are either 0 or 1

$$y = 0.y_1y_2y_3\ldots y_{m_b}$$

$$\begin{bmatrix} c_{1,1} \cdots c_{1,m} \\ c_{2,1} \cdots c_{2,m} \\ \vdots \\ c_{m,1} \cdots c_{m,m} \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \\ \vdots \\ k_m \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

first two rows are responsible for the first two digits of y

Intuition of the generator matrix — 2D

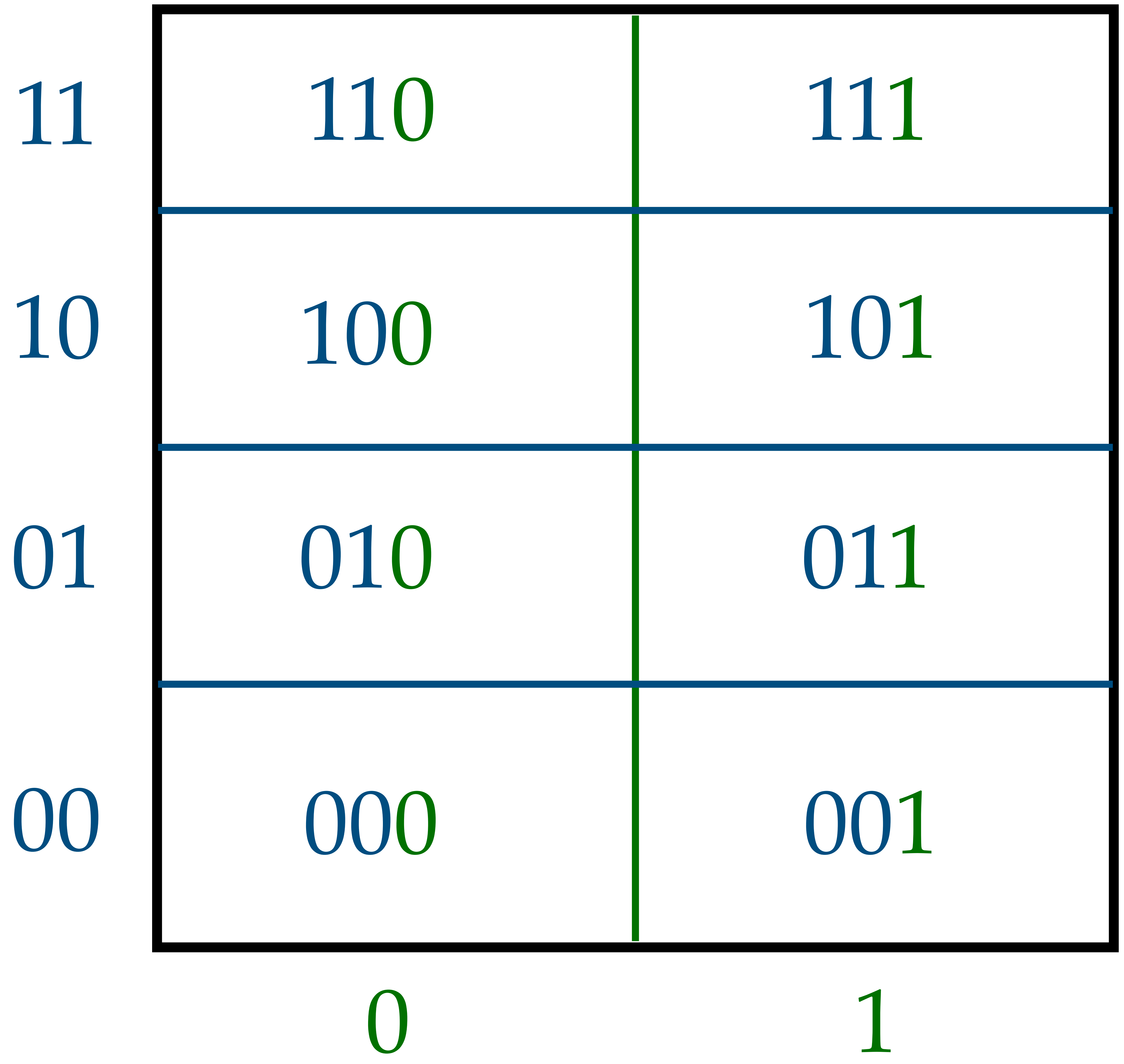
$$\begin{bmatrix} c_{1,1} & c_{1,2} & c_{1,3} \\ c_{2,1} & c_{2,2} & c_{2,3} \\ c_{3,1} & c_{3,2} & c_{3,3} \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

$$\begin{bmatrix} c'_{1,1} & c'_{1,2} & c'_{1,3} \\ c'_{2,1} & c'_{2,2} & c'_{2,3} \\ c'_{3,1} & c'_{3,2} & c'_{3,3} \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix} = \begin{bmatrix} y'_1 \\ y'_2 \\ y'_3 \end{bmatrix}$$

Intuition of the generator matrix — 2D

$$\begin{bmatrix} c_{1,1} & c_{1,2} & c_{1,3} \\ c_{2,1} & c_{2,2} & c_{2,3} \\ c_{3,1} & c_{3,2} & c_{3,3} \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

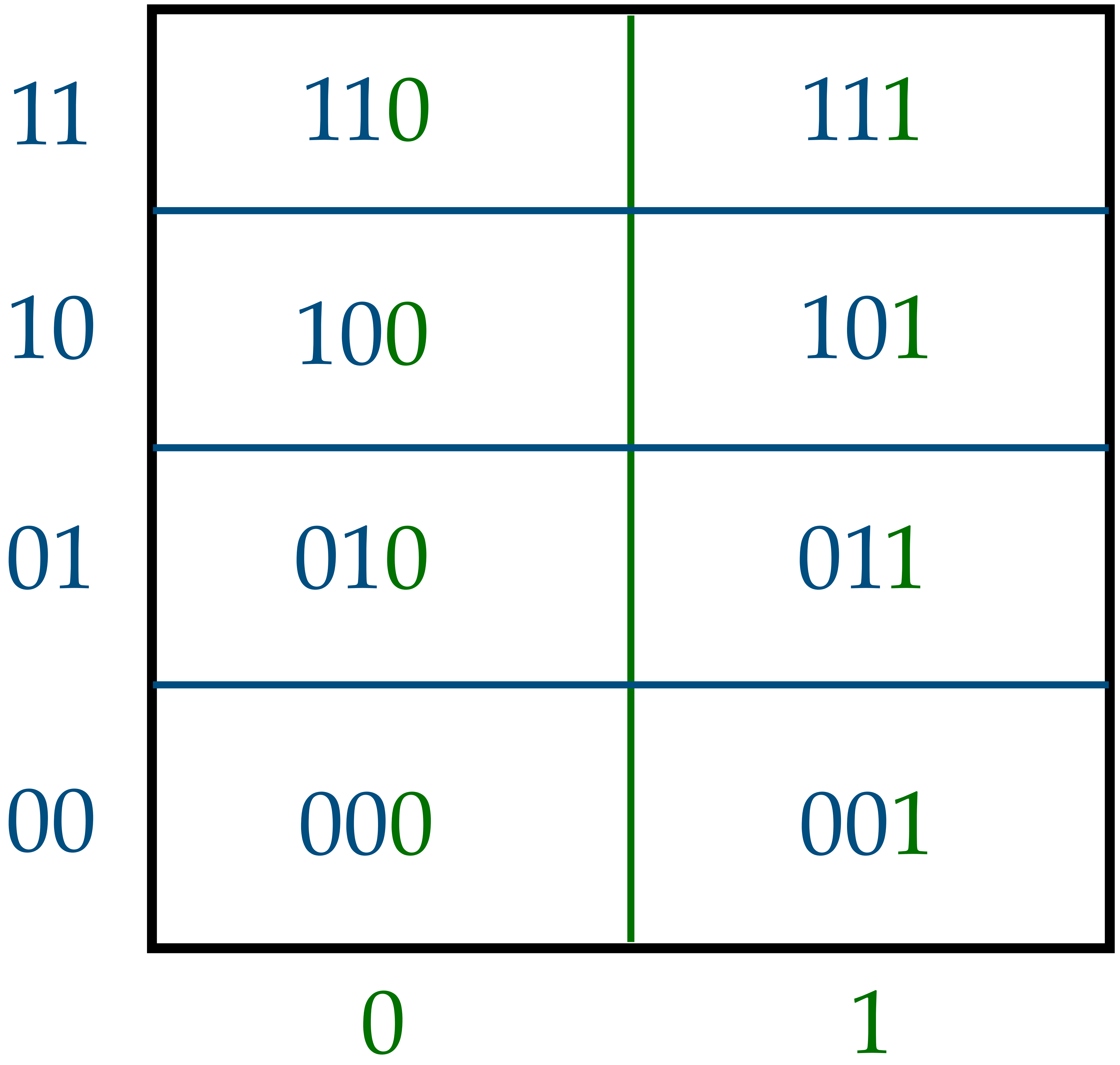
$$\begin{bmatrix} c'_{1,1} & c'_{1,2} & c'_{1,3} \\ c'_{2,1} & c'_{2,2} & c'_{2,3} \\ c'_{3,1} & c'_{3,2} & c'_{3,3} \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix} = \begin{bmatrix} y'_1 \\ y'_2 \\ y'_3 \end{bmatrix}$$



Intuition of the generator matrix — 2D

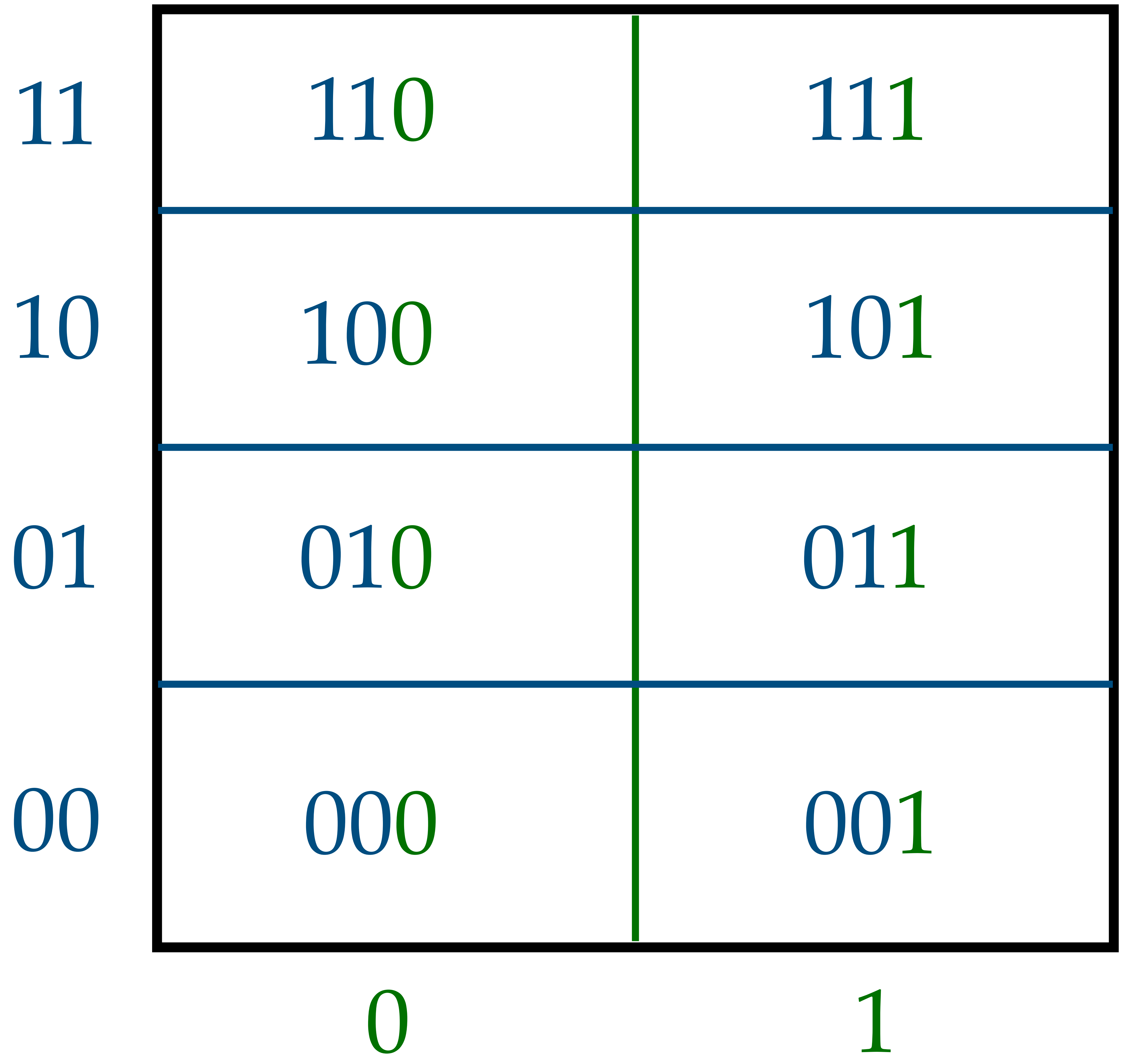
$$\begin{bmatrix} c_{1,1} & c_{1,2} & c_{1,3} \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix} = \begin{bmatrix} y_1 \end{bmatrix}$$

$$\begin{bmatrix} c'_{1,1} & c'_{1,2} & c'_{1,3} \\ c'_{2,1} & c'_{2,2} & c'_{2,3} \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix} = \begin{bmatrix} y'_1 \\ y'_2 \end{bmatrix}$$



