AN OVERVIEW OF QUASI-RANDOM SEQUENCES / SETS

Colas Schretter

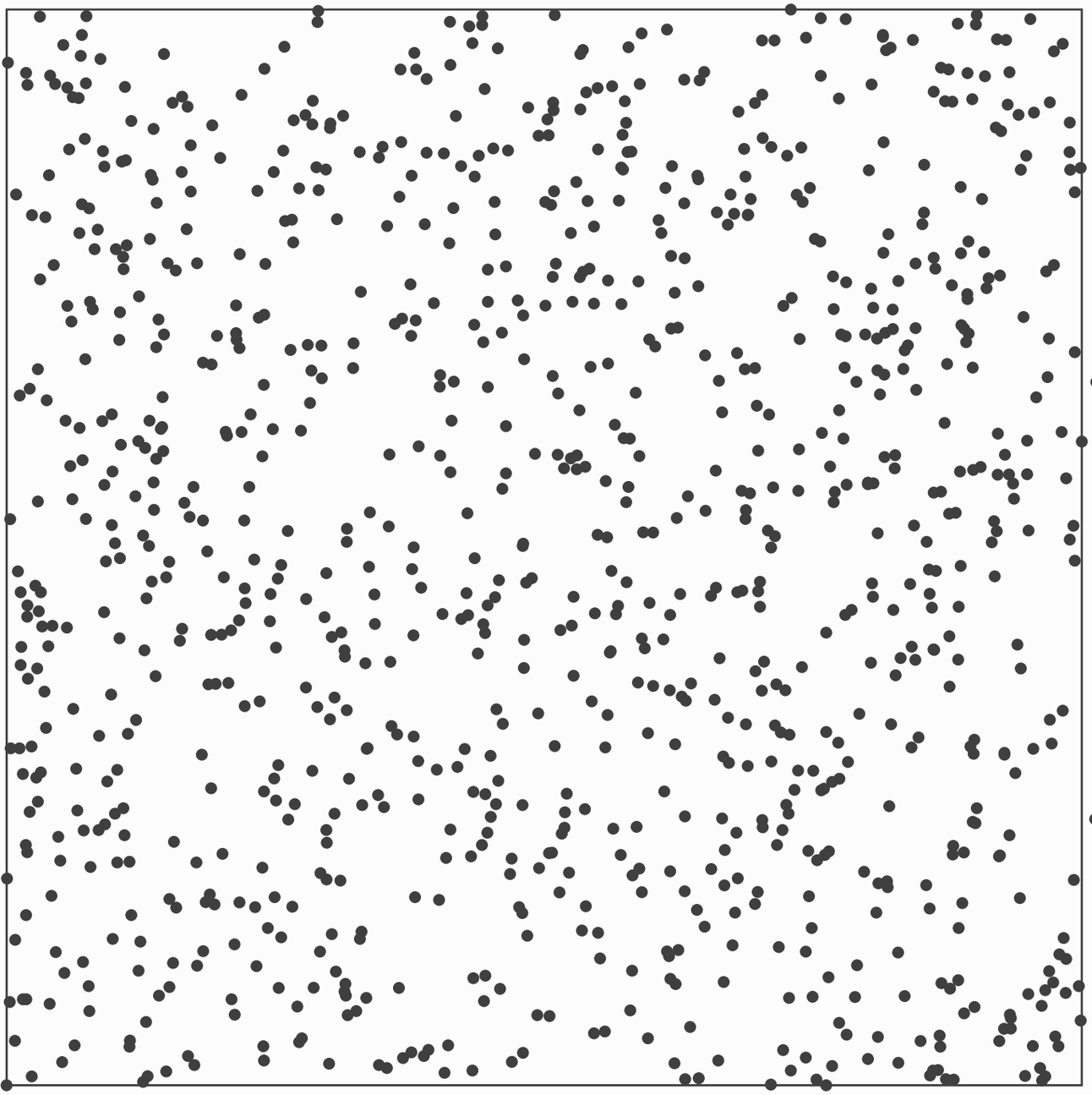
Brussels Summer School of Mathematics, August 1-7, 2013

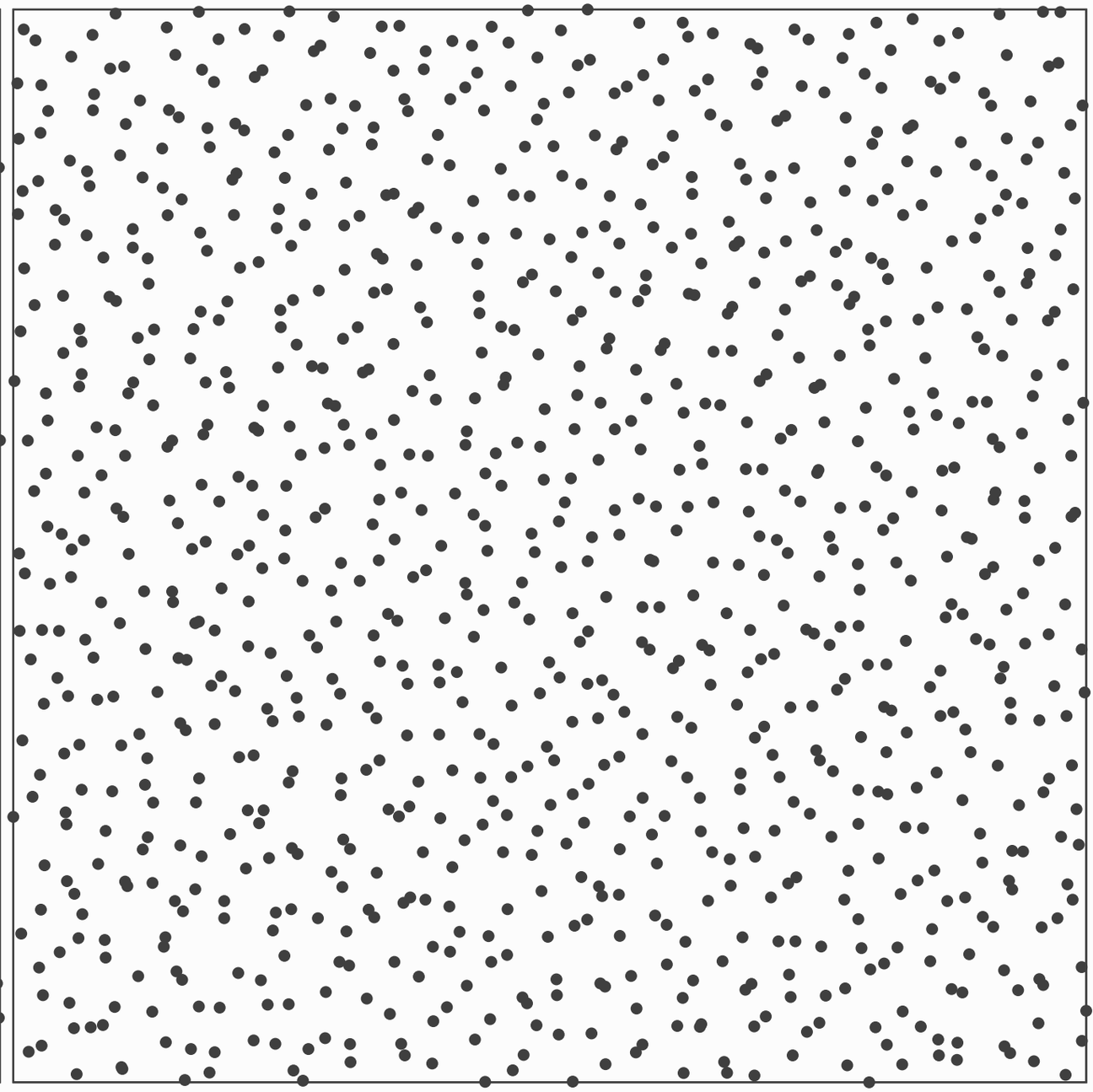
ULB UNIVERSITÉ LIBRE DE BRUXELLES

quasi- (*comb. form*)
seemingly; apparently but not really

EXAMPLE *quasi-American | quasi-scientific*
ORIGIN *from Latin* ***quasi 'as if, almost.'***

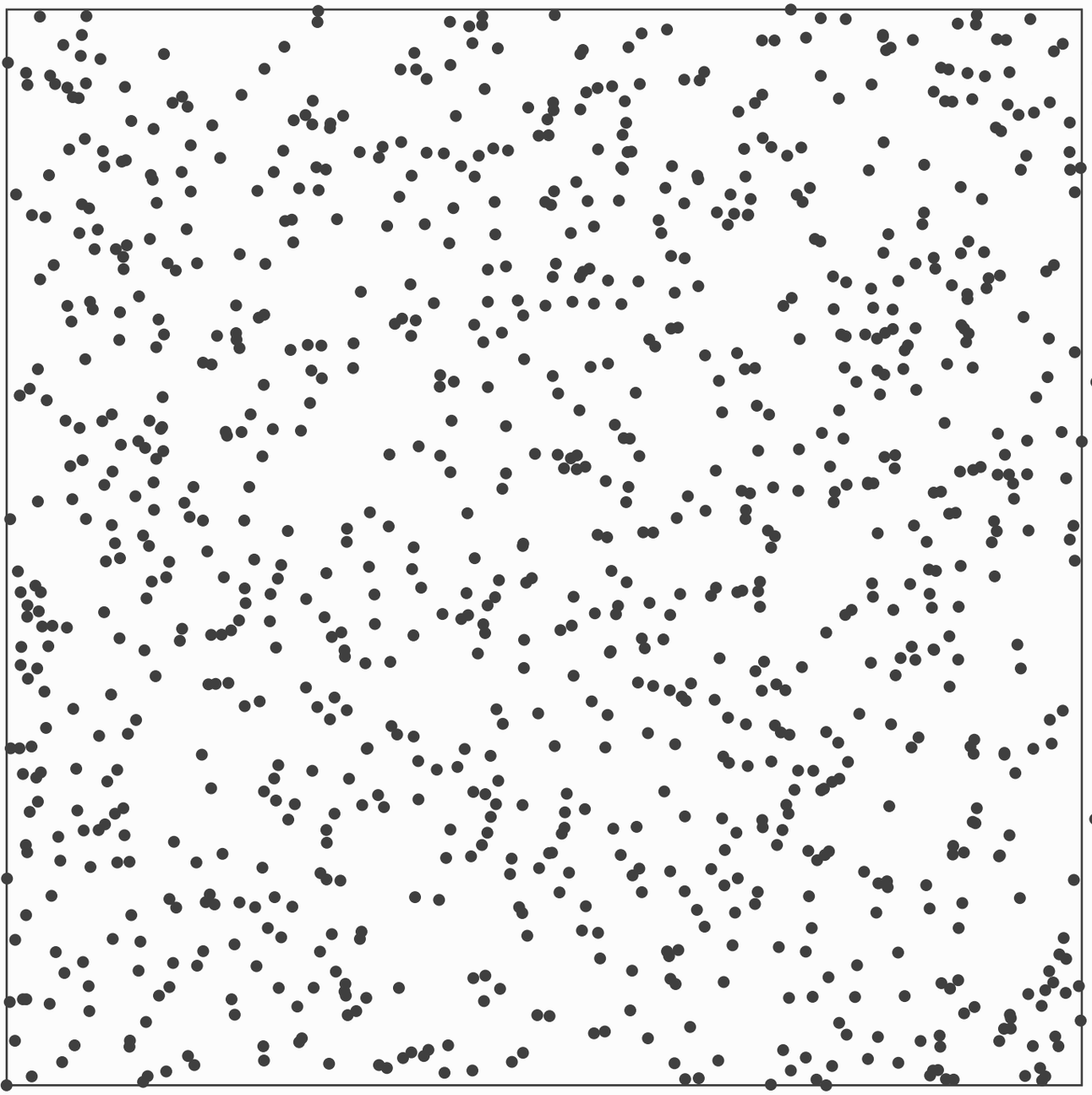
GEOMETRICAL INTUITION

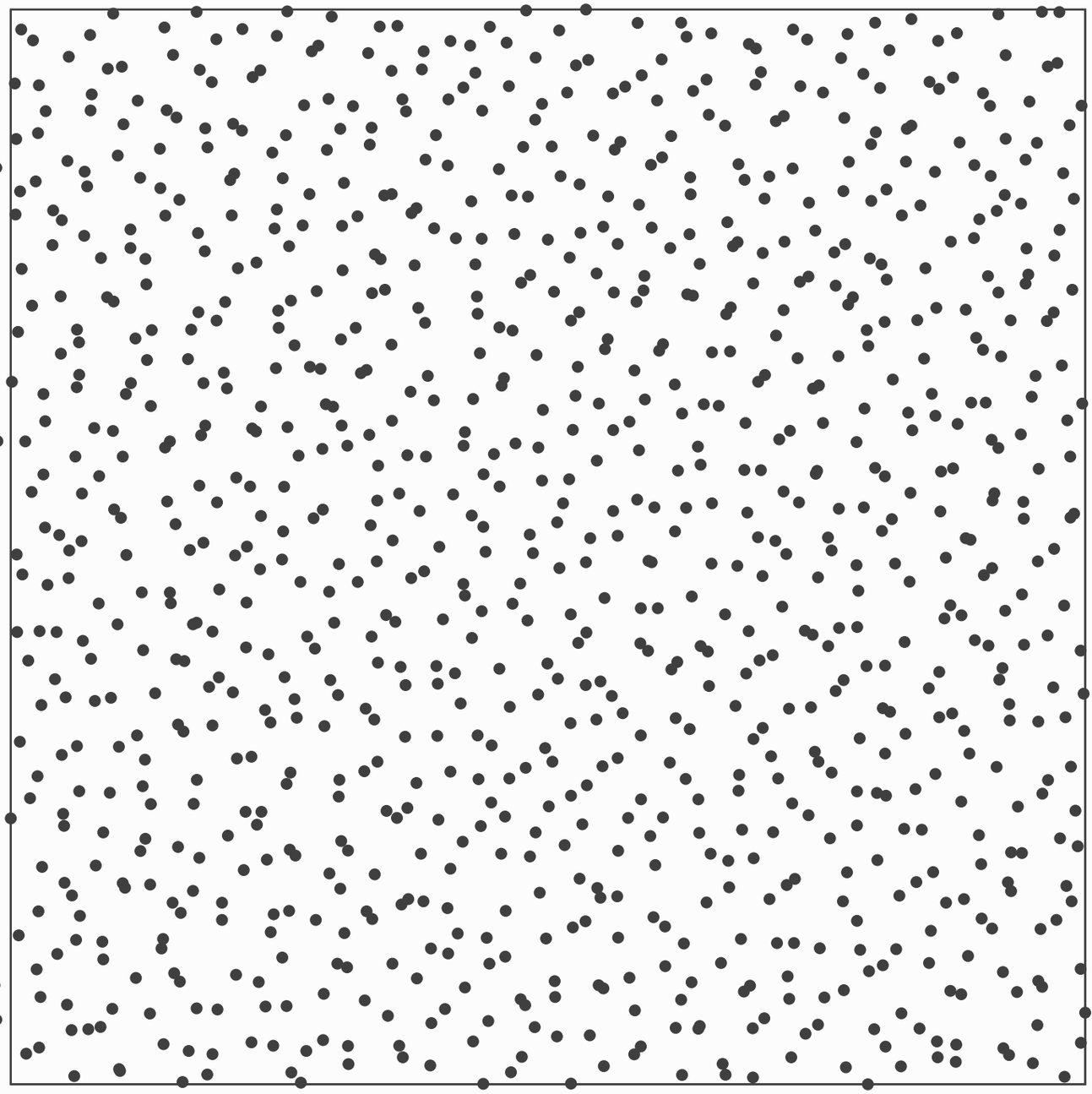




QUESTION Which of the two patterns is non-random and contains correlations among points?

GEOMETRICAL INTUITION



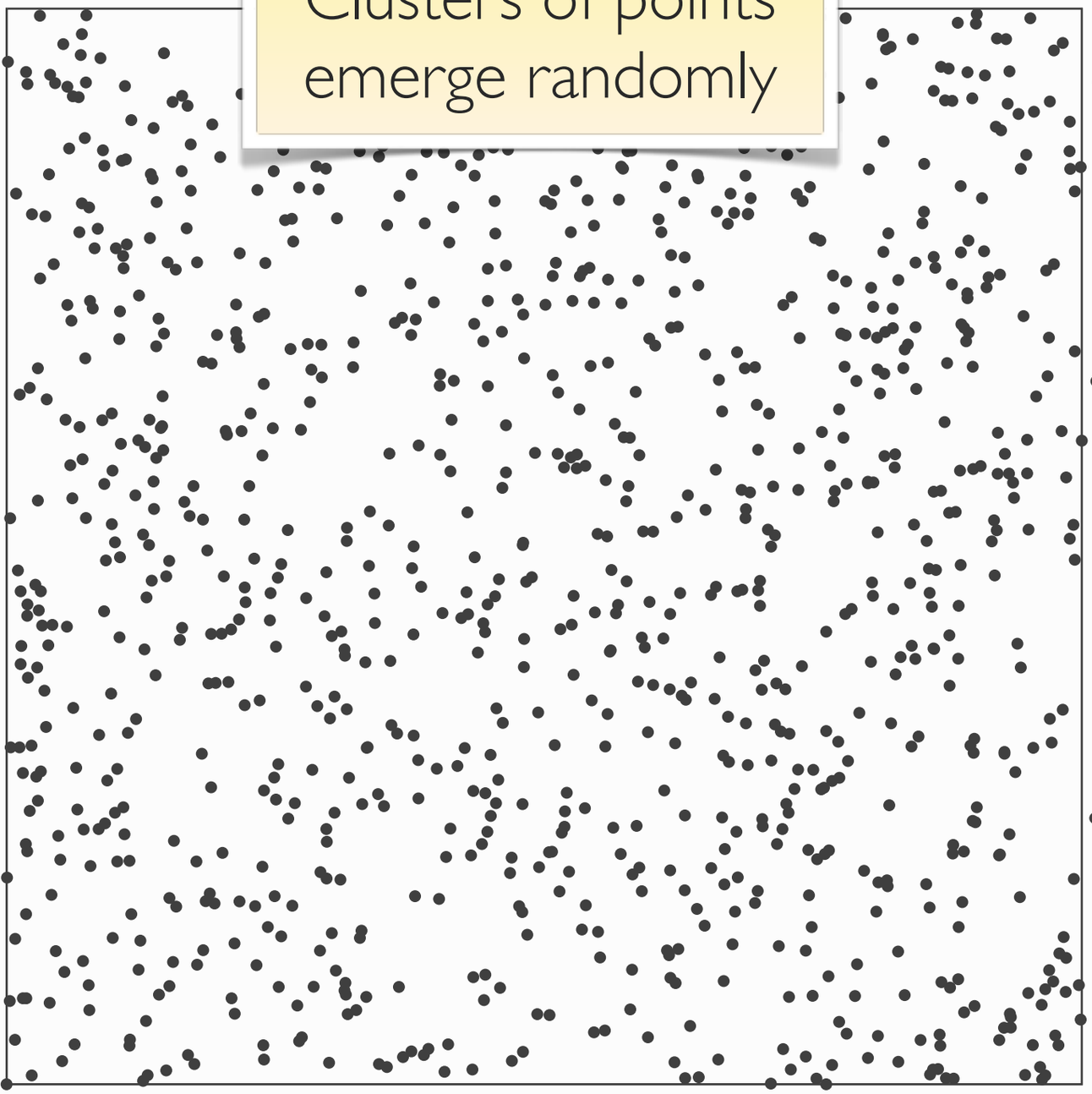


QUESTION If you scatter black grains on a white plate, how will look the distribution of grains?

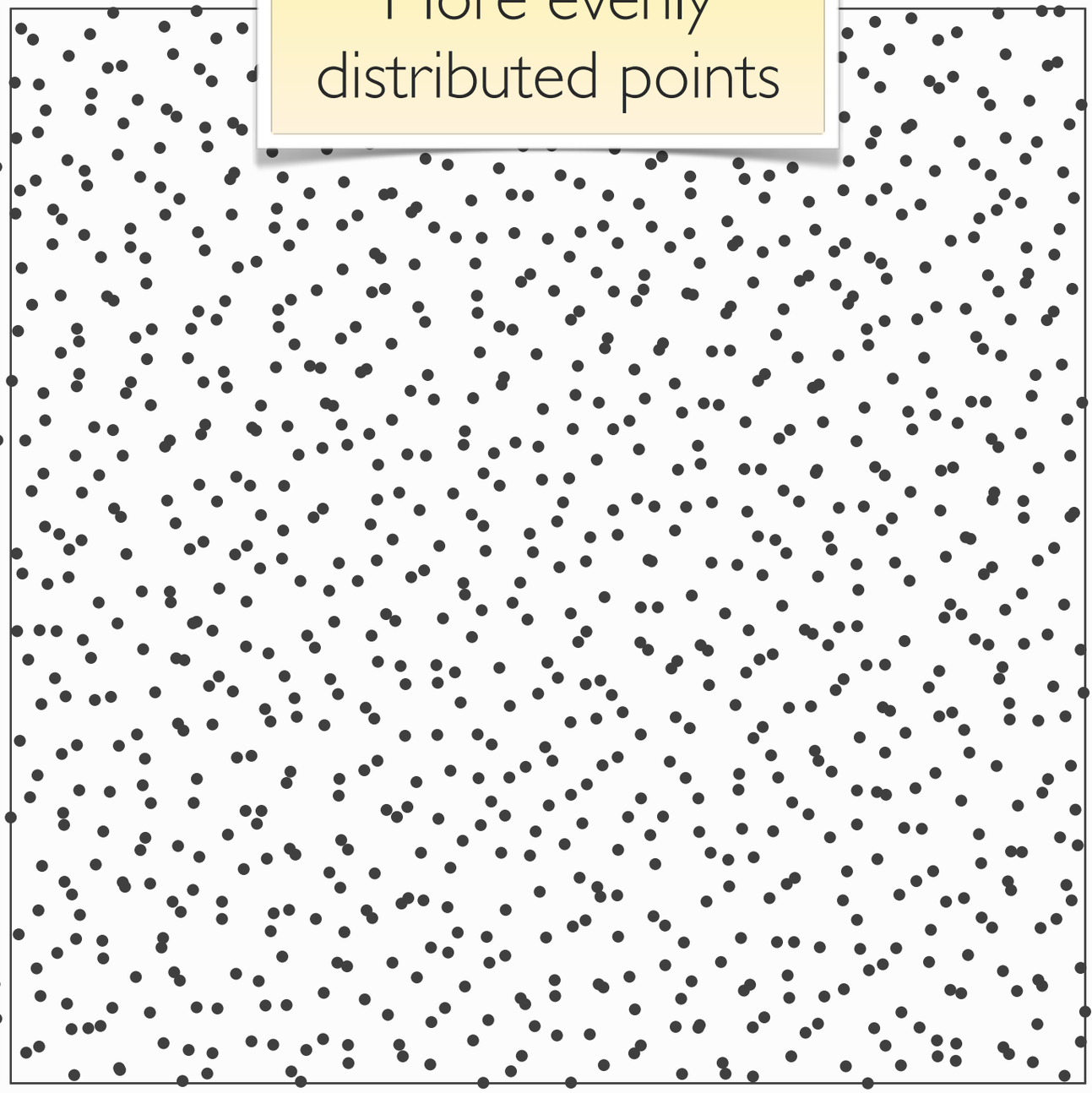
GEOMETRICAL INTUITION

Clusters of points emerge randomly

More evenly distributed points



Random (no correlation)



Quasi-random

VOCABULARY

MONTE CARLO • PSEUDO-RANDOM
LOW-DISCREPANCY • HAMMERSLEY
HALTON • VAN DER CORPUT • GRID
SEQUENCE • SET • GOLDEN RATIO
LATTICE • BLUE NOISE • STRATIFIED
SAMPLES • NONUNIFORM SAMPLING
INVERSION METHODS • THE HILBERT
SPACE FILLING CURVE...

ACKNOWLEDGMENTS

Collaborators

Leif Kobbelt, RWTH University, Germany

Paul-Olivier Dehaye, ETH Zürich, Switzerland

Harald Niederreiter, RICAM Institute, Austria

Sponsors

FP7 Marie Curie ERG (PERG06-GA-2009-256519)

Madrid-MIT M+Vision Consortium Fellowship

COMPUTER **RENDERING**

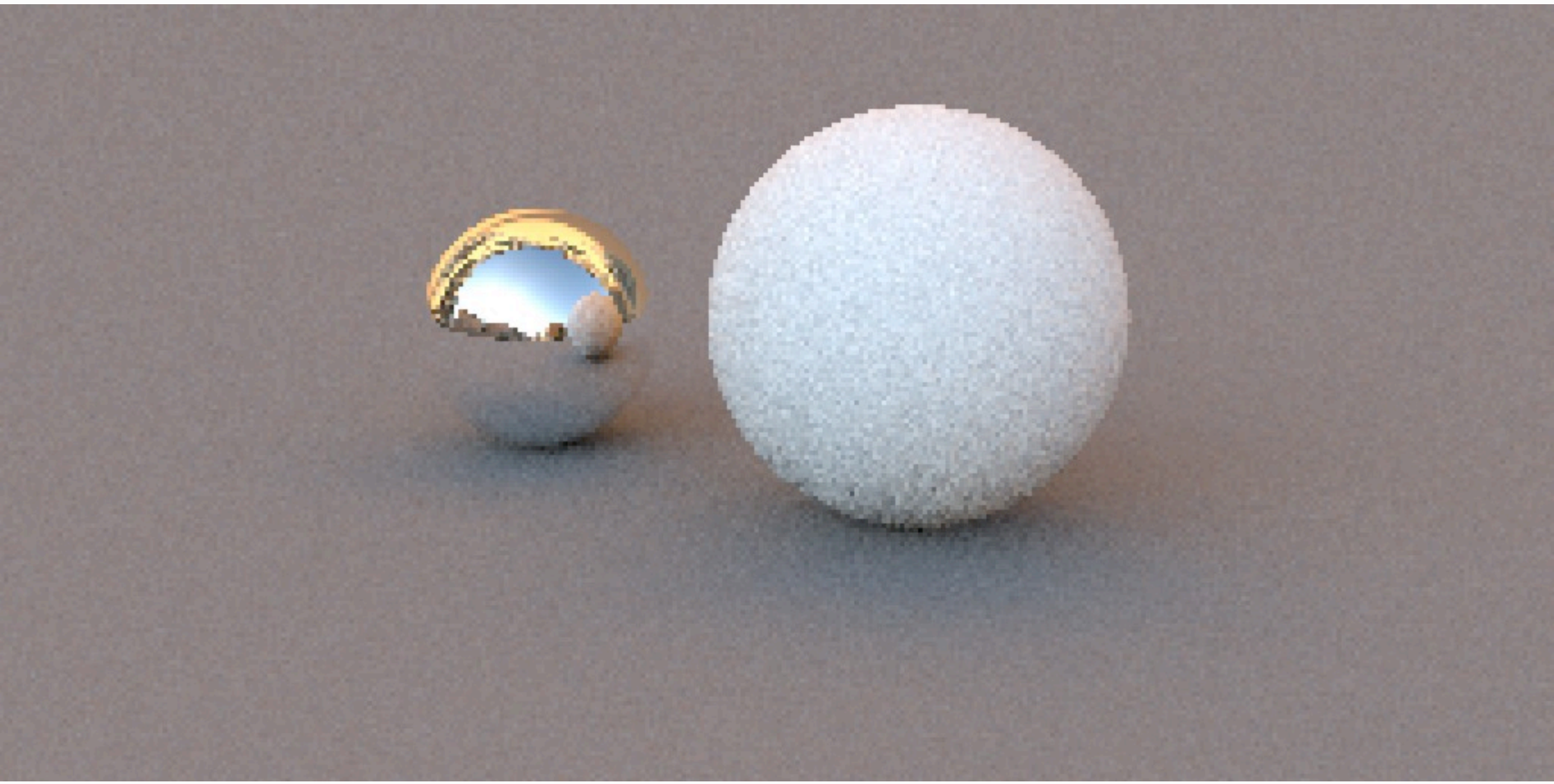
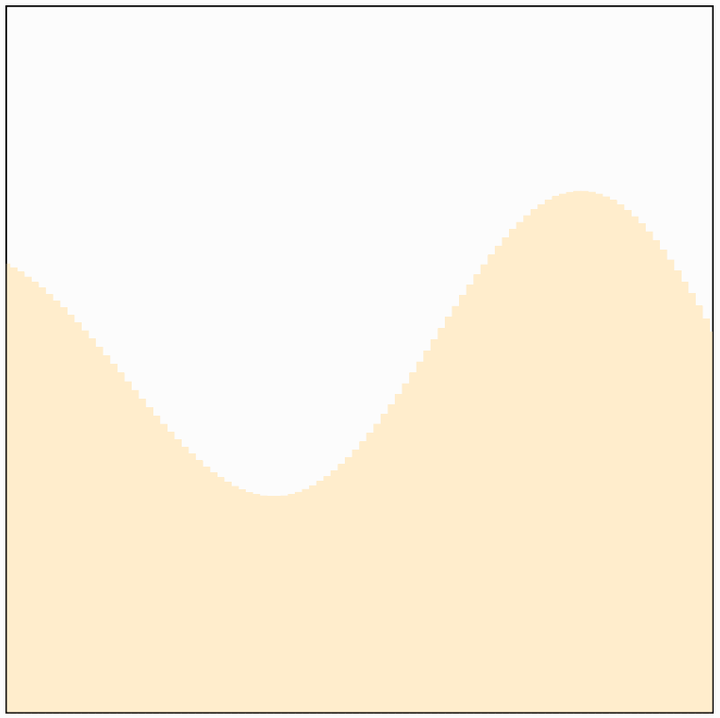


Image-based lighting, 64 quasi-random samples per pixel

MONTE CARLO INTEGRAL

MONTE CARLO INTEGRAL

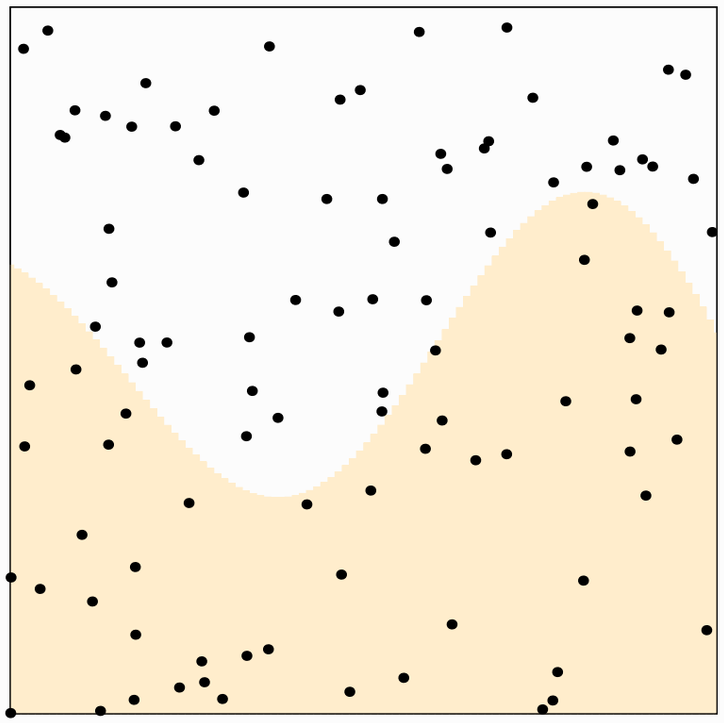


2D function
defined on **D = [0,1)x [0,1)**

$$I = \iint_D f\,(x, y)\,\mathrm{d}x\,\mathrm{d}y$$

Is analytical integration always possible?

MONTE CARLO INTEGRAL



2D function
defined on **D = [0,1)x [0,1)**

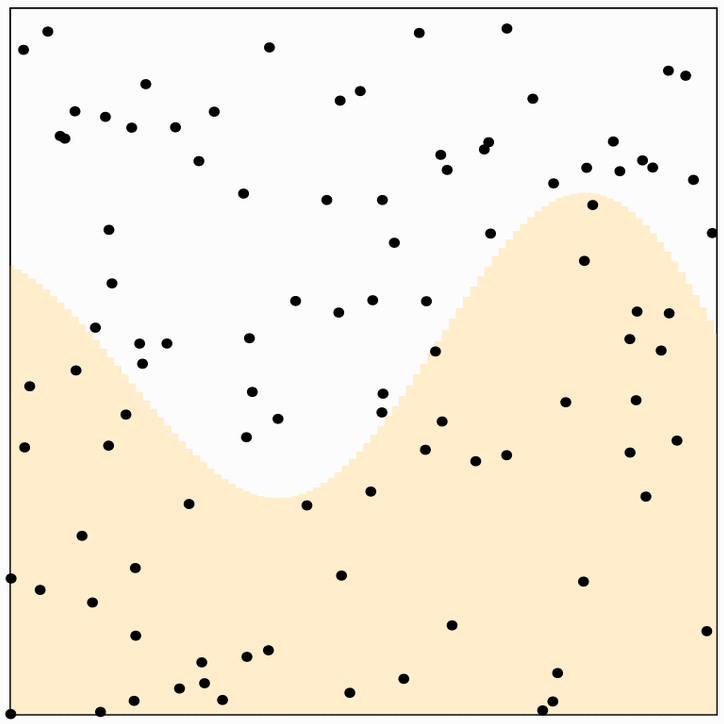
$$I = \iint_D f(x,y)\,\mathrm{d}x\,\mathrm{d}y$$

$$\approx \frac{1}{N}\sum_{i=1}^{N} f(x_i,y_i) = \hat{I}_N$$

$$\text{with } (x_i,y_i) \in D \quad \forall i \in [1..N]$$

Approximation with a discrete sum

MONTE CARLO INTEGRAL



2D function
defined on **D = [0,1)x [0,1)**

$$I = \iint_D f(x,y)\,\mathrm{d}x\,\mathrm{d}y$$

$$\approx \frac{1}{N}\sum_{i=1}^{N} f(x_i,y_i) = \hat{I}_N$$

$$\text{with } (x_i,y_i) \in D \quad \forall i \in [1..N]$$

Approximation with a discrete sum

QUESTION What is the expected mean integration error when the N *point samples* are chosen at random?

RATE OF CONVERGENCE

$$\sigma^2 = Var\left(f\right) = \iint\limits_D \left[f\left(x,y\right) - I\right]^2 \mathrm{d}x\, \mathrm{d}y$$

RATE OF CONVERGENCE

$$\sigma^2 = Var\left(f\right) = \iint\limits_D \left[f\left(x,y\right) - I\right]^2 \mathrm{d}x\ \mathrm{d}y$$

$$\rightarrow E\left[\left(I - \hat{I}_N\right)^2\right] = \frac{\sigma^2}{N}$$

Assume a constant mean squared error (MSE) that is averaged over all N samples.

RATE OF CONVERGENCE

$$\sigma^2 = Var\left(f\right) = \iint_D \left[f\left(x,y\right) - I\right]^2 \mathrm{d}x\, \mathrm{d}y$$

$$\rightarrow E\left[\left(I - \hat{I}_N\right)^2\right] = \frac{\sigma^2}{N}$$

Assume a constant mean squared error (MSE) that is averaged over all N samples.

Result: The squared error between the true integral and the Monte Carlo approximation decreases linearly with the number of samples.

QUESTION Is this a "*fast*" rate of convergence?

BERNOULLI TRIALS

We want to approximate the mean value by flipping coins and counting 0 for **heads** (H) and 100 for **tails**

QUESTION How many times to you need to flip the coin to approach the mean value 50 with 1% error?

BERNOULLI TRIALS

One flip

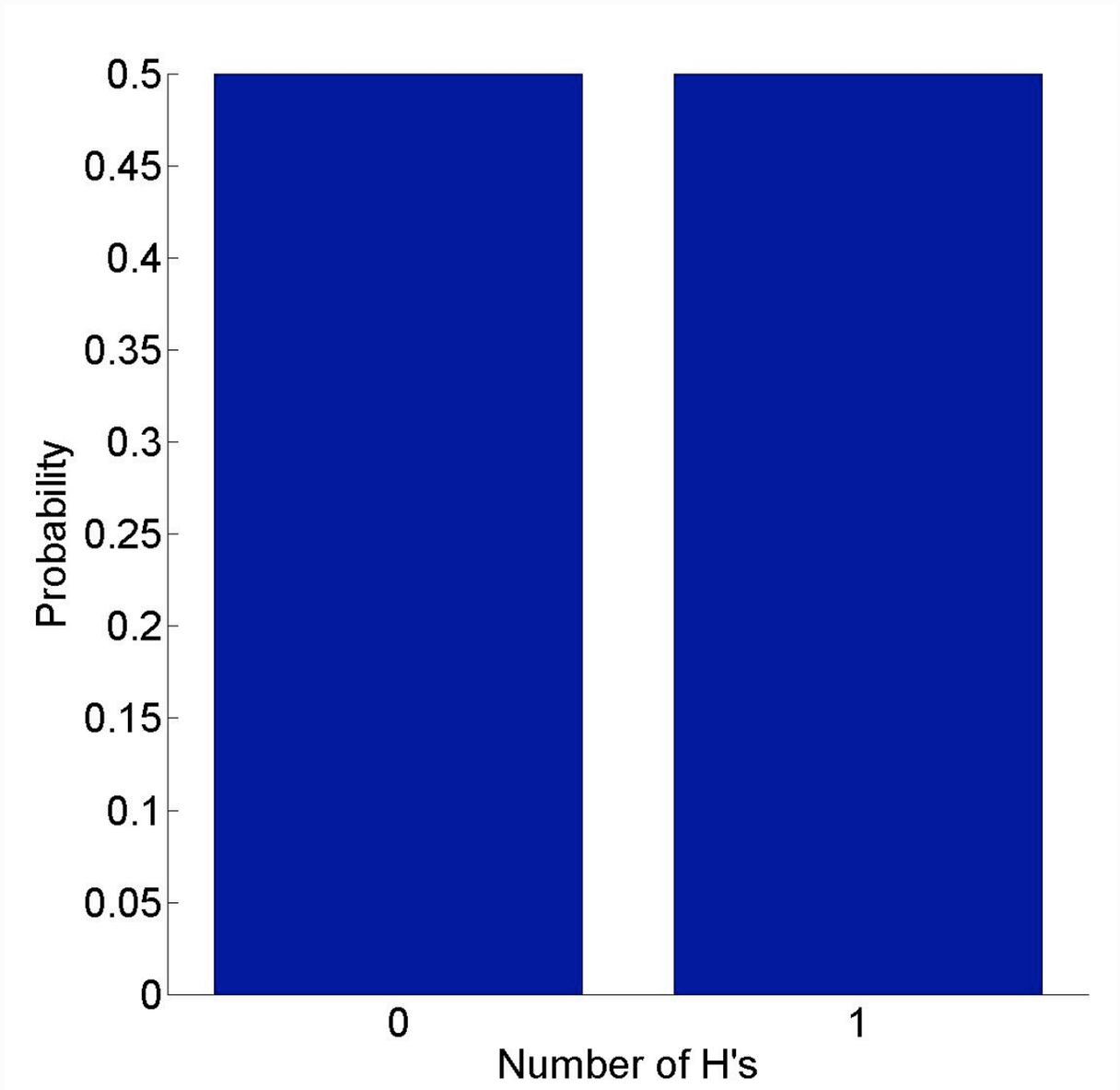
Mean:

0 or 100

Average error:

(50 + 50) / 2

= **50**



BERNOULLI TRIALS

Two flips

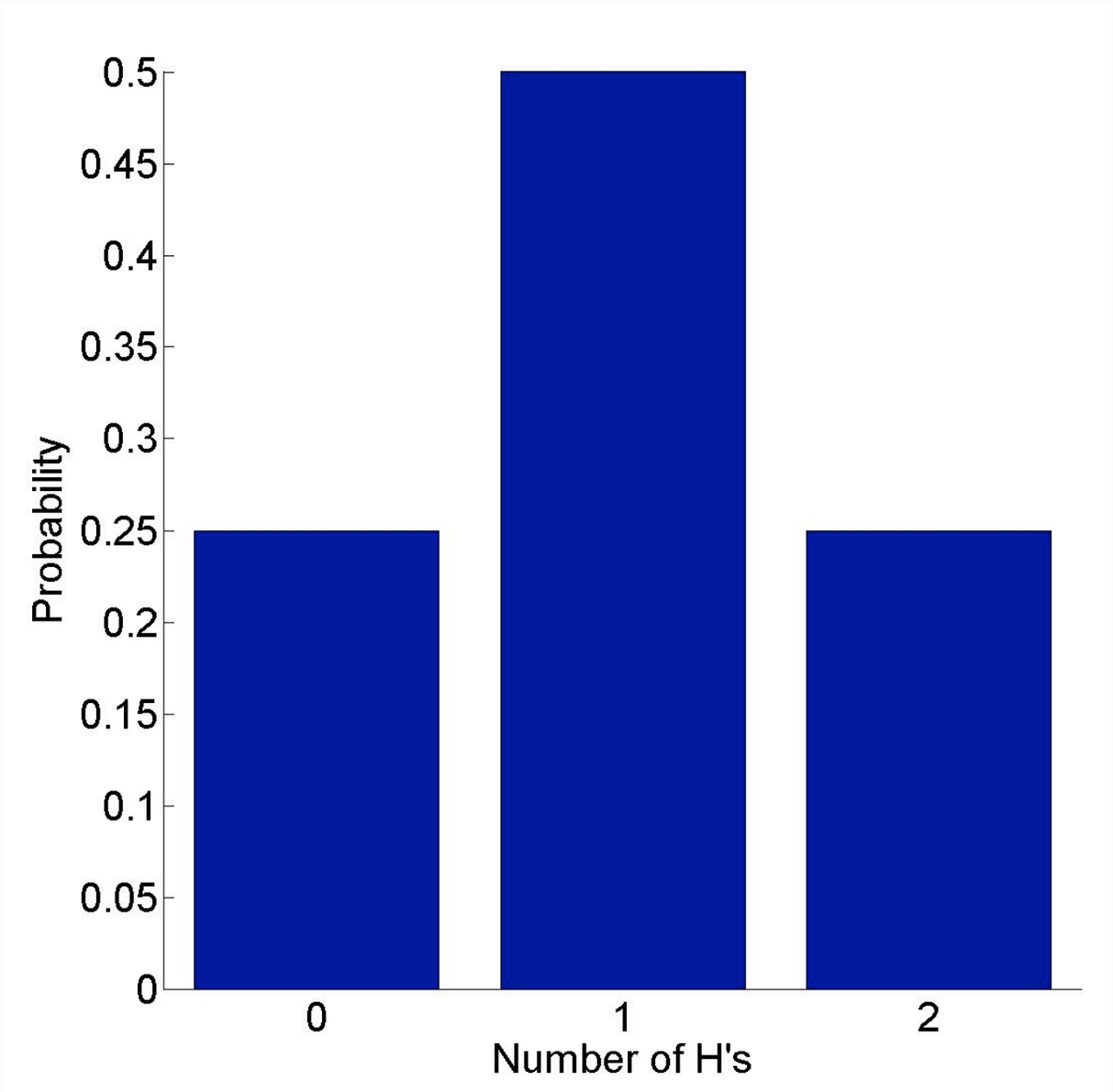
Mean:

0, 50(x2) or 100

Average error:

$(50 + 0(x2) + 50) / 2^2$

$= \mathbf{25}$



BERNOULLI TRIALS

Three flips

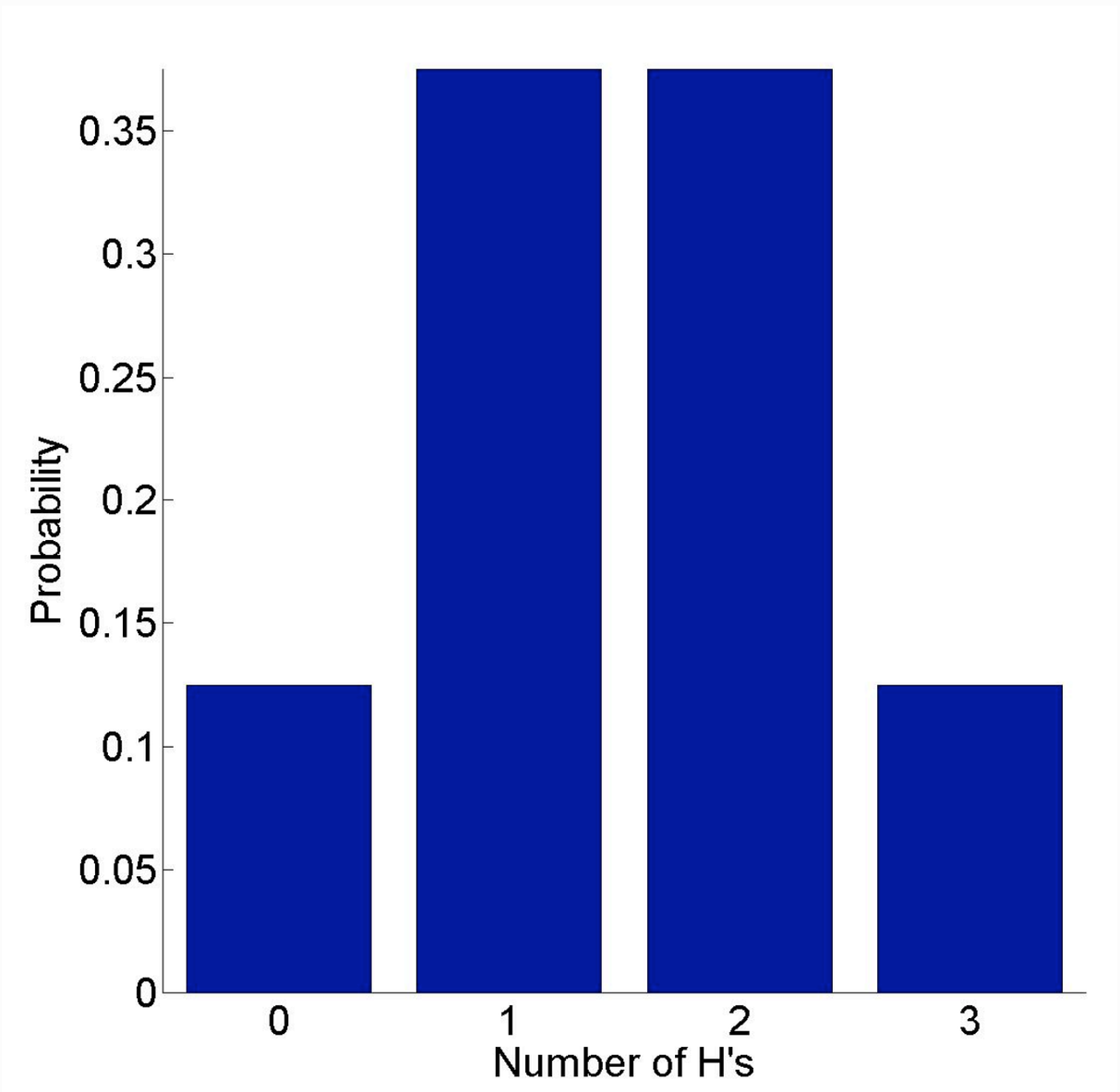
Mean:

0, 33.3(x3), 66.6(x3) or 100

Average error:

$(50 + 16.6(\times 6) + 50) / 2^3$

$= \mathbf{25}$



BERNOULLI TRIALS

Three flips

Mean:

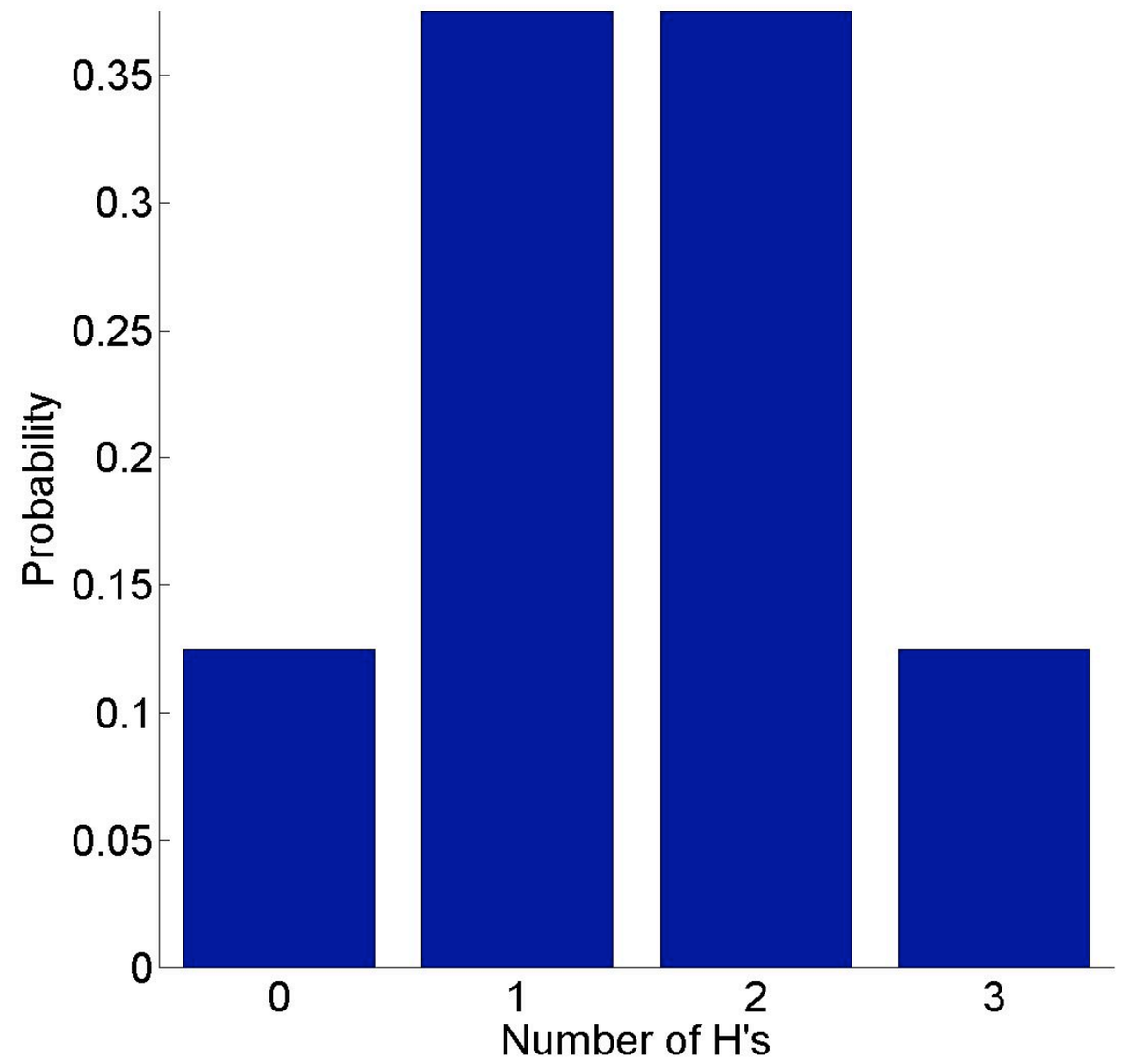
0, 33.3(x3), 66.6(x3) or 100

Average error:

$(50 + 16.6(x6) + 50) / 2^3$

$= \mathbf{25}$

Flipping 3 instead of 2 coins, does not improve the average error!



