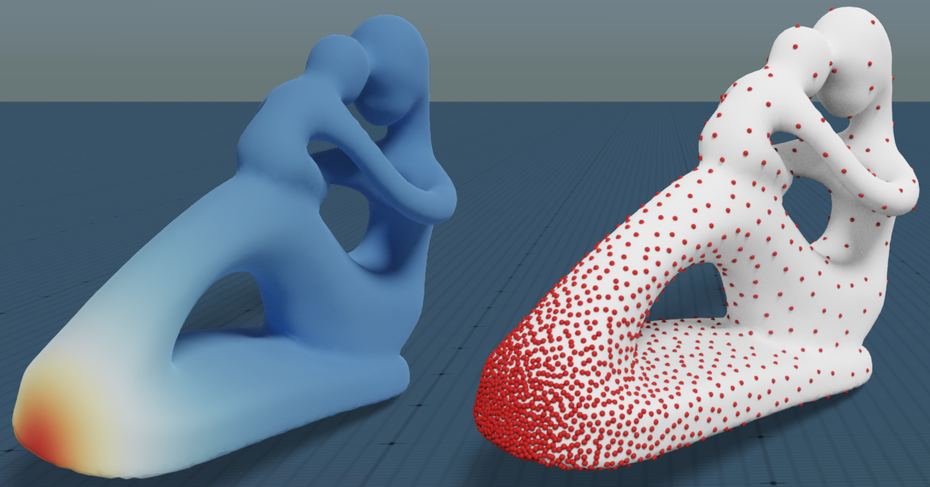
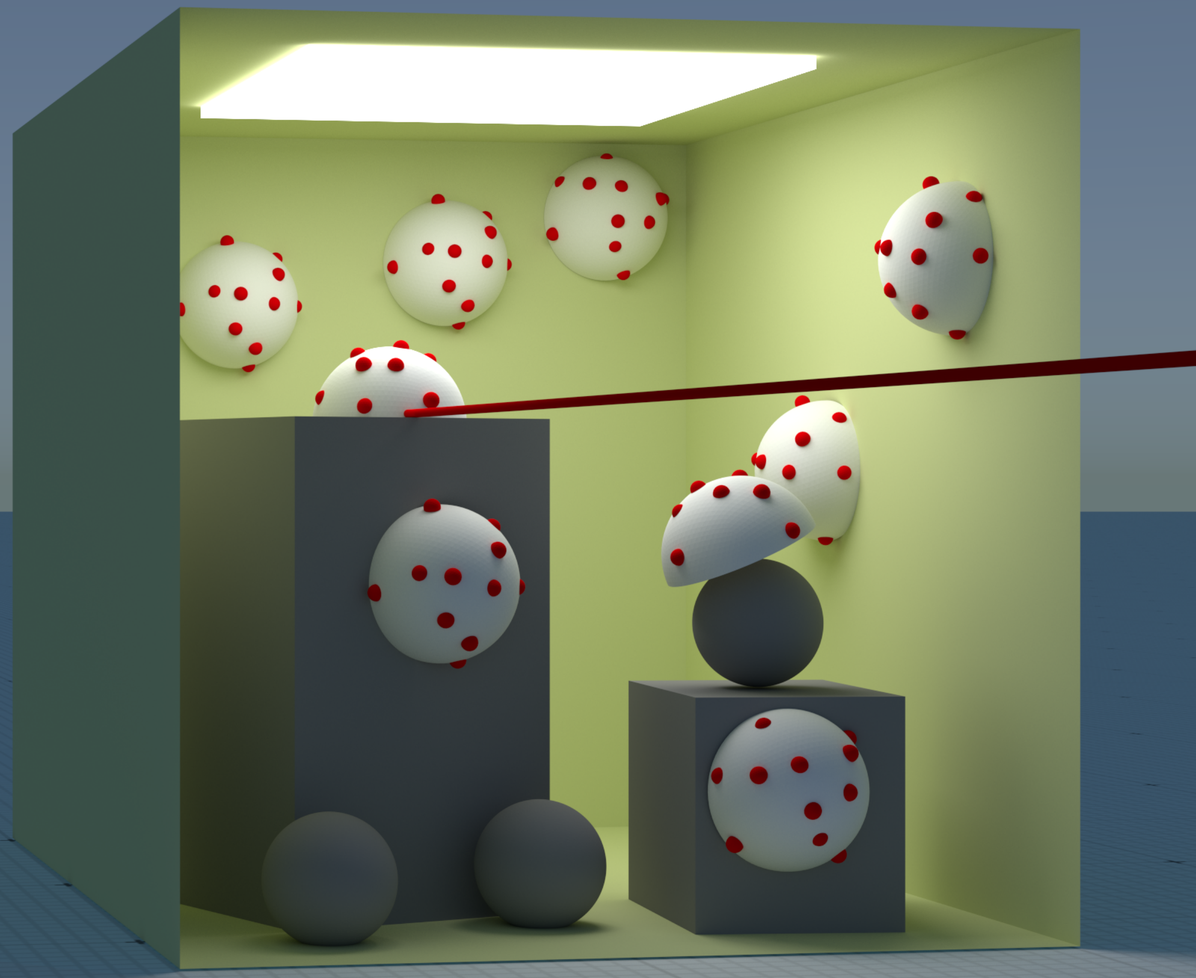
Échantillonnage de points en informatique graphique et transport optimal