Intuition of the generator matrix — 2D

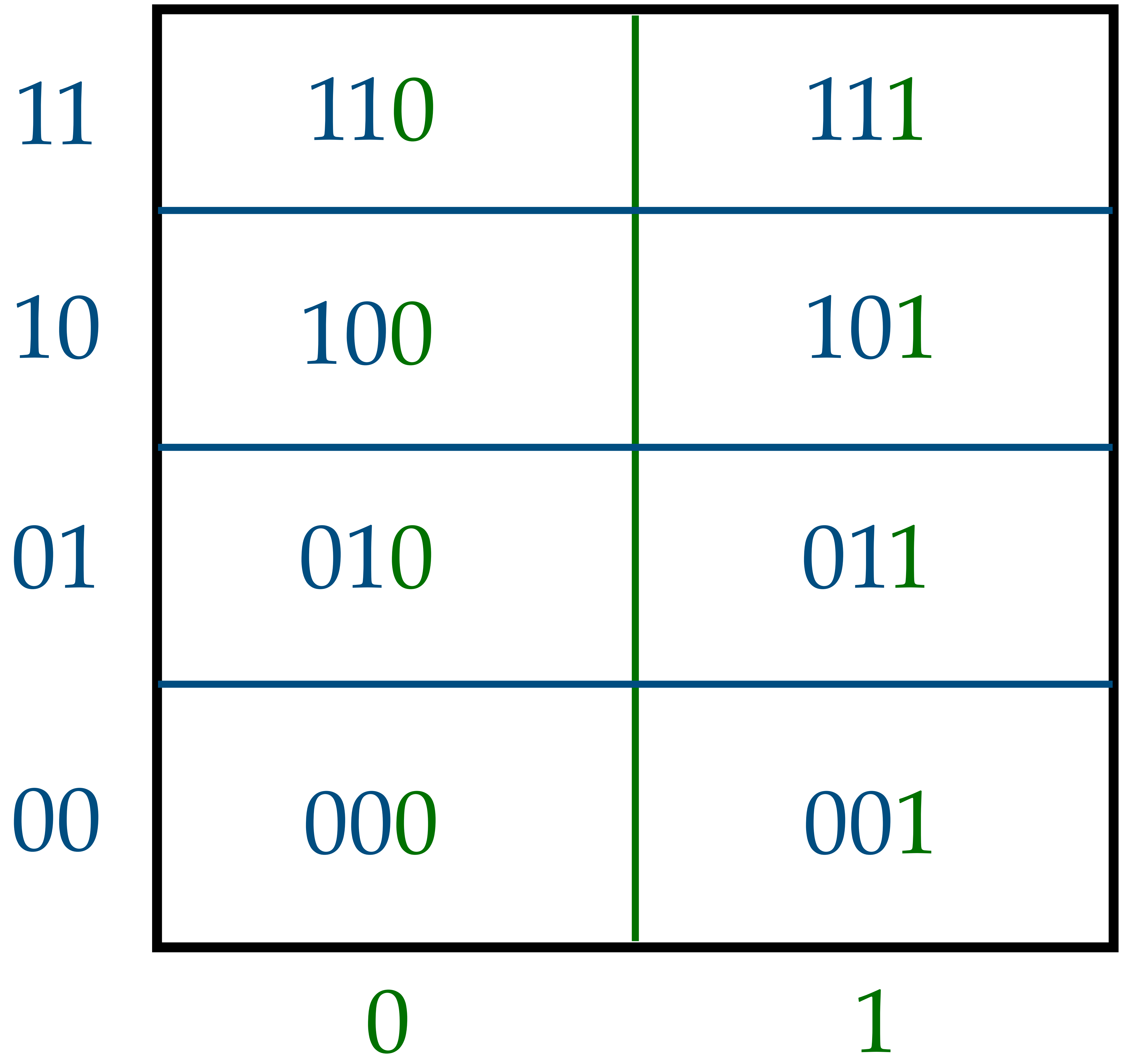
$$\begin{bmatrix} c_{1,1} & c_{1,2} & c_{1,3} \\ c'_{1,1} & c'_{1,2} & c'_{1,3} \\ c'_{2,1} & c'_{2,2} & c'_{2,3} \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y'_1 \\ y'_2 \end{bmatrix}$$



Intuition of the generator matrix — 2D

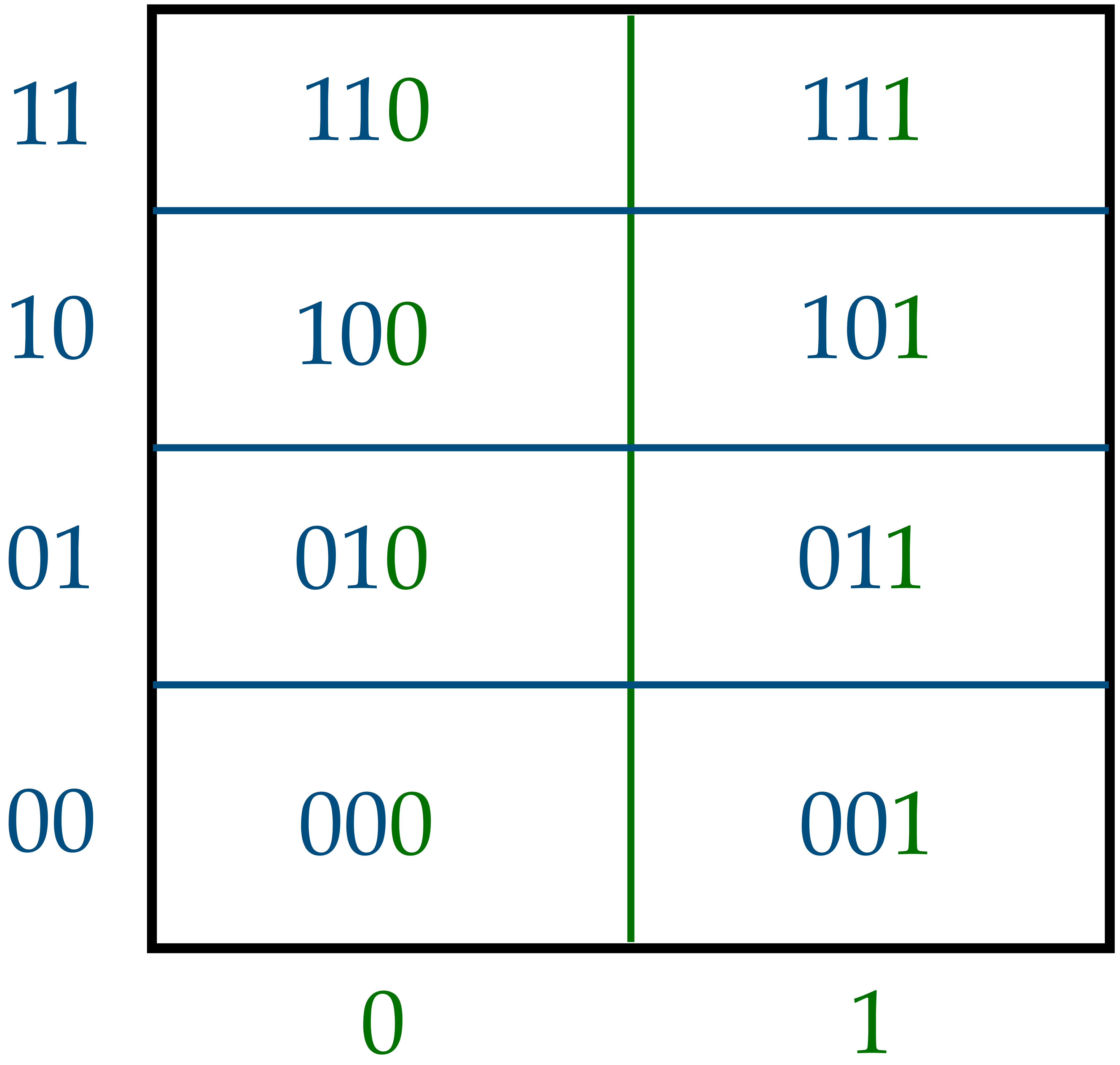
$$\begin{bmatrix} c_{1,1} & c_{1,2} & c_{1,3} \\ c'_{1,1} & c'_{1,2} & c'_{1,3} \\ c'_{2,1} & c'_{2,2} & c'_{2,3} \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y'_1 \\ y'_2 \end{bmatrix}$$

digital nets property = bijection between two vectors



Intuition of the generator matrix — 2D

$$\begin{bmatrix} c_{1,1} & c_{1,2} & c_{1,3} \\ c'_{1,1} & c'_{1,2} & c'_{1,3} \\ c'_{2,1} & c'_{2,2} & c'_{2,3} \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y'_1 \\ y'_2 \end{bmatrix}$$



digital nets property = bijection between two vectors

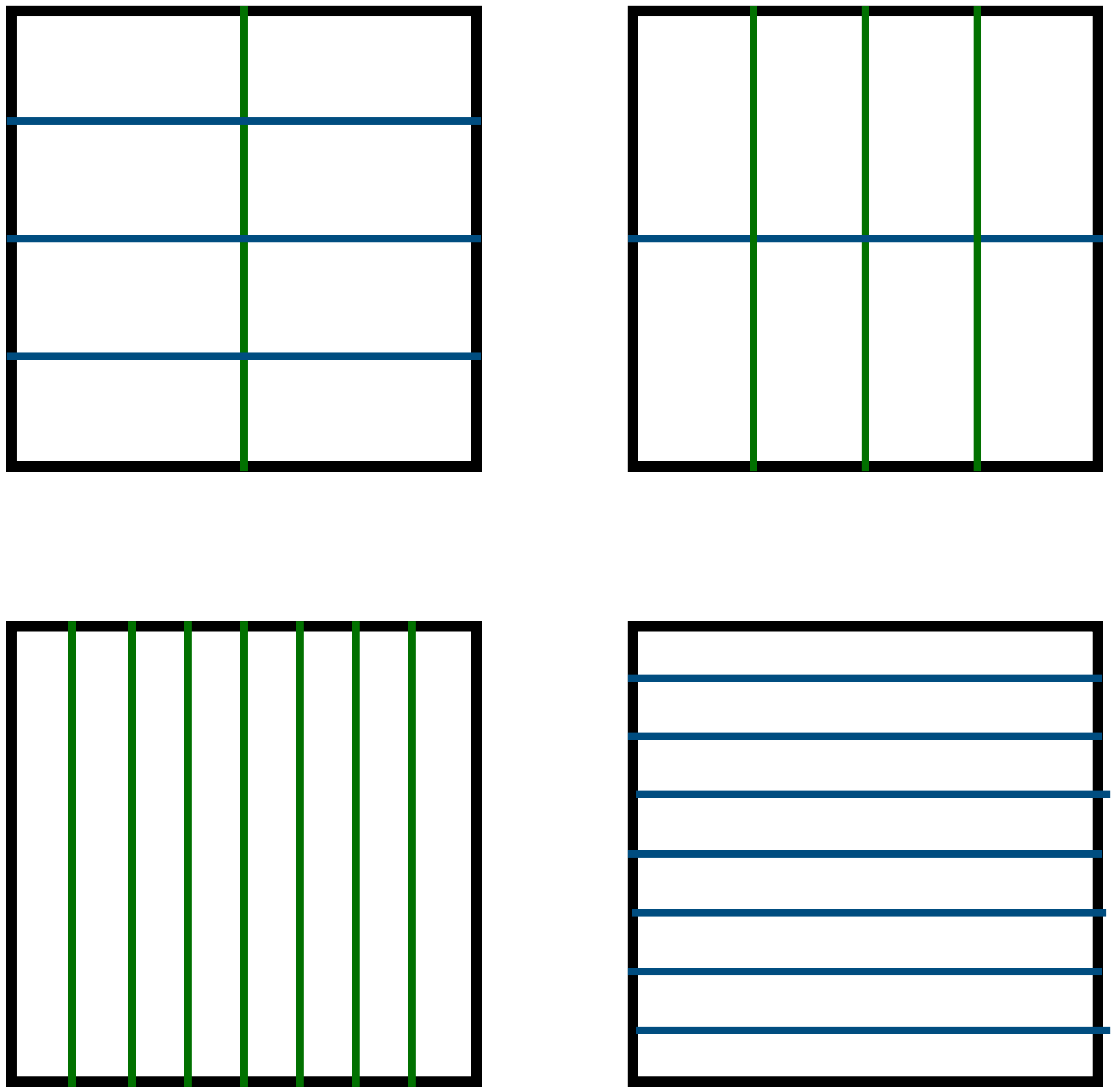
bijection = the matrix being invertible!! (i.e. det != 0)

MatBuilder: Mastering Sampling Uniformity Over Projections

LOÏS PAULIN, Univ Lyon, UCBL, CNRS, INSA Lyon, LIRIS, France
NICOLAS BONNEEL, Univ Lyon, CNRS, INSA Lyon, UCBL, LIRIS, France
DAVID COEURJOLLY, Univ Lyon, CNRS, INSA Lyon, UCBL, LIRIS, France
JEAN-CLAUDE IEHL, Univ Lyon, UCBL, CNRS, INSA Lyon, LIRIS, France
ALEXANDER KELLER, NVIDIA, Germany
VICTOR OSTROMOUKHOV, Univ Lyon, UCBL, CNRS, INSA Lyon, LIRIS, France

Designing generator matrices

enumerate all elementary intervals
(or even just subregions of the domain)



MatBuilder: Mastering Sampling Uniformity Over Projections

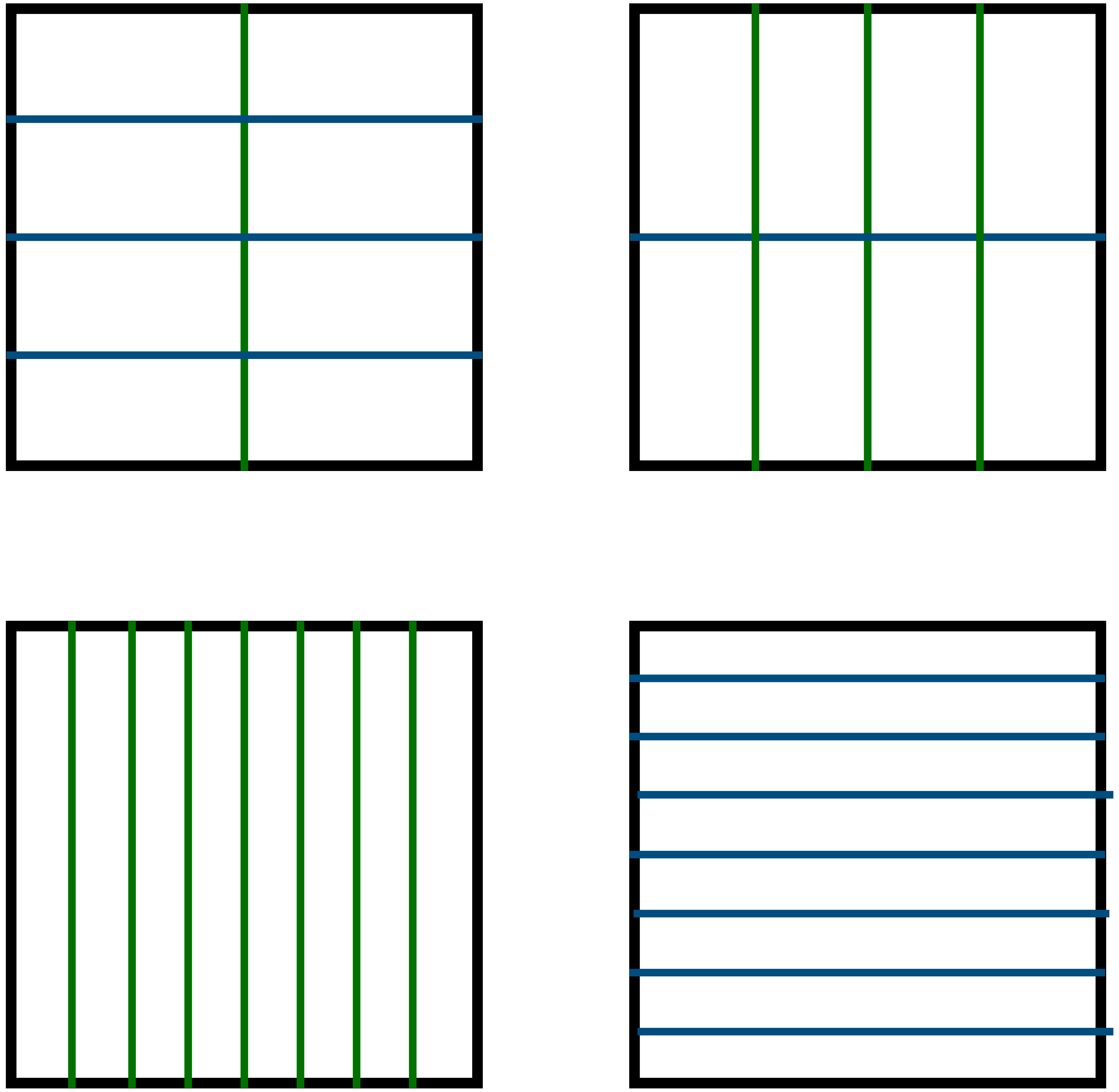
LOÏS PAULIN, Univ Lyon, UCBL, CNRS, INSA Lyon, LIRIS, France
NICOLAS BONNEEL, Univ Lyon, CNRS, INSA Lyon, UCBL, LIRIS, France
DAVID COEURJOLLY, Univ Lyon, CNRS, INSA Lyon, UCBL, LIRIS, France
JEAN-CLAUDE IEHL, Univ Lyon, UCBL, CNRS, INSA Lyon, LIRIS, France
ALEXANDER KELLER, NVIDIA, Germany
VICTOR OSTROMOUKHOV, Univ Lyon, UCBL, CNRS, INSA Lyon, LIRIS, France