BERNOULLI TRIALS

Four flips

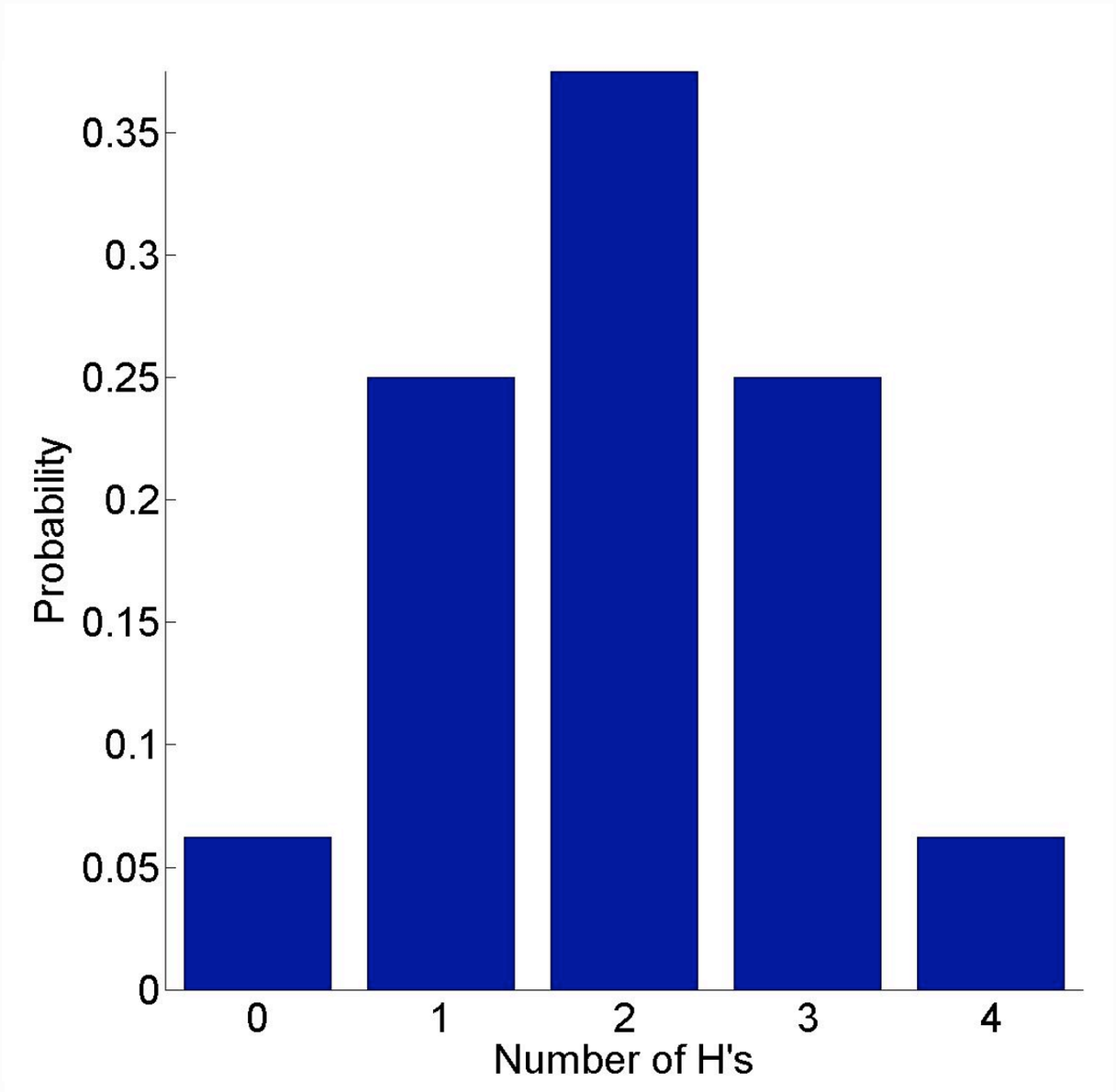
Mean:

0, 25(x4), 50(x6), 75(x4) or 100

Average error:

$(50(x2) + 25(x8)) / 2^4$

$= \mathbf{18.75}$



BERNOULLI TRIALS

Five flips

Mean:

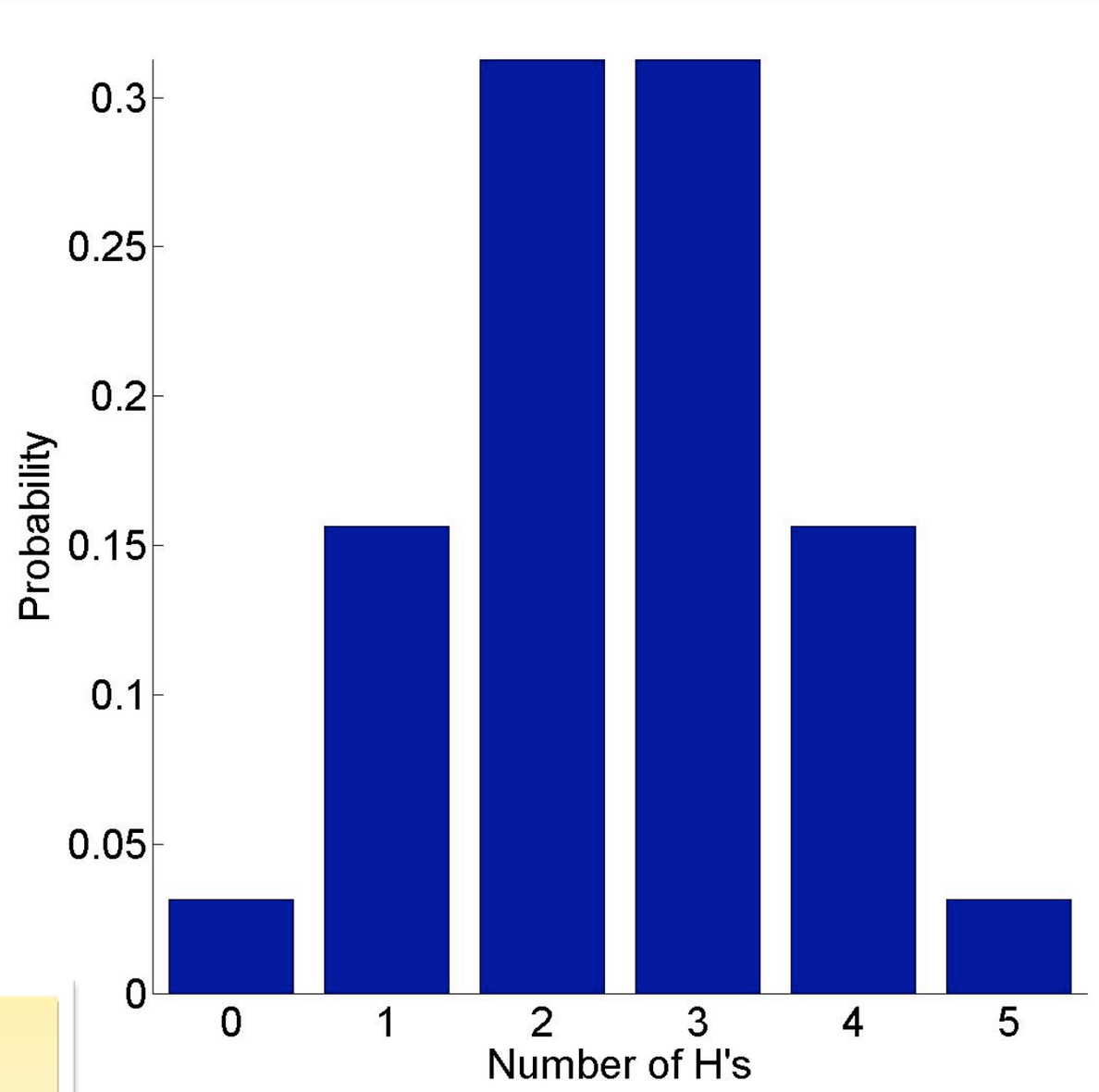
0, 20(x5), 40(x10), 60(x10), 80(x5) or 100

Average error:

(50(x2) + 30(x10) + 10(x20)) / 2^5

= **18.75**

Flipping 5 instead of 4 coins, does not improve the average error!



BERNOULLI TRIALS

Six flips

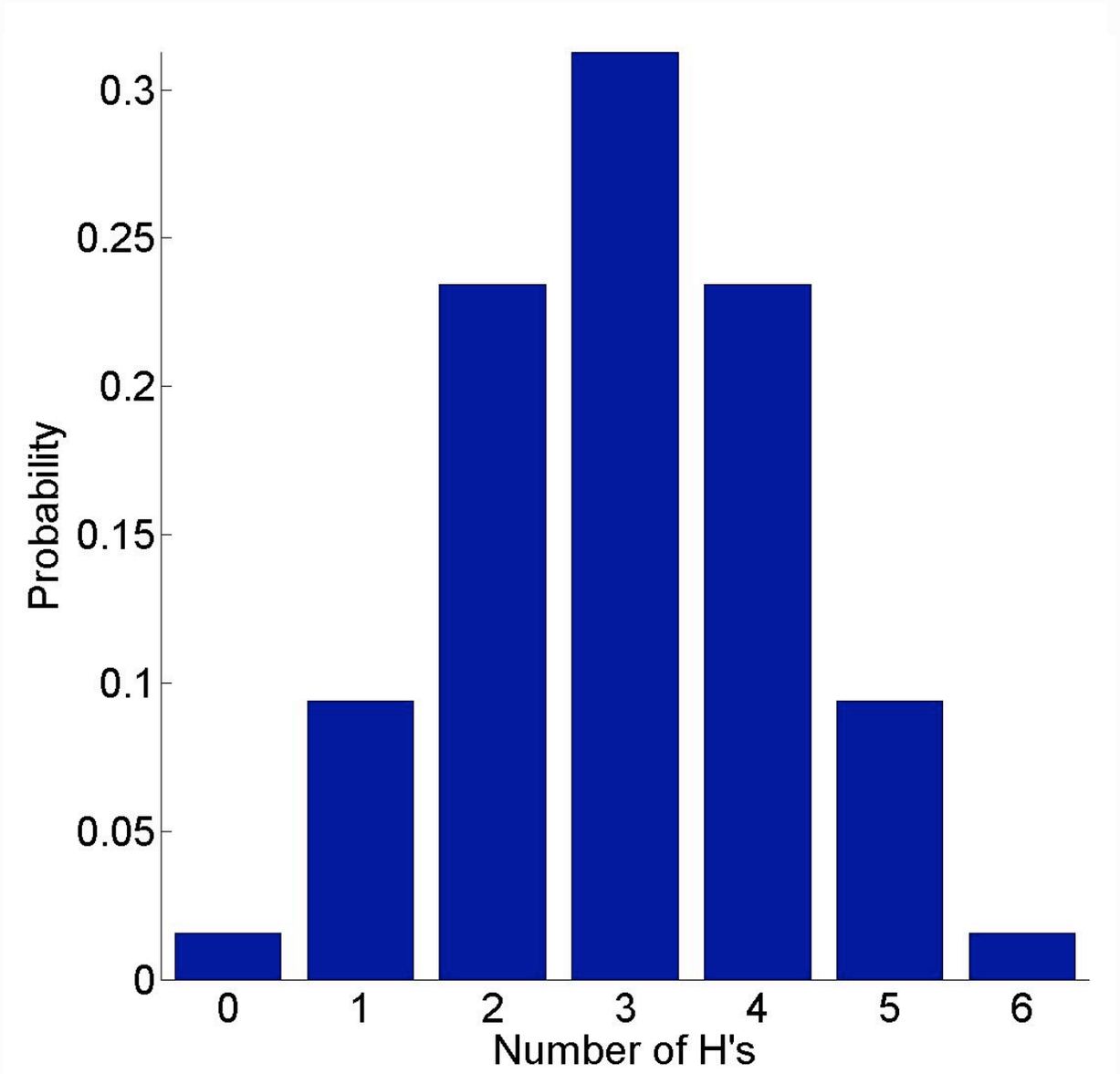
Mean:

0, 16.6(x6), 33.3(x15), 50(x20), 66.6(x15), 83.3(x6) or 100

Average error:

(50(x2) + 33.3(x12) + 16.6(x30)) / 2^6

= **15.625**



BERNOULLI TRIALS

Seven flips

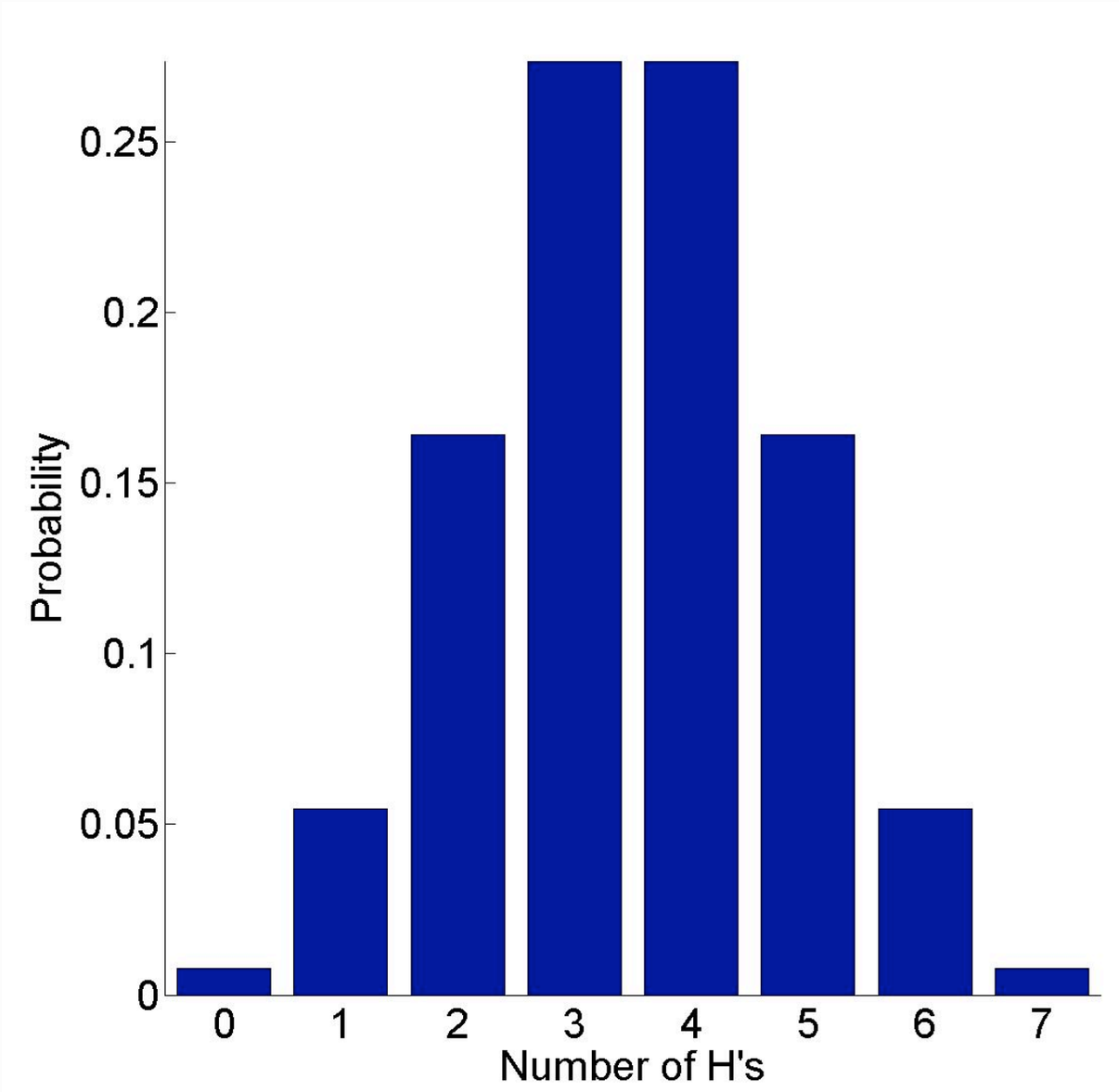
Mean:

0, 14.3(x7), 28.6(x21), 42.6(x35), 57.1(x35), 71.4(x21), 85.7(x7) or 100

Average error:

(50(x2) + 85.7(x14) + 71.4(x42) + 14.3(x70)) / 2^7

= **15.625**



BERNOULLI TRIALS

Eight flips

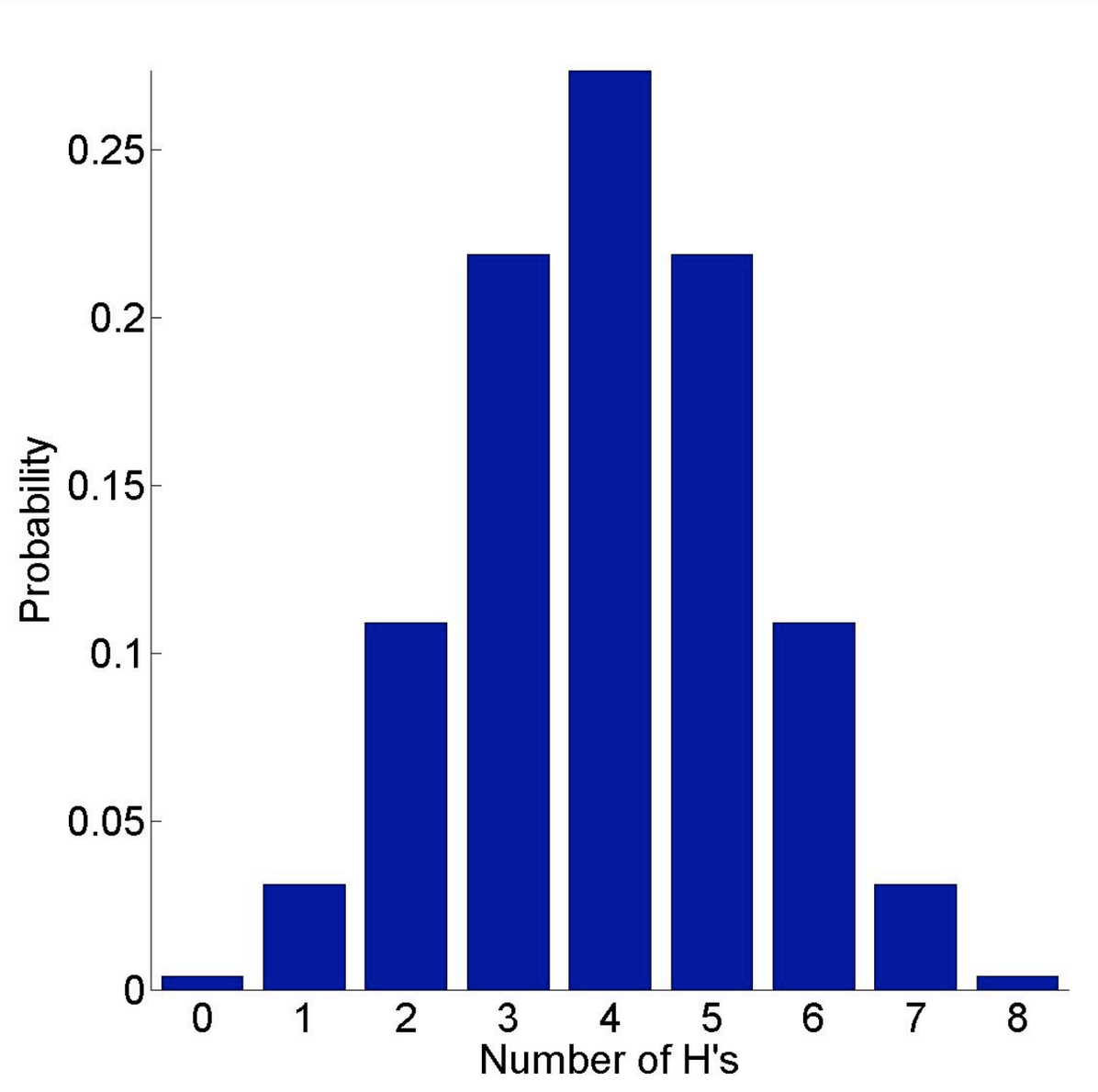
Mean:

0, 12.5(x8), 25(x28),
37.5(x56), 50(x70), 62.5(x56),
75(x28), 87.5(x8) or 100

Average error:

(50(x2) + 37.5(x16) +
25(x56) + 12.5(x112) +
0(x70)) / 2^8

= **13.671875**



BERNOULLI TRIALS

Hundred flips

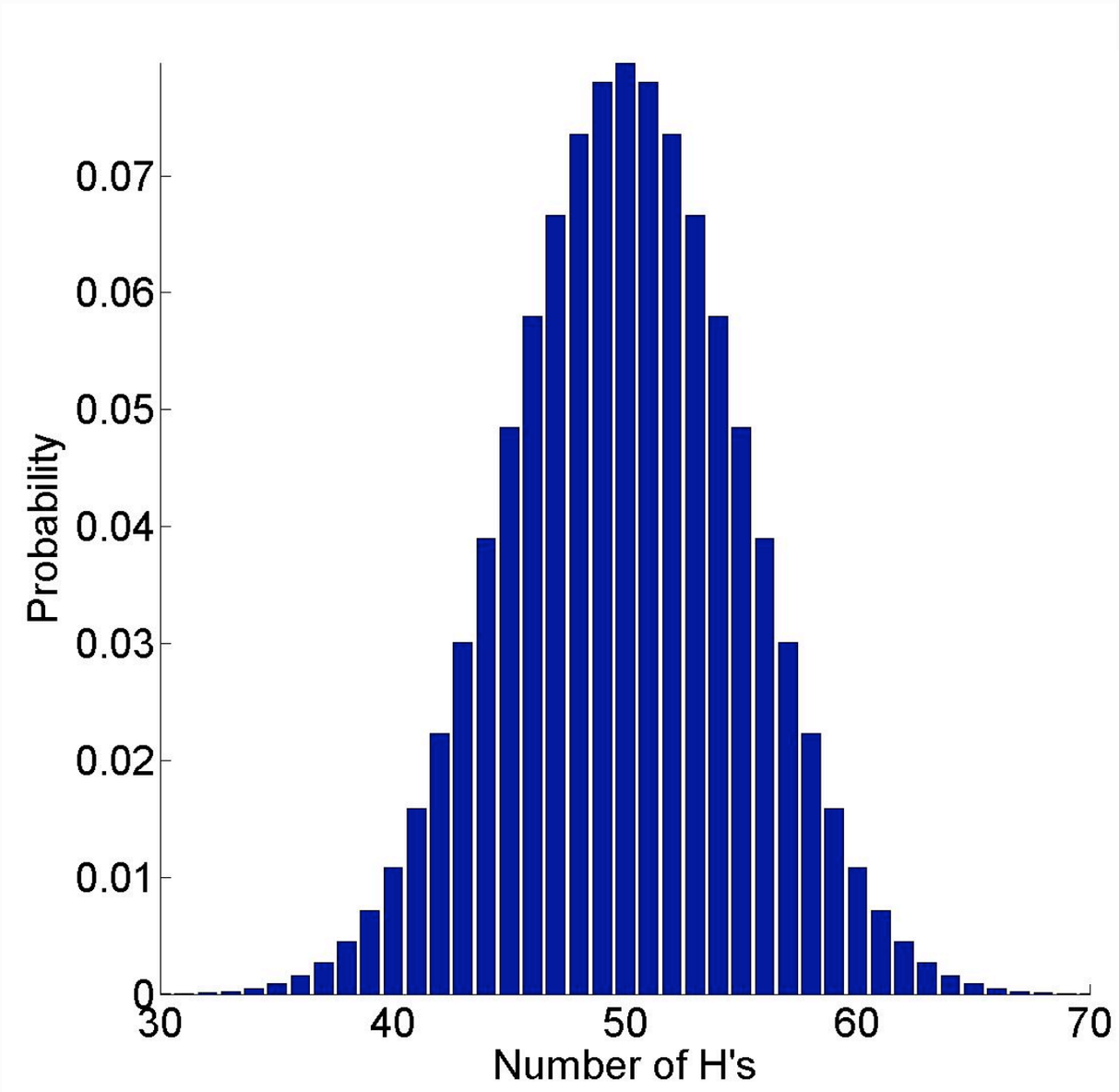
Mean:

...

Average error:

...

= **3.979462**



AVERAGE ERROR

QUESTION How many times to you need to flip the coin to approach the mean value 50 with 1% error?

1 flip: avg. error = **50**

2 flips: avg. error = **25** = 50 - 50 / 2

4 flips: avg. error = **18.75** = 25 - 25 / 4

6 flips: avg. error = **15.625** = 18.75 - 18.75 / 6

8 flips: avg. error = **13.671875** = 15.625 - 15.625 / 8

Observation: The relative error improvement is inversely proportional to the number of sample drawn so far.

AVERAGE ERROR

QUESTION How many times to you need to flip the coin to approach the mean value 50 with 1% error?

1 flip: avg. error = **50**

2 flips: avg. error = **25** = 50 - 50 / 2

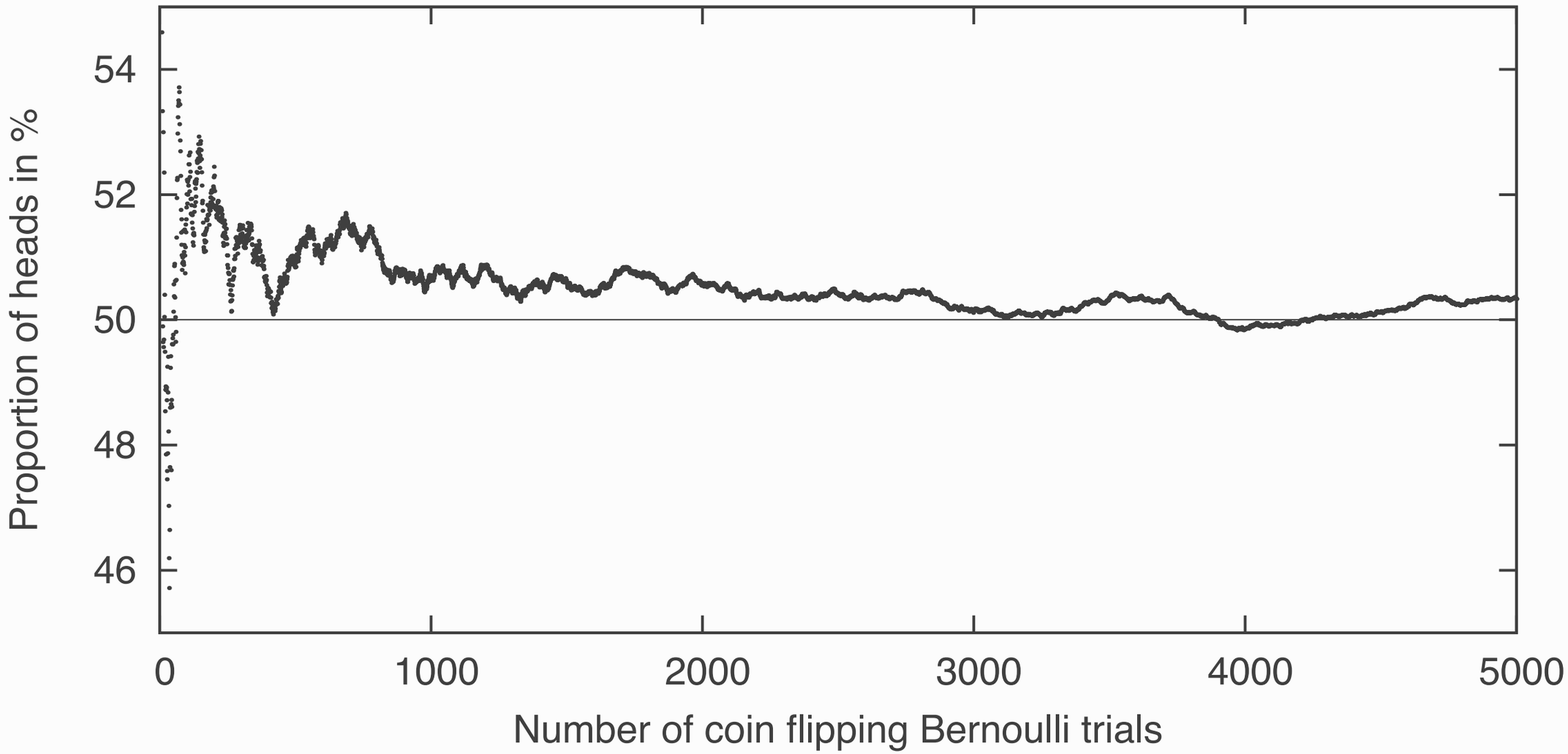
4 flips: avg. error =**18.75** = 25 - 25 / 4

6 flips: avg. error =**15.625** = 18.75 - 18.75 / 6

8 flips: avg. error =**13.671875** = 15.625 - 15.625 / 8

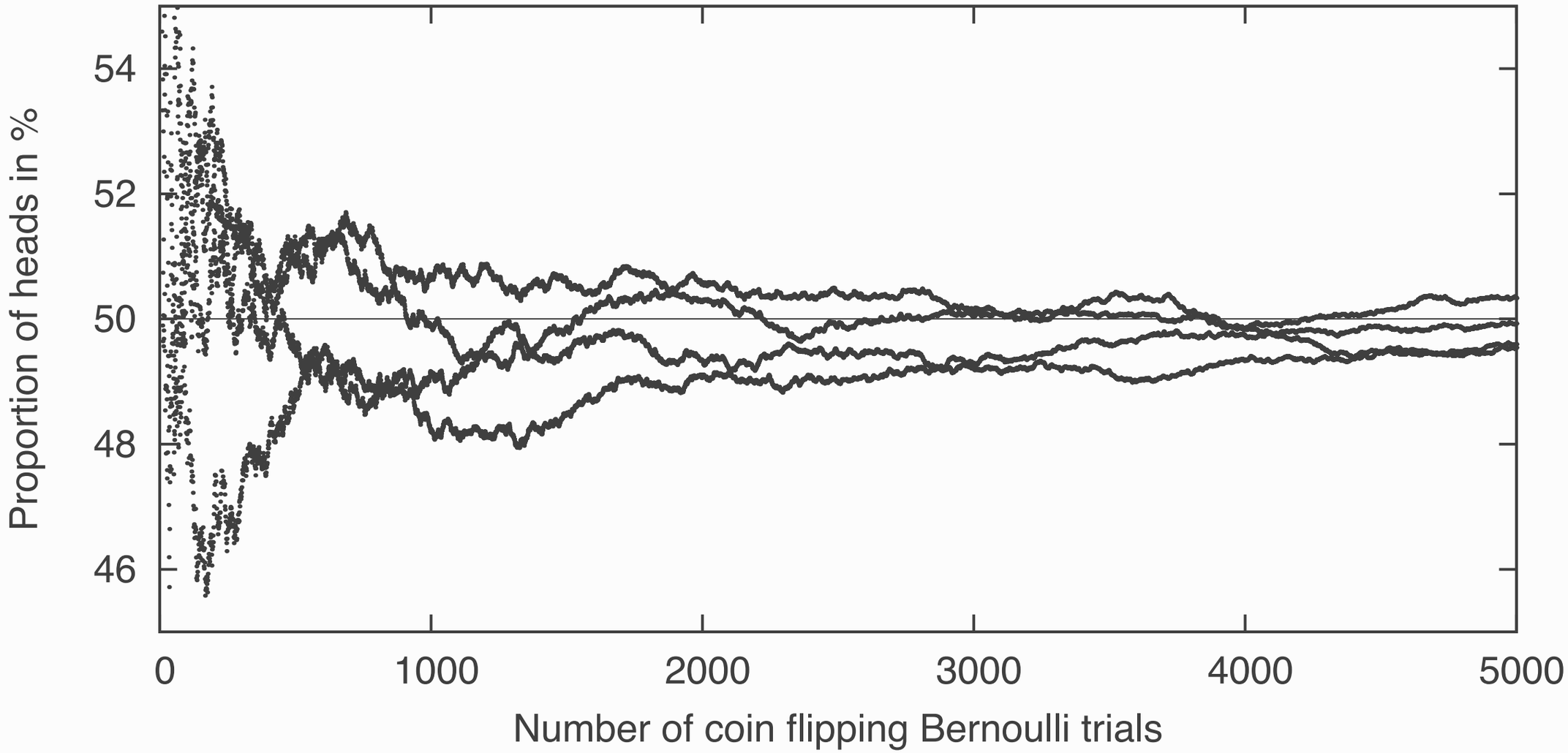
1592 flips: avg. error = **0.9997015** = 1.00033 - 1.00033 / 1592

LAW OF LARGE NUMBERS



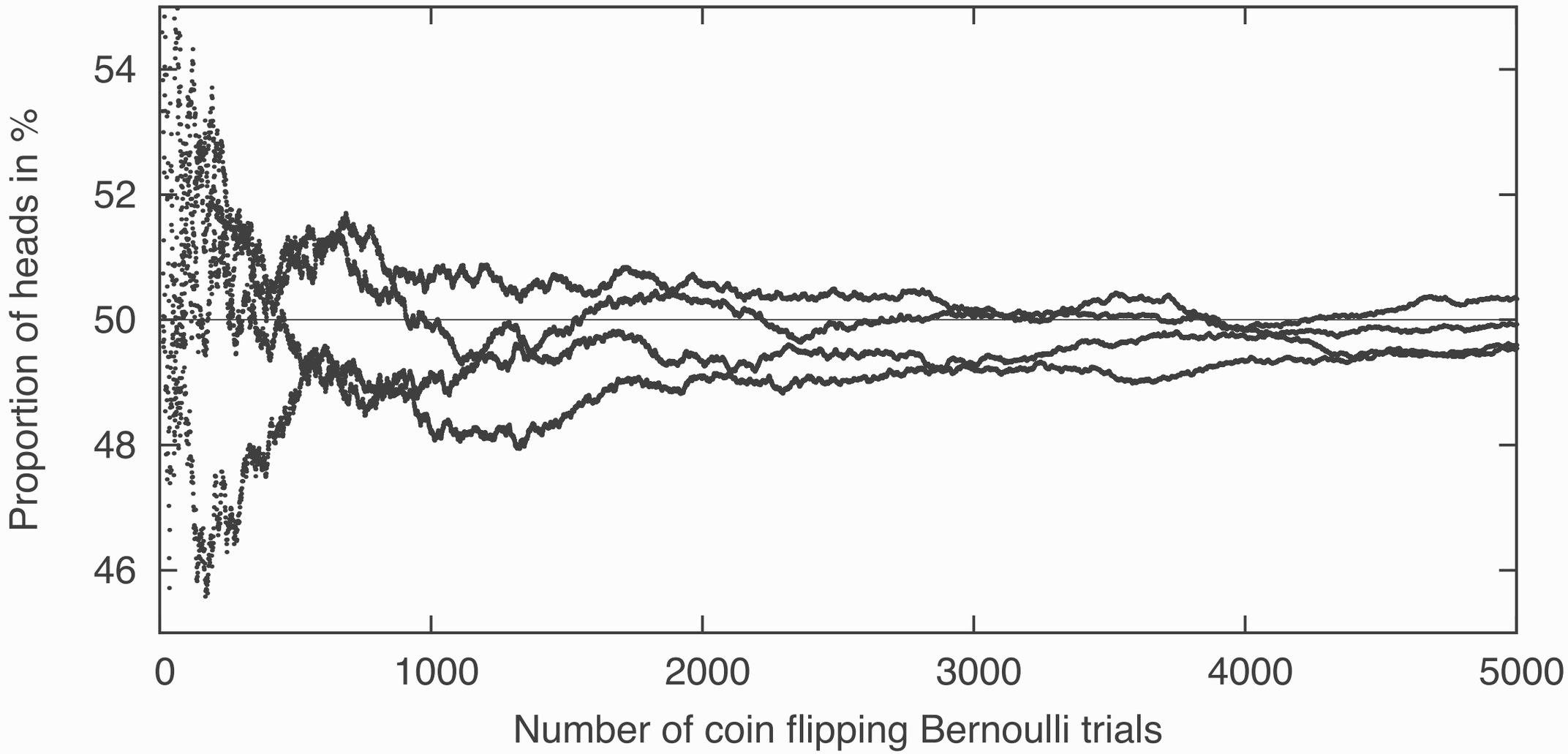
As the number of independent *randomly* generated trials increases, their average tends to their theoretical mean.

LAW OF LARGE NUMBERS



As the number of independent *randomly* generated trials increases, their average tends to their theoretical mean.

LAW OF LARGE NUMBERS

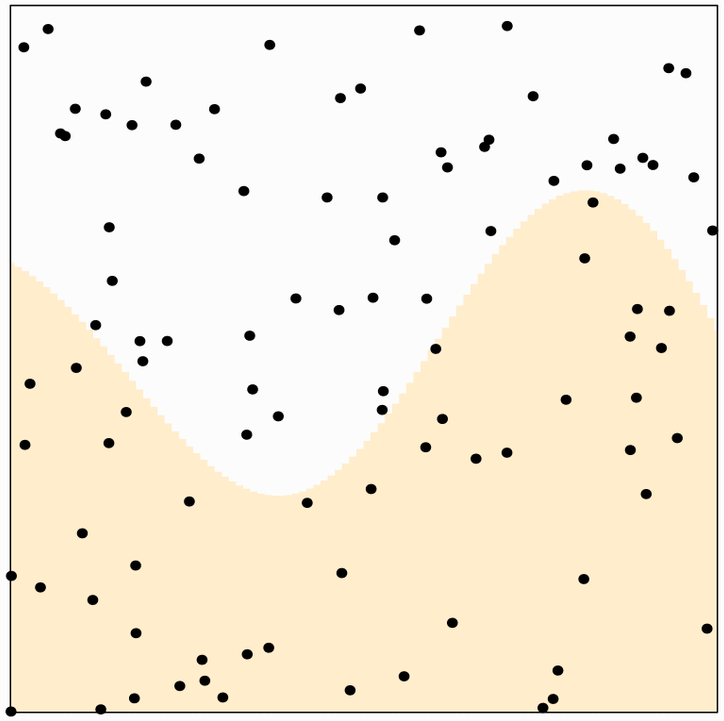


As the number of independent *randomly* generated trials increases, their average tends to their theoretical mean.

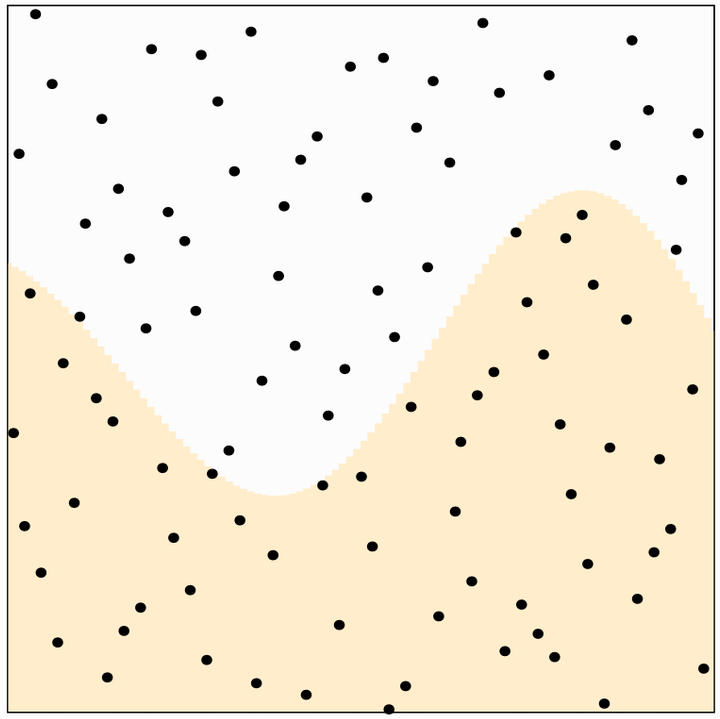
Consequence: If it is possible to improve the convergence rate, then there is only one way: *Do not pick point samples at random!*