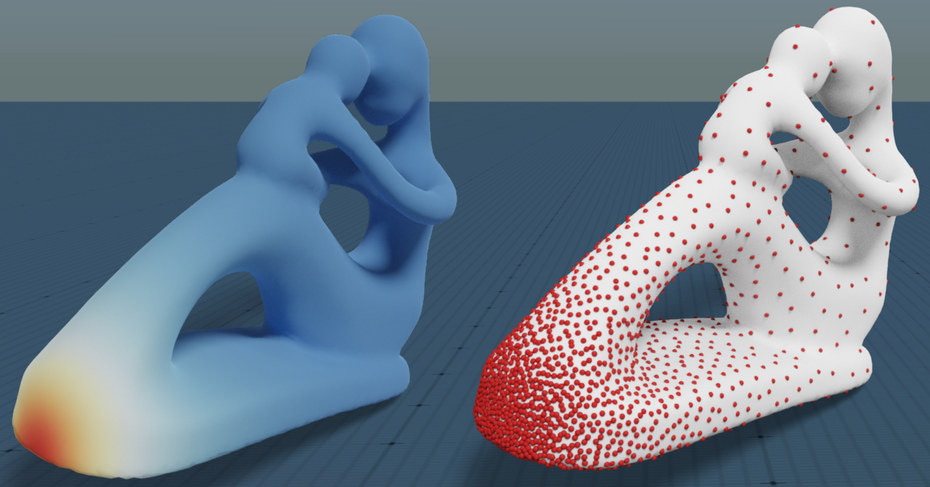


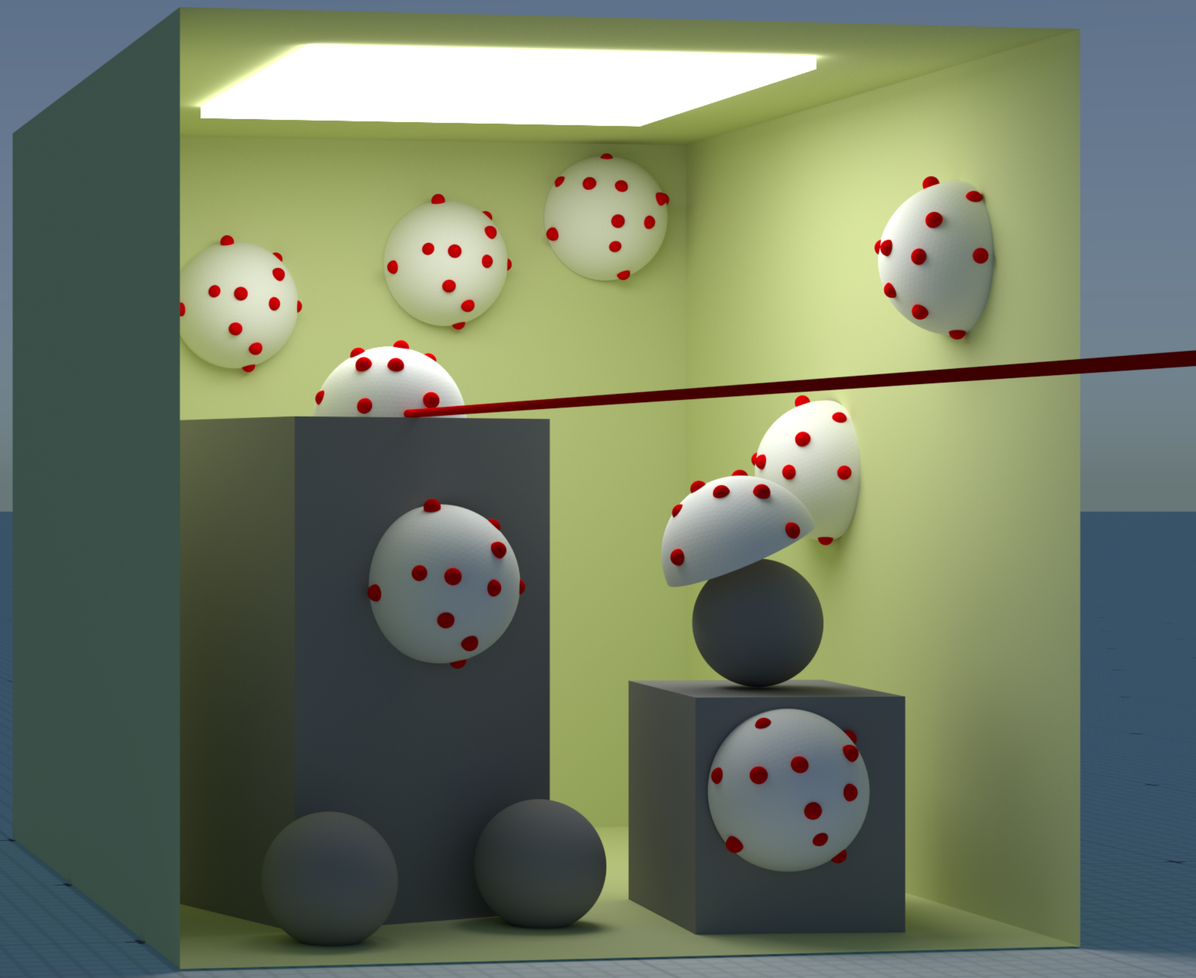
David Coeurjolly

Module Image

ECOLE DOCTORALE
LIRMM
UNIVERSITÉ DE MONTPELLIER

Échantillonnage de points en informatique graphique et transport optimal

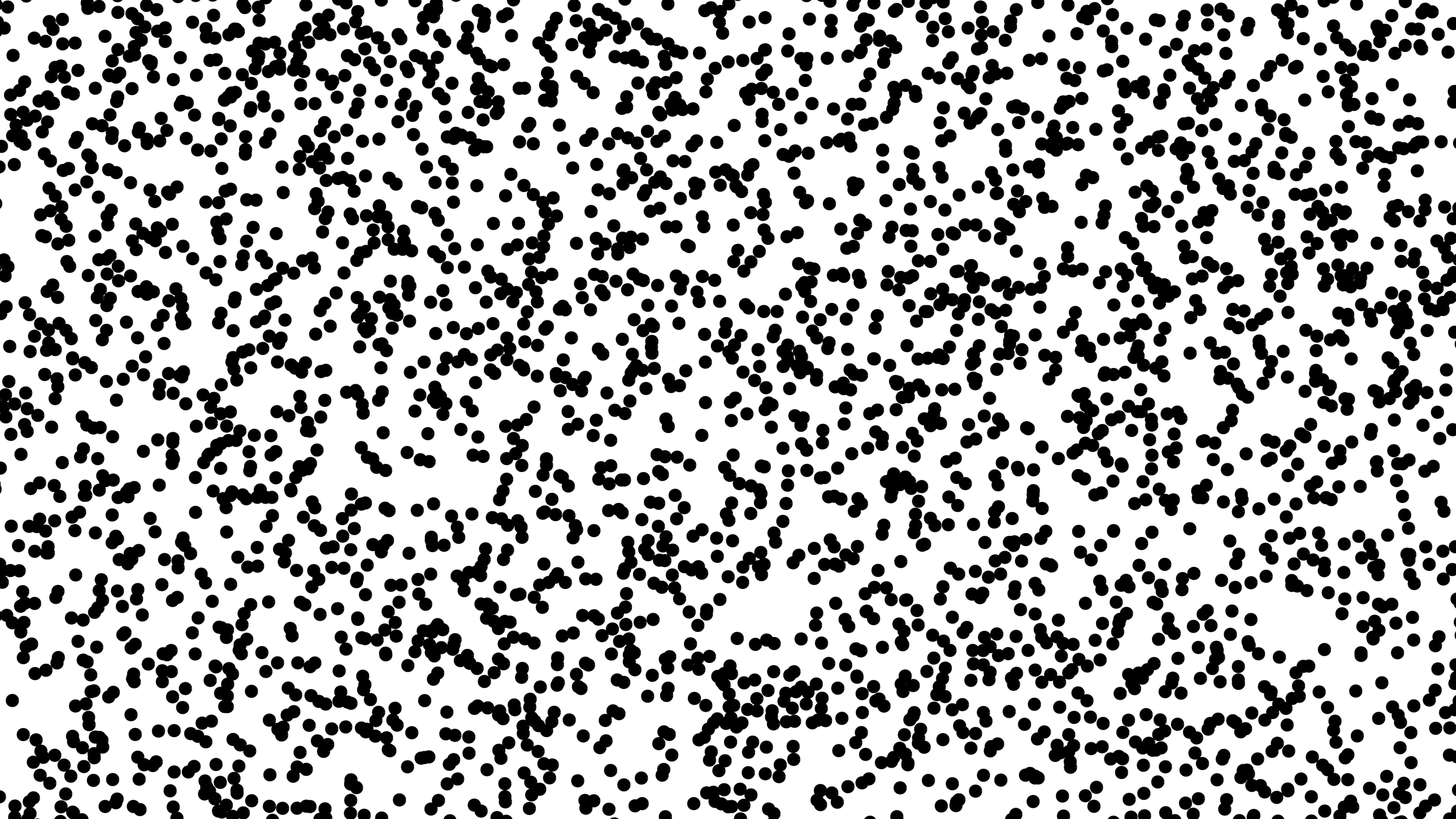


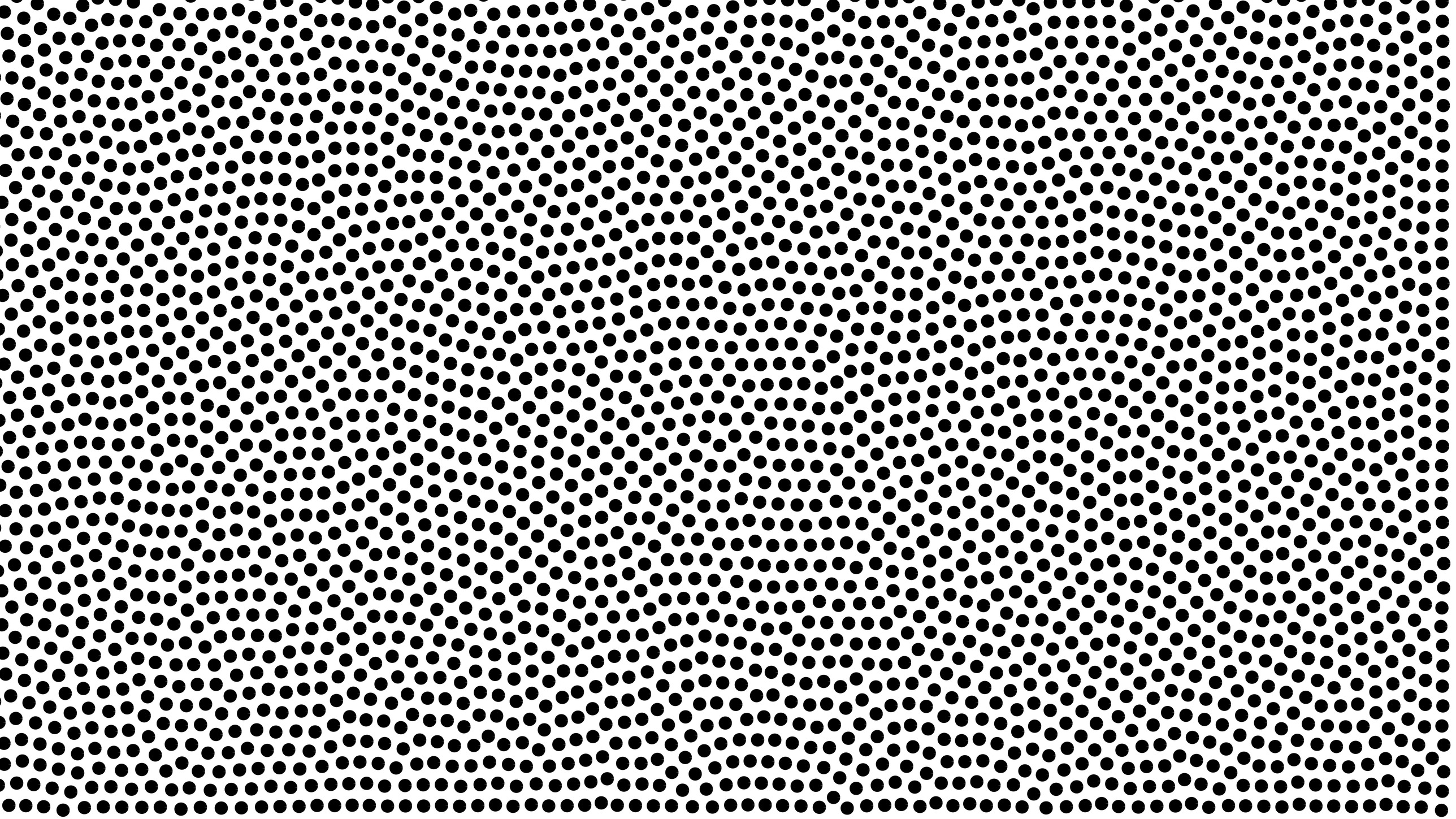


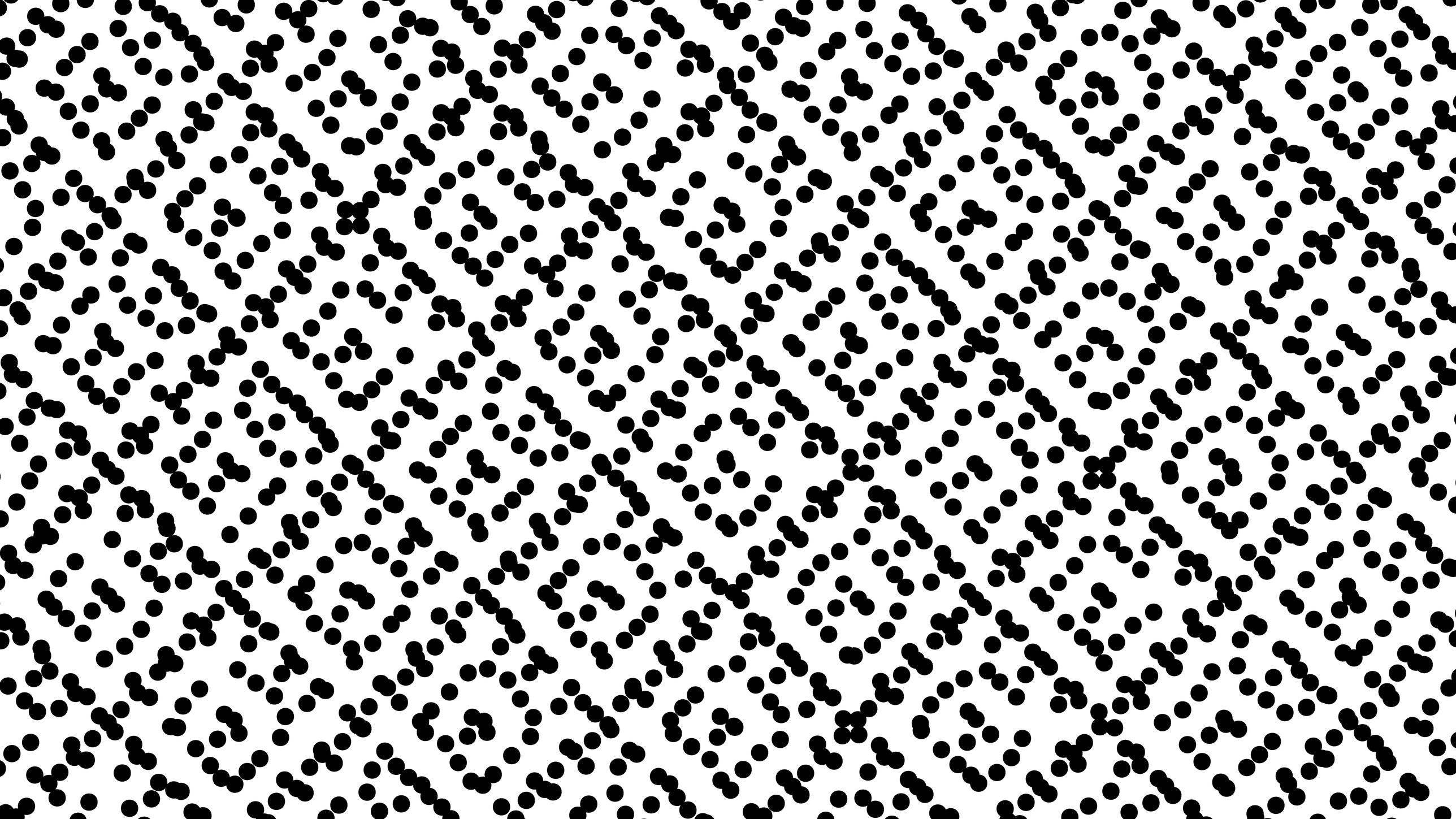
David Coeurjolly

Module Image

ECOLE DOCTORALE
LIRMM
UNIVERSITÉ DE MONTPELLIER



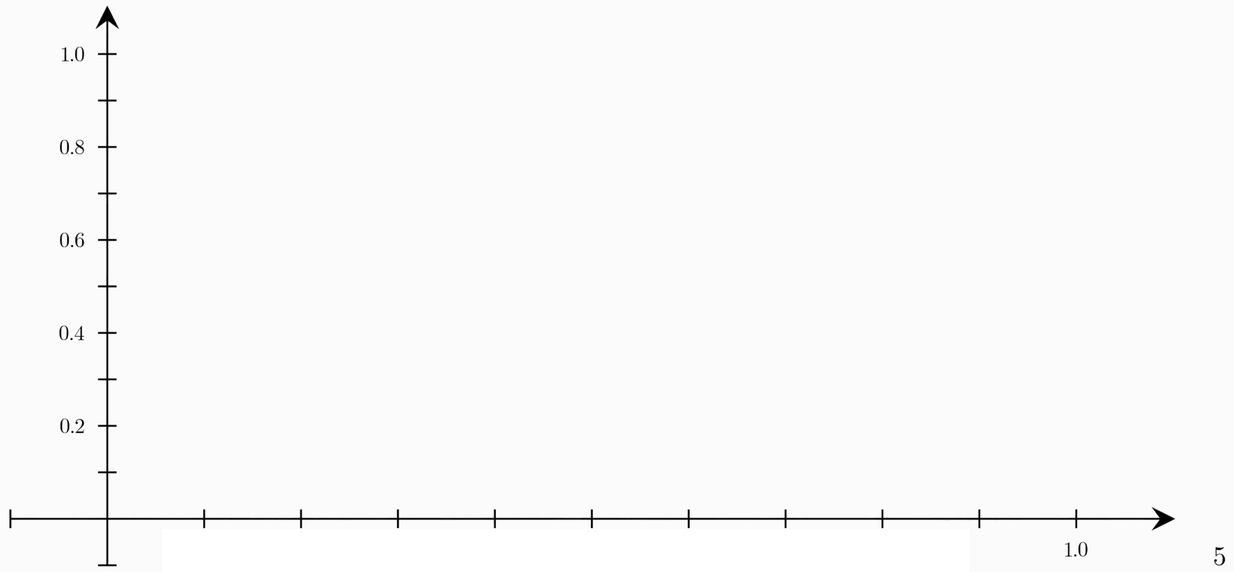


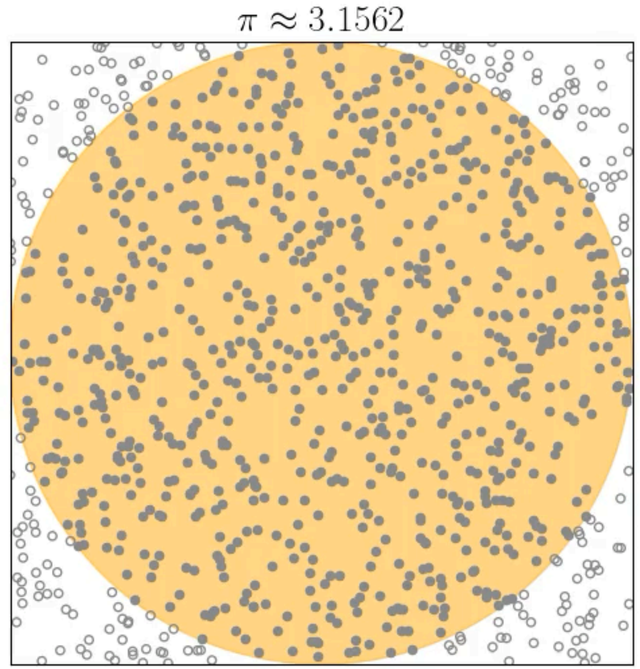


Monte Carlo Integration in Rendering

$$\int_\Omega f dx \approx \frac{1}{n} \sum_i f(x_i)$$

Quasi-Monte Carlo



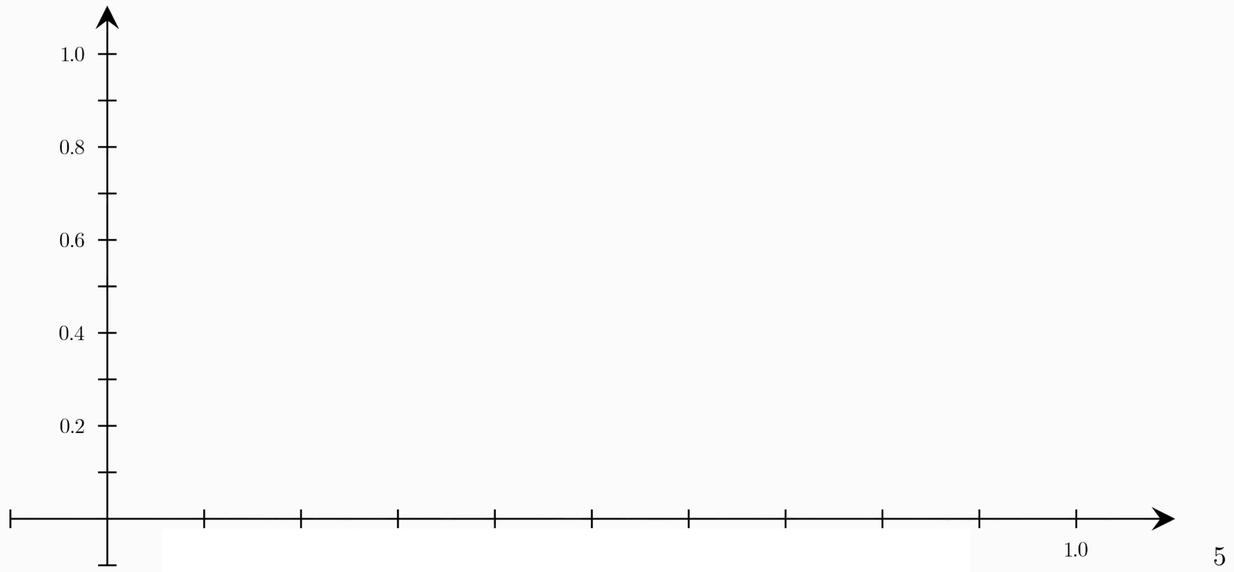


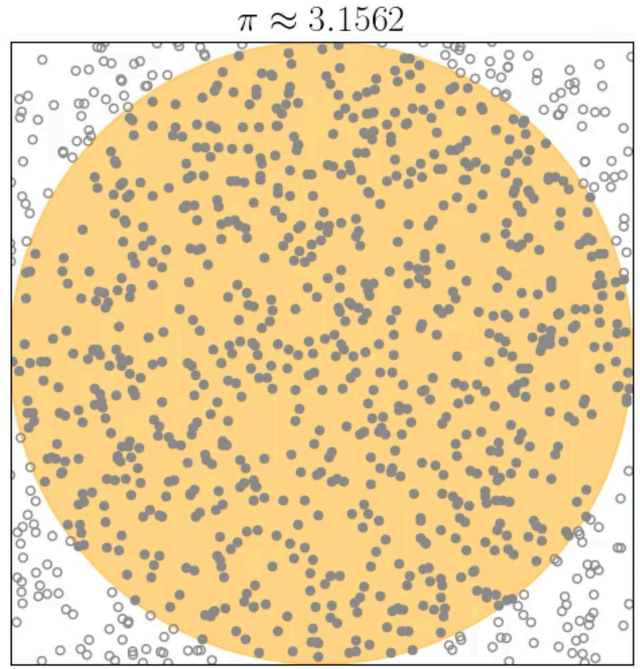
Whitenoise

Monte Carlo Integration in Rendering

$$\int_\Omega f dx \approx \frac{1}{n} \sum_i f(x_i)$$

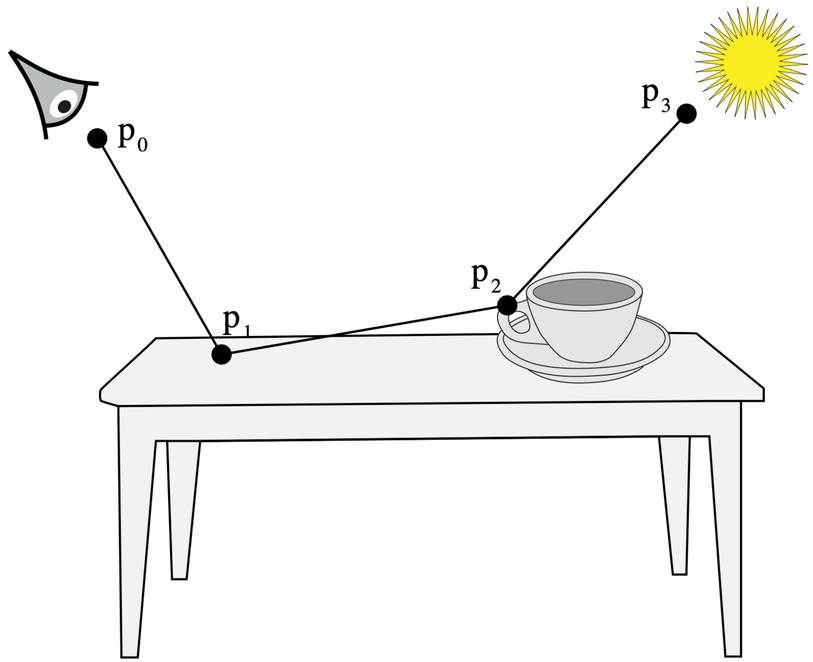
Quasi-Monte Carlo



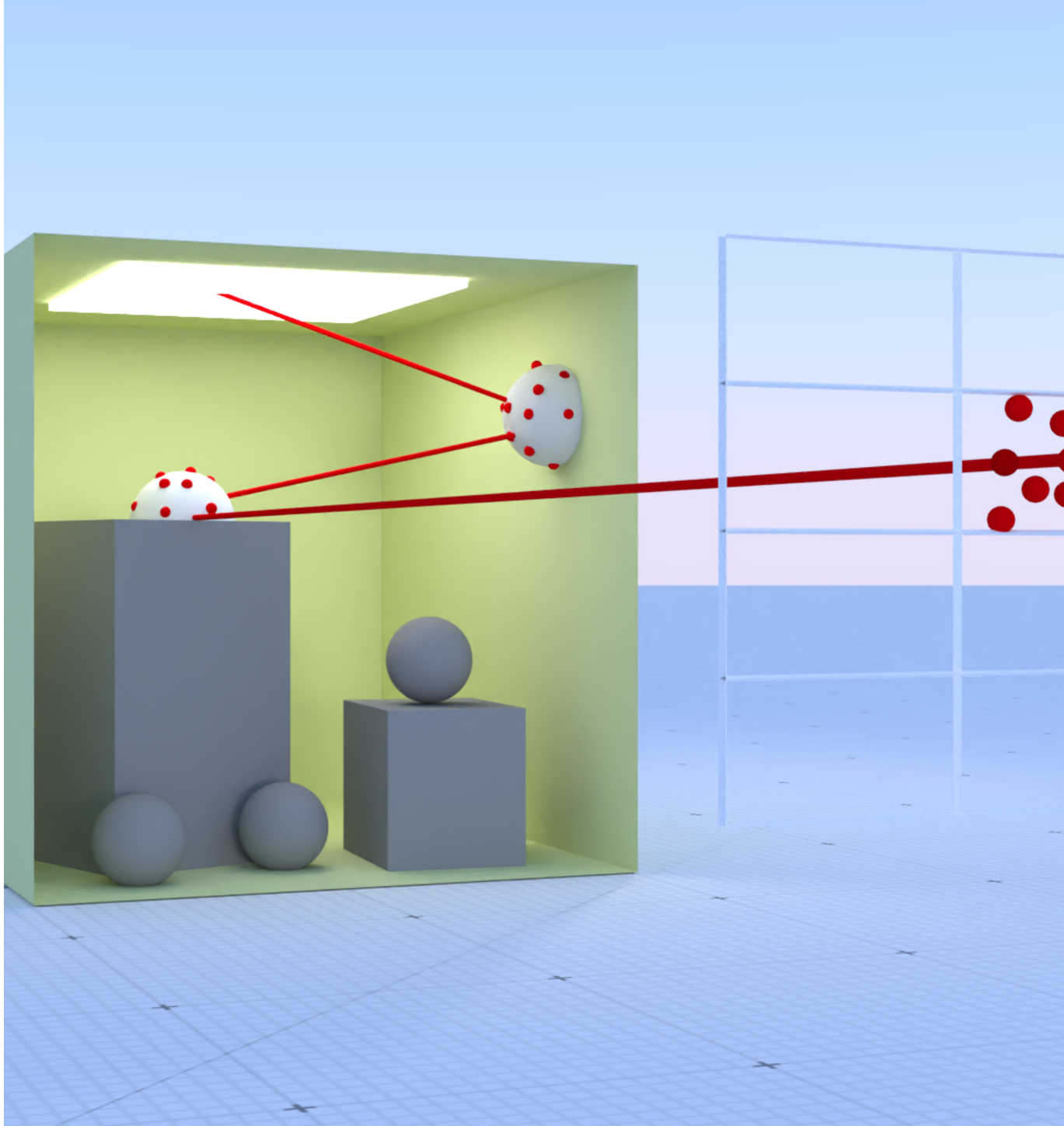


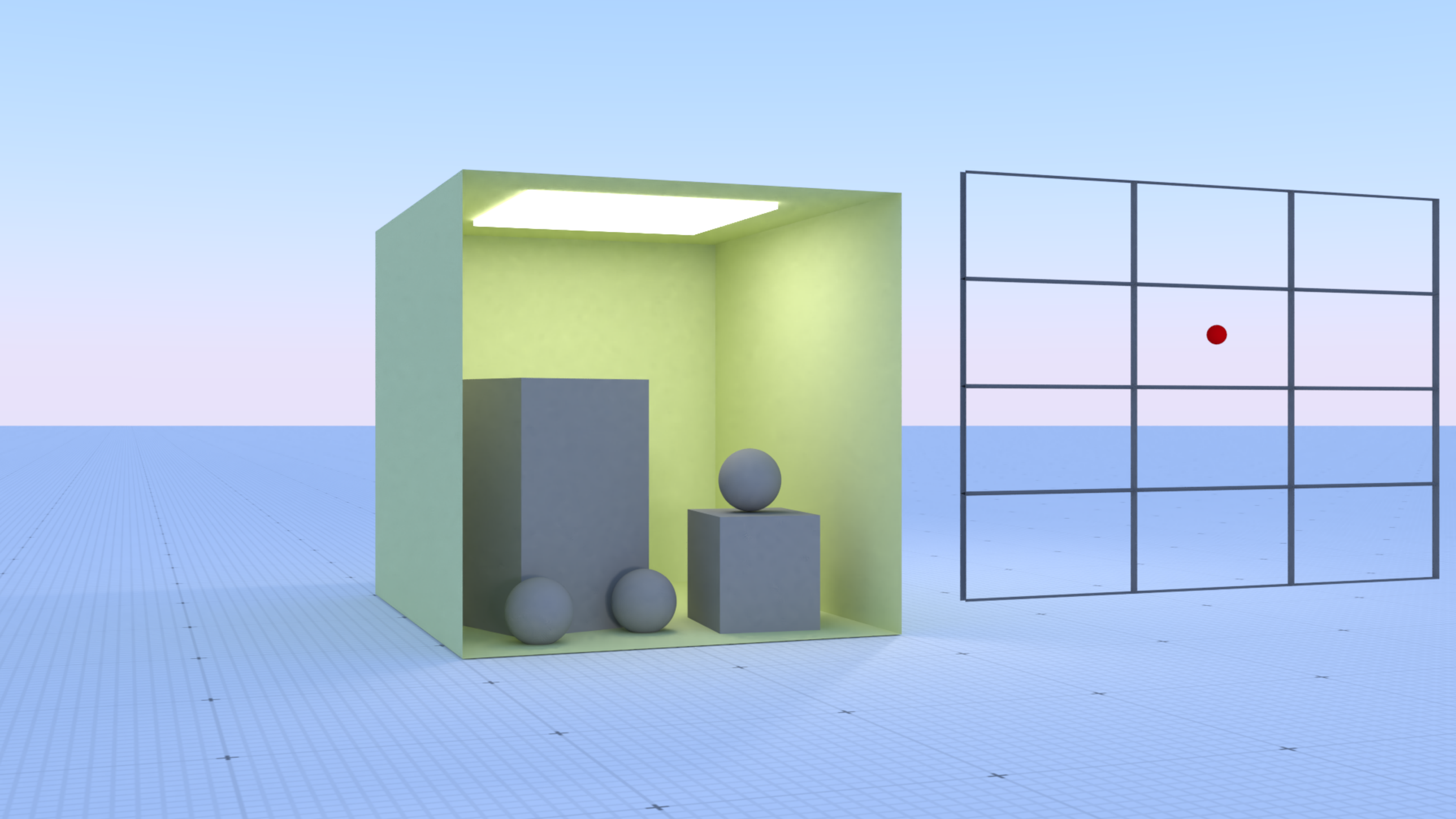
Whitenoise

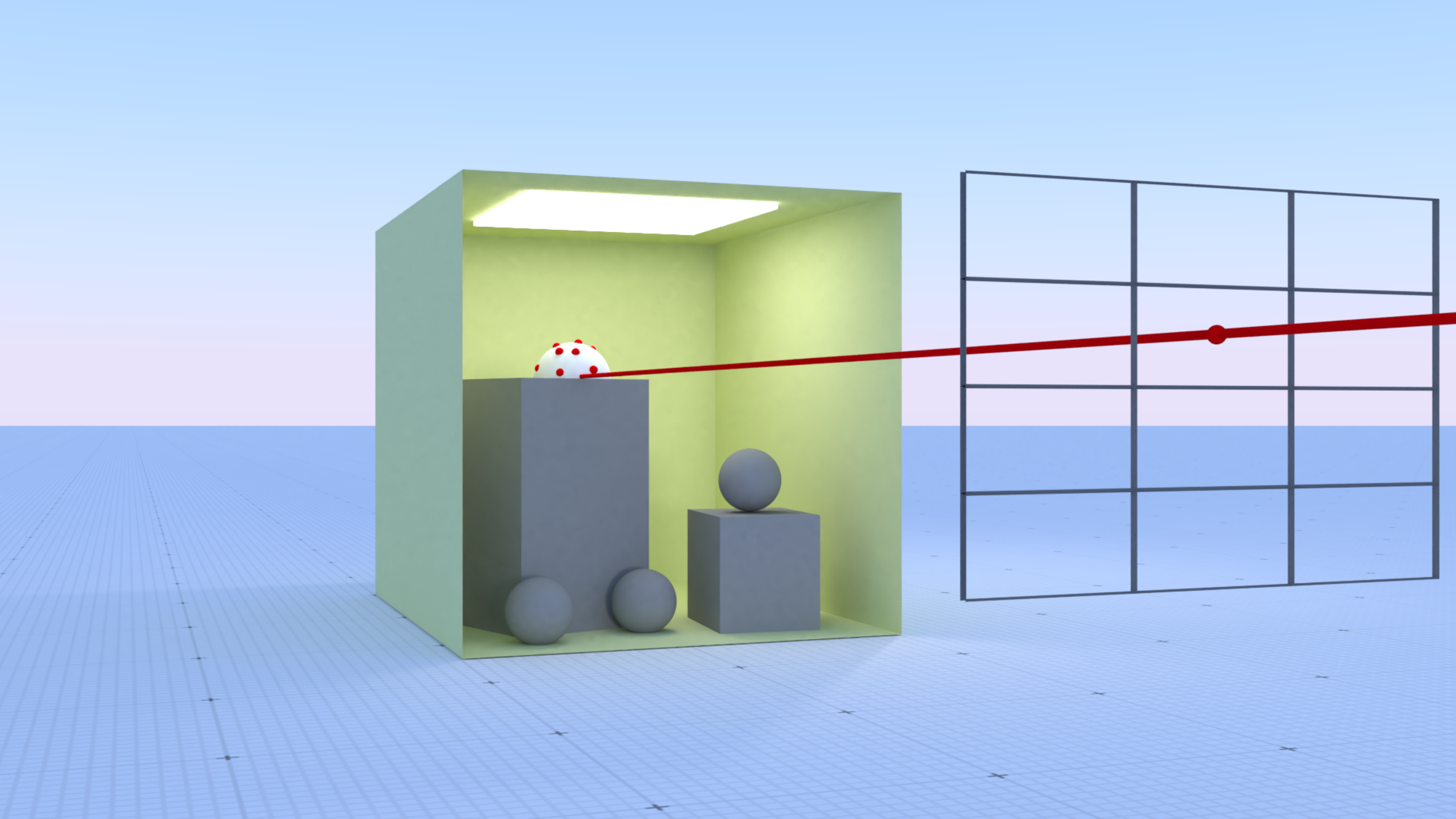
Path tracing

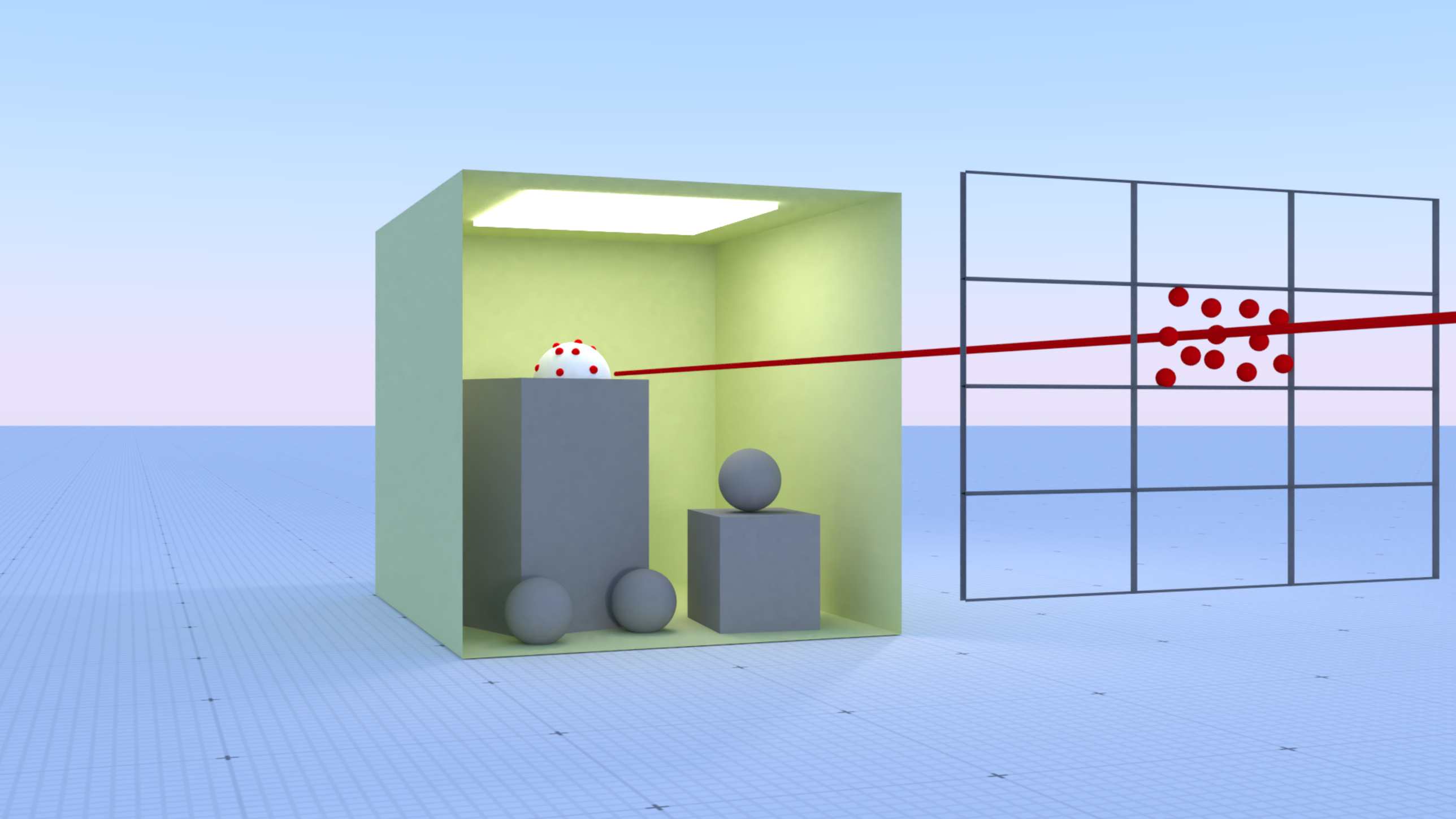


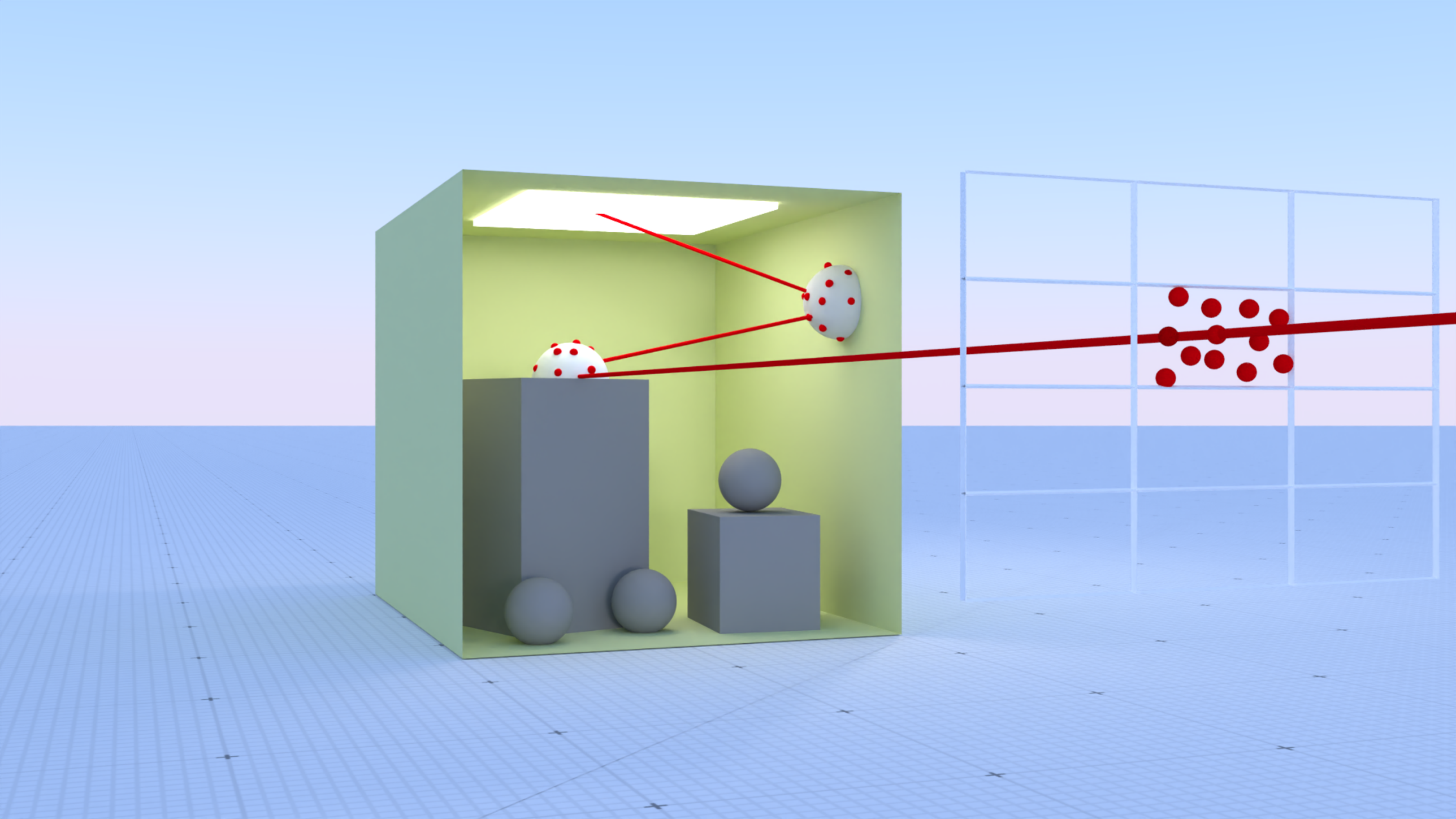
$$L_o(p,\omega_o) = L_e(p,\omega_o) + \int_{\mathbb{S}^2} f(p,\omega_o,\omega_i) L_i(p,\omega_i) |\,\omega_i \cdot \vec{n}\,| \, d\omega_i$$

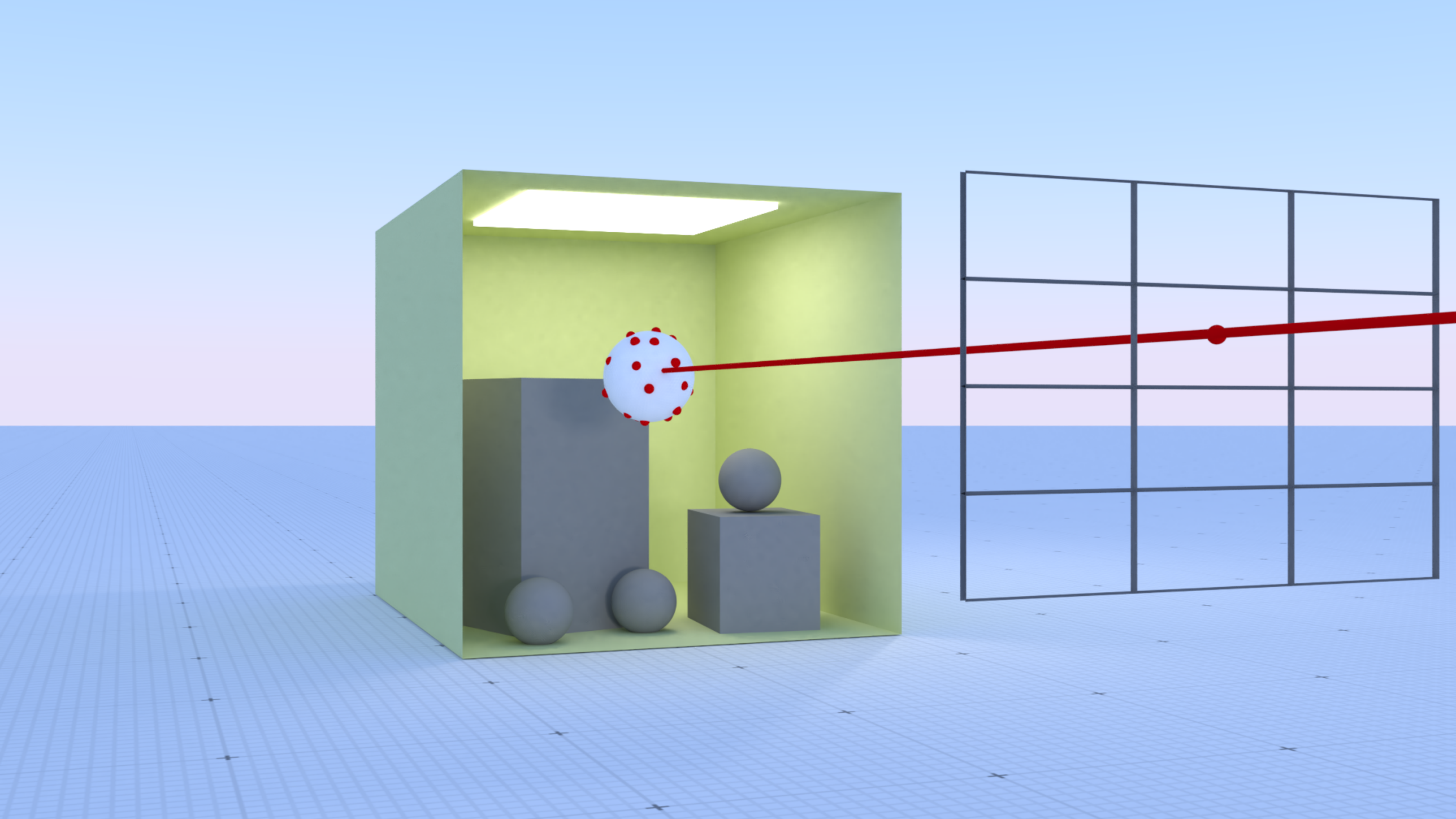


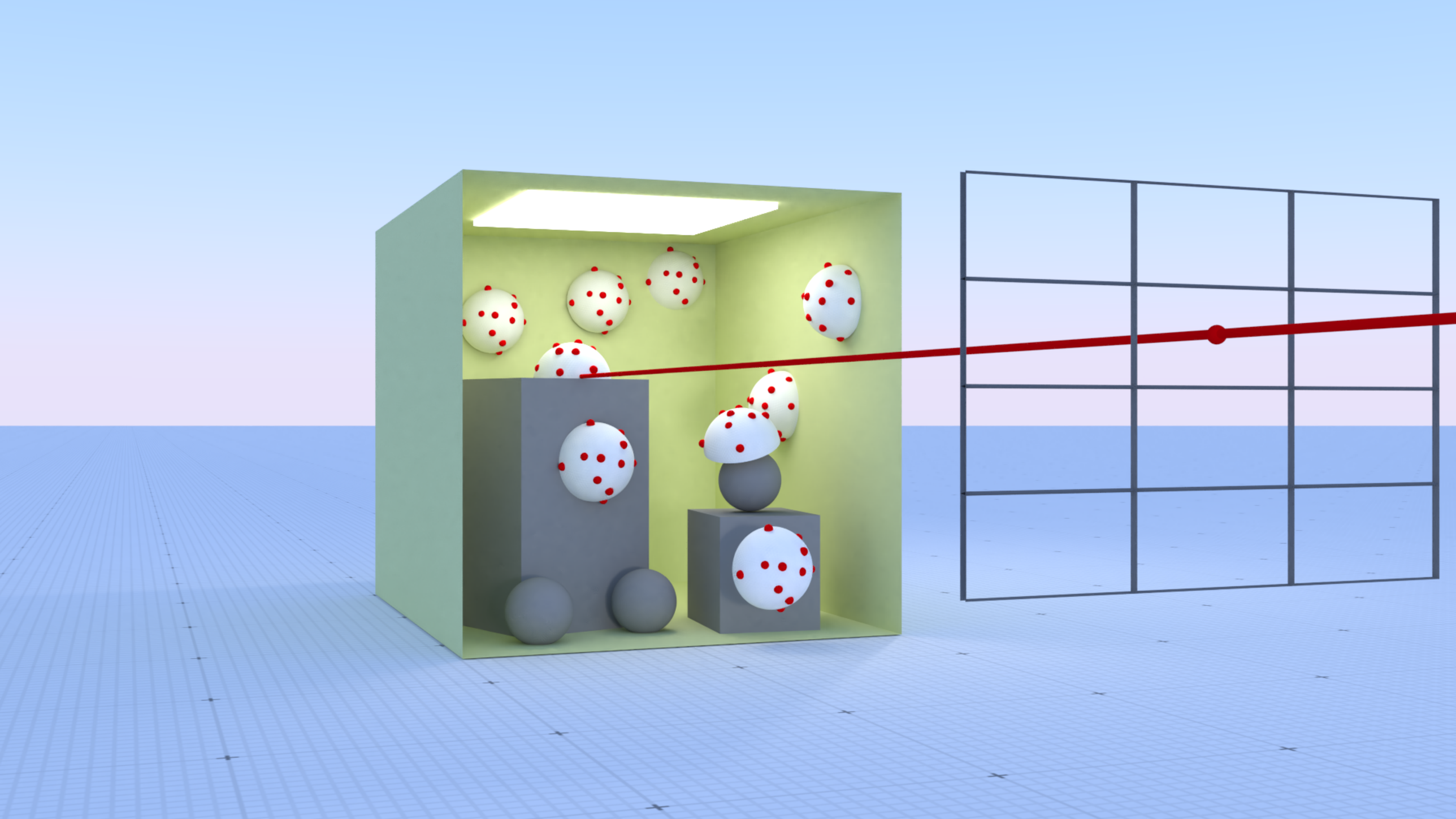













1 spp


2 spp


4 spp


8 spp


16 spp


32 spp

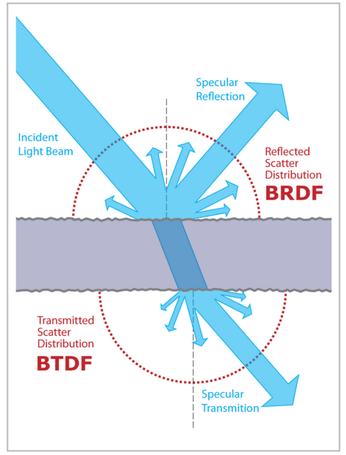

64 spp

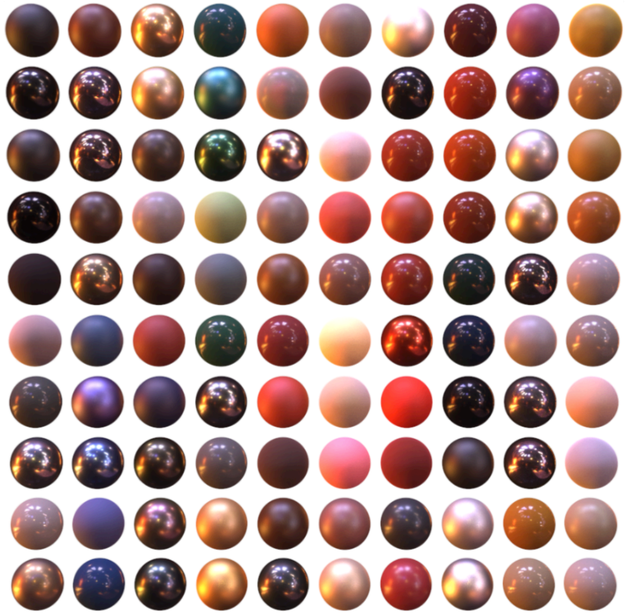

128 spp

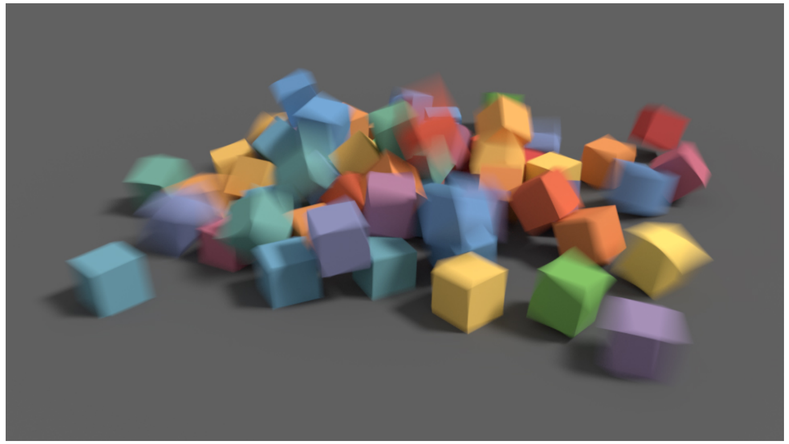

256 spp

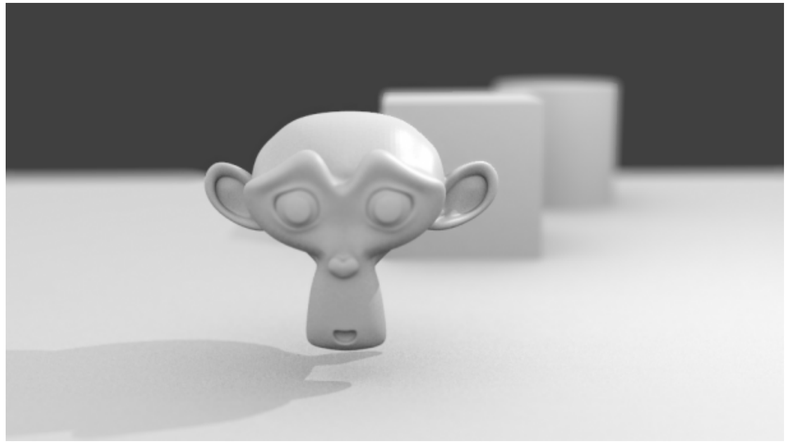

∞ spp

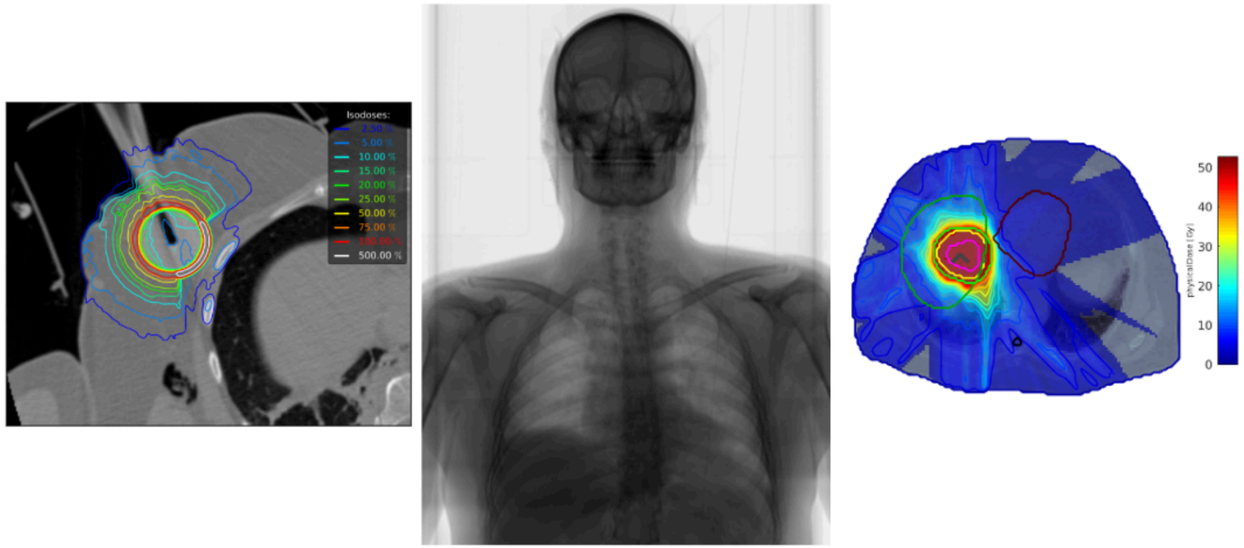
Material and physically based rendering



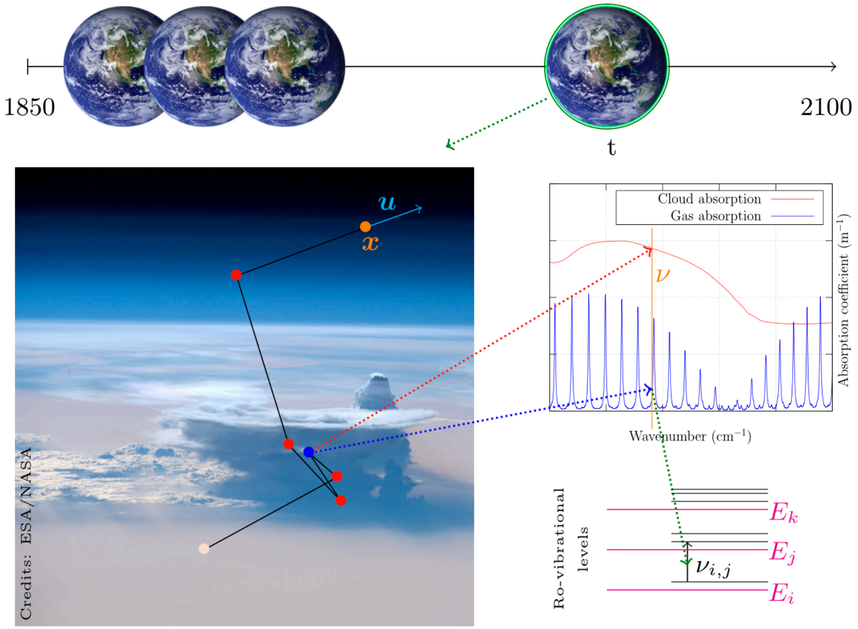








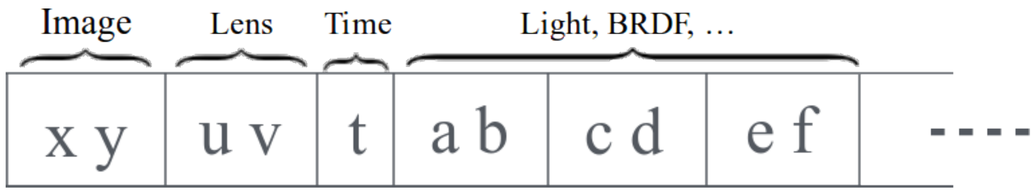
[ANR MOCAMED]



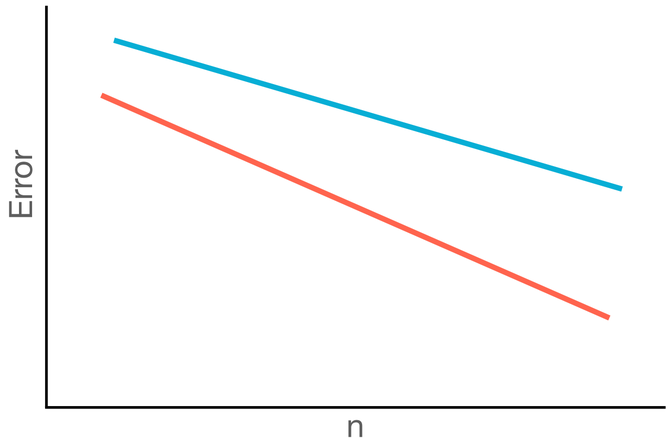
[Spectrally refined unbiased Monte Carlo estimate of the Earth's global radiative cooling]

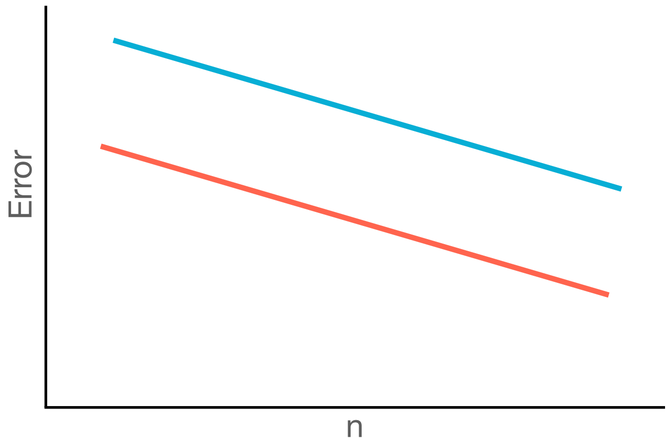
Wrap-up: MC Rendering Objectives

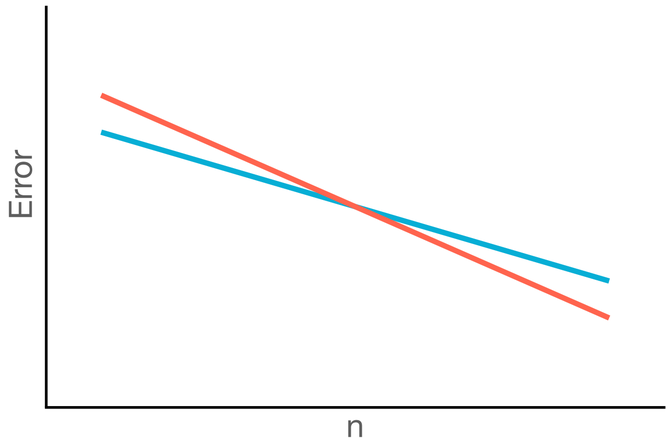
- Sampling in $[0,1)^s$ domains
- Stochastic samplers
- Asymptotic variance reduction in MC/QMC
- Low error on low sample counts
- Rather limited number of dimensions (~40)
- Fast, adaptive and progressive sampler
- Some projective subspaces of the $[0,1)^s$ may be specific



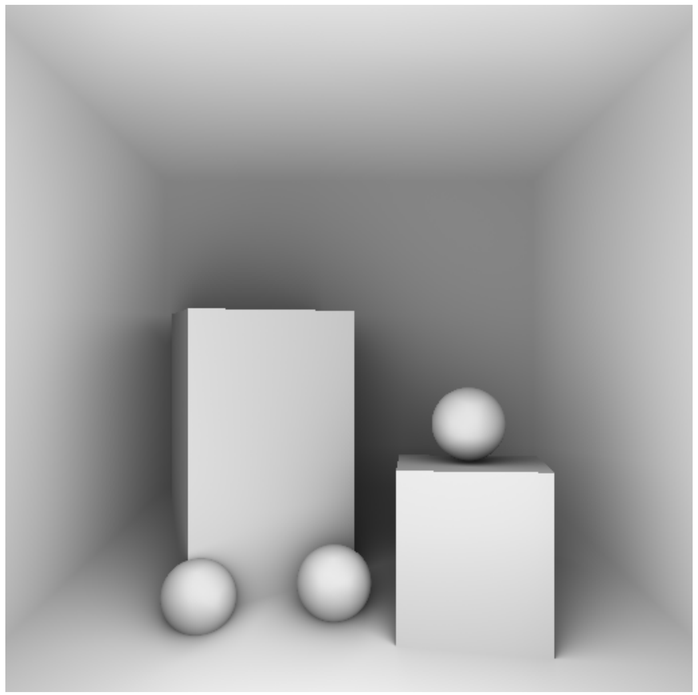
Convergence speed

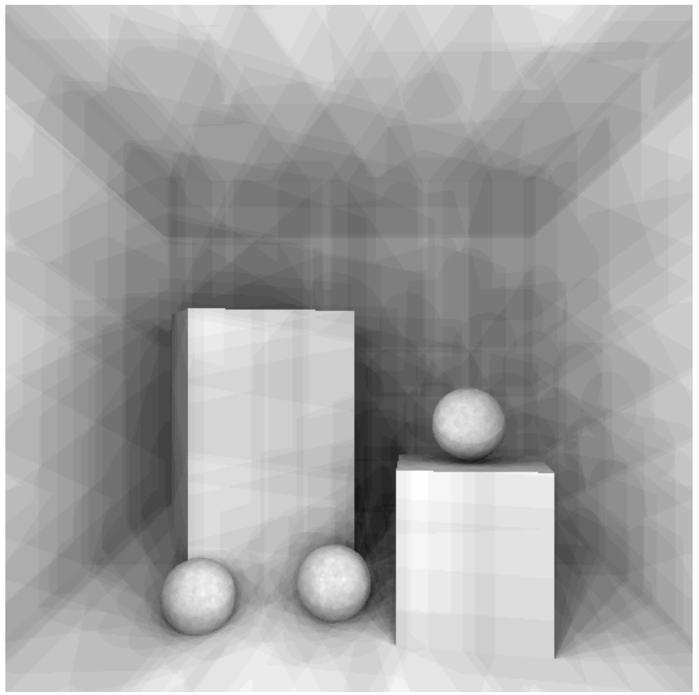


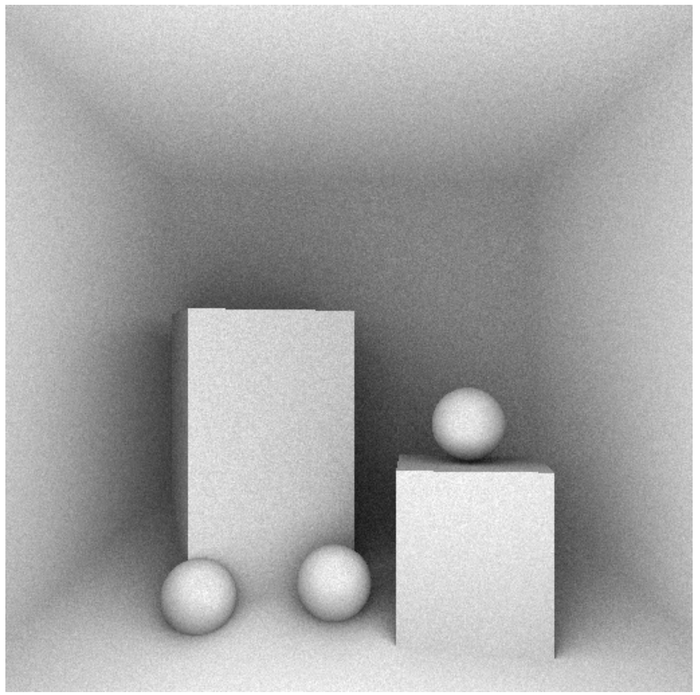




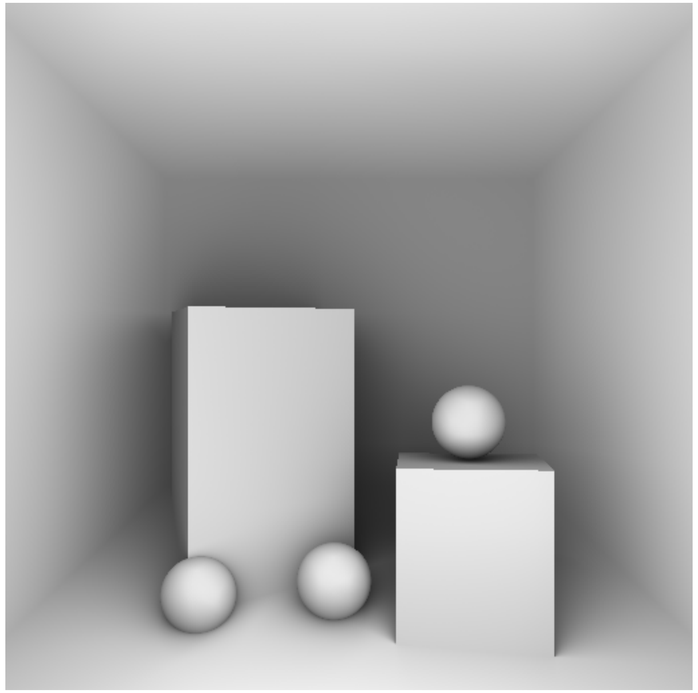
Noise vs aliasing

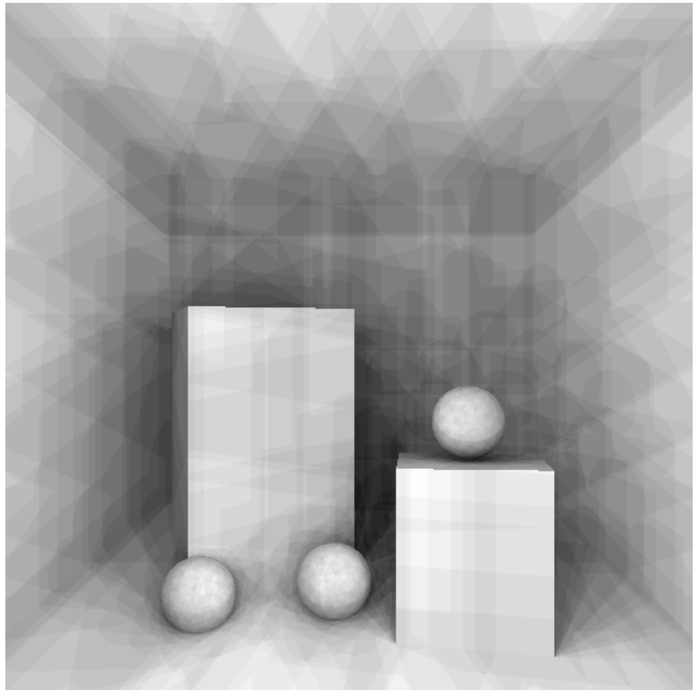


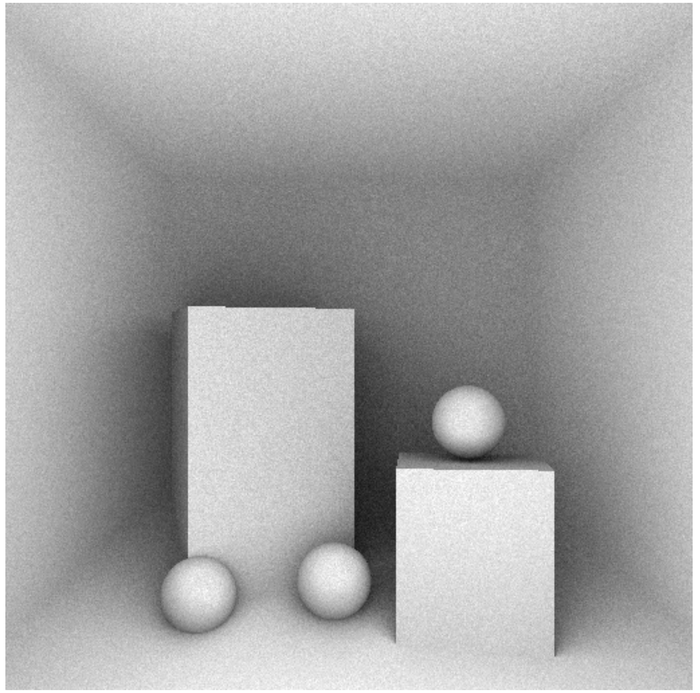




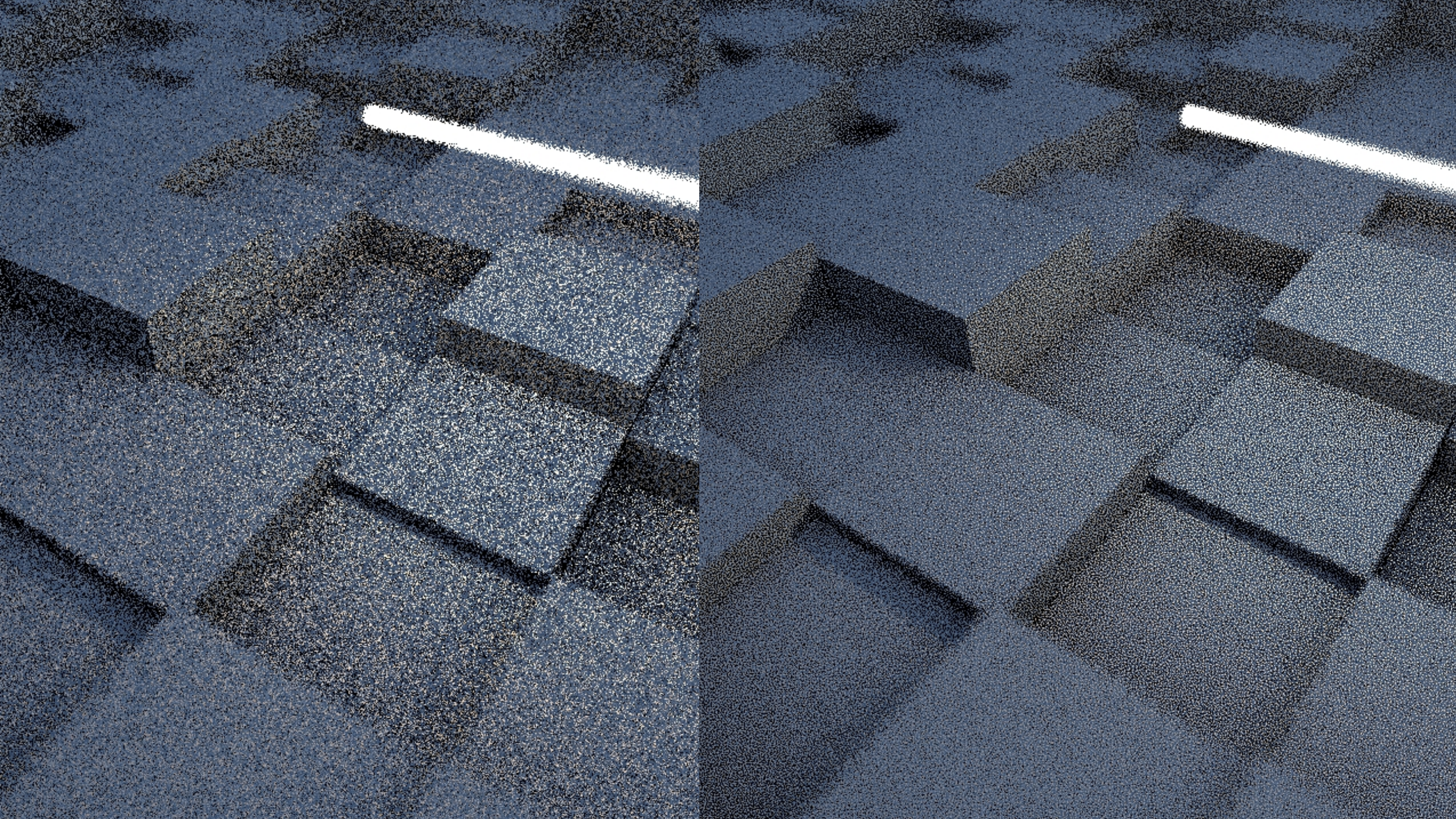
Noise vs aliasing







⇒ Stochastic point process or scrambling strategies

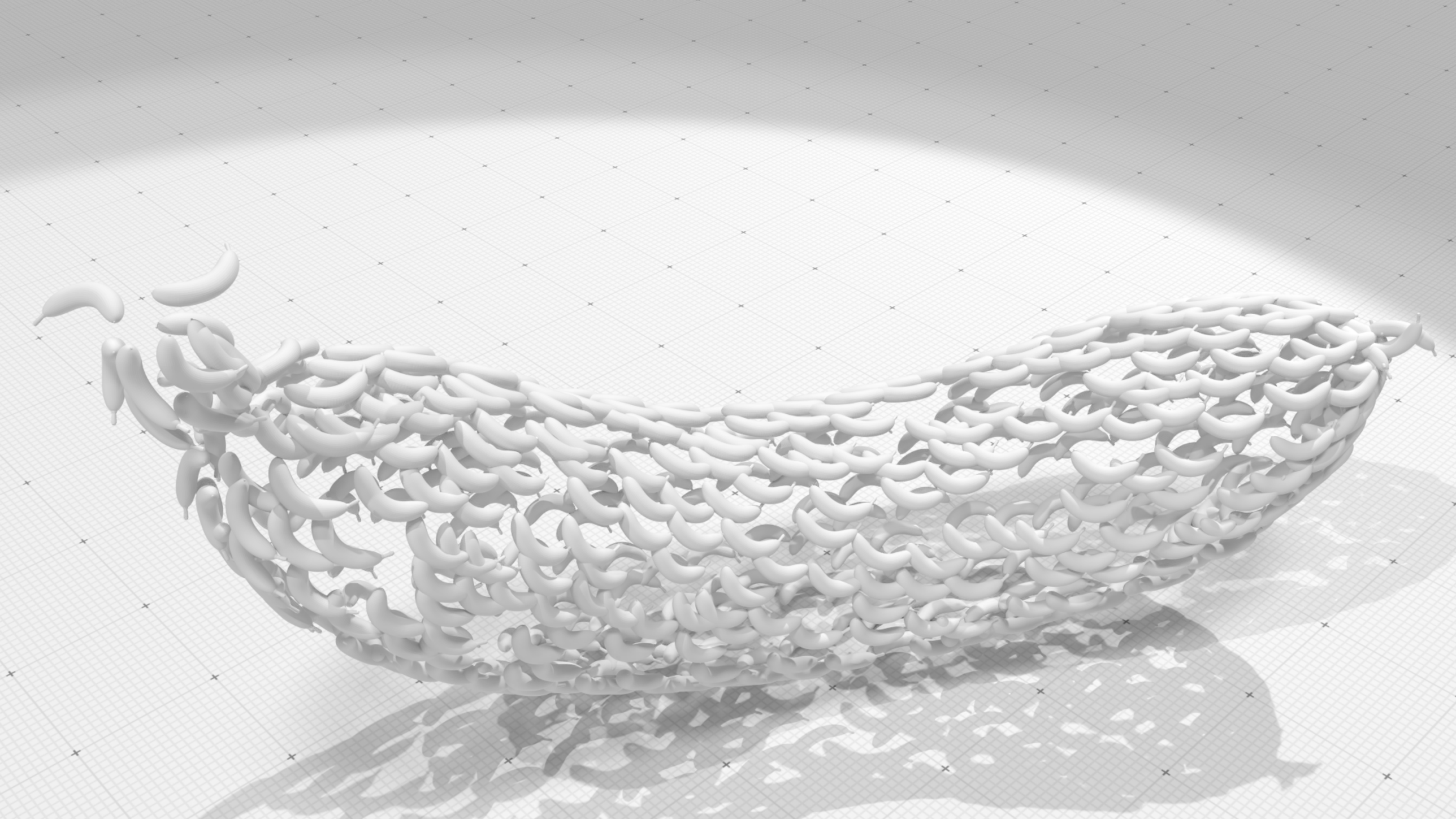


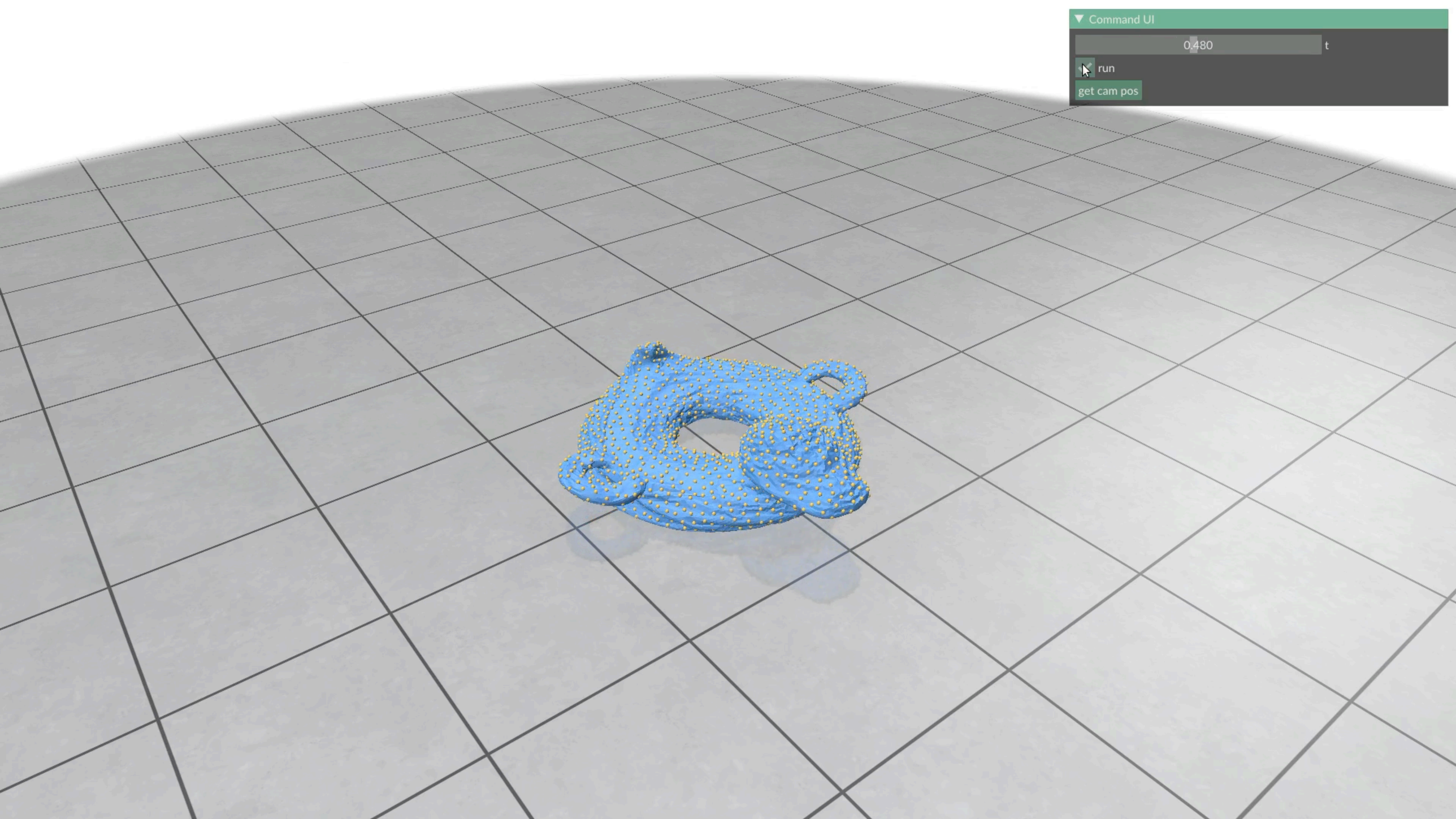
(Advanced techniques)