Designing generator matrices

enumerate all elementary intervals
(or even just subregions of the domain)

write down the constraints of different sub matrices

det(A) != 0 det(B) != 0 det(C) != 0 det(D) != 0



MatBuilder: Mastering Sampling Uniformity Over Projections

LOÏS PAULIN, Univ Lyon, UCBL, CNRS, INSA Lyon, LIRIS, France
NICOLAS BONNEEL, Univ Lyon, CNRS, INSA Lyon, UCBL, LIRIS, France
DAVID COEURJOLLY, Univ Lyon, CNRS, INSA Lyon, UCBL, LIRIS, France
JEAN-CLAUDE IEHL, Univ Lyon, UCBL, CNRS, INSA Lyon, LIRIS, France
ALEXANDER KELLER, NVIDIA, Germany
VICTOR OSTROMOUKHOV, Univ Lyon, UCBL, CNRS, INSA Lyon, LIRIS, France

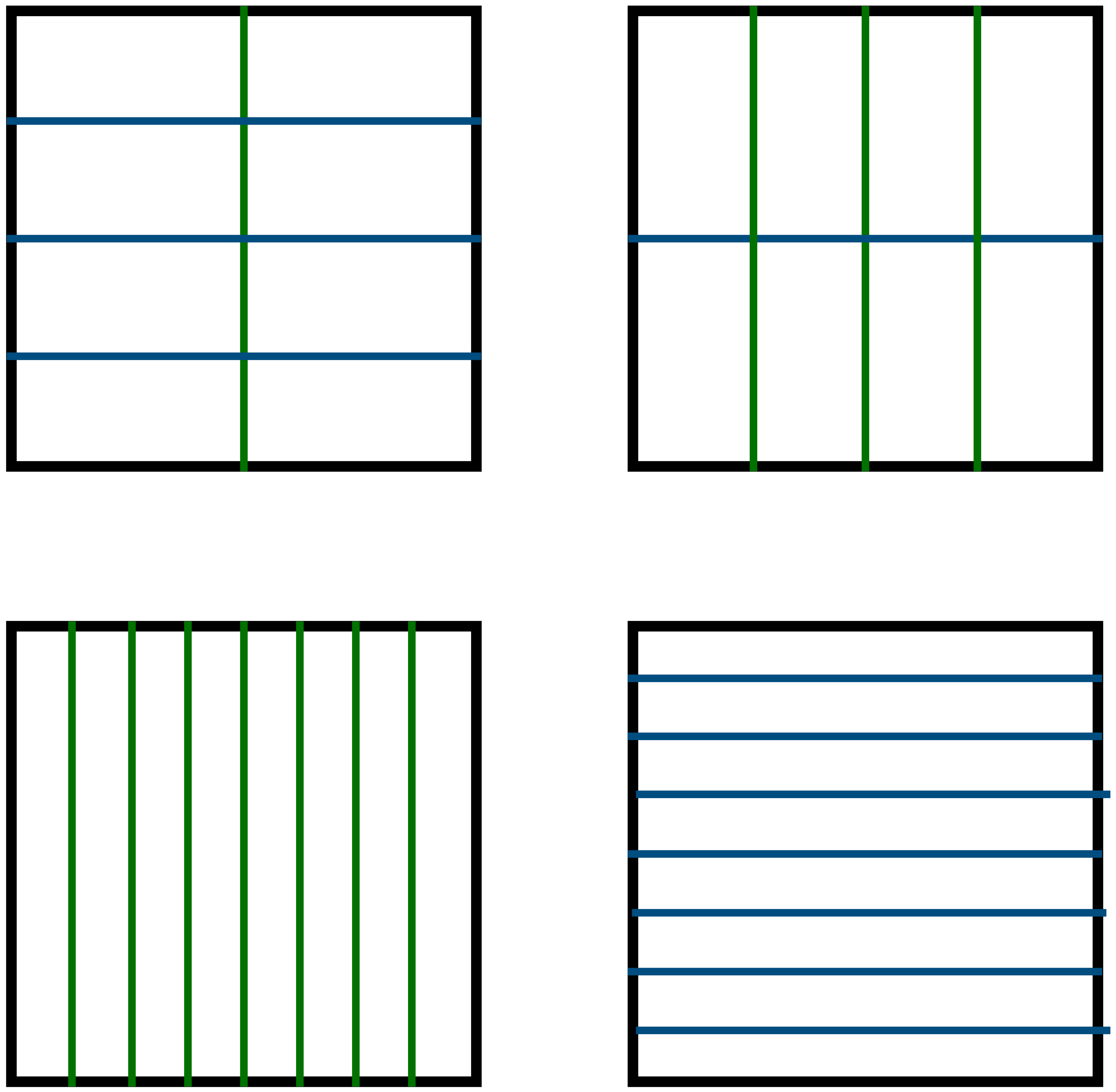
Designing generator matrices

enumerate all elementary intervals
(or even just subregions of the domain)

write down the constraints of different sub matrices

det(A) != 0 det(B) != 0 det(C) != 0 det(D) != 0

solve for the polynomial systems (in general NP hard, but there are many known solutions in number theory, and fast greedy approximation exists)



MatBuilder: Mastering Sampling Uniformity Over Projections

LOÏS PAULIN, Univ Lyon, UCBL, CNRS, INSA Lyon, LIRIS, France
NICOLAS BONNEEL, Univ Lyon, CNRS, INSA Lyon, UCBL, LIRIS, France
DAVID COEURJOLLY, Univ Lyon, CNRS, INSA Lyon, UCBL, LIRIS, France
JEAN-CLAUDE IEHL, Univ Lyon, UCBL, CNRS, INSA Lyon, LIRIS, France
ALEXANDER KELLER, NVIDIA, Germany
VICTOR OSTROMOUKHOV, Univ Lyon, UCBL, CNRS, INSA Lyon, LIRIS, France

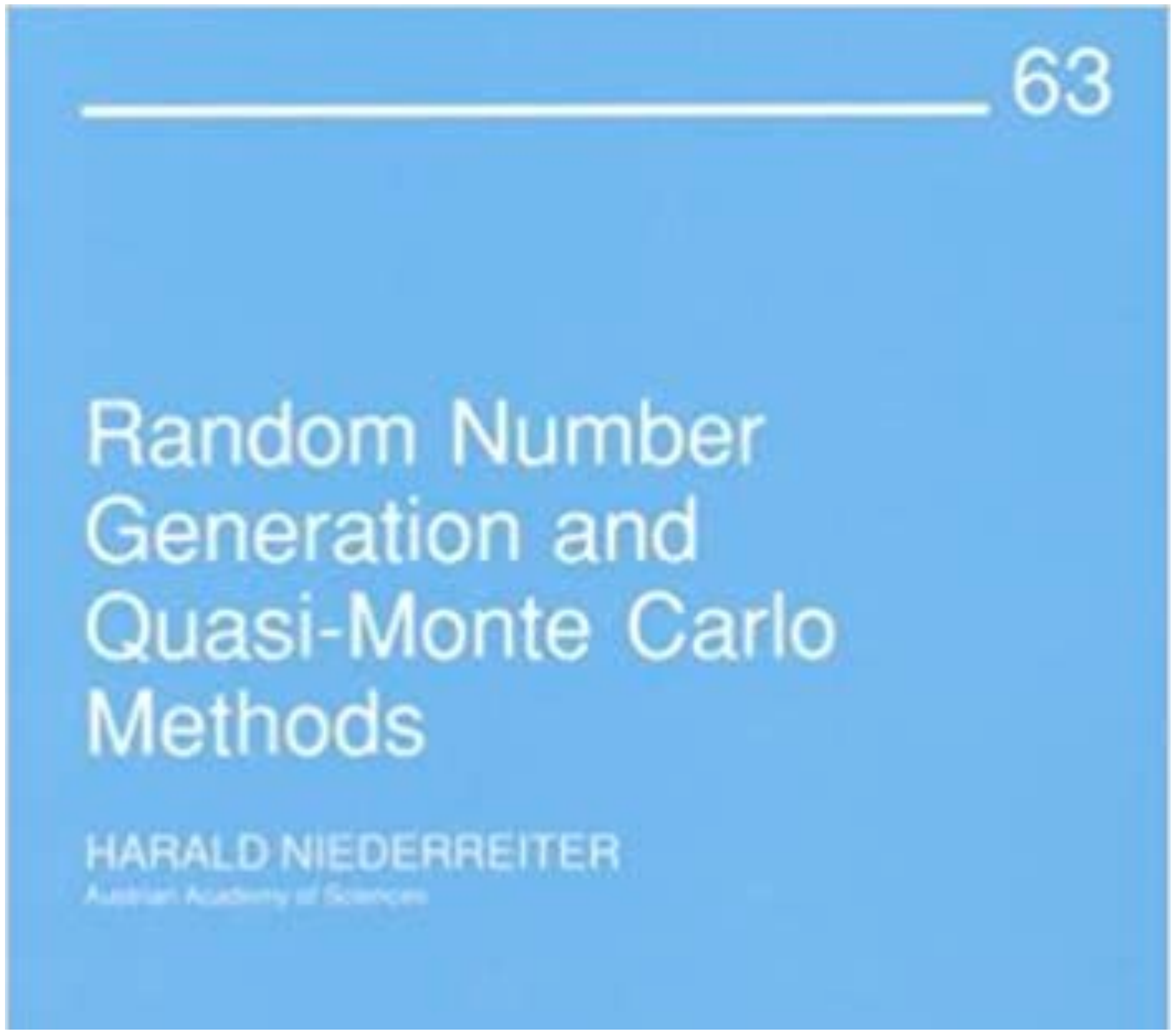
Example of generator matrices

Sobol's algorithm produces a (0,s)-sequence on base 2

base = 2, m = 6

$$C_1 = \begin{bmatrix} 1&0&0&0&0&0 \\ 0&1&0&0&0&0 \\ 0&0&1&0&0&0 \\ 0&0&0&1&0&0 \\ 0&0&0&0&1&0 \\ 0&0&0&0&0&1 \end{bmatrix} \qquad C_2 = \begin{bmatrix} 1&0&0&0&0&0 \\ 1&1&0&0&0&0 \\ 1&0&1&0&0&0 \\ 1&1&1&1&0&0 \\ 1&0&0&0&1&0 \\ 1&1&0&0&1&1 \end{bmatrix}$$

Some good references



Efficient Multidimensional Sampling

Thomas Kollig and Alexander Keller

Department of Computer Science, Kaiserslautern University, Germany

MatBuilder: Mastering Sampling Uniformity Over Projections

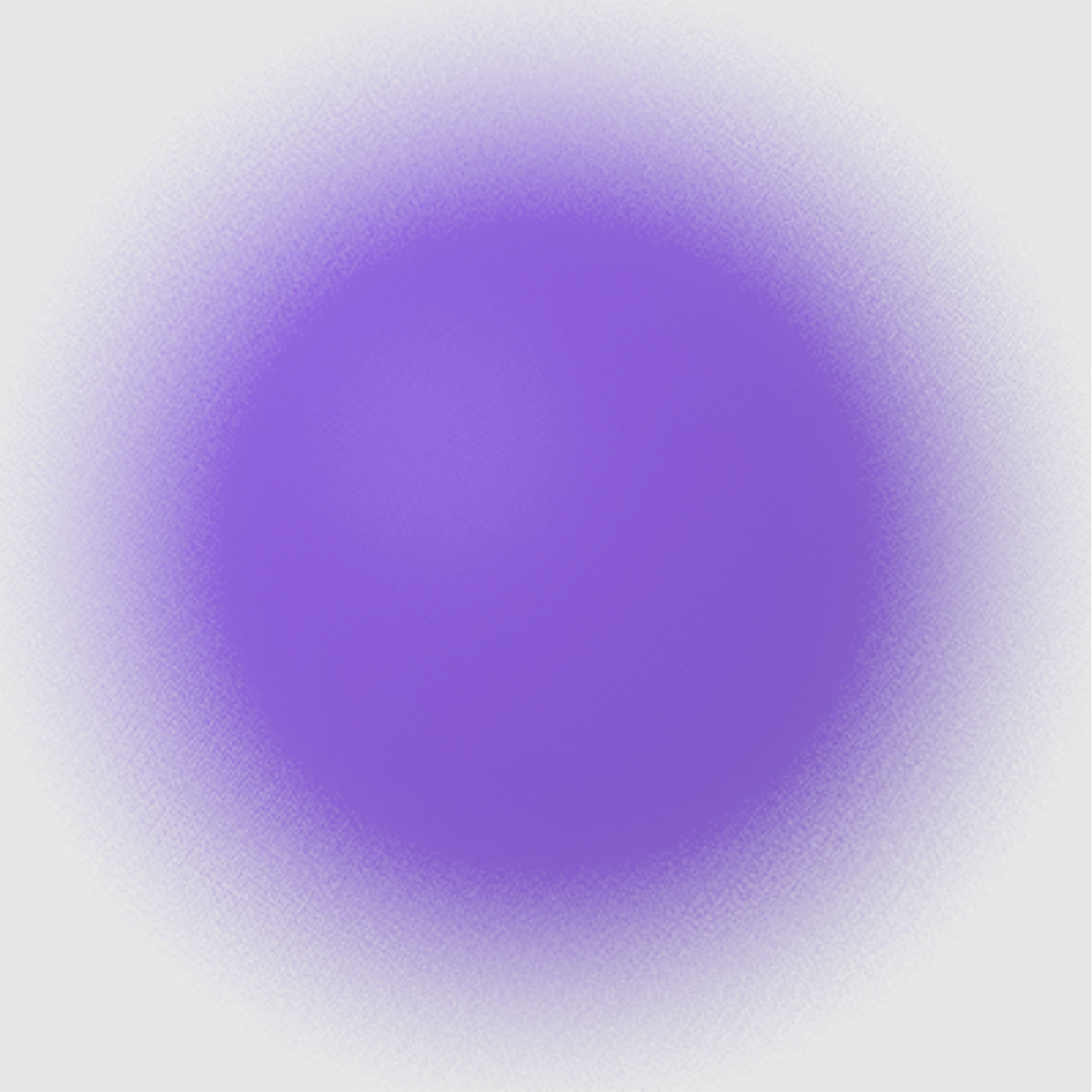
LOÏS PAULIN, Univ Lyon, UCBL, CNRS, INSA Lyon, LIRIS, France
NICOLAS BONNEEL, Univ Lyon, CNRS, INSA Lyon, UCBL, LIRIS, France
DAVID COEURJOLLY, Univ Lyon, CNRS, INSA Lyon, UCBL, LIRIS, France
JEAN-CLAUDE IEHL, Univ Lyon, UCBL, CNRS, INSA Lyon, LIRIS, France
ALEXANDER KELLER, NVIDIA, Germany
VICTOR OSTROMOUKHOV, Univ Lyon, UCBL, CNRS, INSA Lyon, LIRIS, France