QUASI-RANDOM POINTS

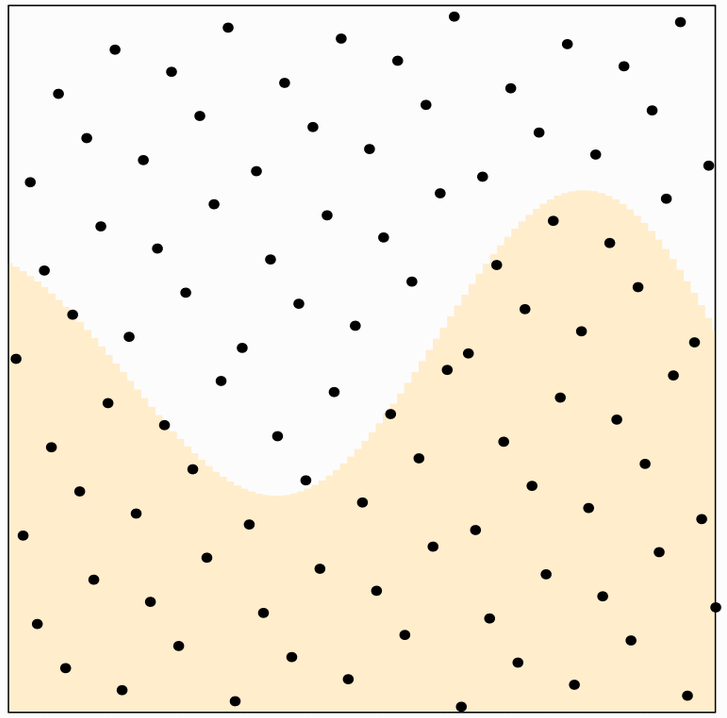
QUASI-RANDOM POINTS



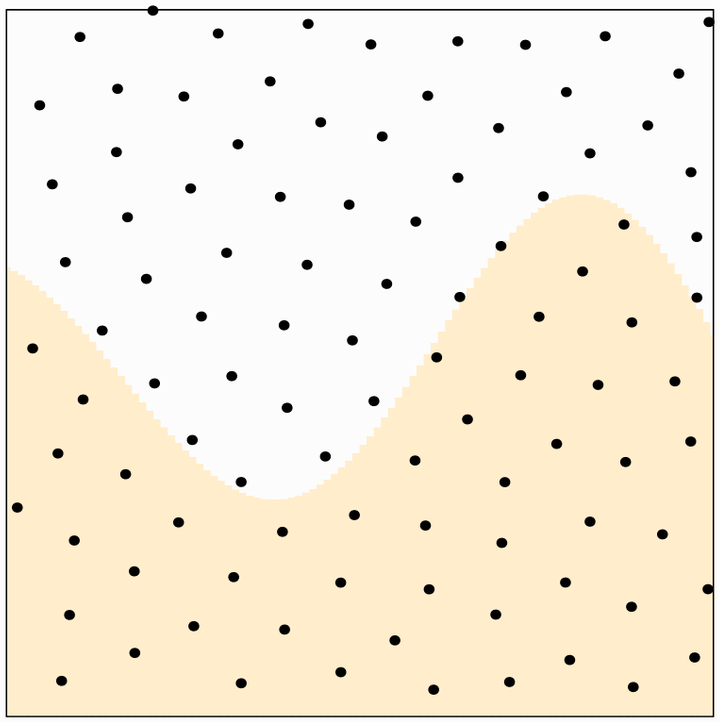
Random



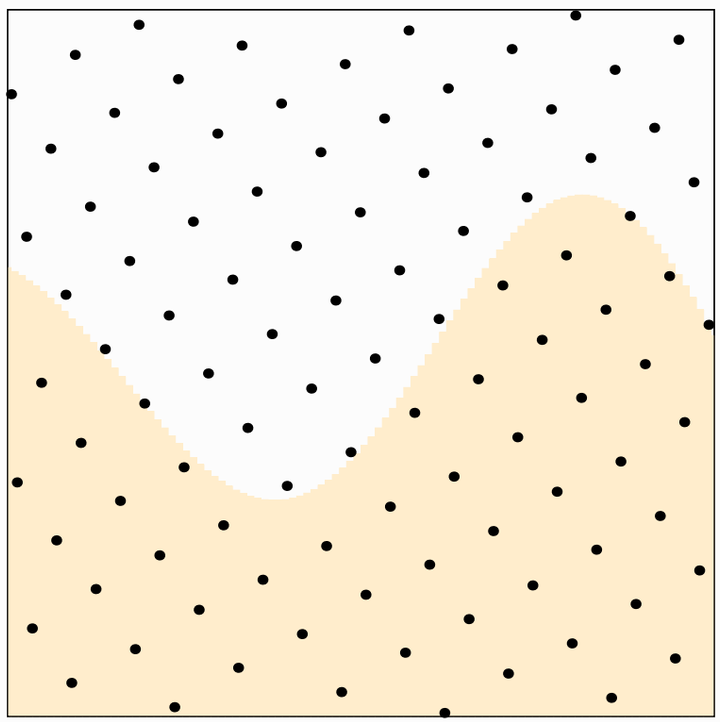
Halton sequence



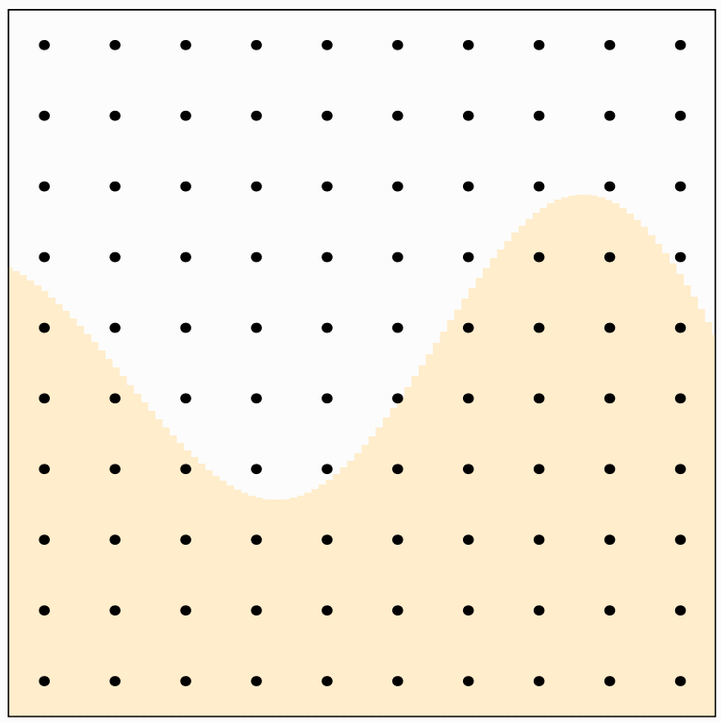
Hammersley set



Blue noise points

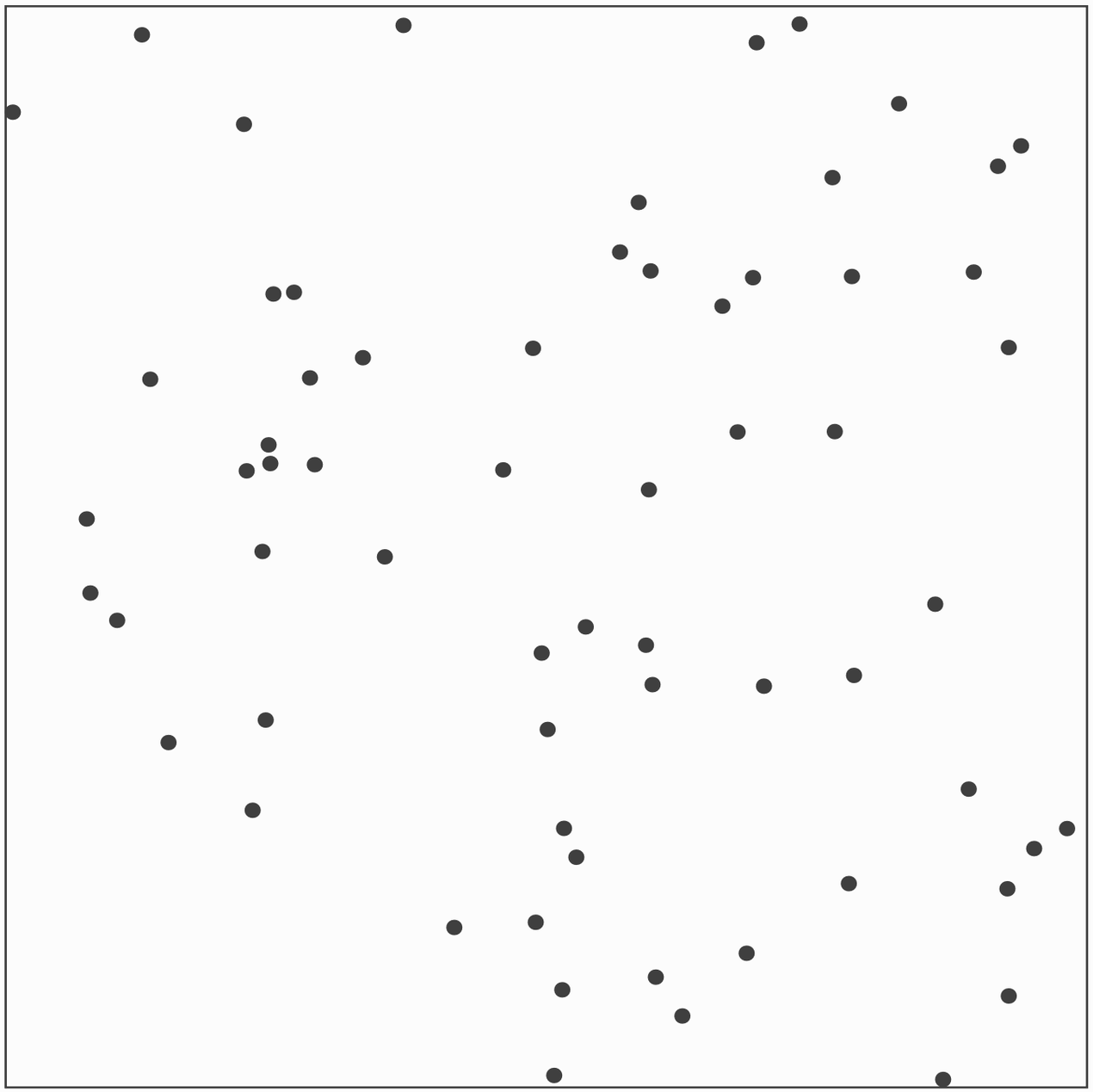


Lattice rule

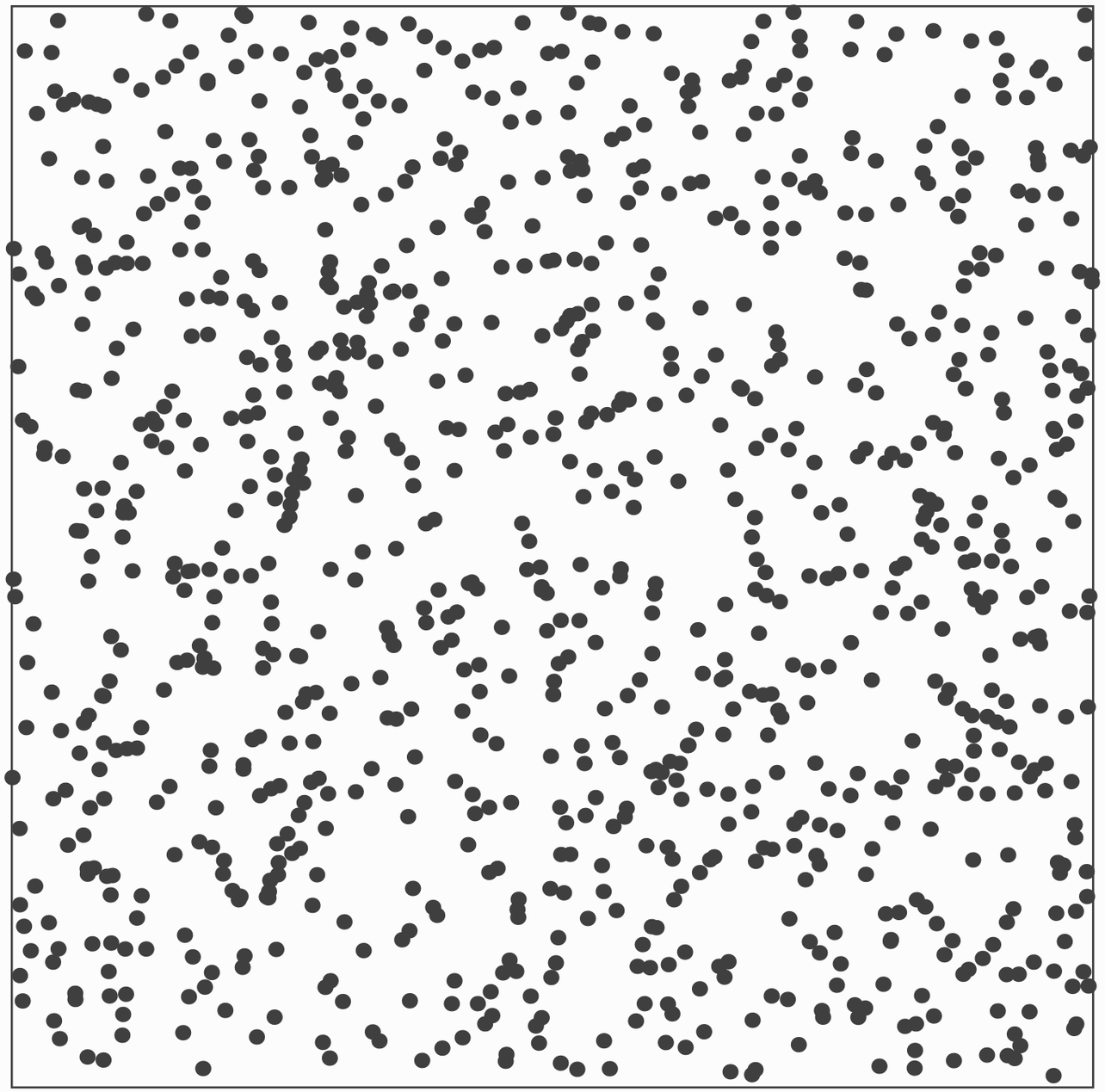


Cartesian grid

RANDOM SAMPLING



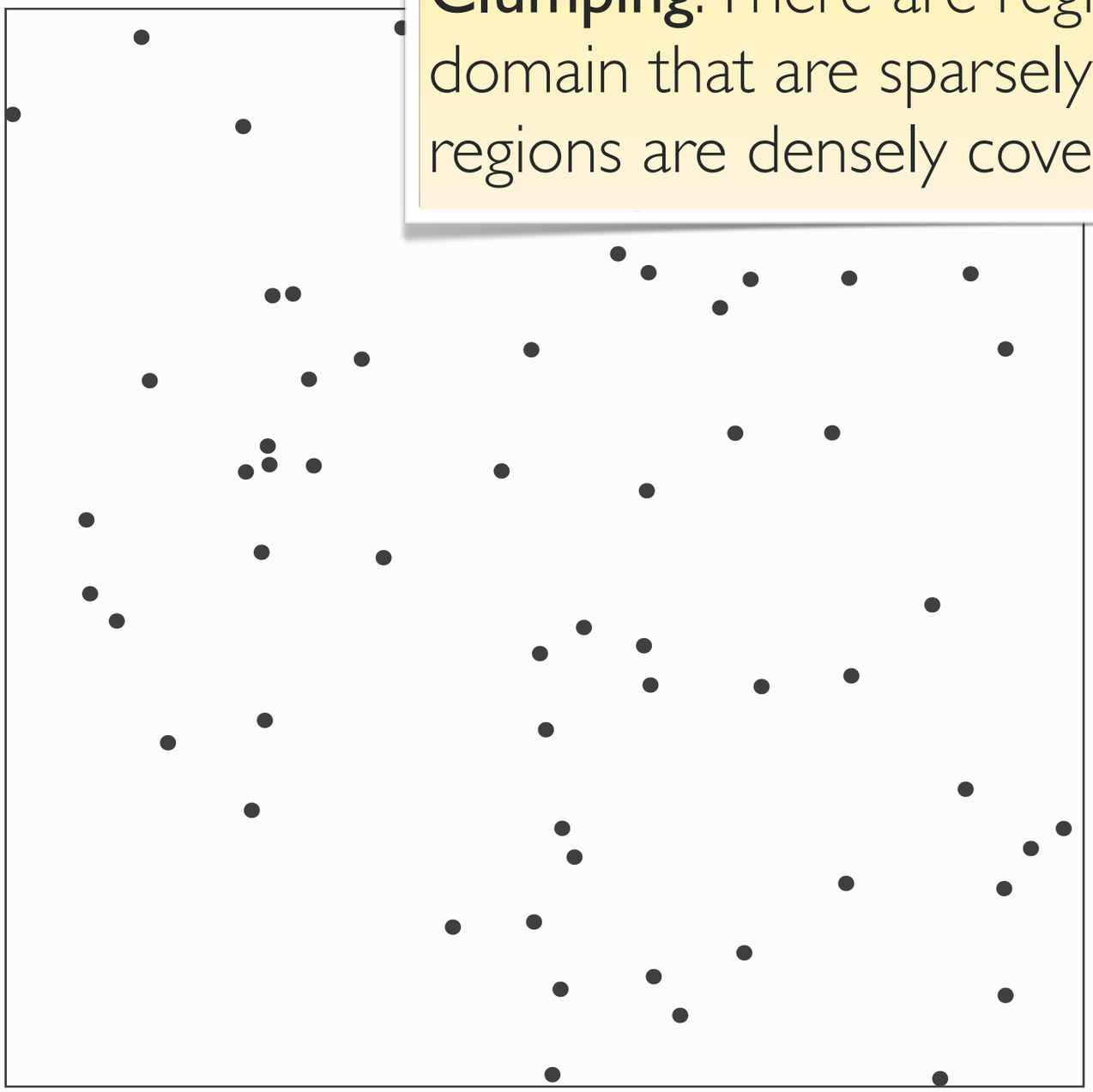
64 random points



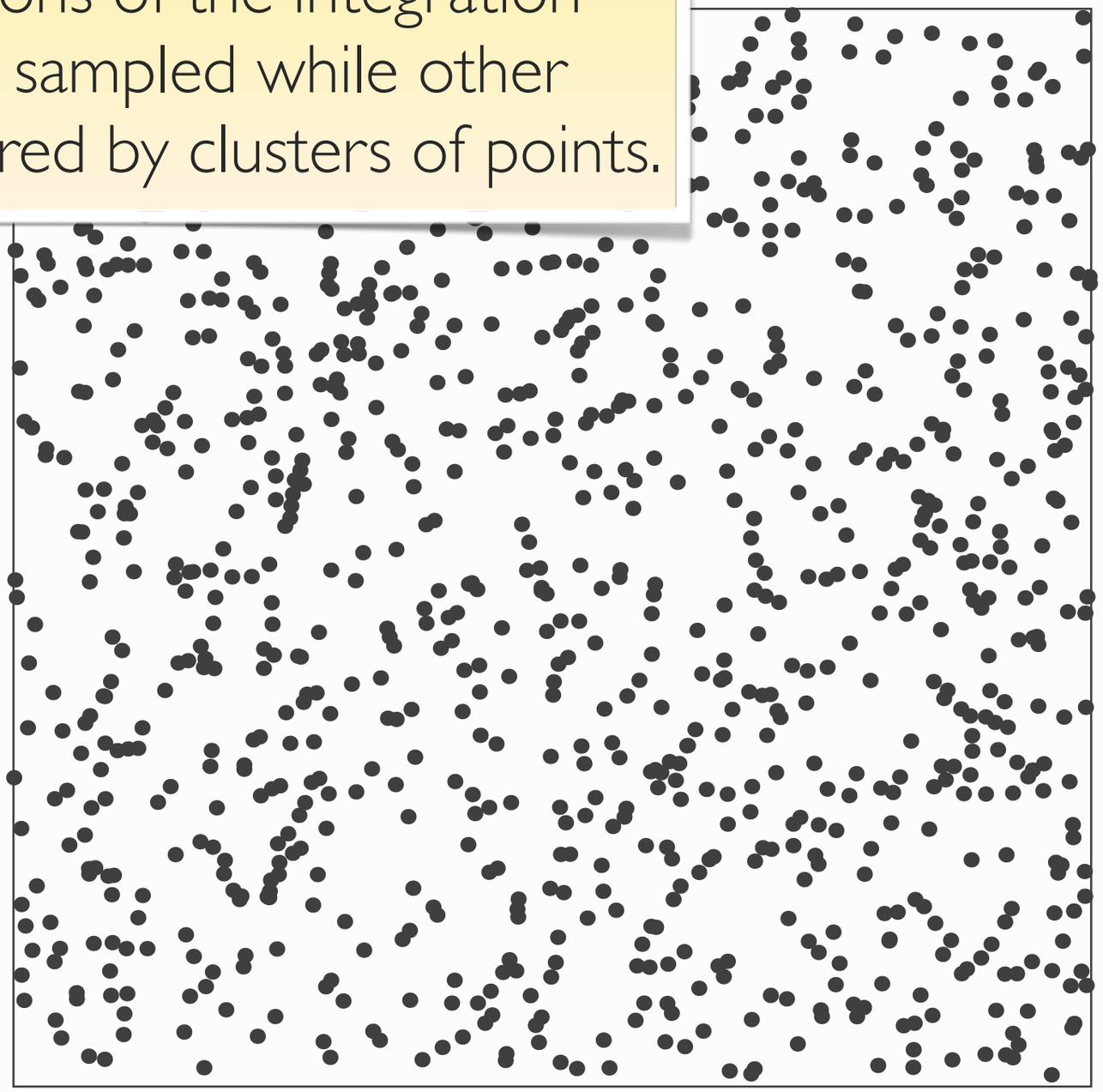
1000 random points

RANDOM SAMPLING

Clumping: There are regions of the integration domain that are sparsely sampled while other regions are densely covered by clusters of points.

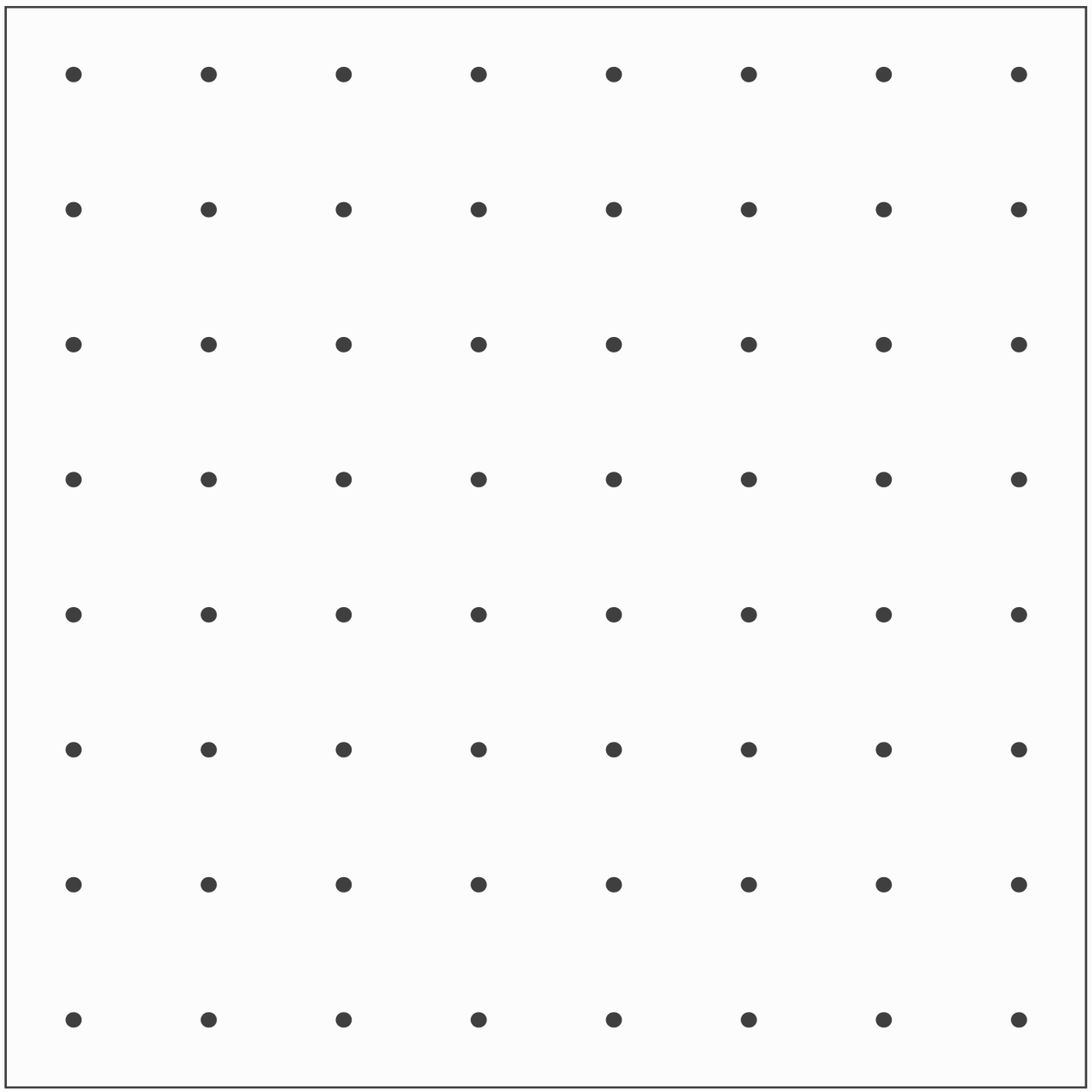


64 random points

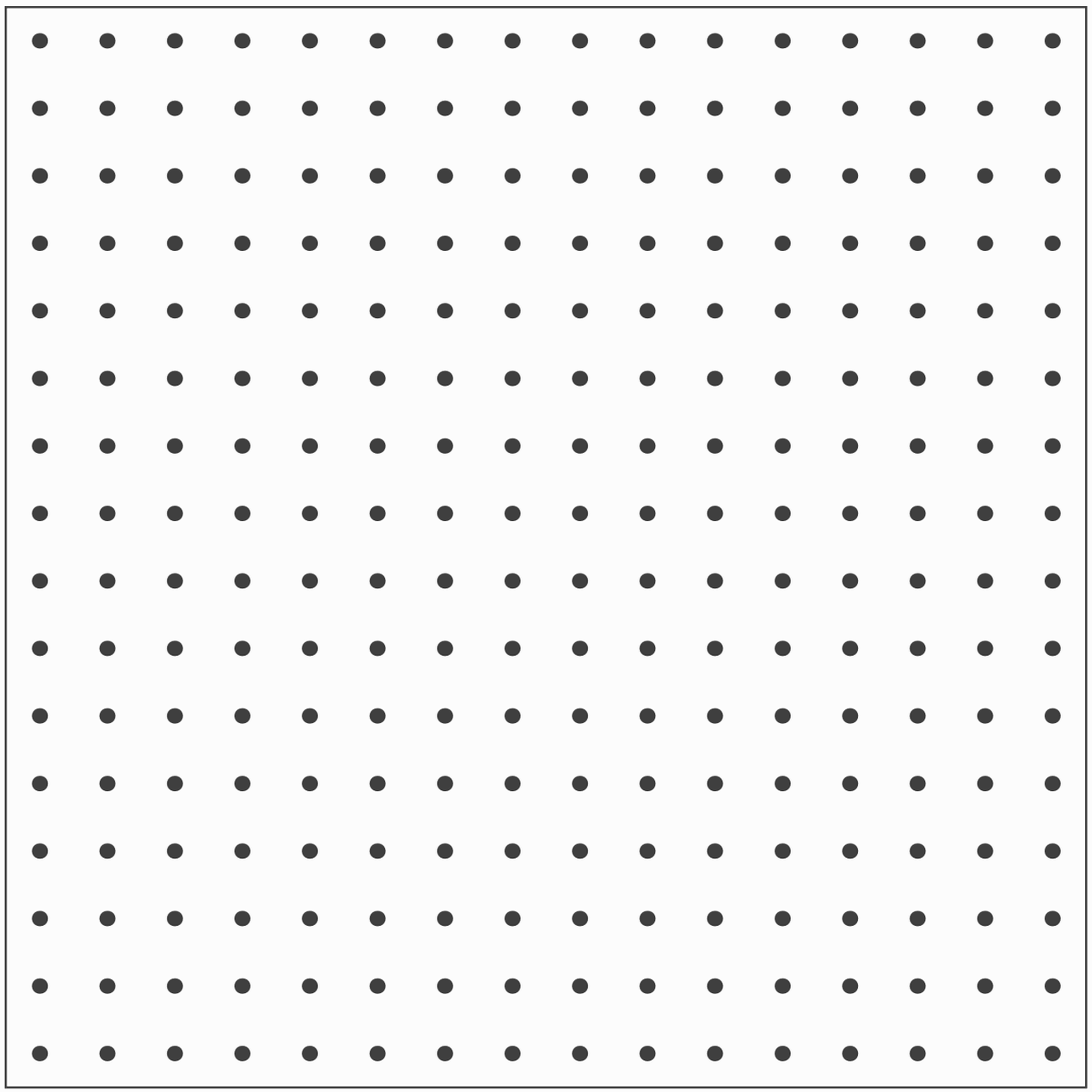


1000 random points

REGULAR **GRIDS**



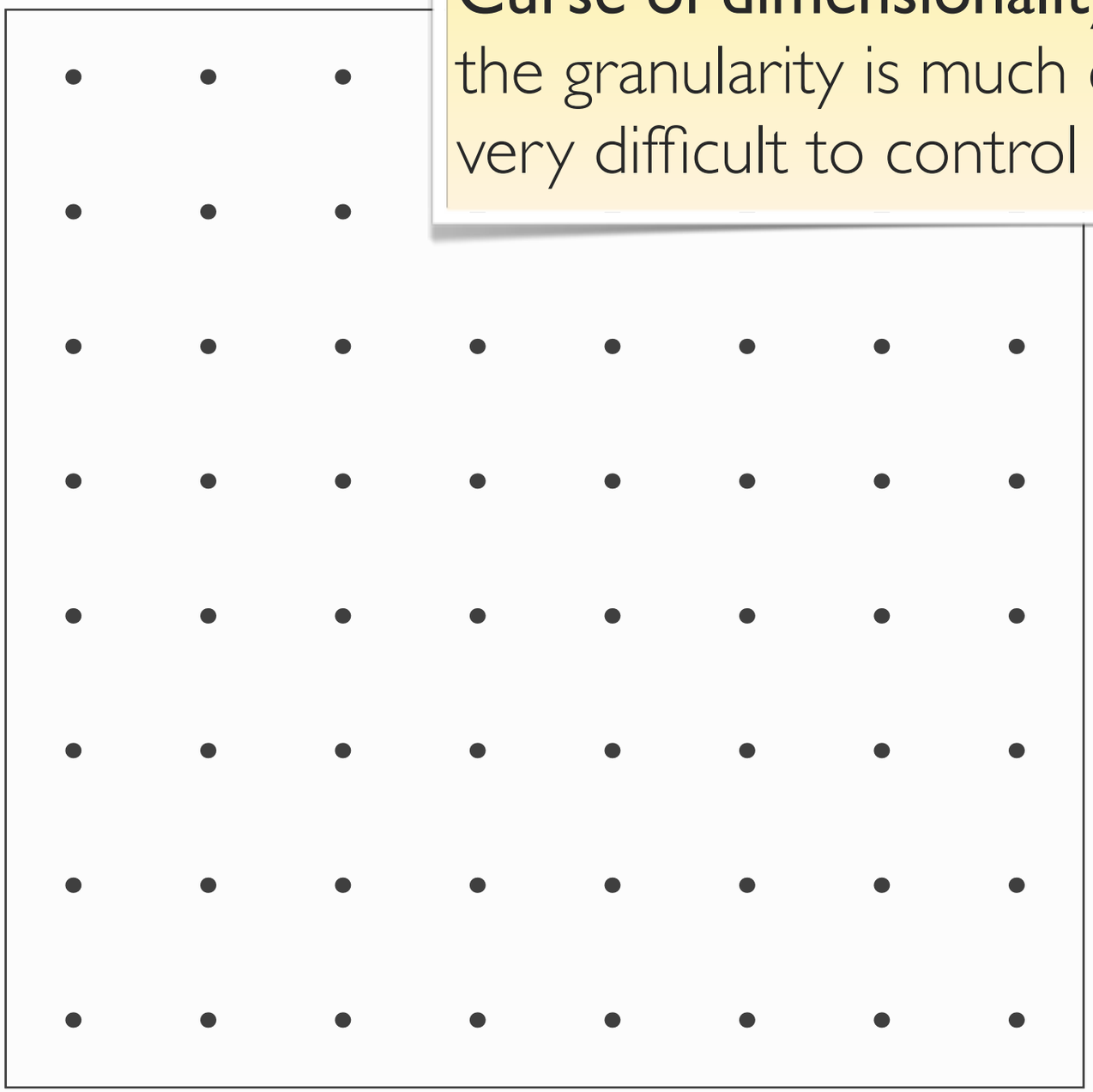
Cartesian grid, $8^2 = 64$ points



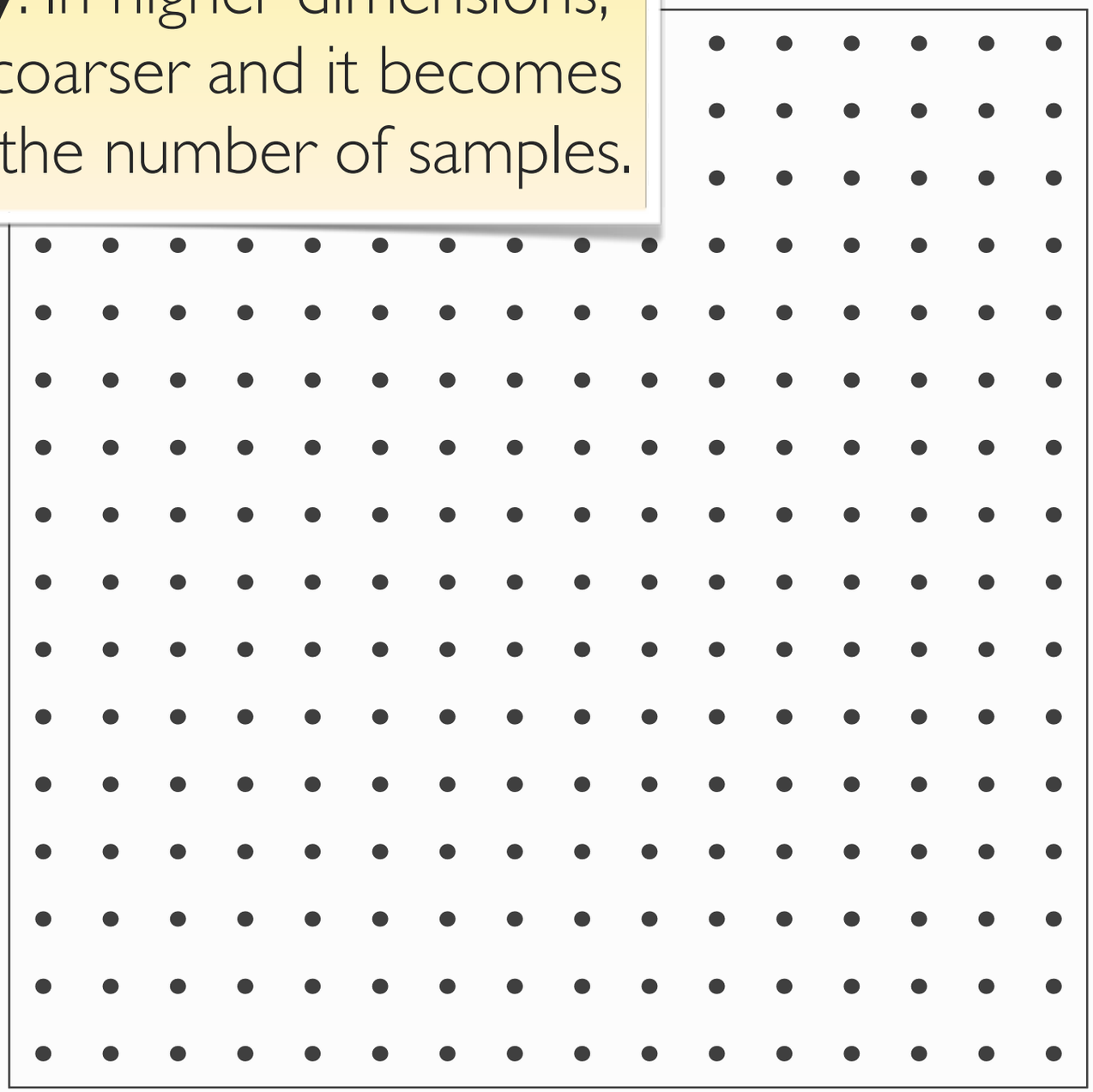
Cartesian grid, $16^2 = 256$ points

REGULAR **GRIDS**

Curse of dimensionality: In higher dimensions, the granularity is much coarser and it becomes very difficult to control the number of samples.

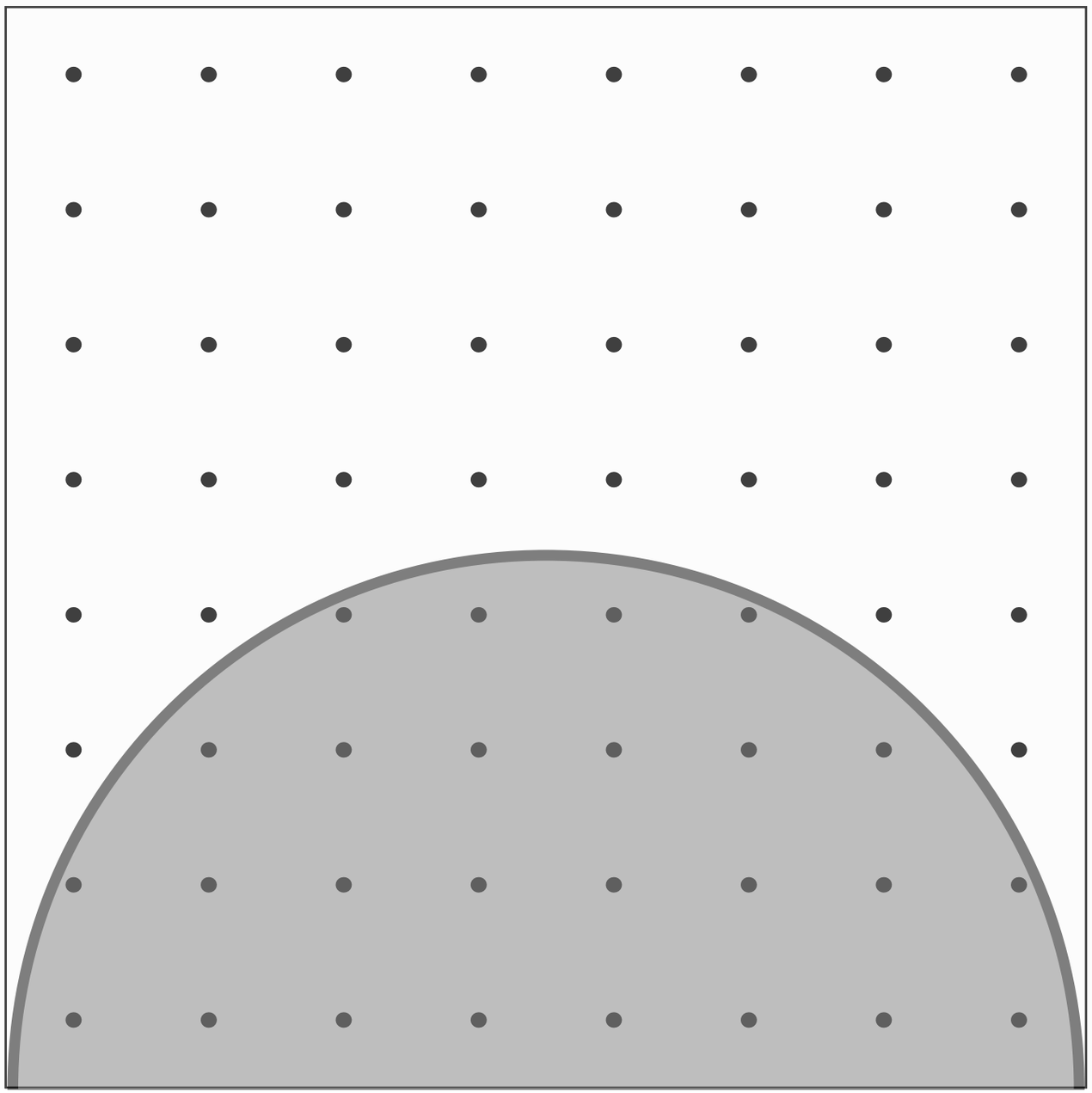


Cartesian grid, $8^2 = 64$ points

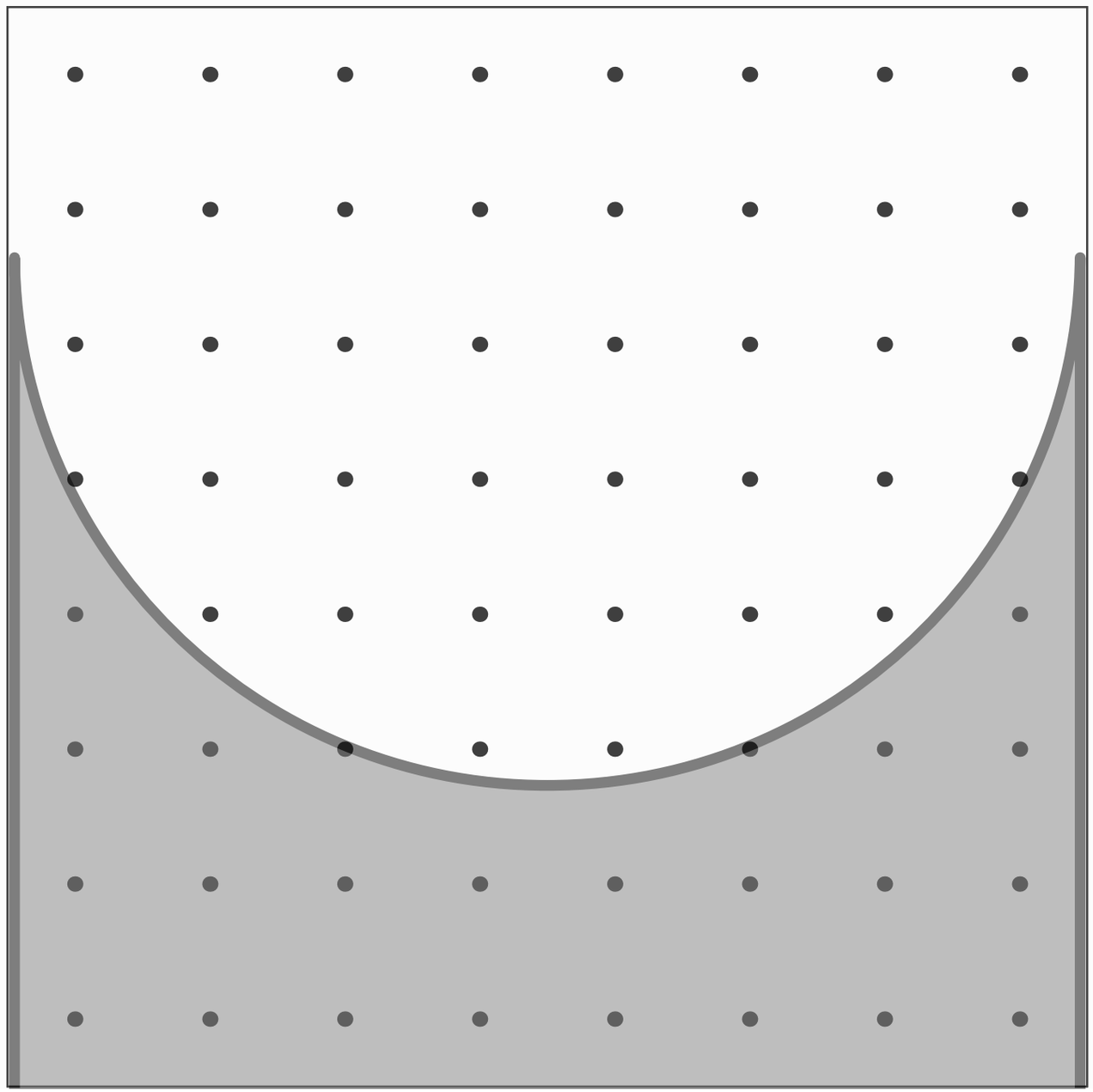


Cartesian grid, $16^2 = 256$ points

CONCAVITY -VEXITY **BIAS**



26 points inside a *concave* function



24 points inside a *convex* function

CONCAVITY -VEXITY **BIAS**

We want to integrate the area of a half unit disc (area: **Pi/2**) enclosed in a unit box (area: $\mathbf{2^2 = 4}$)

- Note that the ratio these two is **(Pi/2)/4 = Pi/8**

- When approximating the integral with $\mathbf{8^2 = 64}$ samples, the proportion of samples inside the shapes should be close to **64 x Pi/8 = 25.132** (not 24 or 26)

CONCAVITY -VEXITY **BIAS**

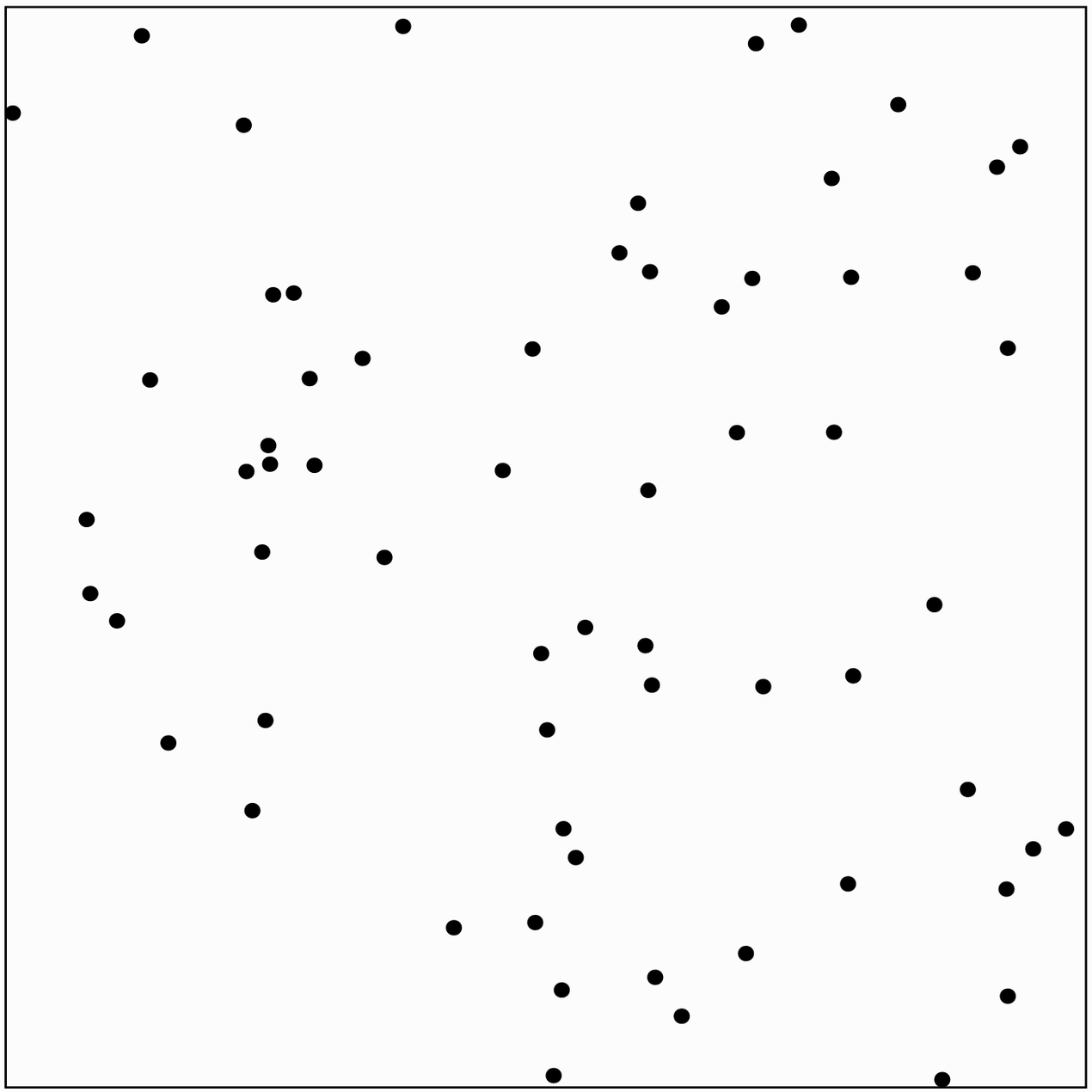
We want to integrate the area of a half unit disc (area: **Pi/2**) enclosed in a unit box (area: $\mathbf{2^2 = 4}$)

- Note that the ratio these two is **(Pi/2)/4 = Pi/8**

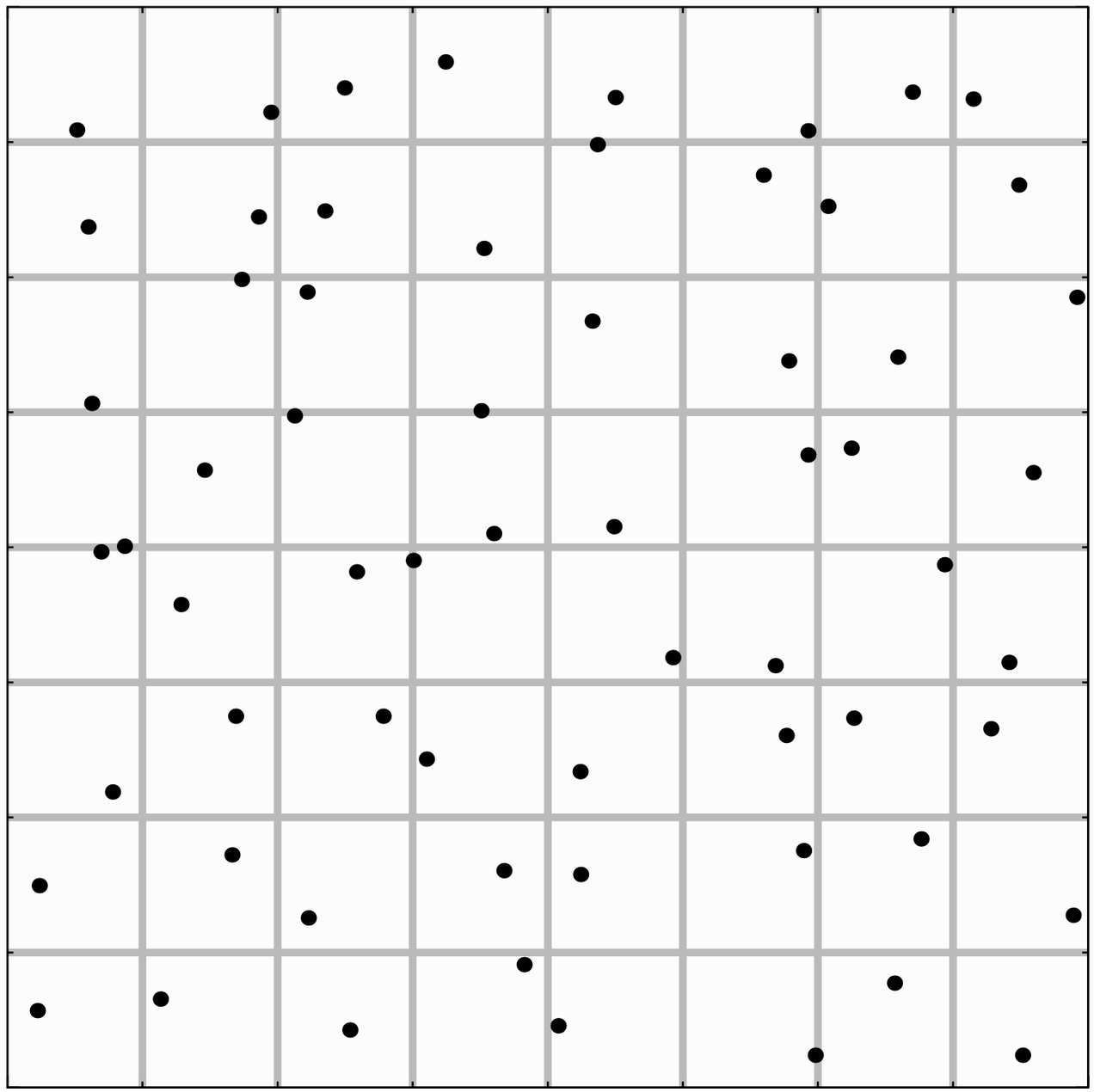
- When approximating the integral with $\mathbf{8^2 = 64}$ samples, the proportion of samples inside the shapes should be close to **64 x Pi/8 = 25.132** (not 24 or 26)

Bias: When using regular grids, the integral is systematically over or under estimated for globally concave or convex functions, respectively (*Dupire & Savine, 1998*)