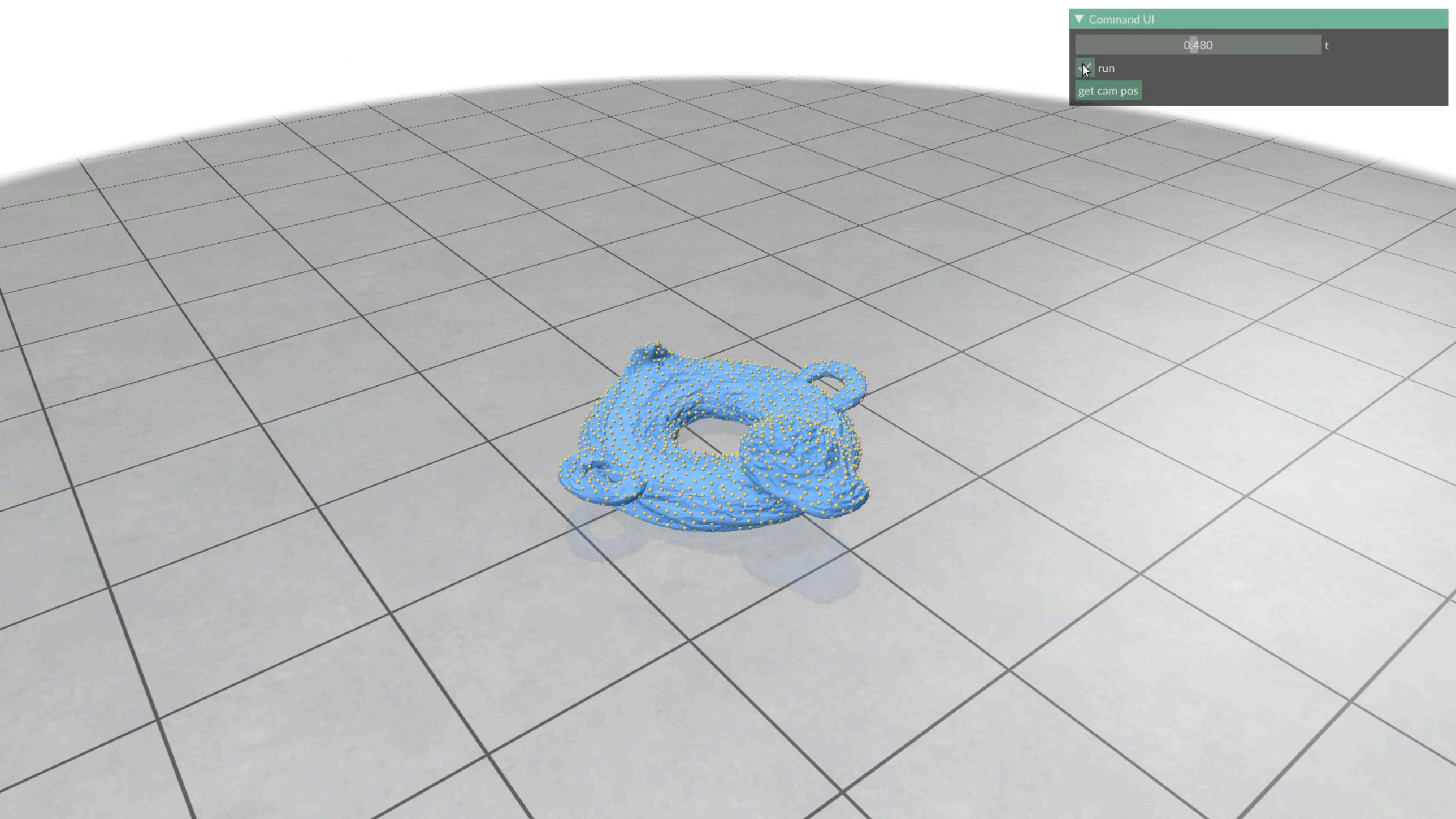
- Non-uniform densities
- Importance / Multiple importance sampling
- Control variates
- Metropolis sampling, Markov chain Monte Carlo…
- Path-reuse
- Gradient domain rendering
- Denoising / Reconstruction
- Screenspace error diffusion
- ...

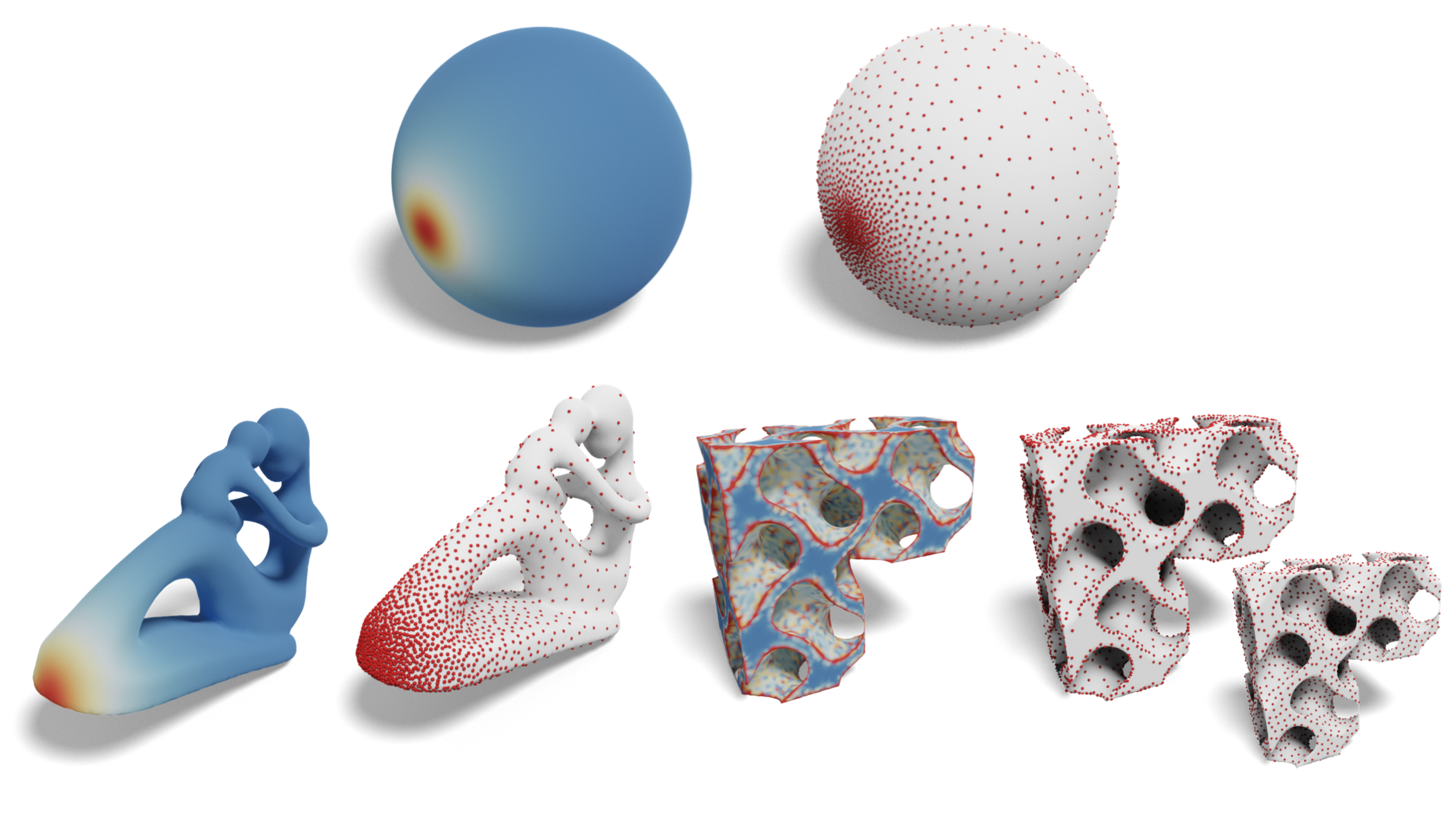
non-Euclidean Domains Sampling

non-Euclidean Domains Sampling

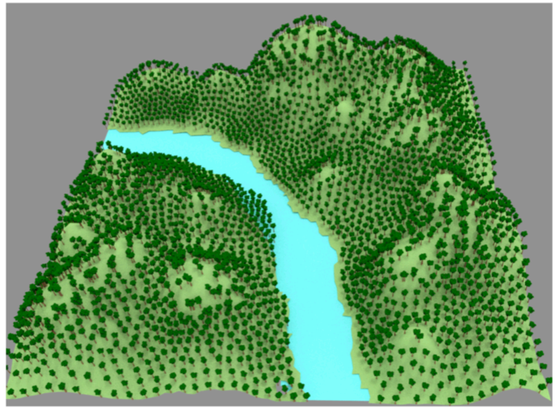


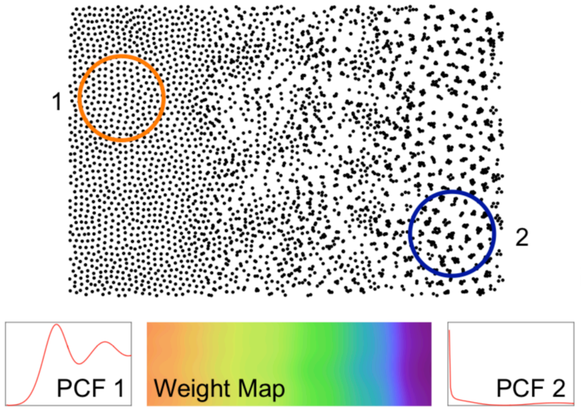

Command UI
0.480
t
run
get cam pos

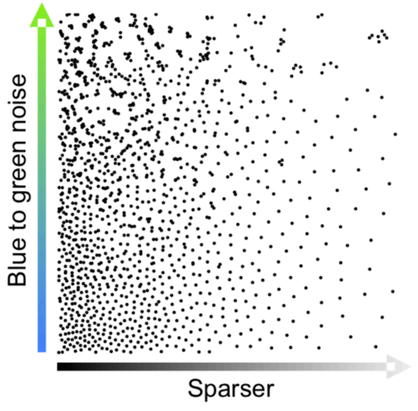

Command UI
0.480
t
run
get cam pos

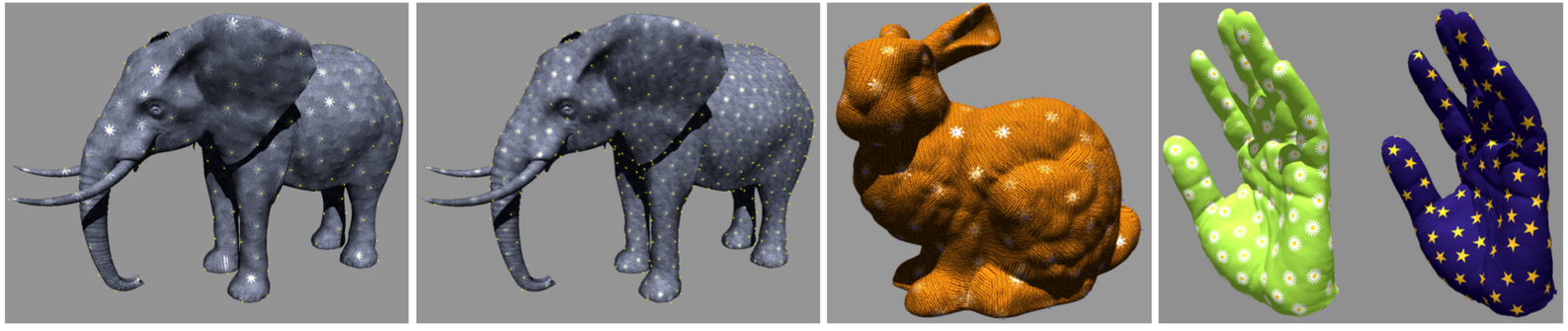


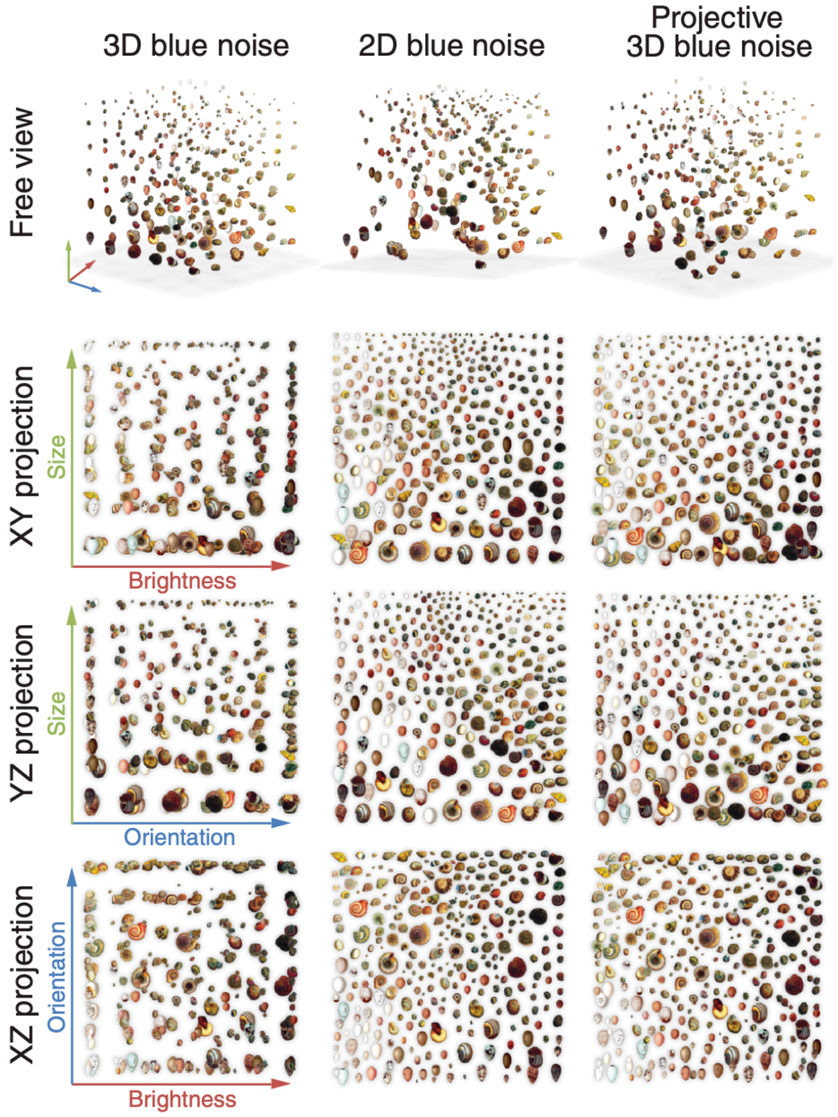
(Alternative use-cases)





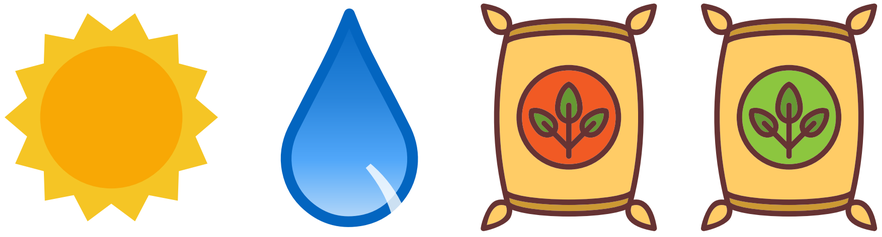






Experimental Design: Orthogonal Arrays

Factors:
$d = 4$

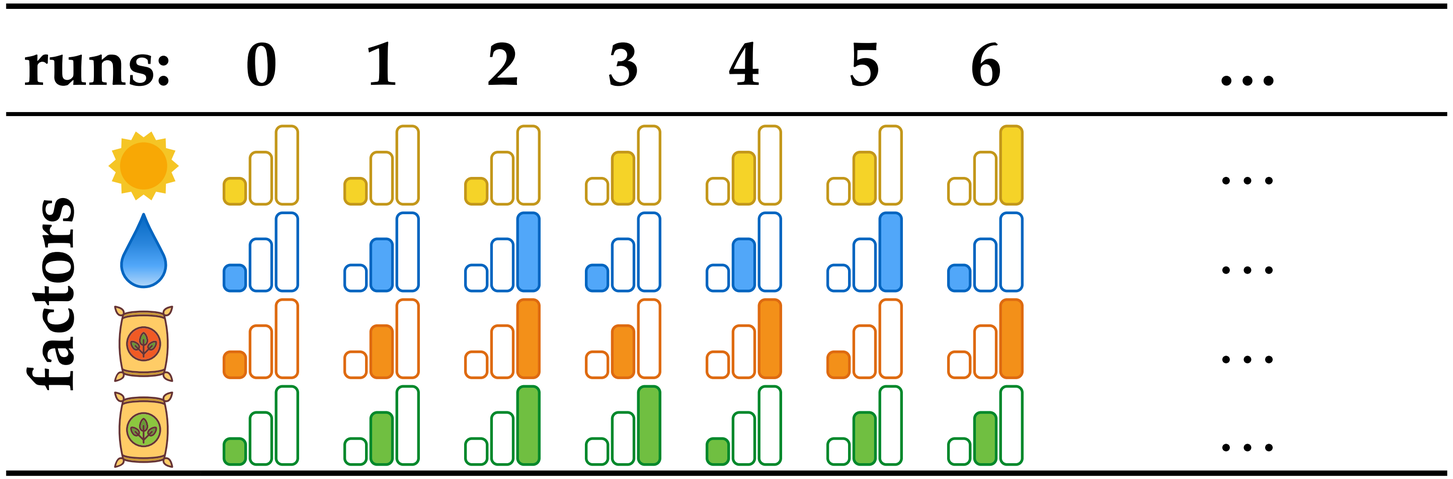


Levels:
$s = 3$

(amounts)

Slides from Jarosz Wojciech, Dartmouth University

An experiment plan



An experiment plan

s discrete levels

$\{0, \ldots, s\text{-}1\}$

runs:	0	1	2	3	4	5	6	…
factors								…
								…
								…
								…

An experiment plan

s discrete levels

$\{0, \ldots, s\text{-}1\}$

runs:		0	1	2	3	4	5	6	…
factors		0	0	0	1	1	1	2	…
		0	1	2	0	1	2	0	…
		0	1	2	1	2	0	2	…
		0	1	2	2	0	1	1	…

An experiment plan

✘ Testing all combinations of factors is expensive: $N = s^d = 81$

runs:	0	1	2	3	4	5	6	…	80
factors 	0	0	0	1	1	1	2	…	2
	0	1	2	0	1	2	0	…	2
	0	1	2	1	2	0	2	…	1
	0	1	2	2	0	1	1	…	0

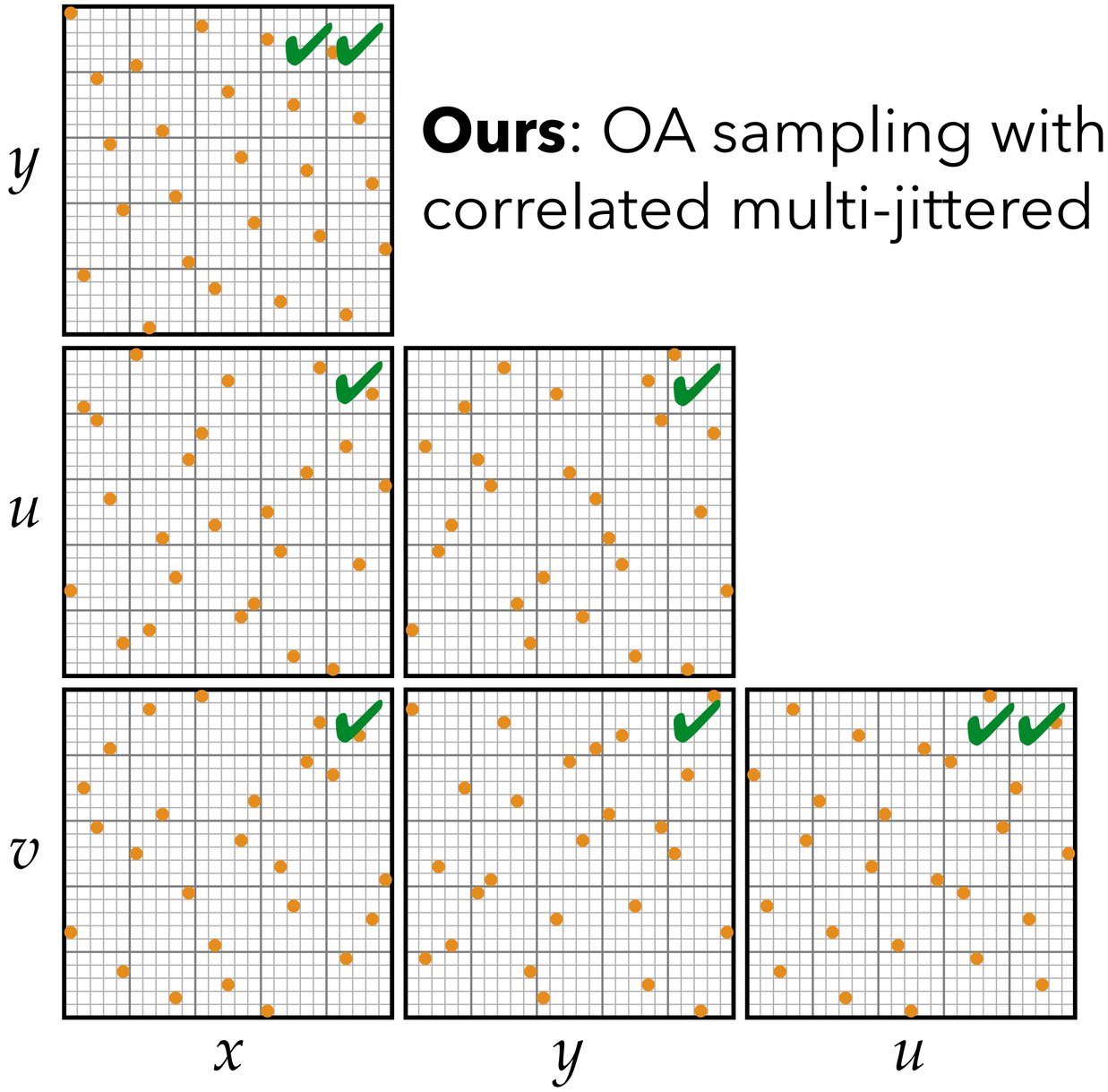
An experiment plan

✘ Testing all combinations of factors is expensive: $N = s^d = 81$

- What if we consider at most 2-way interactions?

runs:	0	1	2	3	4	5	6	…	80
factors	0	0	0	1	1	1	2	…	2
	0	1	2	0	1	2	0	…	2
	0	1	2	1	2	0	2	…	1
	0	1	2	2	0	1	1	…	0

Orthogonal Array with strength 2



Ours: OA sampling with correlated multi-jittered offsets

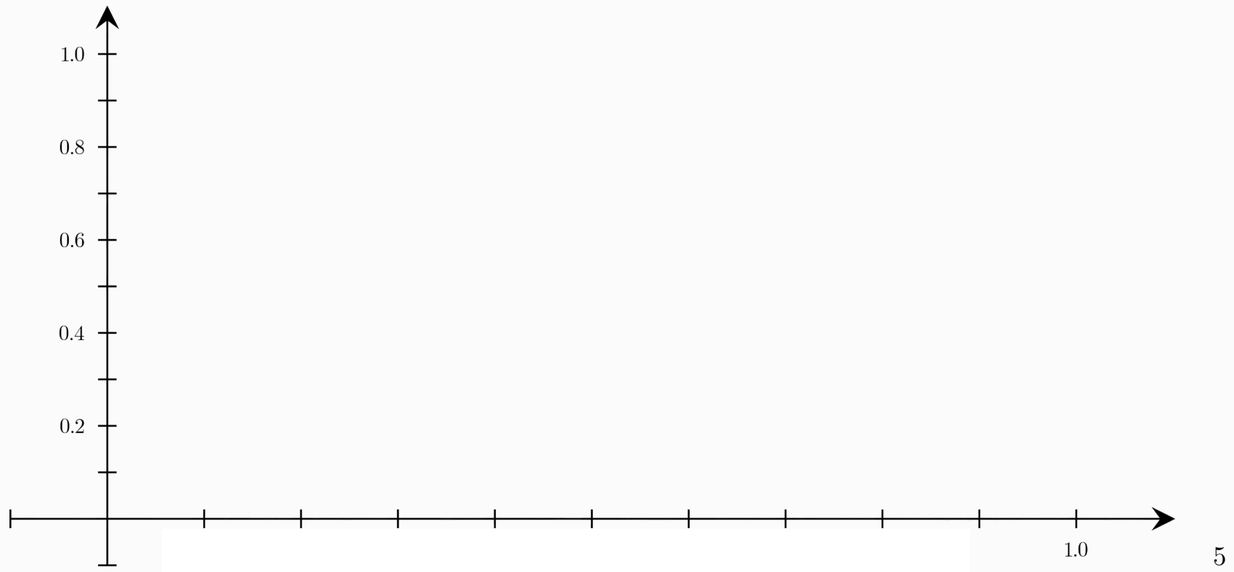
Outline

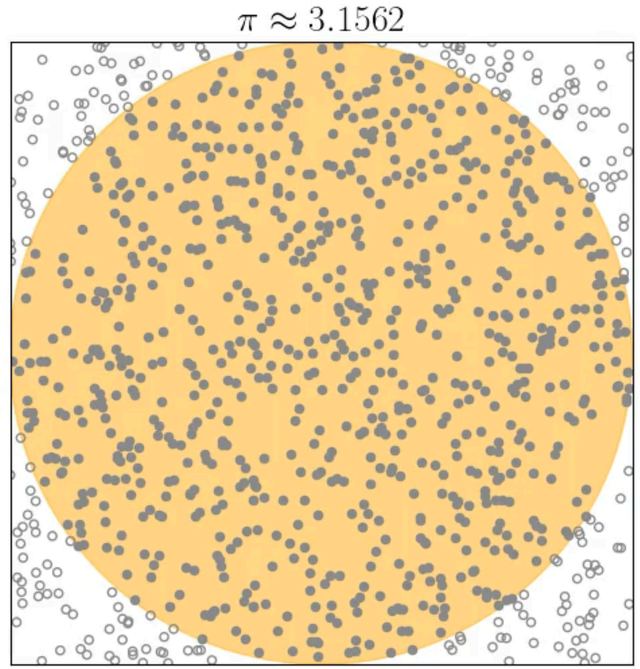
- Samplers 101
- Uniformity measures
- Low discrepancy sequences and algebraic samplers
- Non-Euclidean Sliced Optimal Transport Sampling

Monte Carlo Integration

$$\int_\Omega f dx \approx \frac{1}{n} \sum_i f(x_i)$$

Quasi-Monte Carlo



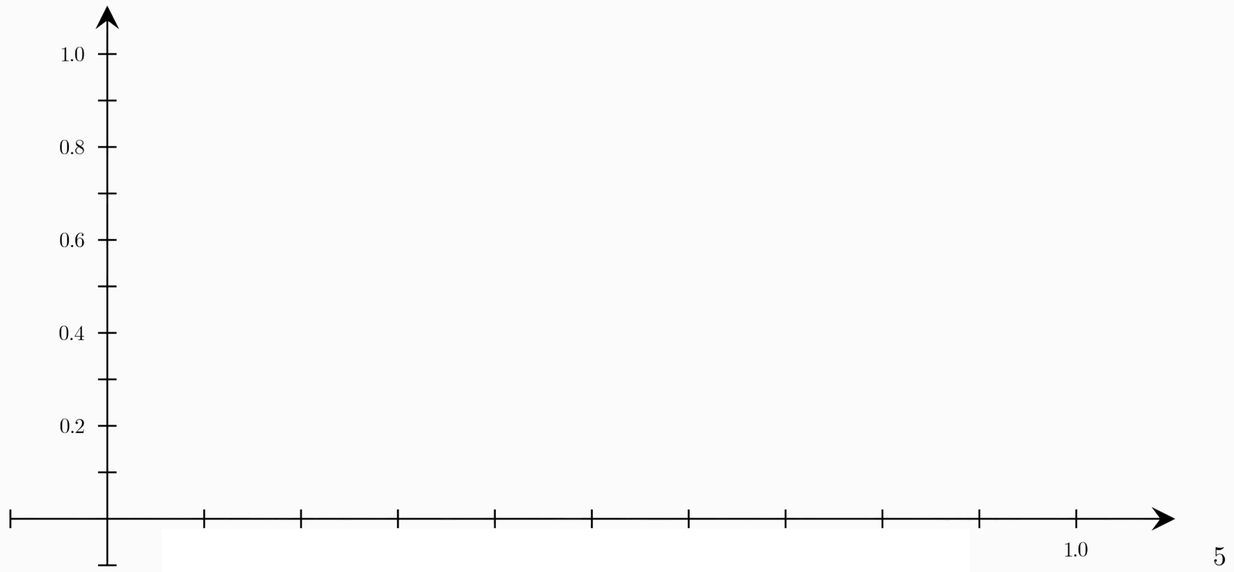


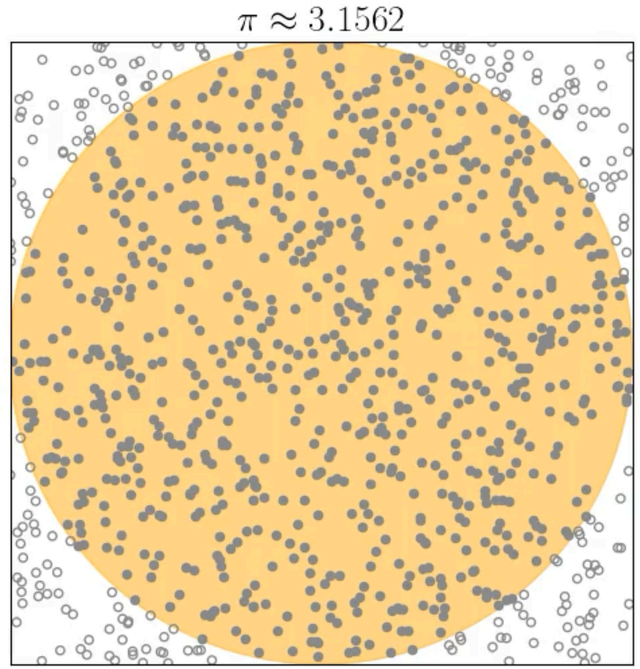
Whitenoise

Monte Carlo Integration

$$\int_\Omega f dx \approx \frac{1}{n} \sum_i f(x_i)$$

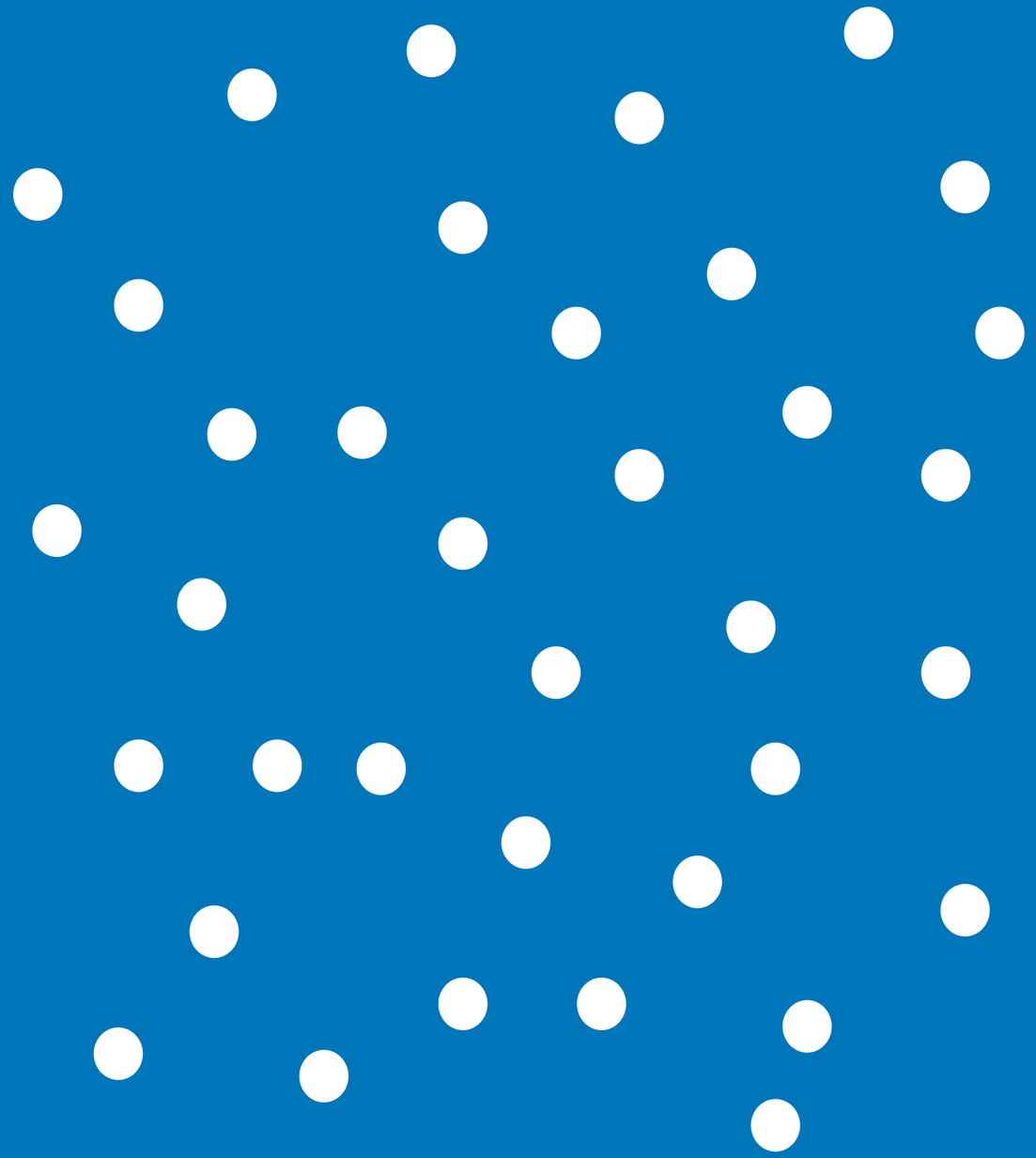
Quasi-Monte Carlo





Whitenoise

Samplers 101



Whitenoise

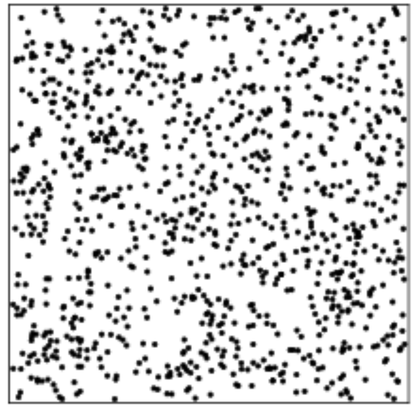
$$x_i \sim \mathcal{U}(0,1)$$

$$\int_\Omega f dx \approx \frac{1}{n}\sum_i f(x_i) \text{ with error in } O\left(\frac{1}{n}\right)$$

Card 1: White noise

Dim Any | Sequence Yes | Class Random | Time Fast

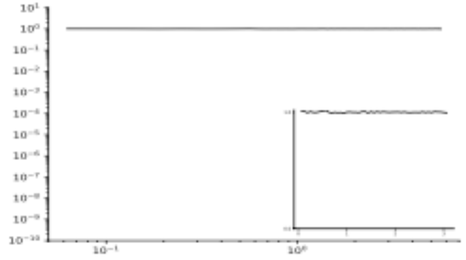
Points



Discrepancy

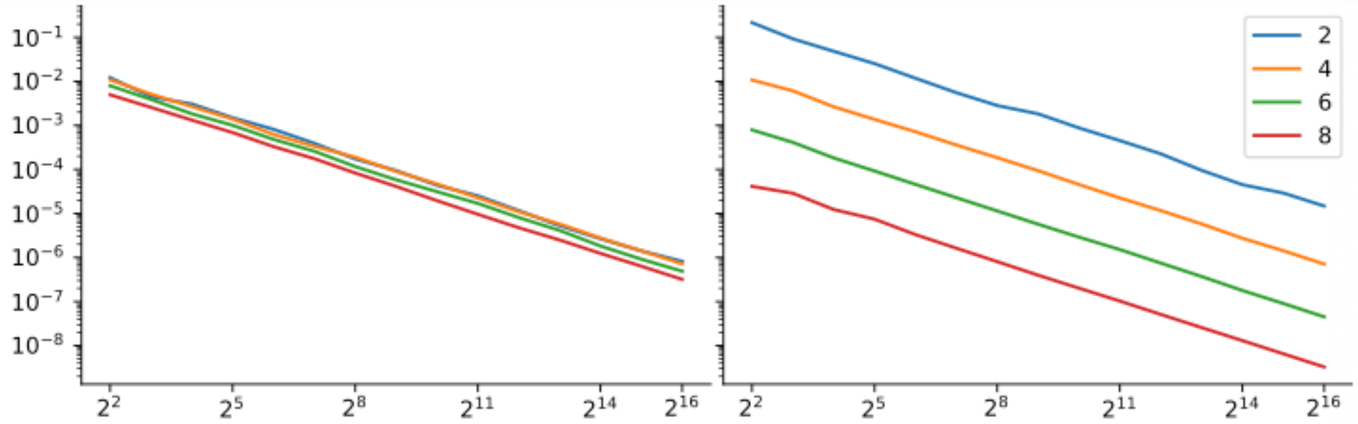
$\mathcal{O}\left(\frac{1}{\sqrt{N}}\right)$ [Lem09]

Radial power spectra



Power Spectra

Integration Error



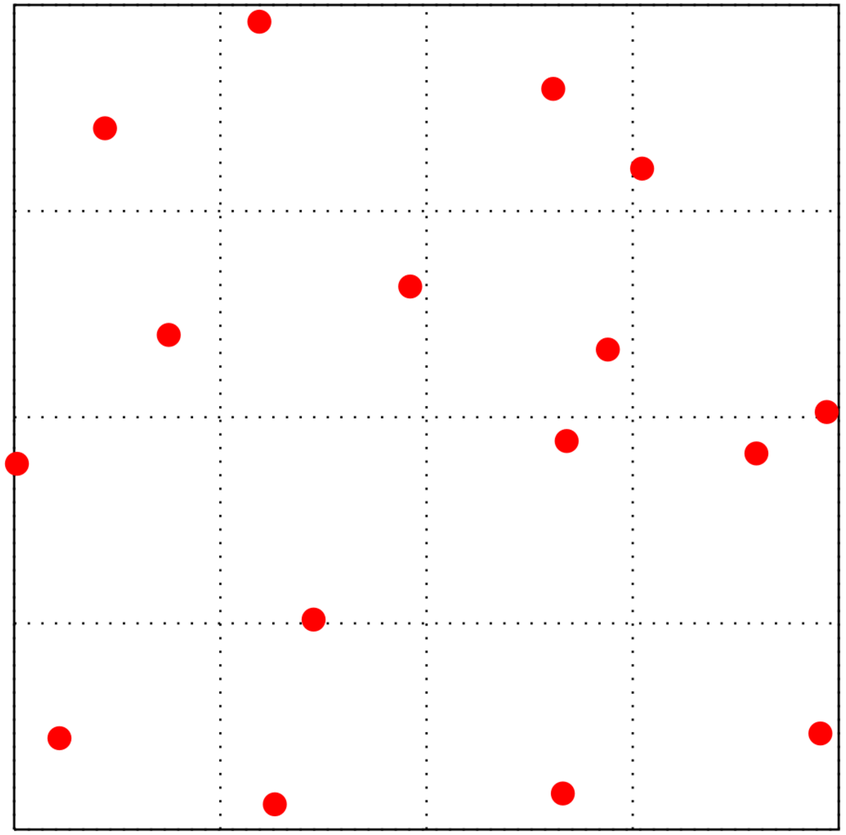
PRNG / rand() / drand48() / pcg32 / .. test01