Optimizing Dyadic Nets

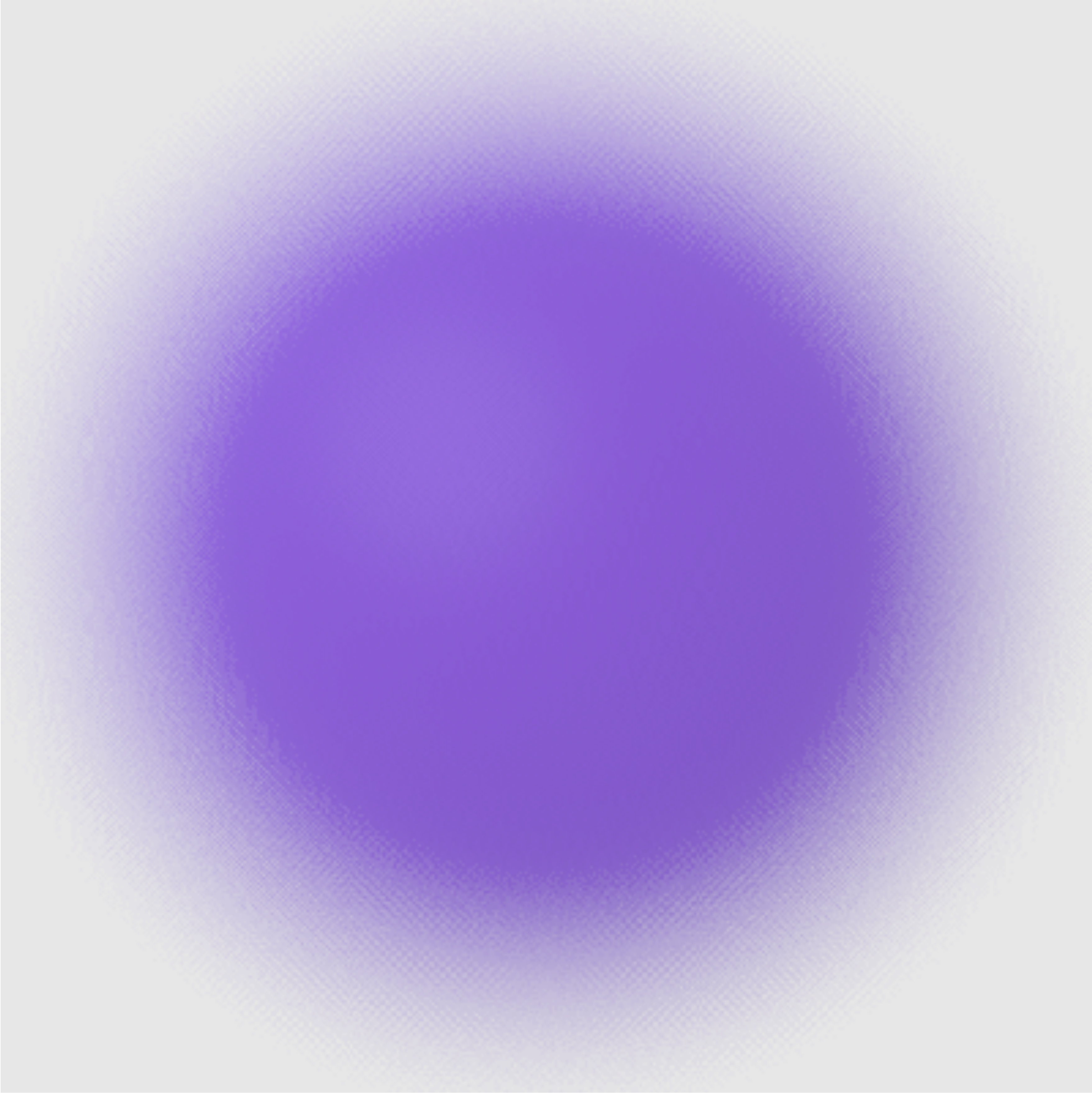
ABDALLA G. M. AHMED, KAUST, KSA
PETER WONKA, KAUST, KSA

Halton vs Sobol’

Sobol’ converges faster than Halton/Hammersley (due to the digital nets property), but introduces structural artifacts



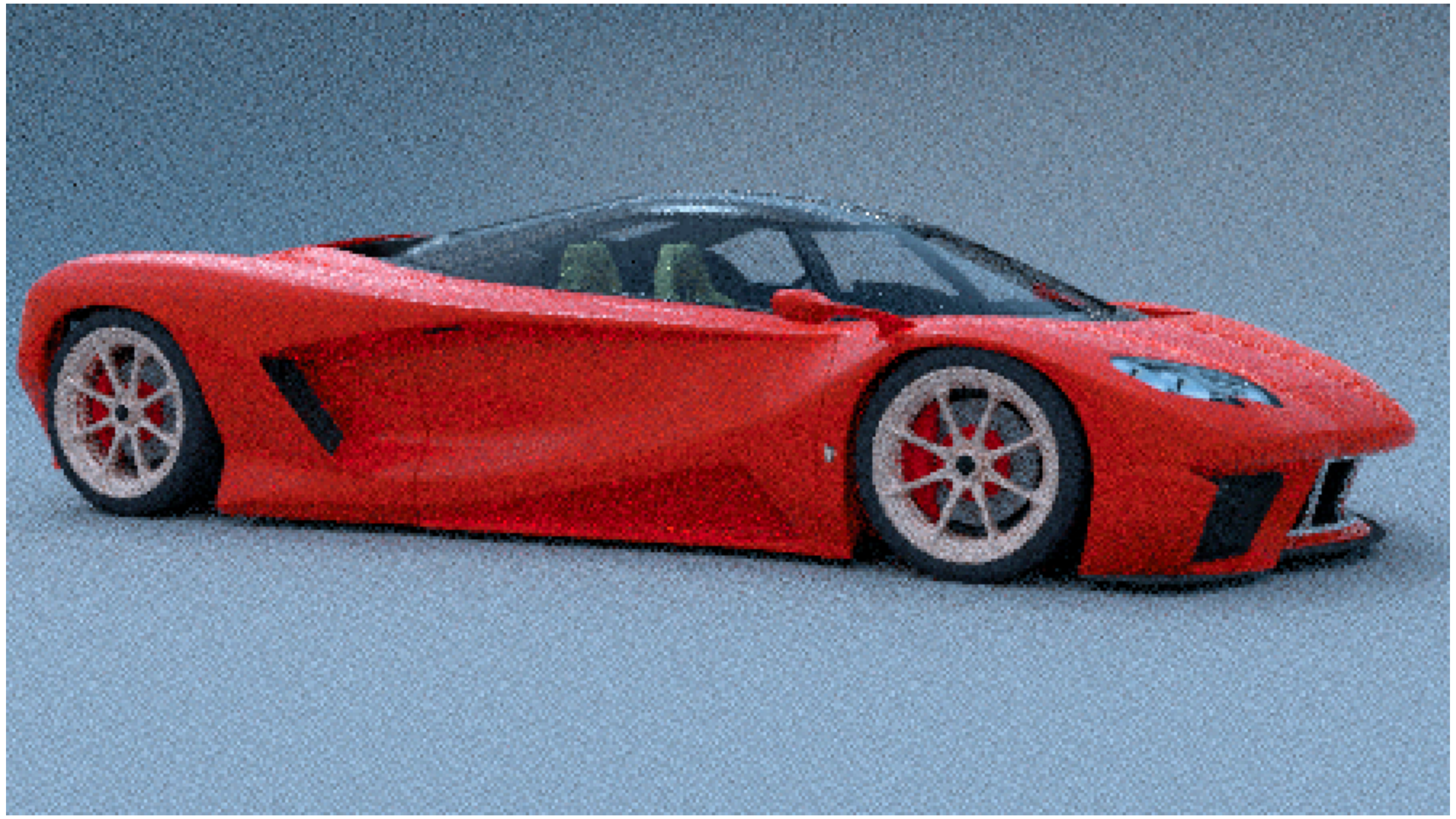
Halton



Sobol’

Halton vs Sobol′

Sobol′ converges faster than Halton/Hammersley (due to the digital nets property), but introduce structural artifacts



Halton

Sobol′

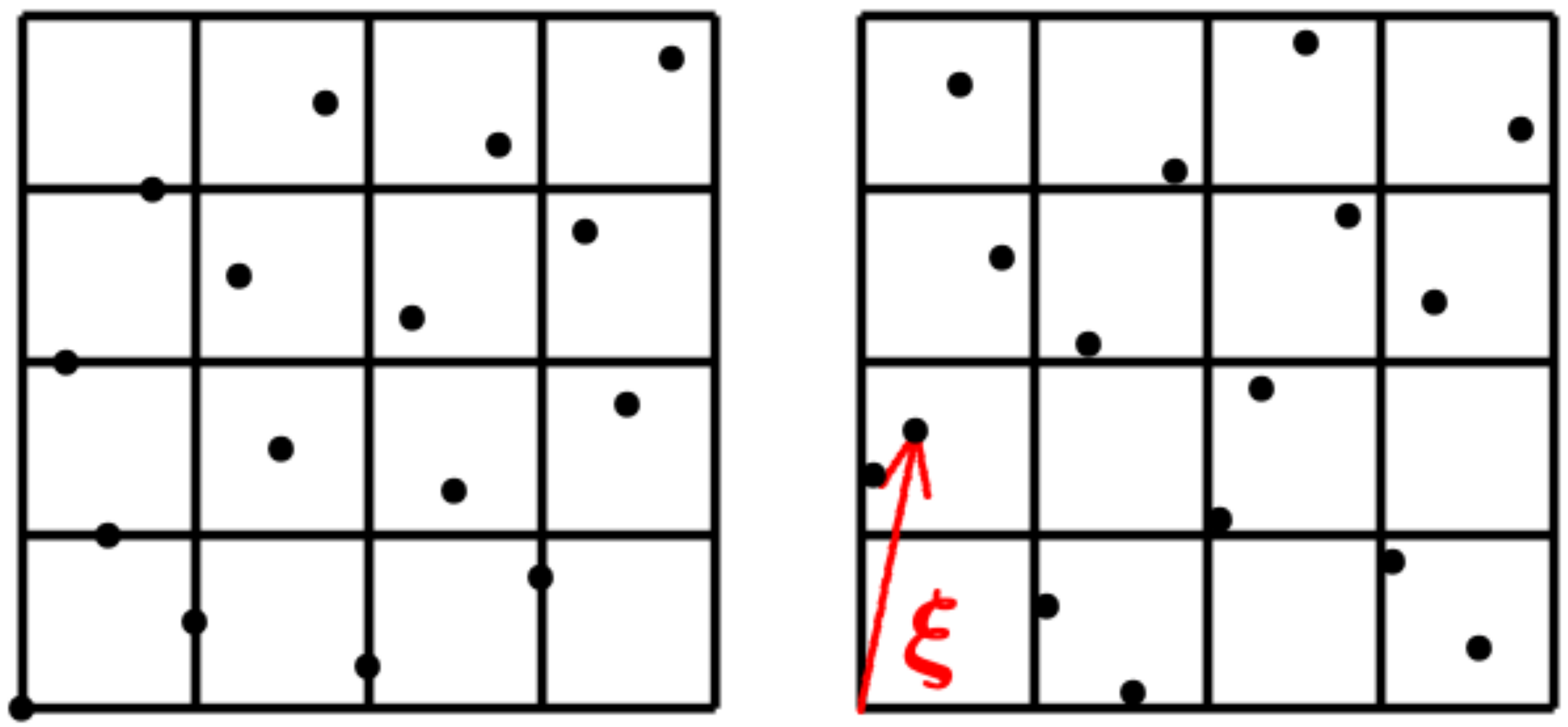
Avoiding structural artifacts in Sobol’ sampling

- Cranley-Patterson rotation
- Owen scrambling

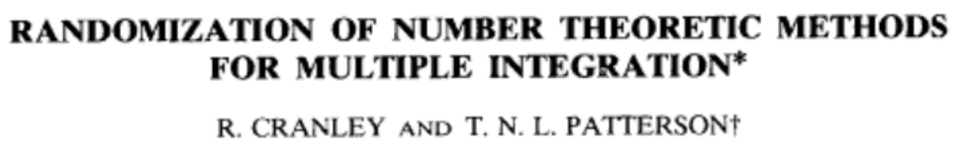
Avoiding structural artifacts in Sobol' sampling

- Cranley-Patterson rotation
- Owen scrambling

add a global random shift to all points in the sequence



can degrade uniformity a little bit


RANDOMIZATION OF NUMBER THEORETIC METHODS FOR MULTIPLE INTEGRATION*

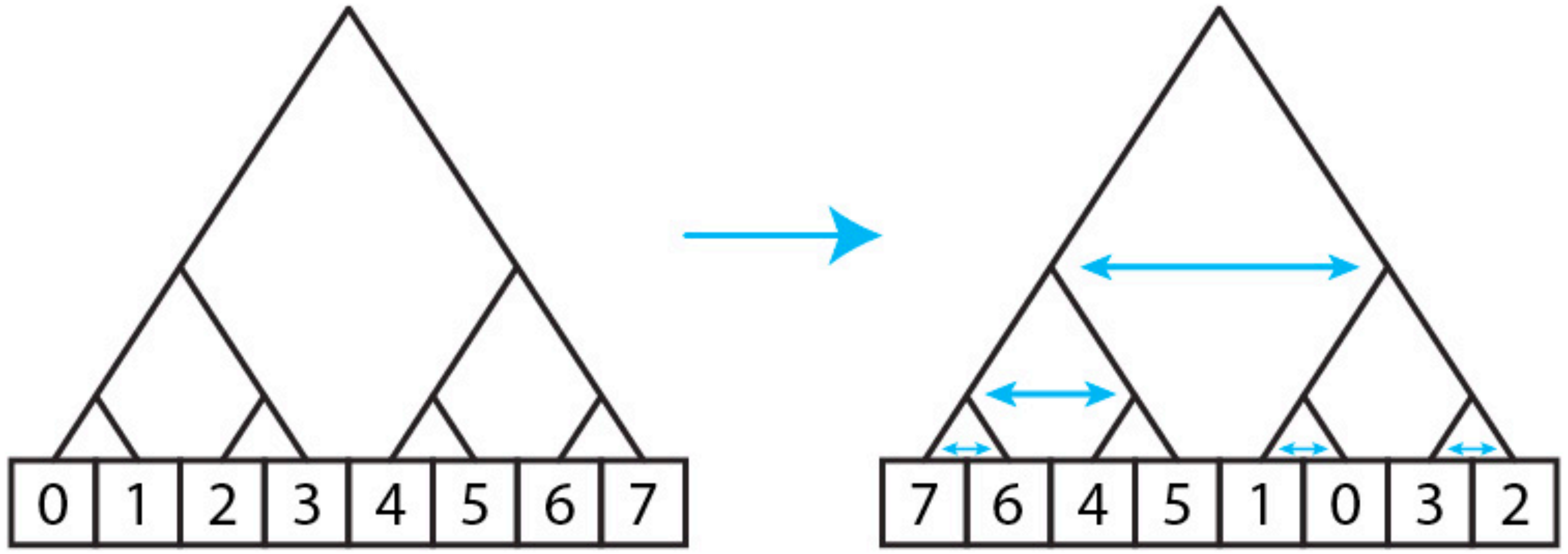
R. CRANLEY AND T. N. L. PATTERSON†

https://www.uni-kl.de/AG-Heinrich/EMS.pdf

Avoiding structural artifacts in Sobol′ sampling

- Cranley-Patterson rotation
- Owen scrambling

hierarchically and randomly scramble the elementary intervals
(similar to Faure's scrambling for Halton sequence)



provably preserves digital nets property and discrepancy!!