STRATIFIED SAMPLES

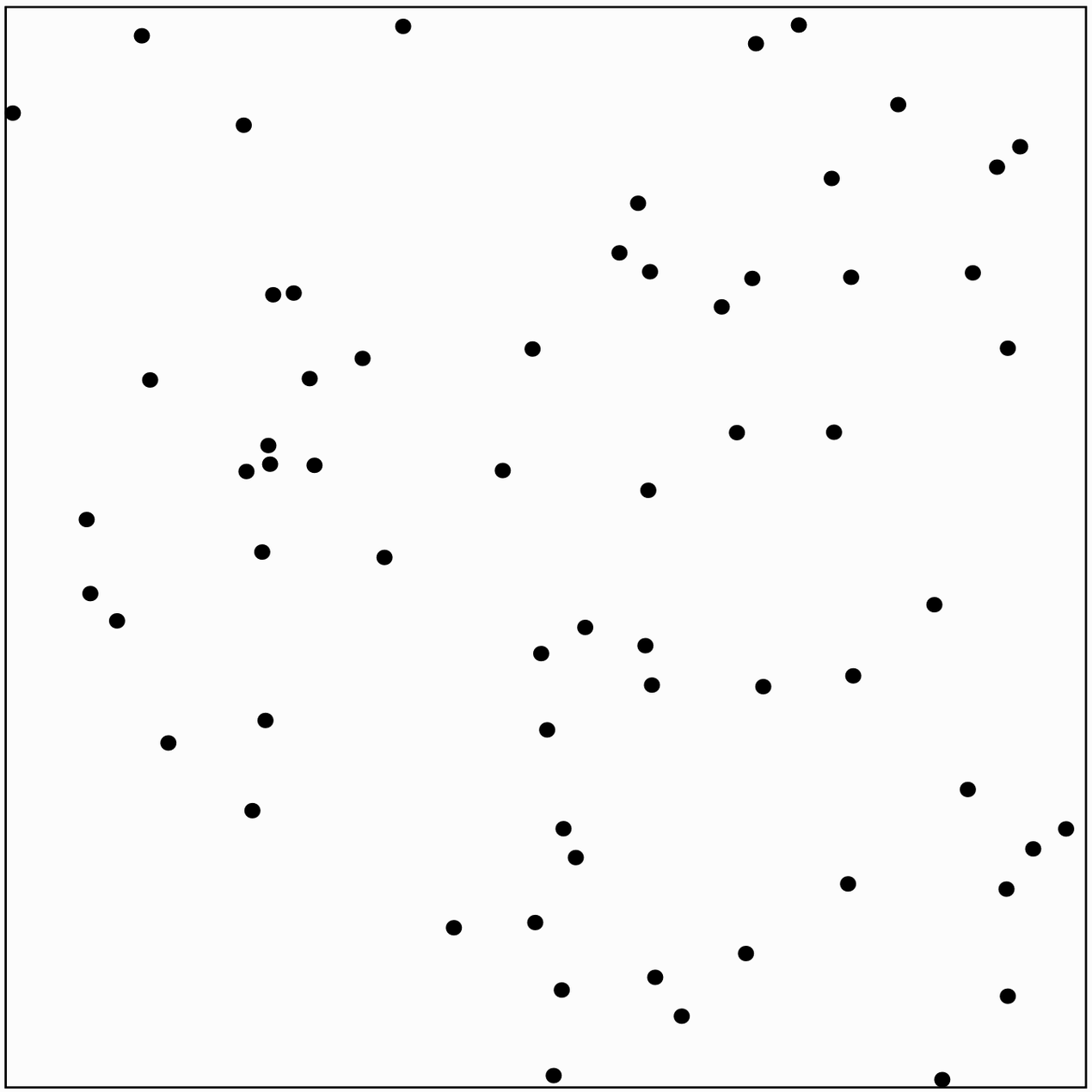


Random samples, 64 points

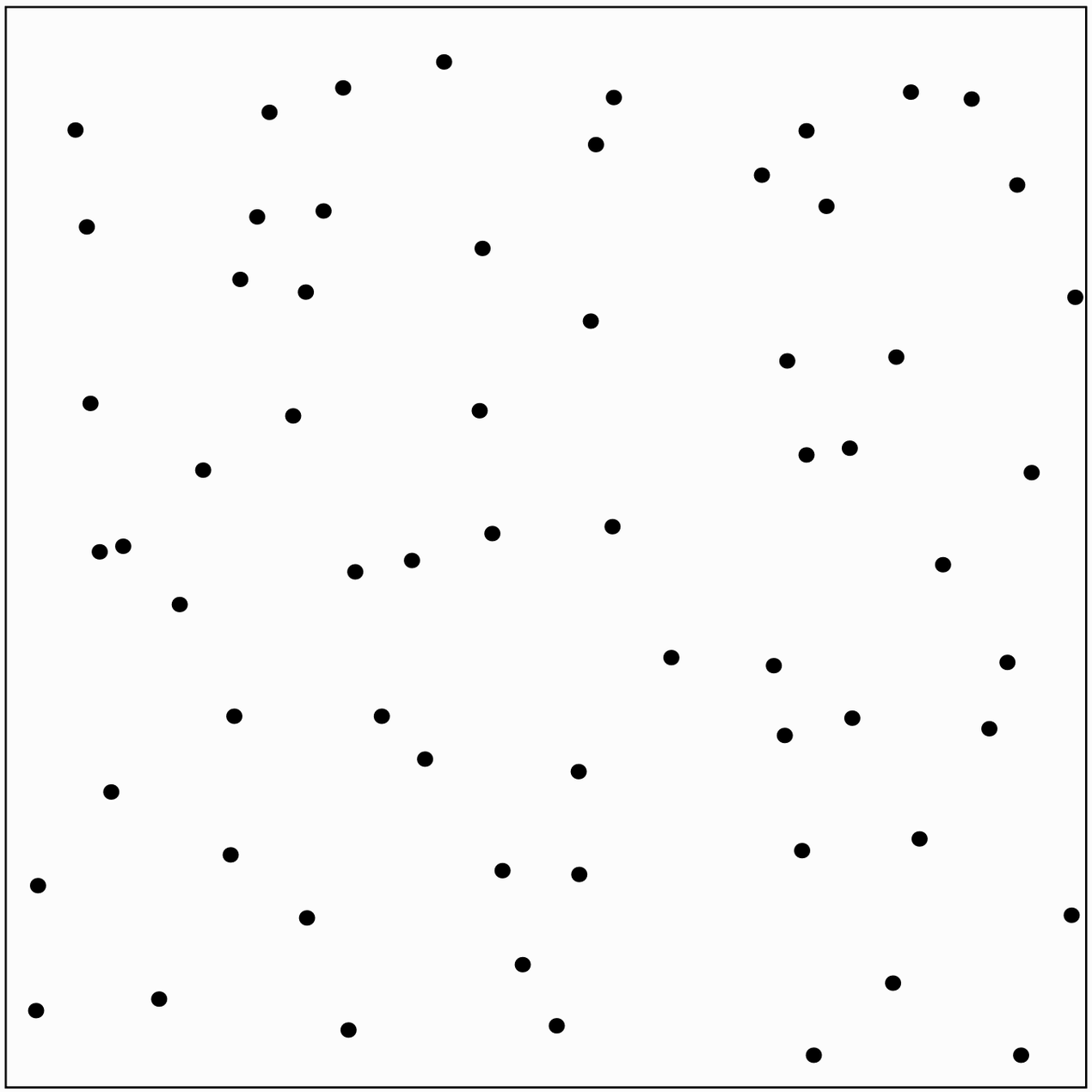


Stratified samples and strata , 64 points

STRATIFIED SAMPLES



Random samples, 64 points



Stratified samples, 64 points

PSEUDO-RANDOM

```
unsigned int state[624];
unsigned int index;

unsigned int twist(const unsigned int u, const unsigned int v) {
    const unsigned int twisted = ((u & 0x80000000) | (v & 0x7FFFFFFF)) >> 1;
    if(v & 1) return twisted ^ 2567483615ul;
    return twisted;
}

void update() {
   index = 0;
   for(unsigned int i = 0; i < 624 - 397; ++i)
       state[i] = state[i + 397] ^ twist(state[i], state[i + 1]);

   for(unsigned int i = 624 - 397; i < 623; ++i)
       state[i] = state[i + 397 - 624] ^ twist(state[i], state[i + 1]);

   state[623] = state[396] ^ twist(state[623], state[0]);
}

unsigned int random() {
   if(index == 624) update();
   unsigned int r = state[index++];
   r ^= (r >> 11);
   r ^= (r << 7) & 2636928640ul;
   r ^= (r << 15) & 4022730752ul;
   r ^= (r >> 18);
   return r;
}
```

Mersene twister number generator (*Matsumoto & Nishimura 2002*)

"*For every randomized algorithm, there is a clever deterministic one.*"

Harald Niederreiter, 1998

LOW-DISCREPANCY POINTS

dis·crep·an·cy (*noun*)
A lack of compatibility or similarity between two or more facts

ORIGIN *early 17th cent.: from Latin* ***discrepantia****, from* ***discrepare 'be discordant,'*** *from* ***dis- 'apart, away'*** + ***crepare 'to creak.'***

dis·crep·an·cy (*noun*)
A lack of ~~compatibility or~~ similarity between ~~two or more~~ facts

ORIGIN *early 17th cent.: from Latin* ***discrepantia****, from* ***discrepare 'be discordant,'*** *from* ***dis- 'apart, away'*** + ***crepare 'to creak.'***

dis·crep·an·cy (*noun*)
A lack of ~~compatibility or~~ similarity between ~~two or more~~ ~~facts~~ points

ORIGIN *early 17th cent.: from Latin* ***discrepantia****, from* ***discrepare 'be discordant,'*** *from* ***dis- 'apart, away'*** + ***crepare 'to creak.'***

DISCREPANCY **CRITERIA**

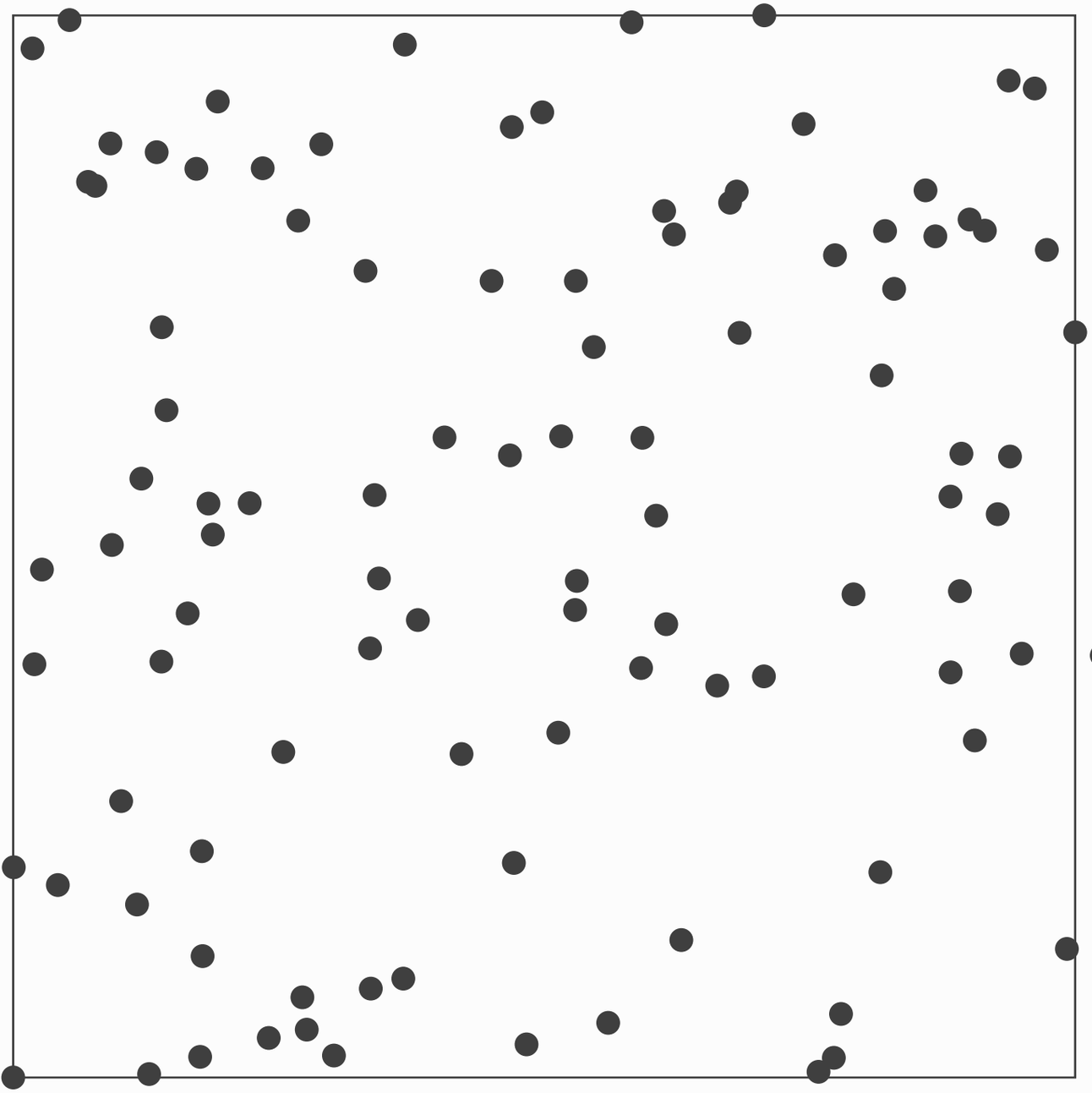
Number Theory

Box discrepancy, local discrepancy, star discrepancy, half-plane discrepancy, edge discrepancy, strip discrepancy, ...

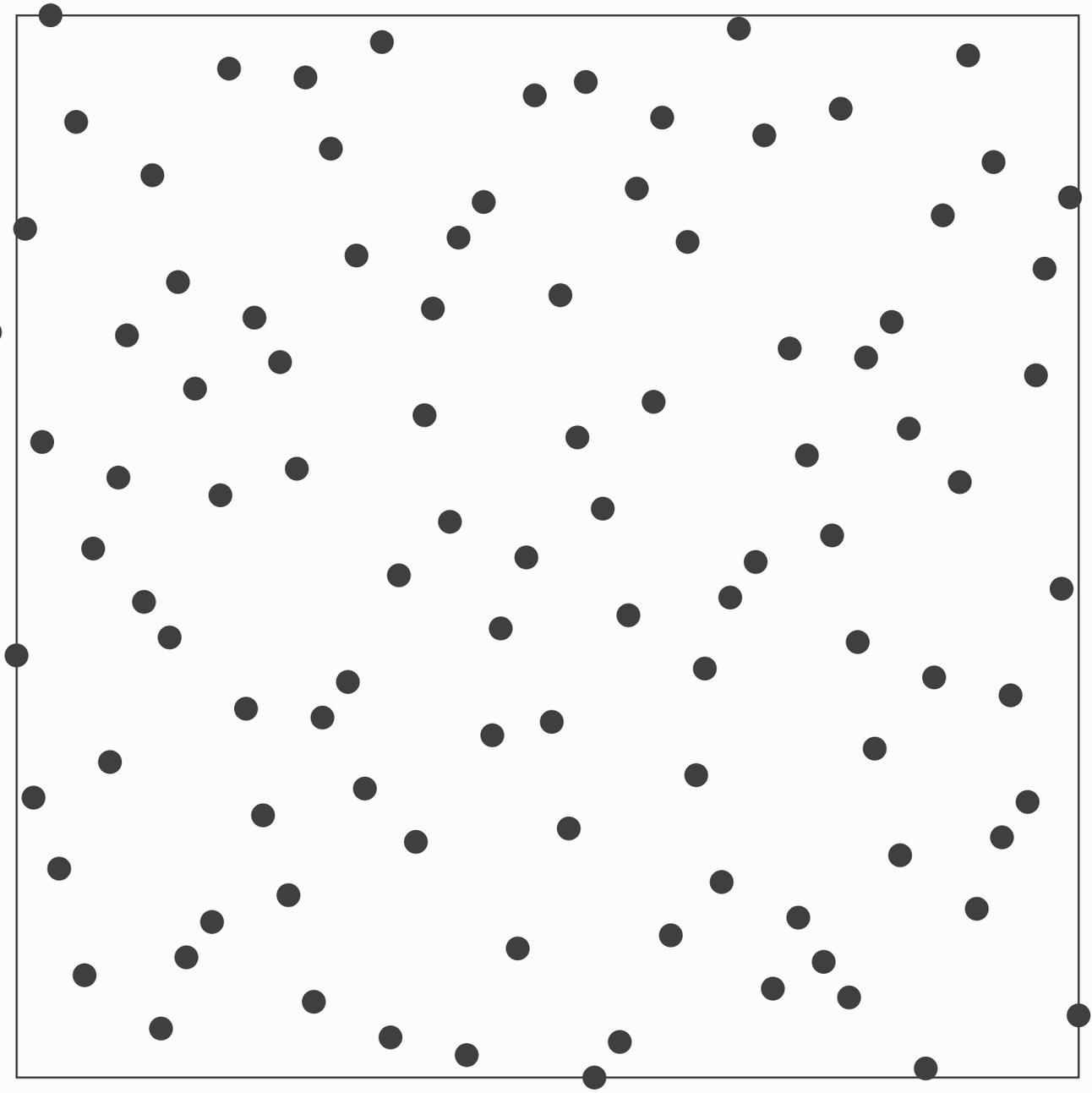
Computer Graphics

Stratified samples, jittering, n-rooks, halftoning, Blue noise spectrum, Poisson disk sampling, minimum maximum distance, ...

BOX DISCREPANCY

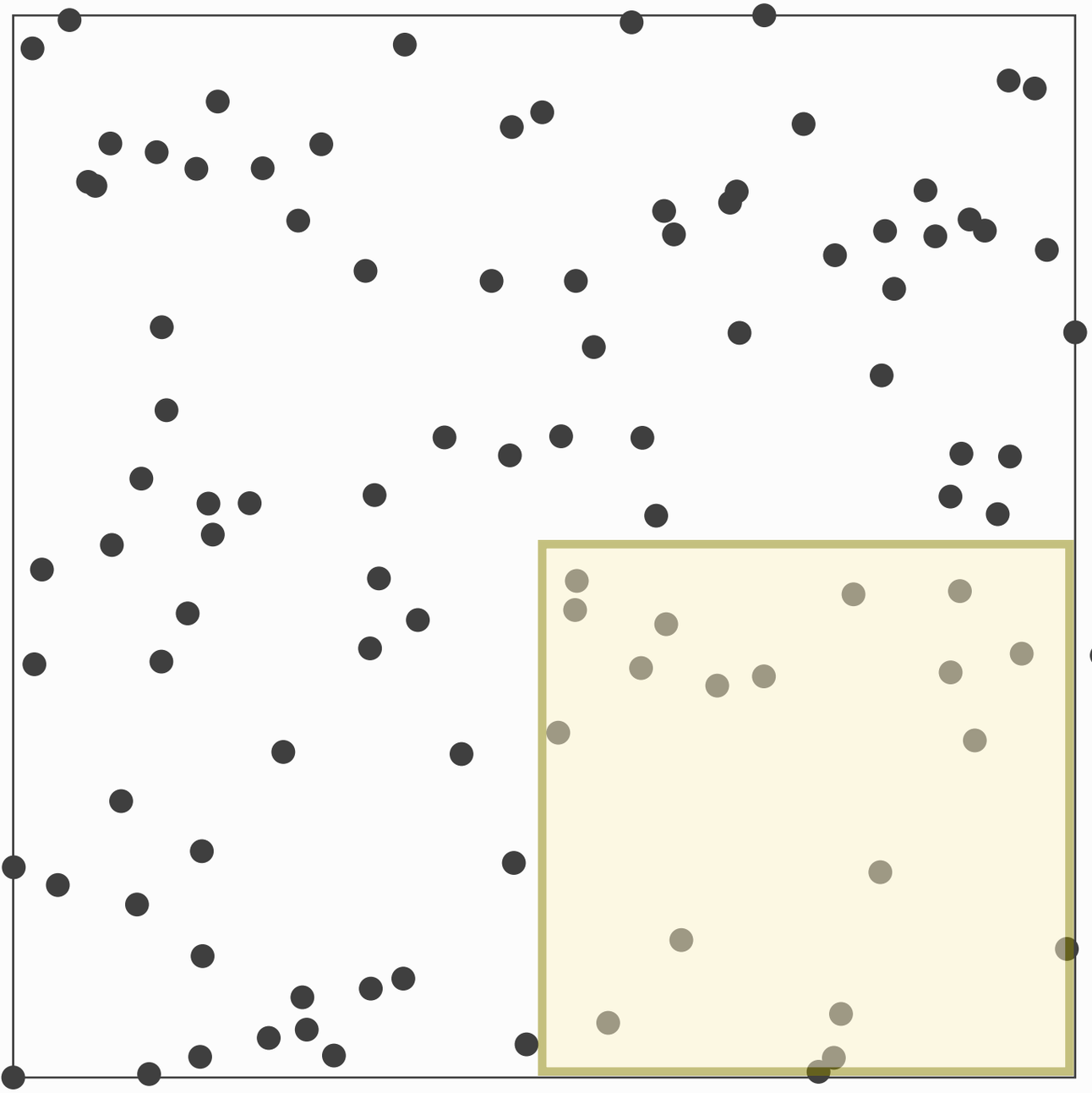


Random (100 points)

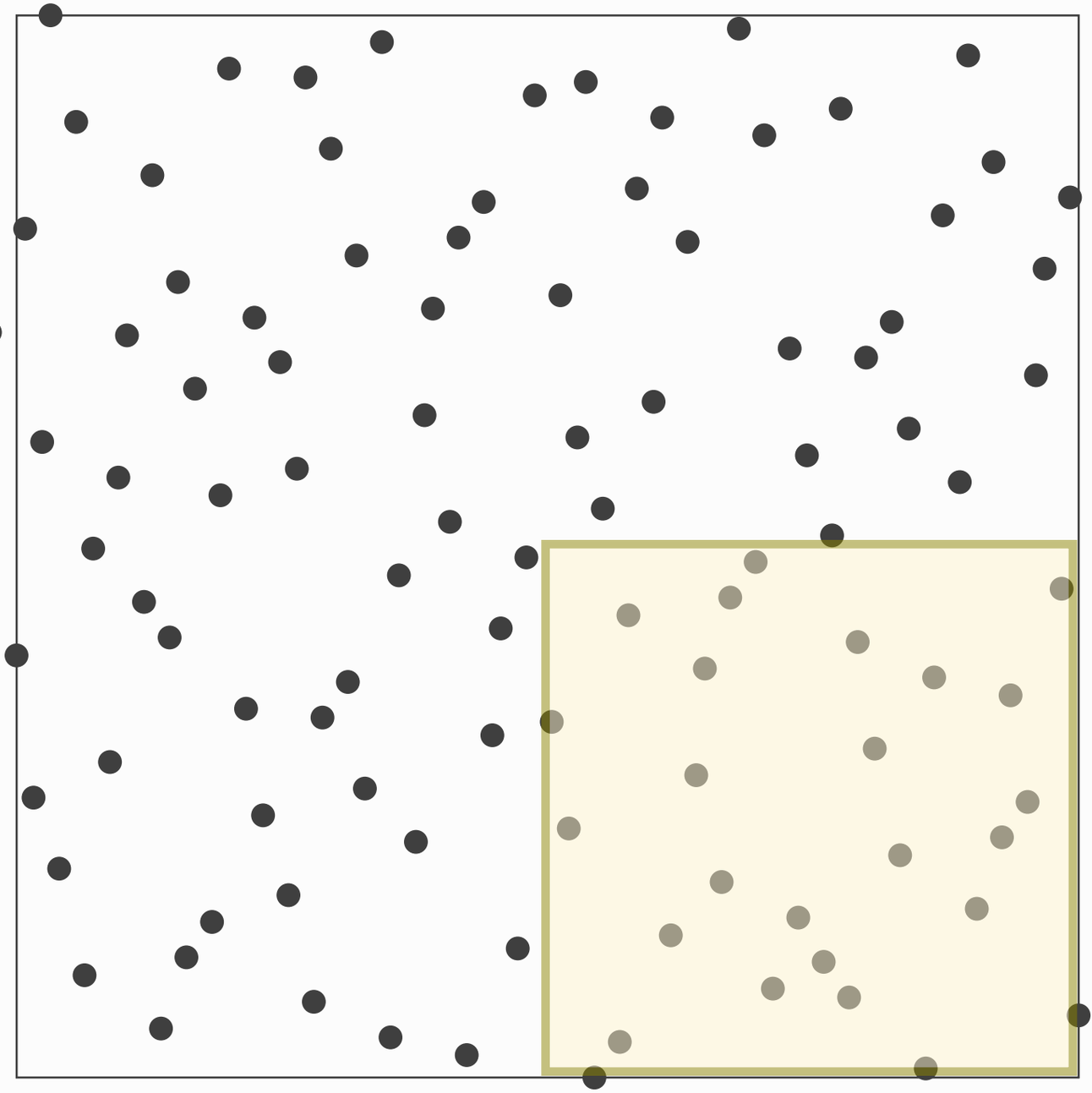


Halton sequence (100 points)

BOX DISCREPANCY

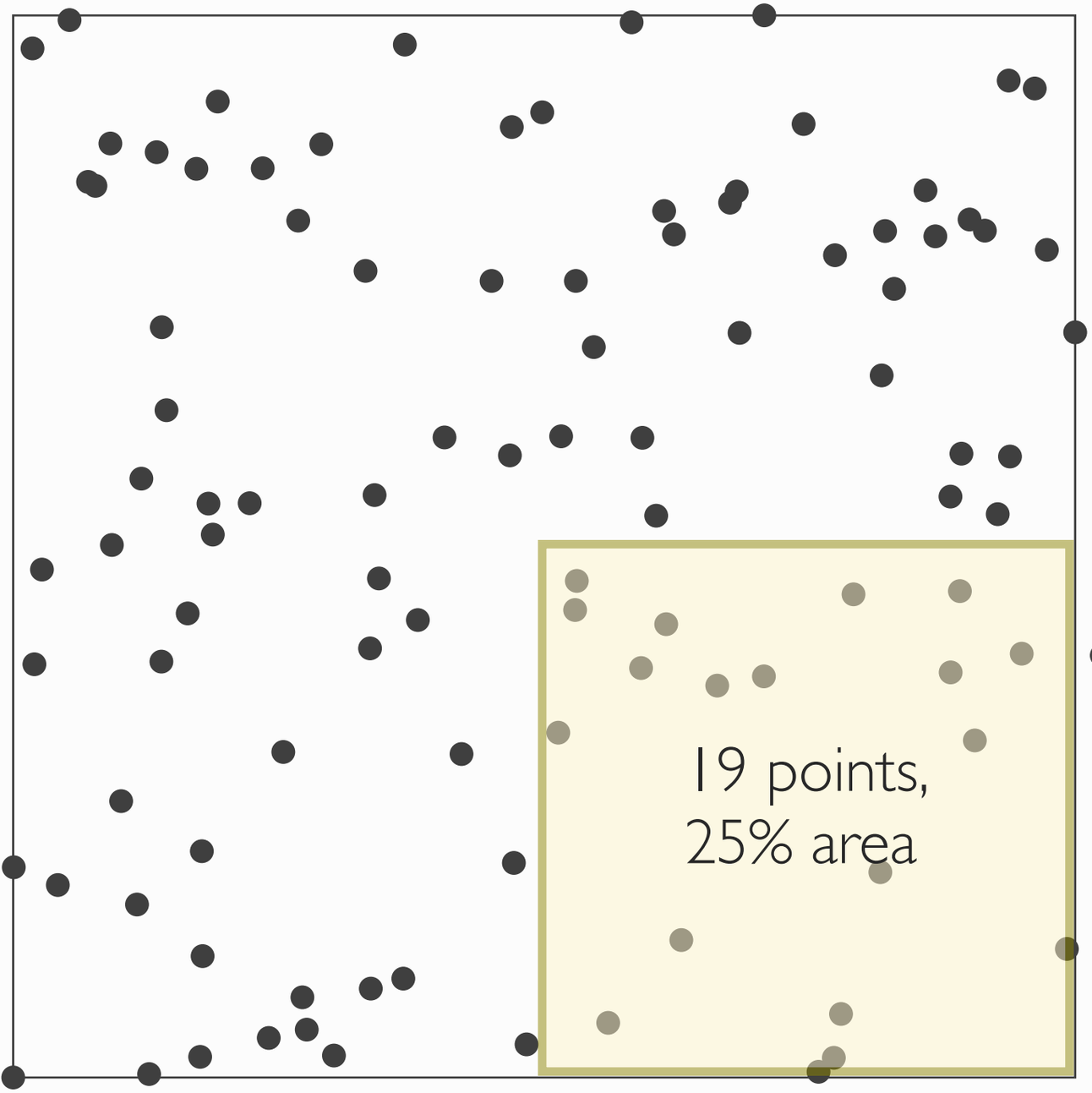


Random (100 points)

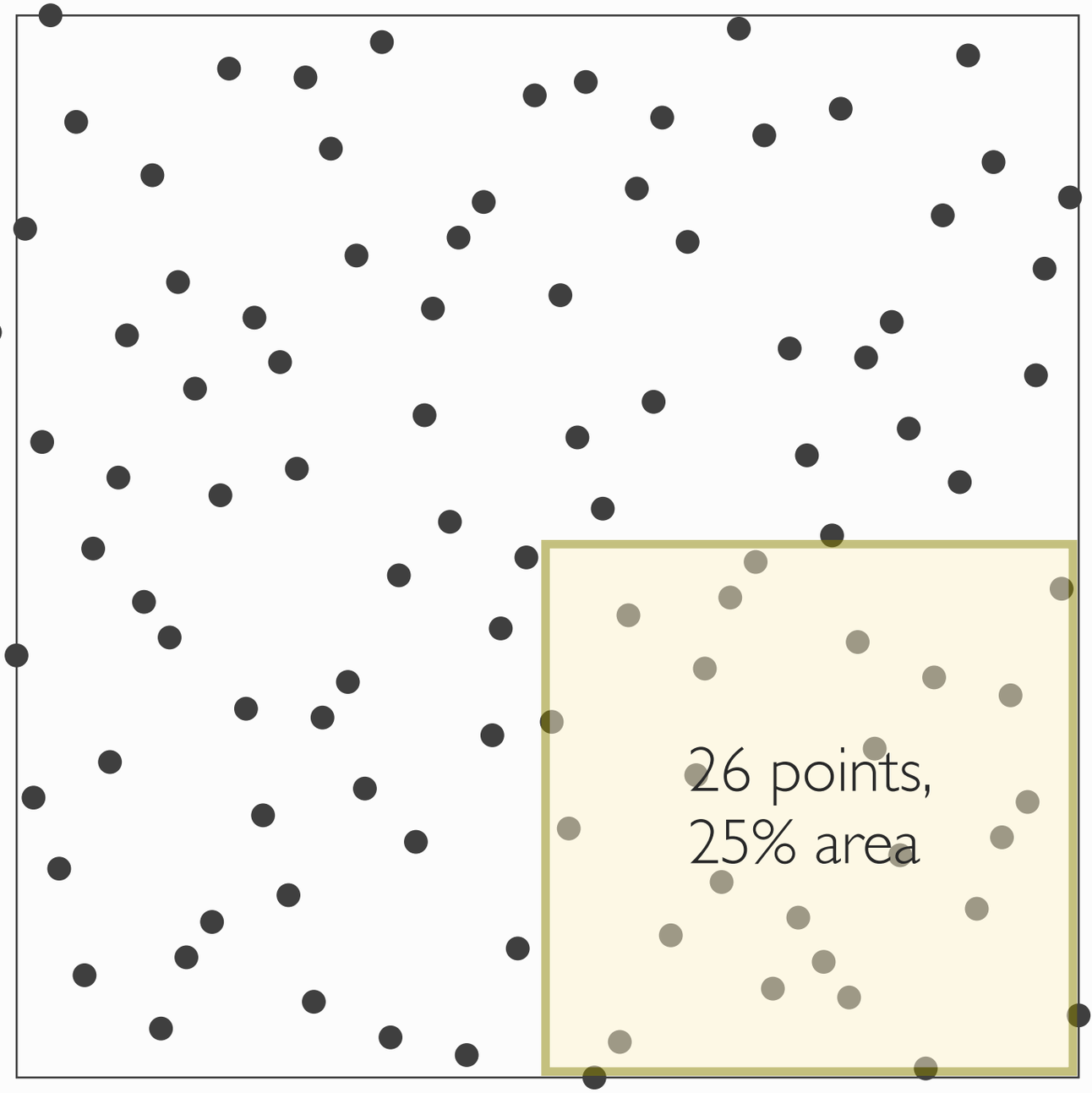


Halton sequence (100 points)

BOX DISCREPANCY

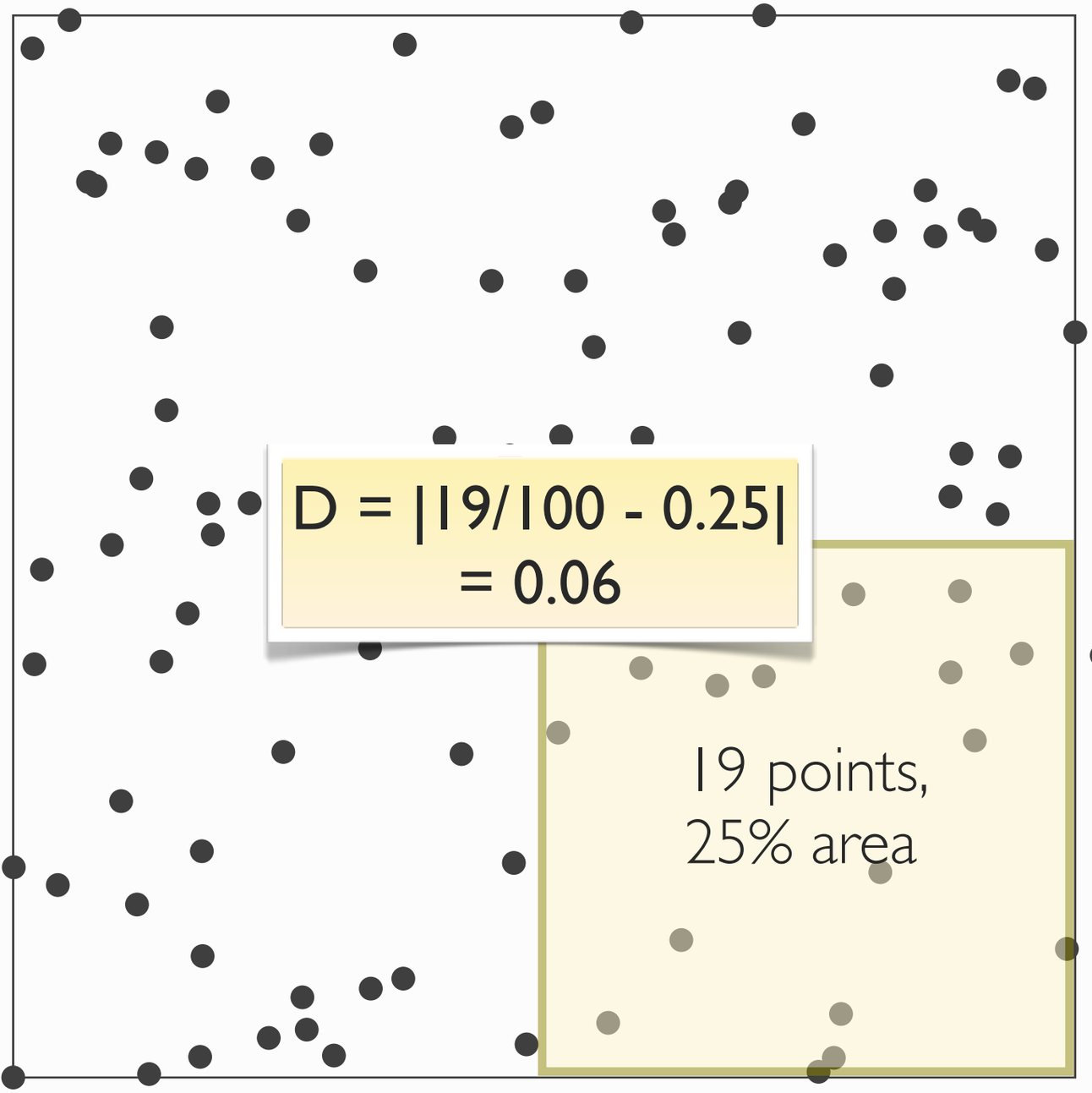


Random (100 points)

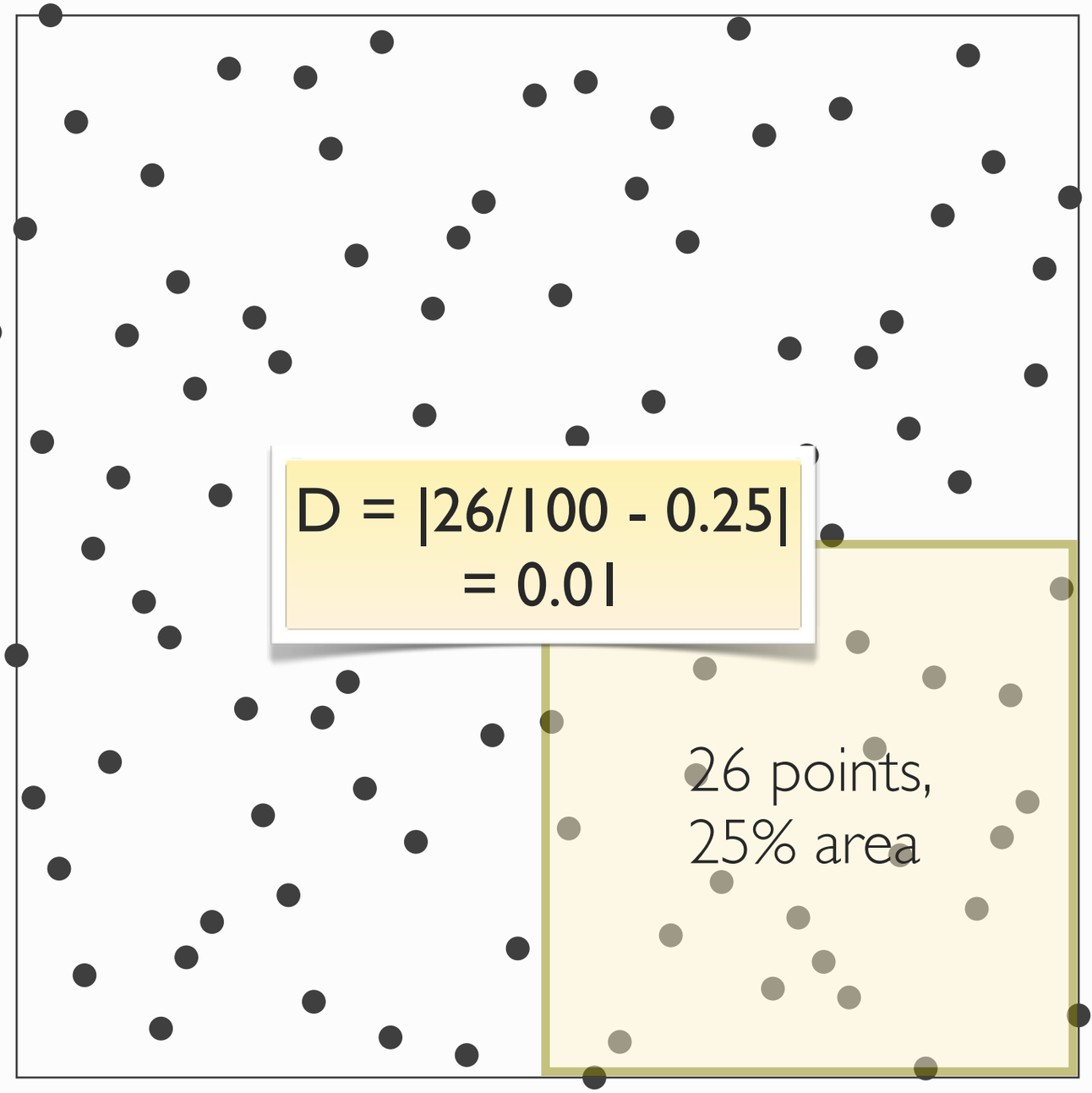


Halton sequence (100 points)

BOX DISCREPANCY

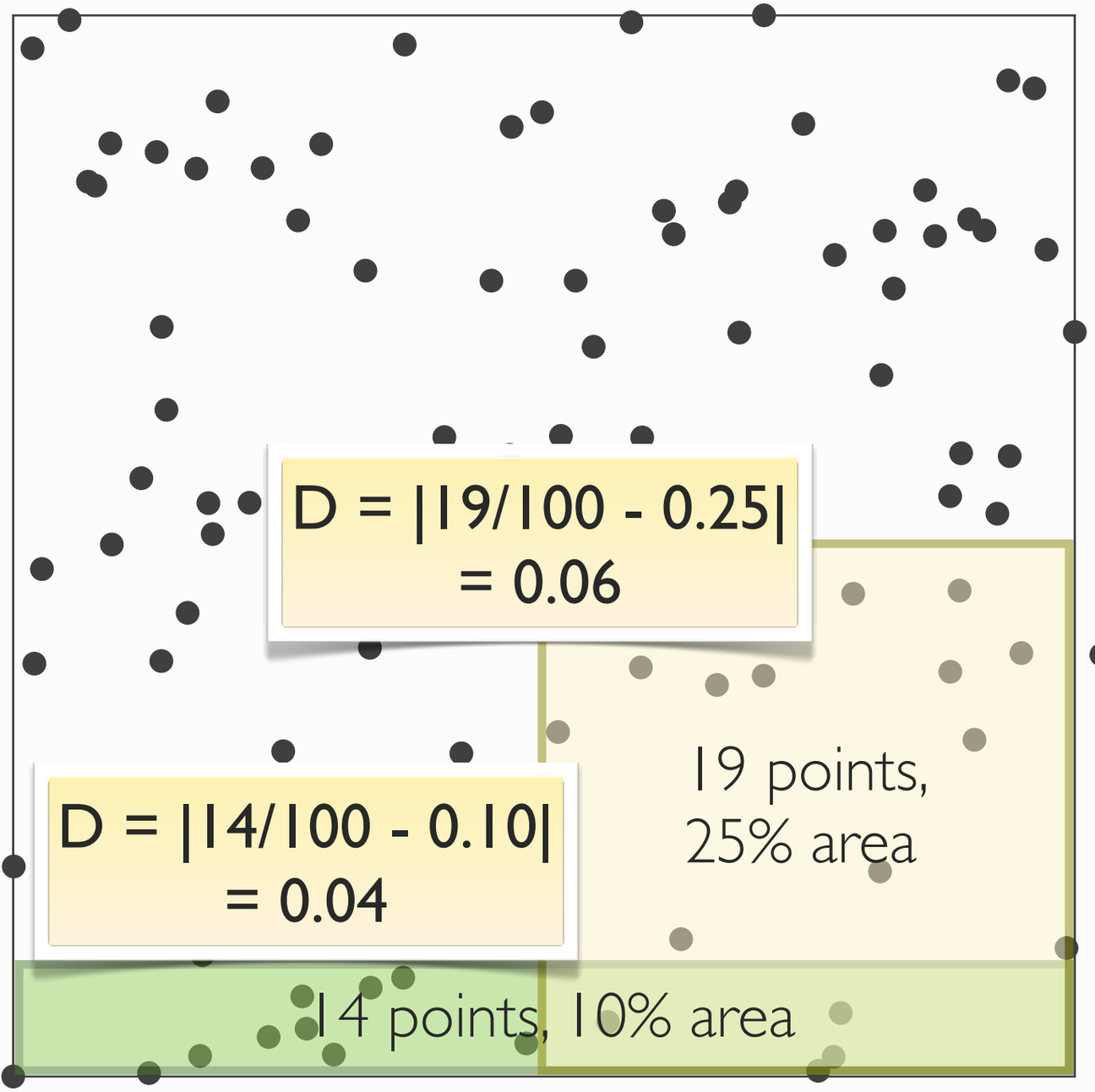


Random (100 points)

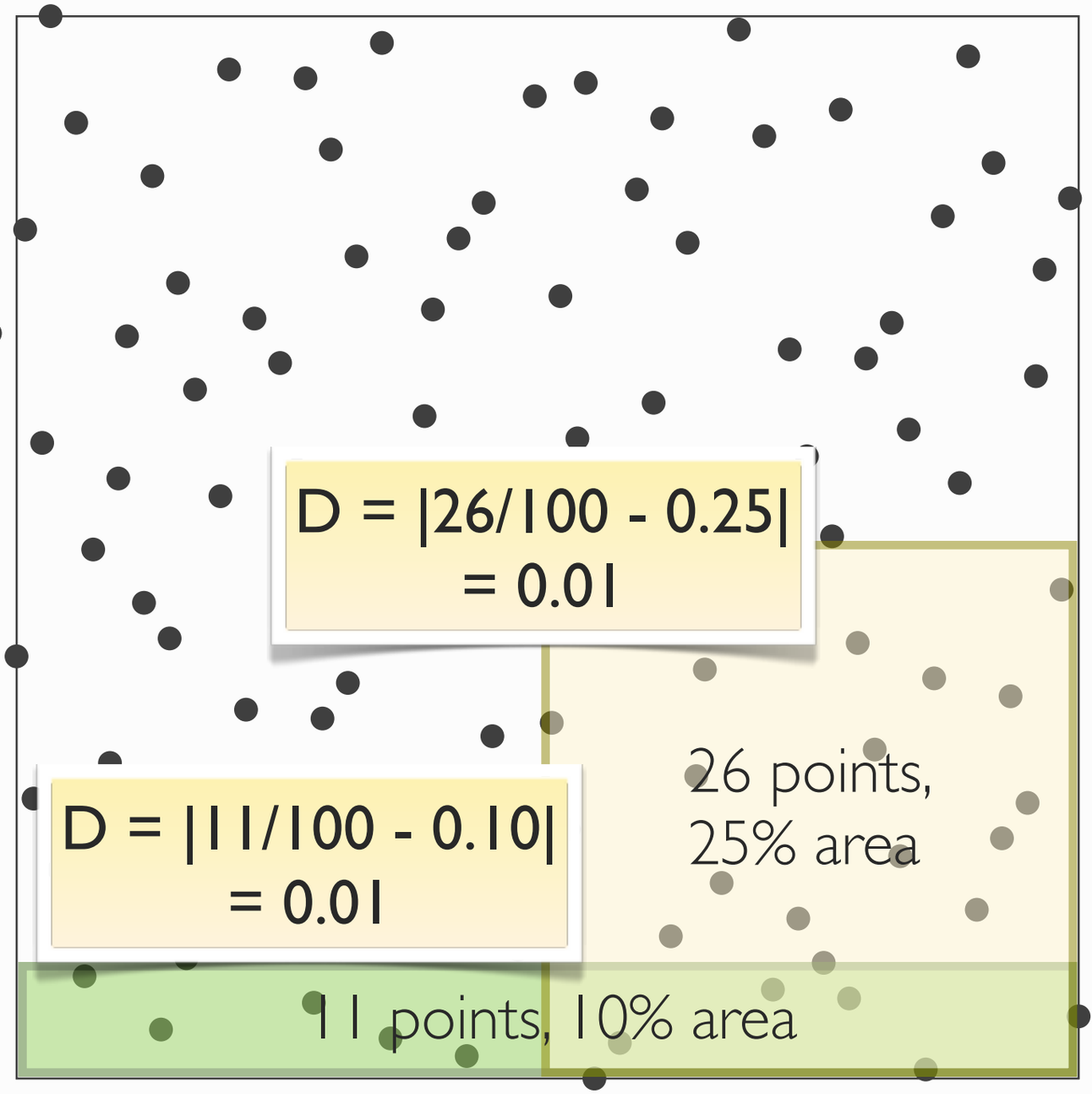


Halton sequence (100 points)

BOX DISCREPANCY



Random (100 points)



Halton sequence (100 points)

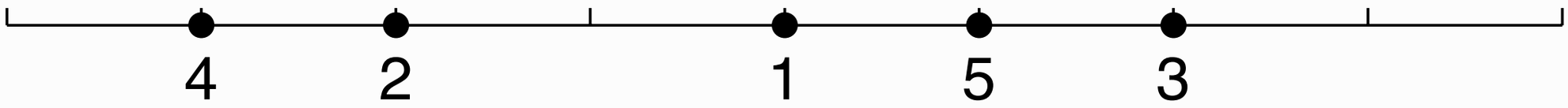
STAR DISCREPANCY

The discrepancy of the point set $S = \{x_1, .., x_N\}$ relative to a collection A of axis-aligned boxes in $[0, 1)^d$ is

$$D^*(S; A) = \sup_{a \in A} \left| \frac{\#\{x_i \in a\}}{N} - \mathrm{vol}(a) \right|$$

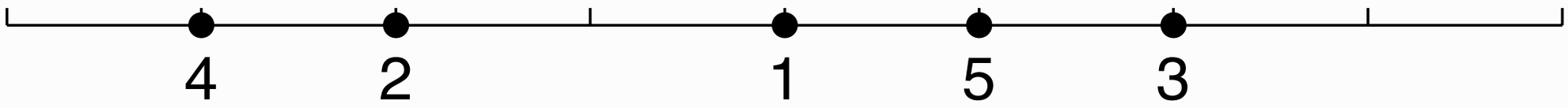
IN **ONE** DIMENSION

VAN DER CORPUT SEQ.



van der Corput sequence

VAN DER CORPUT SEQ.



van der Corput sequence

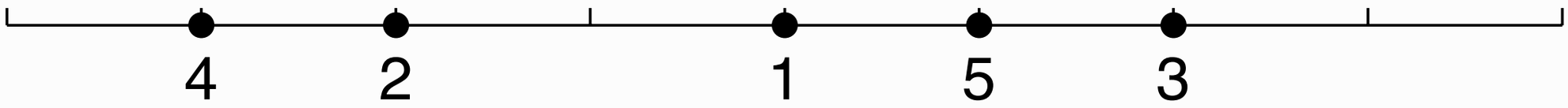
$$H_2(i) =$$

```
FUNCTION Halton(index)
  result = 0
  f = 1 / 2
  i = index
  WHILE(i > 0)
    result += f * (i % 2)
    i = floor(i / 2)
    f = f / 2;
  END
  RETURN result
END
```

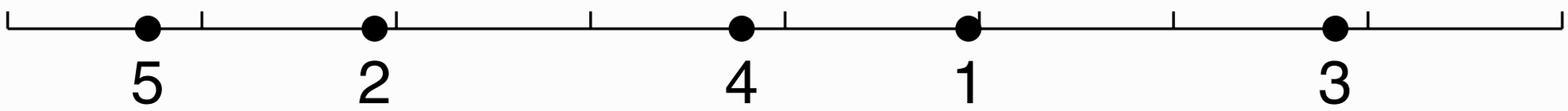
=

```
// Implementation with bitwise operations, Kollig & Keller 2002
double RI_vdC(uint bits, uint s = 0) {
   bits = (bits << 16) | (bits >> 16);
   bits = ((bits & 0x00ff00ff) << 8) | ((bits & 0xff00ff00) >> 8);
   bits = ((bits & 0x0f0f0f0f) << 4) | ((bits & 0xf0f0f0f0) >> 4);
   bits = ((bits & 0x33333333) << 2) | ((bits & 0xcccccccc) >> 2);
   bits = ((bits & 0x55555555) << 1) | ((bits & 0xaaaaaaaa) >> 1);

   // facultative scrambling for randomization
   bits ^= s;

   // final normalization in the range [0,1)
   return (double) bits / (double) 0x100000000LL;
}
```

GOLDEN RATIO SEQ.

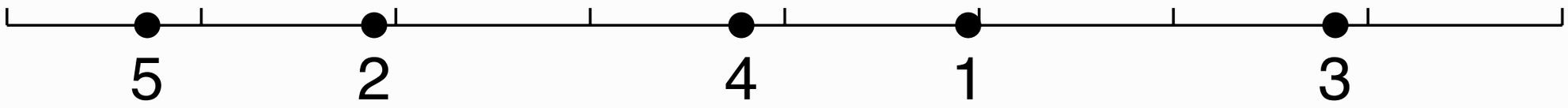


van der Corput sequence



Golden ratio sequence

GOLDEN RATIO SEQ.