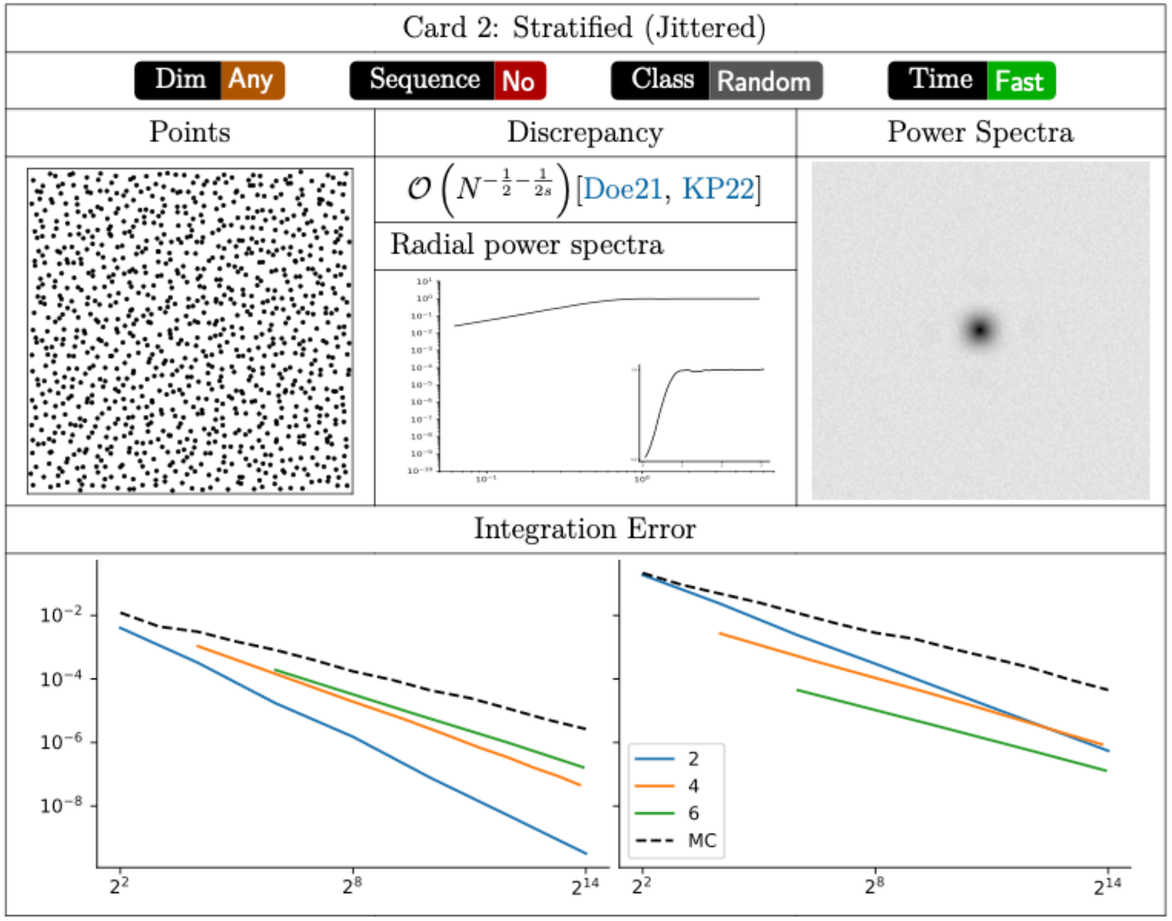
```
Pointset whitenoise2D(size_t N, unsigned int seed = 12345)
{
  std::random_device rd;  // a seed source for the random number engine
  std::mt19937 generator(rd()); // mersenne_twister_engine seeded with rd()
  std::uniform_real_distribution<double> distribution(0.0,1.0);

  Pointset P(N);
  for(auto i=0; i < N; ++i)
    P[i] = { distribution(generator),distribution(generator) };

  return P;
}
```

```
double custom_rand() {
  static unsigned x = 1;
  x = (x * 3) % 100;
  return x / 100.0;
}
```

```
double trig(double x)
{
  double y = 43757.5453*sin( std::abs(x) * 12.9898);
  return y - floor(y);
}
```

Stratified sampling

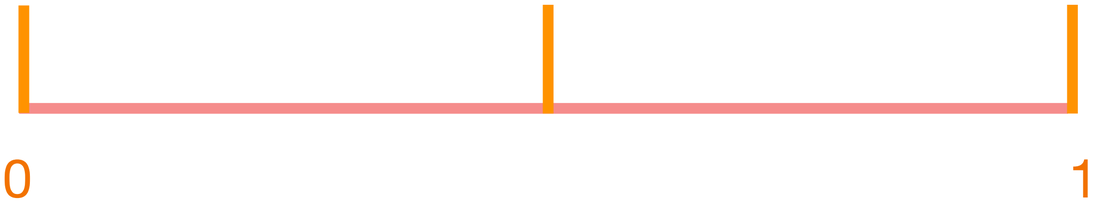




van der Corput

Index $a \in \mathbb{N} \to (a_0, \ldots, a_{m-1})_2$

$$.(a_0, \ldots, a_{m-1})_2 \to x = \frac{1}{2^m} \sum_{j=0}^{m-1} a_j 2^j \in [0,1)$$

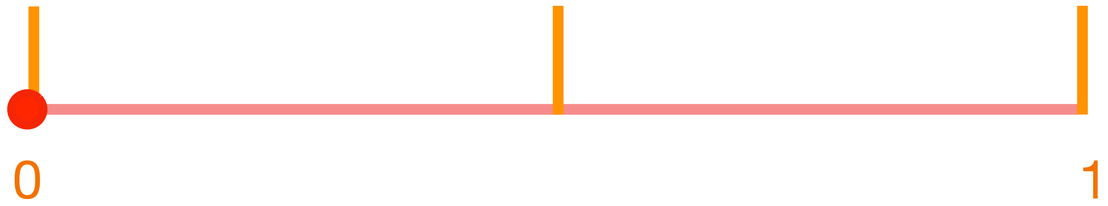


van der Corput

Index $a \in \mathbb{N} \rightarrow (a_0, \ldots, a_{m-1})_2$

$$. (a_0, \ldots, a_{m-1})_2 \rightarrow x = \frac{1}{2^m} \sum_{j=0}^{m-1} a_j 2^j \in [0,1)$$

$0 \rightarrow (00000000)_2 \rightarrow 0$



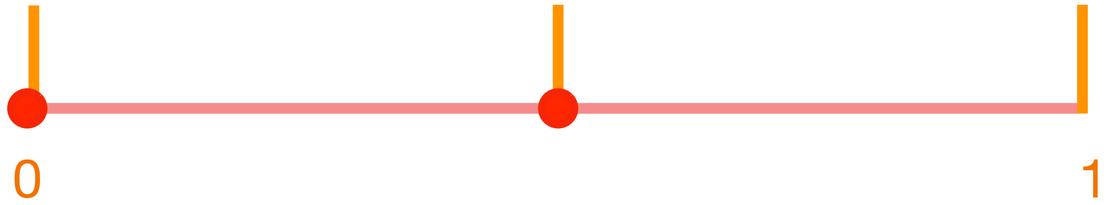
van der Corput

Index $a \in \mathbb{N} \rightarrow (a_0, \ldots, a_{m-1})_2$

$$. (a_0, \ldots, a_{m-1})_2 \rightarrow x = \frac{1}{2^m} \sum_{j=0}^{m-1} a_j 2^j \in [0,1)$$

$$0 \rightarrow (00000000)_2 \rightarrow 0$$

$$1 \rightarrow (00000001)_2 \rightarrow \frac{1}{2}$$



van der Corput

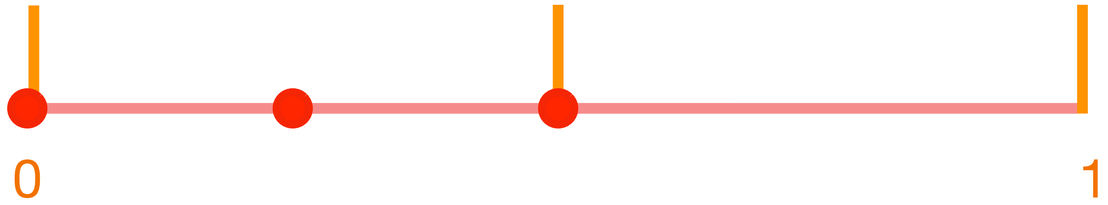
Index $a \in \mathbb{N} \to (a_0, \ldots, a_{m-1})_2$

$$. (a_0, \ldots, a_{m-1})_2 \to x = \frac{1}{2^m} \sum_{j=0}^{m-1} a_j 2^j \in [0,1)$$

$$0 \to (00000000)_2 \to 0$$

$$1 \to (00000001)_2 \to \frac{1}{2}$$

$$2 \to (00000010)_2 \to \frac{1}{4}$$



van der Corput

Index $a \in \mathbb{N} \to (a_0, \ldots, a_{m-1})_2$

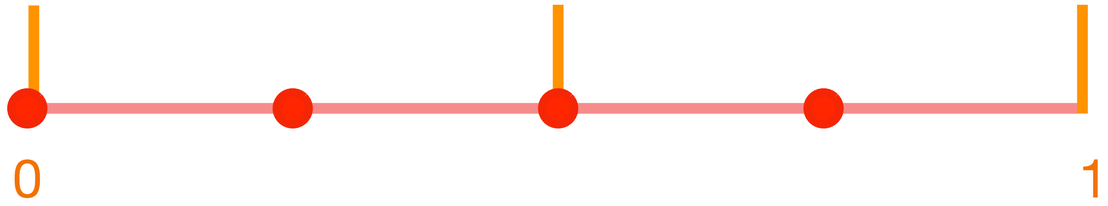
$$. (a_0, \ldots, a_{m-1})_2 \to x = \frac{1}{2^m} \sum_{j=0}^{m-1} a_j 2^j \in [0,1)$$

$$0 \to (00000000)_2 \to 0$$

$$1 \to (00000001)_2 \to \frac{1}{2}$$

$$2 \to (00000010)_2 \to \frac{1}{4}$$

$$3 \to (00000011)_2 \to \frac{1}{2} + \frac{1}{4}$$



van der Corput

Index $a \in \mathbb{N} \to (a_0, \ldots, a_{m-1})_2$

$$. (a_0, \ldots, a_{m-1})_2 \to x = \frac{1}{2^m} \sum_{j=0}^{m-1} a_j 2^j \in [0,1)$$

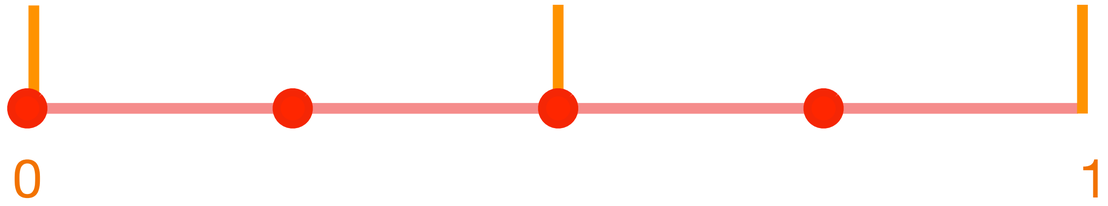
$$0 \to (00000000)_2 \to 0$$

$$1 \to (00000001)_2 \to \frac{1}{2}$$

$$2 \to (00000010)_2 \to \frac{1}{4}$$

$$3 \to (00000011)_2 \to \frac{1}{2} + \frac{1}{4}$$

• • •



van der Corput

Index $a \in \mathbb{N} \rightarrow (a_0, \ldots, a_{m-1})_2$

$$. (a_0, \ldots, a_{m-1})_2 \rightarrow x = \frac{1}{2^m} \sum_{j=0}^{m-1} a_j 2^j \in [0,1)$$

$$0 \rightarrow (00000000)_2 \rightarrow 0$$

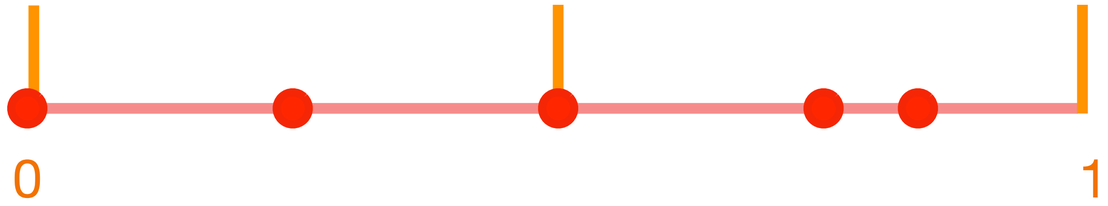
$$1 \rightarrow (00000001)_2 \rightarrow \frac{1}{2}$$

$$2 \rightarrow (00000010)_2 \rightarrow \frac{1}{4}$$

$$3 \rightarrow (00000011)_2 \rightarrow \frac{1}{2} + \frac{1}{4}$$

• • •

$$21 \rightarrow (00010101)_2 \rightarrow \frac{1}{2} + \frac{1}{8} + \frac{1}{32}$$



van der Corput

Index $a \in \mathbb{N} \to (a_0, \ldots, a_{m-1})_2$

$$. (a_0, \ldots, a_{m-1})_2 \to x = \frac{1}{2^m} \sum_{j=0}^{m-1} a_j 2^j \in [0,1)$$

$$0 \to (00000000)_2 \to 0$$

$$1 \to (00000001)_2 \to \frac{1}{2}$$

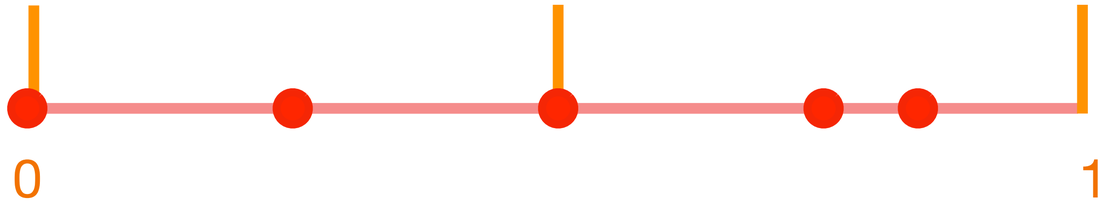
$$2 \to (00000010)_2 \to \frac{1}{4}$$

$$3 \to (00000011)_2 \to \frac{1}{2} + \frac{1}{4}$$

• • •

$$21 \to (00010101)_2 \to \frac{1}{2} + \frac{1}{8} + \frac{1}{32}$$

• • •

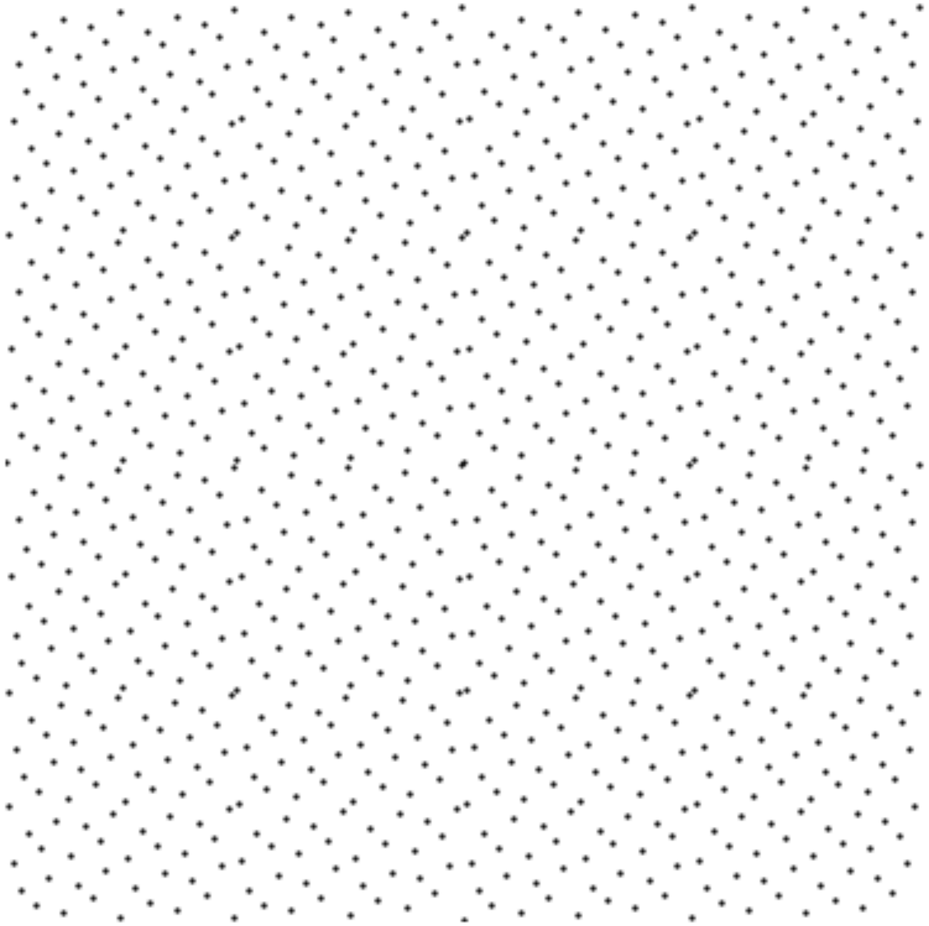


Hammersley

Index $i \in \{0 \ldots N\} \rightarrow \left(i/N, vdC(i)\right)$

Hammersley

Index $i \in \{0 \ldots N\} \rightarrow \left(i/N, vdC(i)\right)$

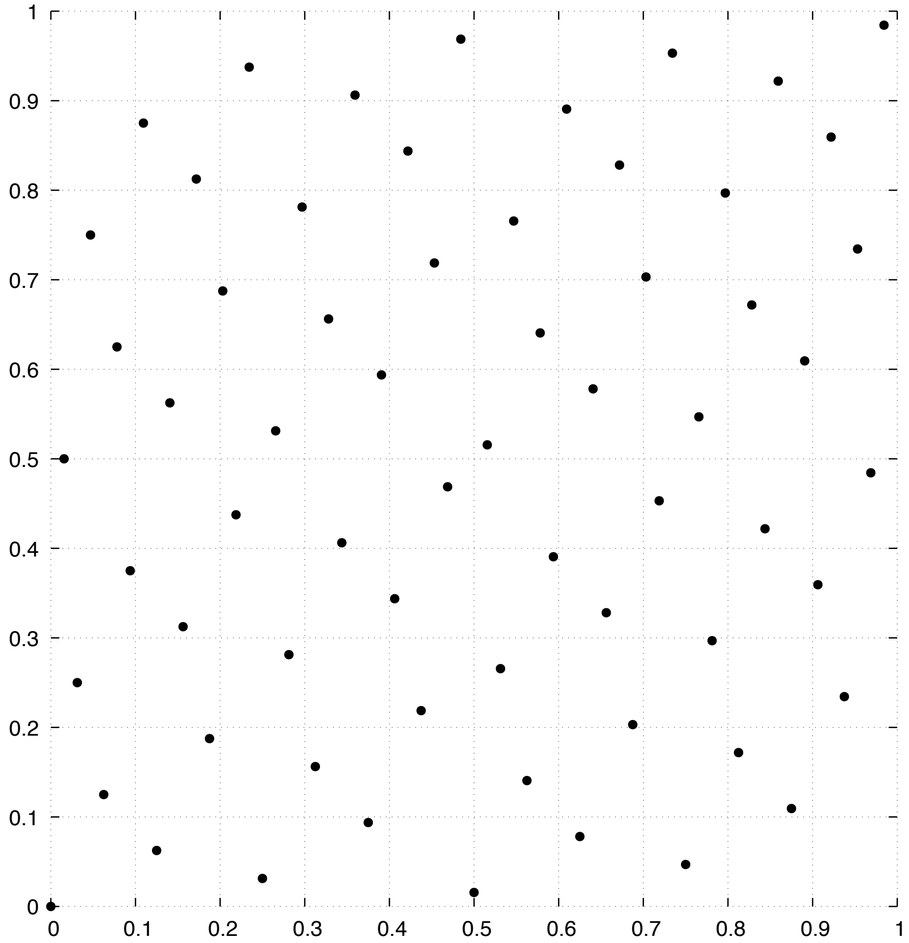


Hammersley

Index $i \in \{0 \ldots N\} \rightarrow \left(i/N, vdC(i)\right)$

Hammersley

Index $i \in \{0 \ldots N\} \rightarrow \left(i/N, vdC(i)\right)$



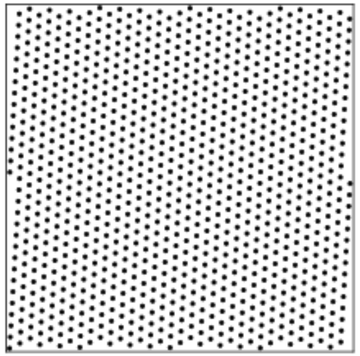
Fibonaci lattice

$$i \in \{0 \ldots N\} \rightarrow \left(\left\lfloor \frac{i}{\phi} \right\rfloor, \frac{i}{N} \right)$$

Card 3: Fibonaci lattice [Zar70]

Dim 2D | Sequence No | Class Lattice | Time Fast

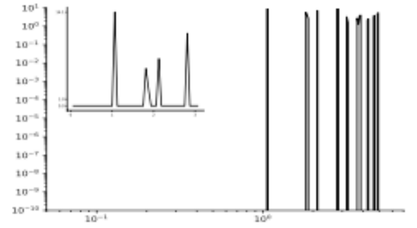
Points



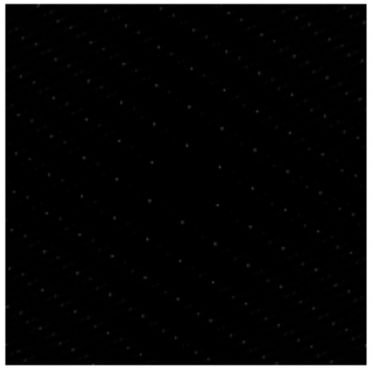
Discrepancy

$\mathcal{O}\left(\frac{\log N}{N}\right)$ [Nie92]

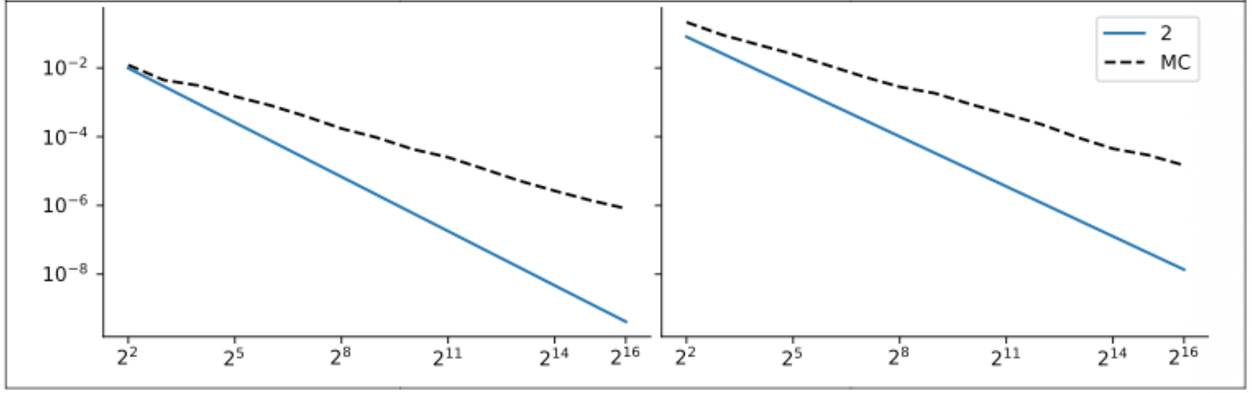
Radial power spectra



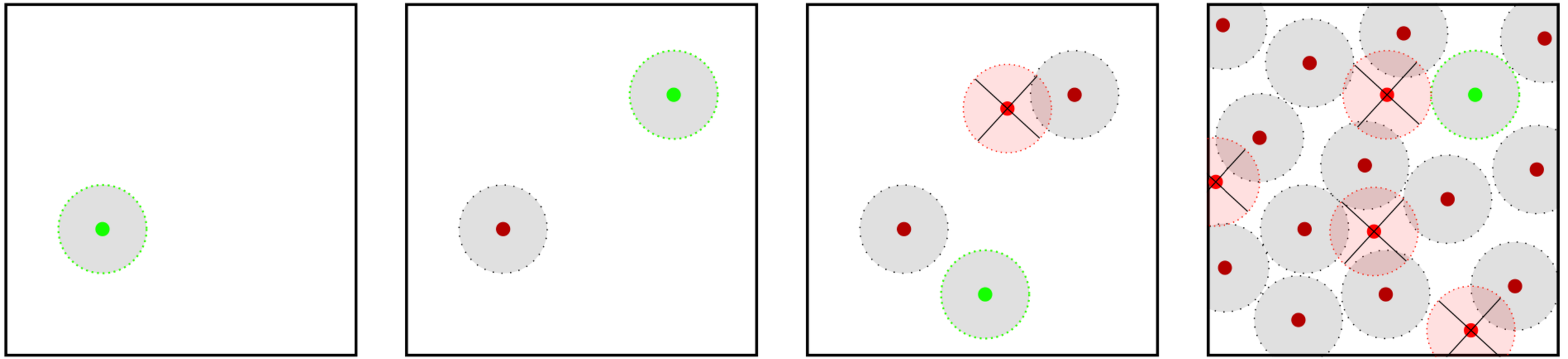
Power Spectra



Integration Error



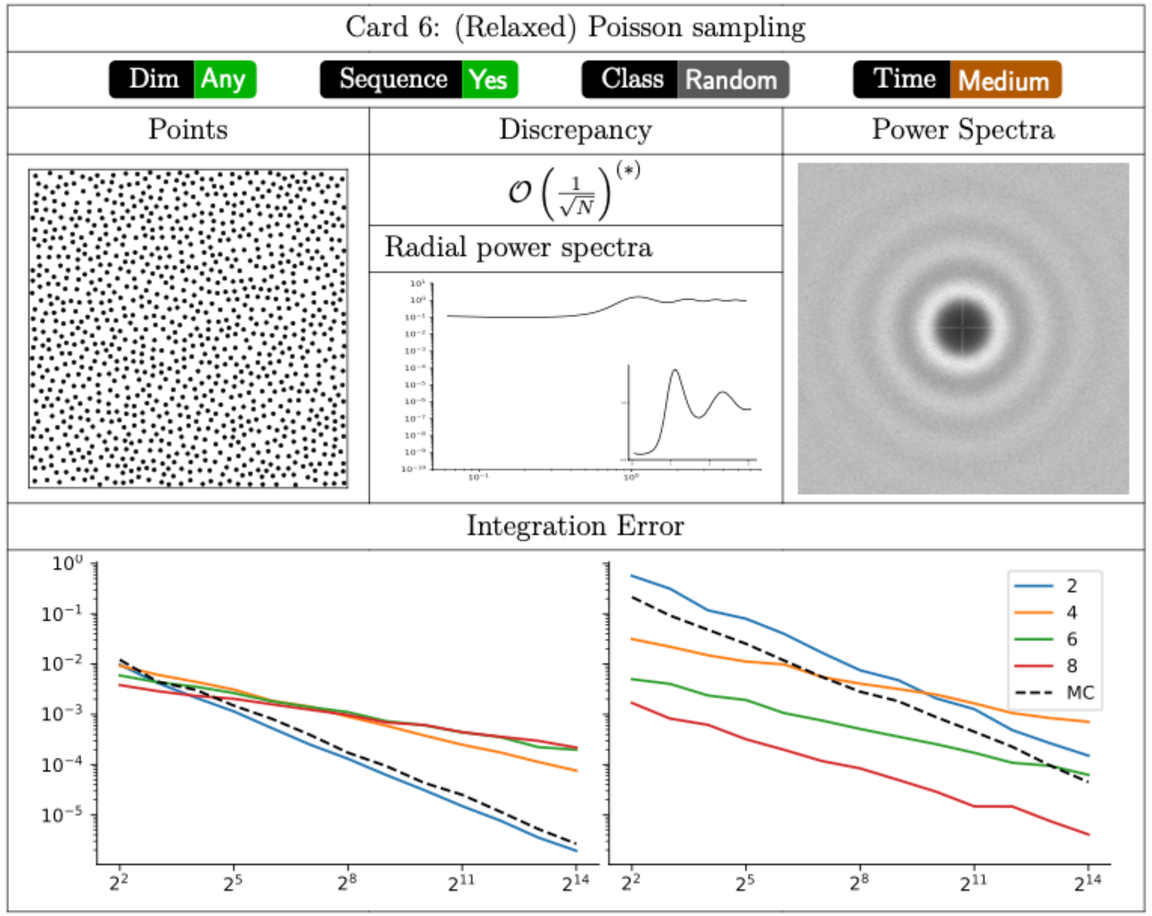
Poisson disk sampling



Iteration 0 Iteration 1 Iteration 2 Iteration 14

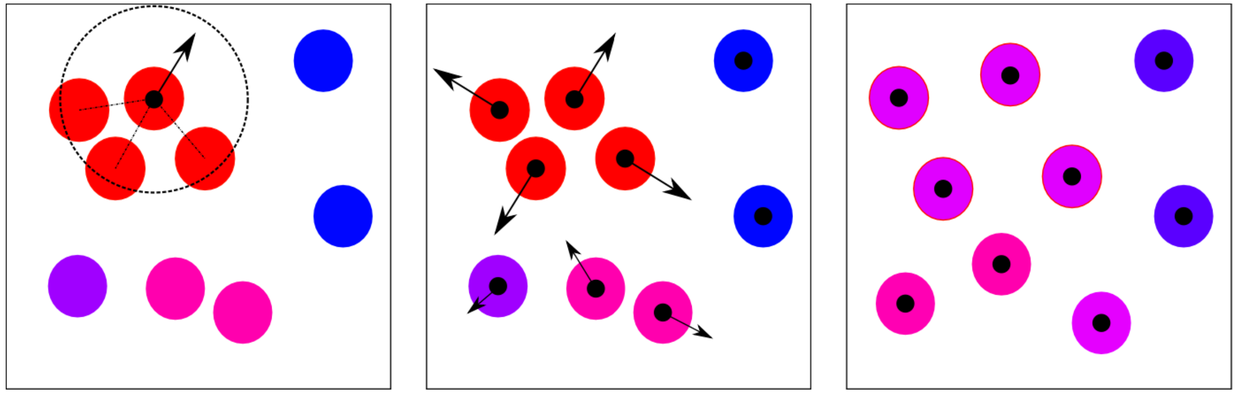
Poisson disk sampling

- Fast implementations/approximations (Bridson 2007)
- Easy to control
  - On non Euclidean domains
  - For custom metrics
  - …



… but not as good as WN (asymptotically)

Blue Noise / Lloyd



Card 6: (Relaxed) Poisson sampling

Dim Any | Sequence Yes | Class Random | Time Medium

Points

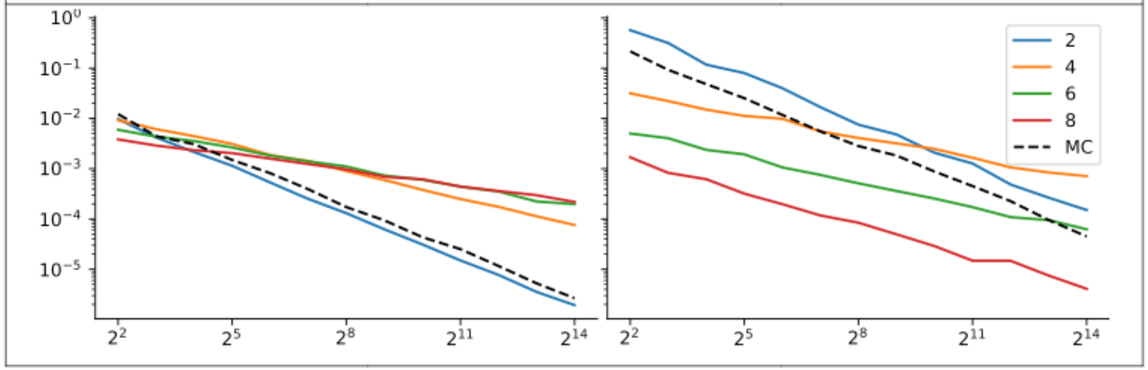
Discrepancy

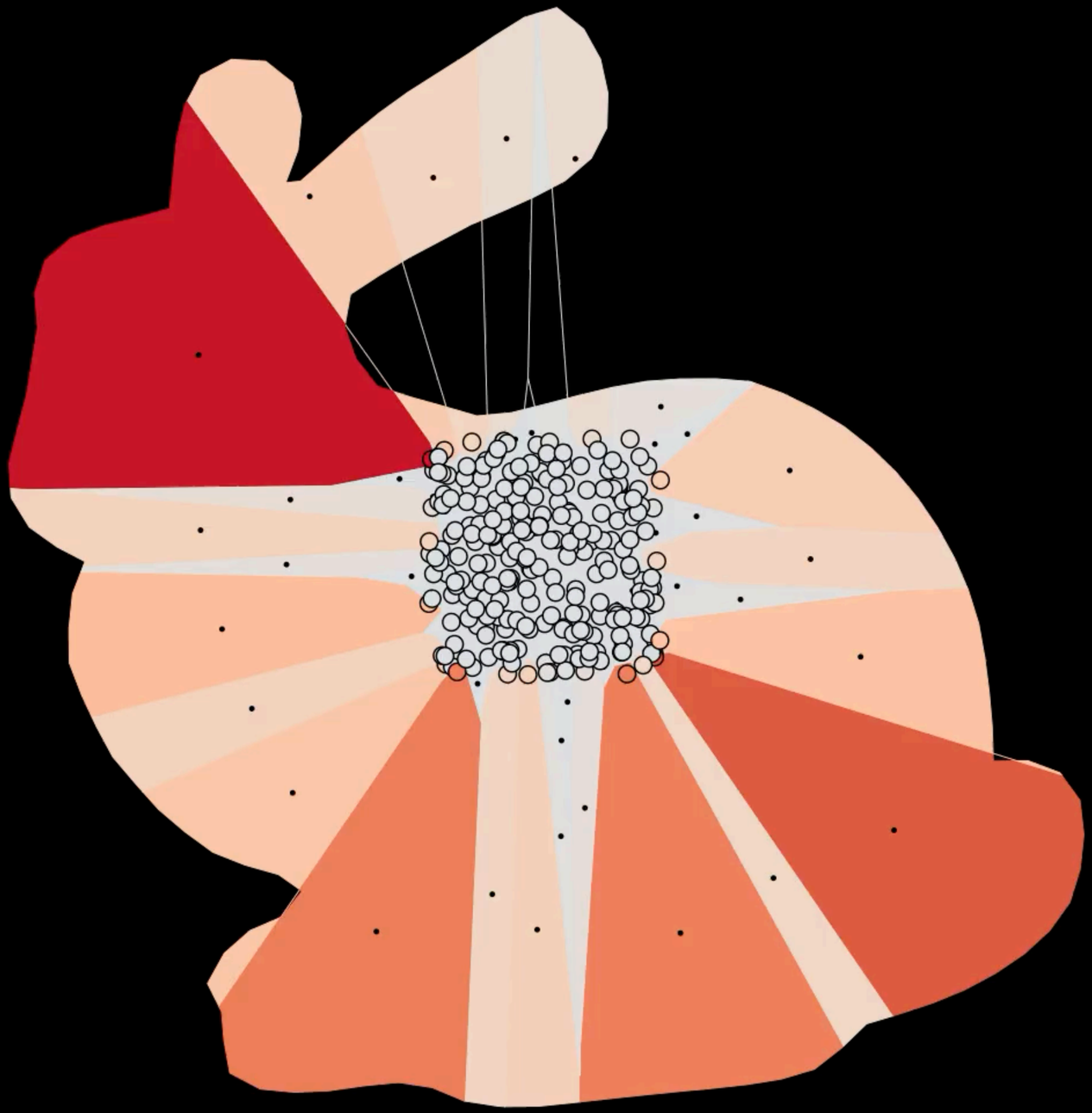
$\mathcal{O}\left(\frac{1}{\sqrt{N}}\right)^{(*)}$

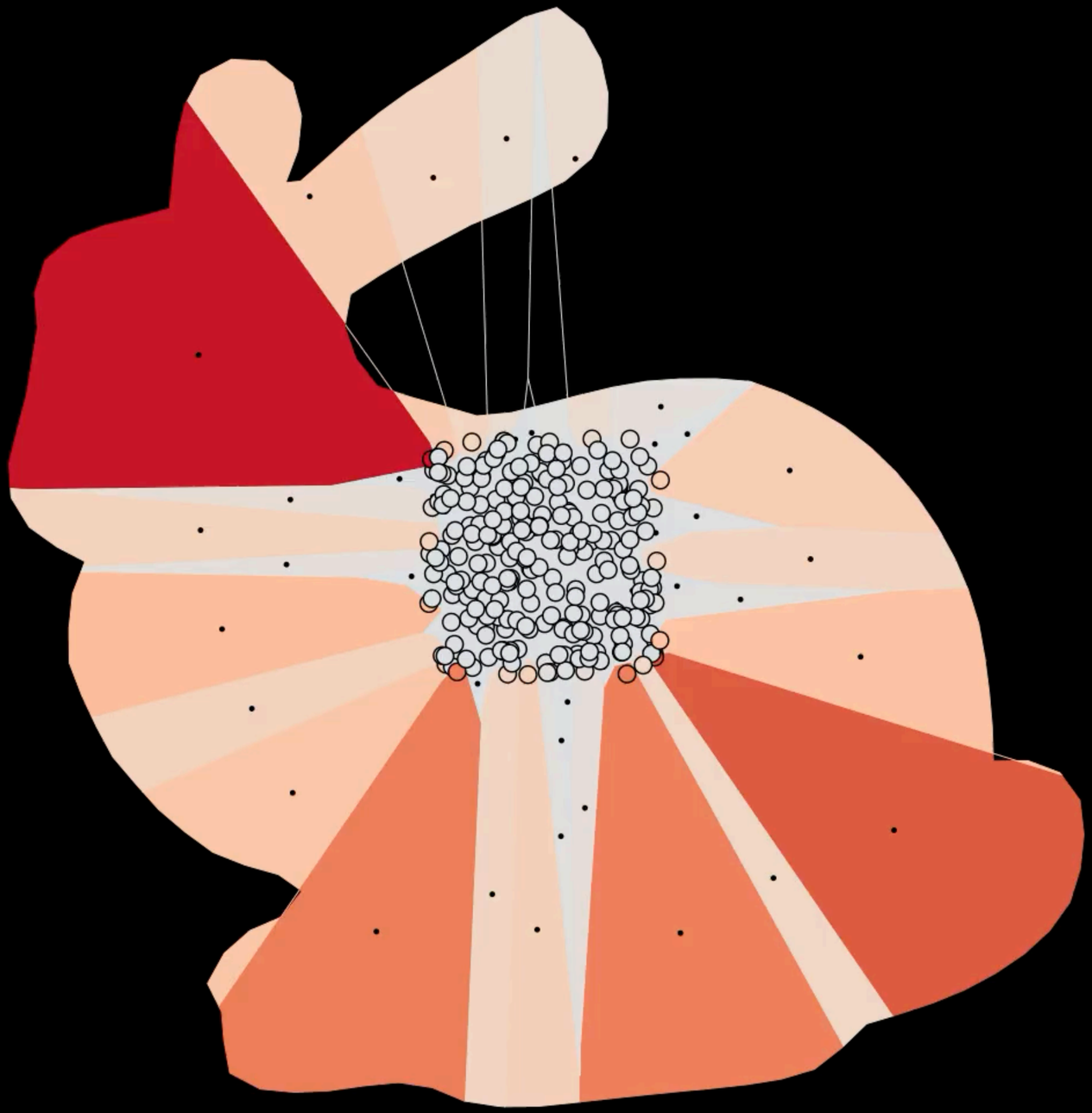
Radial power spectra

Power Spectra

Integration Error



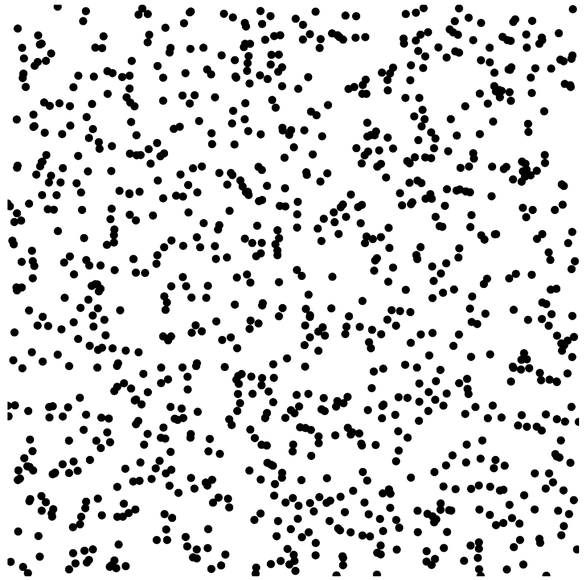




Motivation: Monte Carlo Integration in 2d

$$I_n := \frac{1}{n} \sum f(x_i) \qquad \Delta_n := \left| \int f dx - I_n \right|$$

Whitenoise

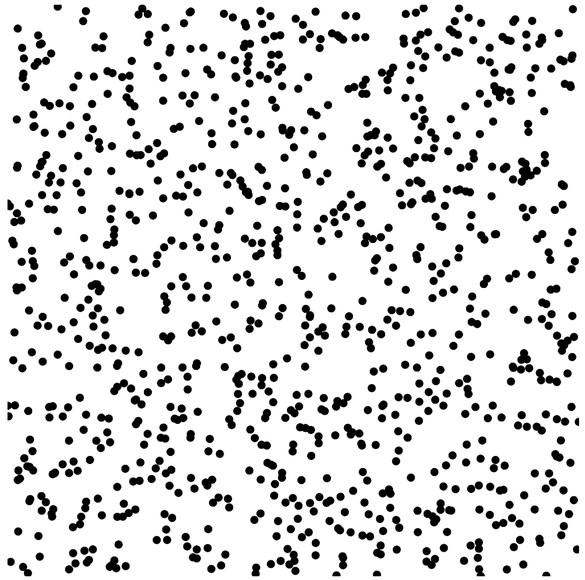


$$Var(I_n) = O\left(\frac{\sigma_f^2}{n}\right)$$

Motivation: Monte Carlo Integration in 2d

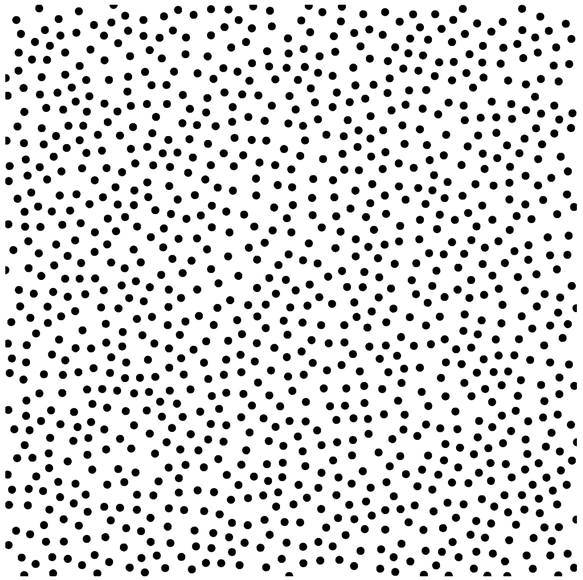
$$I_n := \frac{1}{n}\sum f(x_i) \qquad \Delta_n := \left| \int f dx - I_n \right|$$