https://andrew-helmer.github.io/tree-shuffling/

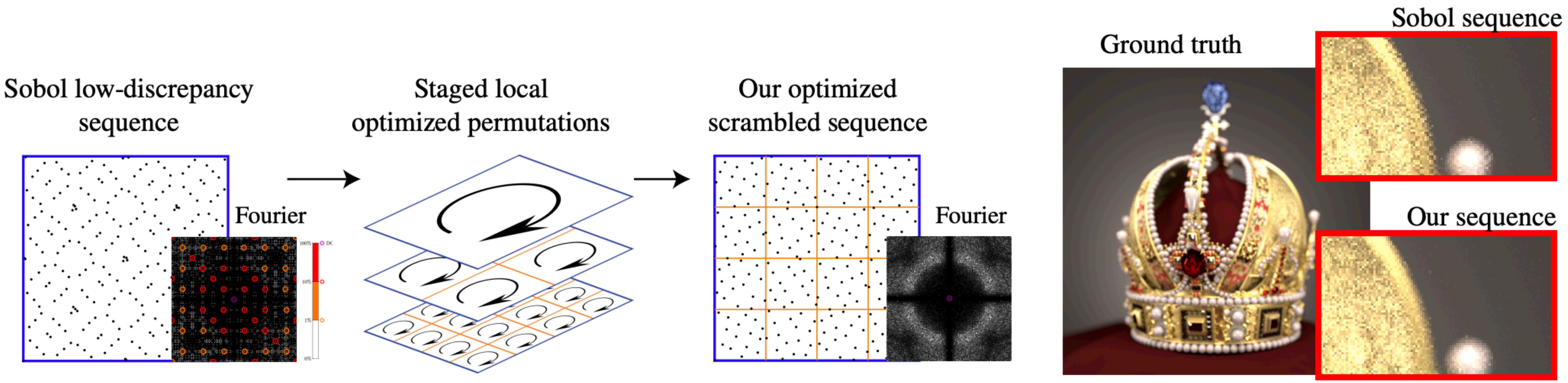
Blue noise + Sobol’

can be done by hacking Owen’s scrambling or the generation matrices

Sequences with Low-Discrepancy Blue-Noise 2-D Projections

Hélène Perrier[1] David Coeurjolly[1] Feng Xie[2] Matt Pharr[3] Pat Hanrahan[2] Victor Ostromoukhov[1]

[1]Université de Lyon, CNRS, LIRIS, France [2]Stanford, USA [3] Google, USA



Incorporation of blue noise in digital nets is still limiting

Low-Discrepancy Blue Noise Sampling

Abdalla G. M. Ahmed[*1] Hélène Perrier [2] David Coeurjolly[2] Victor Ostromoukhov[2]
Jianwei Guo[3] Dong-Ming Yan[3] Hui Huang [4,5] Oliver Deussen[1,5]

[1]University of Konstanz [2]Université de Lyon
[3]NLPR, Institute of Automation, CAS [4]Shenzhen University [5]SIAT

Optimizing Dyadic Nets

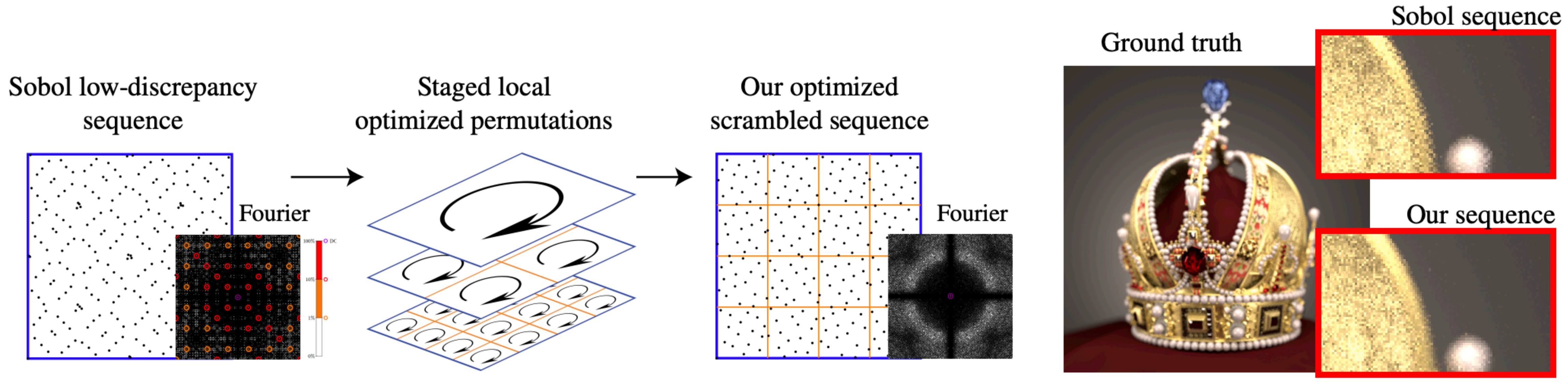
ABDALLA G. M. AHMED, KAUST, KSA
PETER WONKA, KAUST, KSA

mostly applies to 2D

Sequences with Low-Discrepancy Blue-Noise 2-D Projections

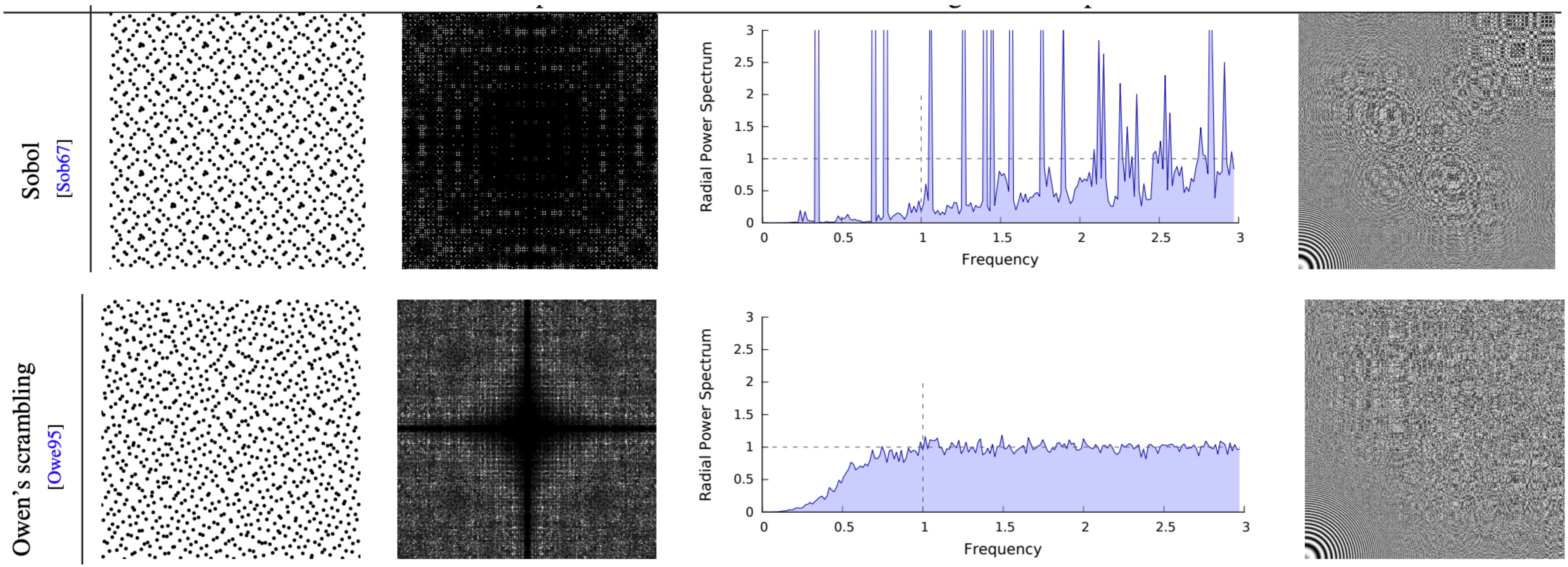
Hélène Perrier[1] David Coeurjolly[1] Feng Xie[2] Matt Pharr[3] Pat Hanrahan[2] Victor Ostromoukhov[1]

[1]Université de Lyon, CNRS, LIRIS, France [2]Stanford, USA [3] Google, USA



needs the # of samples to be power of 16

Power spectrum of Sobol' sampling



Sequences with Low-Discrepancy Blue-Noise 2-D Projections

Hélène Perrier[1] David Coeurjolly[1] Feng Xie[2] Matt Pharr[3] Pat Hanrahan[2] Victor Ostromoukhov[1]

[1]Université de Lyon, CNRS, LIRIS, France [2]Stanford, USA [3] Google, USA

2019

Blue-noise Low-Discrepancy Sequences

Perrier et al.

Sequences with Low-Discrepancy Blue-Noise 2-D Projections

Hélène Perrier[1] David Coeurjolly[1] Feng Xie[2] Matt Pharr[3] Pat Hanrahan[2] Victor Ostromoukhov[1]

[1]Université de Lyon, CNRS, LIRIS, France [2]Stanford, USA [3] Google, USA

2019

Connection to optimal transport/ Wasserstein distance

- Rubinstein-Kantorovich theorem

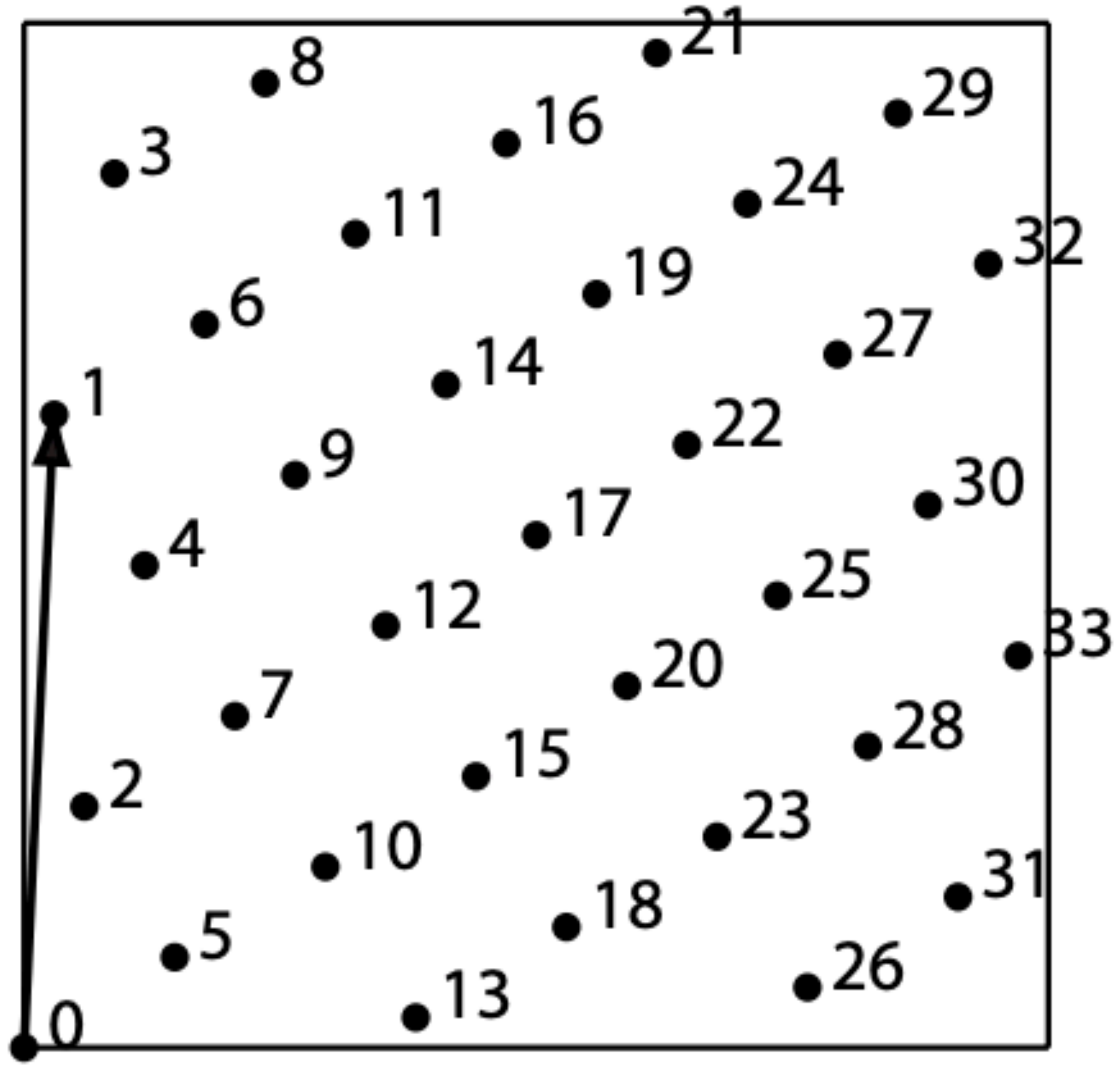
$$\left| \int f(x)\, dx - \frac{1}{n} \sum_{i=1}^{n} f(x^i) \right| \leq \mathrm{Lip}(f) \,.\, W_1(X, 1_\Omega),$$

instead of using discrepancies, measure the earth mover distance

Sliced Optimal Transport Sampling

LOIS PAULIN, Univ Lyon, CNRS
NICOLAS BONNEEL, Univ Lyon, CNRS
DAVID COEURJOLLY, Univ Lyon, CNRS
JEAN-CLAUDE IEHL, Univ Lyon, CNRS
ANTOINE WEBANCK, Univ Lyon, CNRS
MATHIEU DESBRUN, ShanghaiTech/Caltech
VICTOR OSTROMOUKHOV, Univ Lyon, CNRS

Rank-1 lattice



Rank-1 Lattices for Efficient Path Integral Estimation

Hongli Liu[1] and Honglei Han[1] † and Min Jiang[2]

[1]State Key Laboratory of Media Convergence and Communication,
and School of Animation and Digital Arts, Communication University of China, China
[2]Framestore, United Kingdom