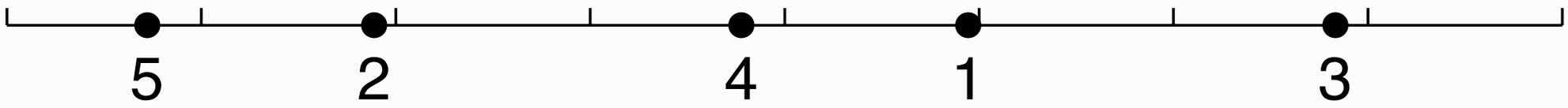
Golden ratio sequence

$$G_s(i) = \{s + i \cdot \phi\}, \quad \forall\, i \geq 1$$

$$\{x\} = x - \lfloor x \rfloor$$

$$\phi = \frac{1+\sqrt{5}}{2} \approx 1.618034\ldots$$

GOLDEN RATIO SEQ.



Golden ratio sequence

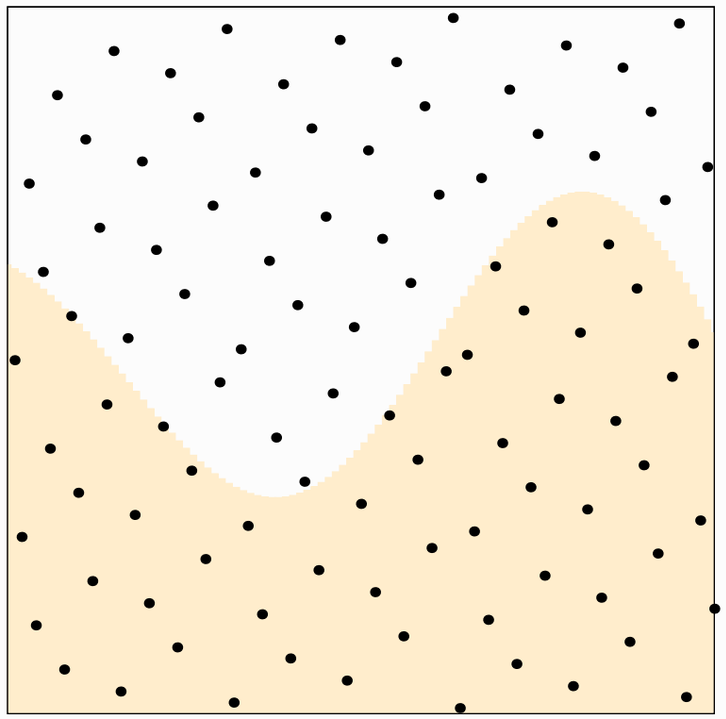
$$G_s(i) = \{s + i \cdot \tau\}, \quad \forall\, i \geq 1$$

$$\{x\} = x - \lfloor x \rfloor$$

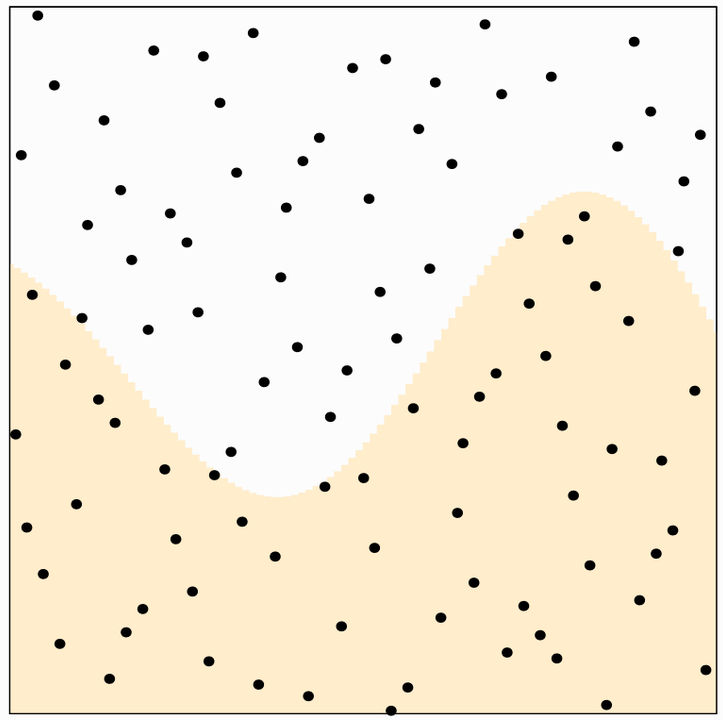
$$\tau = \frac{1}{\phi} = \phi - 1 = \frac{\sqrt{5} - 1}{2} \approx 0.618034\ldots$$

IN **TWO** DIMENSIONS

HAMMERSLEY AND HALTON



Hammersley set

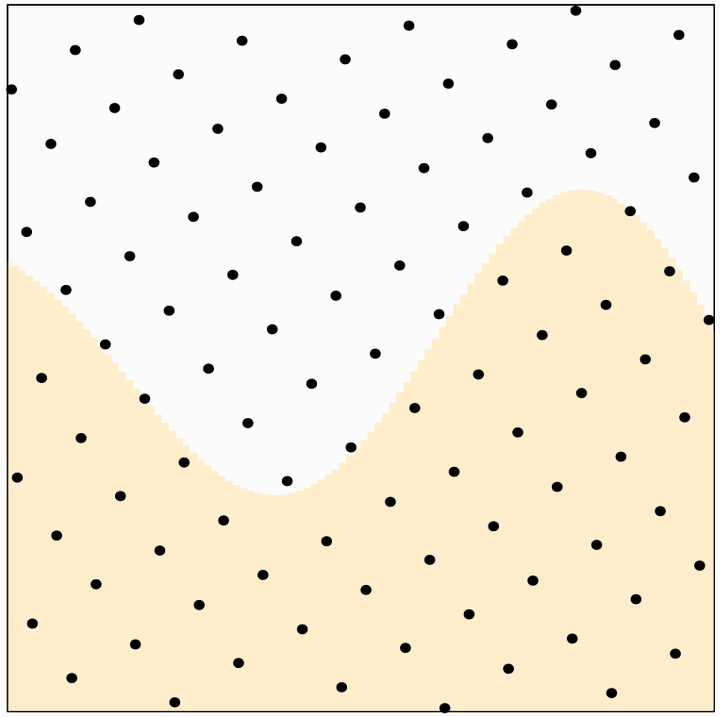


Halton sequence

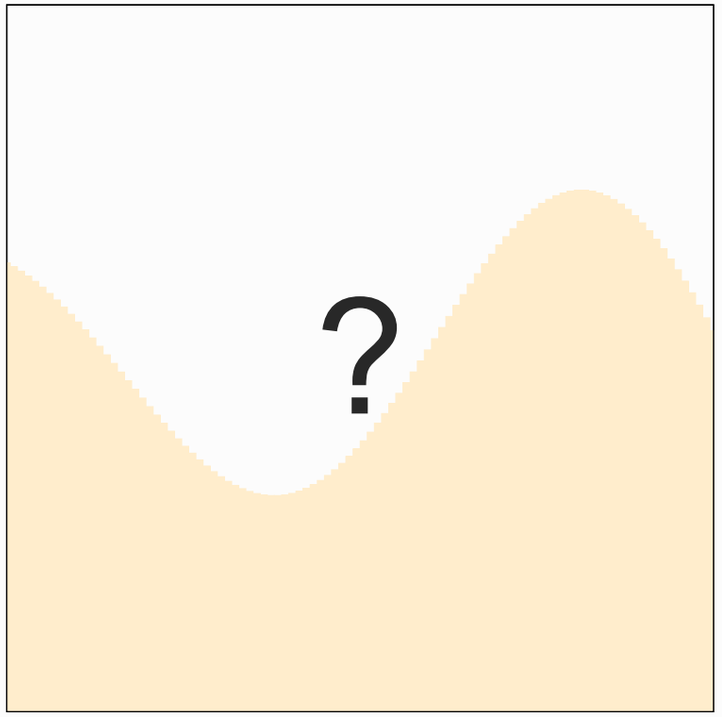
$$\left\{\left(\frac{i}{N}, H_2\left(i\right)\right)\right\}_{i=1}^{N}$$

$$\left(H_2\left(i\right), H_3\left(i\right)\right), \quad \forall\, i \geq 1$$

GOLDEN LATTICE RULE



Golden lattice

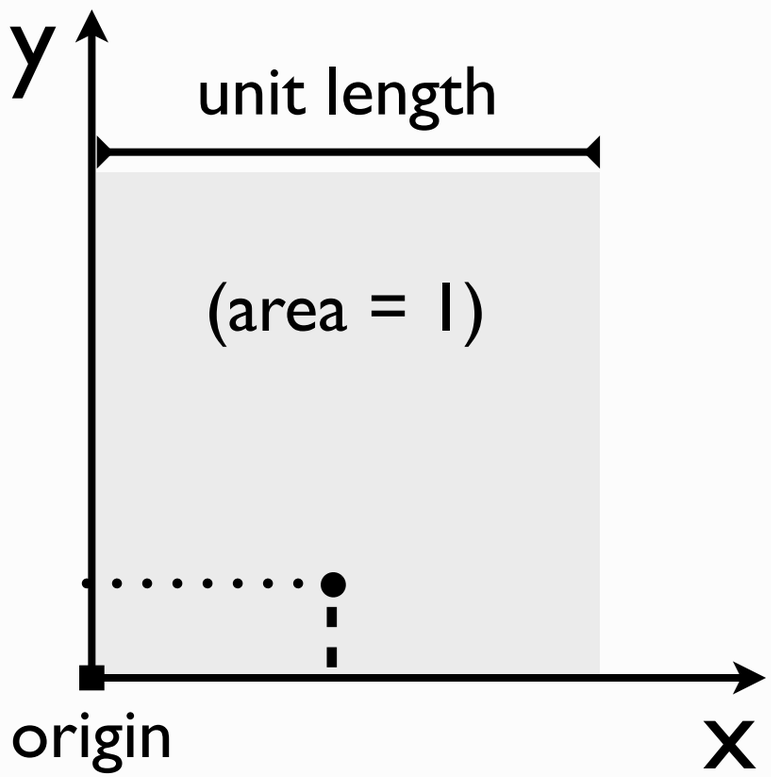


Golden sequence

$$\left\{\left(\frac{i}{N}, G_s\left(i\right)\right)\right\}_{i=1}^{N}$$

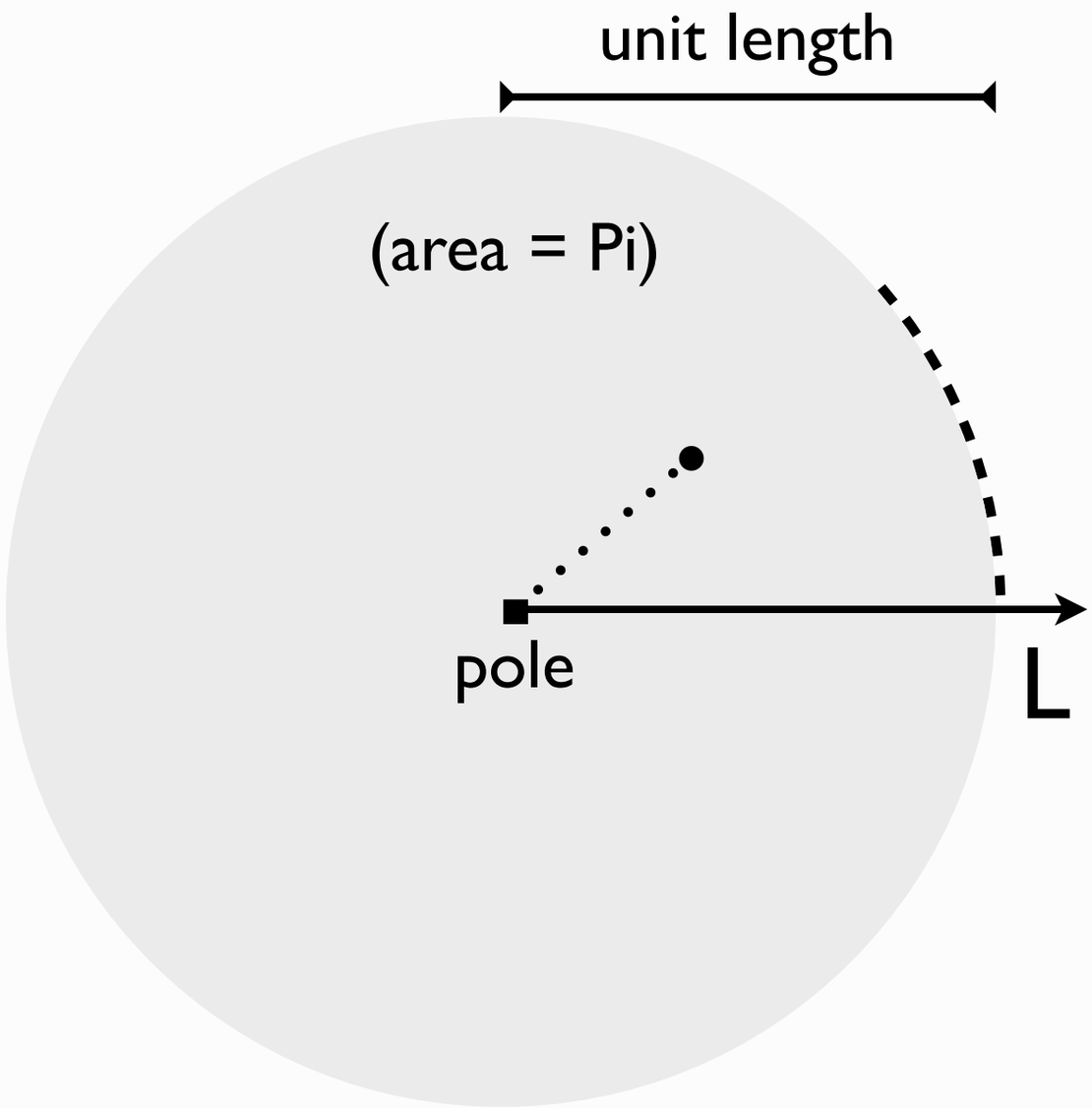
$$\left(G_s\left(i\right), ?\right), \quad \forall\, i \geq 1$$

CARTESIAN/POLAR SYSTEMS



vertical coordinate · · · · · · · ·
horizontal coordinate - -

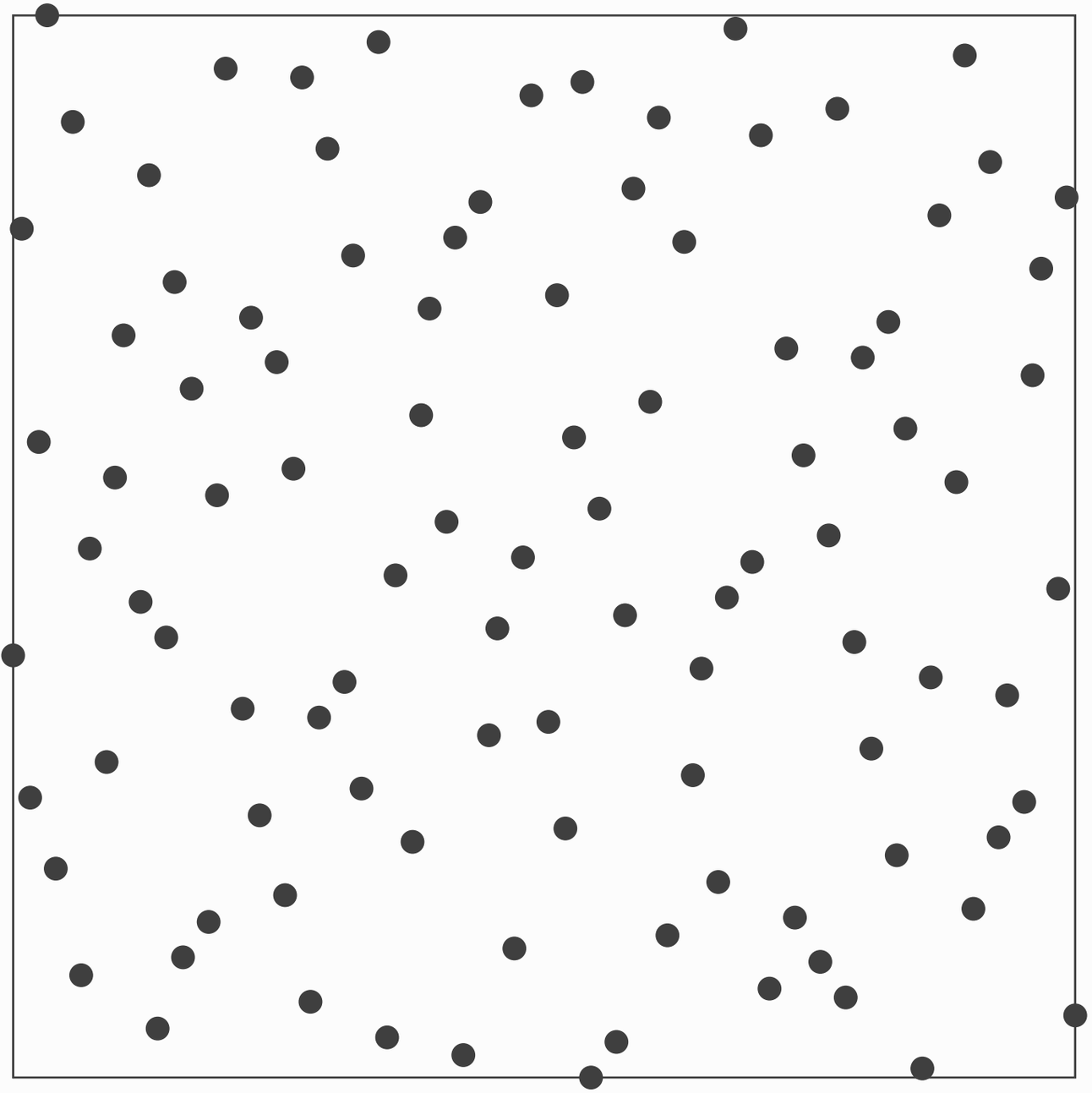
Cartesian coordinate system



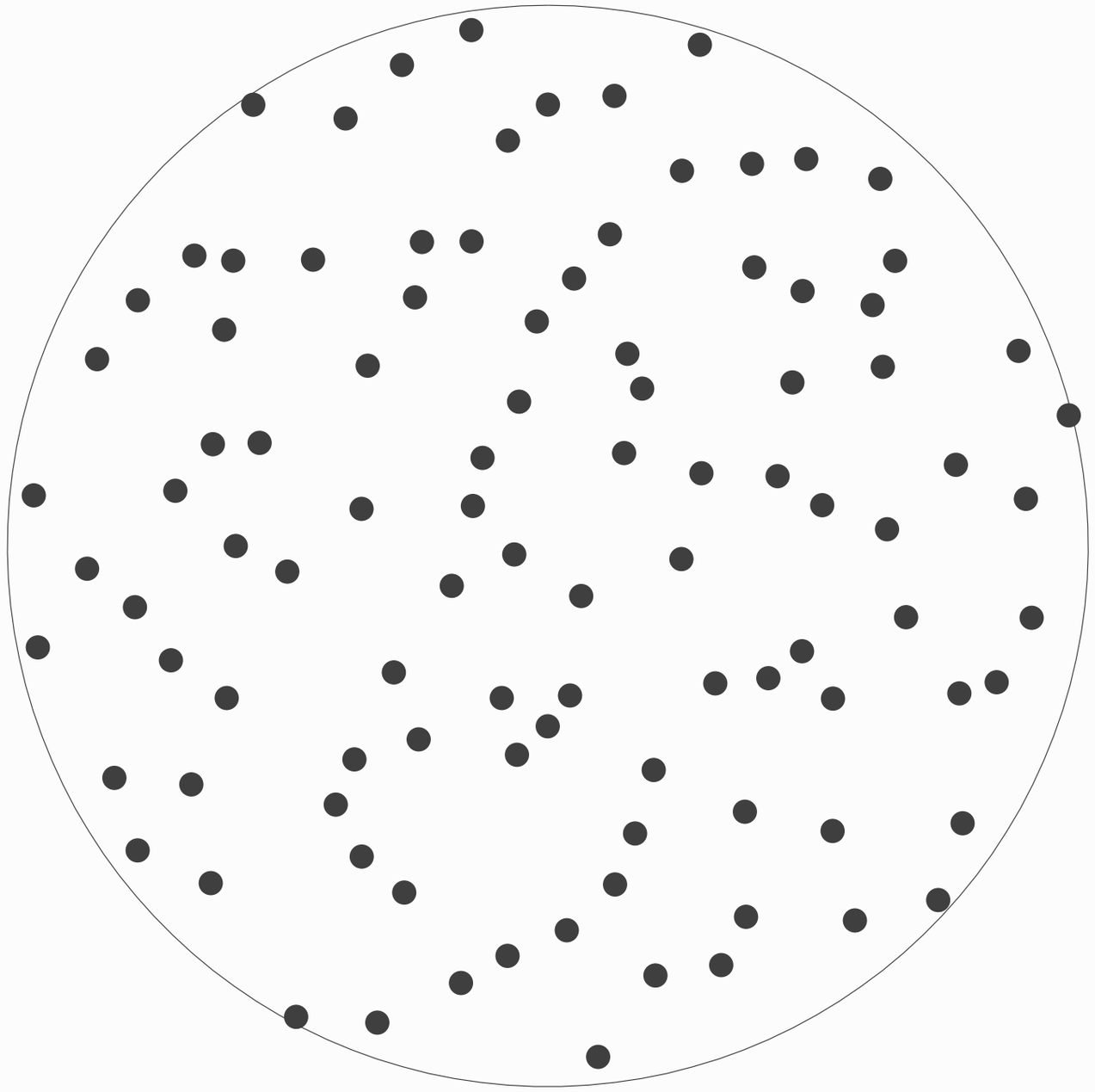
radius · · · · · · · ·
polar angle - - - - - - - - (2xPi)

Polar coordinate system

HALTON SEQUENCE

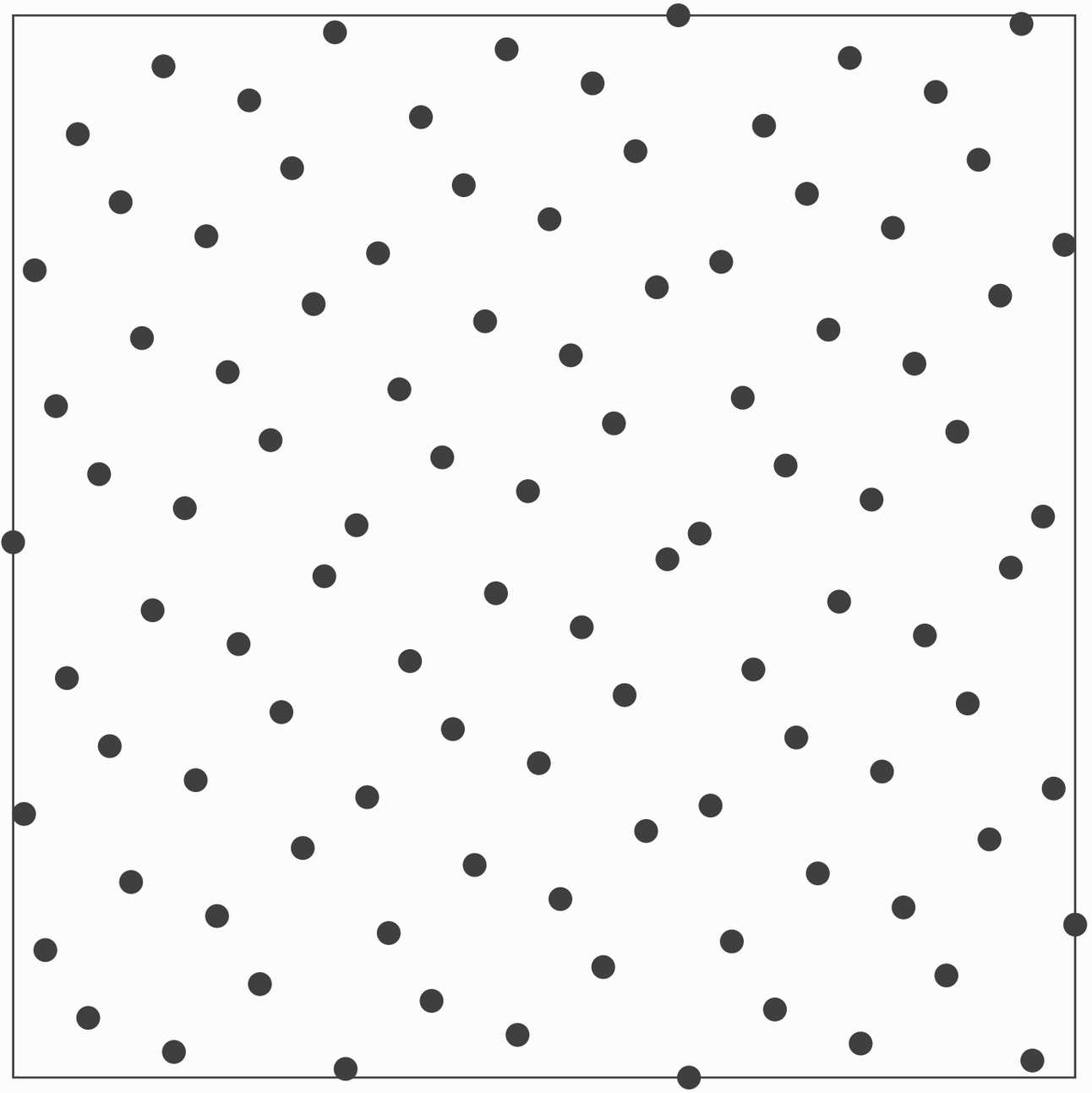


Cartesian coordinates

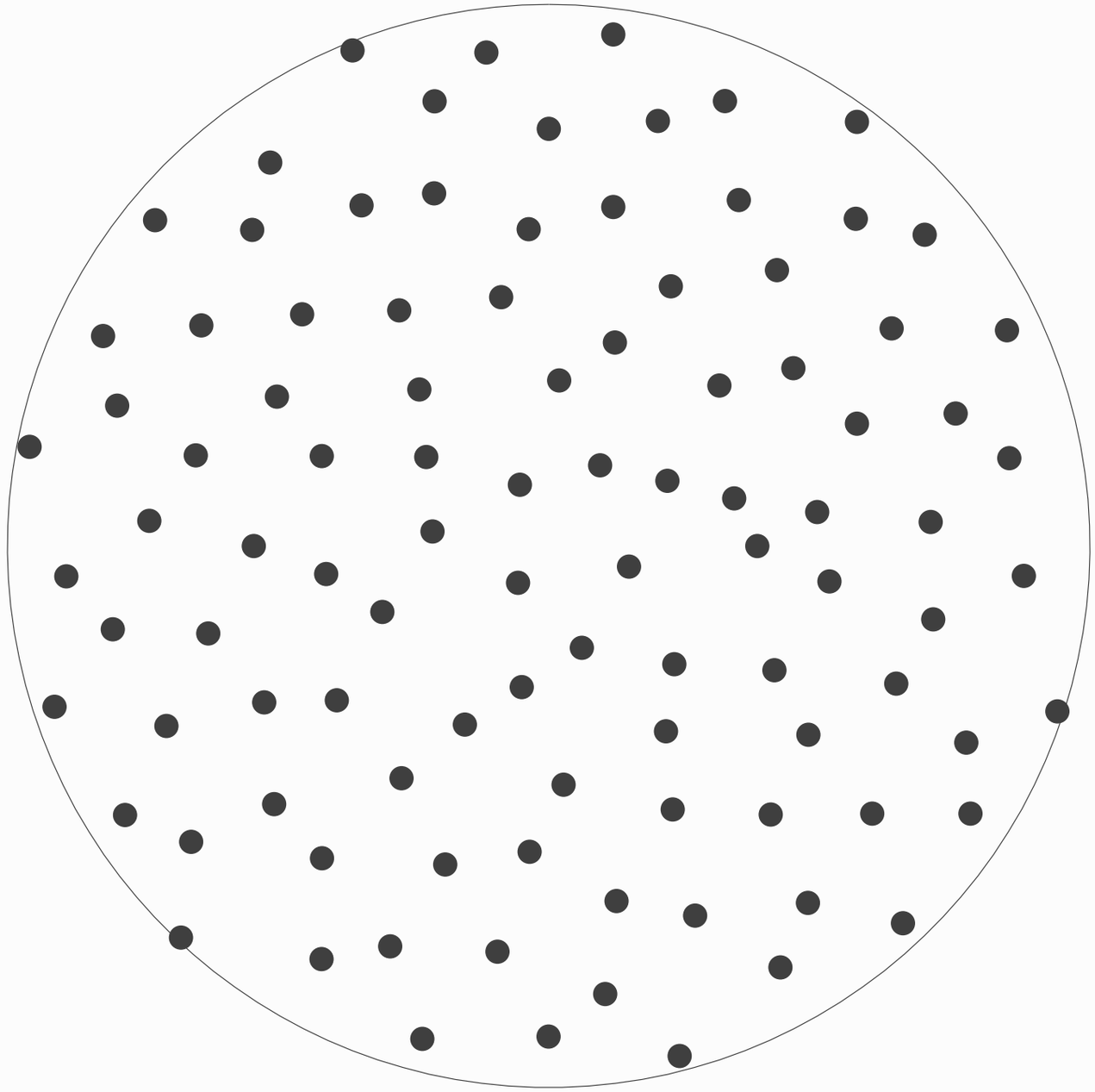


Polar coordinates

HAMMERSLEY SET

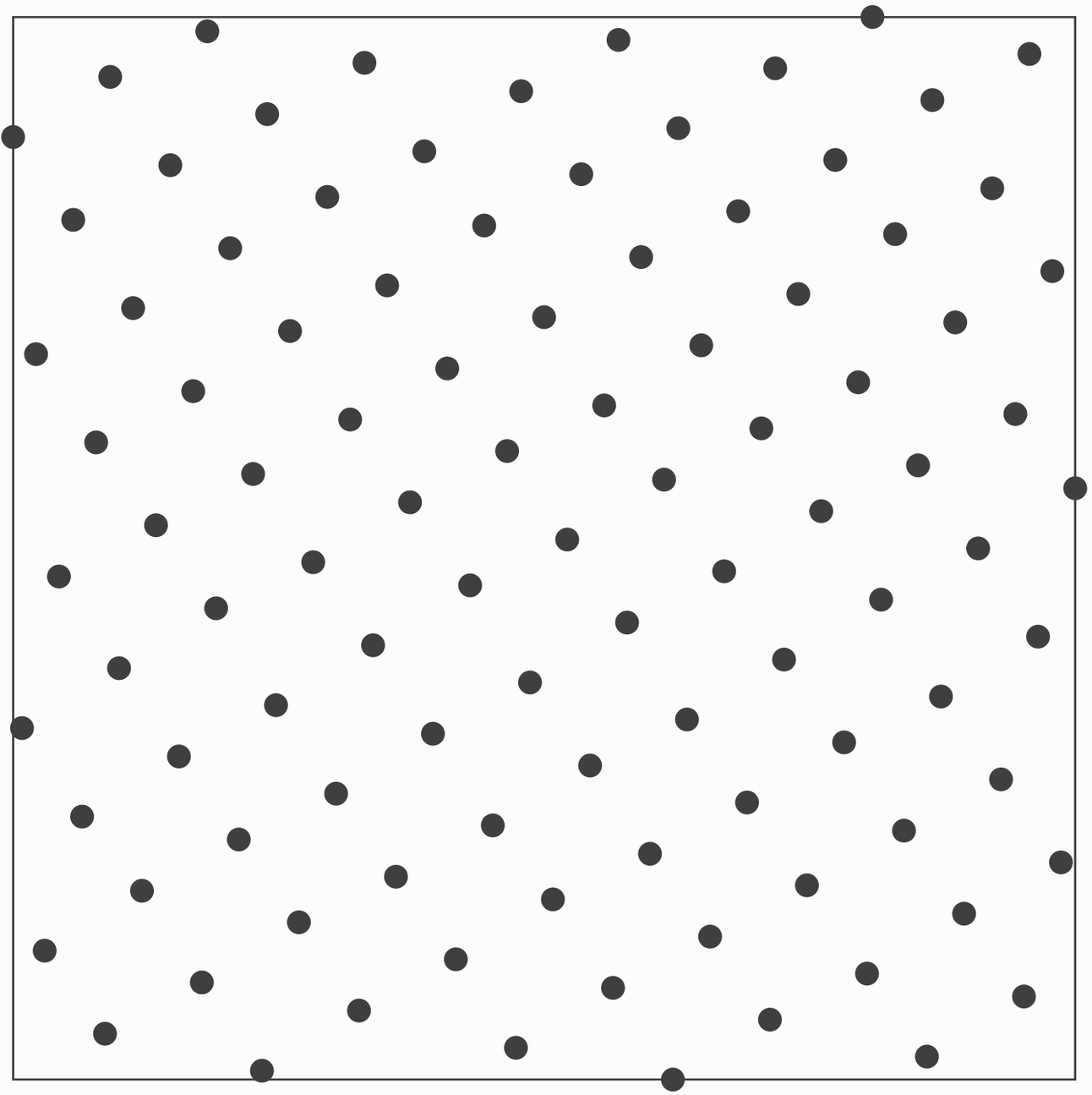


Cartesian coordinates

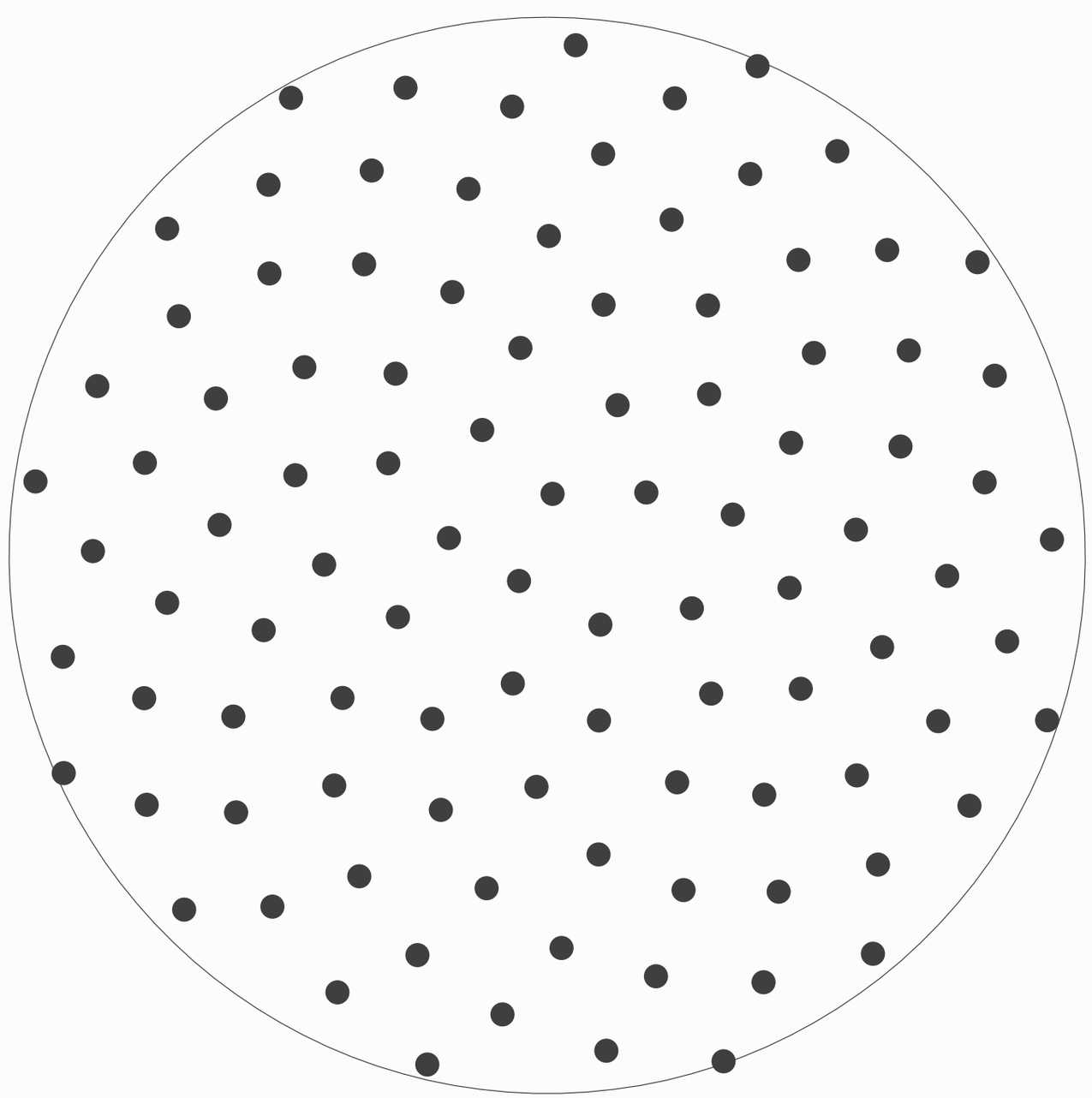


Polar coordinates

GOLDEN LATTICE



Cartesian coordinates



Polar coordinates

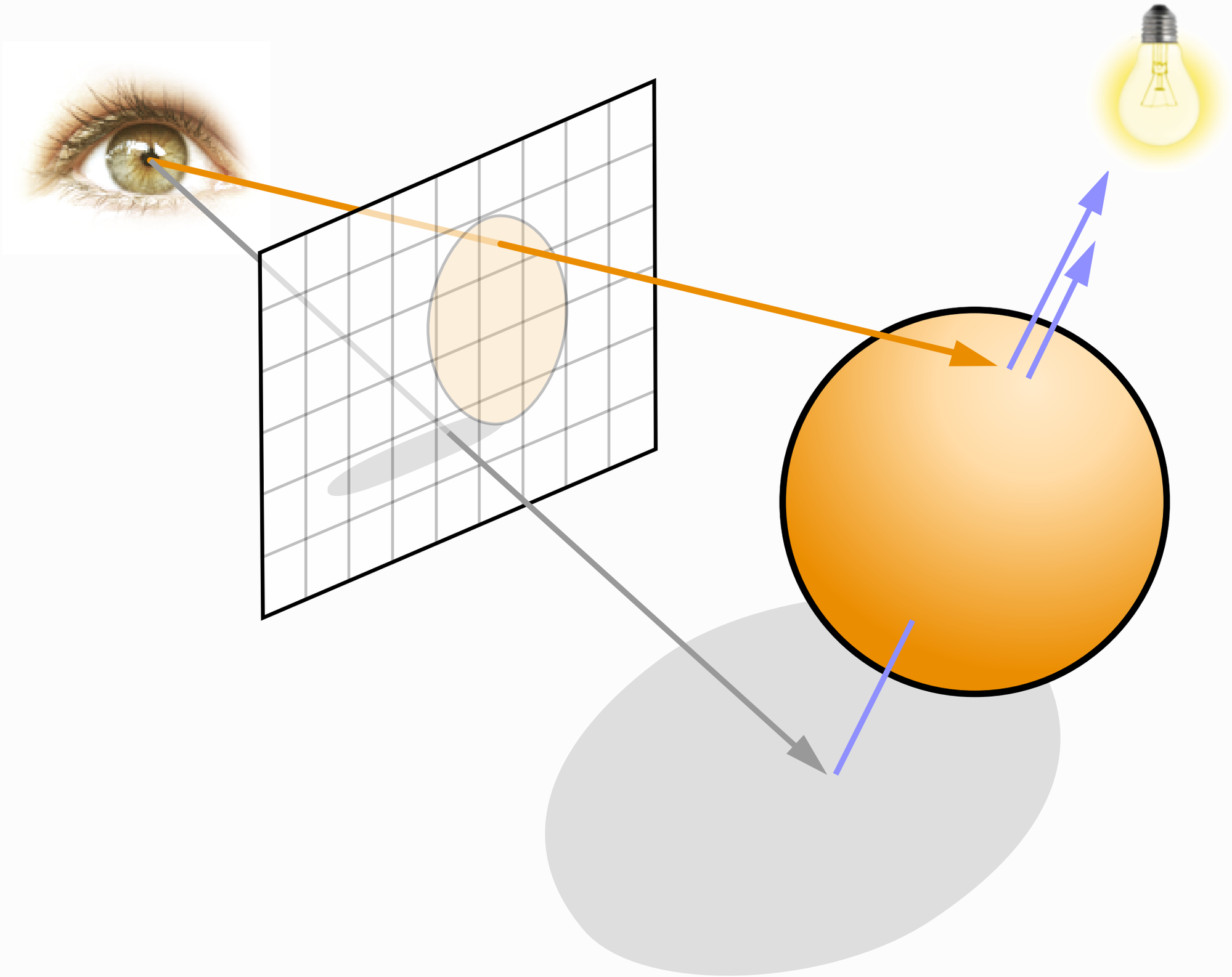
PHYLLOTAXIS INSPIRATIONS

Strawberry seeds

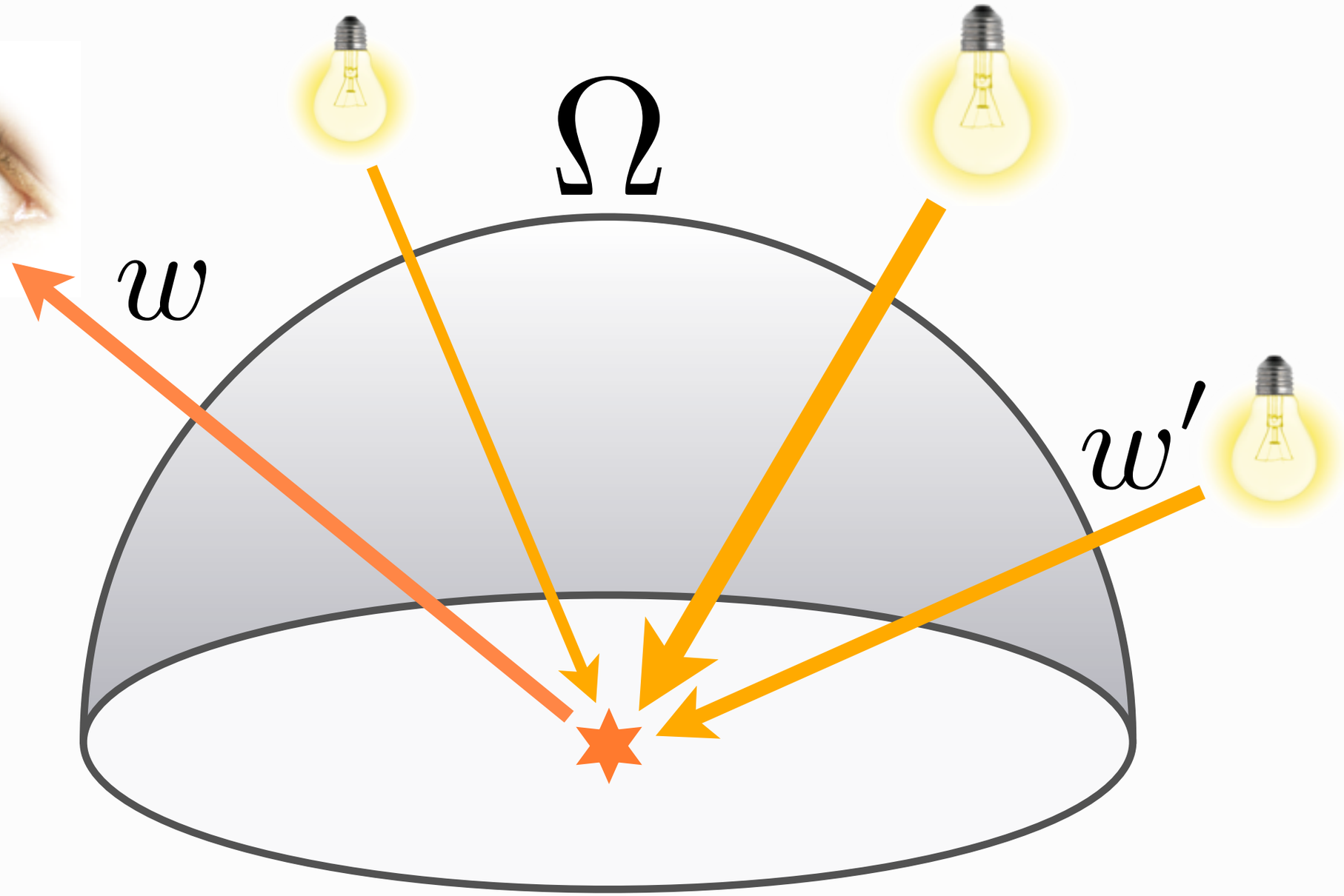
Flower seeds

COMPUTER **RENDERING**

RAY-TRACING IN 3D SCENES



RENDERING EQUATION



$$L_w\left(x, w\right) = \int_{\Omega} L_i\left(w'\right)\ V\left(x, w'\right)\ \beta_r\left(w, w'\right)\ \cos\left(w, w'\right)\ dw'$$