Whitenoise



$$Var(I_n) = O\left(\frac{\sigma_f^2}{n}\right)$$

Poisson Disk

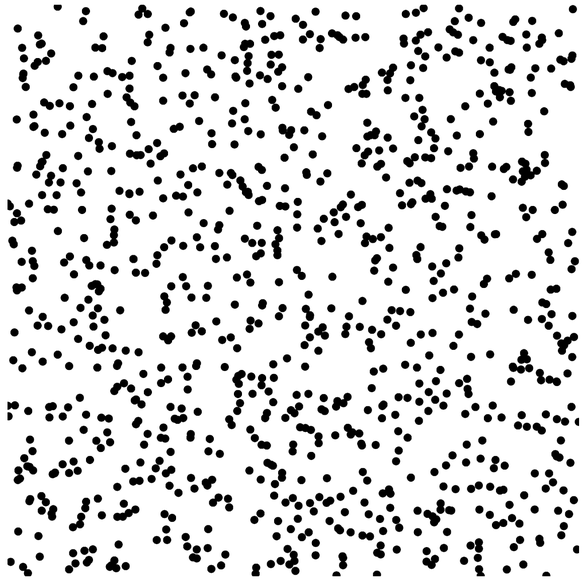


$$Var(I_n) = O\left(\frac{1}{n}\right)$$

Motivation: Monte Carlo Integration in 2d

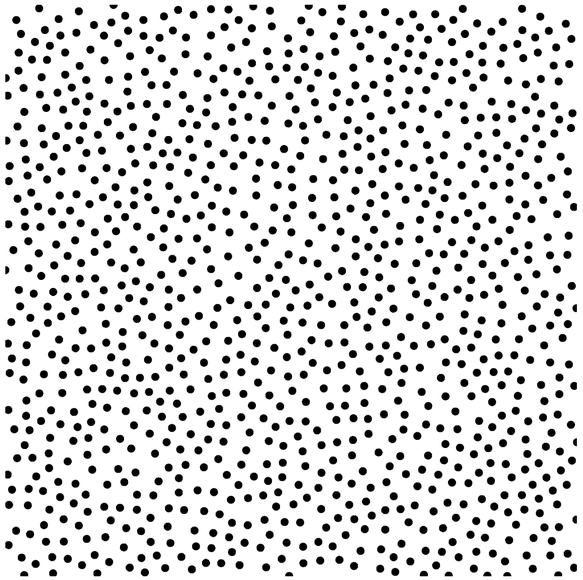
$$I_n := \frac{1}{n}\sum f(x_i) \qquad \Delta_n := \left| \int f dx - I_n \right|$$

Whitenoise



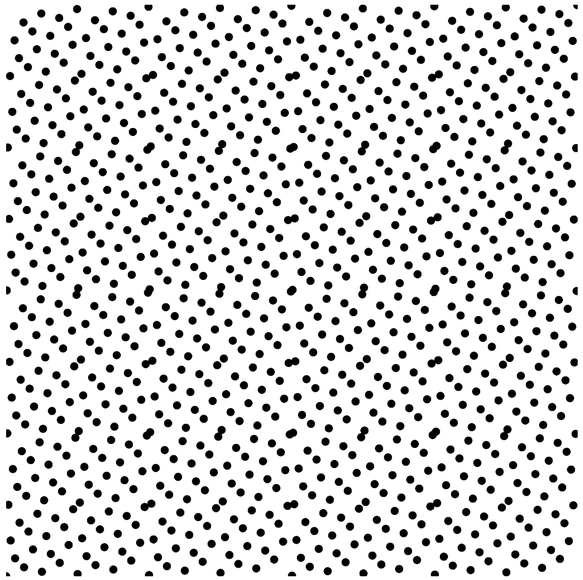
$$Var(I_n) = O\left(\frac{\sigma_f^2}{n}\right)$$

Poisson Disk



$$Var(I_n) = O\left(\frac{1}{n}\right)$$

Low discrepancy sequences

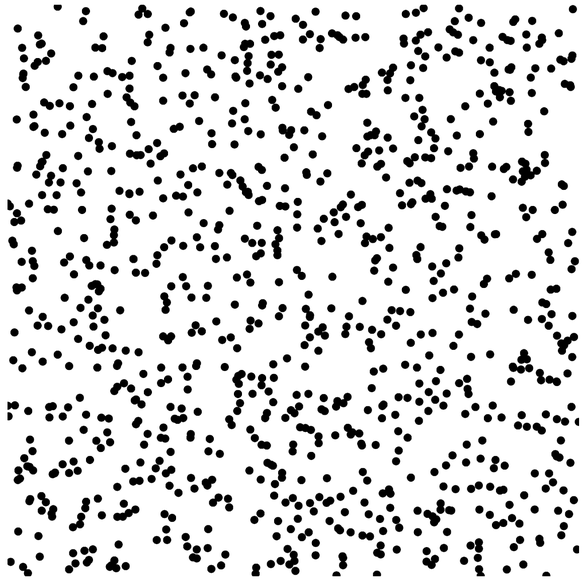


$$\Delta_n^2 = O\left(\frac{log(n)^{2(s-1)}}{n^2}\right)$$

Motivation: Monte Carlo Integration in 2d

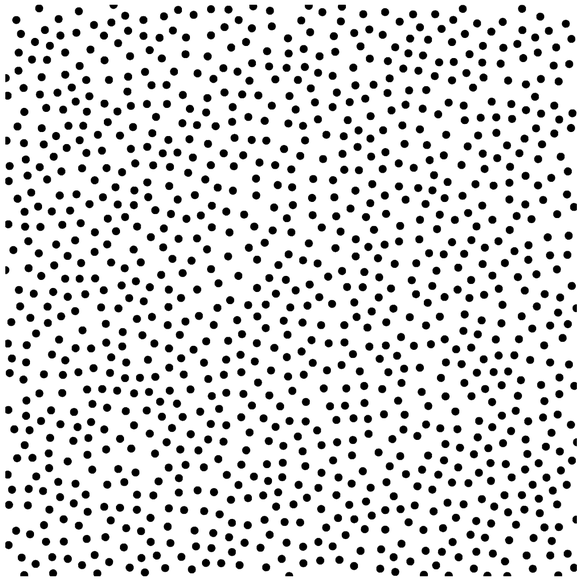
$$I_n := \frac{1}{n}\sum f(x_i) \qquad \Delta_n := \left| \int f dx - I_n \right|$$

Whitenoise



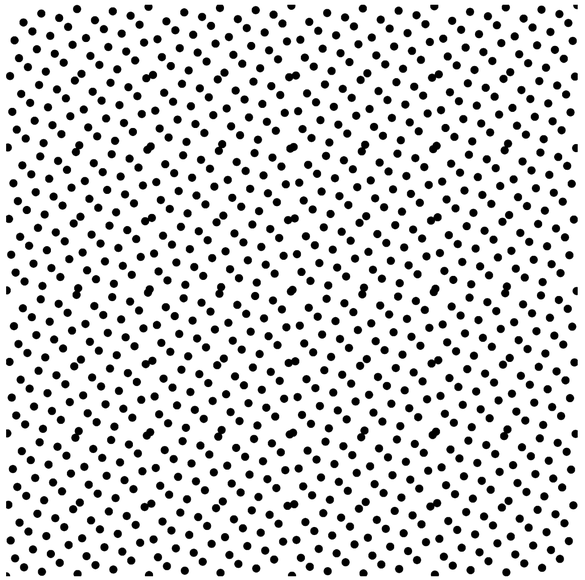
$$Var(I_n) = O\left(\frac{\sigma_f^2}{n}\right)$$

Poisson Disk



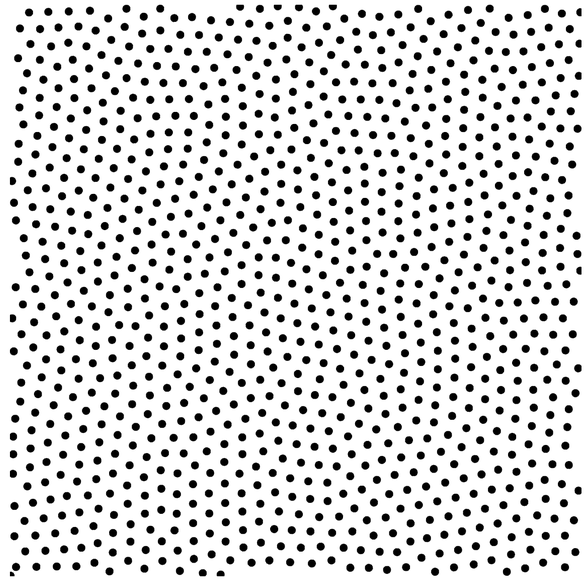
$$Var(I_n) = O\left(\frac{1}{n}\right)$$

Low discrepancy sequences



$$\Delta_n^2 = O\left(\frac{log(n)^{2(s-1)}}{n^2}\right)$$

Blue noise sampling

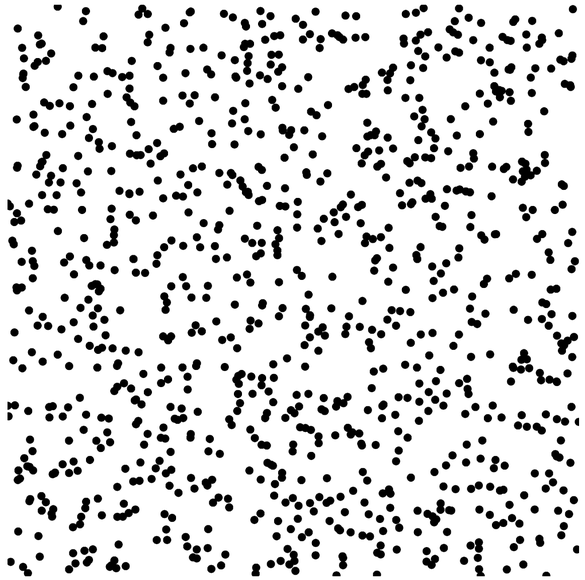


$$Var(I_n) = O\left(\frac{1}{n^{1+1/s}}\right)$$

Motivation: Monte Carlo Integration in 2d

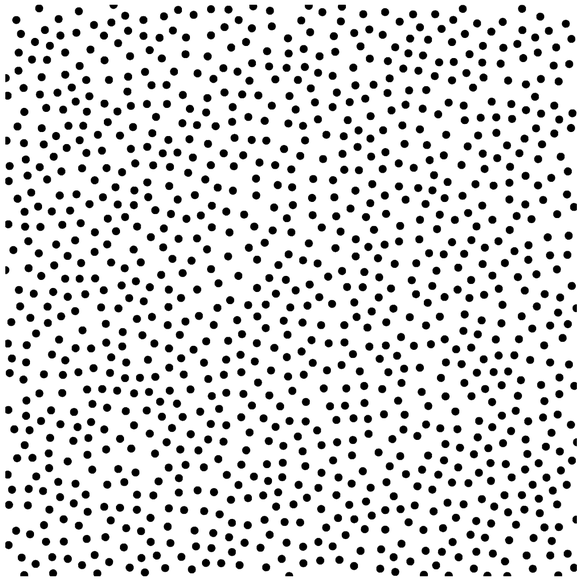
$$I_n := \frac{1}{n}\sum f(x_i) \qquad \Delta_n := \left| \int f dx - I_n \right|$$

Whitenoise



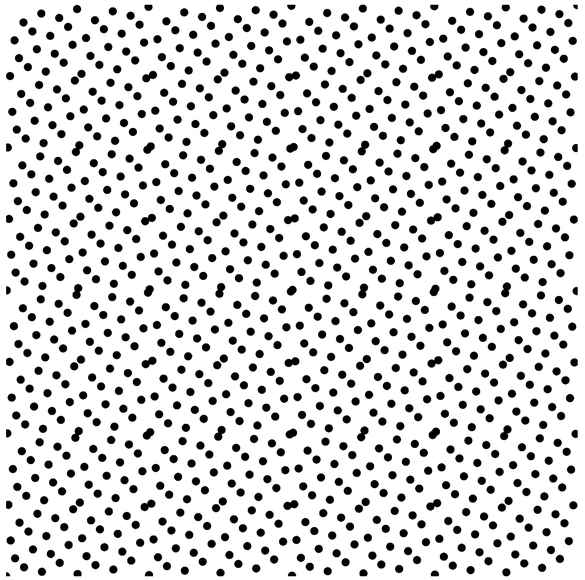
$$Var(I_n) = O\left(\frac{\sigma_f^2}{n}\right)$$

Poisson Disk



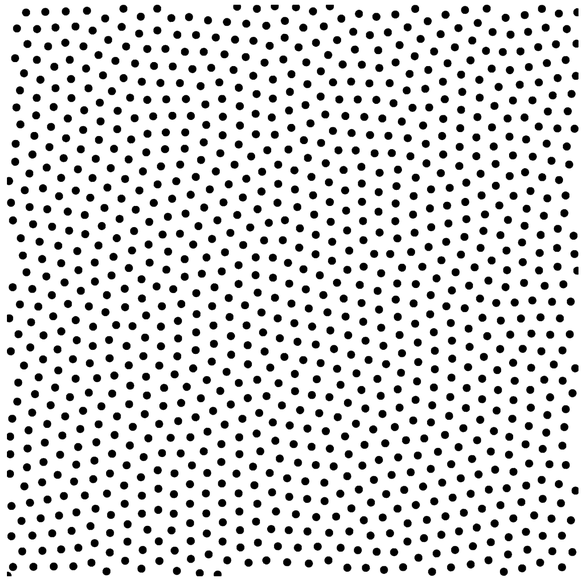
$$Var(I_n) = O\left(\frac{1}{n}\right)$$

Low discrepancy sequences



$$\Delta_n^2 = O\left(\frac{log(n)^{2(s-1)}}{n^2}\right)$$

Blue noise sampling



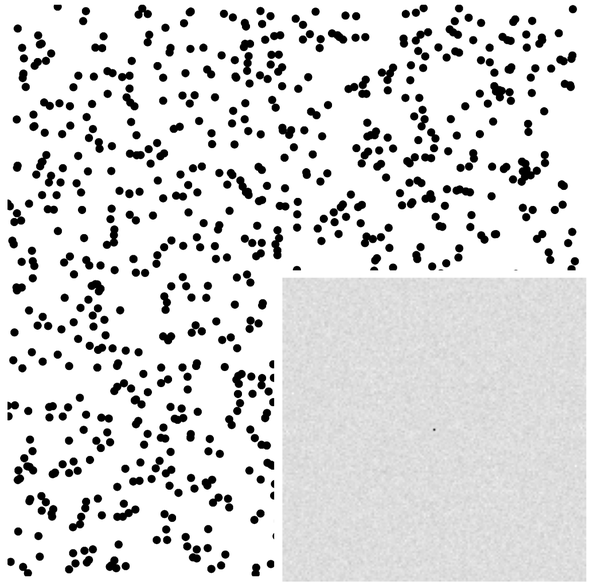
$$Var(I_n) = O\left(\frac{1}{n^{1+1/s}}\right)$$

Uniformity measure + error bound ⇒ convergence speed

Motivation: Monte Carlo Integration in 2d

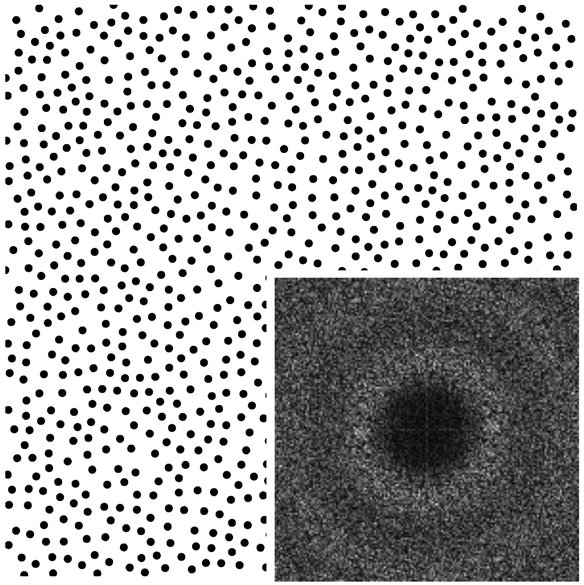
$$I_n := \frac{1}{n}\sum f(x_i) \qquad \Delta_n := \left| \int f dx - I_n \right|$$

Whitenoise



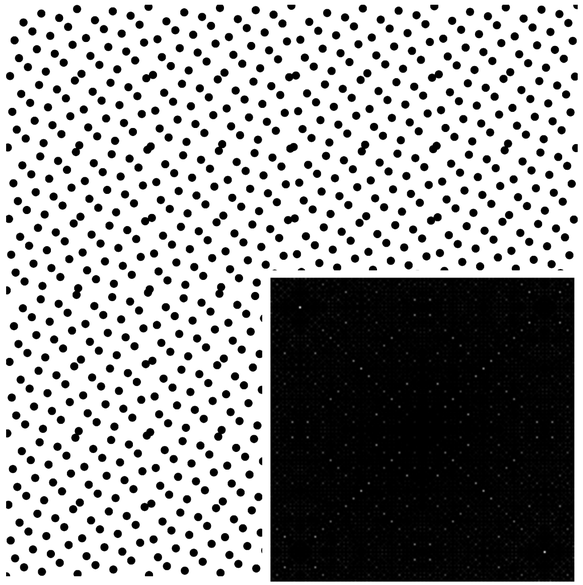
$$Var(I_n) = O\left(\frac{\sigma_f^2}{n}\right)$$

Poisson Disk



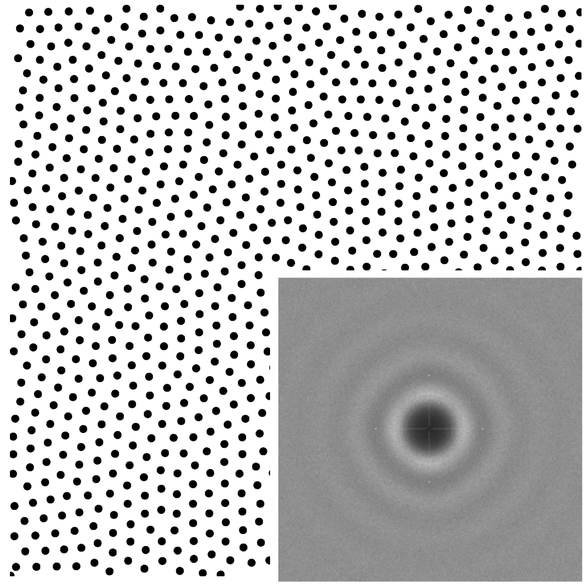
$$Var(I_n) = O\left(\frac{1}{n}\right)$$

Low discrepancy sequences



$$\Delta_n^2 = O\left(\frac{log(n)^{2(s-1)}}{n^2}\right)$$

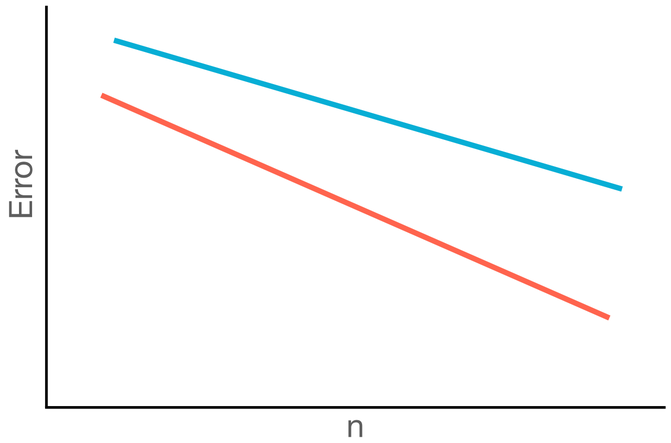
Blue noise sampling

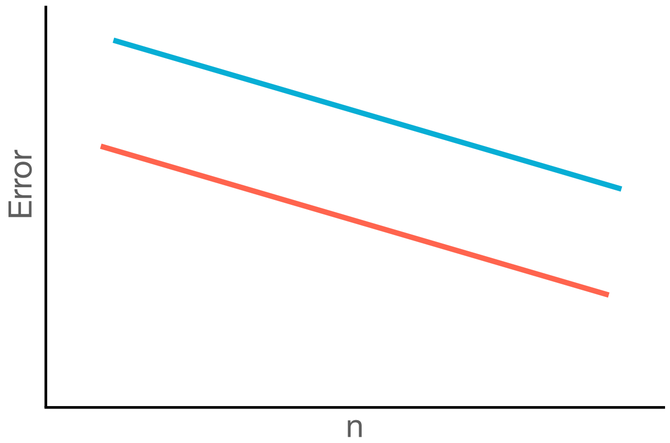


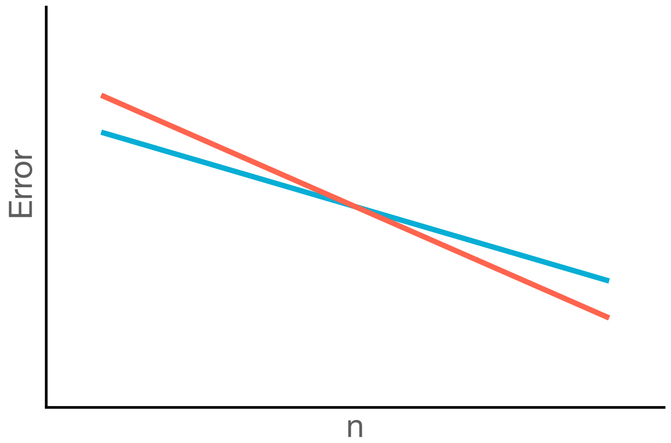
$$Var(I_n) = O\left(\frac{1}{n^{1+1/s}}\right)$$

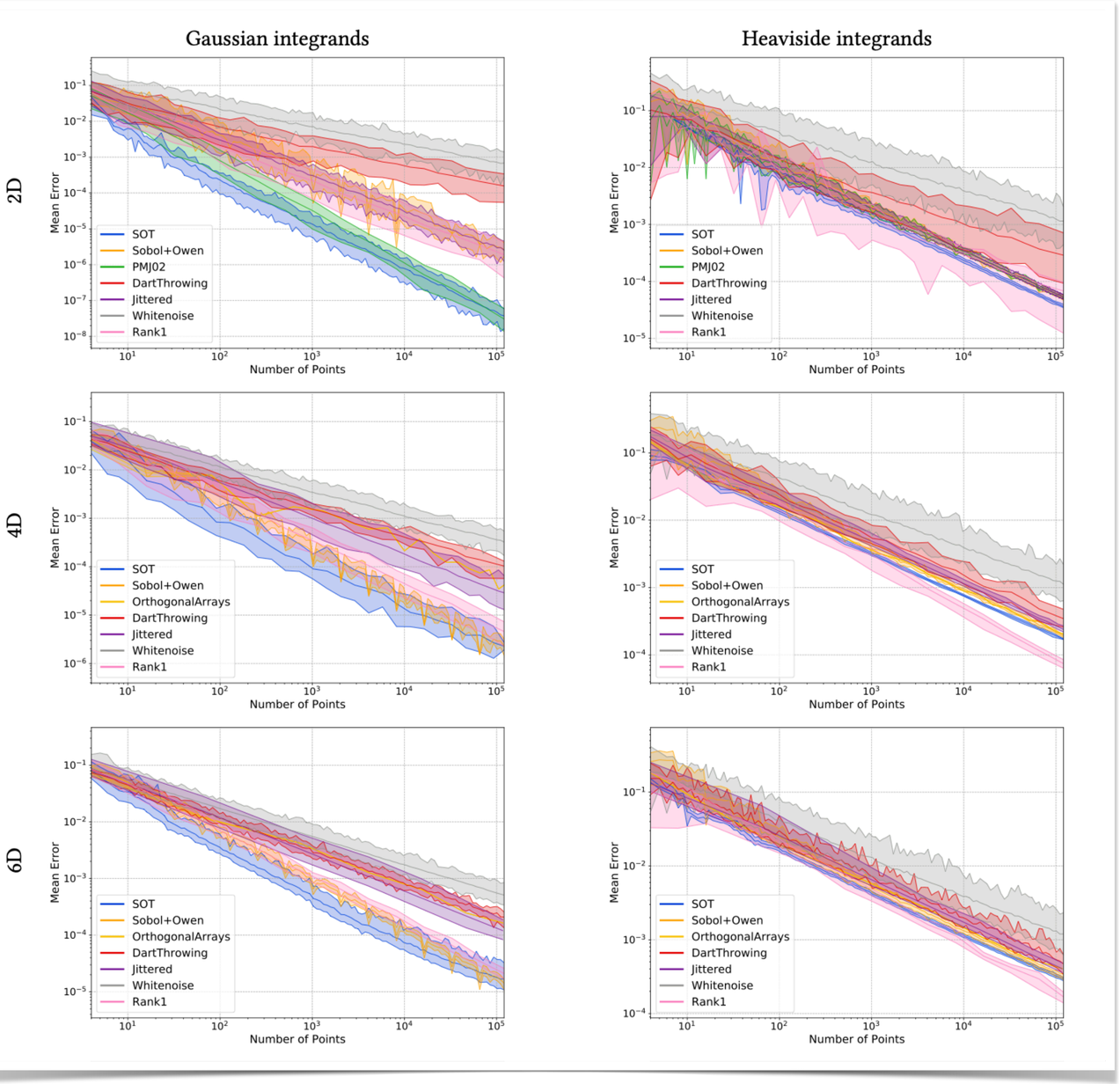
Uniformity measure + error bound ⇒ convergence speed

Convergence speed

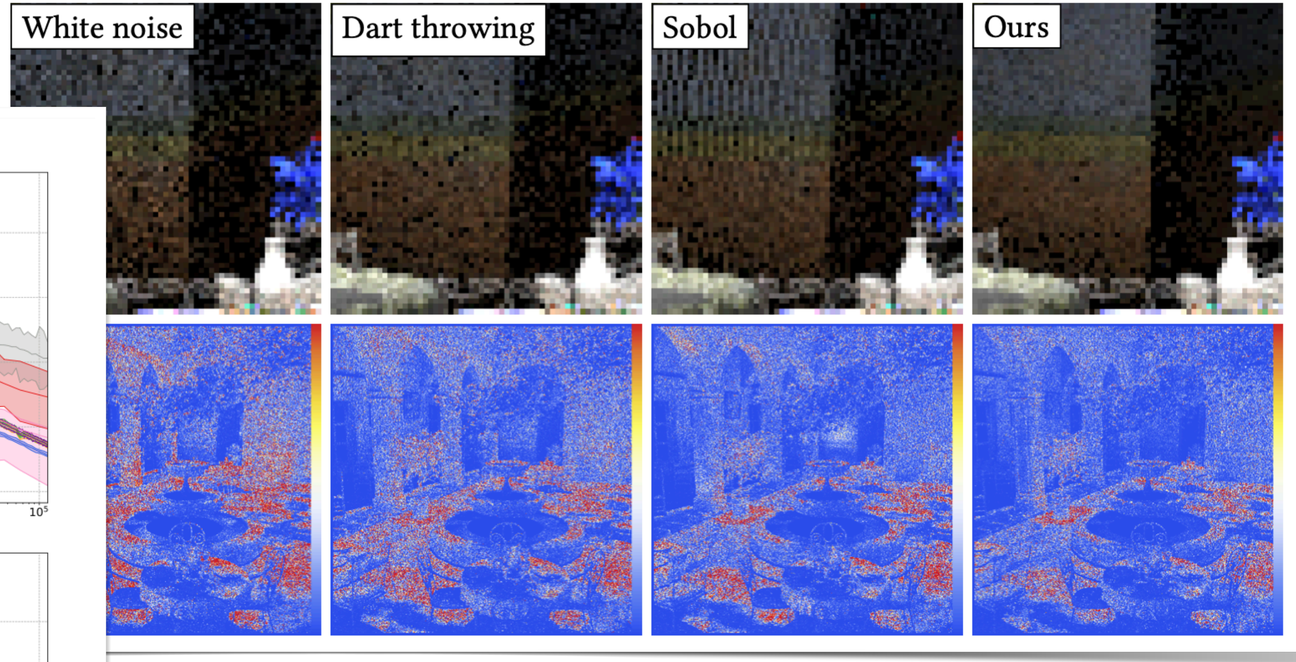


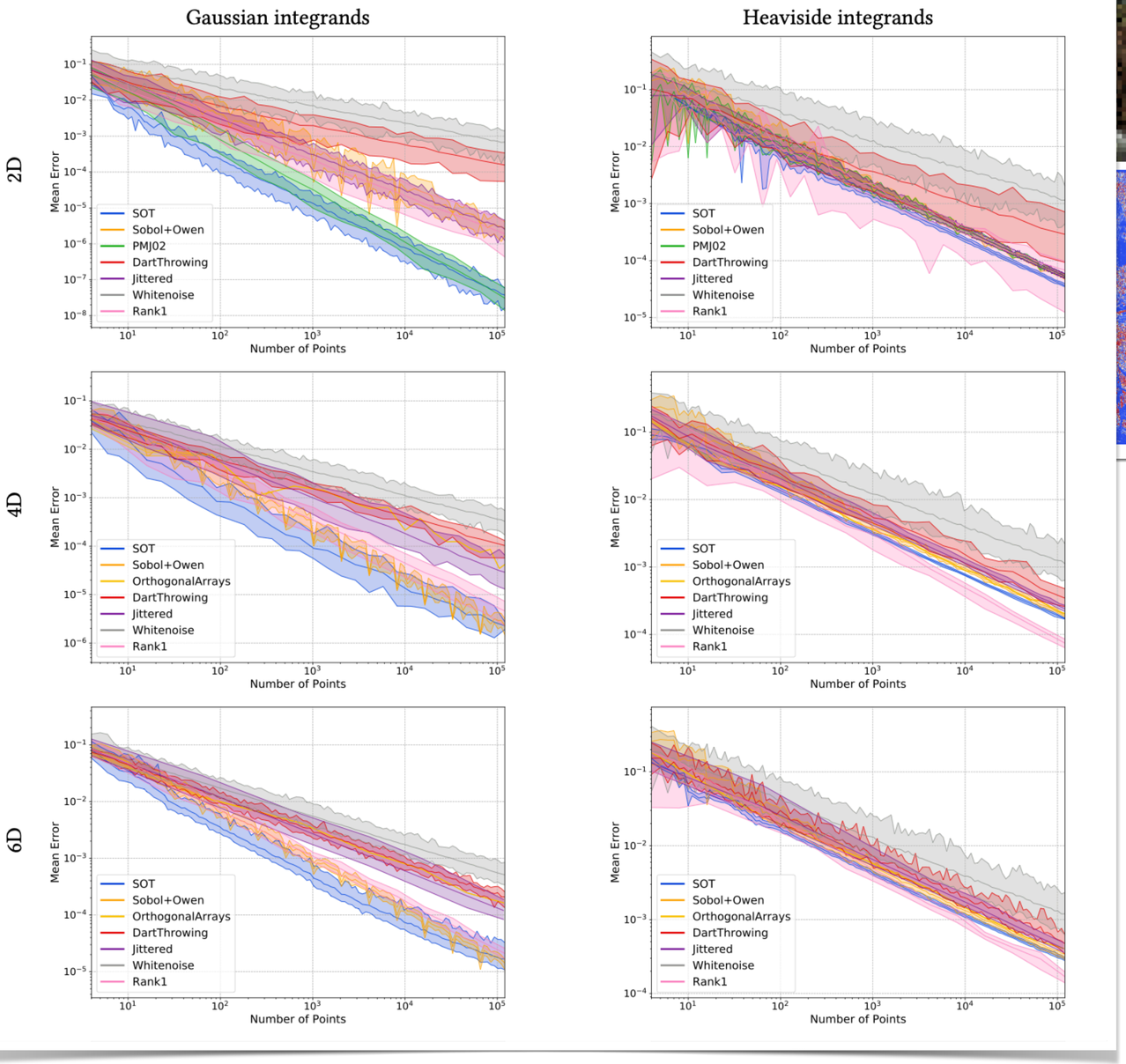


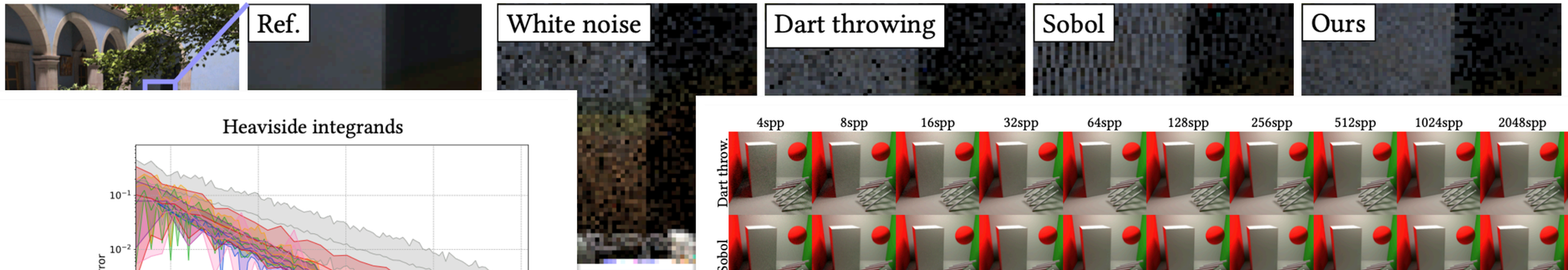


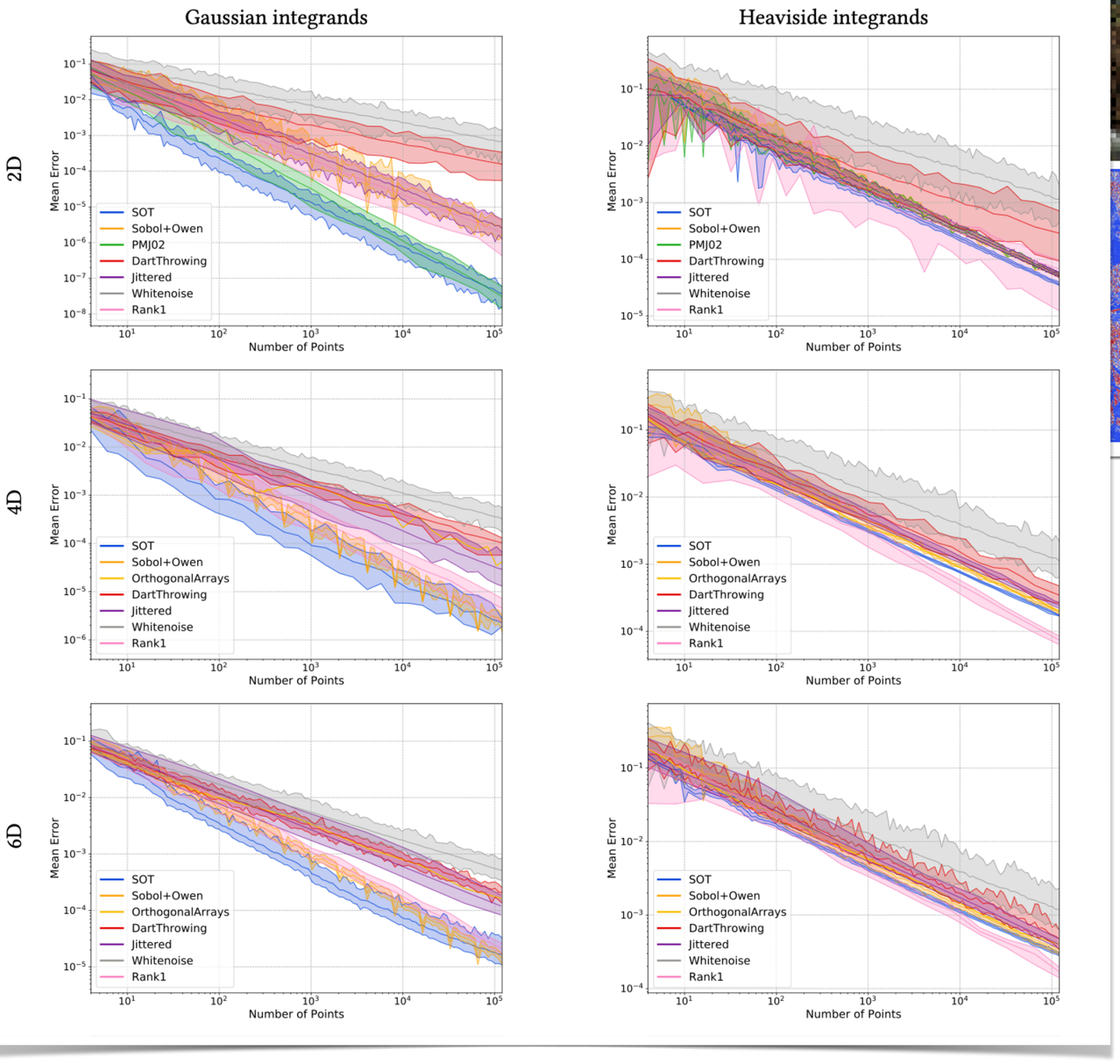

Gaussian integrands
Heaviside integrands
2D
4D
6D
Mean Error
Number of Points
SOT
Sobol+Owen
PMJ02
OrthogonalArrays
DartThrowing
Jittered
Whitenoise
Rank1

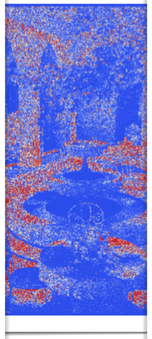

Ref.

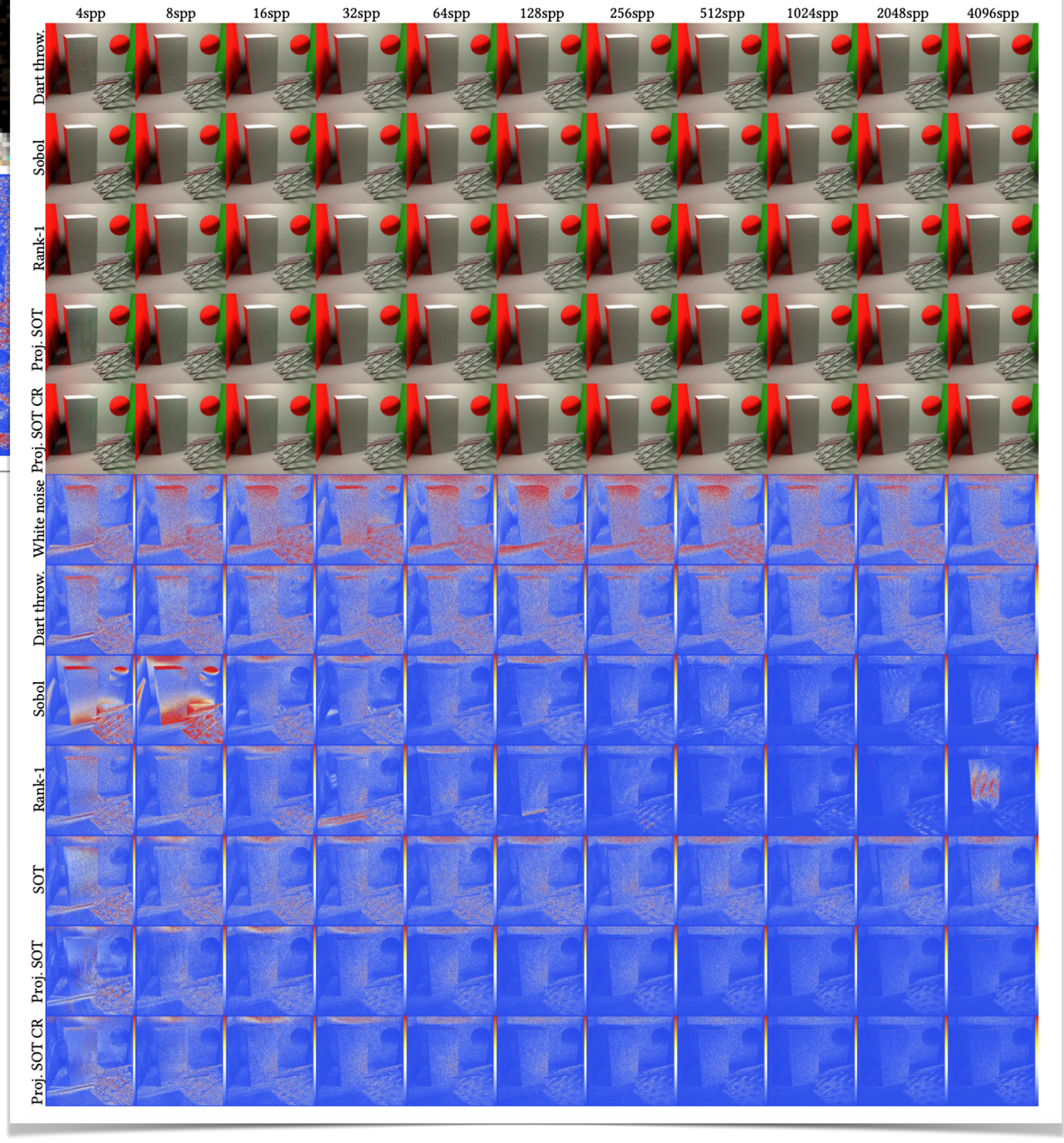

White noise
Dart throwing
Sobol
Ours

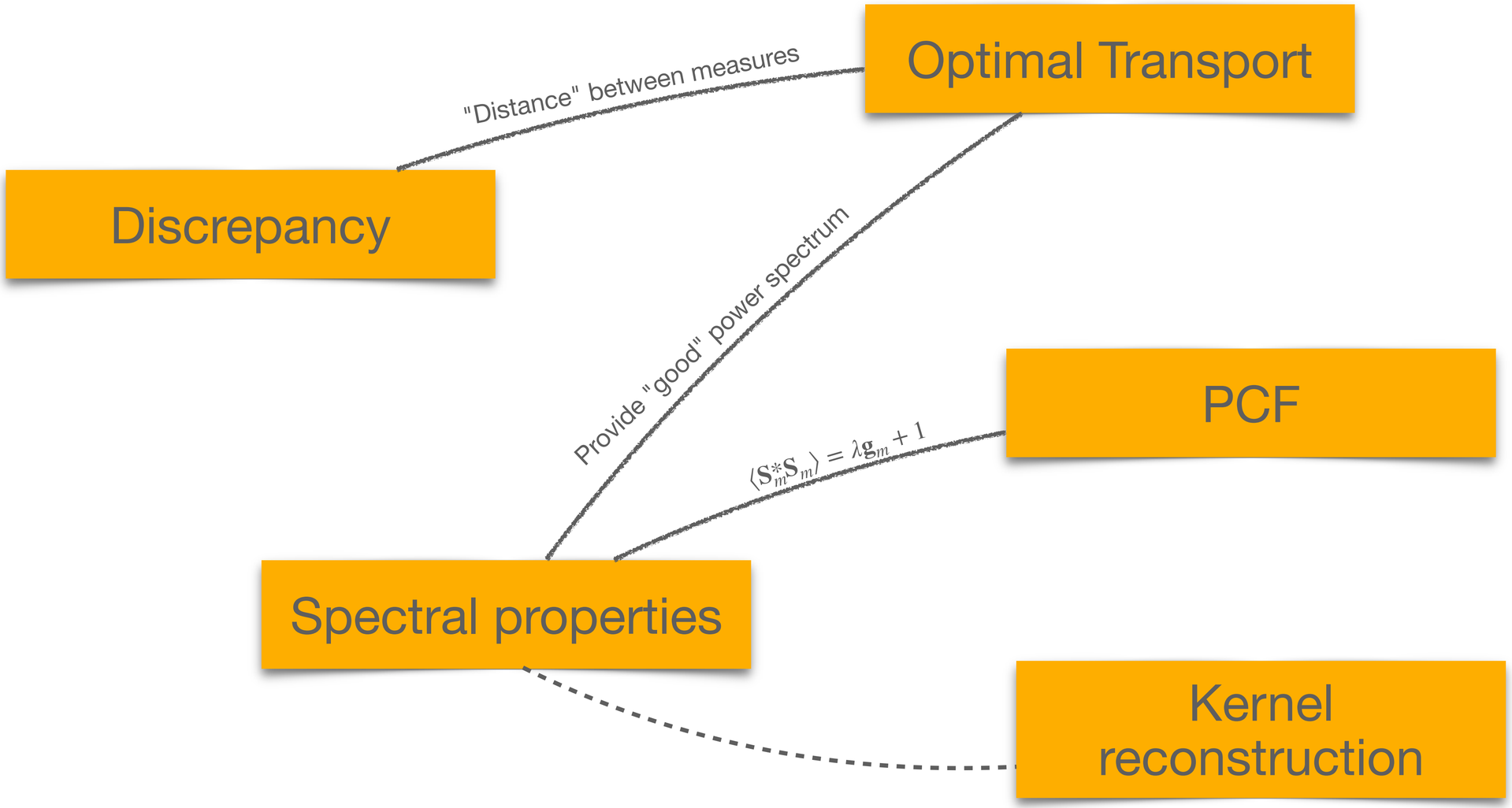

Gaussian integrands
Heaviside integrands
2D
4D
6D
Mean Error
Number of Points
SOT
Sobol+Owen
PMJ02
OrthogonalArrays
DartThrowing
Jittered
Whitenoise
Rank1


Ref.
White noise
Dart throwing
Sobol
Ours


Gaussian integrands
Heaviside integrands
2D
4D
6D
Mean Error
Number of Points
SOT
Sobol+Owen
PMJ02
OrthogonalArrays
DartThrowing
Jittered
Whitenoise
Rank1




4spp
8spp
16spp
32spp
64spp
128spp
256spp
512spp
1024spp
2048spp
4096spp
Dart throw.
Sobol
Rank-1
Proj. SOT
Proj. SOT CR
White noise
SOT


Optimal Transport
"Distance" between measures
Discrepancy
Provide "good" power spectrum
PCF
$\langle \mathbf{S}_m^* \mathbf{S}_m \rangle = \lambda \mathbf{g}_m + 1$
Spectral properties
Kernel reconstruction

Various routes

Various routes

- Low Discrepancy Sequences / Pointsets [Sobol, Halton, Faure, Niederreiter, Keller...]
    - fast (matrix/vector product in base 2, or Euclidean division), nD (-ish), $O(log(n)^{s-1}/n)$ integration error, could be randomized through Owen's scrambling

Various routes

- Low Discrepancy Sequences / Pointsets [Sobol, Halton, Faure, Niederreiter, Keller...]
    - fast (matrix/vector product in base 2, or Euclidean division), nD (-ish), $O(log(n)^{s-1}/n)$ integration error, could be randomized through Owen's scrambling

Discrepancy

Various routes

- Low Discrepancy Sequences / Pointsets [Sobol, Halton, Faure, Niederreiter, Keller...]
  - fast (matrix/vector product in base 2, or Euclidean division), nD (-ish), $O(log(n)^{s-1}/n)$ integration error, could be randomized through Owen's scrambling
- Poisson disk [Bridson 07, Wei 2008, Bowers 10, Guo et al 15, Ebeida 12...]
  - Good sample distributions for low sample counts, focus: fast algorithms (default is $O(n^2)$), genericity (anisotropic metric, on manifolds...)