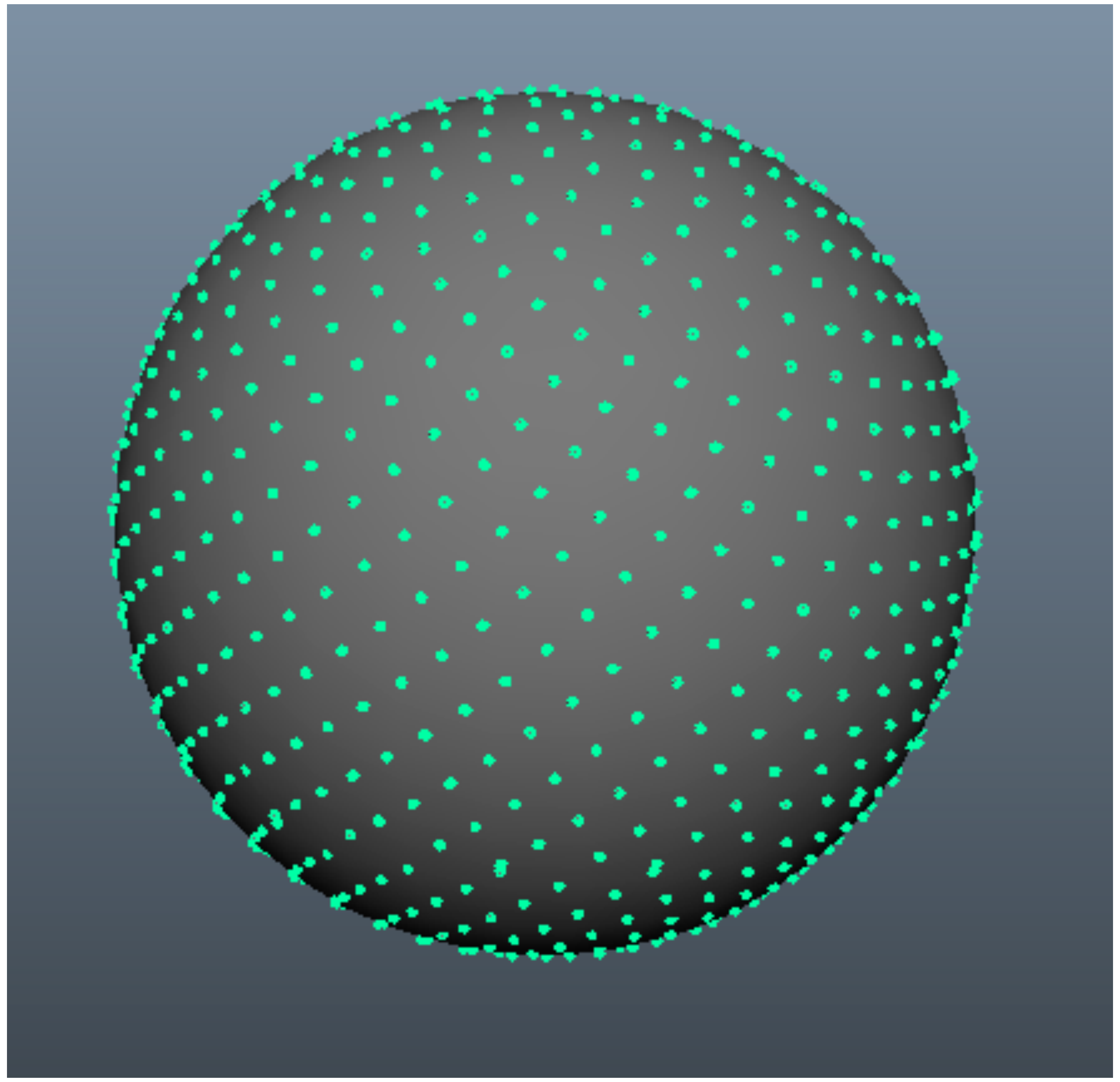
2021

Image Synthesis by Rank-1 Lattices

S. Dammertz and A. Keller
Institute of Media Informatics, Ulm University, Germany

2006

Spherical Fibonacci lattice



Spherical Fibonacci Point Sets for Illumination Integrals

R. Marques[1], C. Bouville[2], M. Ribardière[2], L. P. Santos[3] and K. Bouatouch[2]

[1]INRIA Rennes, France
[2]IRISA Rennes, France
[3]Universidade do Minho, Braga, Portugal

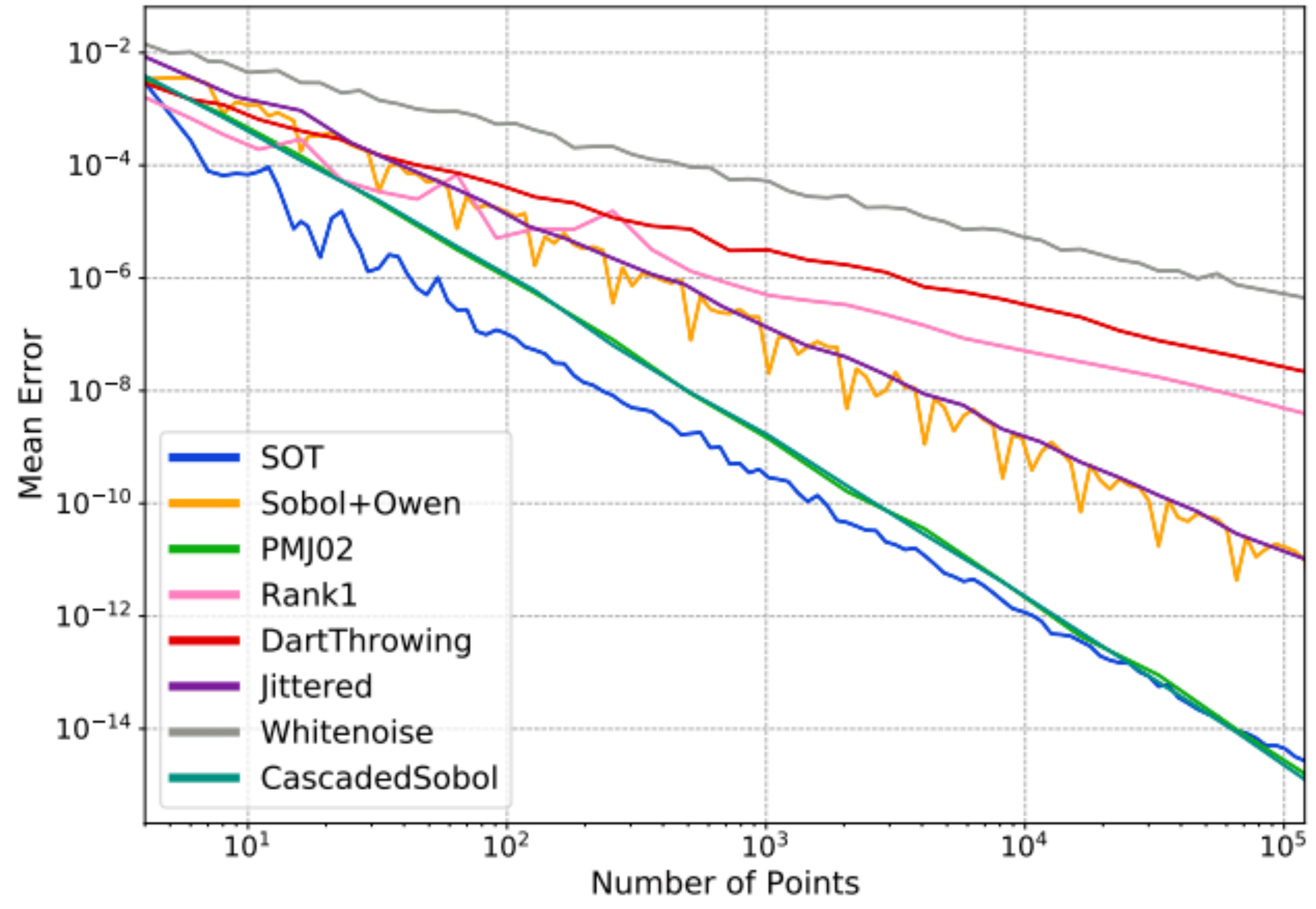
https://math.stackexchange.com/questions/3291489/can-the-fibonacci-lattice-be-extended-to-dimensions-higher-than-3

Comparison between low-discrepancy sequences and jittering

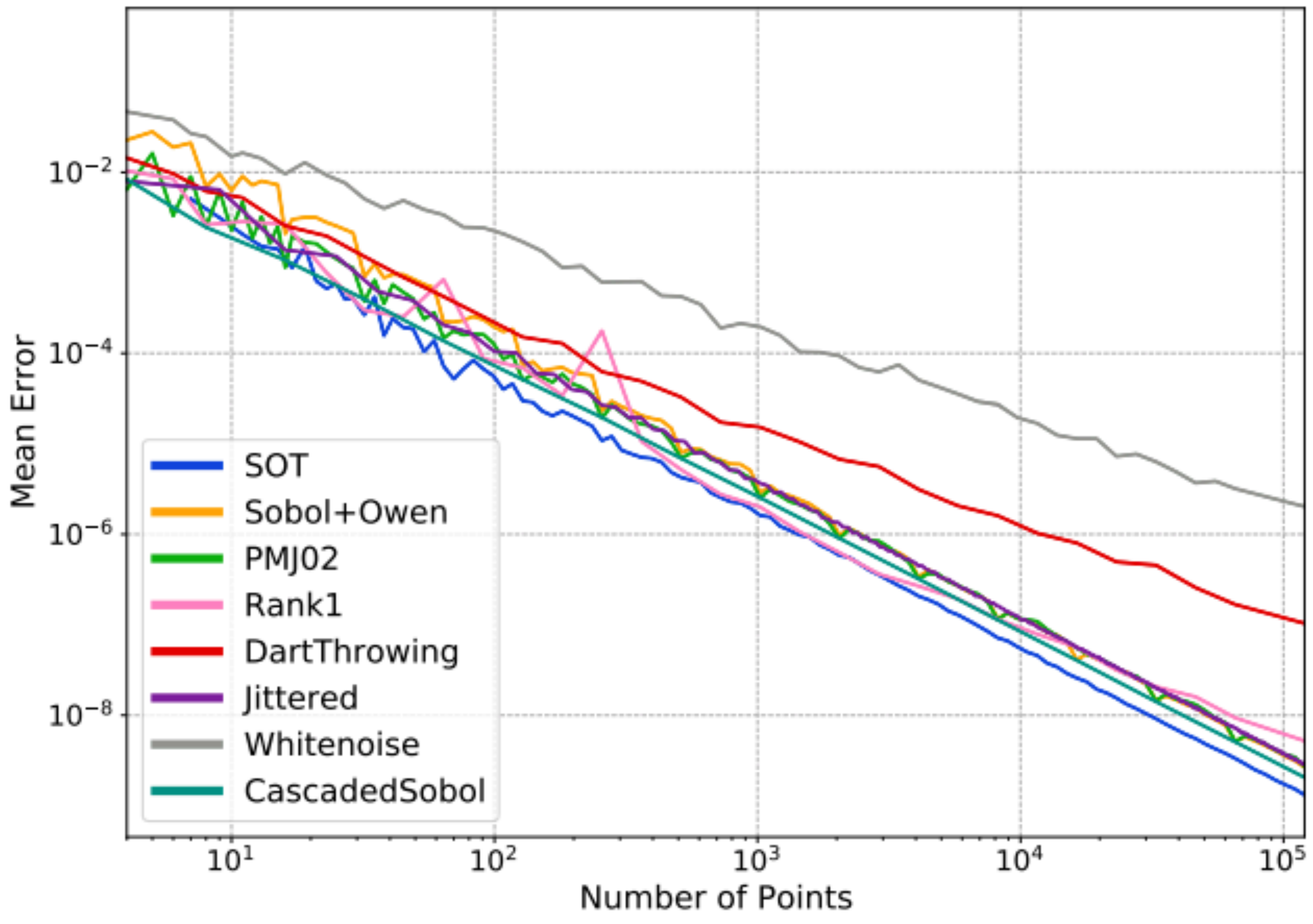
at low dimension, for smooth integrals, digital nets often outperform jittering
(but extensions of progressive multi-jittering, PMJ02, is as good at 2D)