QUASI-RANDOM SAMPLING

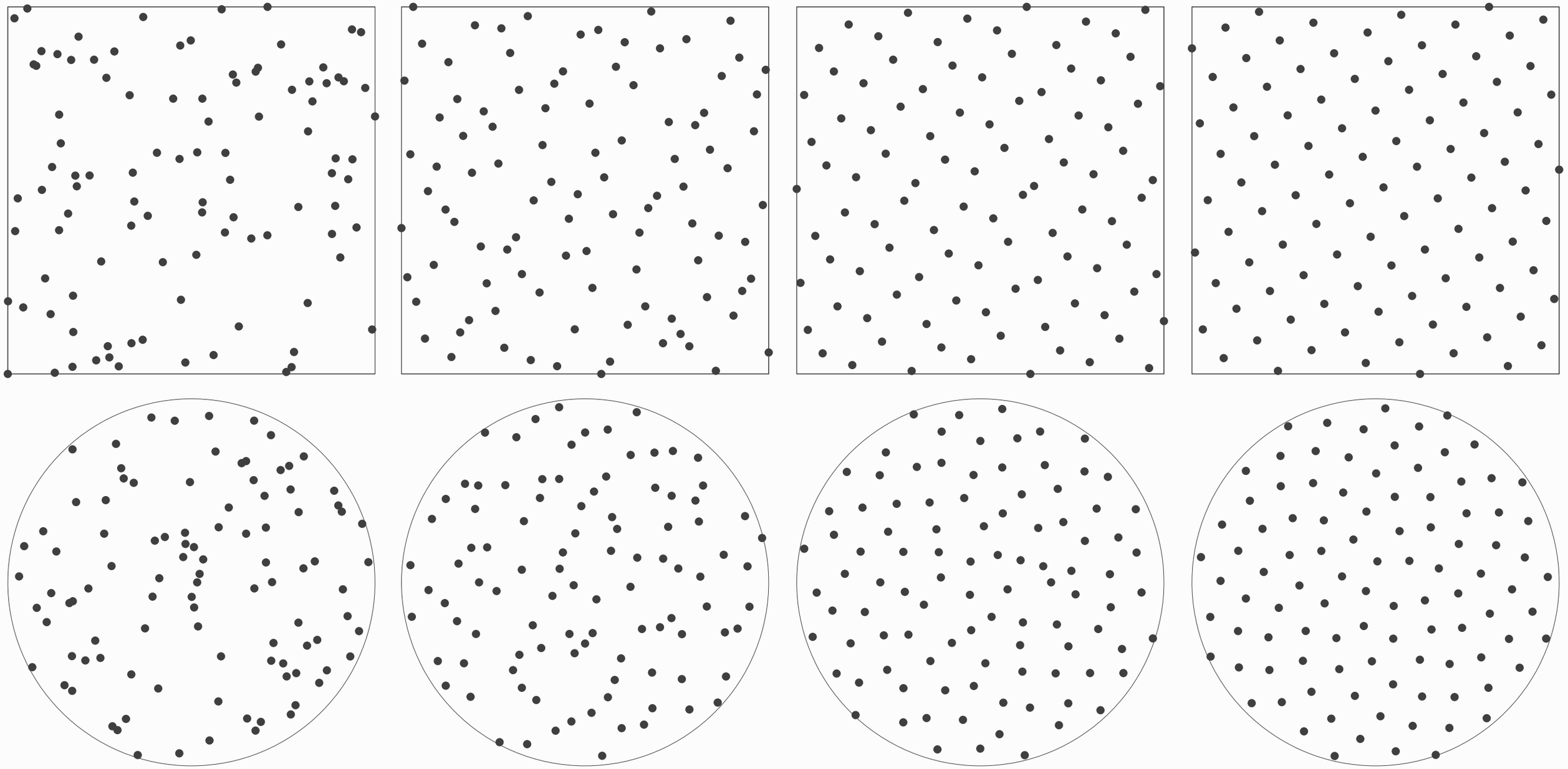
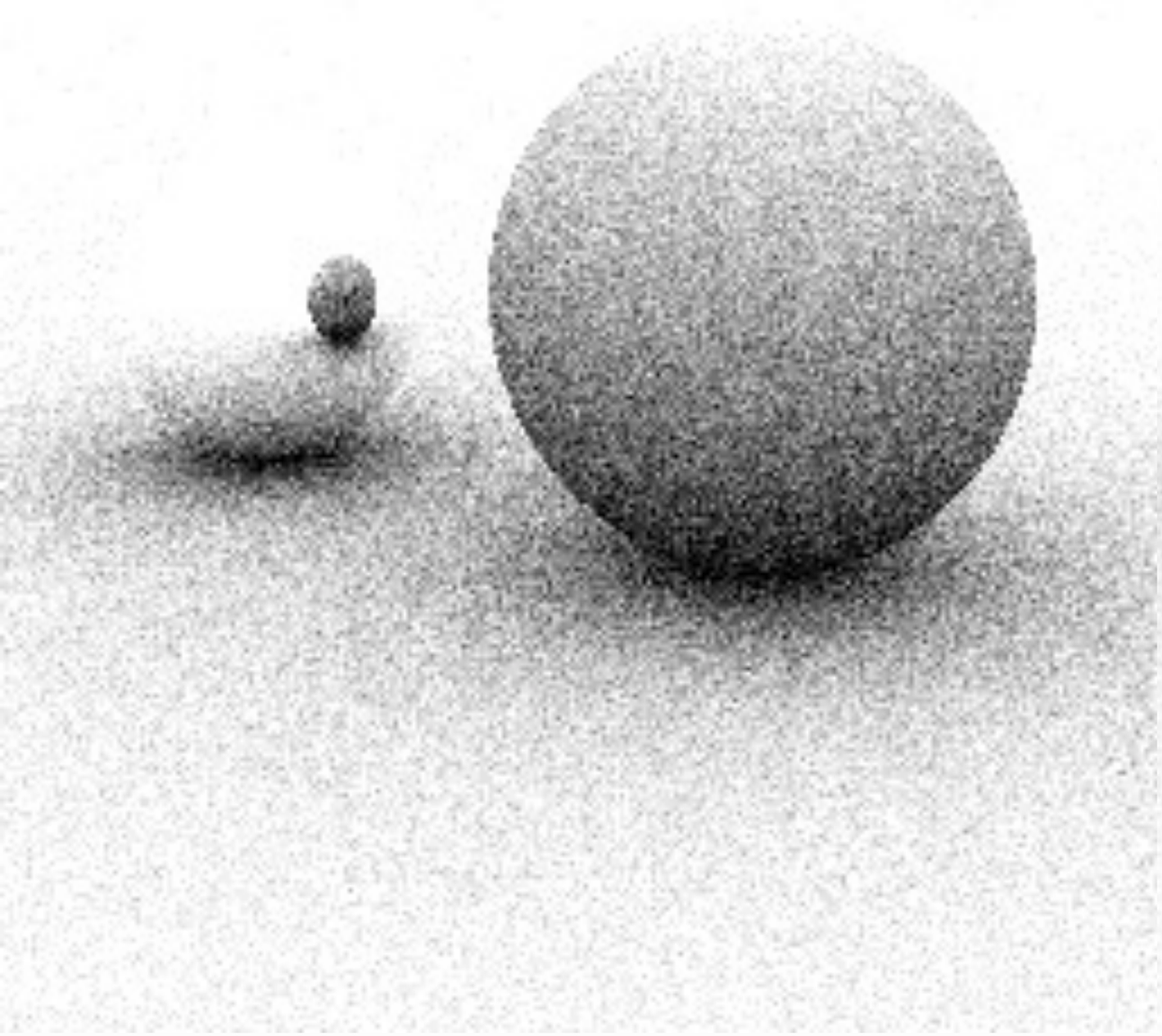
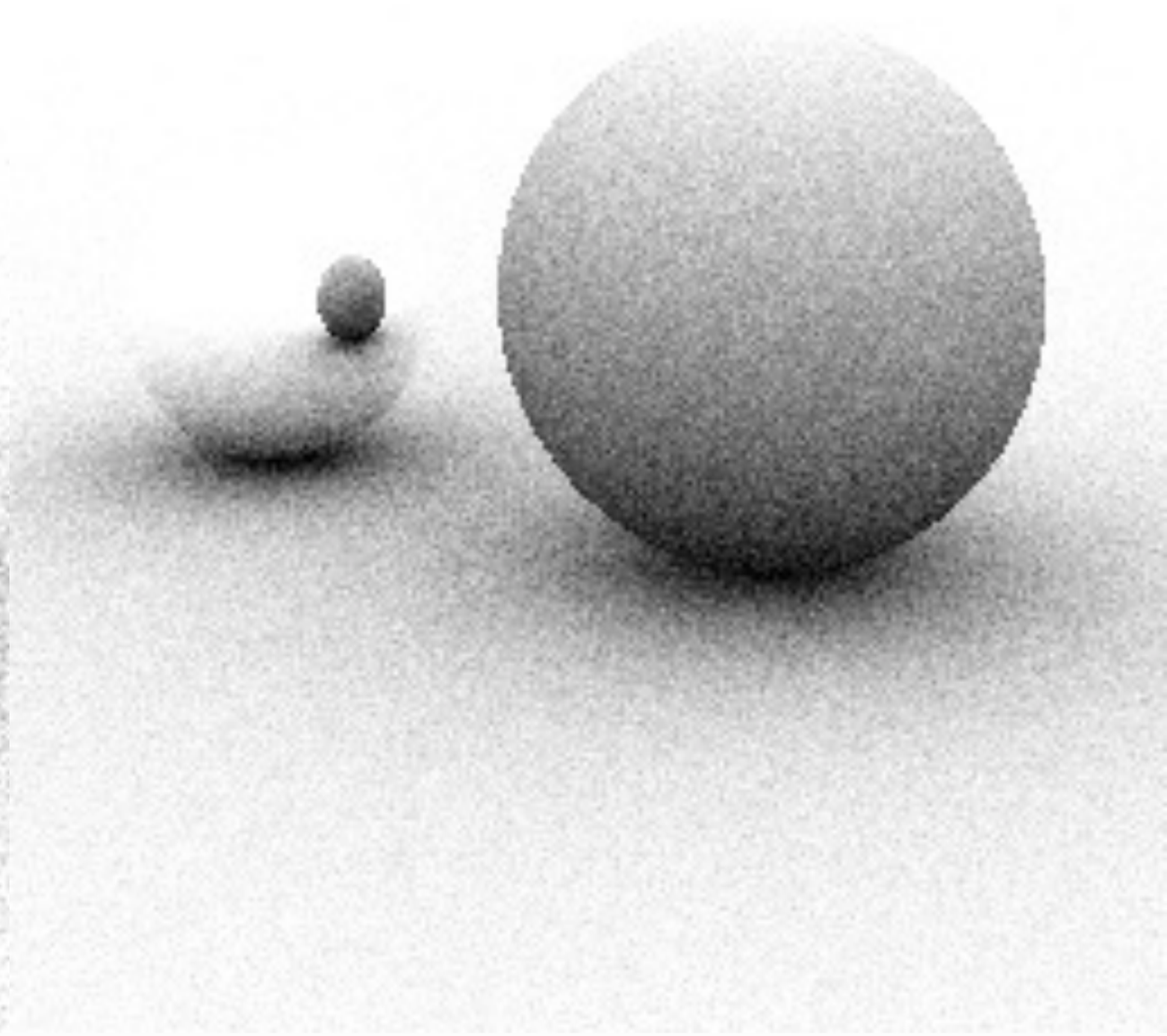


Figure from *Schretter, Dehaye and Kobbelt, 2012*

AMBIENT OCCLUSION

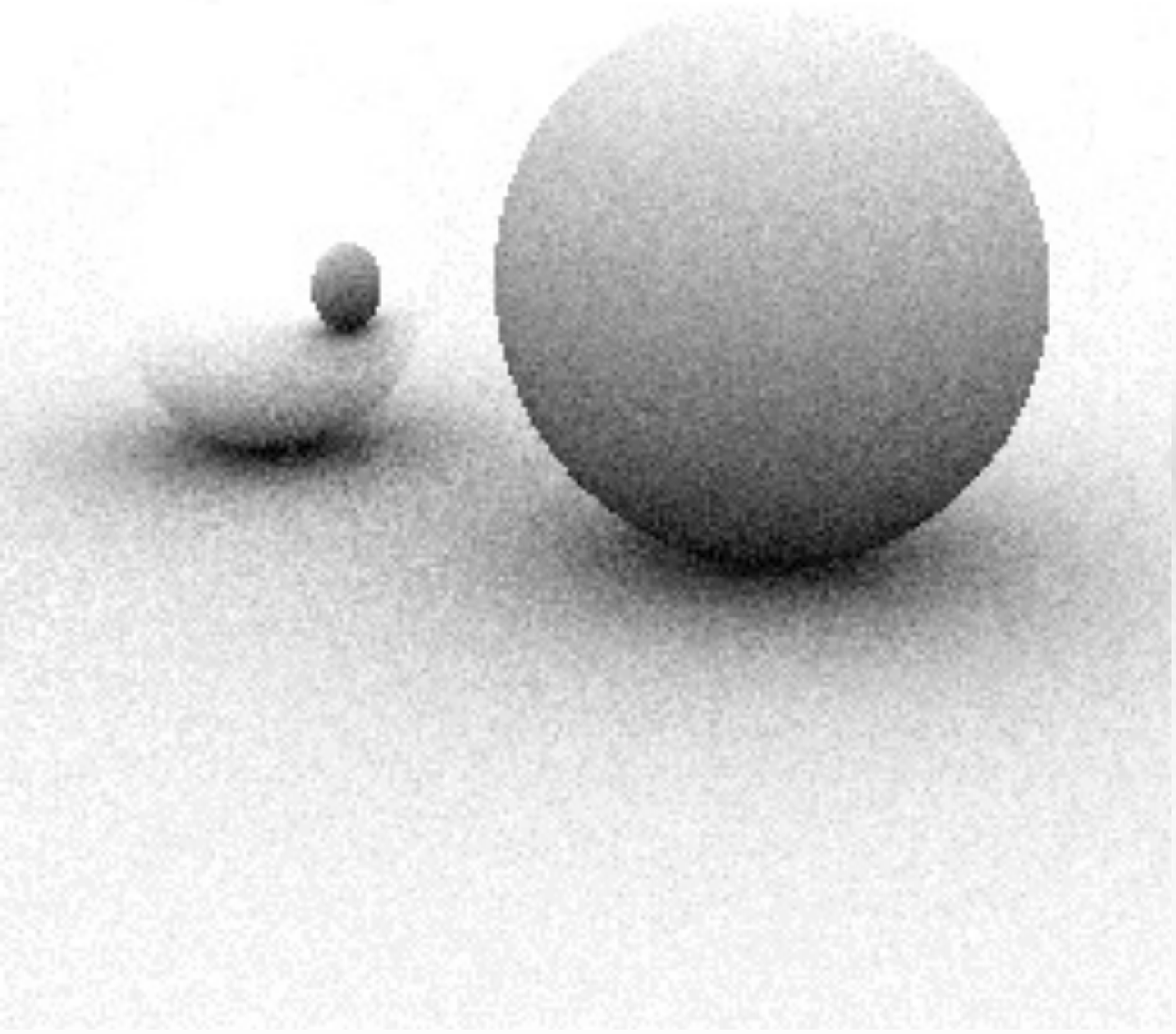


Random (16 spp)

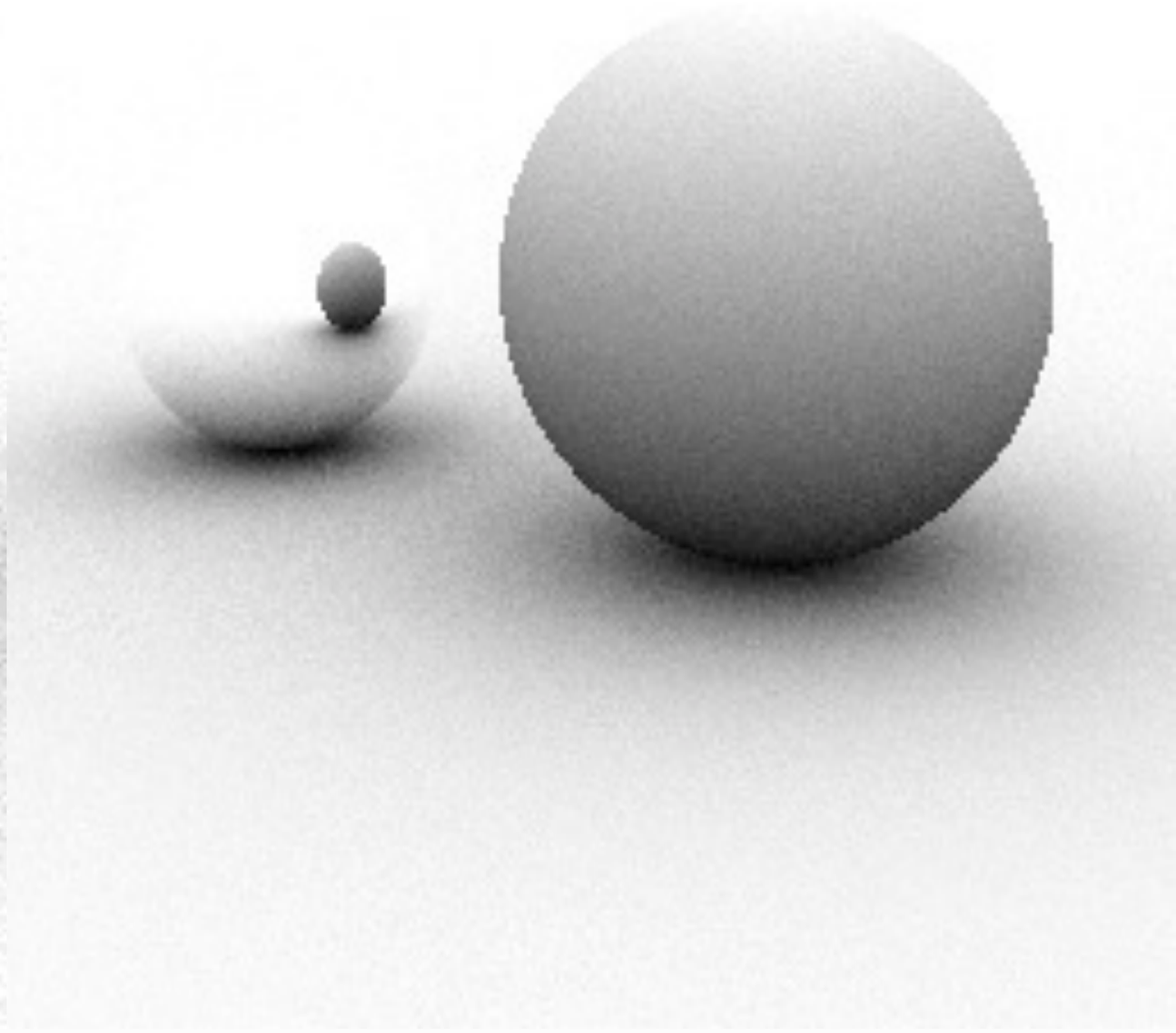


Random (64 spp)

AMBIENT OCCLUSION



Golden lattice (16 spp)

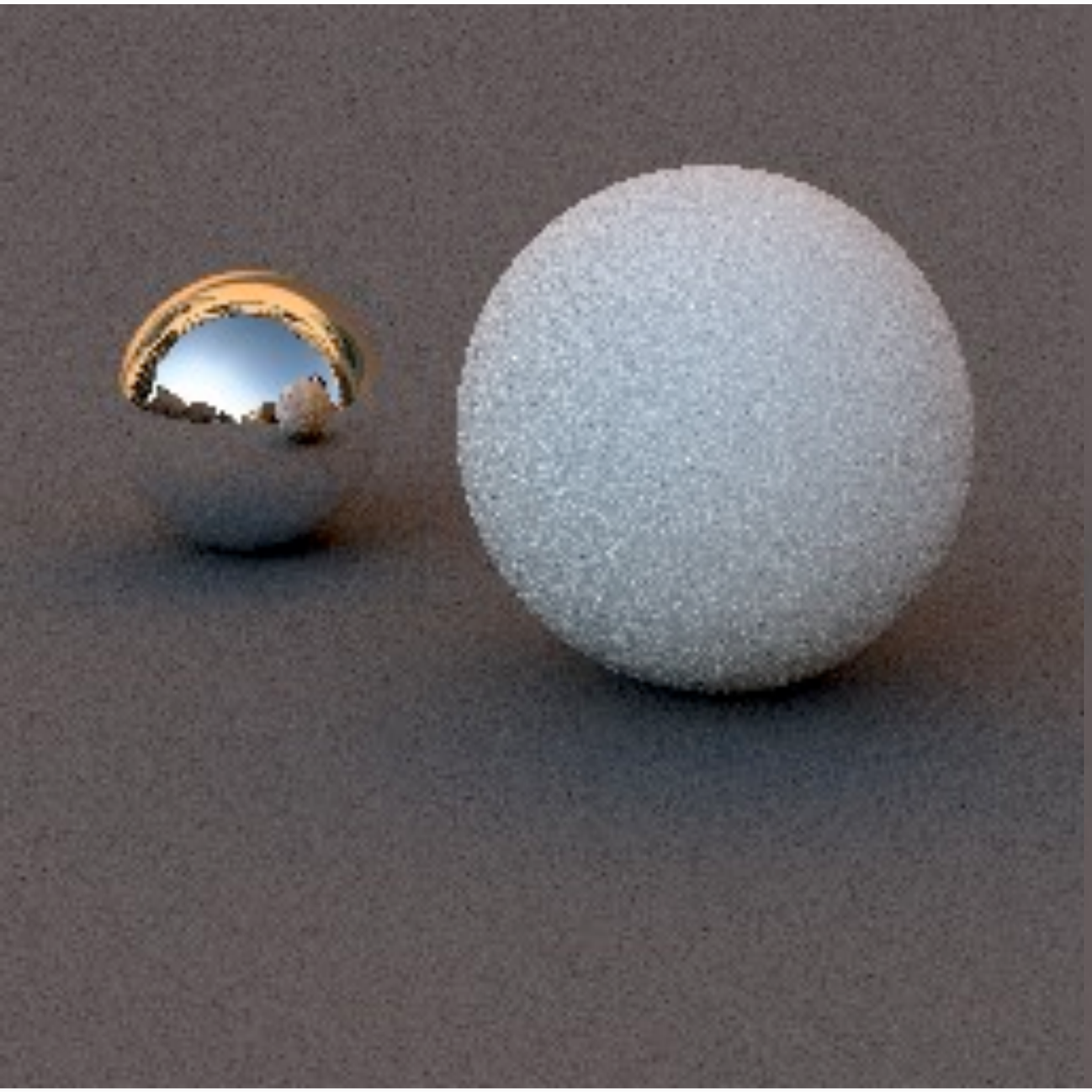


Golden lattice (64 spp)

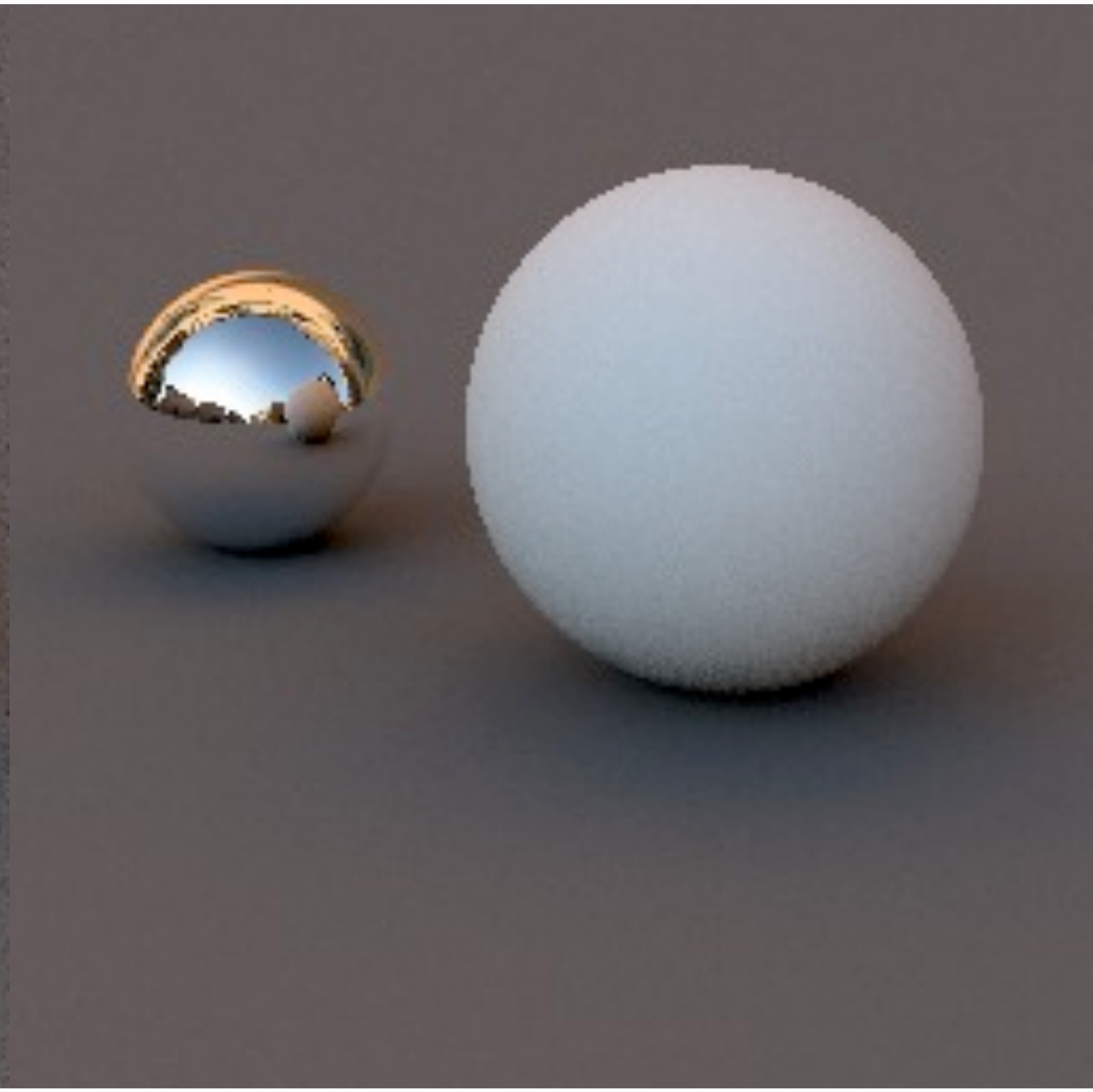
HDR LIGHT PROBE

Pisa light probe, *courtesy of Paul Debevec*

IMAGE-BASED LIGHTING



Golden lattice (64 spp)



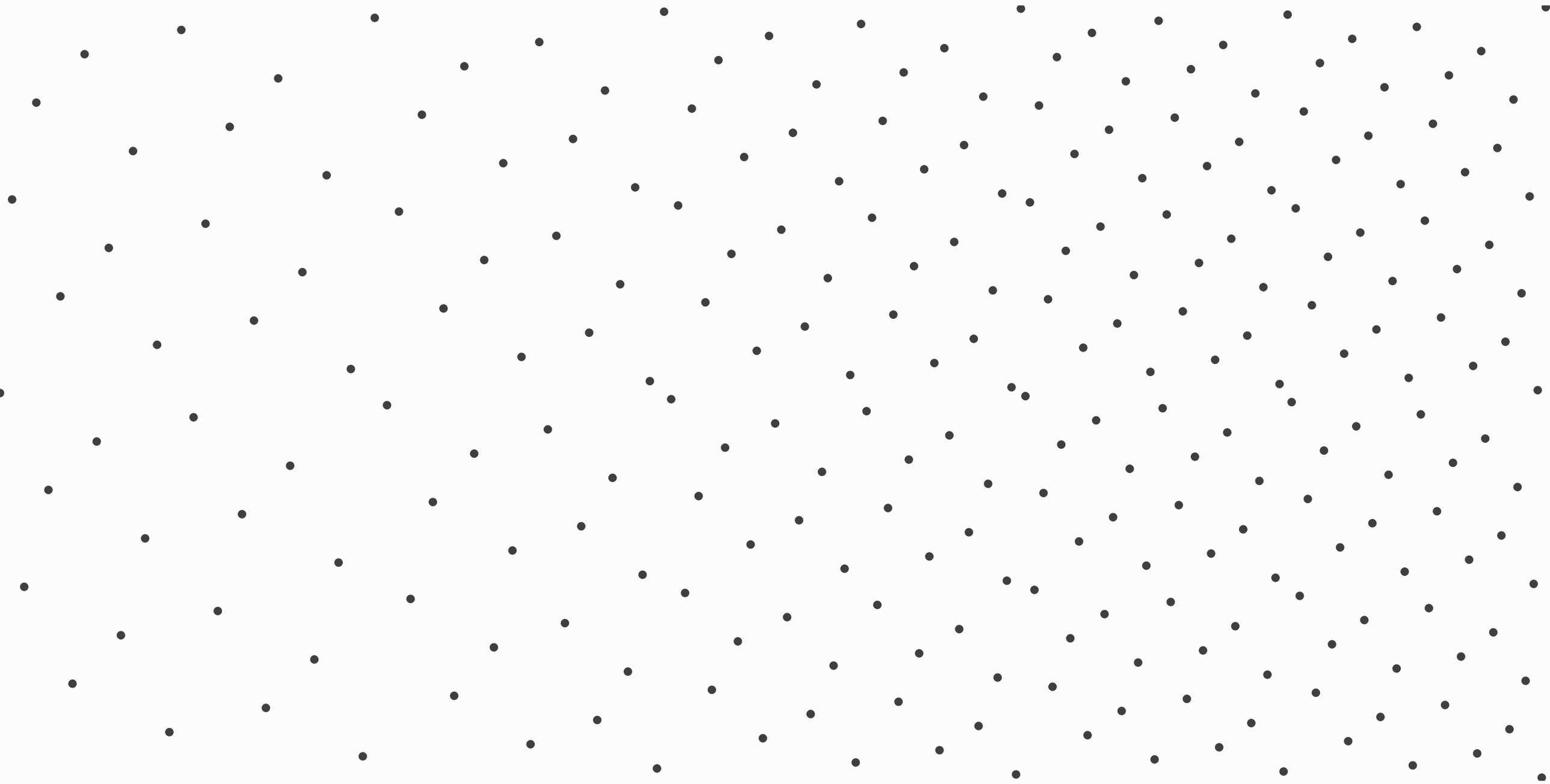
Golden lattice (256 spp)

NON-UNIFORM SAMPLING

PROBABILITY DENSITY

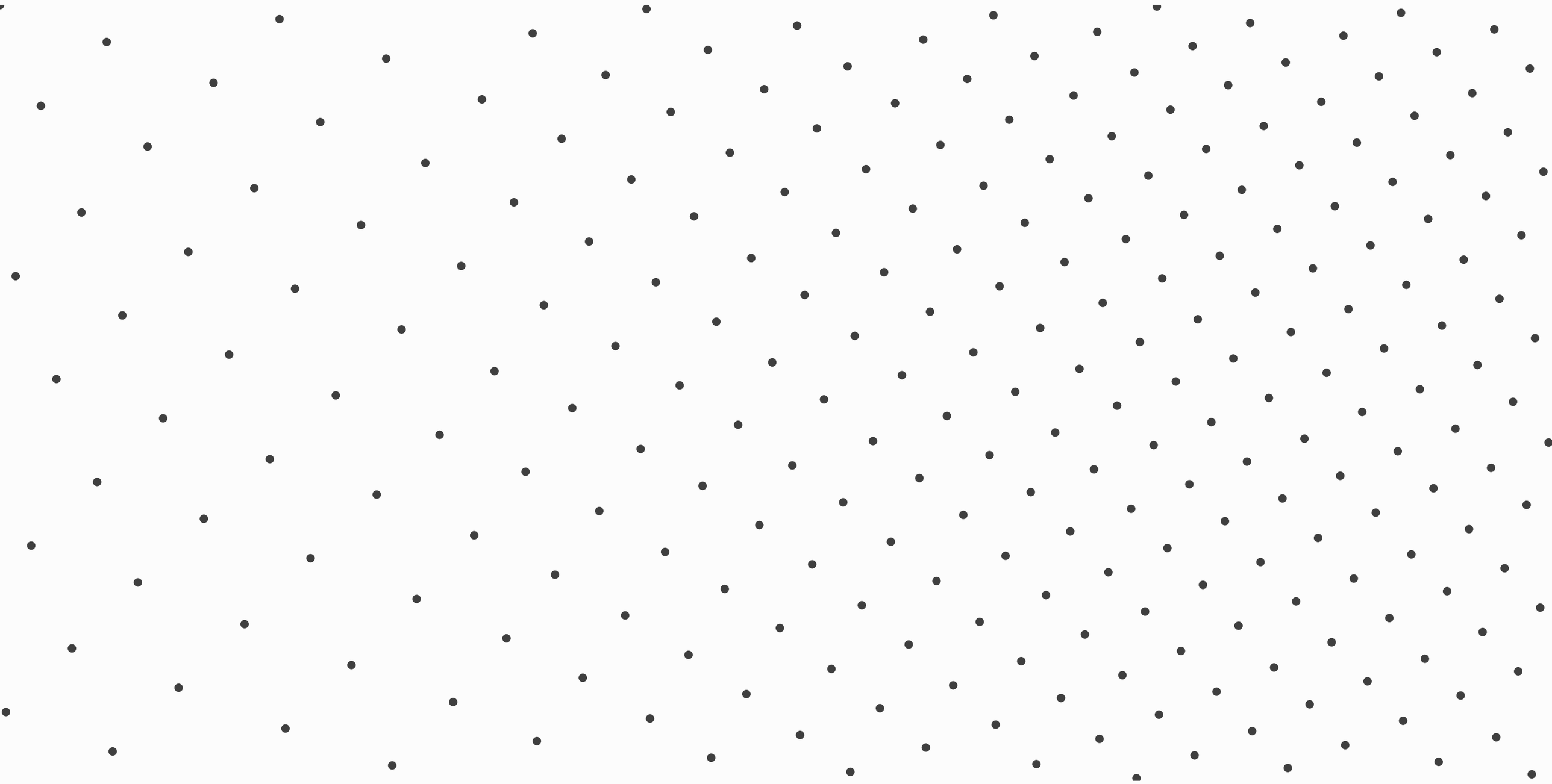
Piecewise-linear ramp gradient

WARPED POINT SET



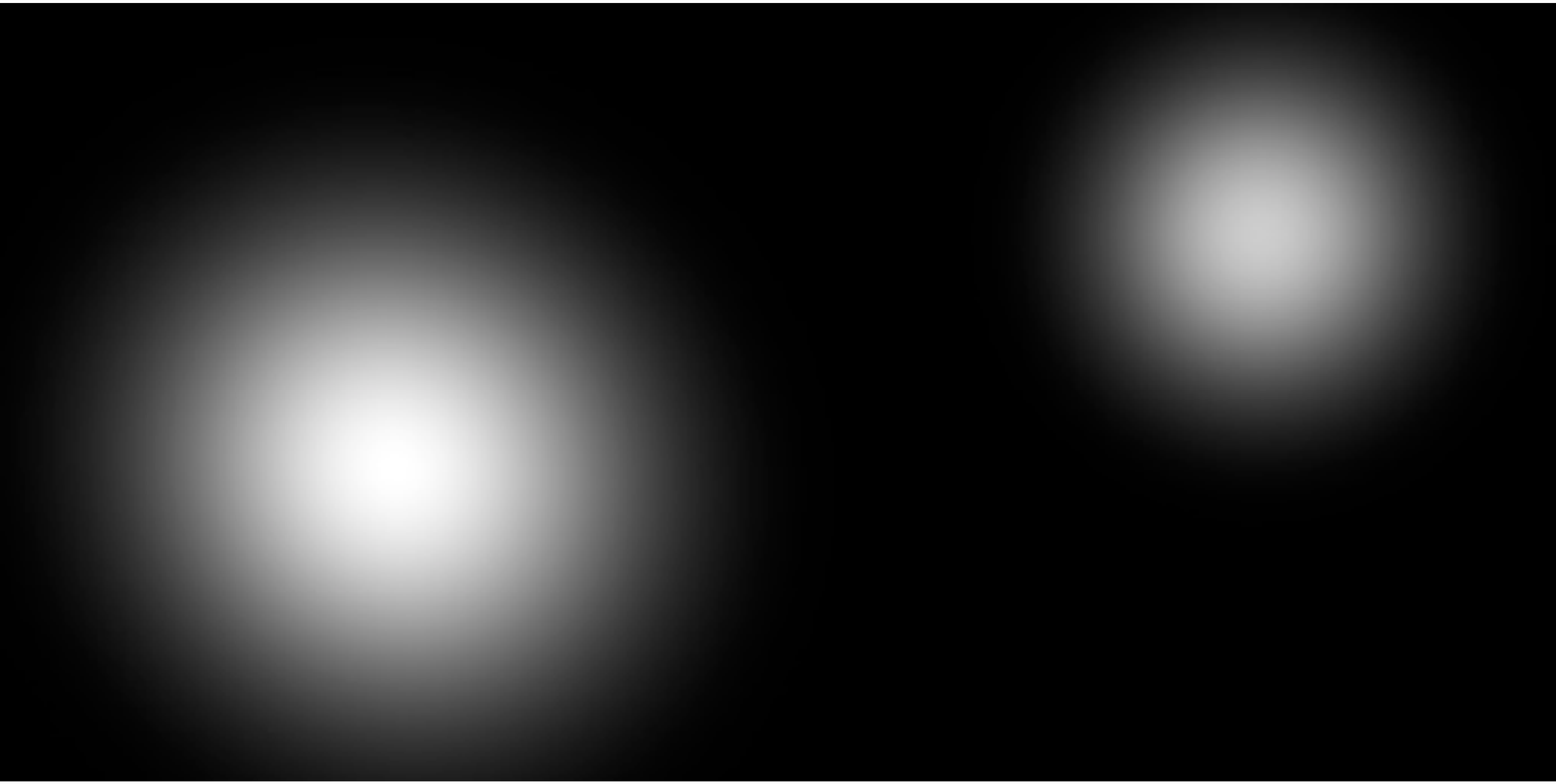
Warped Hammersley set (256 points)

WARPED POINT SET



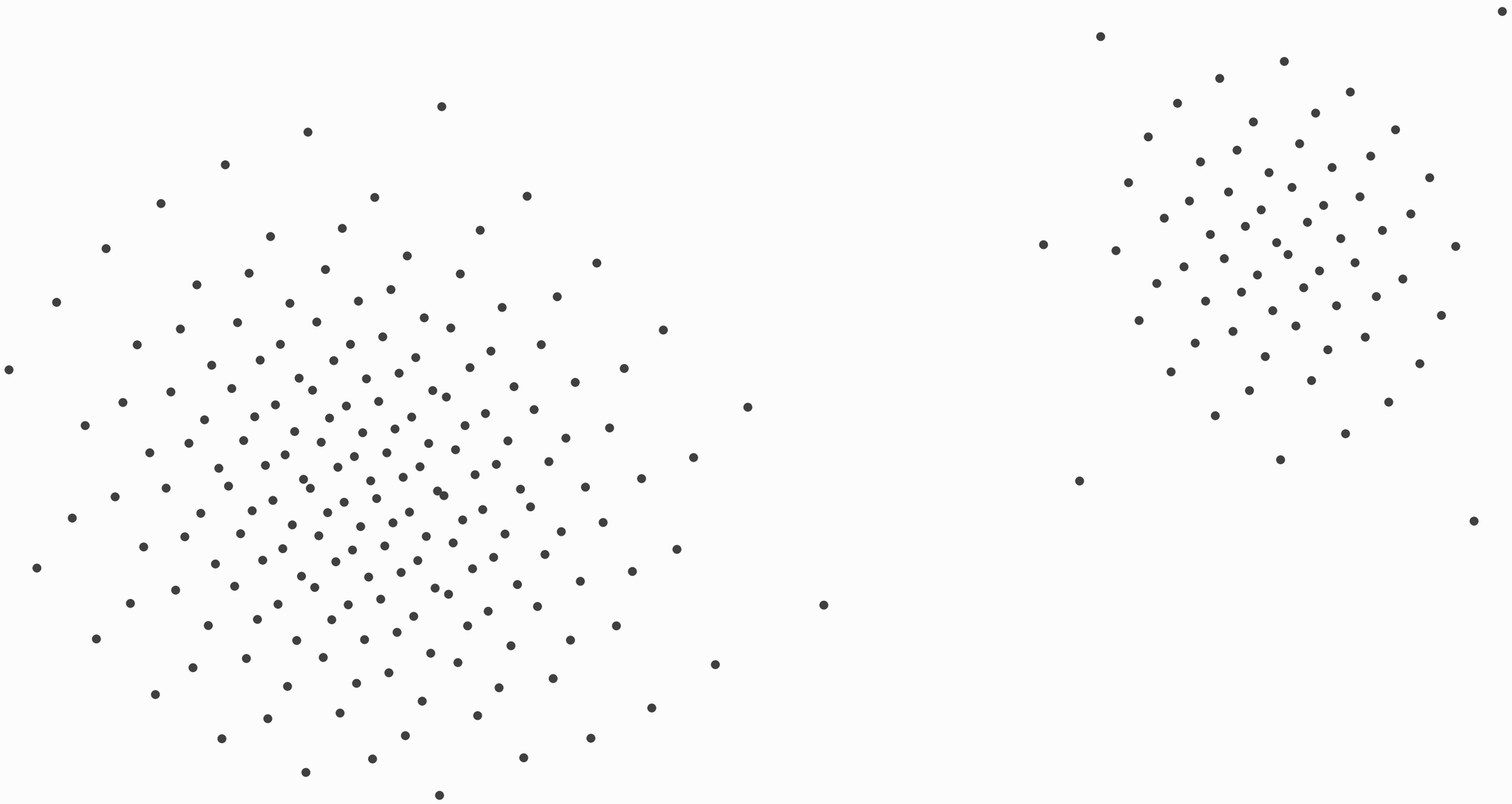
Warped golden lattice (256 points)

PROBABILITY DENSITY



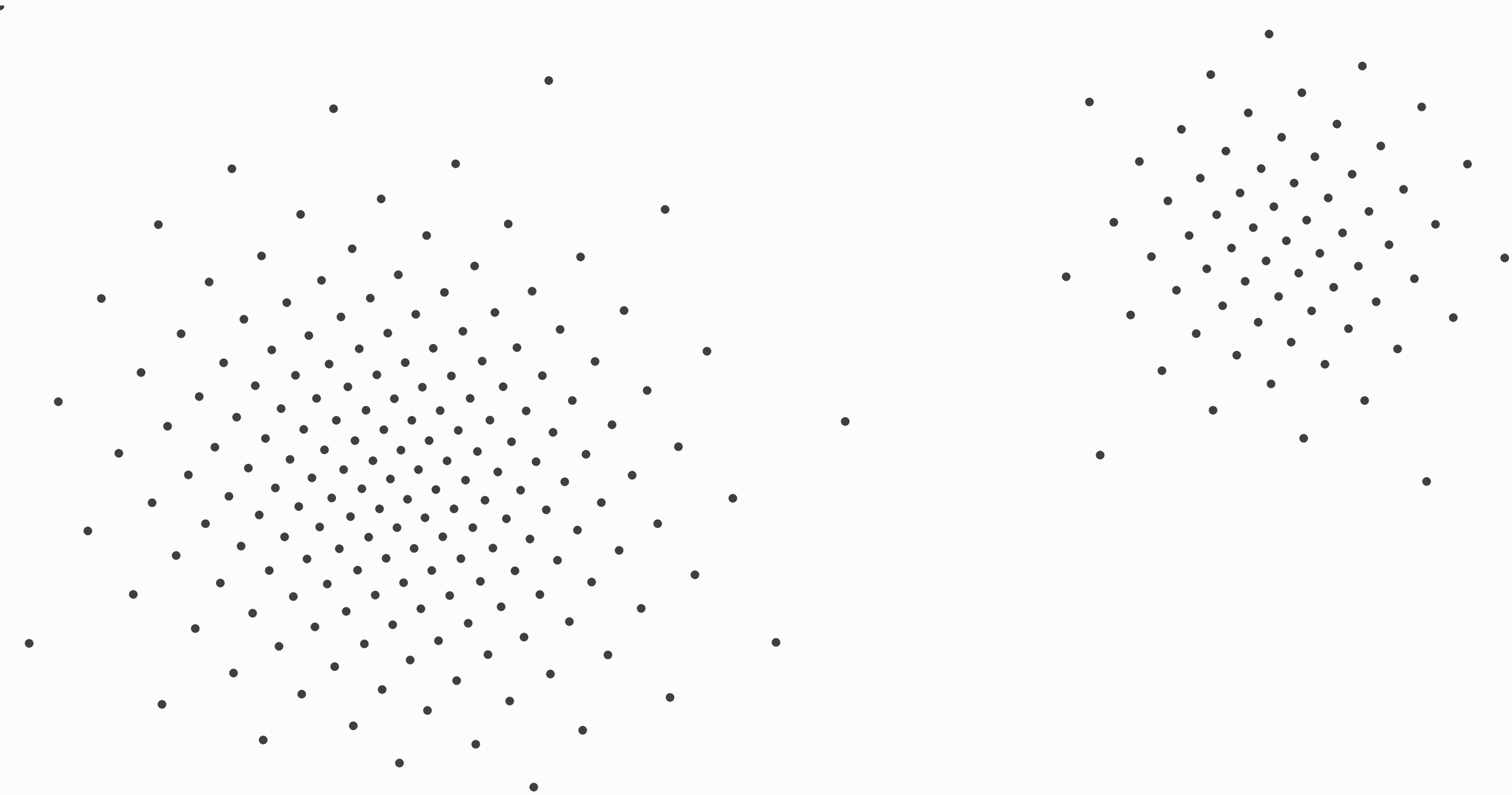
Mixture of two multivariate Gaussians

WARPED POINT SET



Warped Hammersley set (256 points)

WARPED POINT SET



Warped golden lattice (256 points)

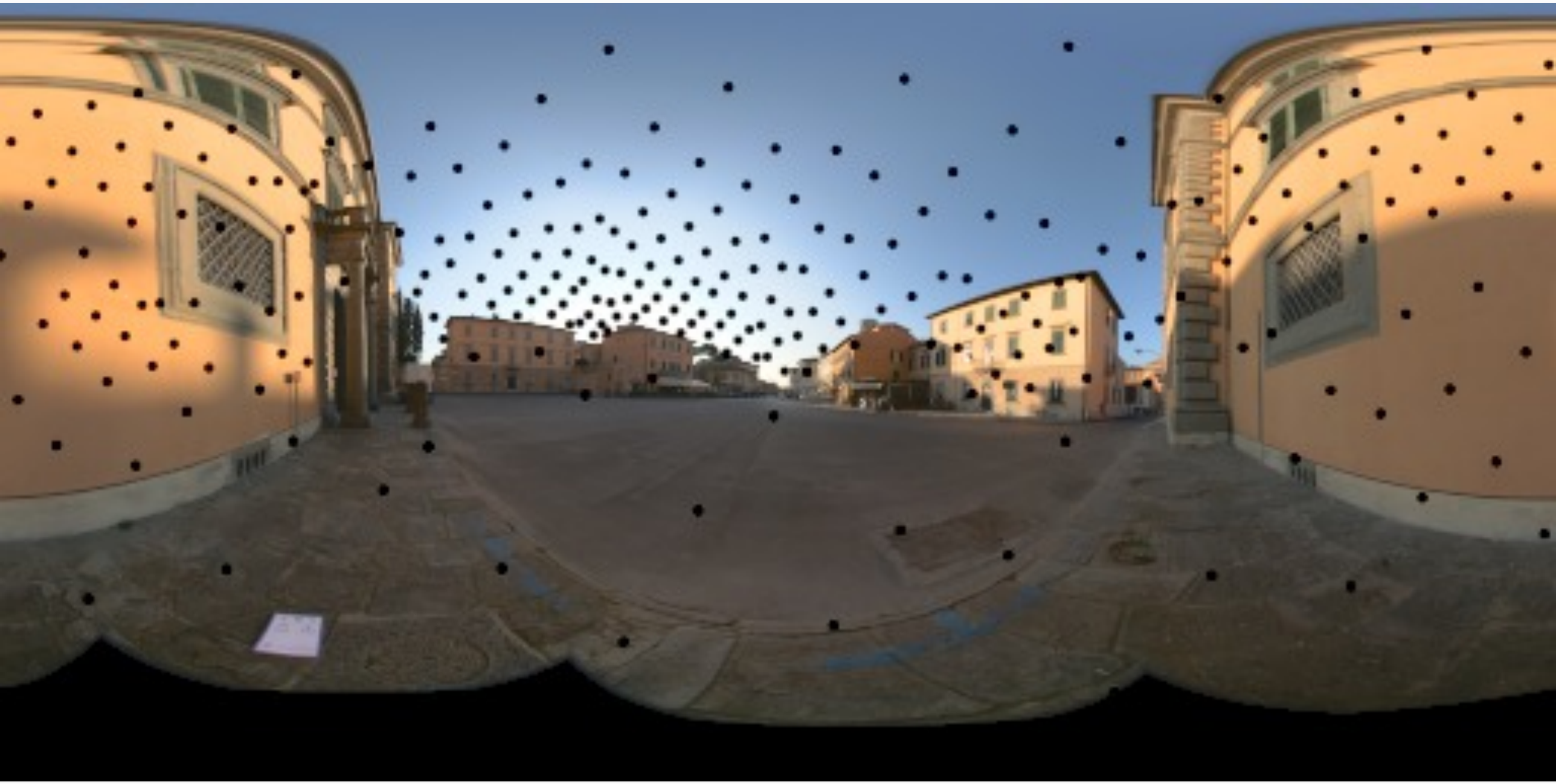
HDR LIGHT PROBE

Pisa light probe, *courtesy of Paul Debevec*

WARPED POINT SET

Warped Hammersley set (256 points)

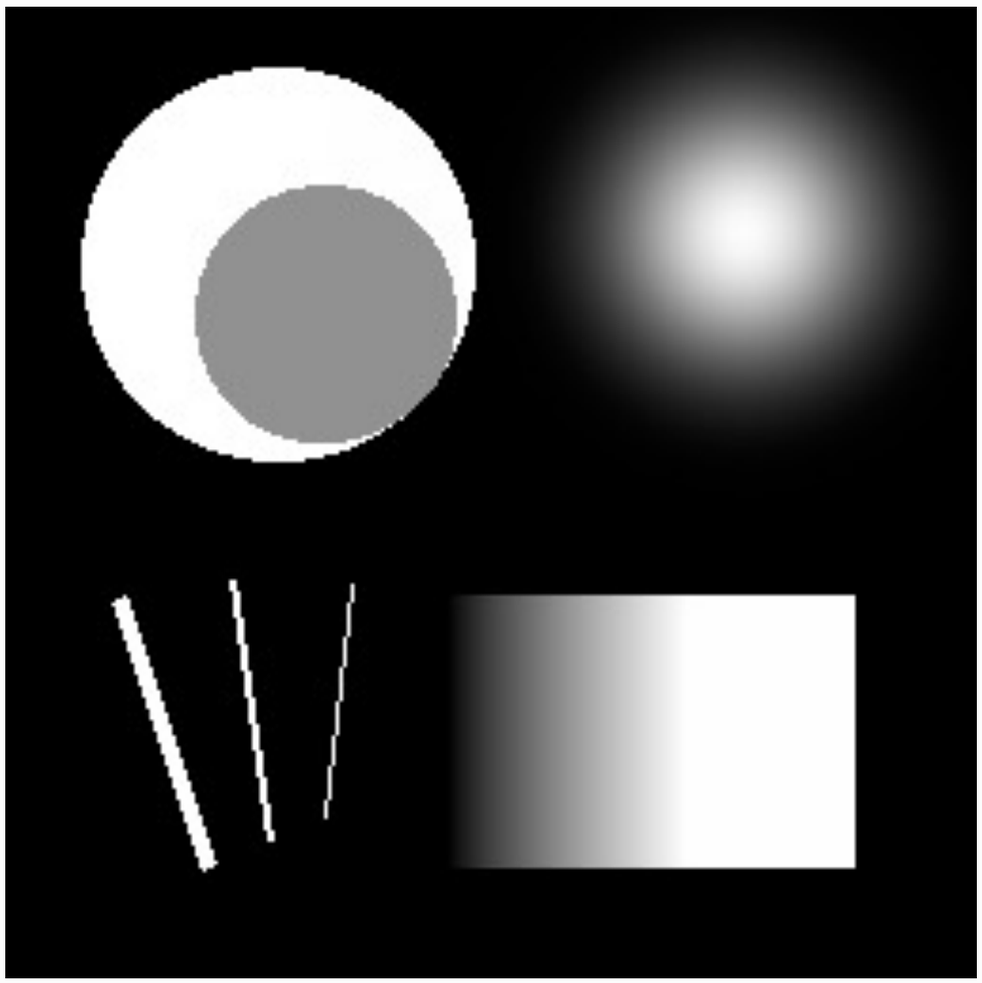
WARPED POINT SET



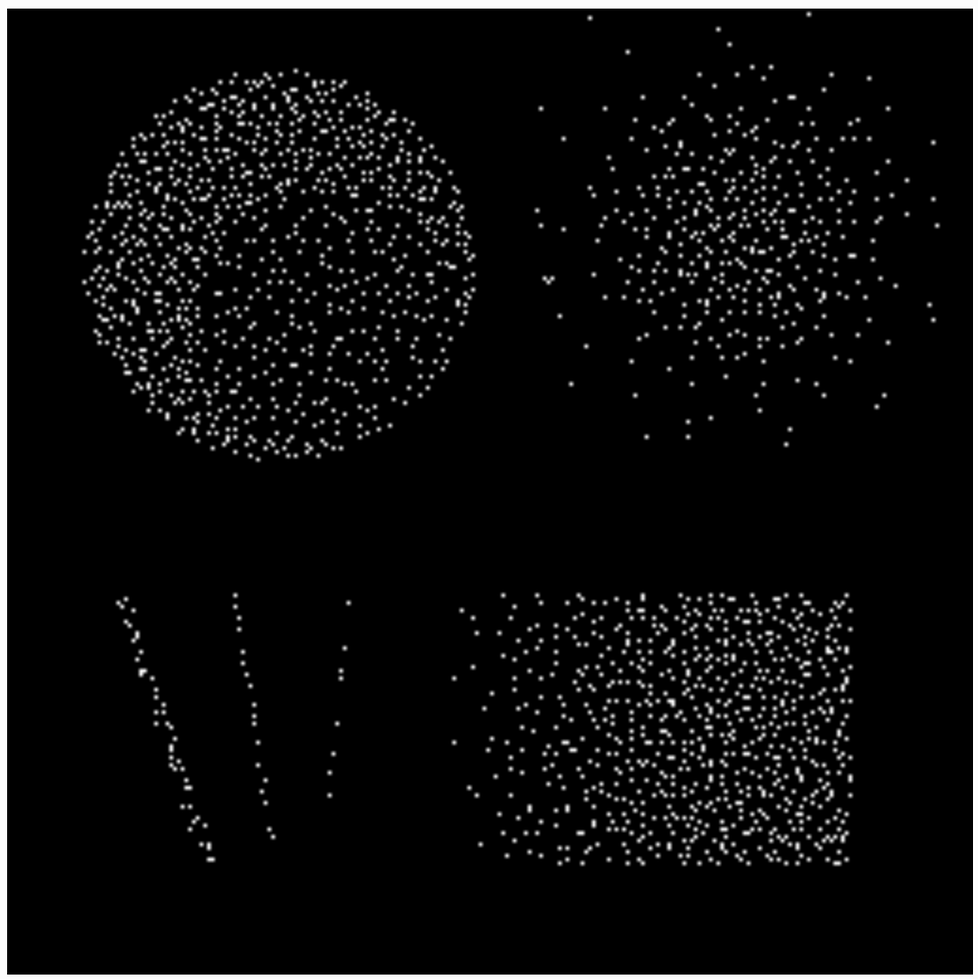
Warped golden lattice (256 points)

PROBLEM STATEMENT

Problem: Generate a *sequence* of low-discrepancy points from a given probability density function (PDF) represented as a discrete grayscale intensity image.