Discrepancy

Various routes

- Low Discrepancy Sequences / Pointsets [Sobol, Halton, Faure, Niederreiter, Keller...]
  - fast (matrix/vector product in base 2, or Euclidean division), nD (-ish), $O(log(n)^{s-1}/n)$ integration error, could be randomized through Owen's scrambling
- Poisson disk [Bridson 07, Wei 2008, Bowers 10, Guo et al 15, Ebeida 12...]
  - Good sample distributions for low sample counts, focus: fast algorithms (default is $O(n^2)$), genericity (anisotropic metric, on manifolds...)

Discrepancy

PCF

Spectral properties

Various routes

- Low Discrepancy Sequences / Pointsets [Sobol, Halton, Faure, Niederreiter, Keller...]
  - fast (matrix/vector product in base 2, or Euclidean division), nD (-ish), $O(log(n)^{s-1}/n)$ integration error, could be randomized through Owen's scrambling
- Poisson disk [Bridson 07, Wei 2008, Bowers 10, Guo et al 15, Ebeida 12...]
  - Good sample distributions for low sample counts, focus: fast algorithms (default is $O(n^2)$), genericity (anisotropic metric, on manifolds...)
- Kernel based approaches [Fatall 11, Ahmed 20]
  - Good sample distributions for low sample counts, target: fast algorithm (default is $O(n^2)$)

Discrepancy

PCF

Spectral properties

Various routes

- Low Discrepancy Sequences / Pointsets [Sobol, Halton, Faure, Niederreiter, Keller...]
  - fast (matrix/vector product in base 2, or Euclidean division), nD (-ish), $O(log(n)^{s-1}/n)$ integration error, could be randomized through Owen's scrambling
- Poisson disk [Bridson 07, Wei 2008, Bowers 10, Guo et al 15, Ebeida 12...]
  - Good sample distributions for low sample counts, focus: fast algorithms (default is $O(n^2)$), genericity (anisotropic metric, on manifolds...)
- Kernel based approaches [Fatall 11, Ahmed 20]
  - Good sample distributions for low sample counts, target: fast algorithm (default is $O(n^2)$)

Discrepancy

PCF

Spectral properties

Spectral properties

Various routes

- Low Discrepancy Sequences / Pointsets [Sobol, Halton, Faure, Niederreiter, Keller...]
  - fast (matrix/vector product in base 2, or Euclidean division), nD (-ish), $O(log(n)^{s-1}/n)$ integration error, could be randomized through Owen's scrambling
- Poisson disk [Bridson 07, Wei 2008, Bowers 10, Guo et al 15, Ebeida 12...]
  - Good sample distributions for low sample counts, focus: fast algorithms (default is $O(n^2)$), genericity (anisotropic metric, on manifolds...)
- Kernel based approaches [Fatall 11, Ahmed 20]
  - Good sample distributions for low sample counts, target: fast algorithm (default is $O(n^2)$)
- Blue noise from Optimal Transport [de Goes et al 12, Qin et al 17, Paulin 20]
  - Targeting blue-noise like power spectrum, versatile tool (to sample non-uniform pdf) but numerically expensive in higher dimensions

Discrepancy

PCF

Spectral properties

Spectral properties

Various routes

- Low Discrepancy Sequences / Pointsets [Sobol, Halton, Faure, Niederreiter, Keller...]
  - fast (matrix/vector product in base 2, or Euclidean division), nD (-ish), $O(log(n)^{s-1}/n)$ integration error, could be randomized through Owen's scrambling
- Poisson disk [Bridson 07, Wei 2008, Bowers 10, Guo et al 15, Ebeida 12...]
  - Good sample distributions for low sample counts, focus: fast algorithms (default is $O(n^2)$), genericity (anisotropic metric, on manifolds...)
- Kernel based approaches [Fatall 11, Ahmed 20]
  - Good sample distributions for low sample counts, target: fast algorithm (default is $O(n^2)$)
- Blue noise from Optimal Transport [de Goes et al 12, Qin et al 17, Paulin 20]
  - Targeting blue-noise like power spectrum, versatile tool (to sample non-uniform pdf) but numerically expensive in higher dimensions

Discrepancy

PCF

Spectral properties

Spectral properties

Optimal Transport

Various routes

- Low Discrepancy Sequences / Pointsets [Sobol, Halton, Faure, Niederreiter, Keller...]
  - fast (matrix/vector product in base 2, or Euclidean division), nD (-ish), $O(log(n)^{s-1}/n)$ integration error, could be randomized through Owen's scrambling
- Poisson disk [Bridson 07, Wei 2008, Bowers 10, Guo et al 15, Ebeida 12...]
  - Good sample distributions for low sample counts, focus: fast algorithms (default is $O(n^2)$), genericity (anisotropic metric, on manifolds...)
- Kernel based approaches [Fatall 11, Ahmed 20]
  - Good sample distributions for low sample counts, target: fast algorithm (default is $O(n^2)$)
- Blue noise from Optimal Transport [de Goes et al 12, Qin et al 17, Paulin 20]
  - Targeting blue-noise like power spectrum, versatile tool (to sample non-uniform pdf) but numerically expensive in higher dimensions
- Mixed solutions

Discrepancy

PCF

Spectral properties

Spectral properties

Optimal Transport

Various routes

- Low Discrepancy Sequences / Pointsets [Sobol, Halton, Faure, Niederreiter, Keller...]
  - fast (matrix/vector product in base 2, or Euclidean division), nD (-ish), $O(log(n)^{s-1}/n)$ integration error, could be randomized through Owen's scrambling

Discrepancy

- Poisson disk [Bridson 07, Wei 2008, Bowers 10, Guo et al 15, Ebeida 12...]
  - Good sample distributions for low sample counts, focus: fast algorithms (default is $O(n^2)$), genericity (anisotropic metric, on manifolds...)

PCF Spectral properties

- Kernel based approaches [Fatall 11, Ahmed 20]
  - Good sample distributions for low sample counts, target: fast algorithm (default is $O(n^2)$)

Spectral properties

- Blue noise from Optimal Transport [de Goes et al 12, Qin et al 17, Paulin 20]
  - Targeting blue-noise like power spectrum, versatile tool (to sample non-uniform pdf) but numerically expensive in higher dimensions

Optimal Transport

- Mixed solutions
  - Optimized LDS with Blue-Noise targets [Ahmed 16, Perrier 18, Ahmed 20]

Discrepancy PCF Spectral properties

Various routes

- Low Discrepancy Sequences / Pointsets [Sobol, Halton, Faure, Niederreiter, Keller...]
  - fast (matrix/vector product in base 2, or Euclidean division), nD (-ish), $O(log(n)^{s-1}/n)$ integration error, could be randomized through Owen's scrambling
- Poisson disk [Bridson 07, Wei 2008, Bowers 10, Guo et al 15, Ebeida 12...]
  - Good sample distributions for low sample counts, focus: fast algorithms (default is $O(n^2)$), genericity (anisotropic metric, on manifolds...)
- Kernel based approaches [Fatall 11, Ahmed 20]
  - Good sample distributions for low sample counts, target: fast algorithm (default is $O(n^2)$)
- Blue noise from Optimal Transport [de Goes et al 12, Qin et al 17, Paulin 20]
  - Targeting blue-noise like power spectrum, versatile tool (to sample non-uniform pdf) but numerically expensive in higher dimensions
- Mixed solutions
  - Optimized LDS with Blue-Noise targets [Ahmed 16, Perrier 18, Ahmed 20]
  - Tiled based Blue-noise for fast (adaptive) sampling [Ostromoukhov, , Wachtel 14]

Discrepancy

PCF

Spectral properties

Spectral properties

Optimal Transport

Discrepancy

PCF

Spectral properties

Aperiodic tilings +

Spectral properties

Various routes

- Low Discrepancy Sequences / Pointsets [Sobol, Halton, Faure, Niederreiter, Keller...]
  - fast (matrix/vector product in base 2, or Euclidean division), nD (-ish), $O(log(n)^{s-1}/n)$ integration error, could be randomized through Owen's scrambling
- Poisson disk [Bridson 07, Wei 2008, Bowers 10, Guo et al 15, Ebeida 12...]
  - Good sample distributions for low sample counts, focus: fast algorithms (default is $O(n^2)$), genericity (anisotropic metric, on manifolds...)
- Kernel based approaches [Fatall 11, Ahmed 20]
  - Good sample distributions for low sample counts, target: fast algorithm (default is $O(n^2)$)
- Blue noise from Optimal Transport [de Goes et al 12, Qin et al 17, Paulin 20]
  - Targeting blue-noise like power spectrum, versatile tool (to sample non-uniform pdf) but numerically expensive in higher dimensions
- Mixed solutions
  - Optimized LDS with Blue-Noise targets [Ahmed 16, Perrier 18, Ahmed 20]
  - Tiled based Blue-noise for fast (adaptive) sampling [Ostromoukhov, , Wachtel 14]
  - Projective sampling (with Blue-noise or LDS targets) [Reinert 15, Paulin 21]

Discrepancy

PCF

Spectral properties

Spectral properties

Optimal Transport

Discrepancy

PCF

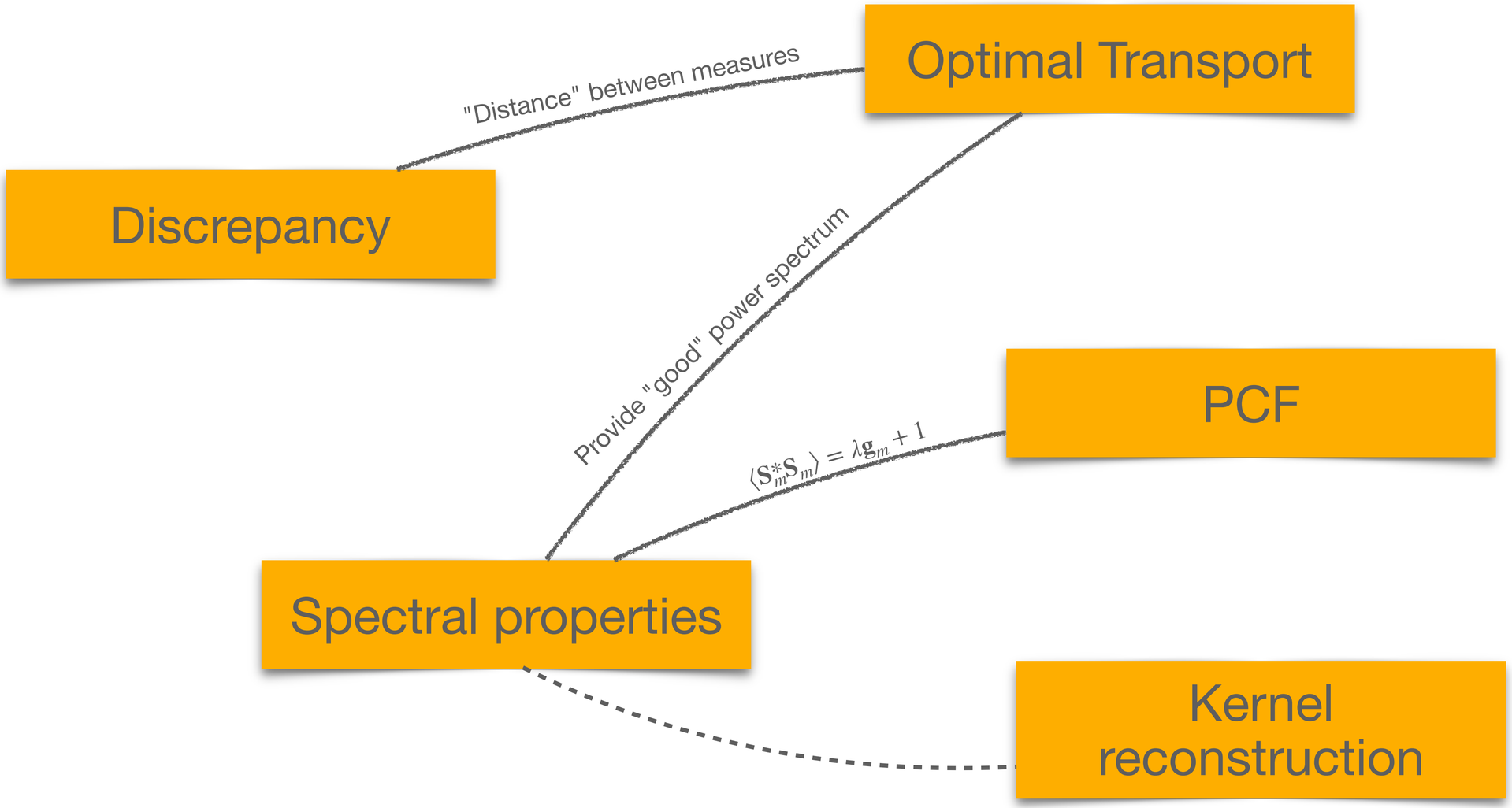
Spectral properties

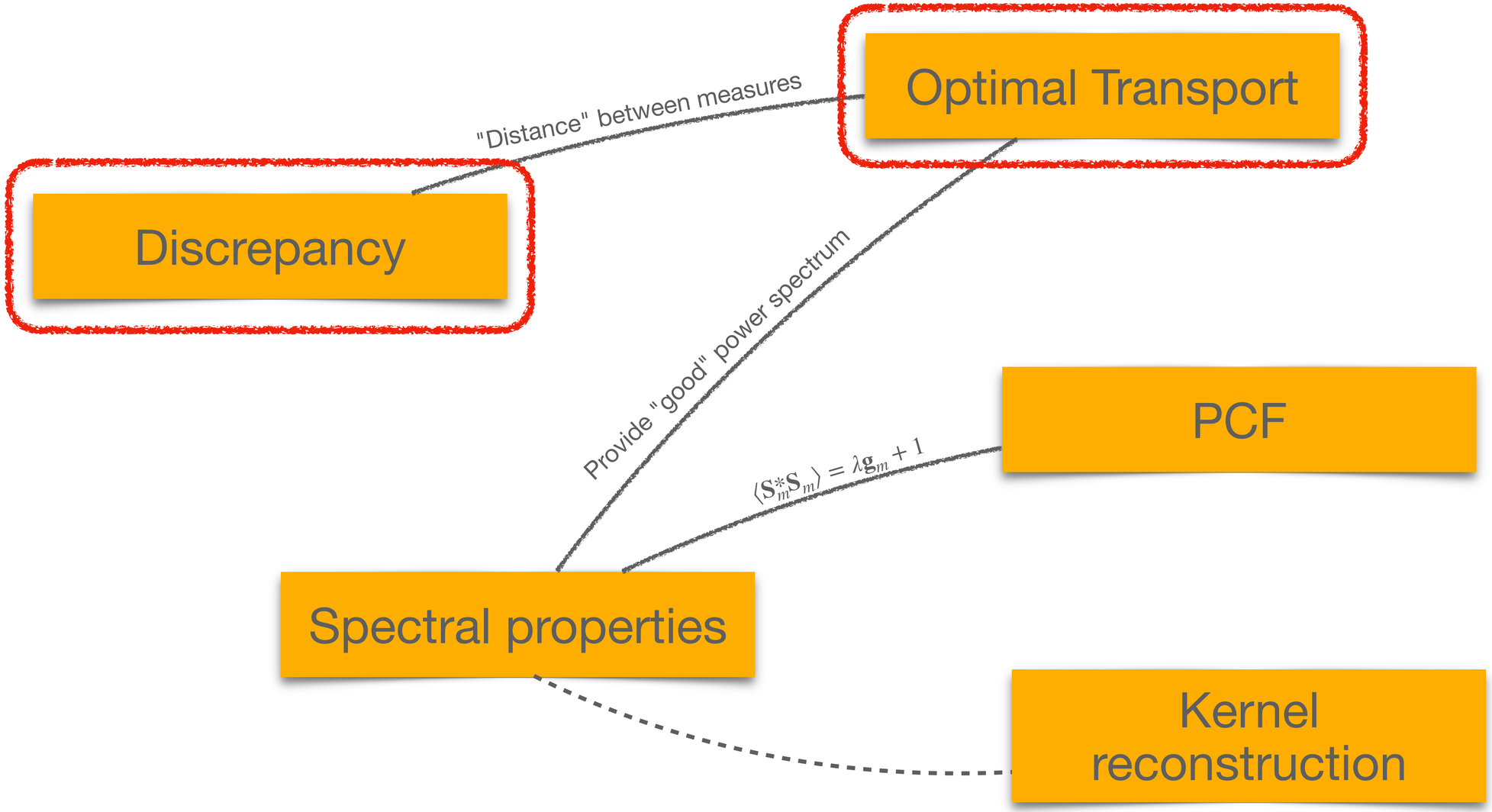
Aperiodic tilings +

Spectral properties

Discrepancy

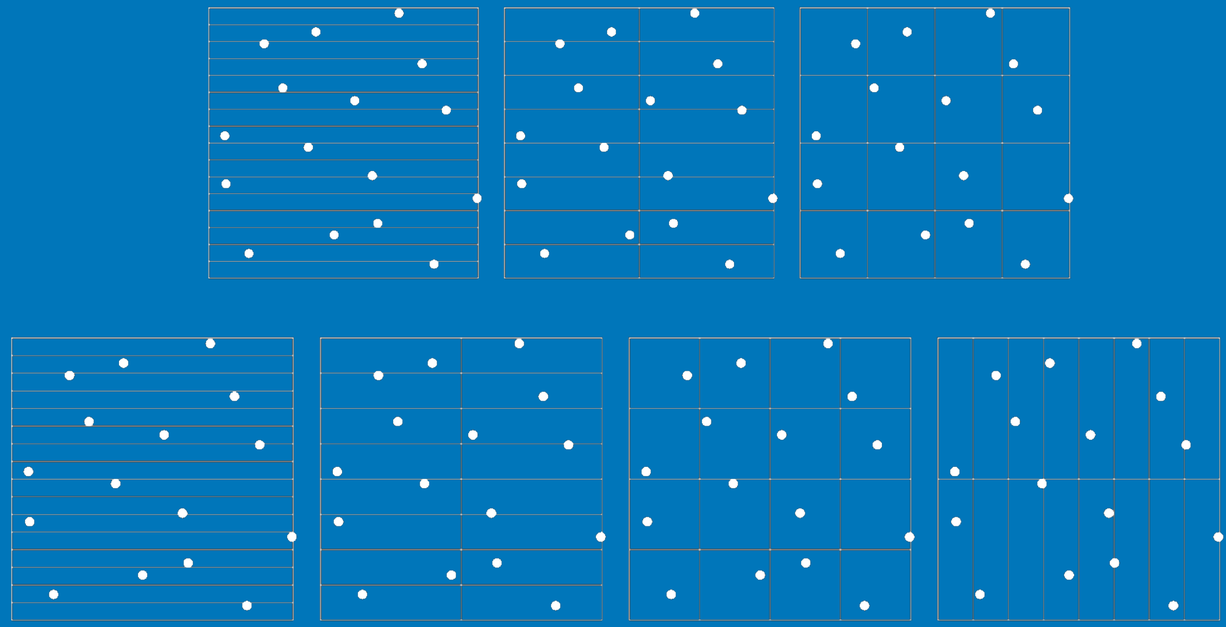
Spectral properties


Optimal Transport
"Distance" between measures
Discrepancy
Provide "good" power spectrum
PCF
$\langle \mathbf{S}_m^* \mathbf{S}_m \rangle = \lambda \mathbf{g}_m + 1$
Spectral properties
Kernel reconstruction

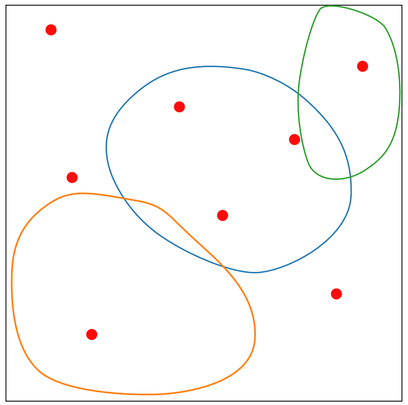

Optimal Transport
"Distance" between measures
Discrepancy
Provide "good" power spectrum
$\langle \mathbf{S}_m^* \mathbf{S}_m \rangle = \lambda \mathbf{g}_m + 1$
PCF
Spectral properties
Kernel reconstruction

UTK

Low discrepancy samplers



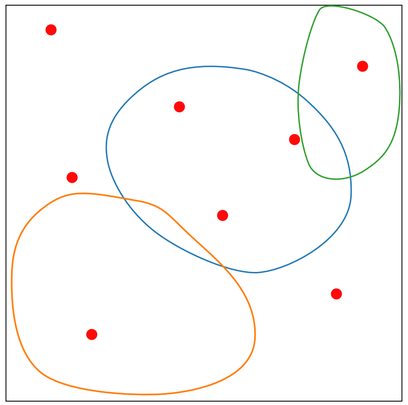
Spatial measures: Discrepancy



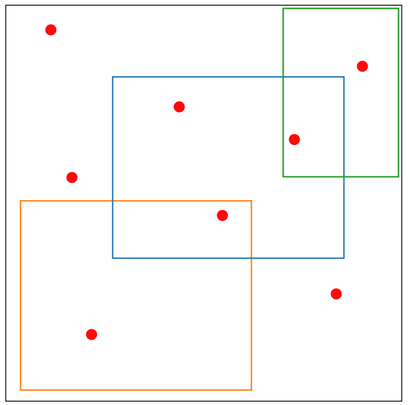
Discrepancy

$$D(X_n, \Omega) := \left| \lambda_2(\Omega) - \frac{A(\Omega, X_n)}{n} \right| \qquad D(X_n) := \max_{\Omega \subset [0,1)^2} D(X_n, \Omega)$$

Spatial measures: Discrepancy



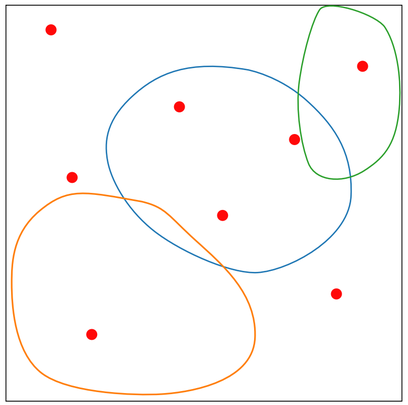
Discrepancy



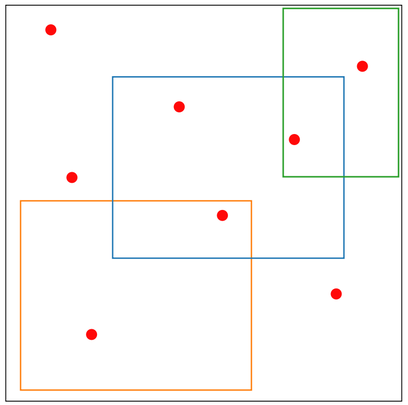
Extreme Discrepancy

$$D(X_n, \Omega) := \left| \lambda_2(\Omega) - \frac{A(\Omega, X_n)}{n} \right| \qquad D(X_n) := \max_{\Omega \subset [0,1)^2} D(X_n, \Omega)$$

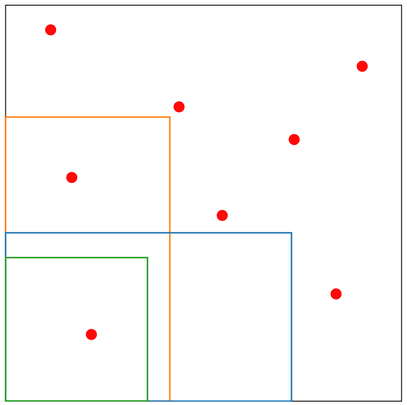
Spatial measures: Discrepancy



Discrepancy



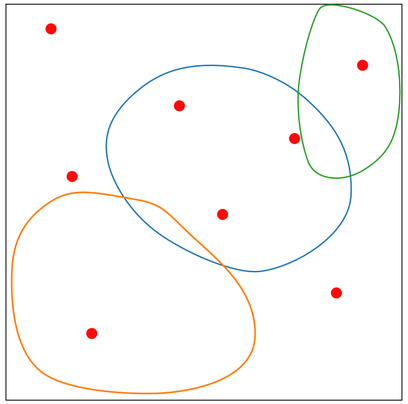
Extreme Discrepancy



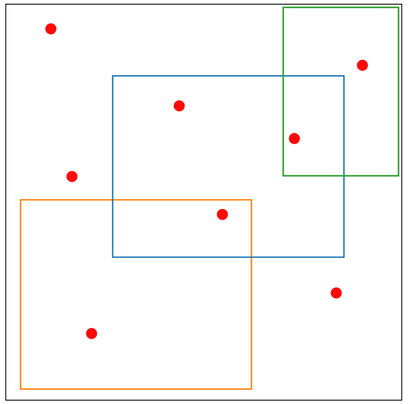
Star Discrepancy

$$D(X_n, \Omega) := \left| \lambda_2(\Omega) - \frac{A(\Omega, X_n)}{n} \right| \qquad D(X_n) := \max_{\Omega \subset [0,1)^2} D(X_n, \Omega)$$

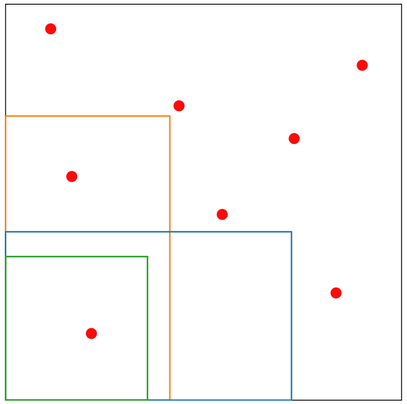
Spatial measures: Discrepancy



Discrepancy



Extreme Discrepancy



Star Discrepancy

- Extensions to 1d Kolmogorov-Smirnov test
- Equivalent measures
- Many variants, numerical approximations...
- [Koksma-Hlawka] inequality:

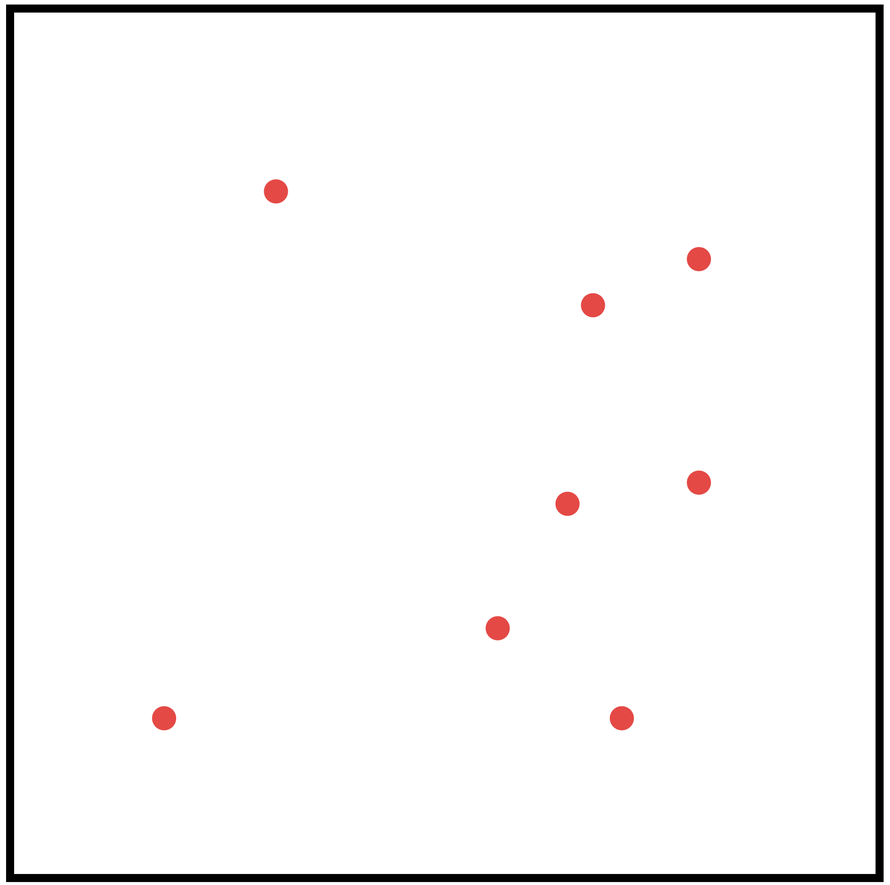
$$\Delta_n \leq V(f) \cdot D(X_n)$$

[Koksma 42, Hlawska 61]

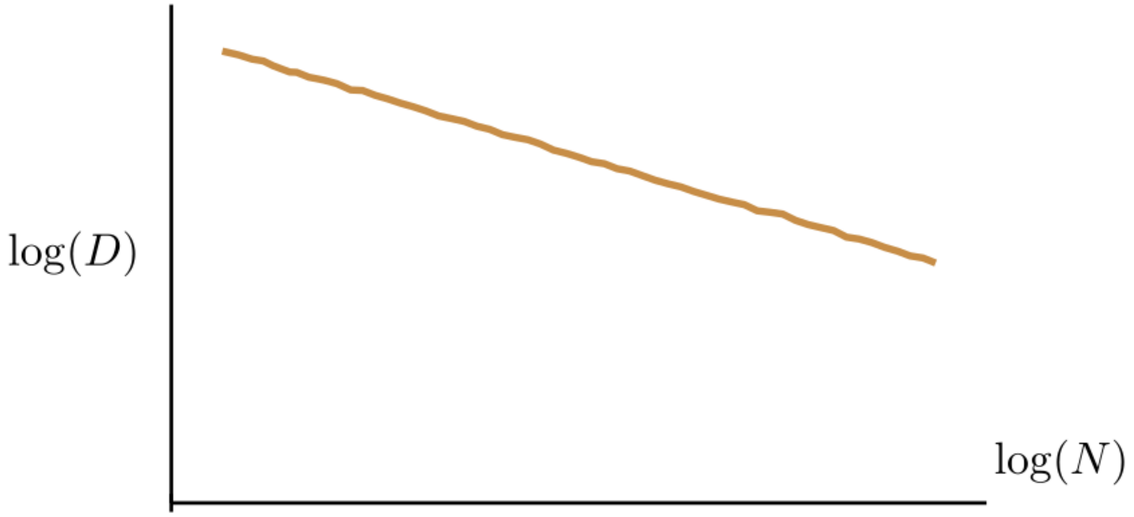
$$D(X_n, \Omega) := \left| \lambda_2(\Omega) - \frac{A(\Omega, X_n)}{n} \right|$$

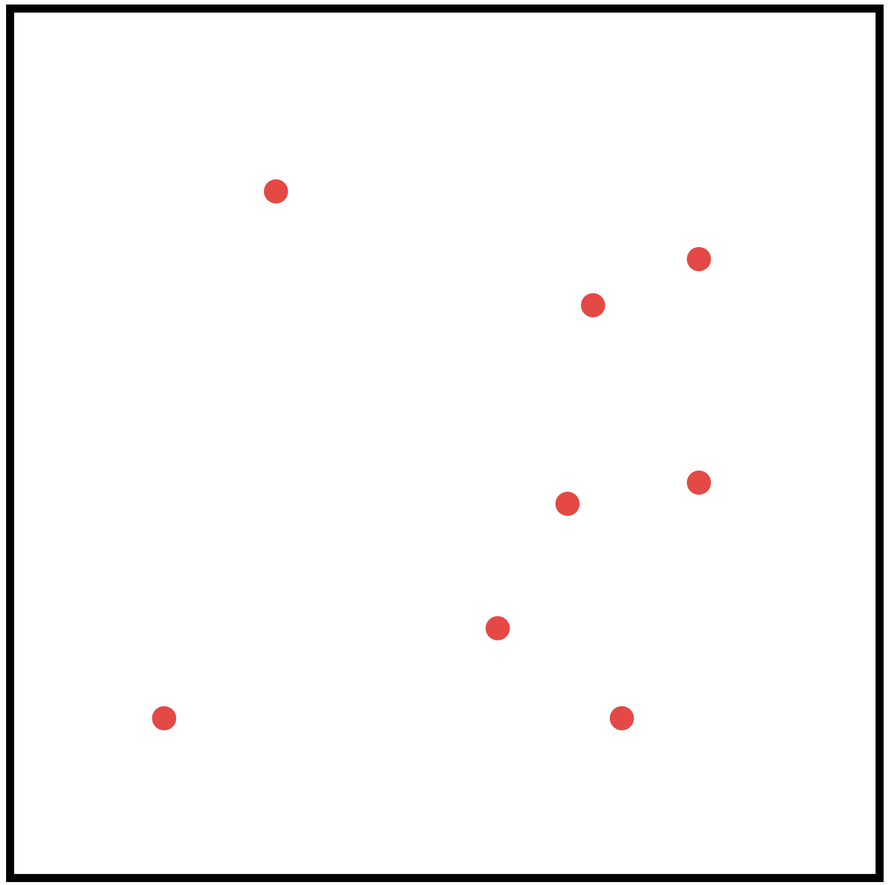
$$D(X_n) := \max_{\Omega \subset [0,1)^2} D(X_n, \Omega)$$