$s = 2$

Gaussian integrands



Heaviside integrands

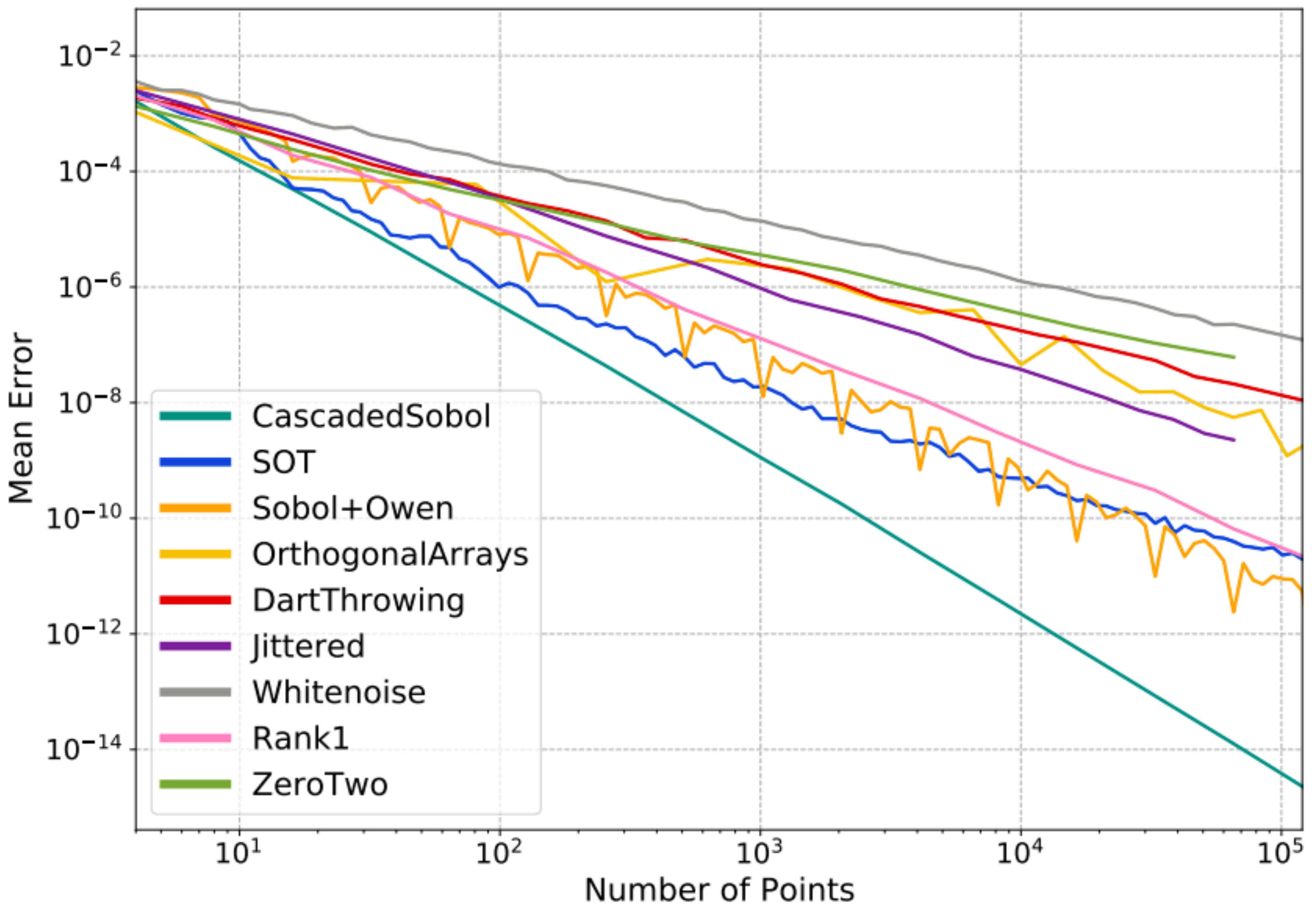


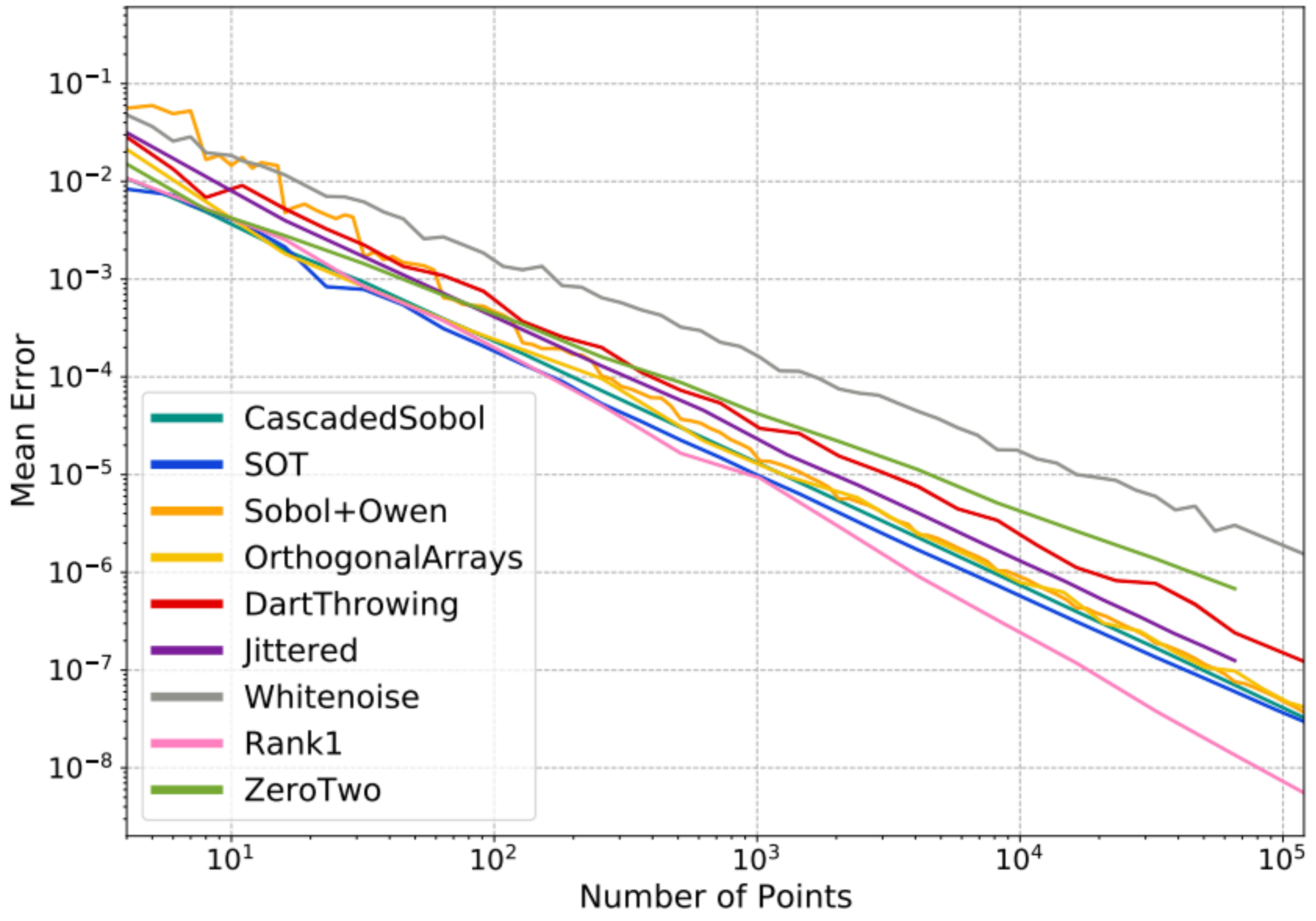
https://perso.liris.cnrs.fr/david.coeurjolly/publication/cascaded2021/Cascaded2021.pdf

Comparison between low-discrepancy sequences and jittering

at mid dimension, for smooth integrals, digital nets still often outperform jittering
(PMJ02 requires uncorrelated jittering to work and is less effective)

$s = 4$



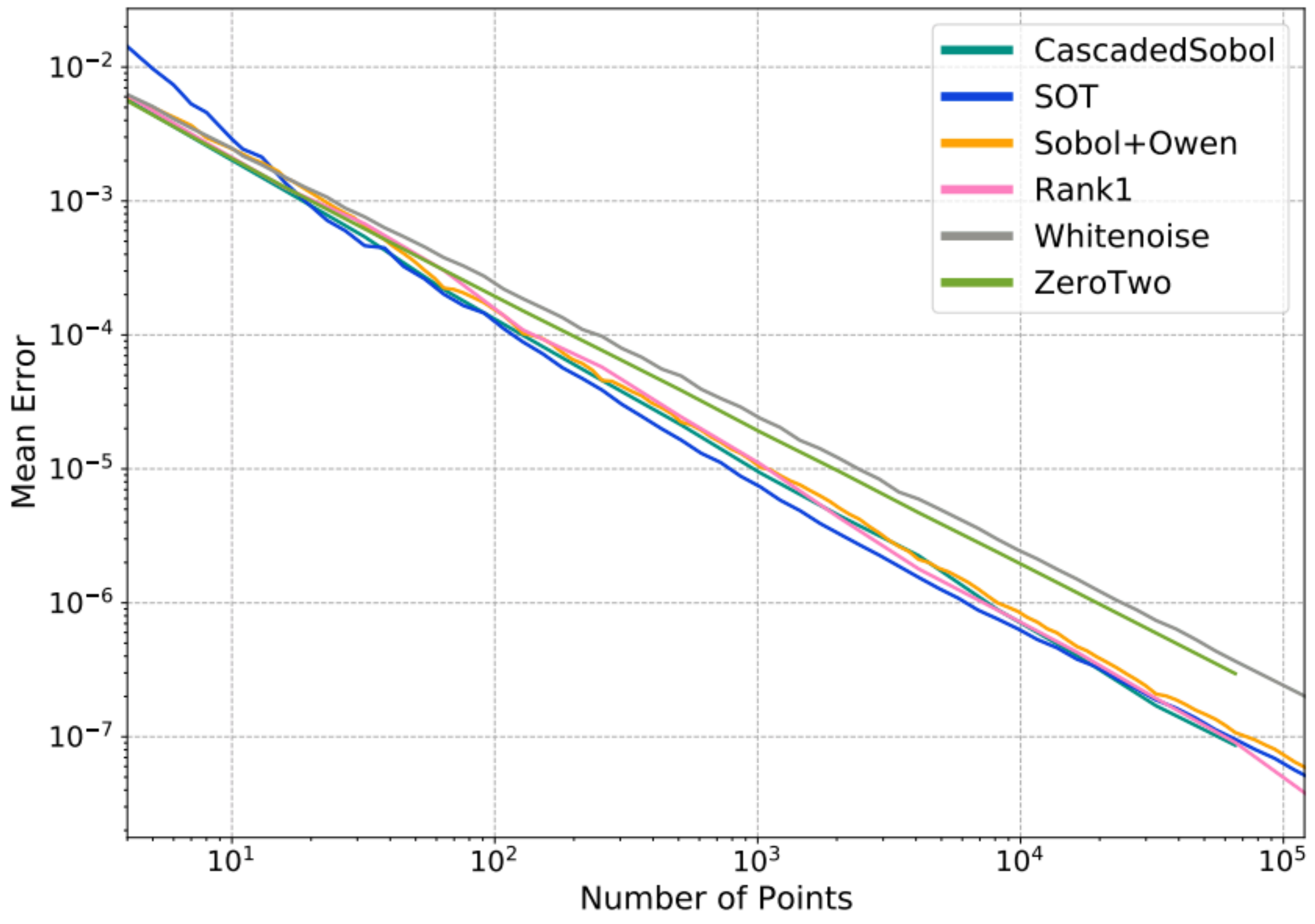


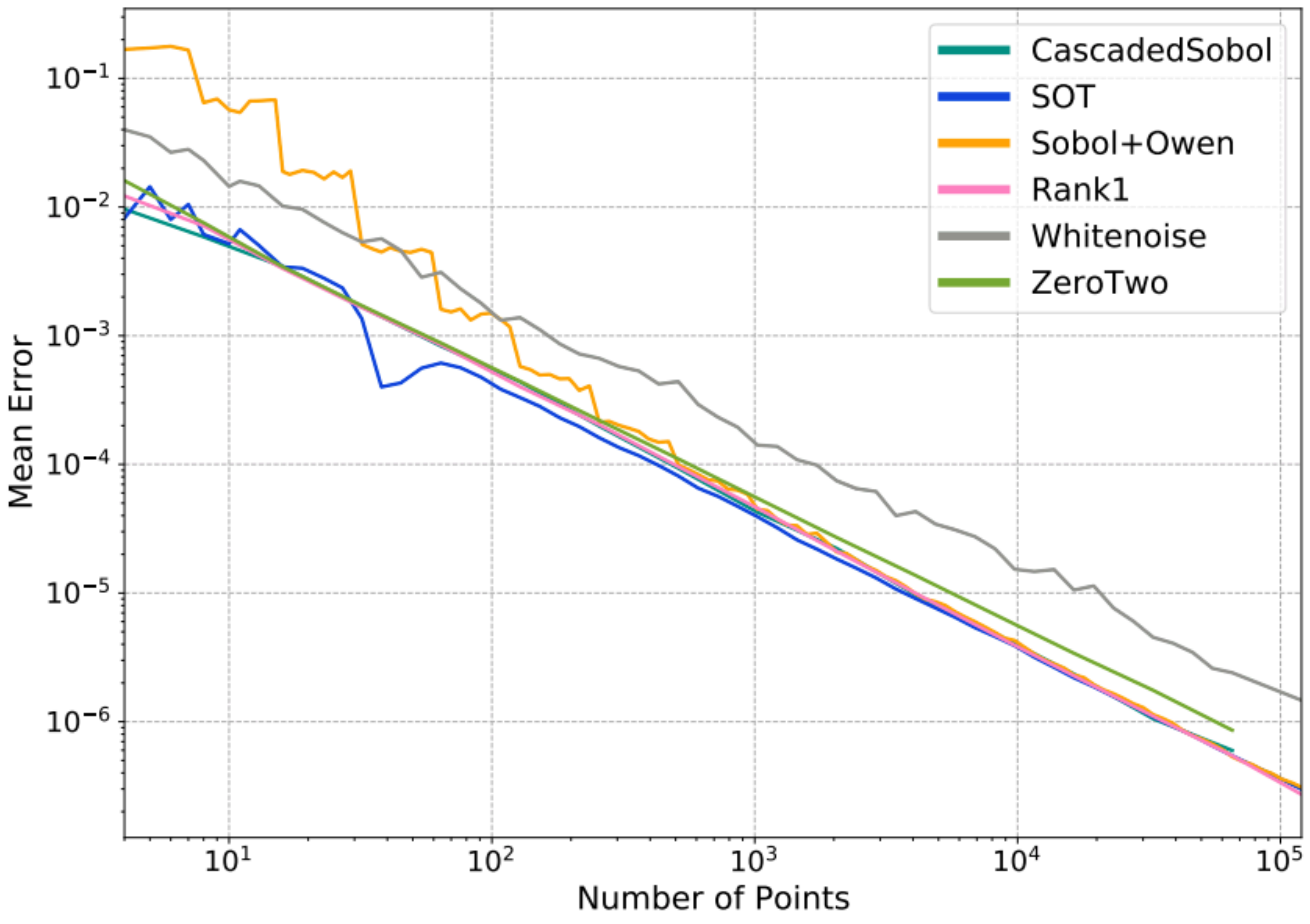
https://perso.liris.cnrs.fr/david.coeurjolly/publication/cascaded2021/Cascaded2021.pdf

Comparison between low-discrepancy sequences and jittering

at high dimension, all methods are similar, except independent white noise

$s = 20$





https://perso.liris.cnrs.fr/david.coeurjolly/publication/cascaded2021/Cascaded2021.pdf

So, which sample sequence should we use?

- PMJ is much easier to combine with blue noise, so it has better perceptual quality
- Digital nets can converge faster in mid dimensional smooth problems (e.g., 4-8D)
- For high-dimensional problems (>10D), you are good as long as you don’t use white noise

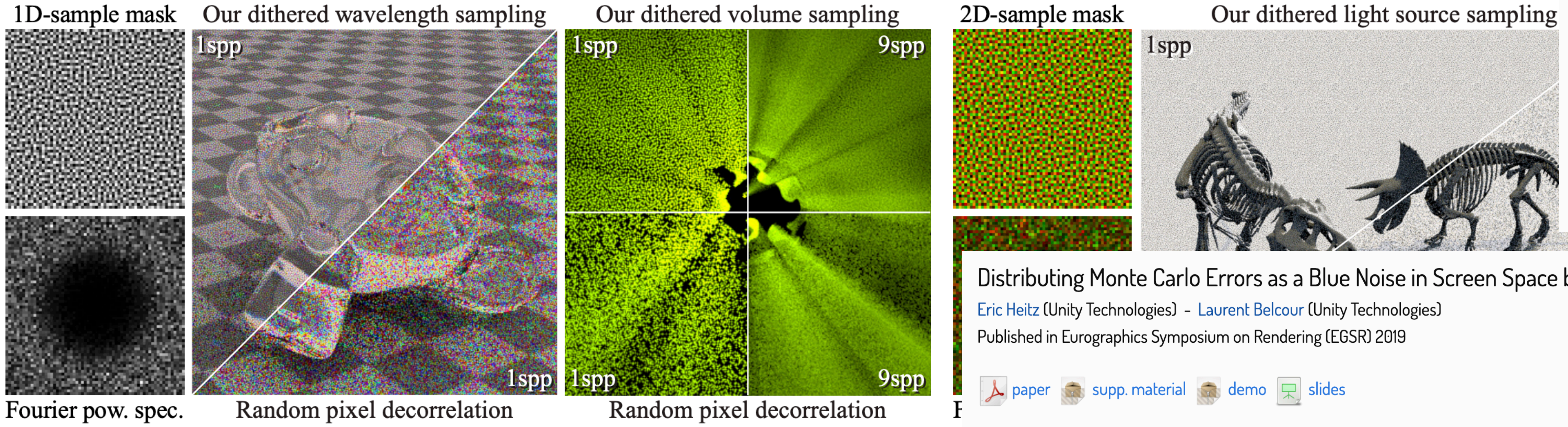
Related topic: blue-noise dithered sampling