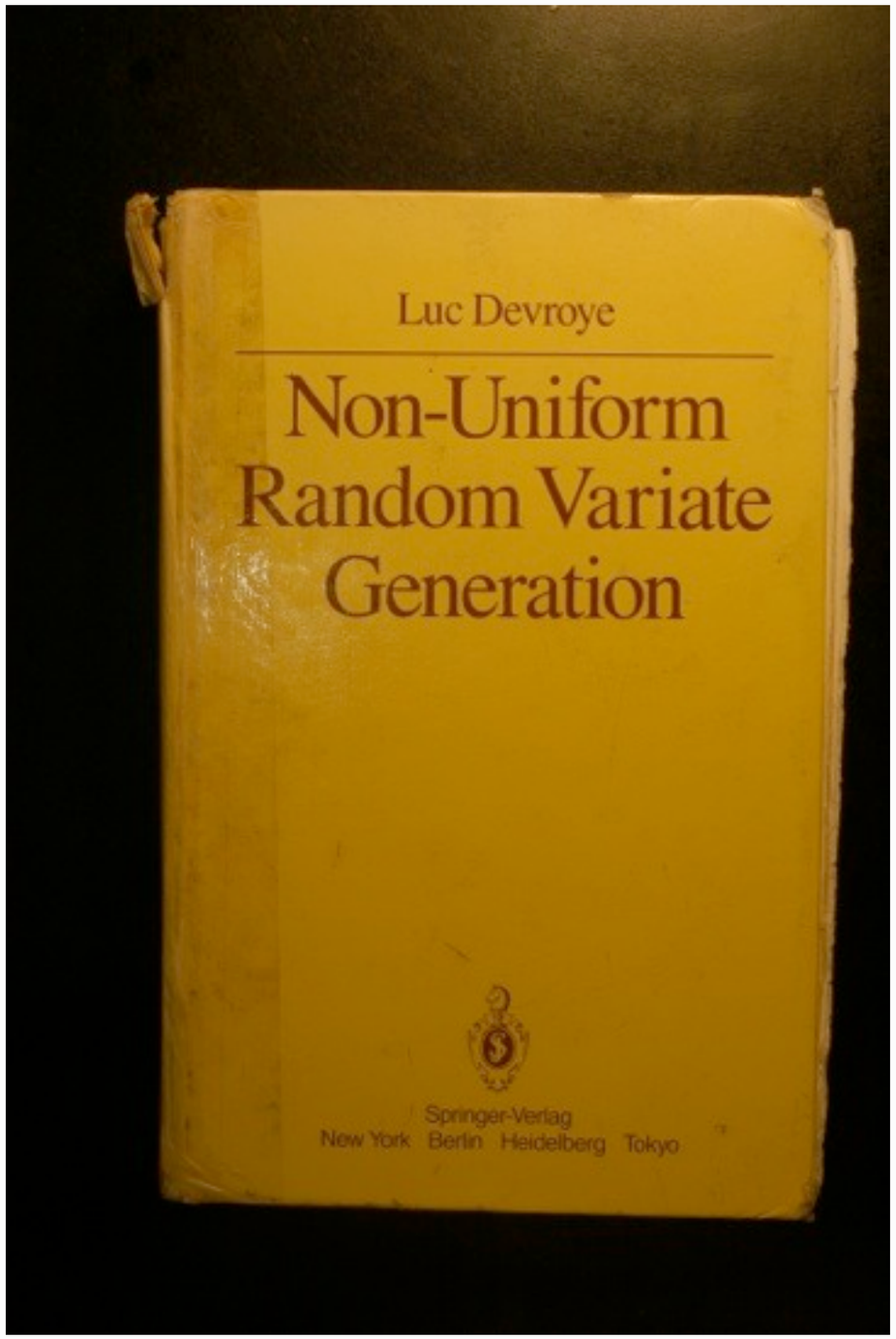


Input: density image



Output: point sequence

THE INVERSION METHOD



THE INVERSION METHOD

Given the cumulative density function **F** from the probability density function (PDF) **f**

1. Generate a uniform *random* number **u** in the interval **[0,1)**
2. With numerical integration, compute **x** such that **F(x) = u**
3. Take **x** to be a non-uniform *random* number drawn from **f**

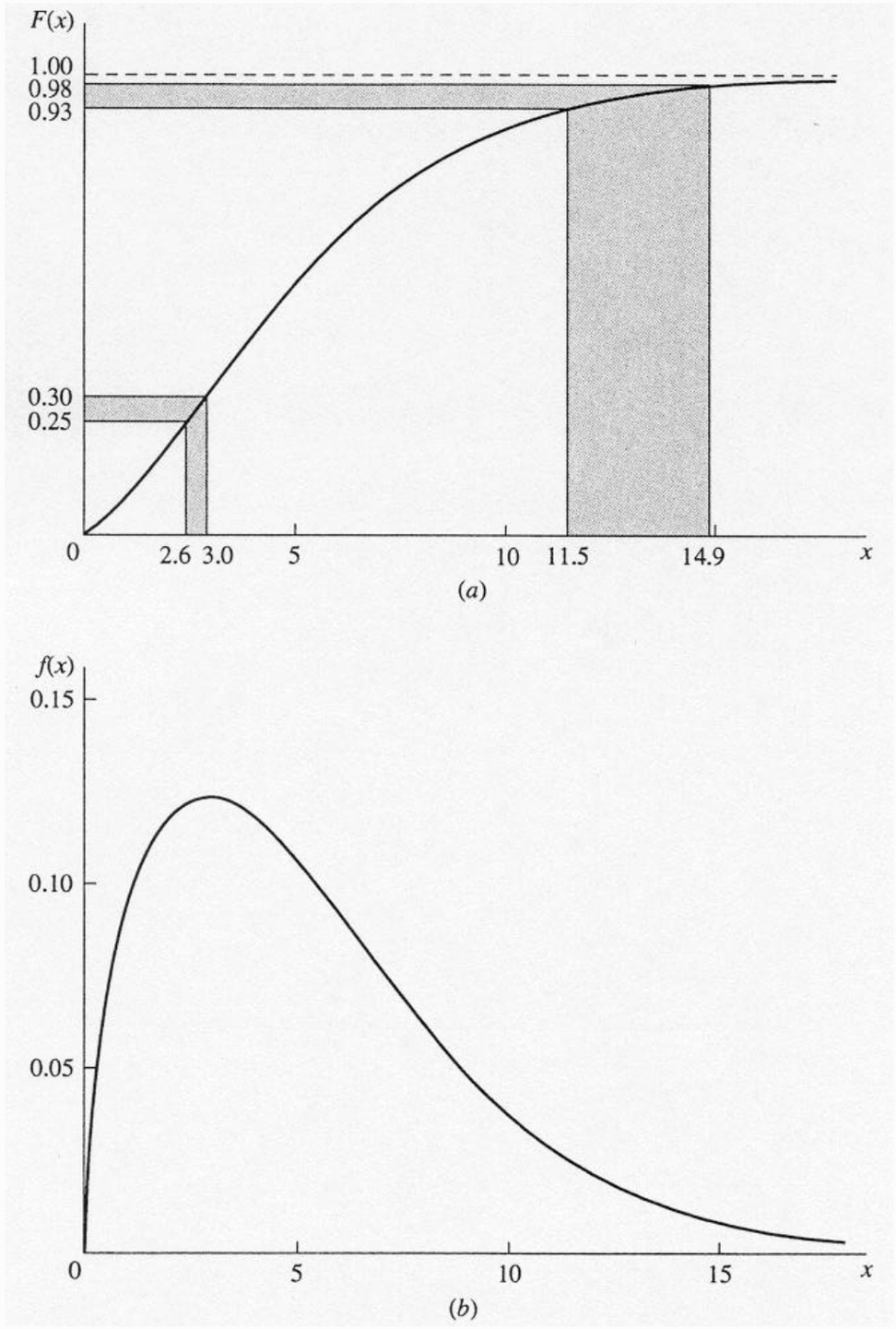


Illustration from D. Kelton, 2002

THE INVERSION METHOD

Given the cumulative density function **F** from the probability density function (PDF) **f**

1. Generate a uniform *random* number **u** in the interval **[0,1)**
2. With numerical integration, compute **x** such that **F(x) = u**
3. Take **x** to be a non-uniform *random* number drawn from **f**

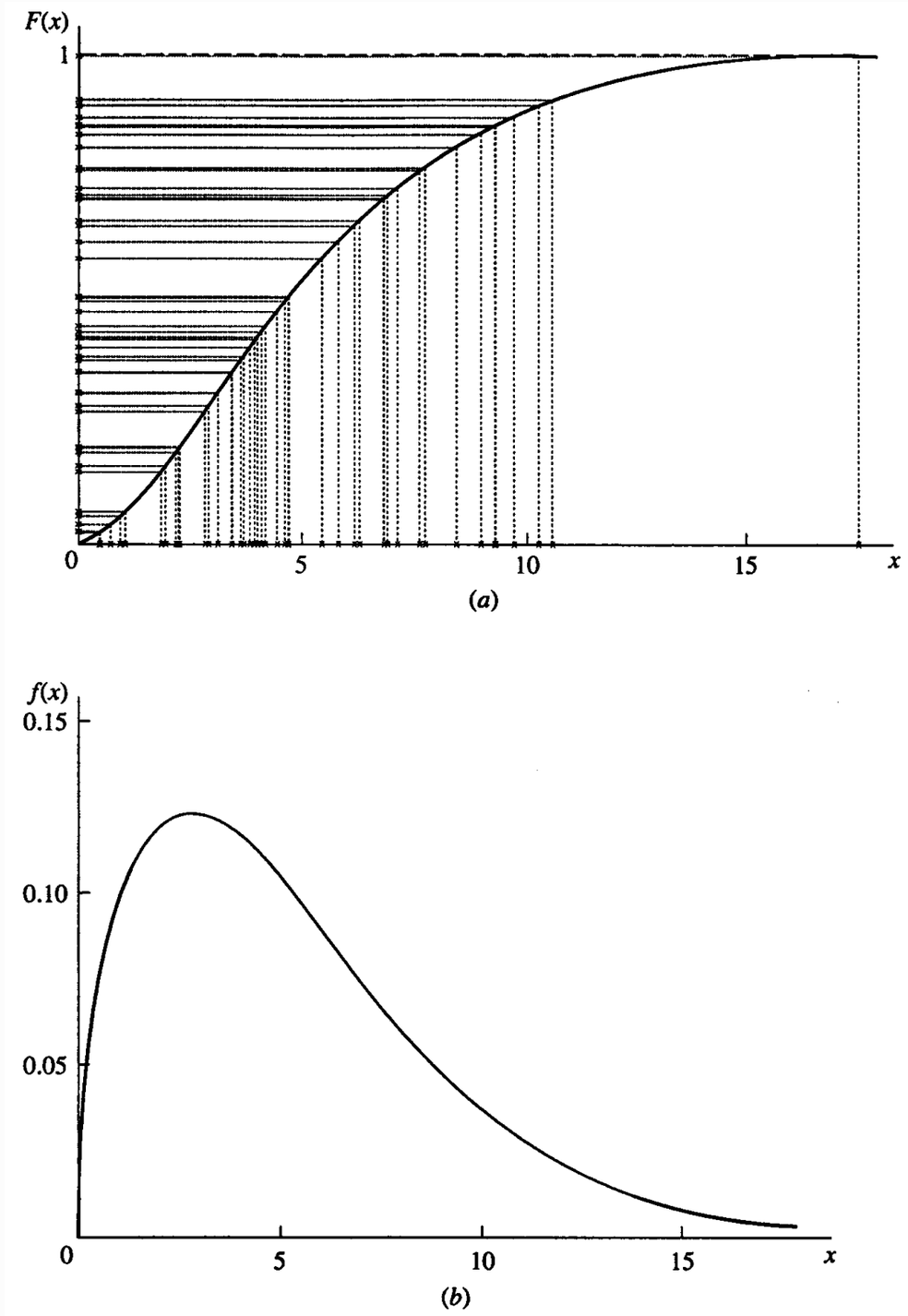


Illustration from D. Kelton, 2002

2D INVERSION METHODS

A- DUAL STEPS METHOD

Generate a low-discrepancy 2D point sequence, e.g. the Halton sequence.

Then, transform the first coordinate with a marginal distribution function and the inversion method.

Then, transform the second coordinate with the corresponding conditional distribution function.

(*Devroye, p. 96, 1986*)

B- SINGLE STEP METHOD

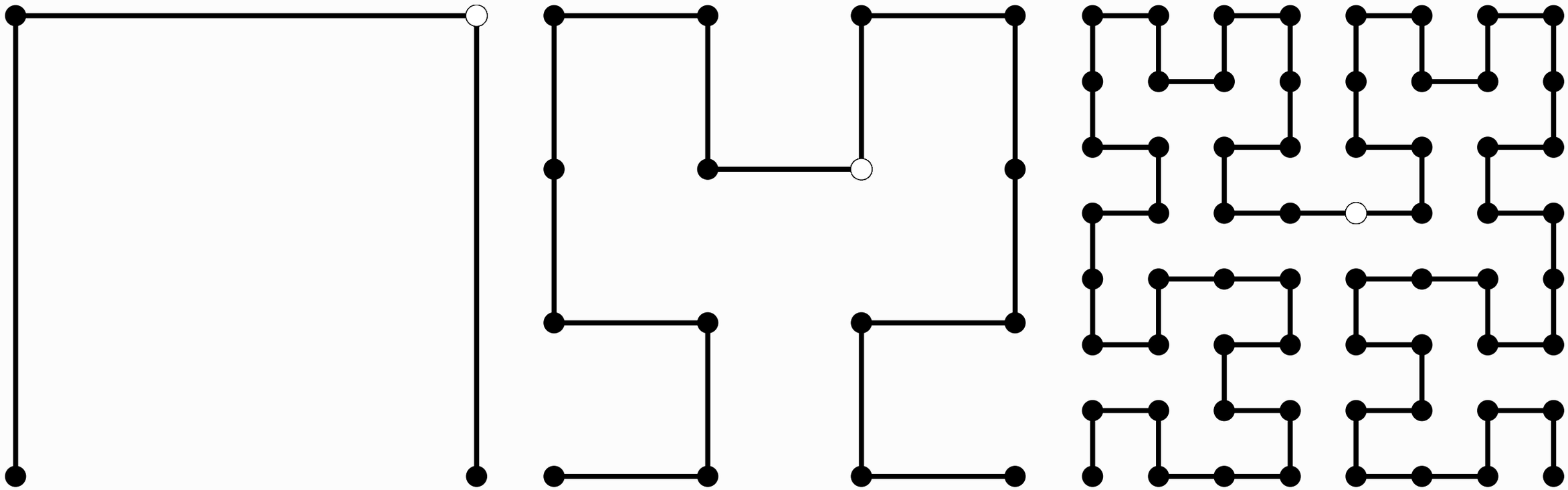
Generate a suitable low-discrepancy 1D sequence, *e.g.* the Golden ratio sequence.

Then, transform the 1D coordinate along a suitable unfolded image, with the inversion method.

Then, map the coordinate to a higher-dimensional point with the inverse Hilbert space filling curve.

(*Schretter and Niederreiter, 2012*)

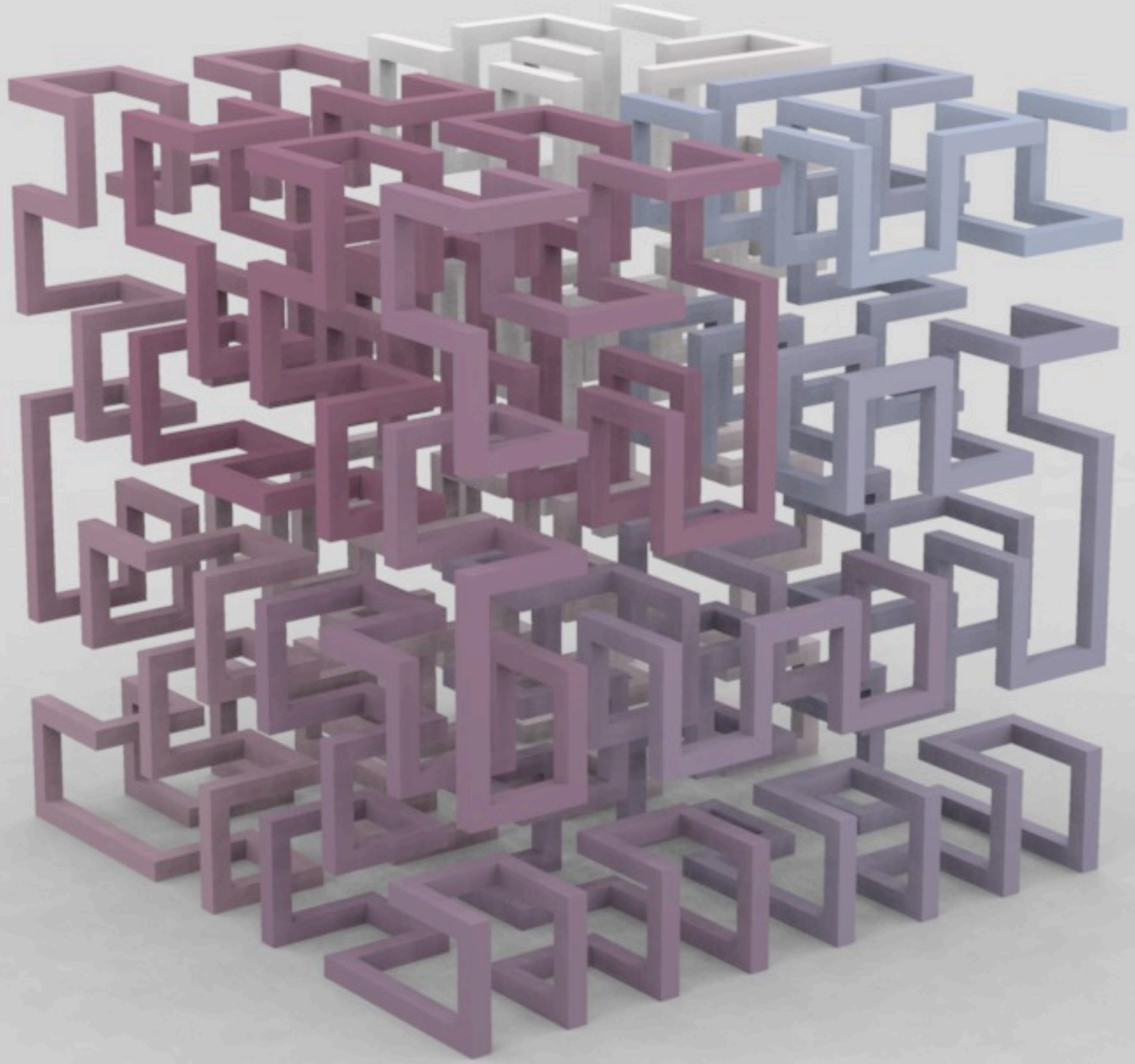
SPACE FILLING CURVE



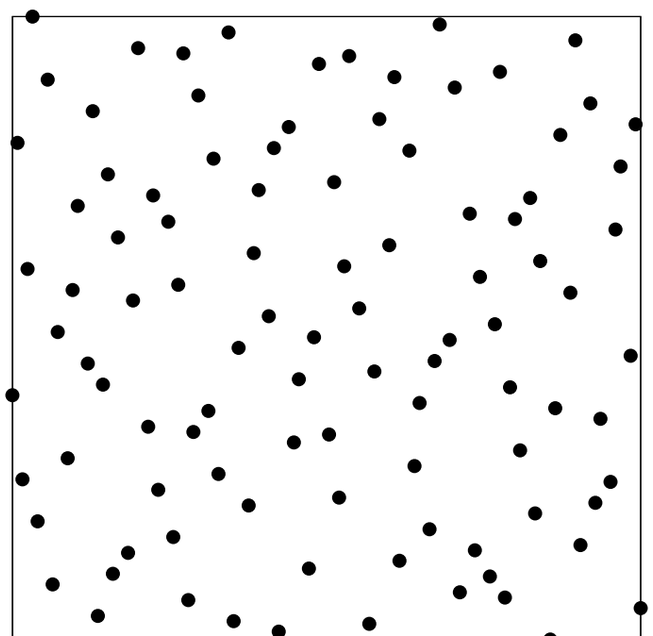
1st order Hilbert SFC 2nd order Hilbert SFC 3rd order Hilbert SFC

The position of the hollow point corresponds to the middle position along the path of the Hilbert SFC.

Arbitrary spatial precision can be reached with a sufficient order of recursion (*order 24 for IEEE floats*).

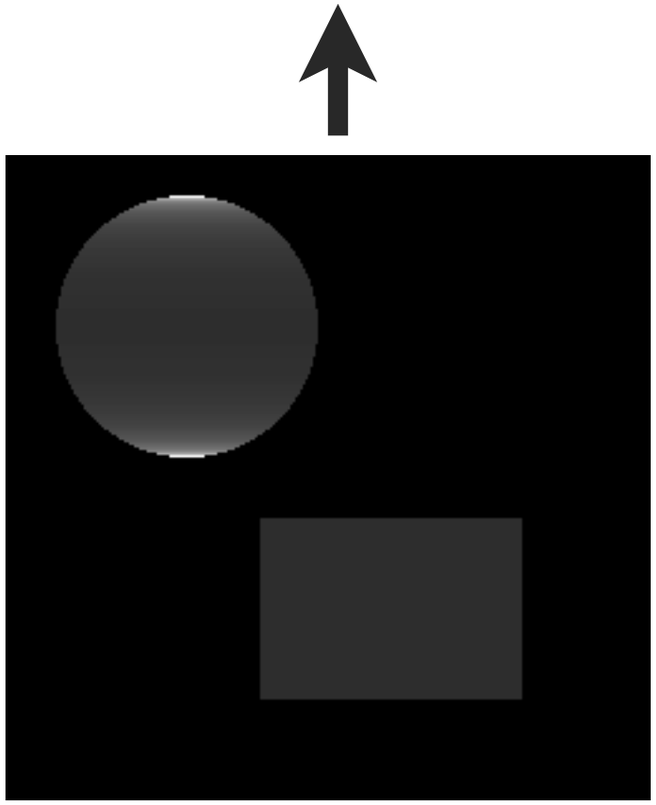


A- DUAL STEPS METHOD

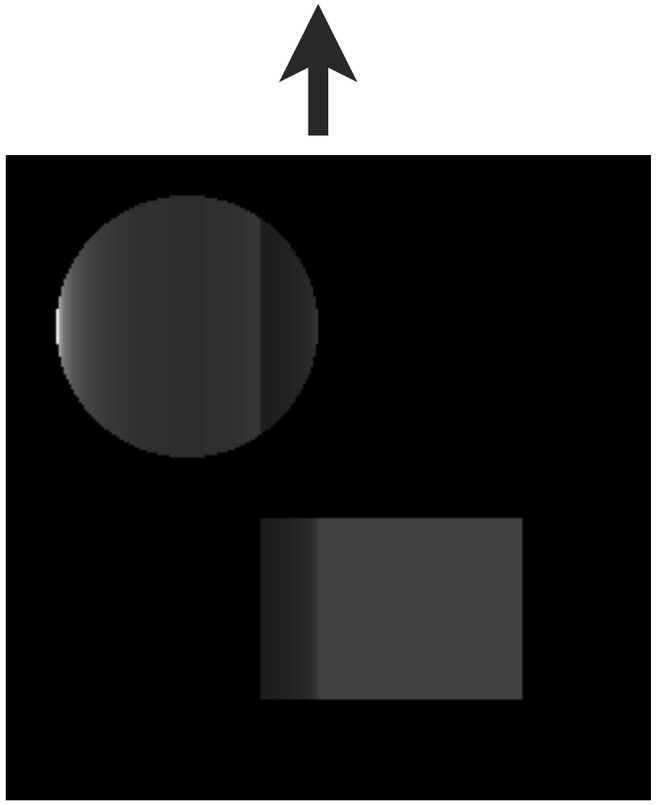


Halton sequence

Density image

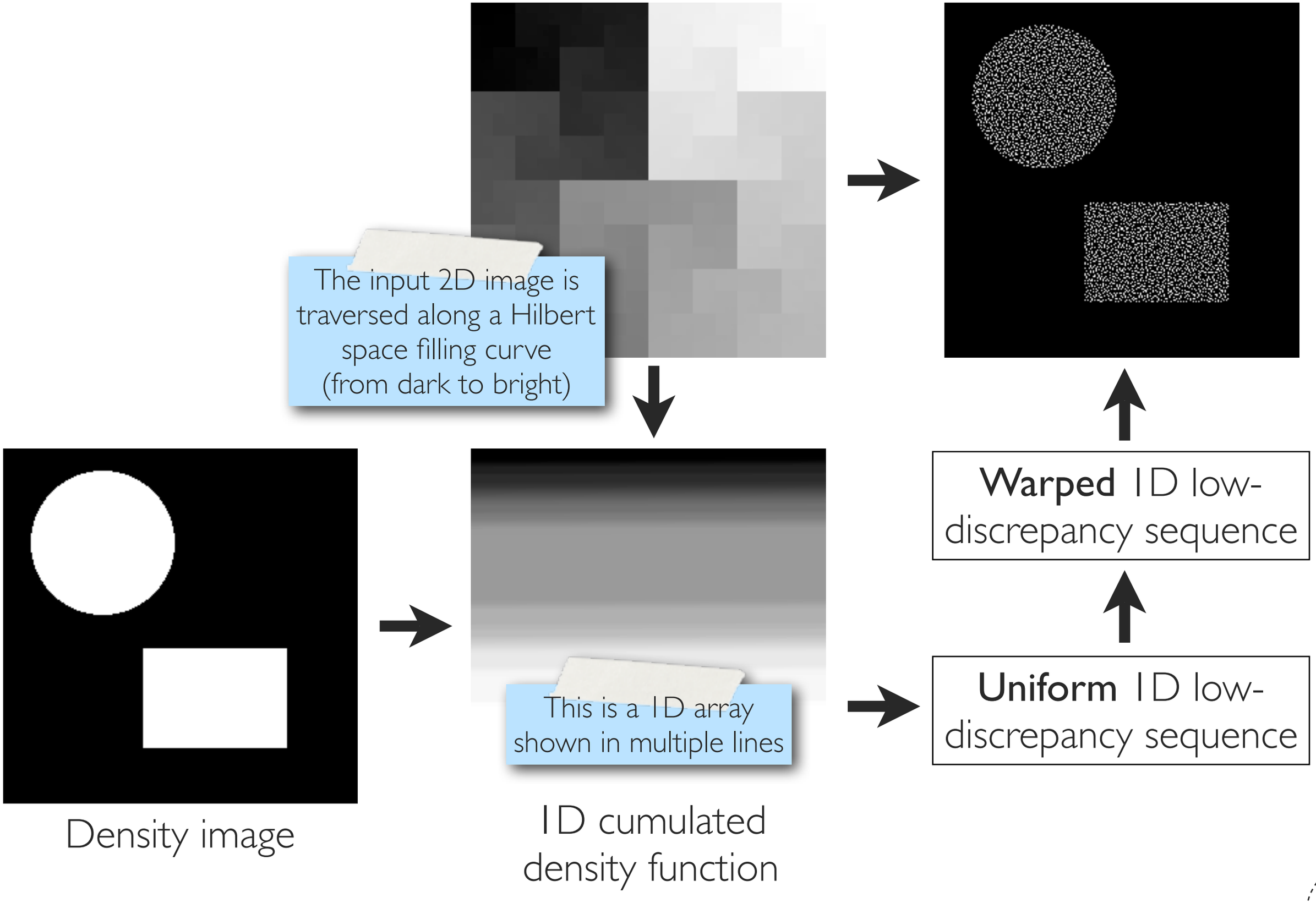


1D marginal (y) and conditional (x) PDFs

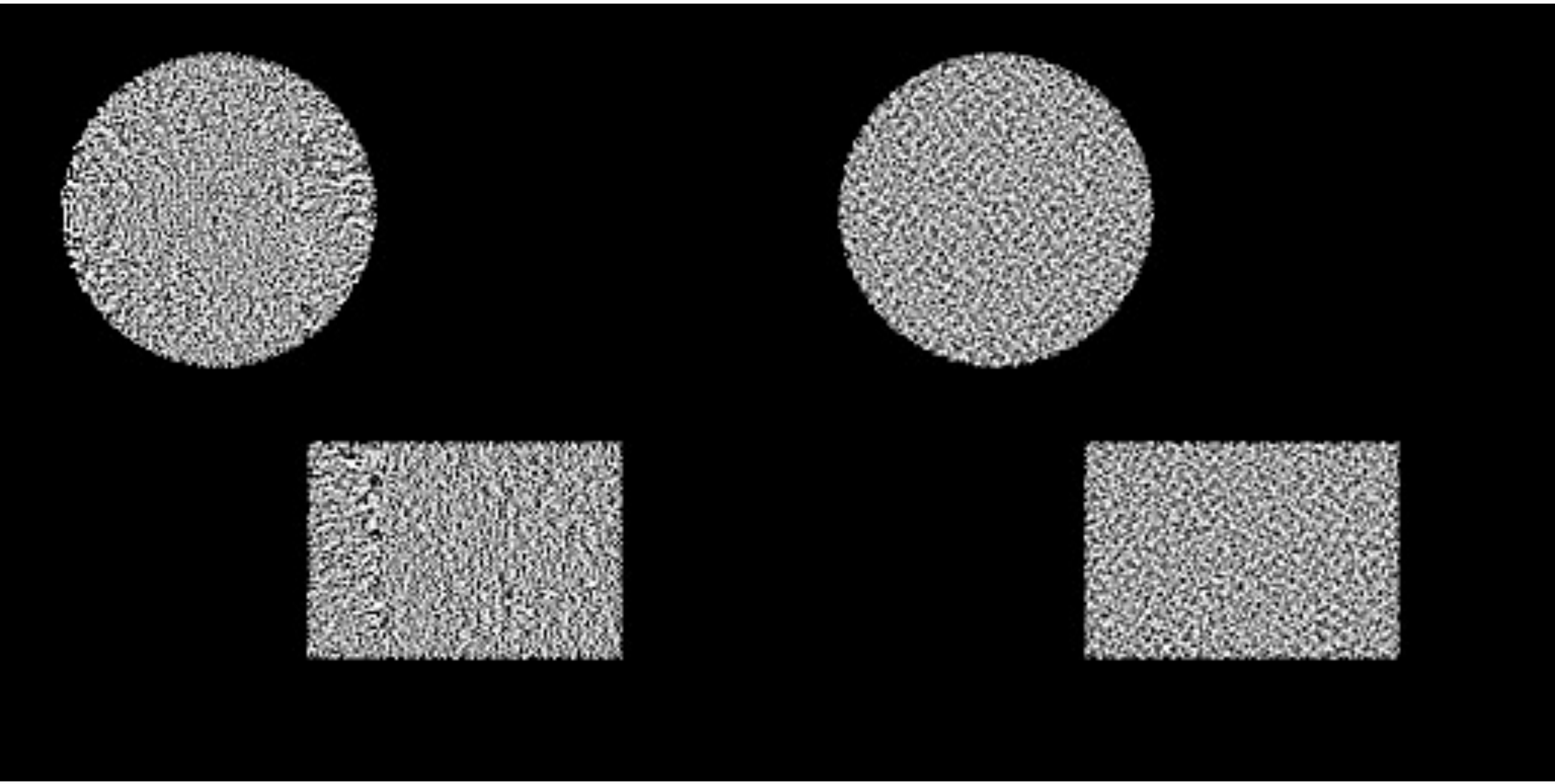


1D marginal (x) and conditional (y) PDFs

B- SINGLE STEP METHOD



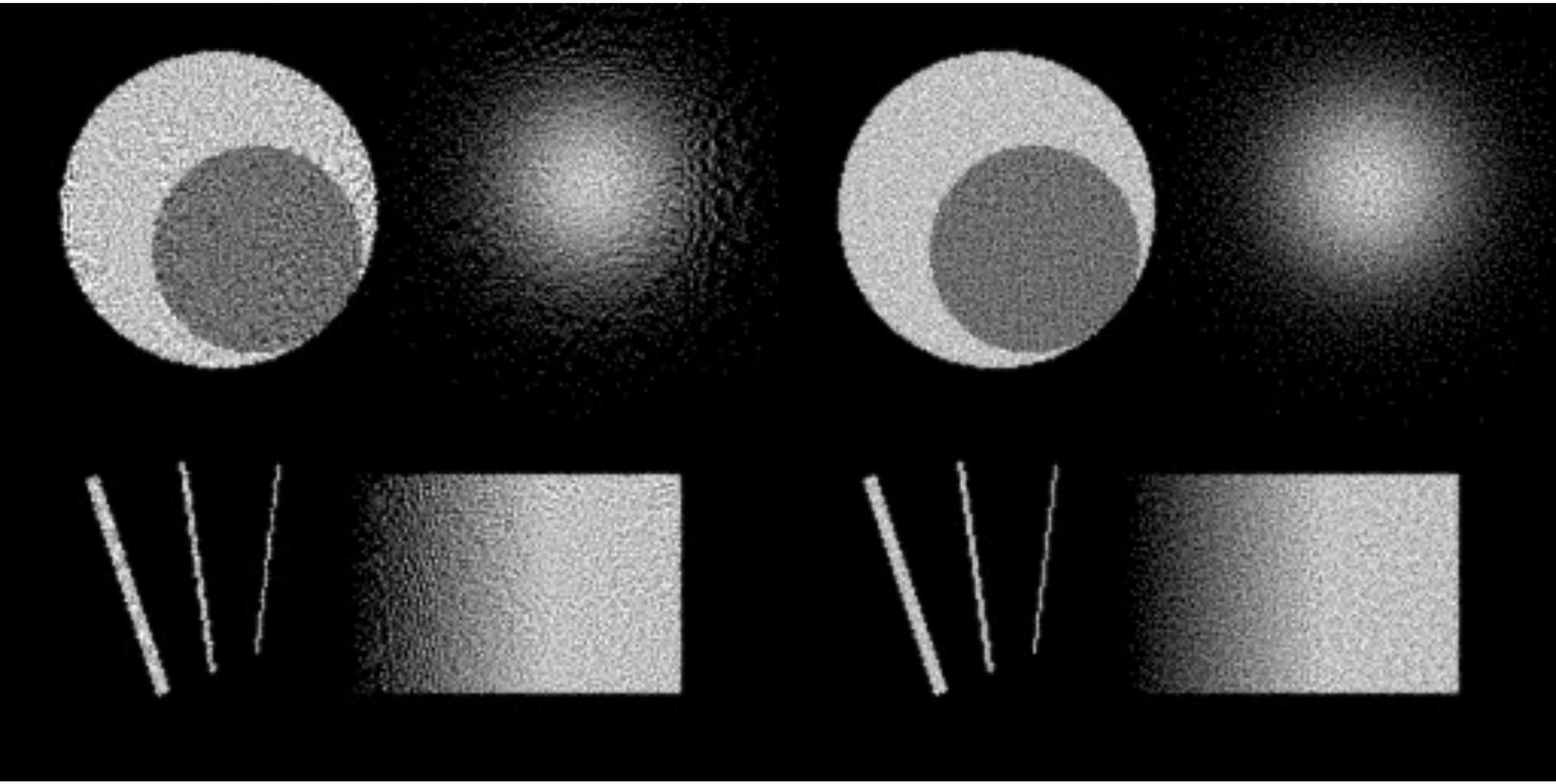
INVERSION **RESULTS**



Dual inversion, 65536 points

Single inversion, 65536 points

INVERSION **RESULTS**

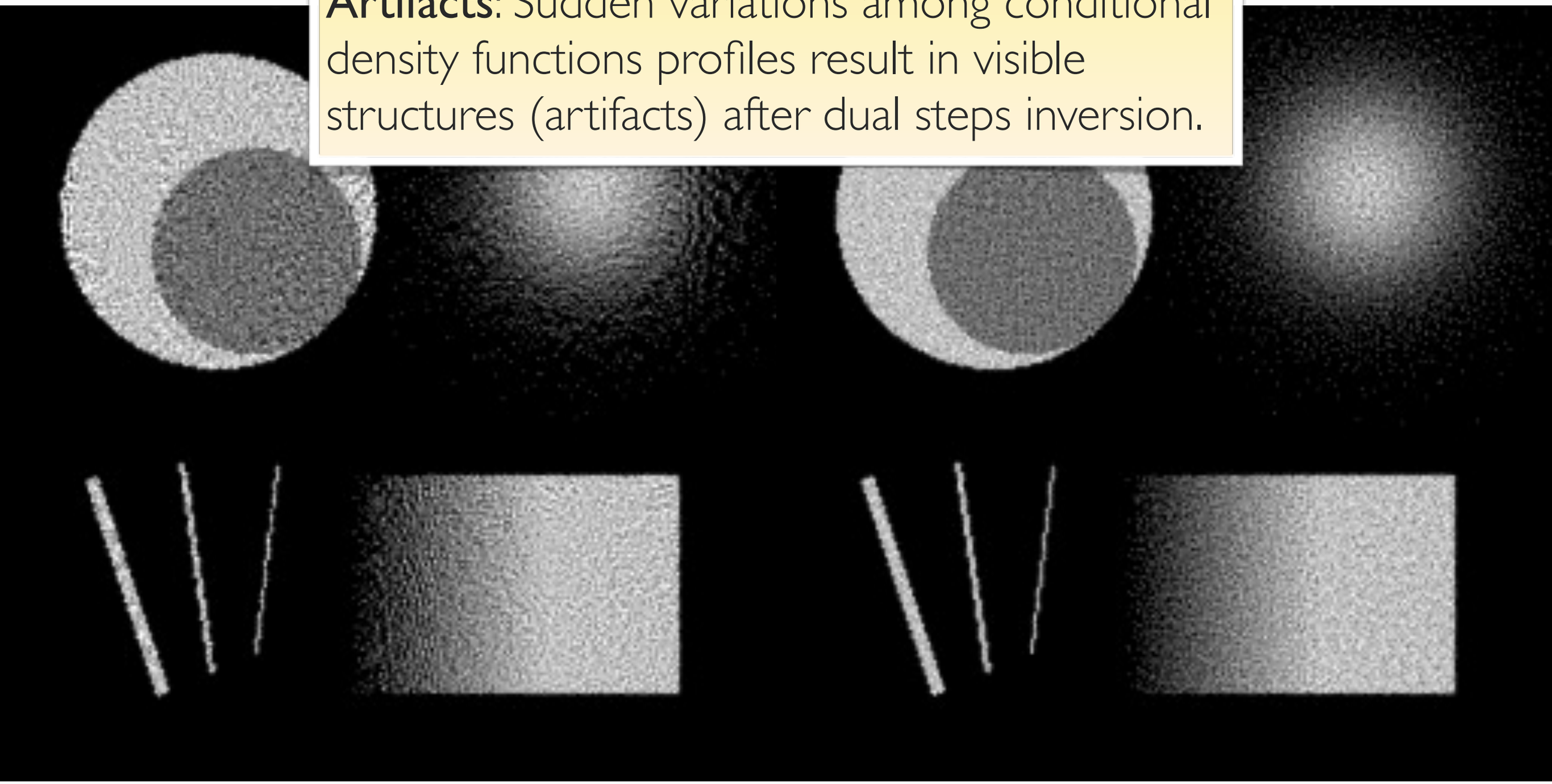


Dual inversion, 65536 points

Single inversion, 65536 points

INVERSION **RESULTS**

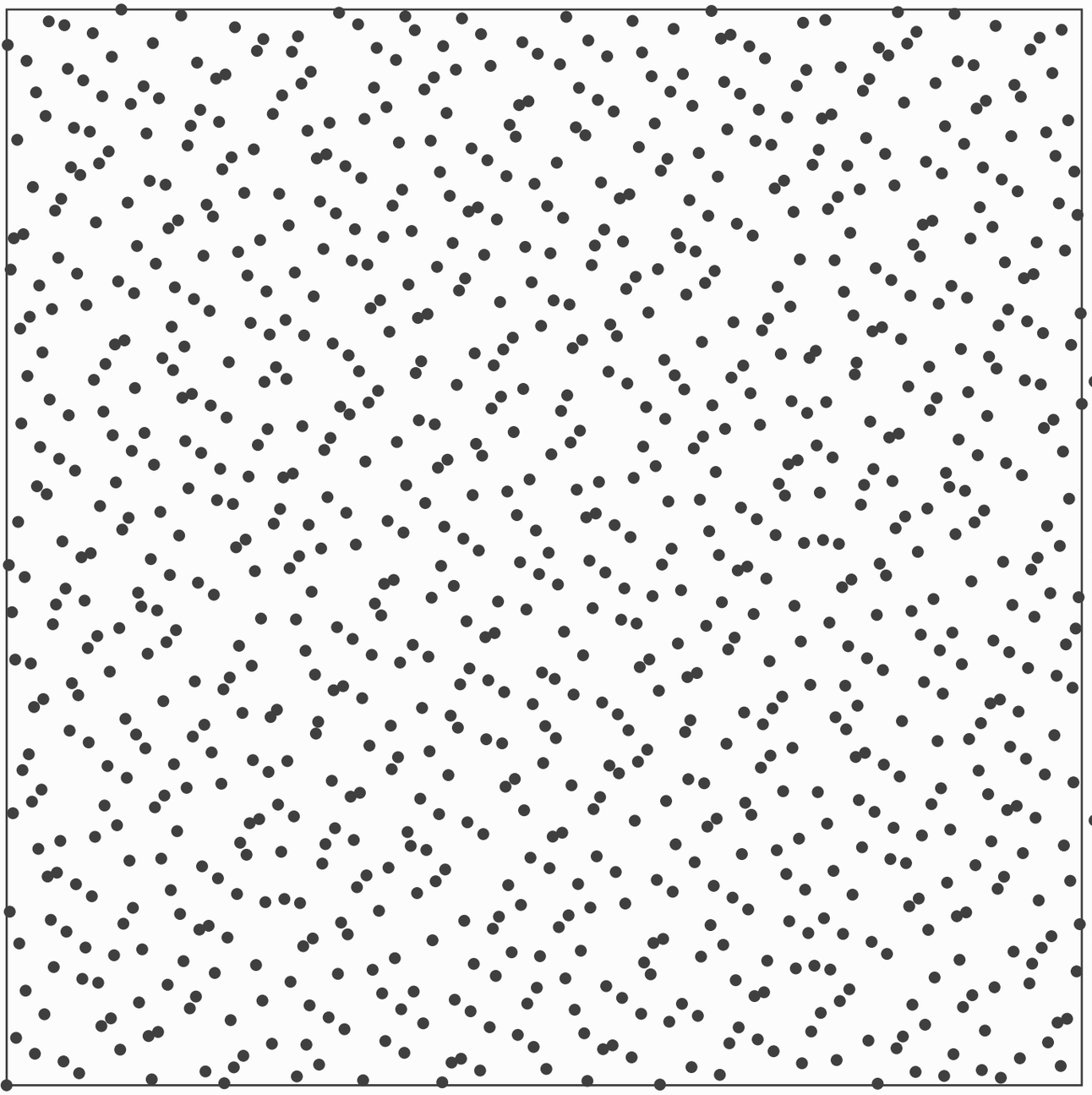
Artifacts: Sudden variations among conditional density functions profiles result in visible structures (artifacts) after dual steps inversion.