Recap stratified sampling

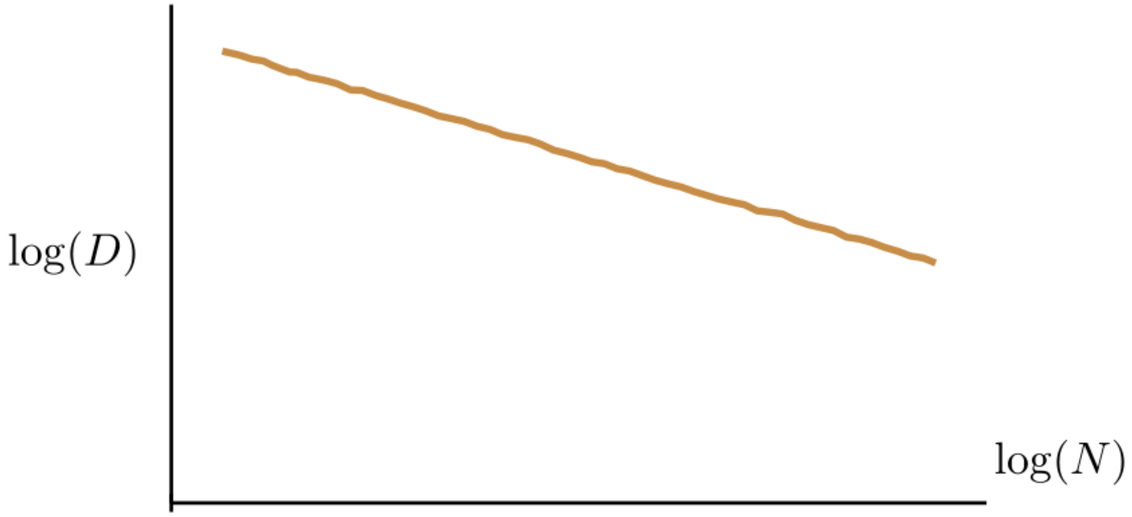


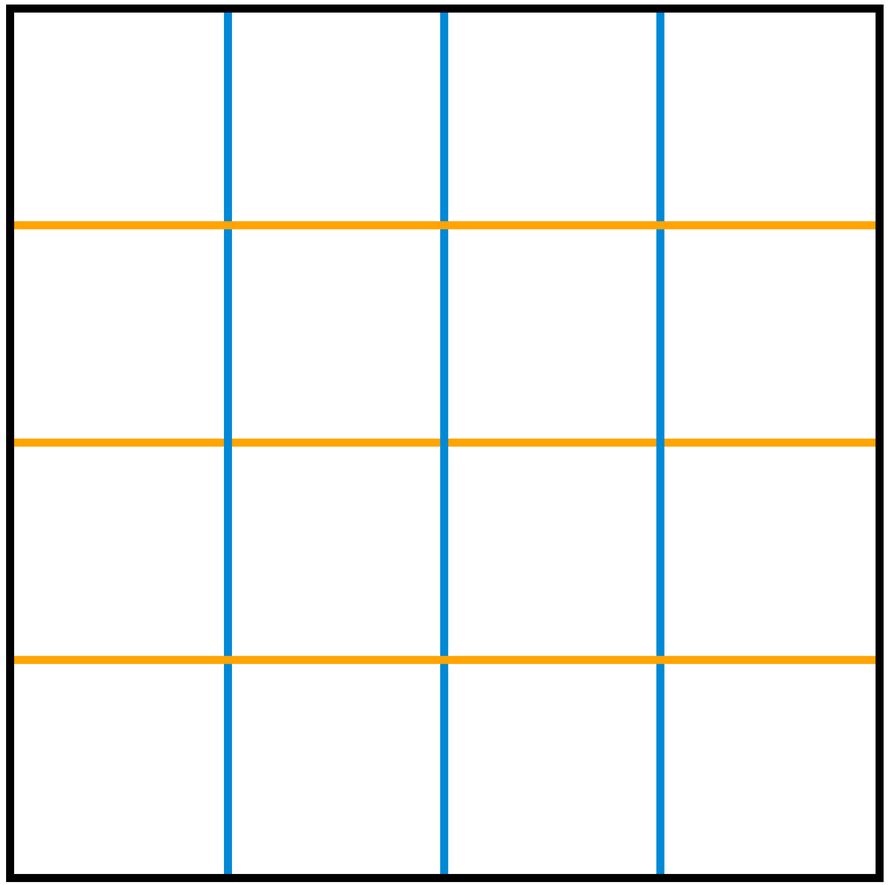
Recap stratified sampling



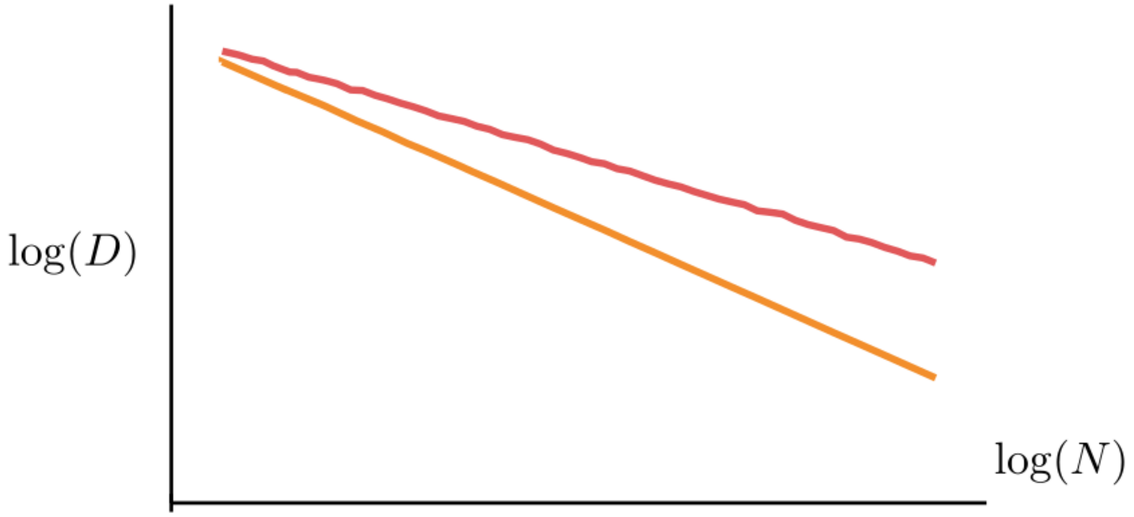


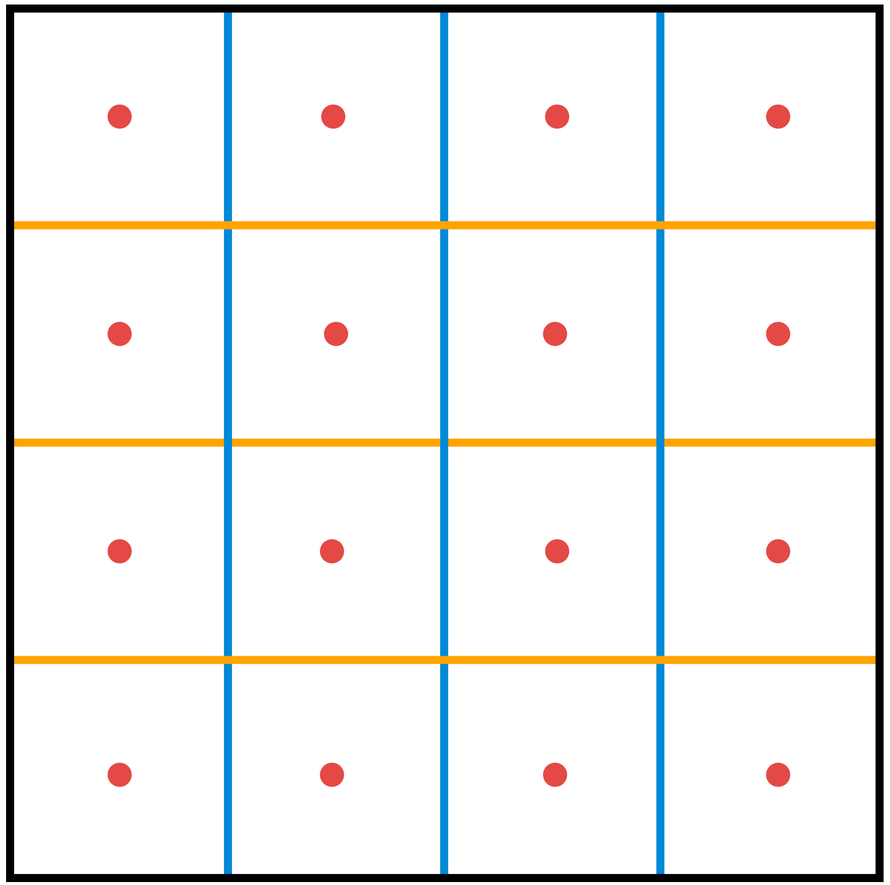
Recap stratified sampling





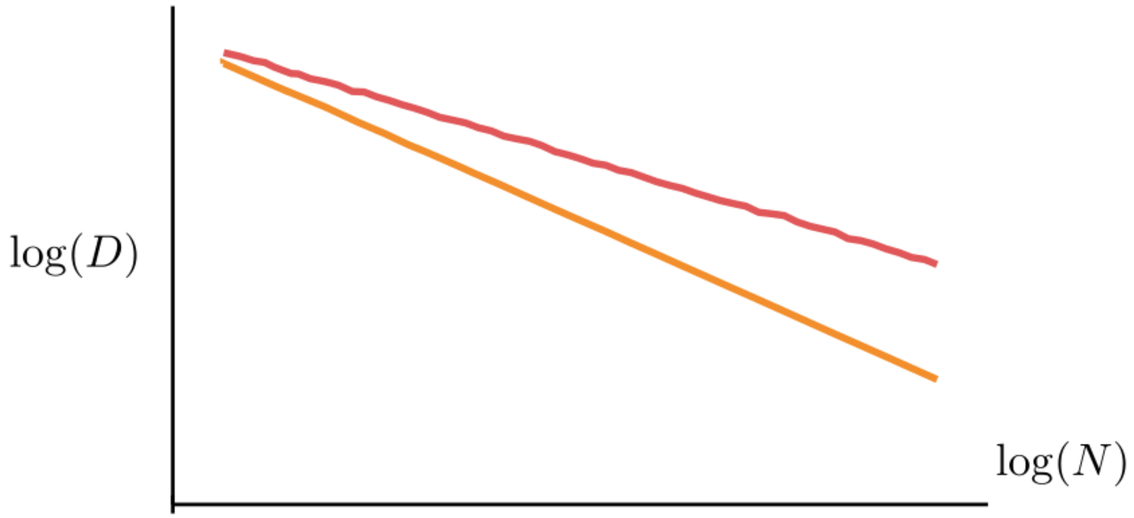
Recap stratified sampling

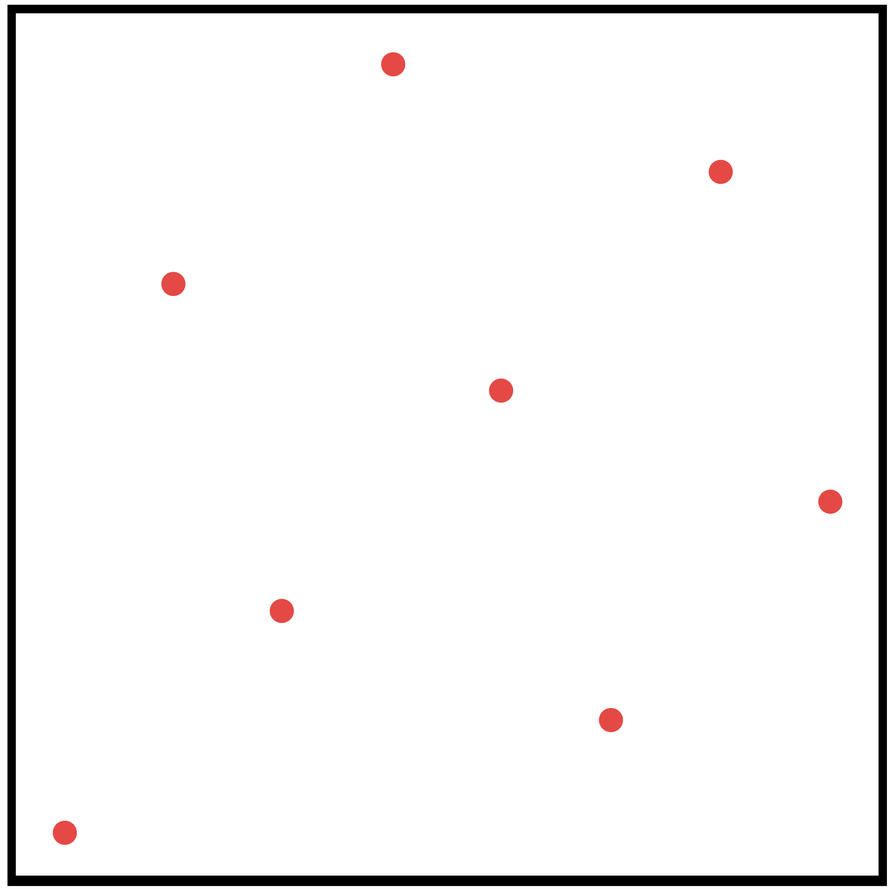




(t,m,s)-nets

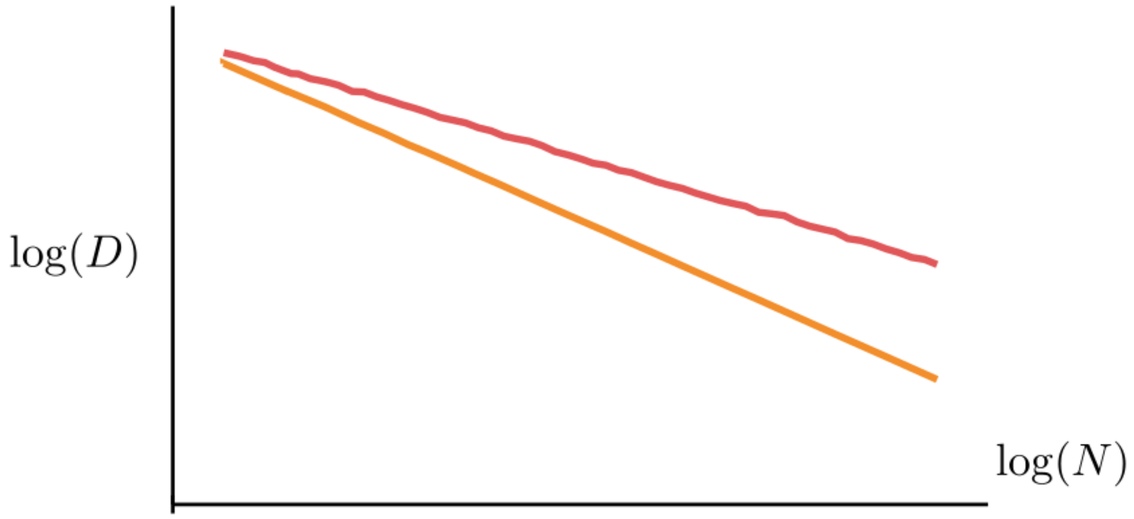
t = 0, m = 3, s = 2, p = 2

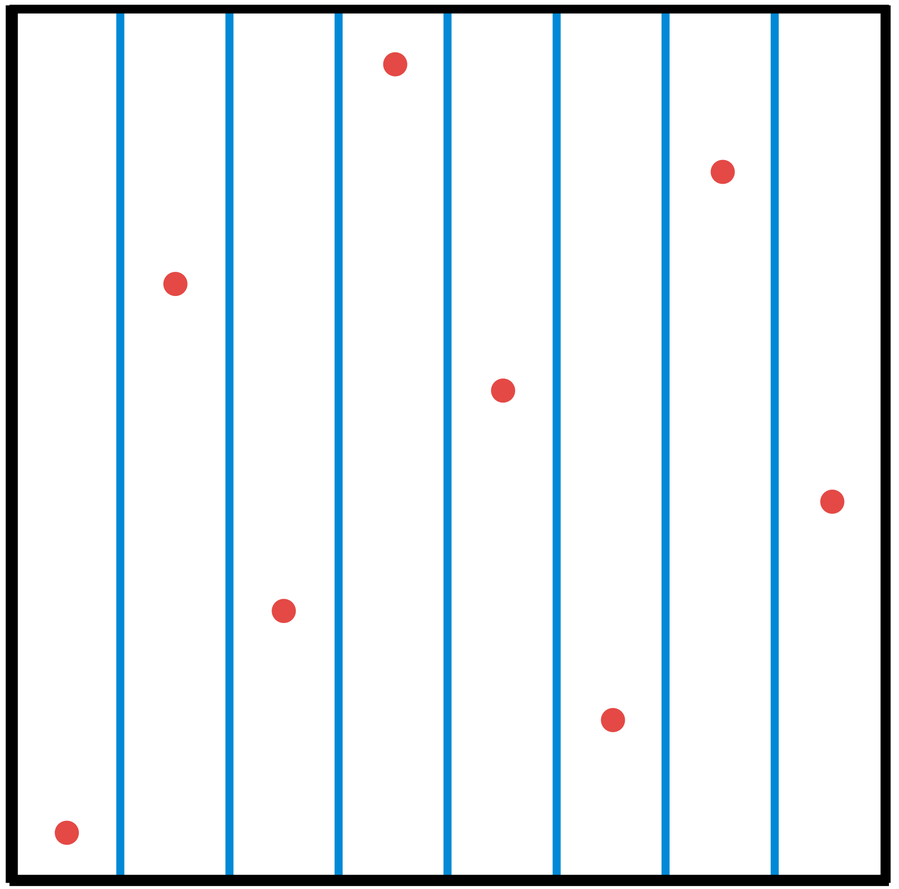




(t,m,s)-nets

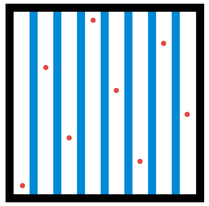
t = 0, m = 3, s = 2, p = 2

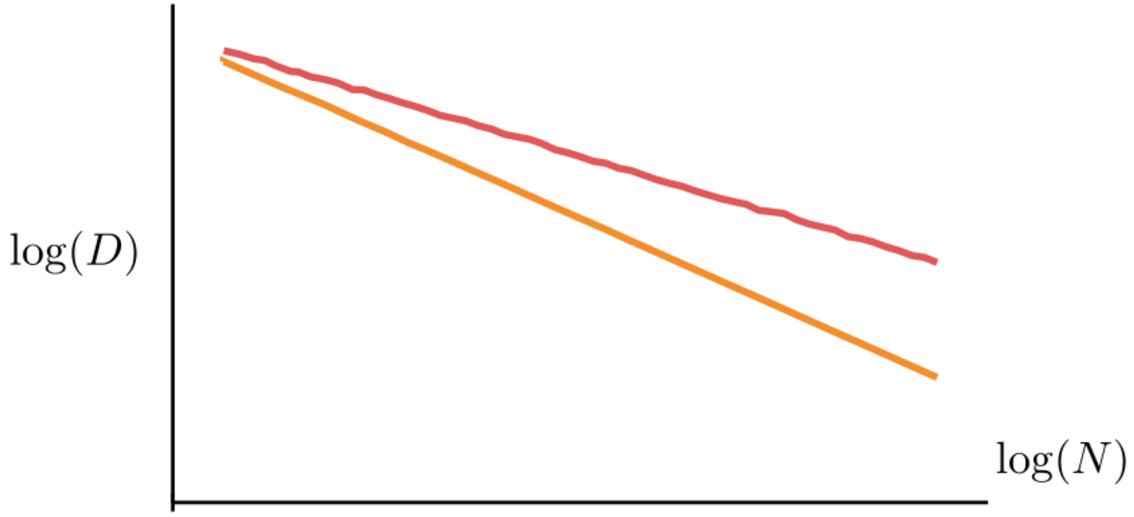


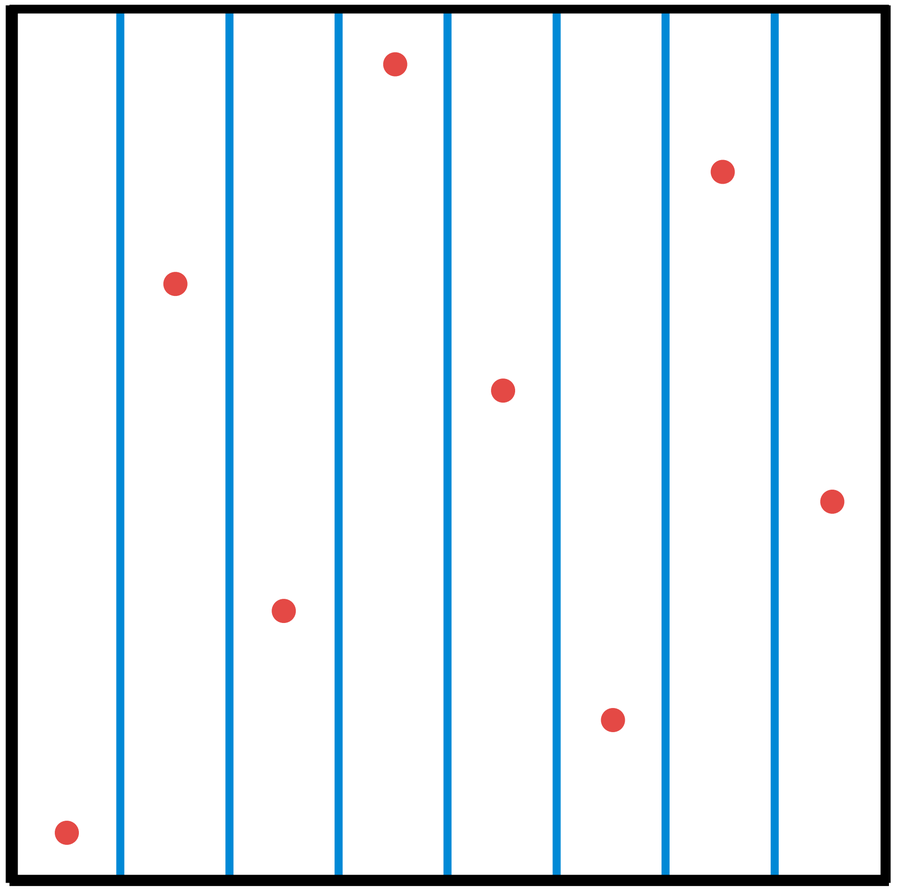


(t,m,s)-nets

t = 0, m = 3, s = 2, p = 2

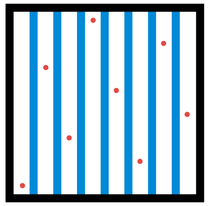


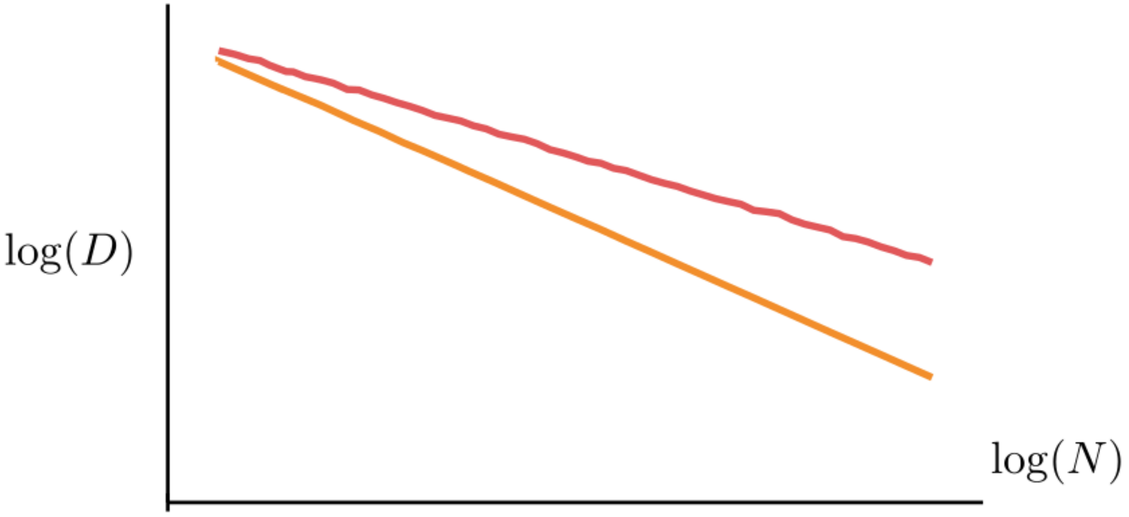


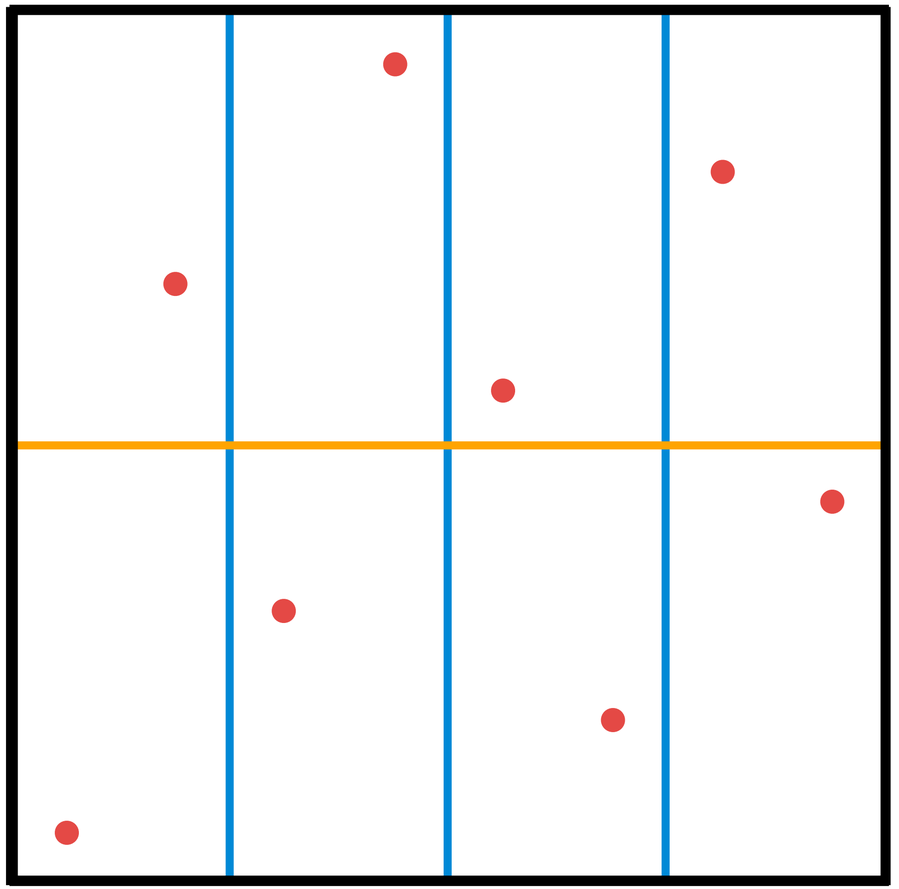


(t,m,s)-nets

t = 0, m = 3, s = 2, p = 2

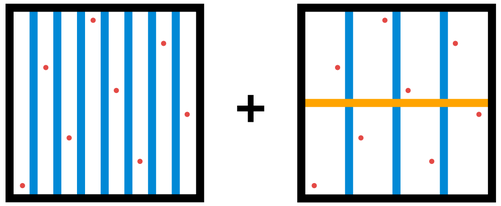


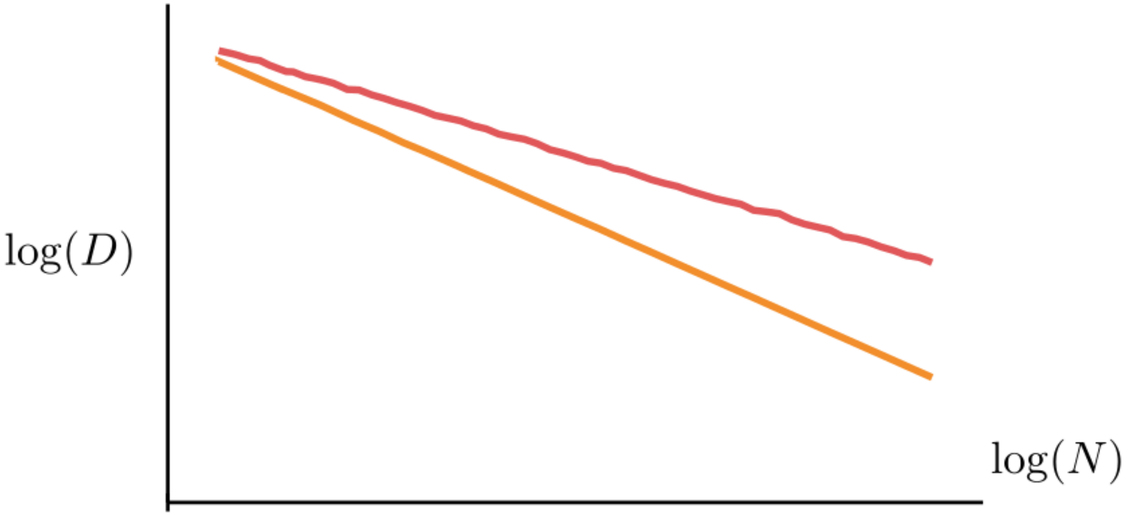


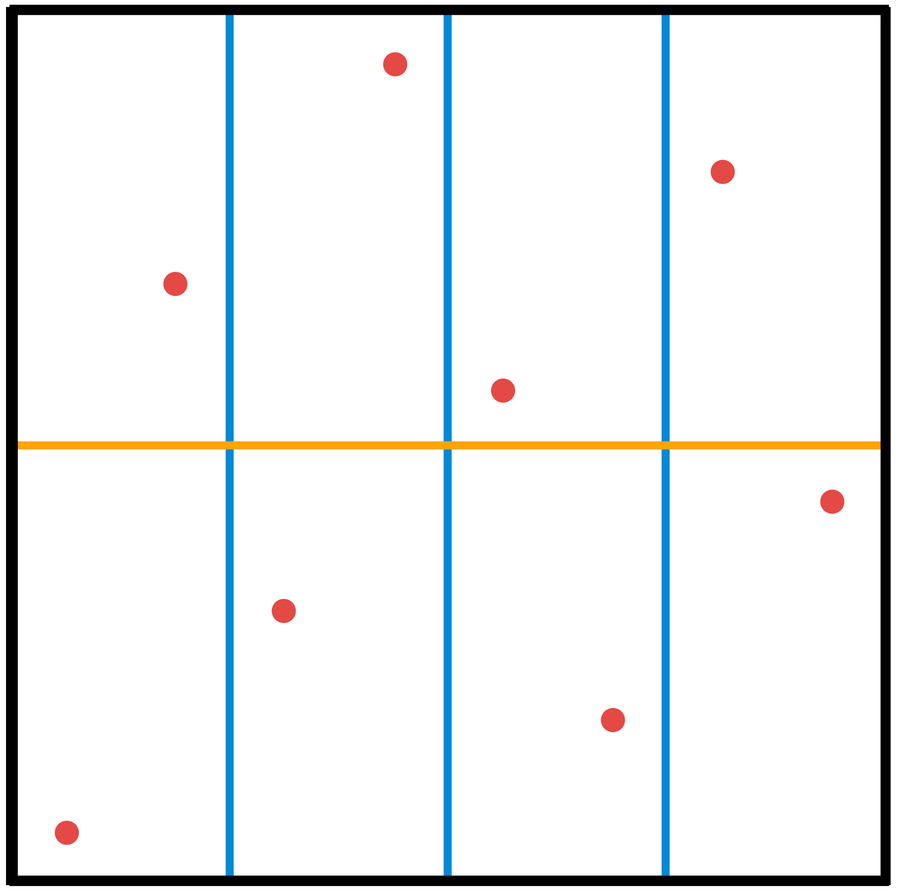


(t,m,s)-nets

t = 0, m = 3, s = 2, p = 2

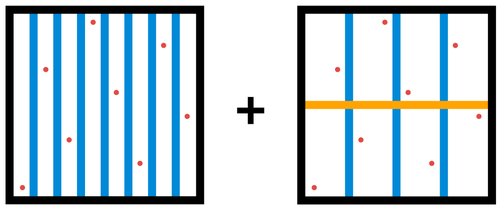


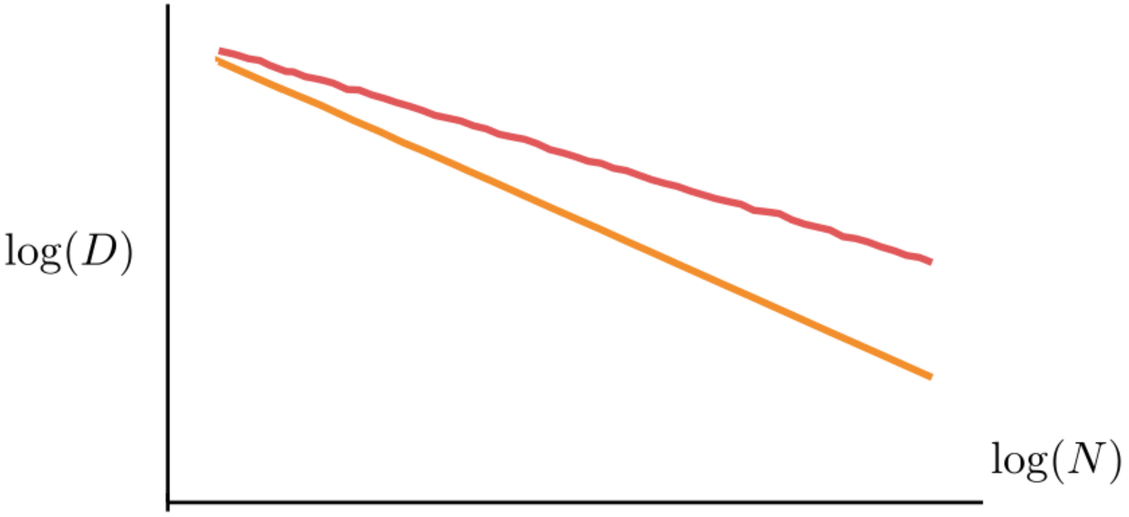


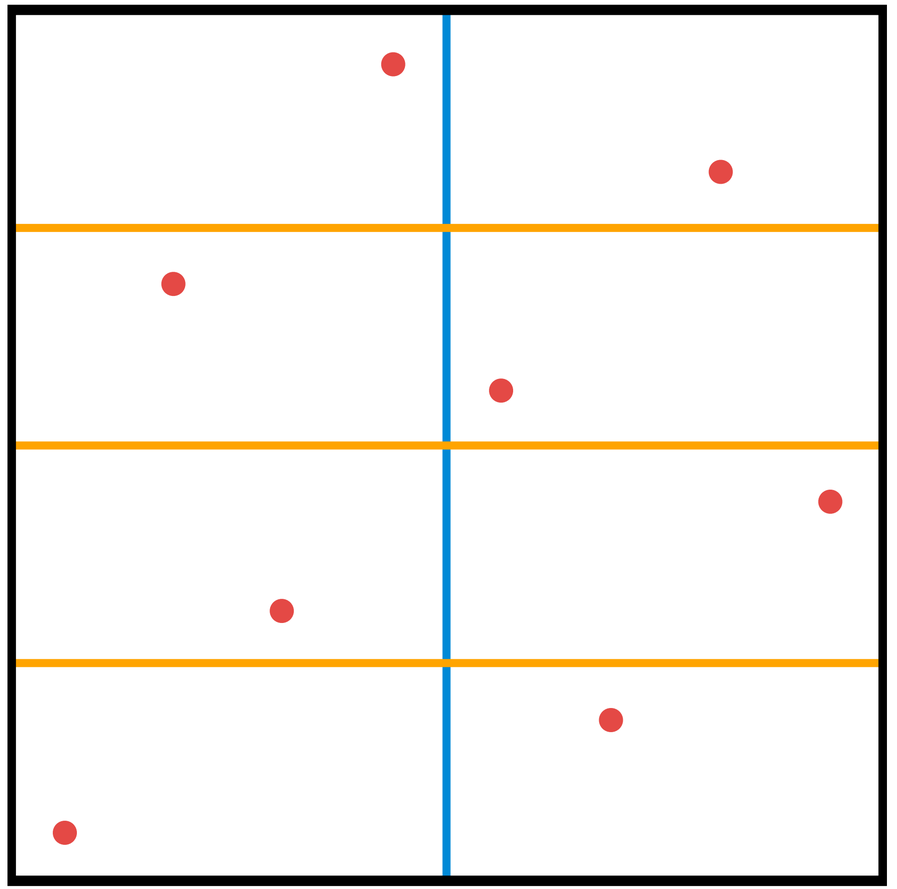


(t,m,s)-nets

t = 0, m = 3, s = 2, p = 2

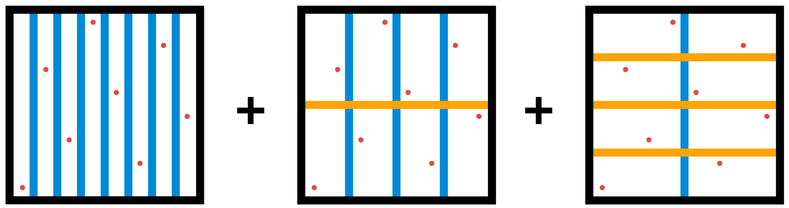


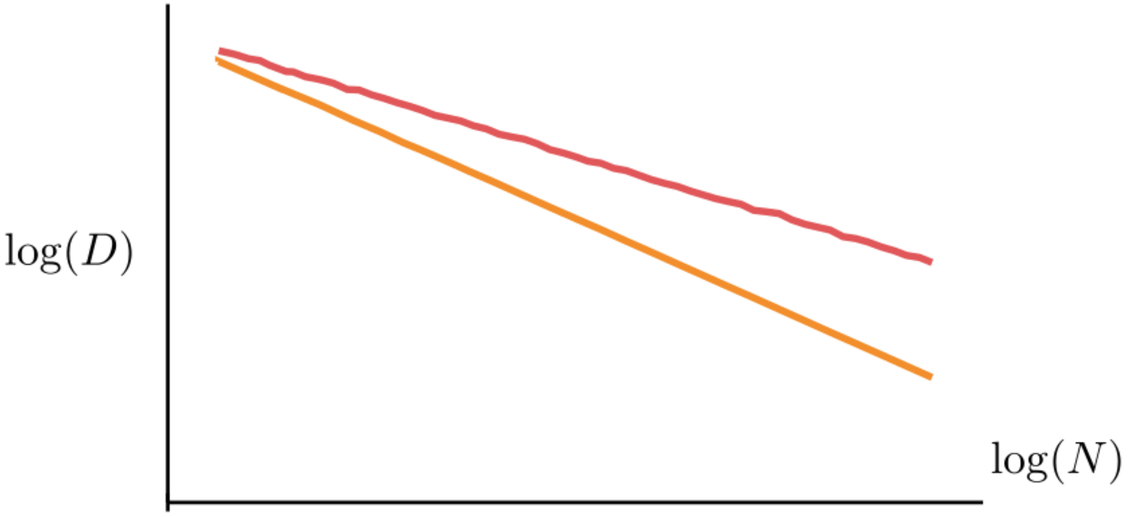


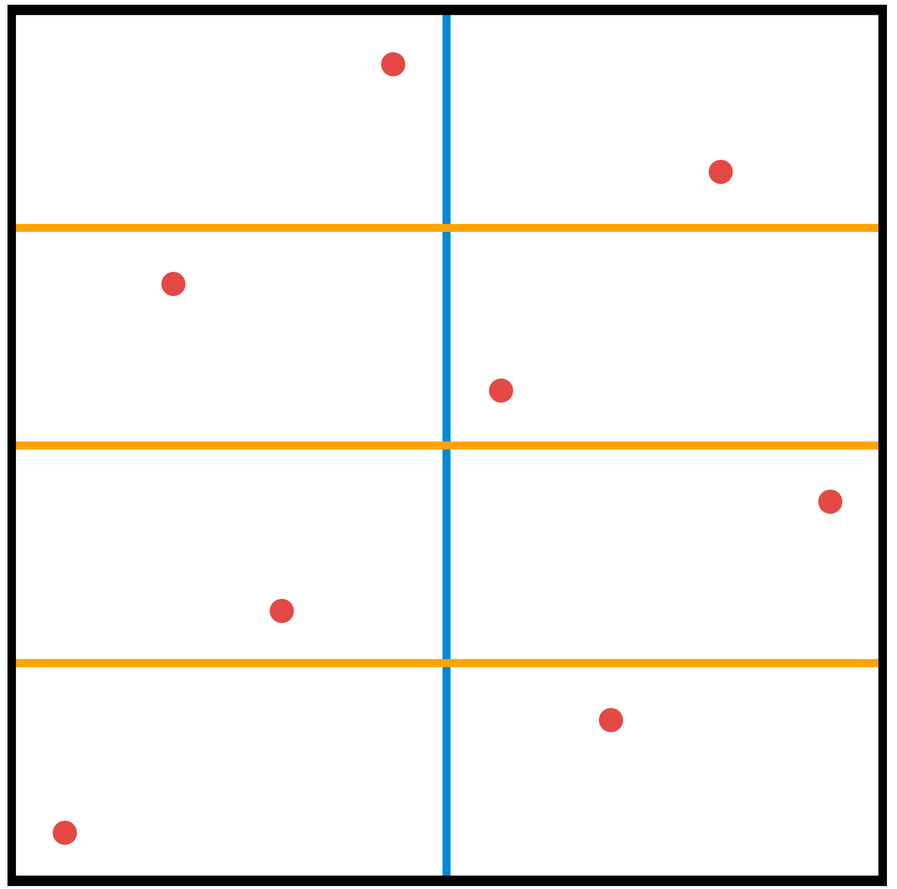


(t,m,s)-nets

t = 0, m = 3, s = 2, p = 2

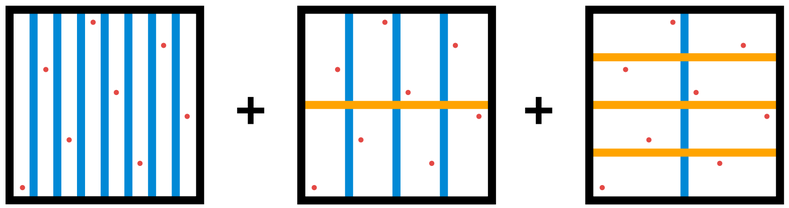


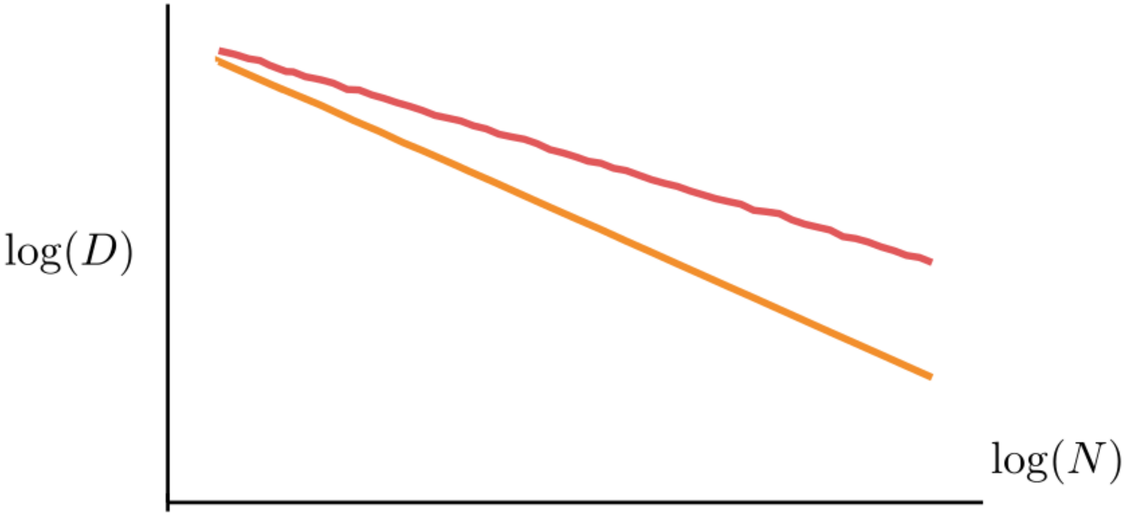


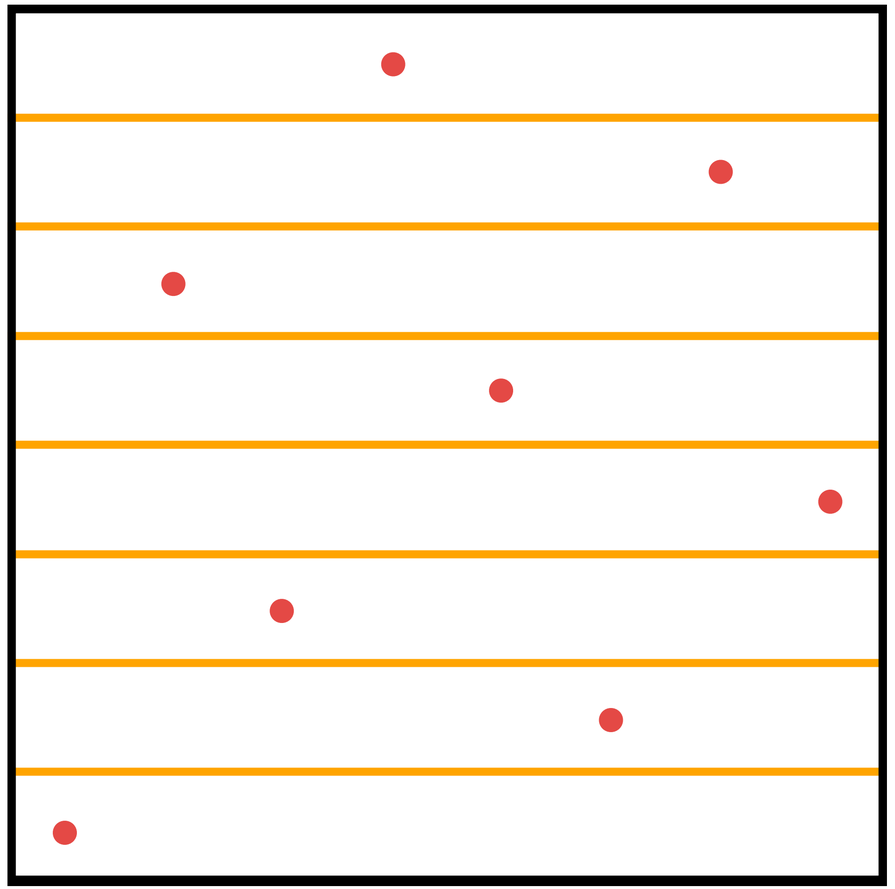


(t,m,s)-nets

t = 0, m = 3, s = 2, p = 2

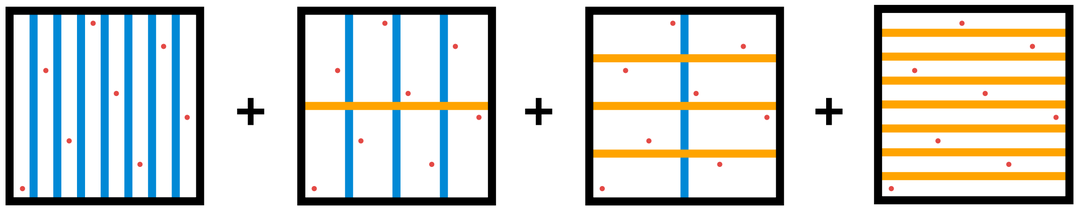


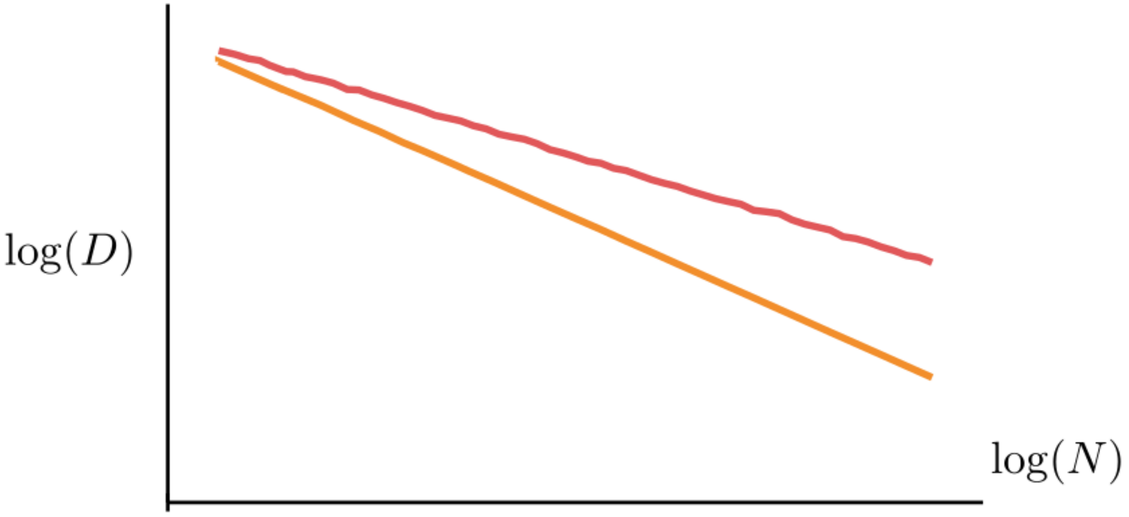


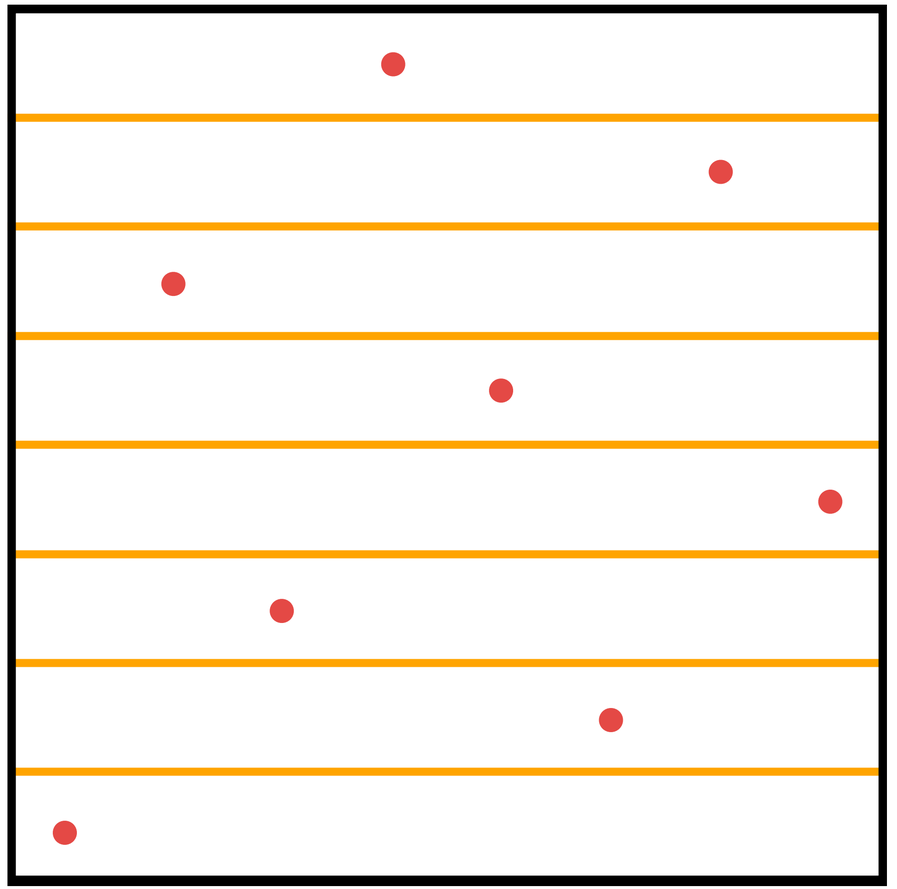


(t,m,s)-nets

t = 0, m = 3, s = 2, p = 2

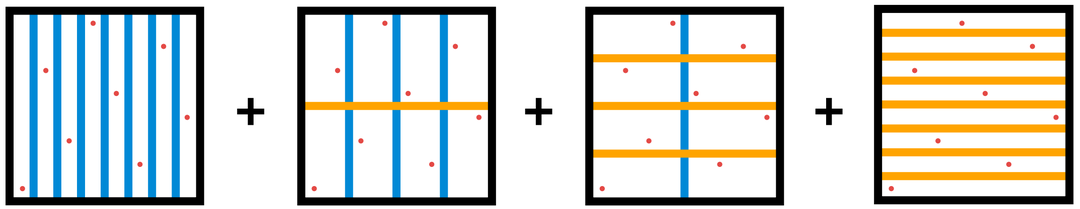


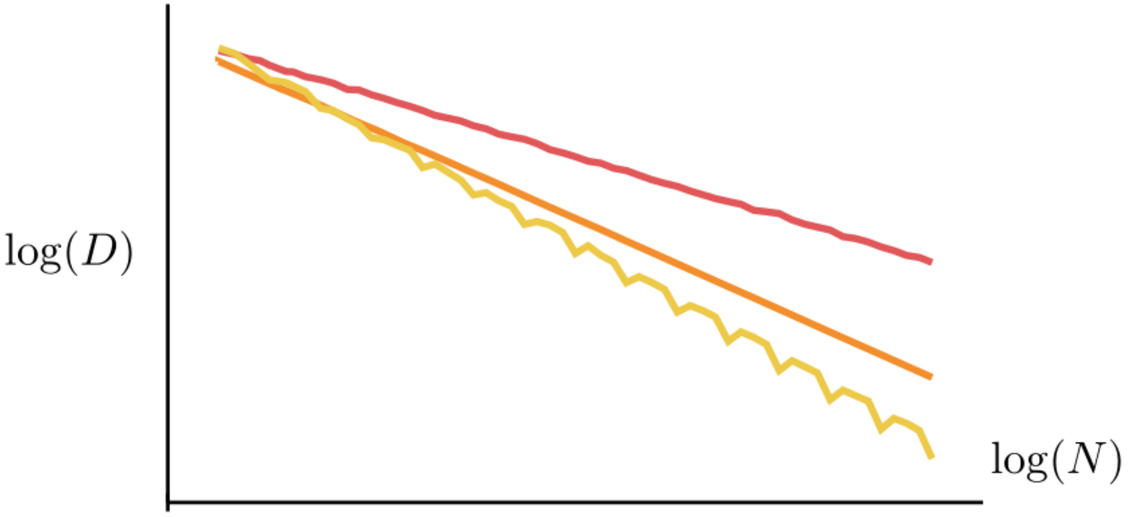


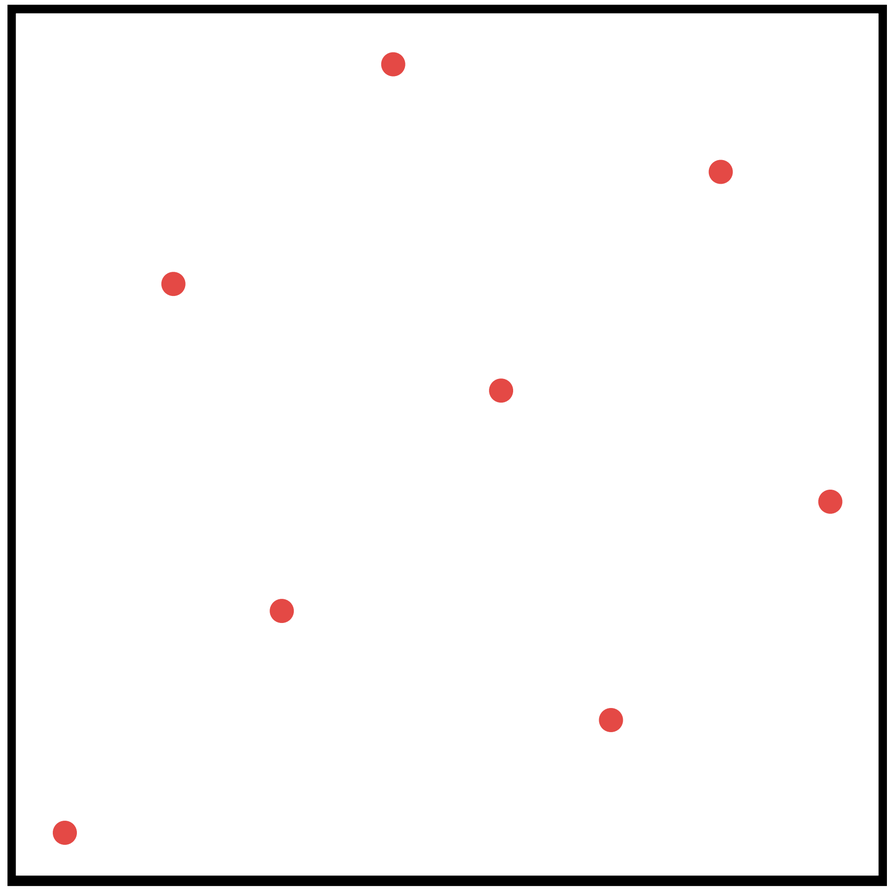


(t,m,s)-nets

t = 0, m = 3, s = 2, p = 2

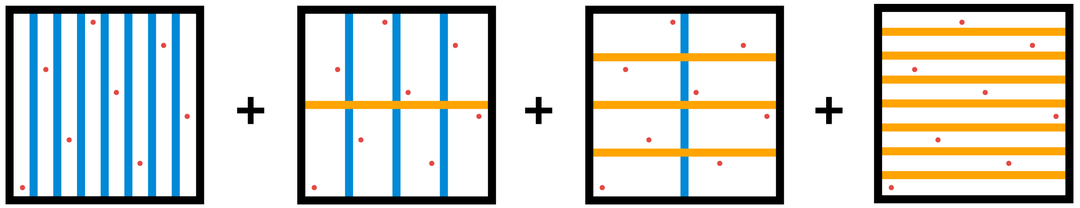


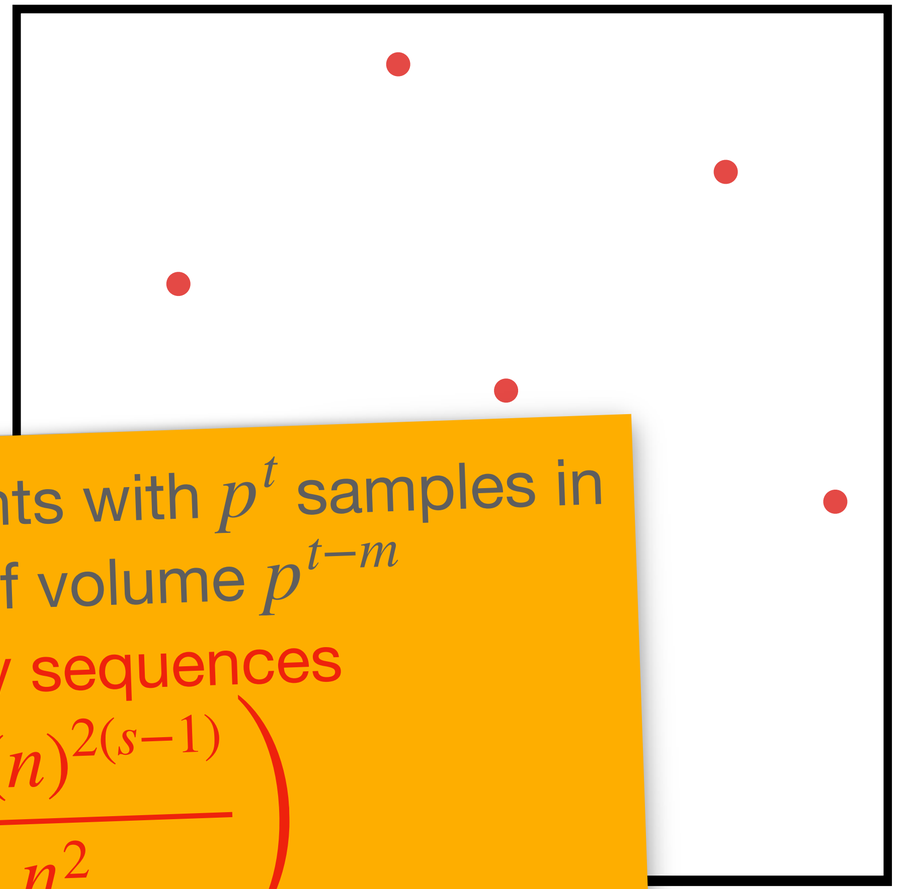


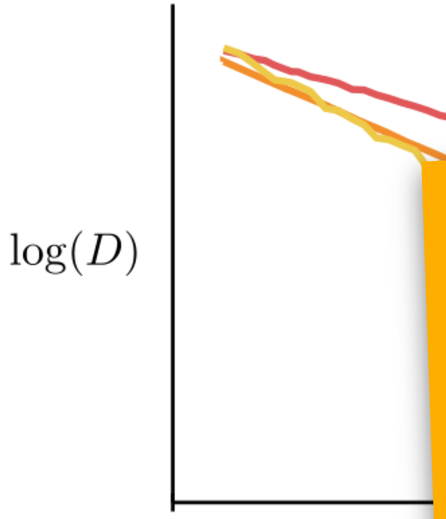


(t,m,s)-nets

t = 0, m = 3, s = 2, p = 2



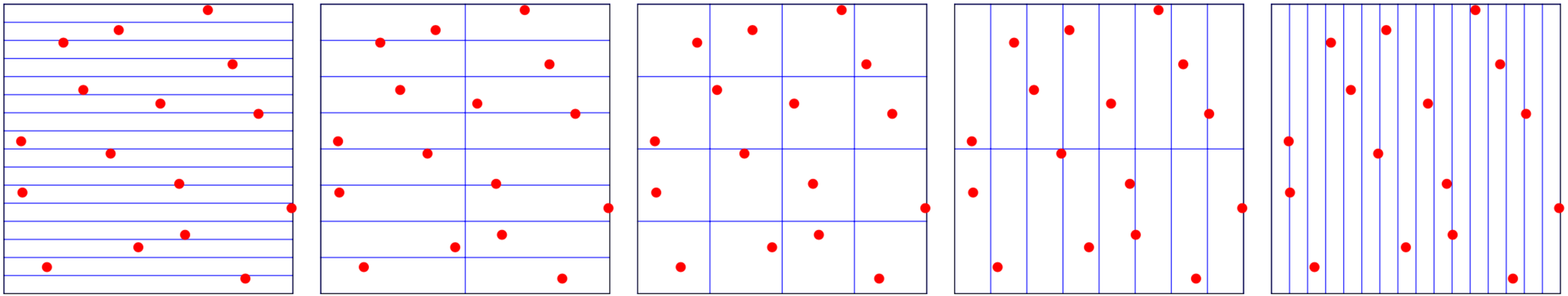




(t,m,s)-net in base *p*: p^m points with p^t samples in elementary intervals of volume p^{t-m}

$\Rightarrow$ Low discrepancy sequences

$$\Rightarrow \Delta_n^2 = O\left(\frac{log(n)^{2(s-1)}}{n^2}\right)$$

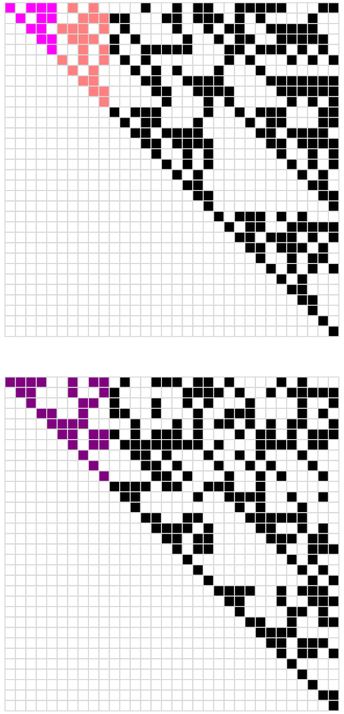


(0,m,2)-nets in base 2: for $n = 2^m$ points, all dyadic partitions of size $1/n$ contain exactly 1 sample