- focus on the reconstruction properties of sampling patterns, instead of integration

often by optimizing the Cranley-Patterson rotation offset in preprocessing

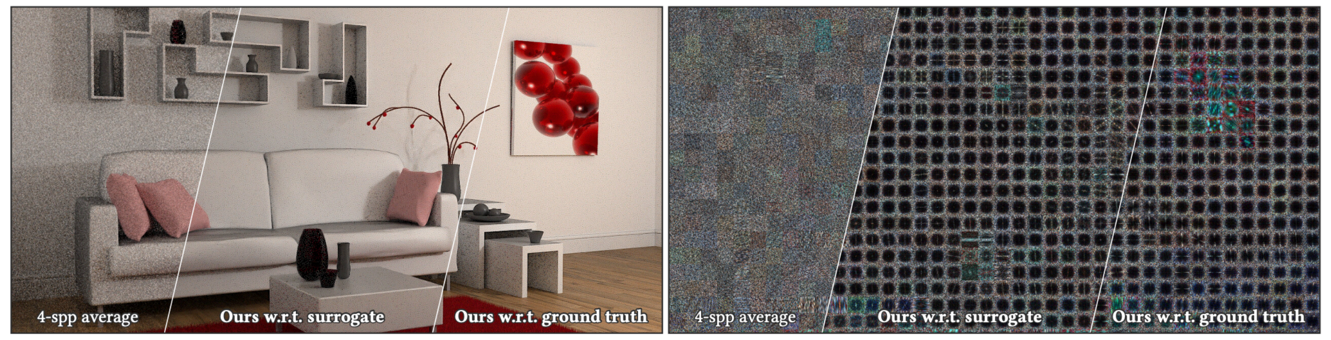
Blue-noise Dithered Sampling

Iliyan Georgiev, Solid Angle
Marcos Fajardo, Solid Angle



Perceptual error optimization for Monte Carlo rendering

VASSILLEN CHIZHOV, MIA Group Saarland University, Max-Planck-Institut für Informatik, Germany
ILIYAN GEORGIEV, Autodesk, United Kingdom
KAROL MYSZKOWSKI, Max-Planck-Institut für Informatik, Germany
GURPRIT SINGH, Max-Planck-Institut für Informatik, Germany

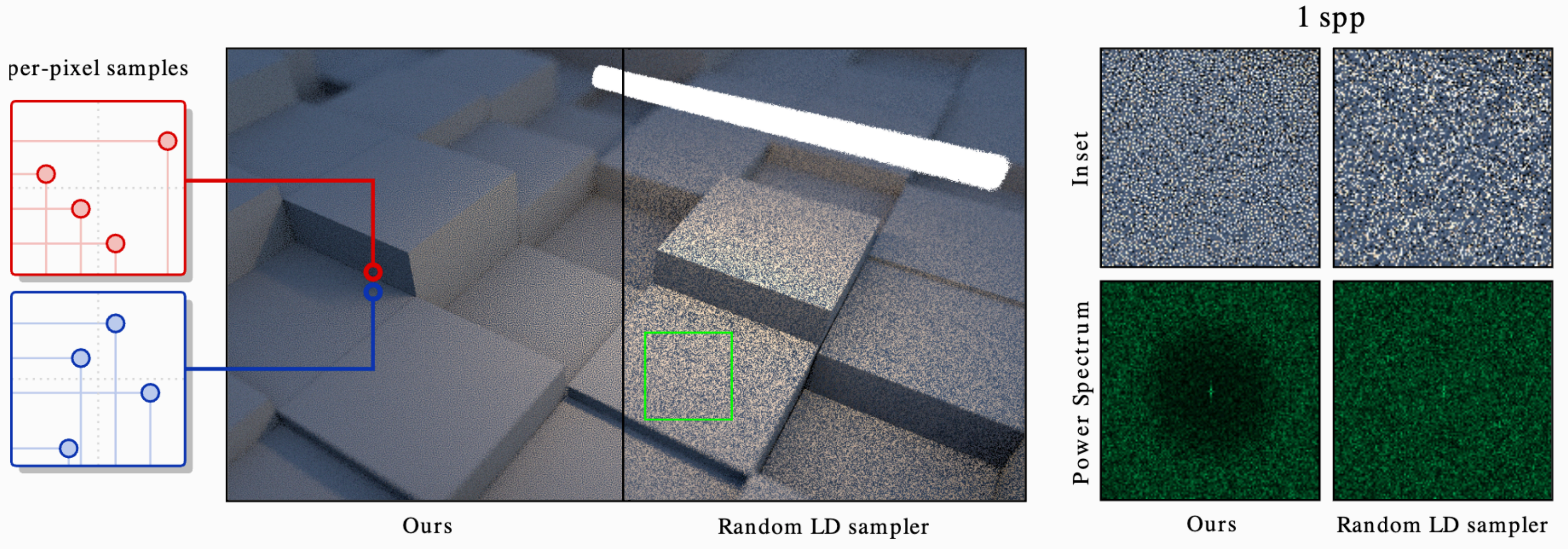


A Low-Discrepancy Sampler that Distributes Monte Carlo Errors as a Blue Noise in Screen Space

Eric Heitz (Unity Technologies) - Laurent Belcour (Unity Technologies) - Victor Ostromoukhov - David Coeurjolly - Jean-Claude Iehl (LIRIS)

Published in ACM SIGGRAPH Talk 2019

paper bib supp. code supp. doc unity demo supp. html video slides



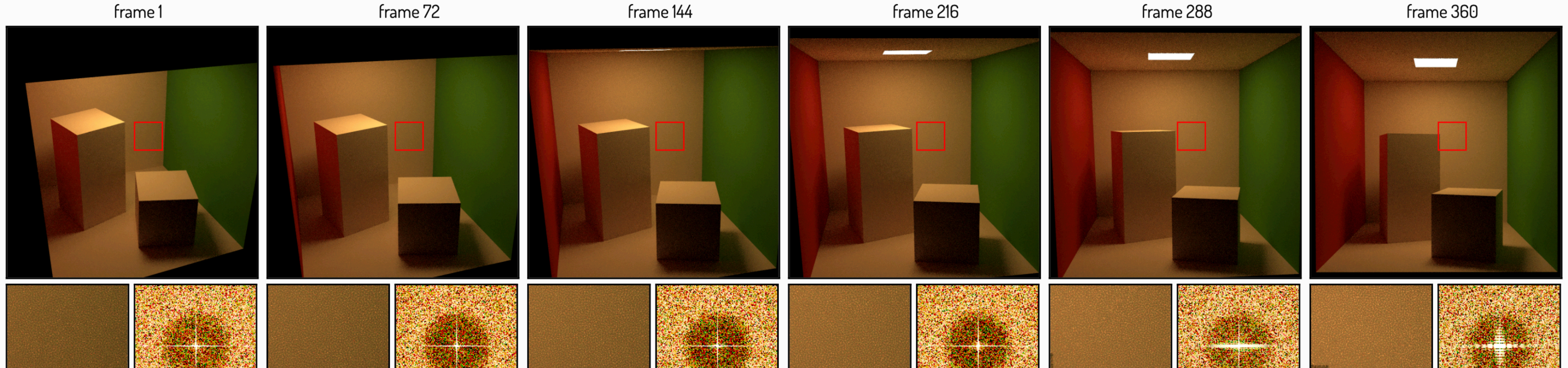
Project Abstract

Distributing Monte Carlo Errors as a Blue Noise in Screen Space by Permuting Pixel Seeds Between Frames

Eric Heitz (Unity Technologies) - Laurent Belcour (Unity Technologies)

Published in Eurographics Symposium on Rendering (EGSR) 2019

paper supp. material demo slides



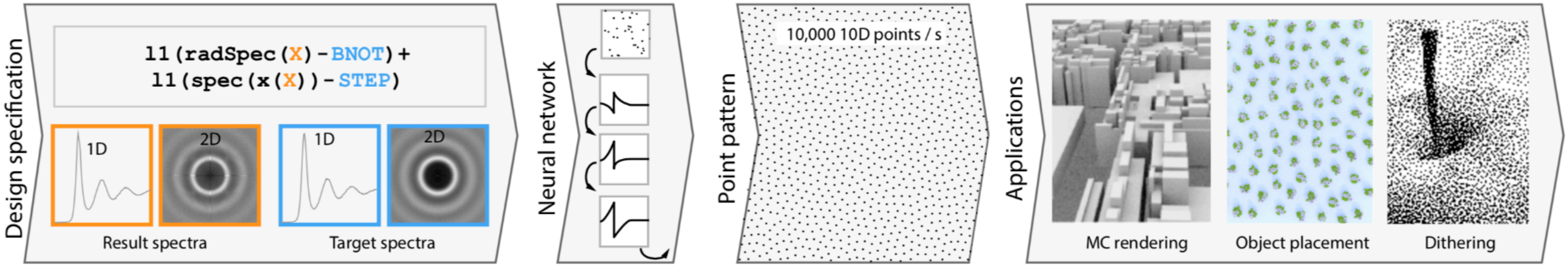
Neural nets for point sampling

Deep Point Correlation Design

Thomas Leimkühler[1], Gurprit Singh[1], Karol Myszkowski[1], Hans-Peter Seidel[1], Tobias Ritschel[2]

[1]Max Planck Institute for Informatics, Saarbrücken, [2]University College London, UK

In ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2019, Volume 38 issue 6)



Next: path-space & Eric Veach

next Monday is holiday!

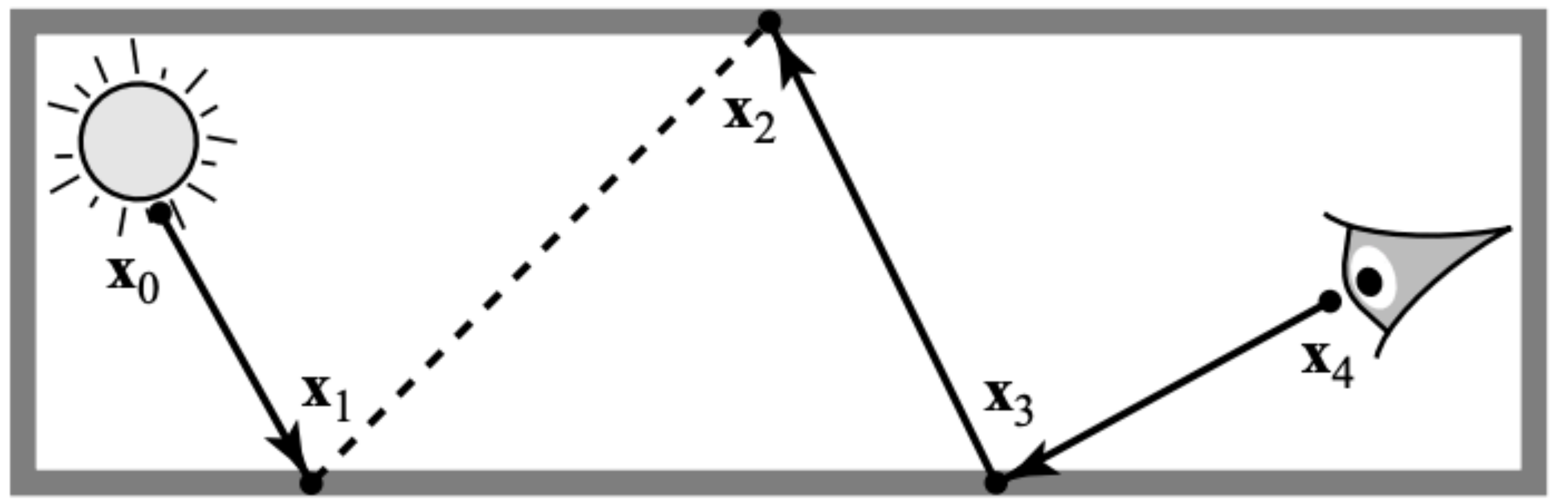
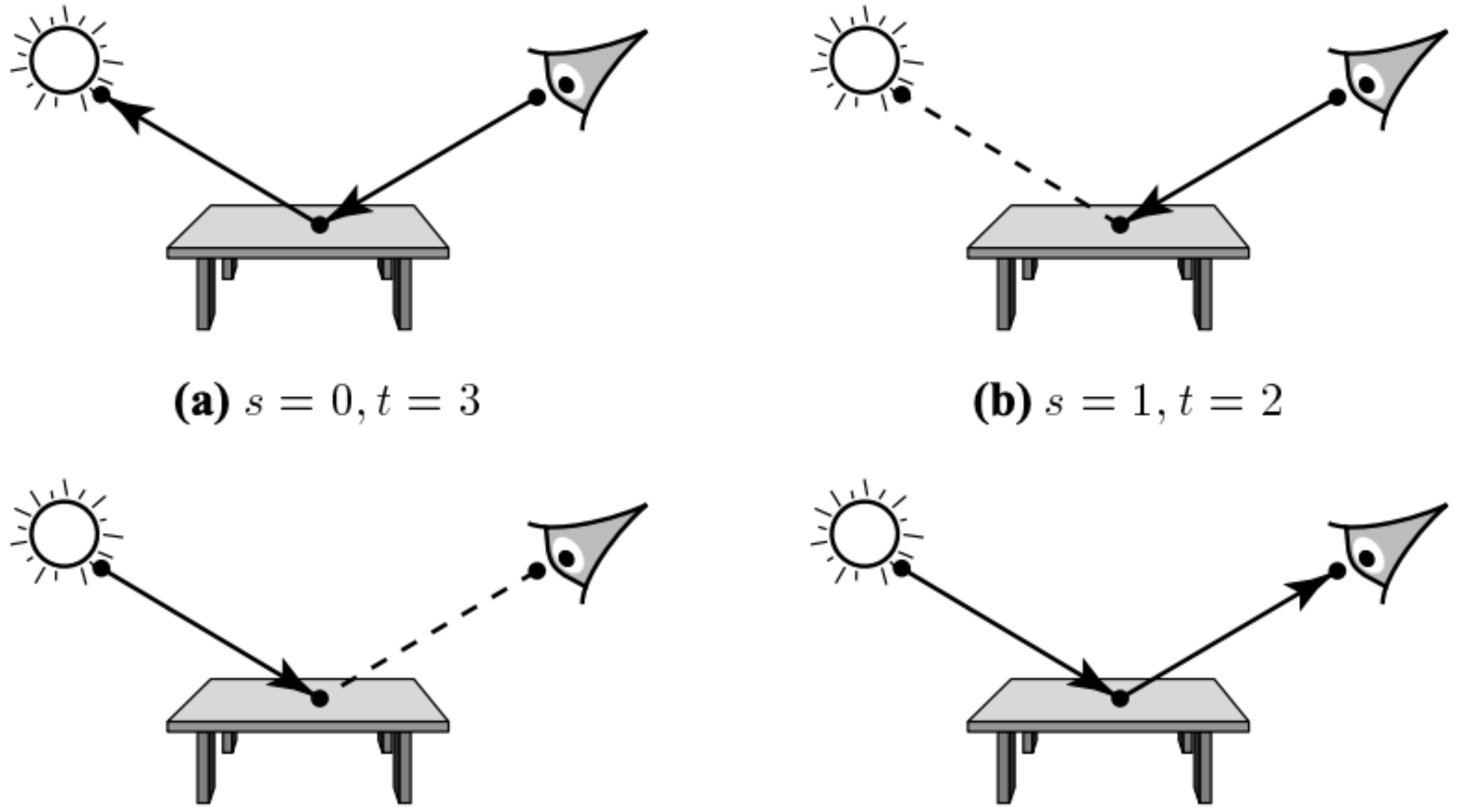


Figure 10.1: A transport path from a light source to the camera lens, created by concat
ing two separately generated pieces.



(a) $s = 0, t = 3$ **(b)** $s = 1, t = 2$

(c) $s = 2, t = 1$ **(d)** $s = 3, t = 0$

ROBUST MONTE CARLO METHODS
FOR LIGHT TRANSPORT SIMULATION

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

by
Eric Veach
December 1997