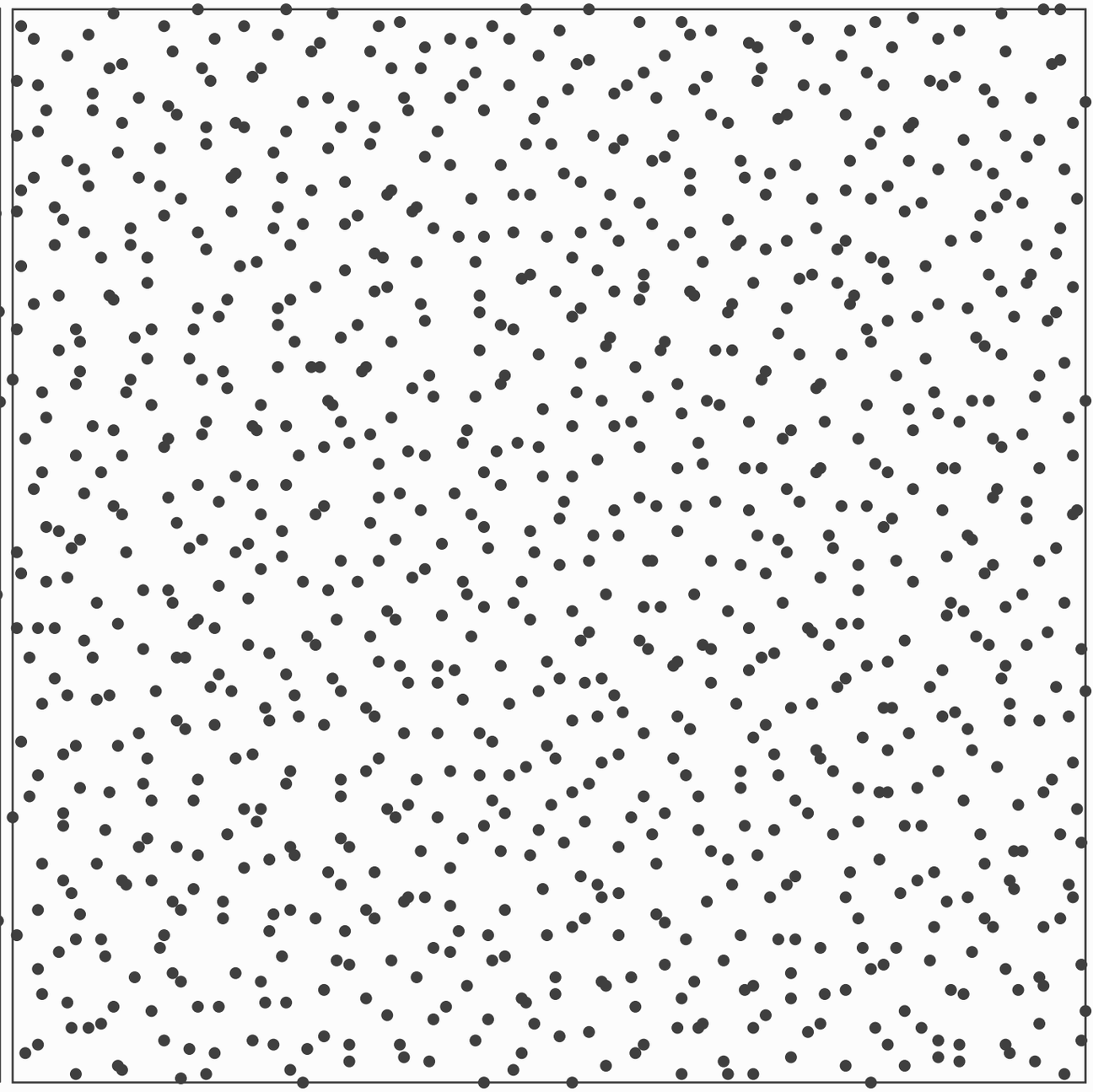
Dual inversion, 65536 points

Single inversion, 65536 points

UNIFORM DENSITY



Halton sequence

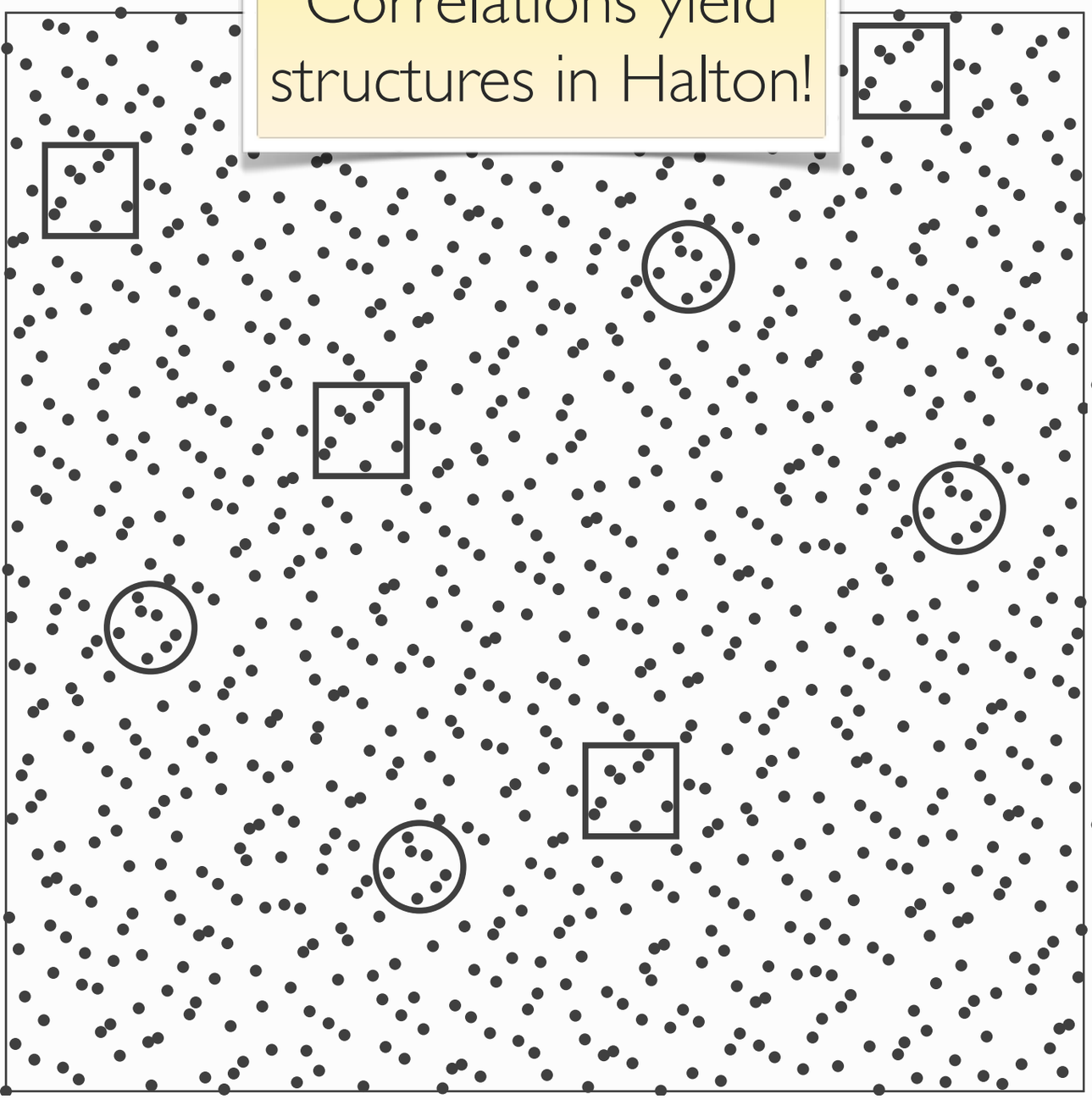


Golden sequence + Hilbert mapping

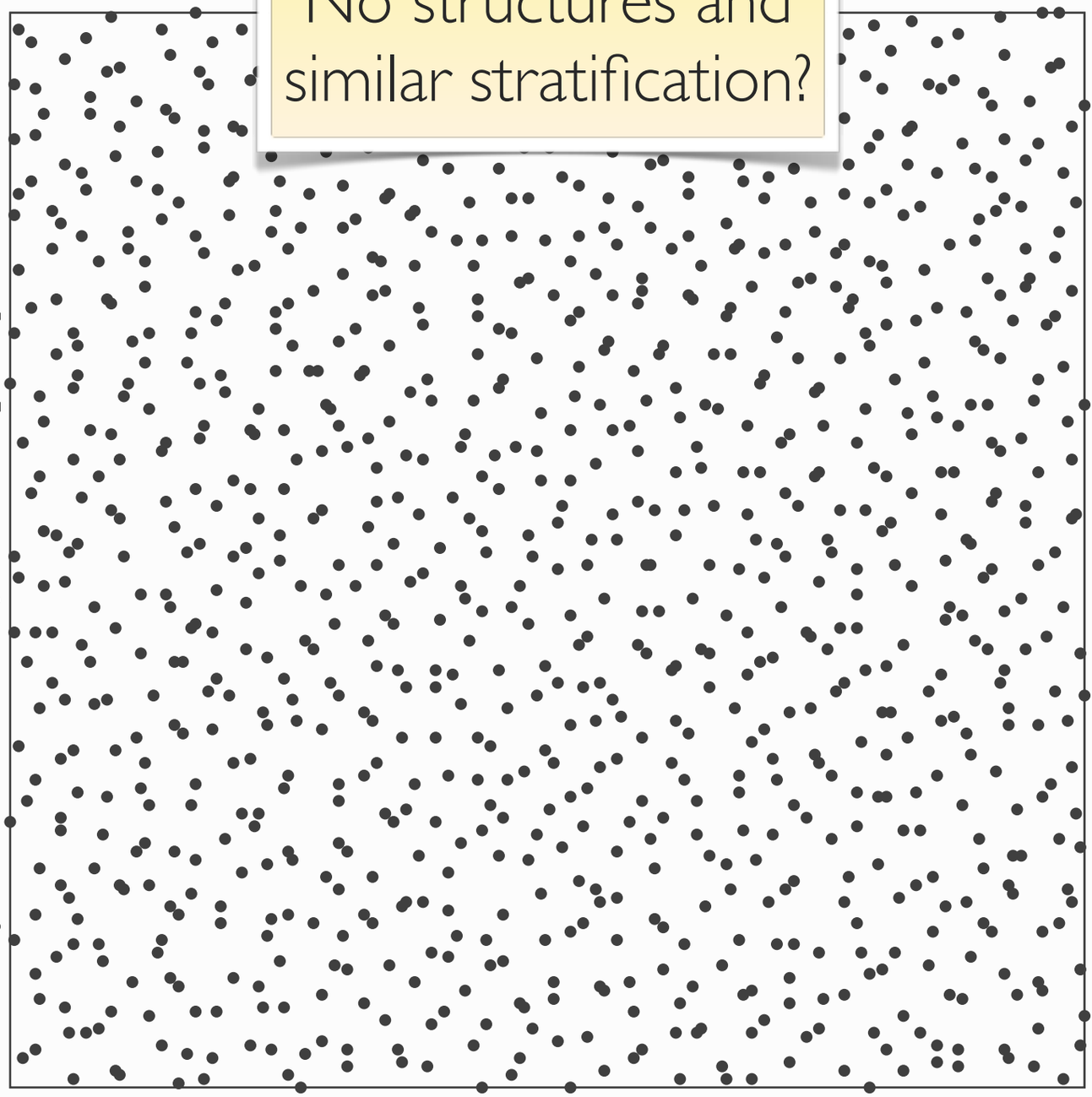
UNIFORM DENSITY

Correlations yield structures in Halton!

No structures and similar stratification?



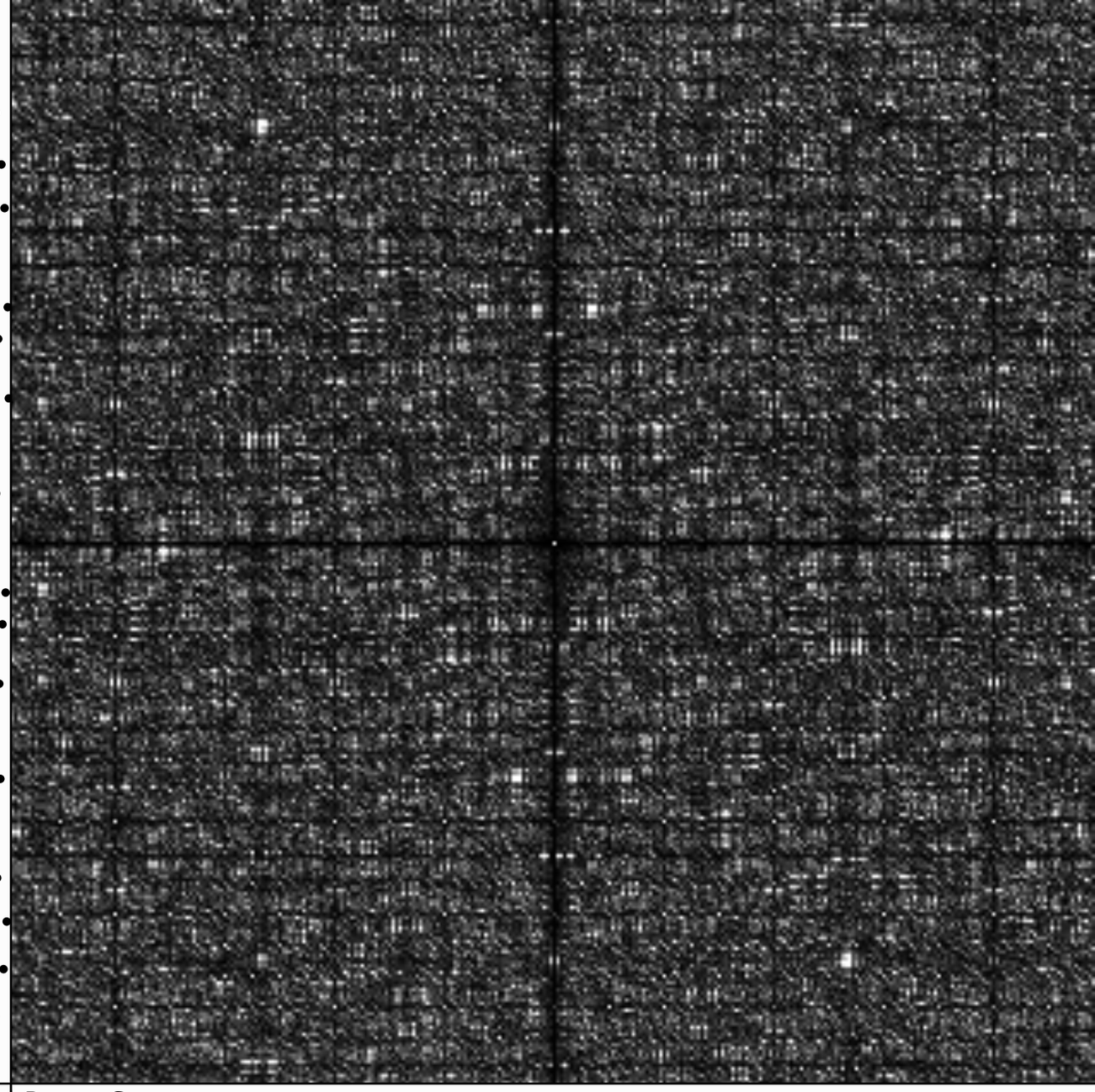
Halton sequence



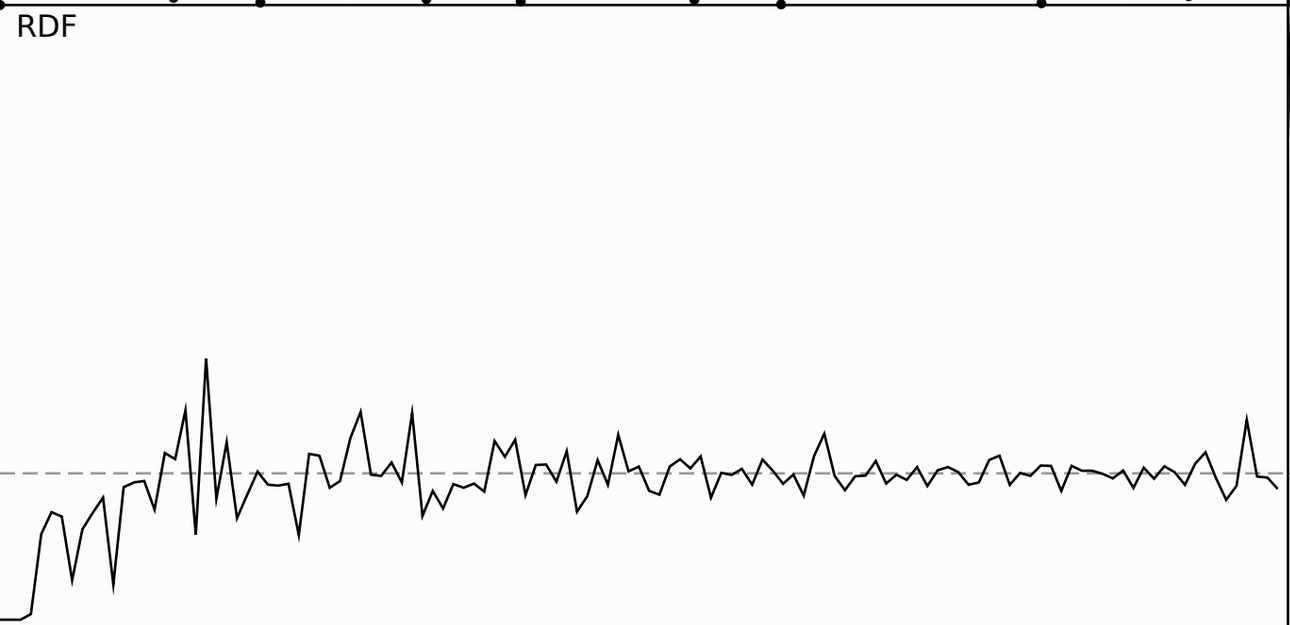
Golden sequence + Hilbert mapping

Gbl. Mindist	0.25539
Avg. Mindist	0.57096
Eff. Nyquist	0.22502
Oscillations	0.93326

Halton sequence



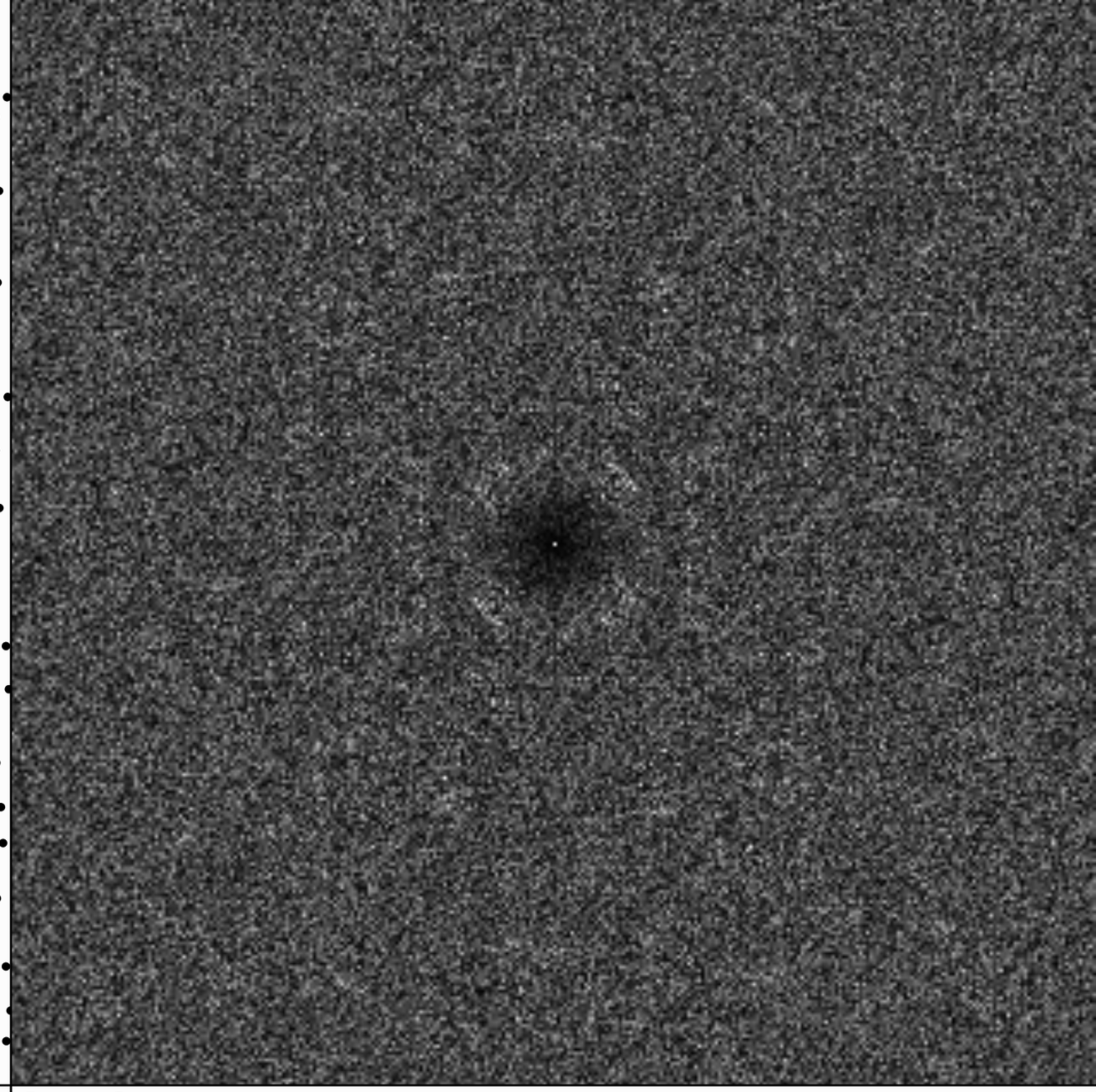
RDF



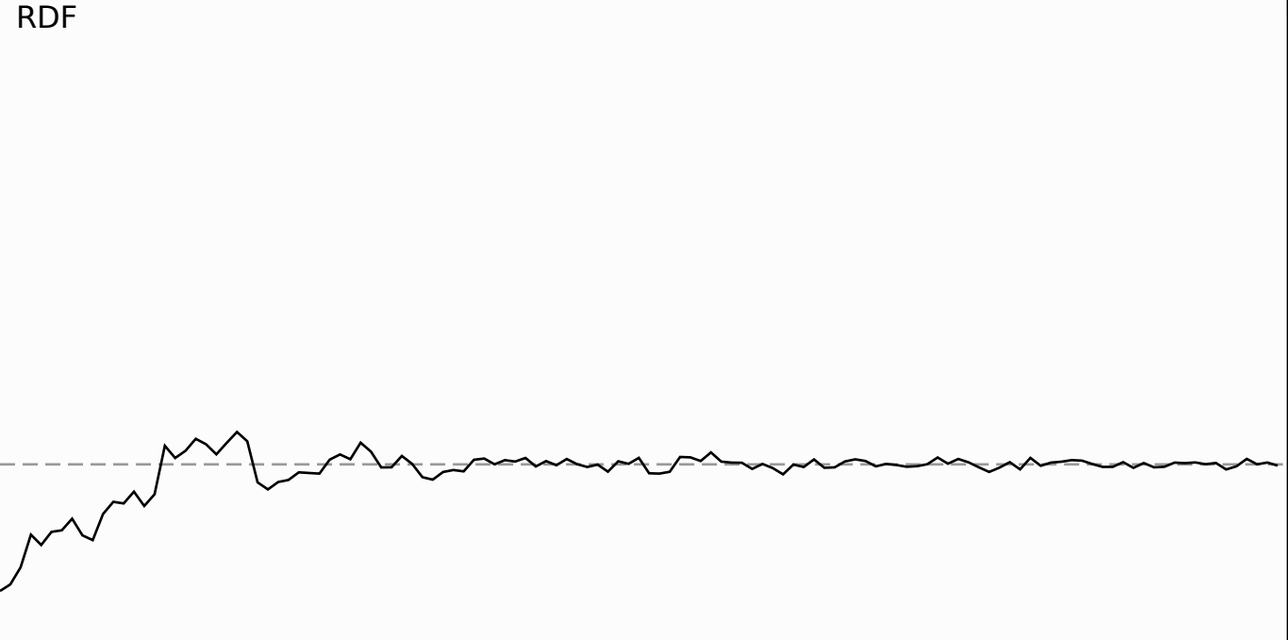
Power Spectrum

Gbl. Mindist	0.05219
Avg. Mindist	0.55083
Eff. Nyquist	0.22002
Oscillations	0.44923

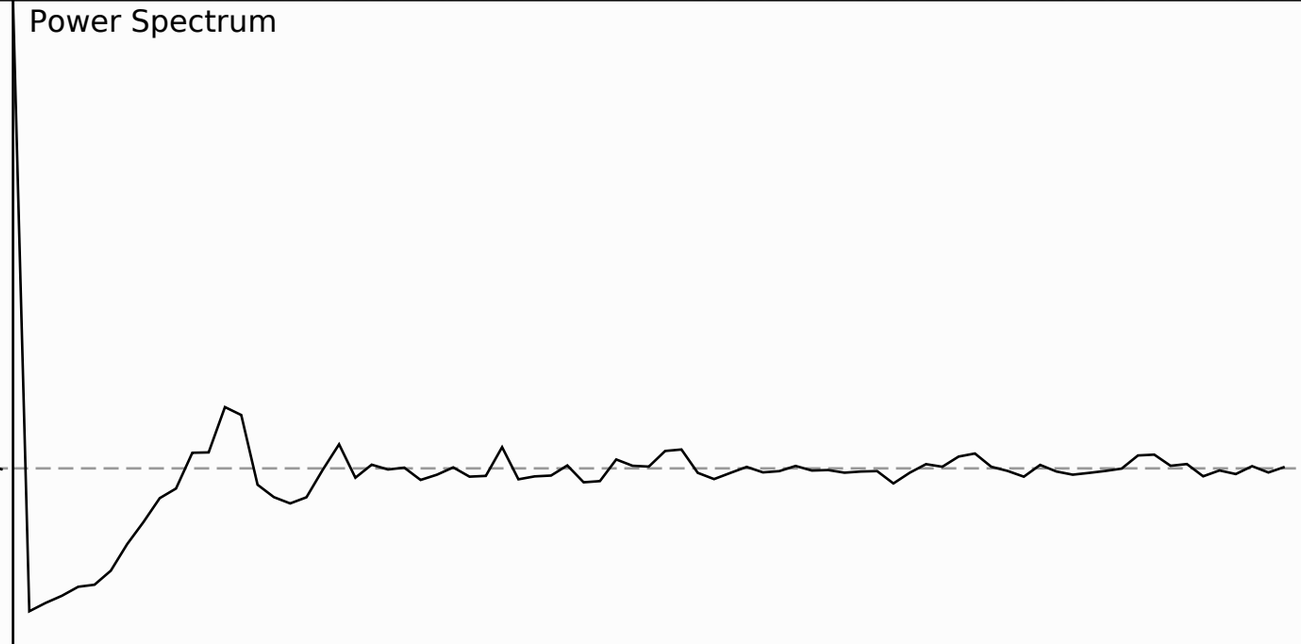
Golden sequence
+ Hilbert mapping



RDF

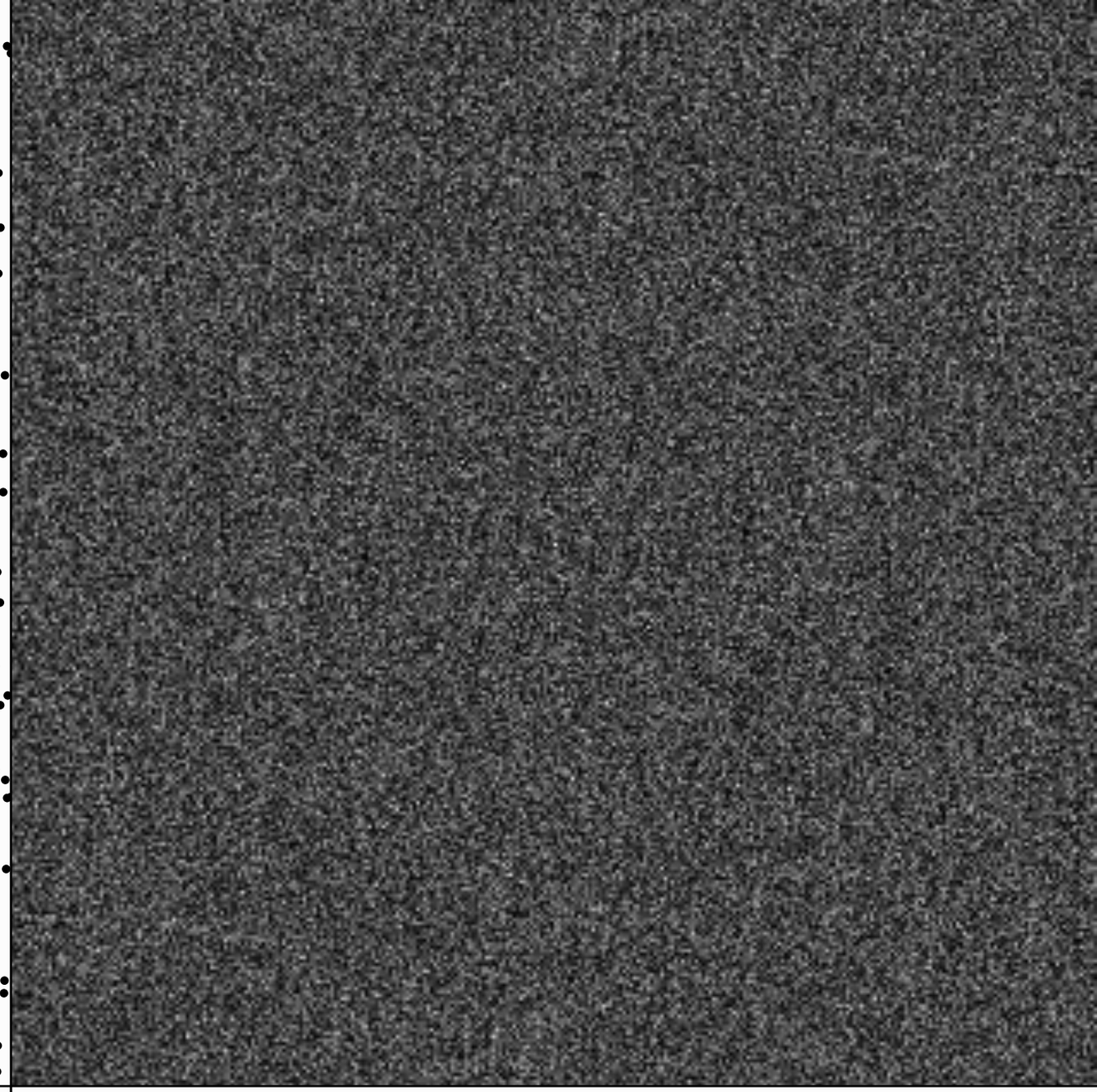


Power Spectrum

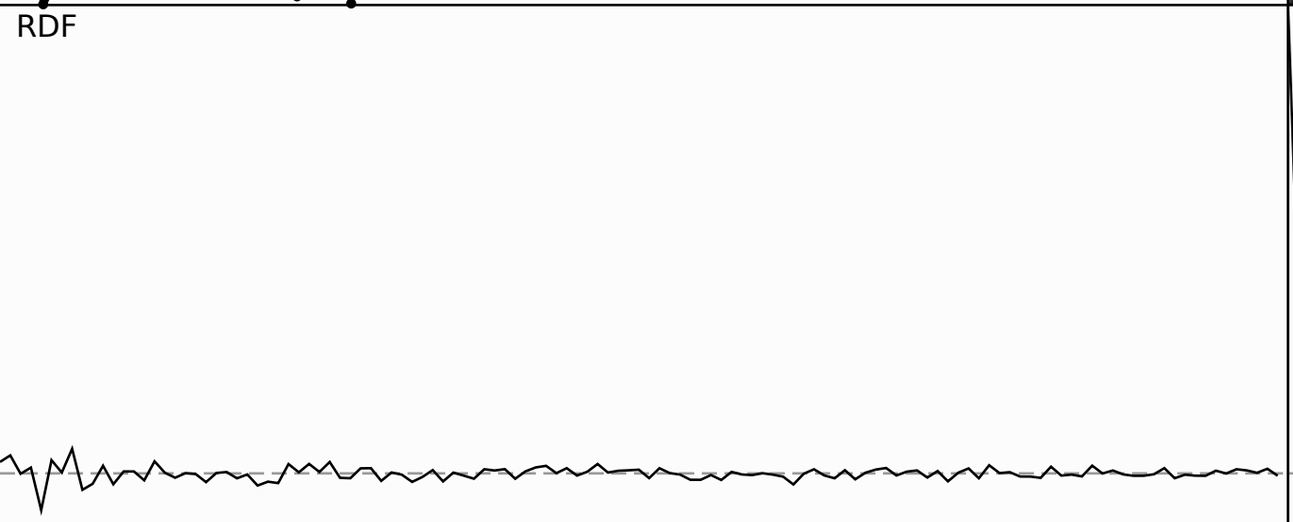


Gbl. Mindist	0.01838
Avg. Mindist	0.46482
Eff. Nyquist	0.00000
Oscillations	0.22947

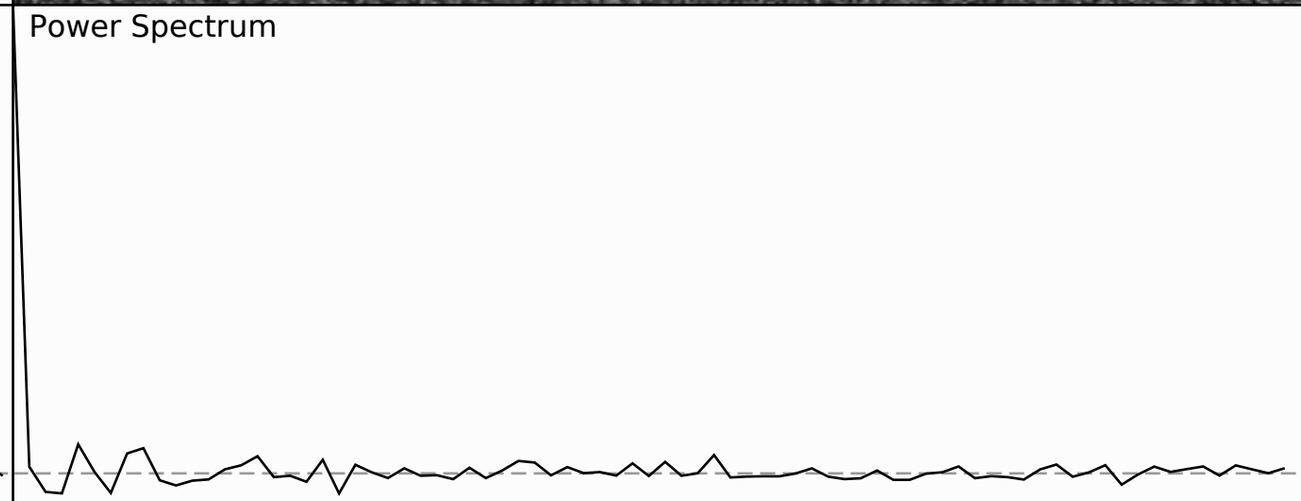
Random sequence



RDF

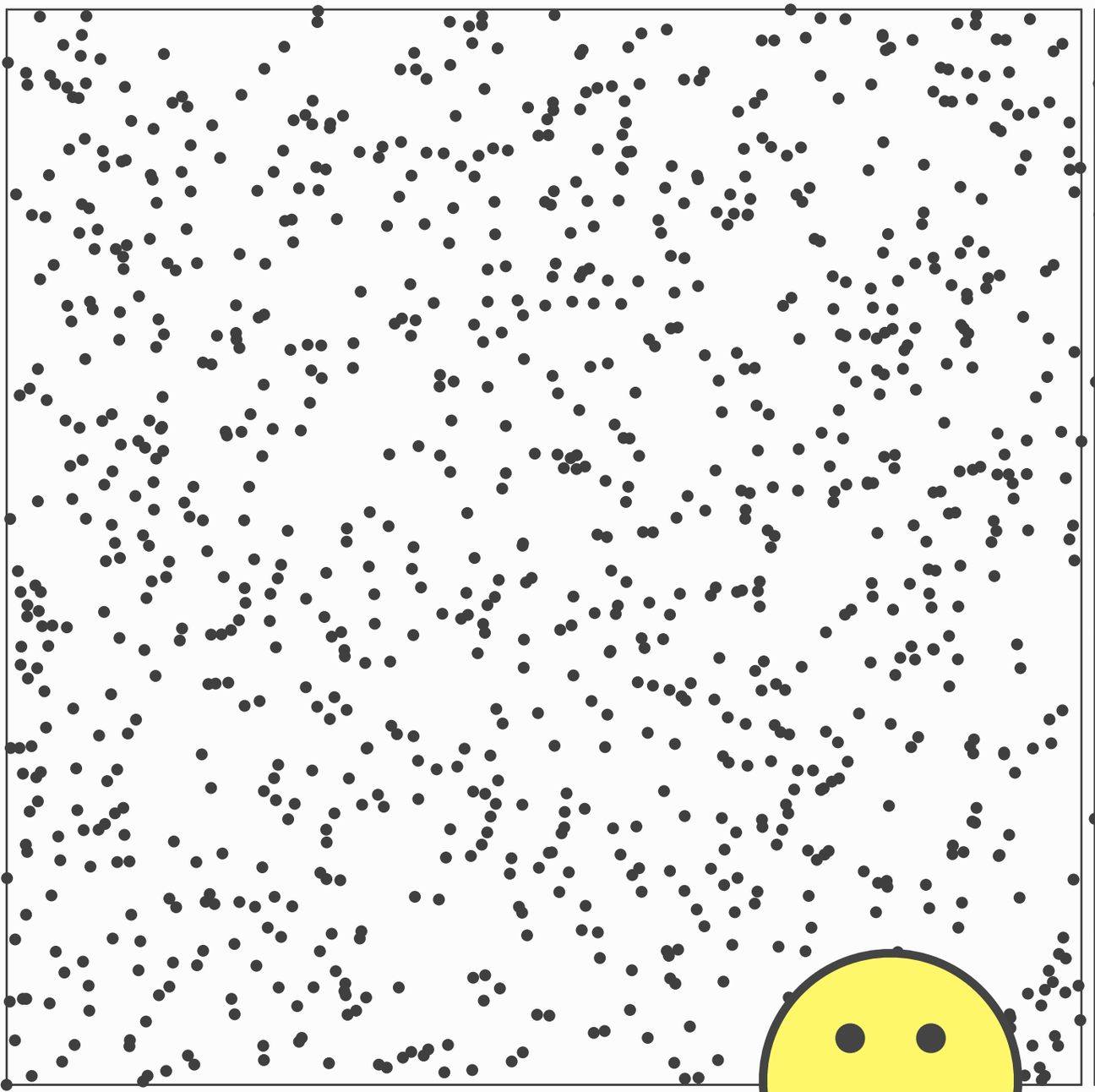


Power Spectrum

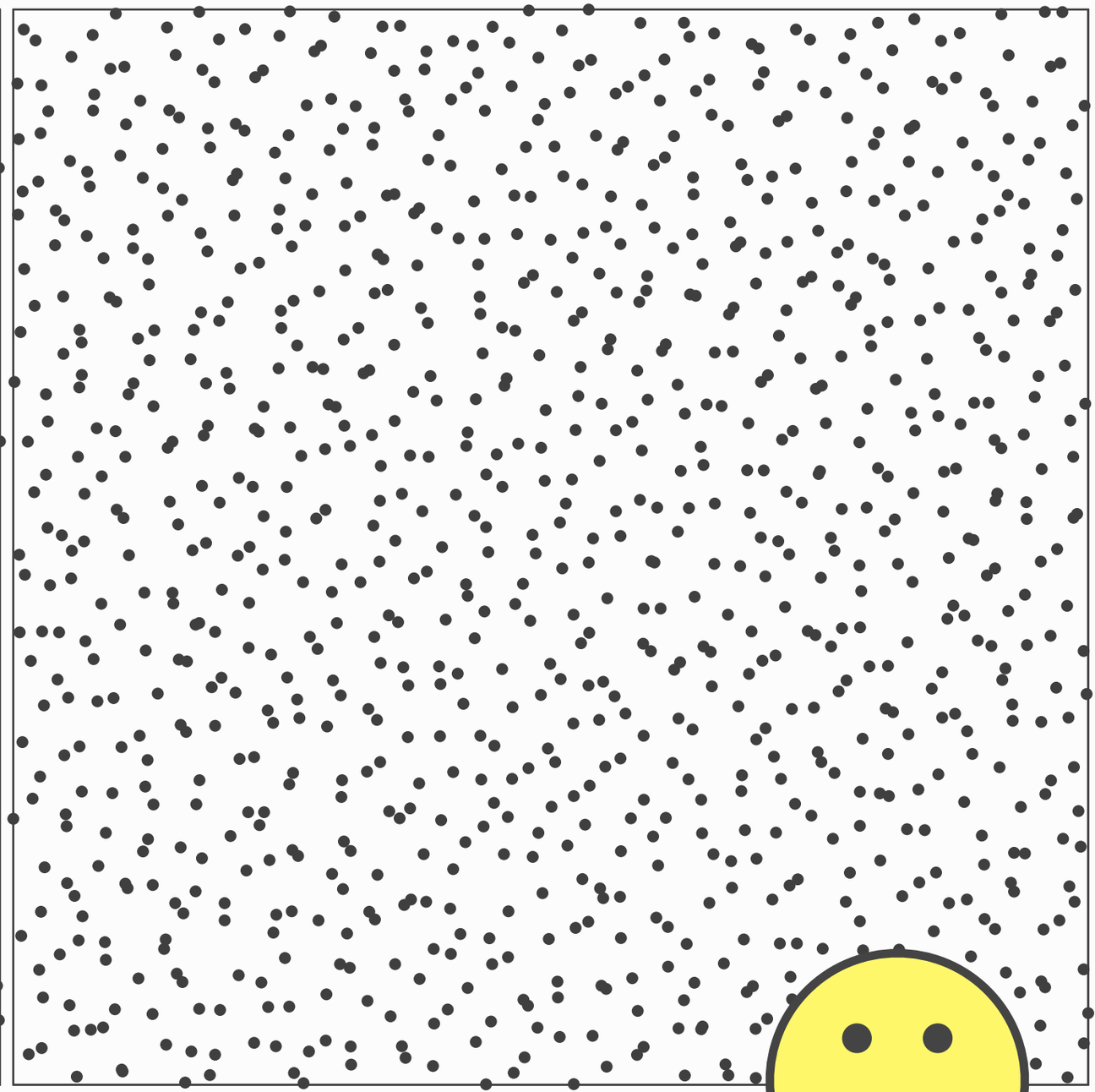


THE **FINAL** WORDS

CONCLUSION

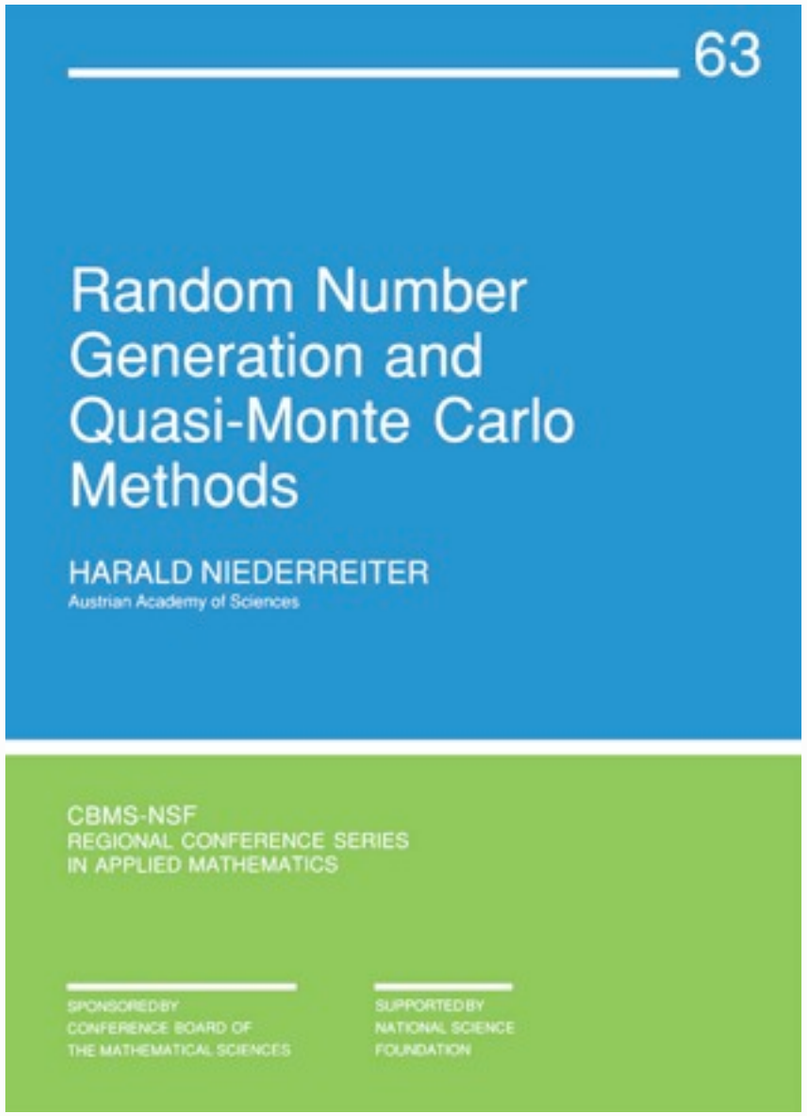


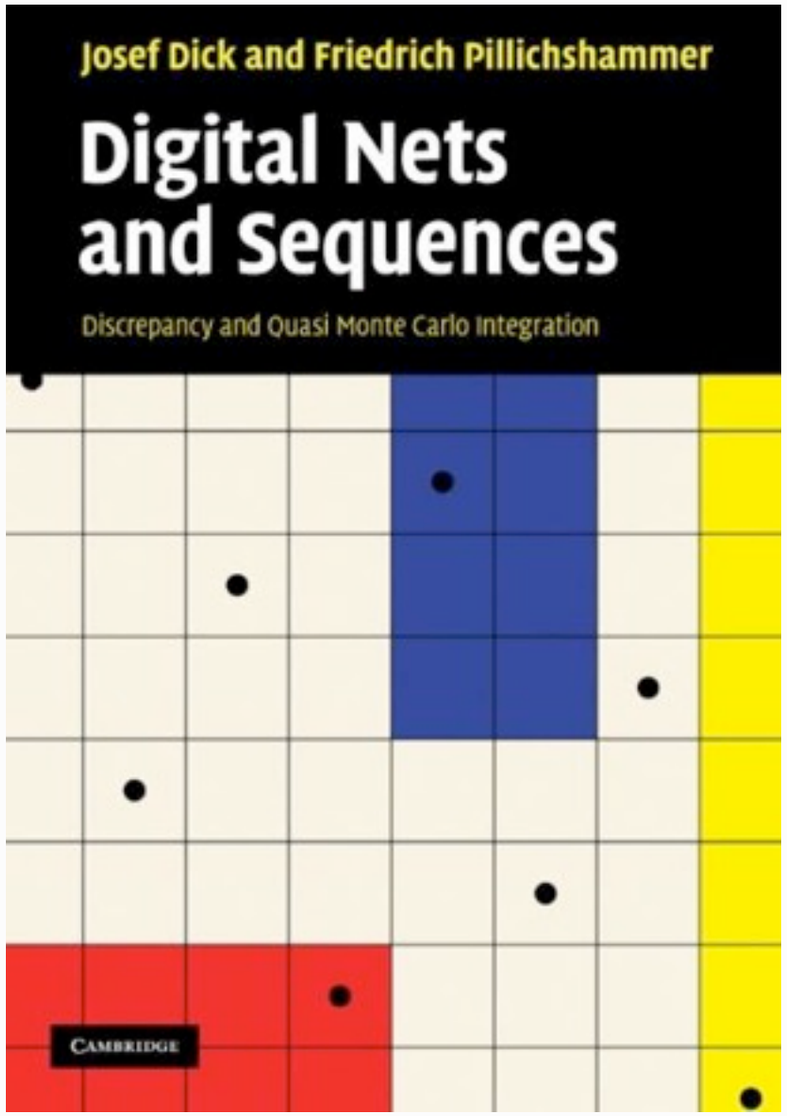
Random

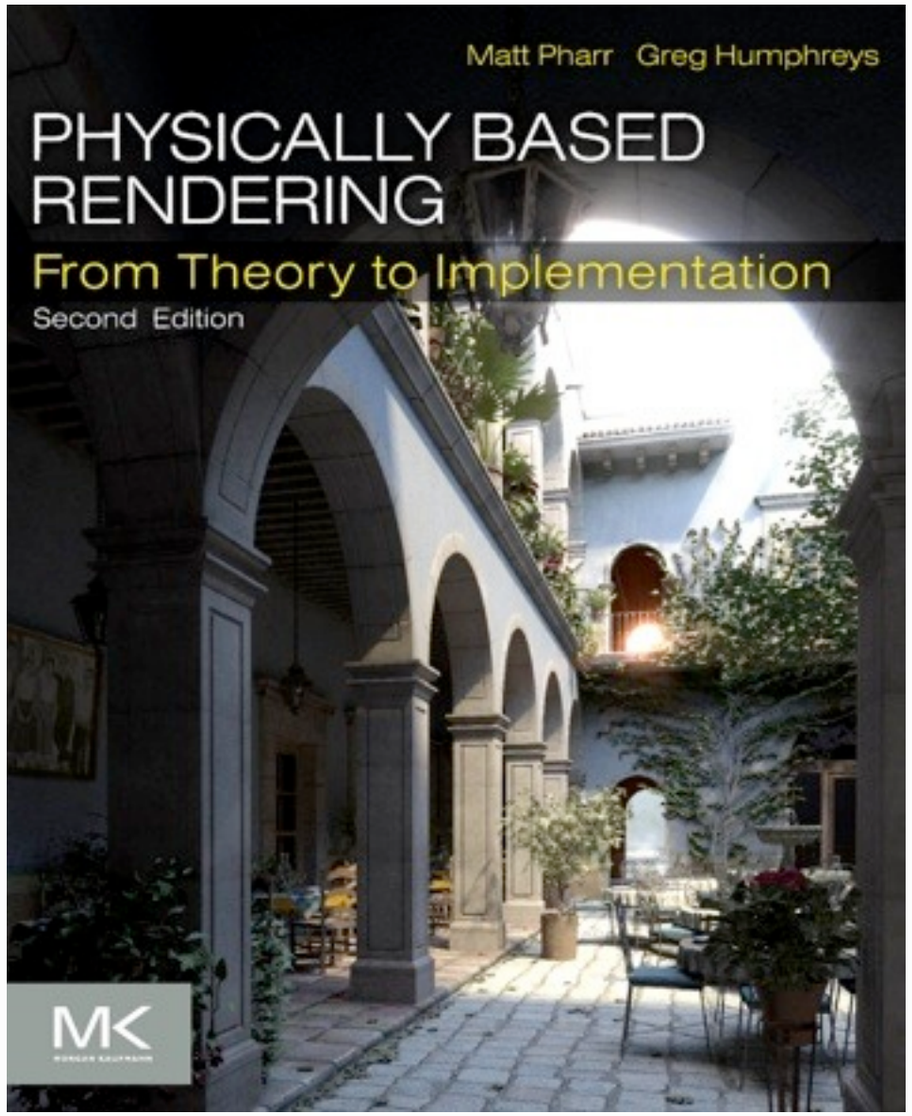


Quasi-random

SUGGESTED **BOOKS**







THANK YOU
FOR NOT USING *random()*

Colas Schretter

colas.schretter@gmail.com

http://homepages.ulb.ac.be/~cschrett/