Sobol' construction [Sobol' 67]

Generator matrices $M_i \in \mathbb{F}_b^{m \times m}$

(ingredients: primitive polynomials in $\mathbb{F}_b$, recursive construction…)



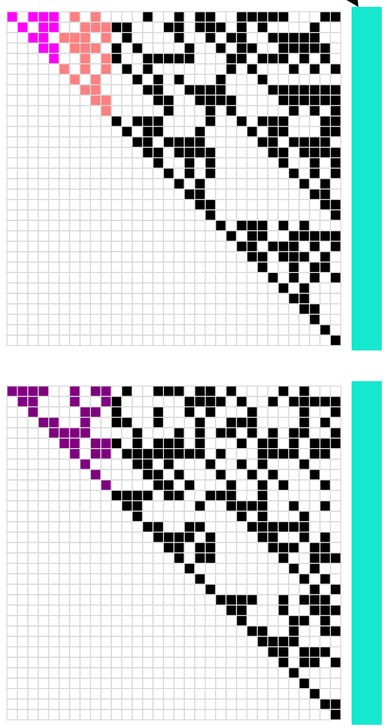
(b=2 here)

Sobol' construction [Sobol' 67]

Index $a \in \mathbb{N} \rightarrow (a_0, \ldots, a_{m-1})_2$

Generator matrices $M_i \in \mathbb{F}_b^{m \times m}$

(ingredients: primitive polynomials in $\mathbb{F}_b$, recursive construction…)

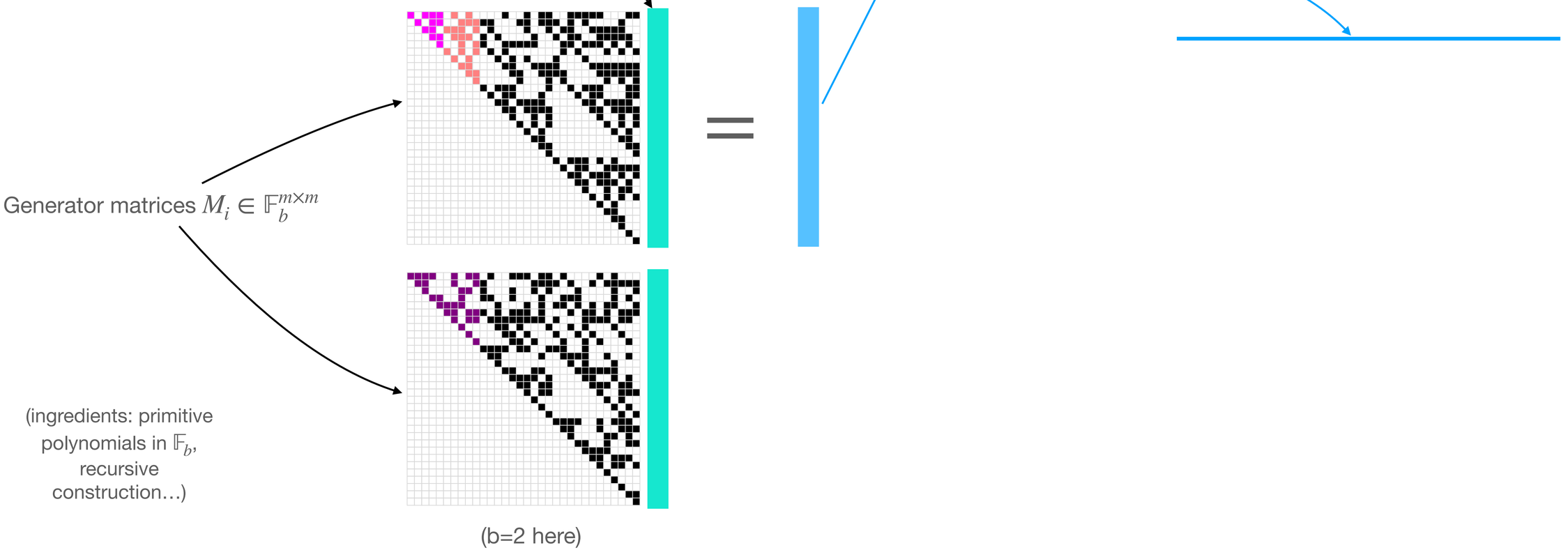


(b=2 here)

Sobol' construction [Sobol' 67]

Index $a \in \mathbb{N} \to (a_0, \ldots, a_{m-1})_2$

$$.(c_0, \ldots, c_{m-1})_2 \to x = \frac{1}{b^m} \sum_{j=0}^{m-1} c_j b^j \in [0,1)$$

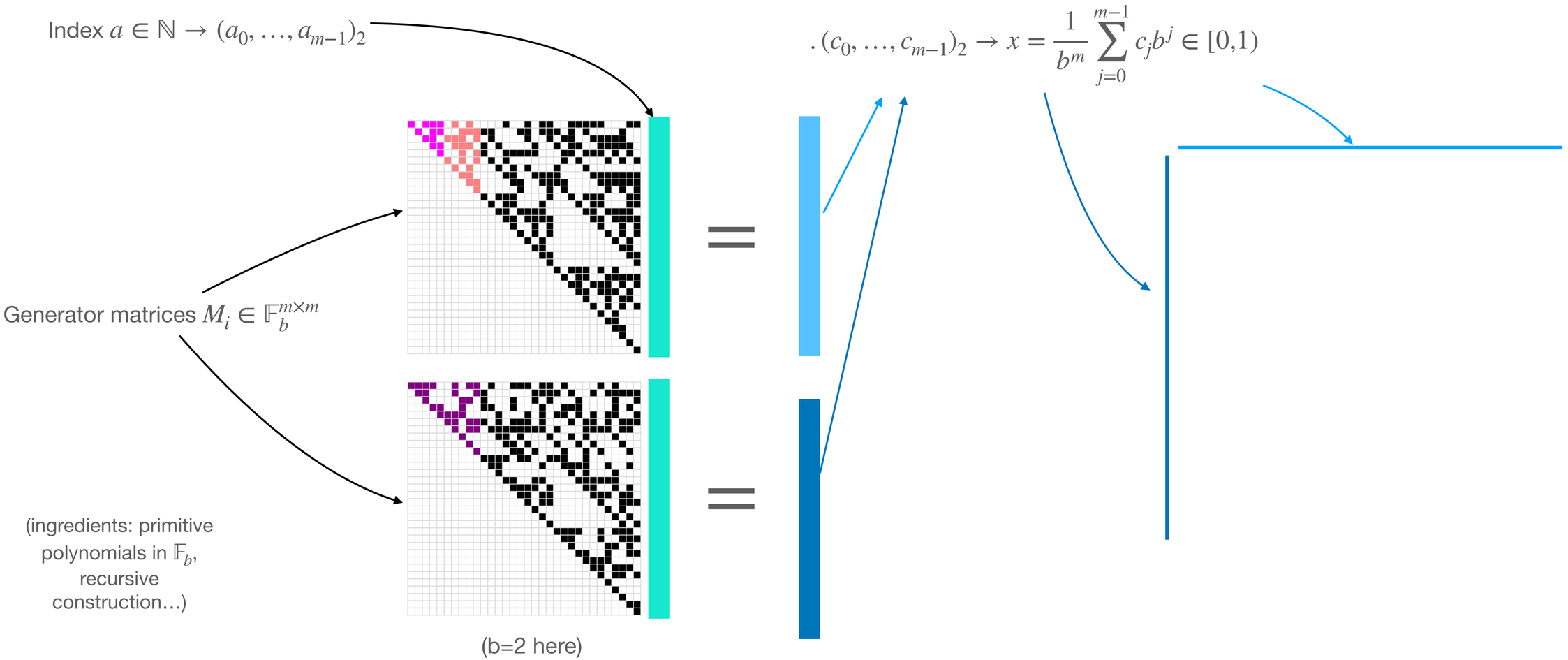


Generator matrices $M_i \in \mathbb{F}_b^{m \times m}$

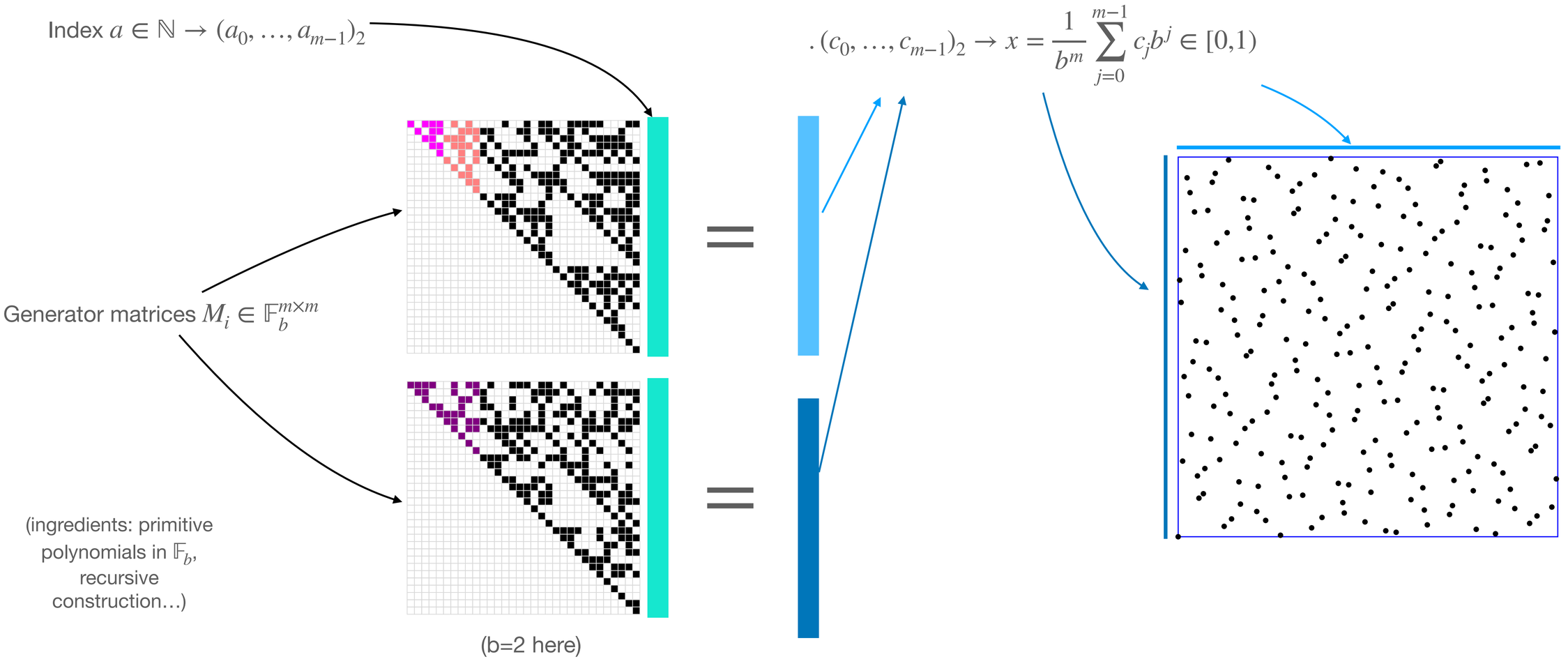
(ingredients: primitive polynomials in $\mathbb{F}_b$, recursive construction…)

(b=2 here)

Sobol' construction [Sobol' 67]



Sobol' construction [Sobol' 67]



Sobol' matrices [Sobol' 67]

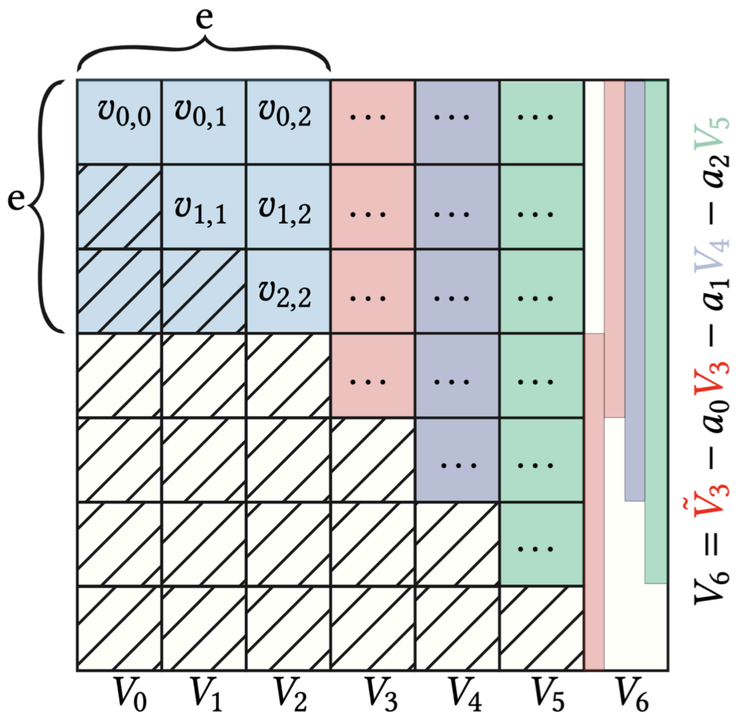
Pick an irreducible polynomial in $\mathbb{F}_p$

$$p(x) = x^3 + a_2 x^2 + a_1 x + a_0$$

Draw a $e \times e$ (non singular upper triangular)

Recursive construction of the columns

$$V_6 = \tilde{V}_3 - a_0 V_3 - a_1 V_4 - a_2 V_5$$



Sobol' matrices [Sobol' 67]

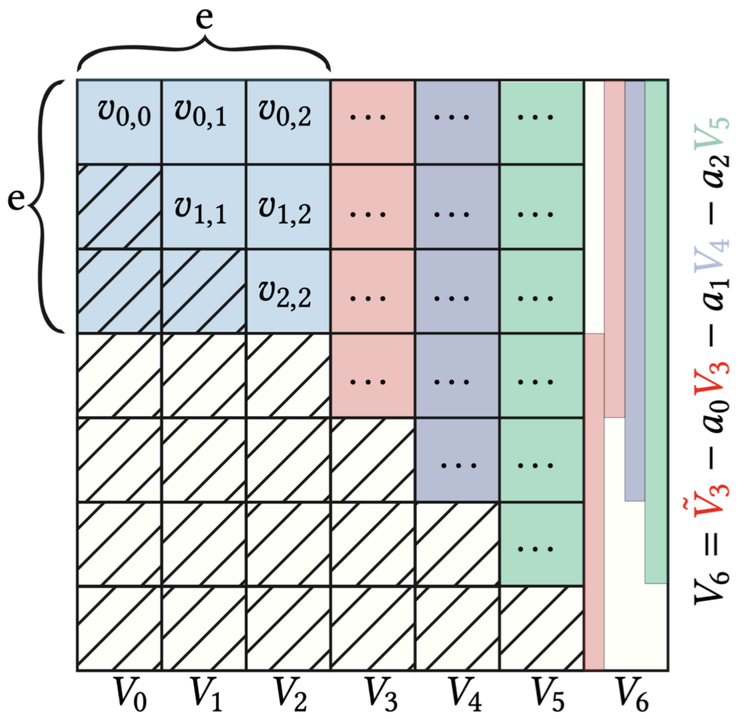
Pick an irreducible polynomial in $\mathbb{F}_p$

$$p(x) = x^3 + a_2 x^2 + a_1 x + a_0$$

Draw a $e \times e$ (non singular upper triangular)

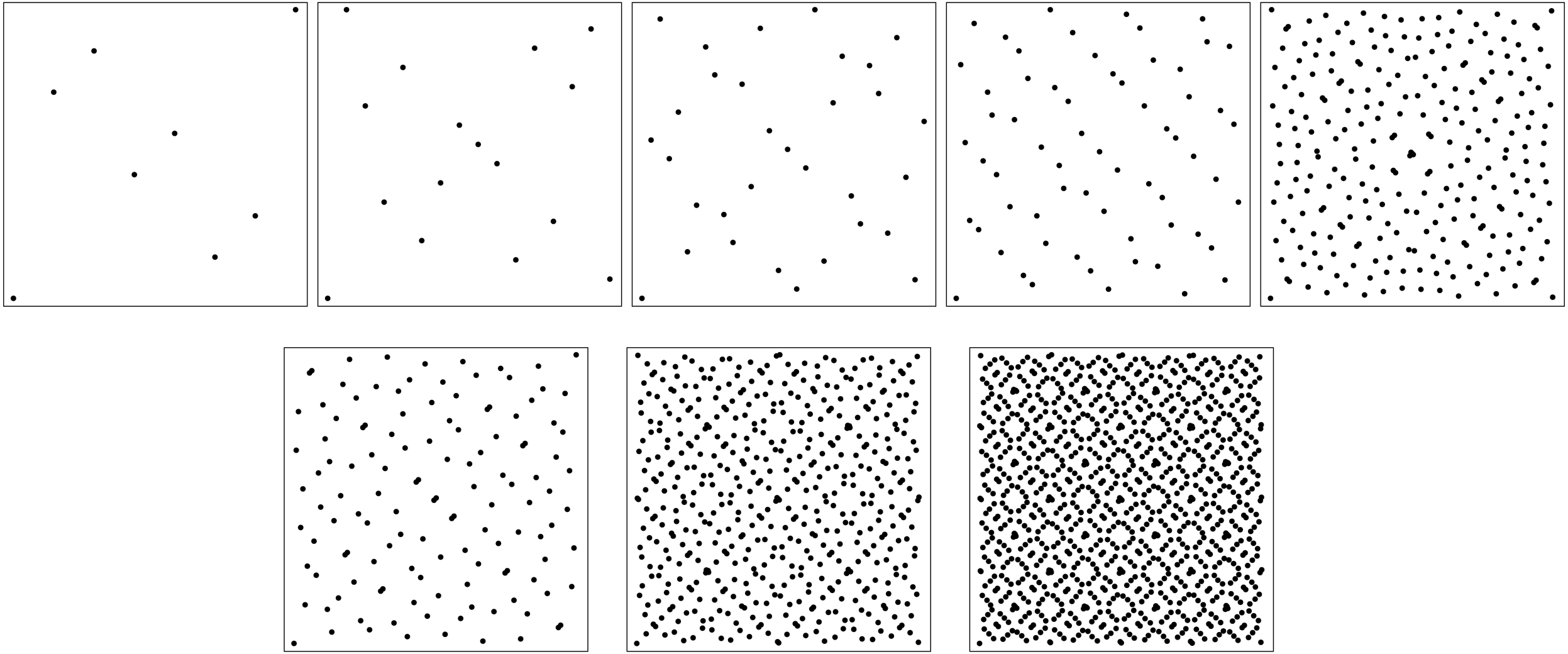
Recursive construction of the columns

$$V_6 = \tilde{V}_3 - a_0 V_3 - a_1 V_4 - a_2 V_5$$

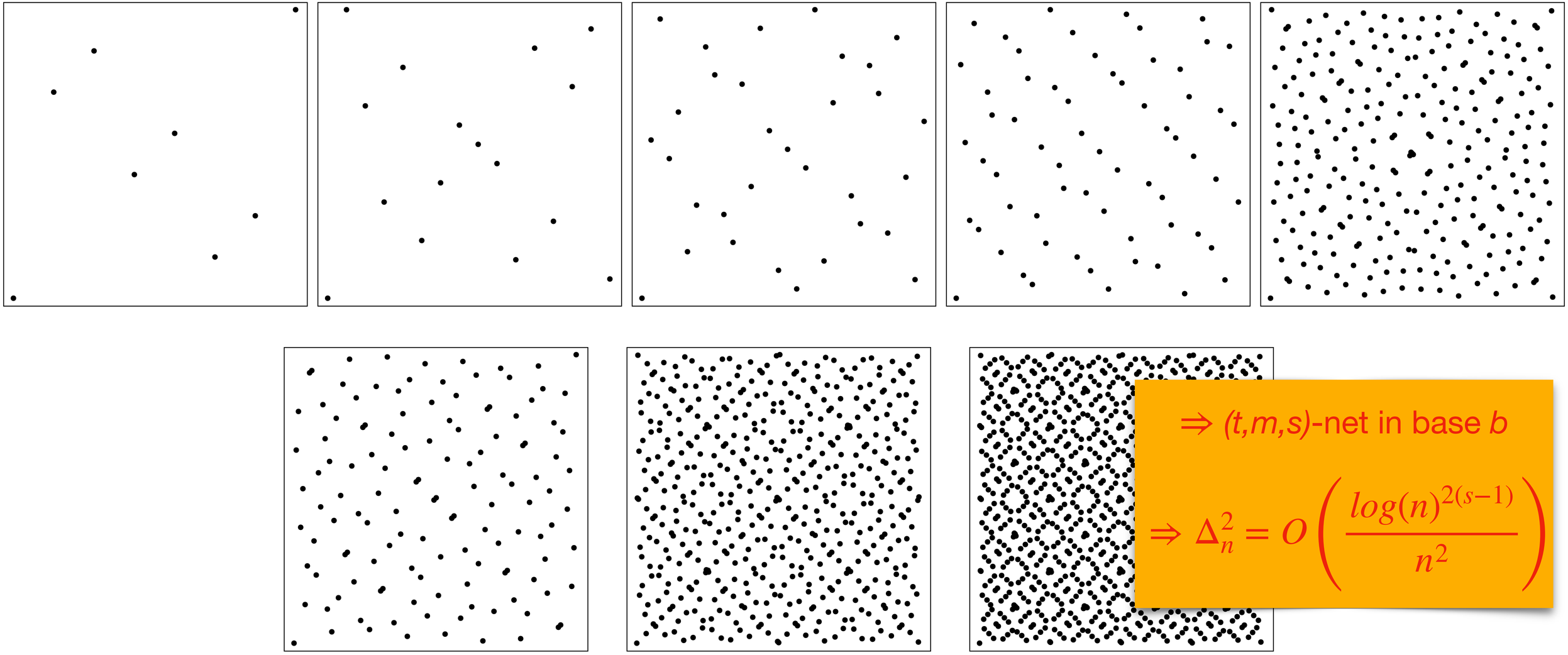


Many extensions: M_i from ILP (Matbuilder), Cascaded Sampling, Quad optimized sampler, primitive → irreducible polynomials ....

Sobol' construction [Sobol' 67]



Sobol' construction [Sobol' 67]



$\Rightarrow$ *(t,m,s)*-net in base *b*

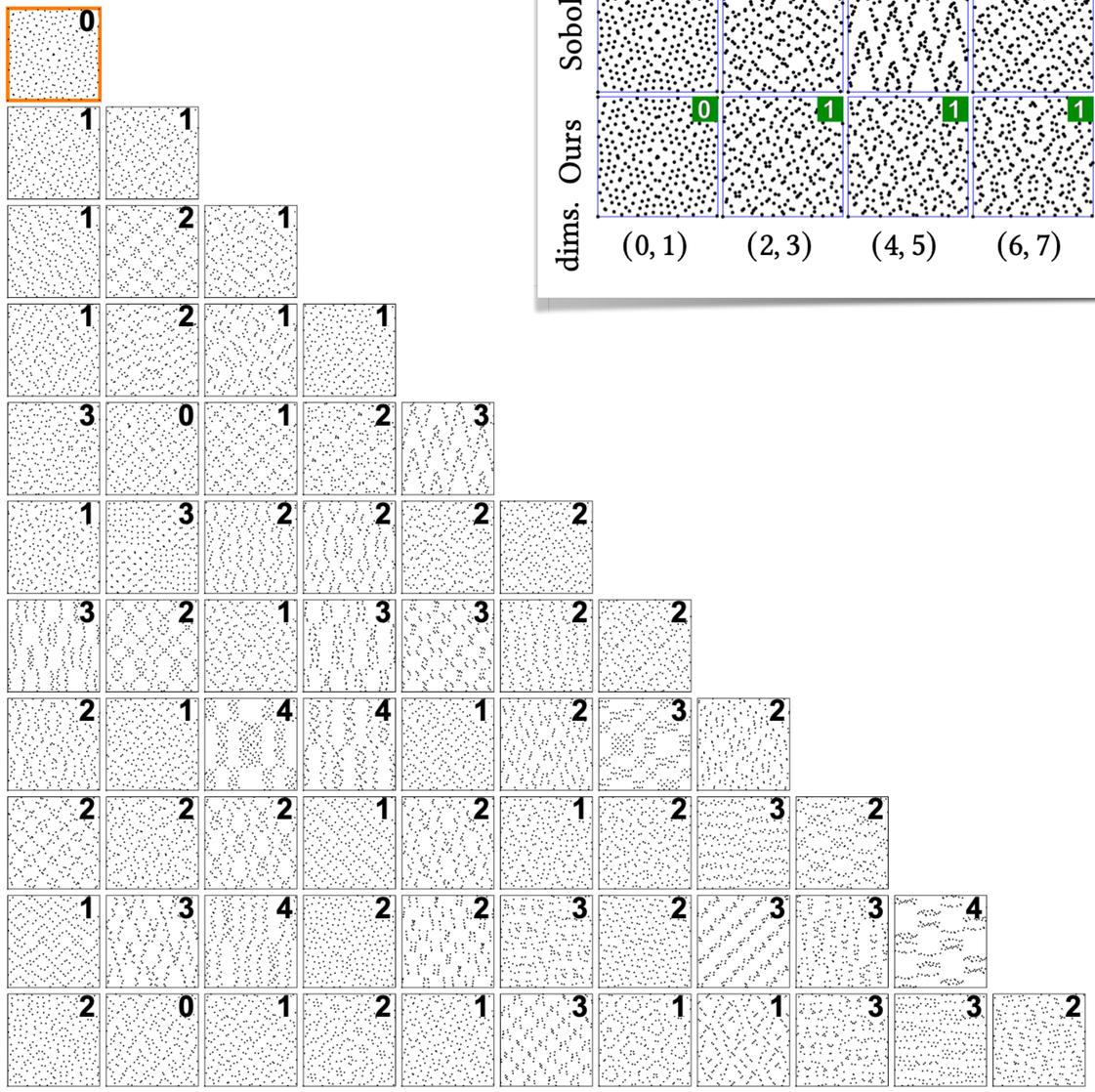
$$\Rightarrow \Delta_n^2 = O\left(\frac{log(n)^{2(s-1)}}{n^2}\right)$$

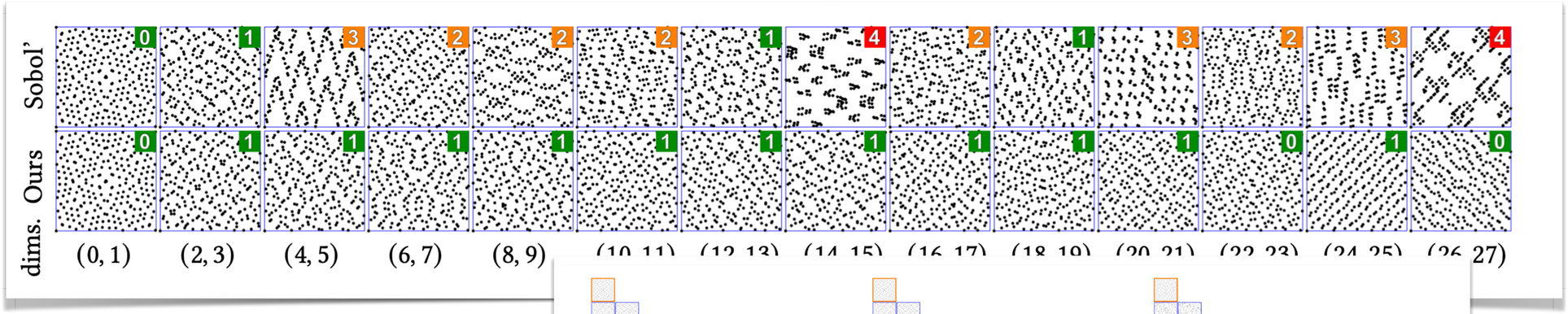
Conclusion

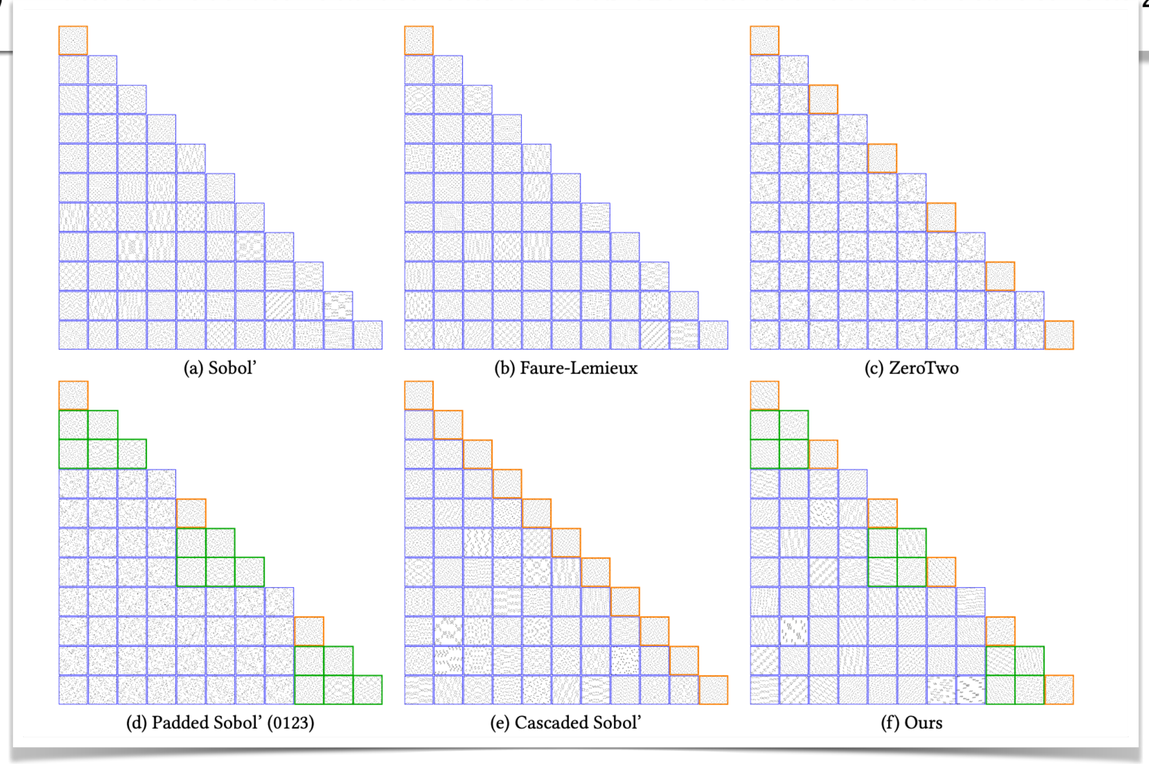
- **Algebraic construction**
  - one small (boolean) matrix per dimension
  - Highly optimized matrix/vector multiplication in GF(2)
  - Strong uniformity guarantees

Projective LDS

Problem


0
1 1
1 2 1
1 2 1 1
3 0 1 2 3
1 3 2 2 2 2
3 2 1 3 3 2 2
2 1 4 4 1 2 3 2
2 2 2 1 2 1 2 3 2
1 3 4 2 2 3 2 3 3 4
2 0 1 2 1 3 1 1 3 3 2


Sobol'
Ours
dims.
0 1 3 2 2 2 1 4 2 1 3 2 3 4
0 1 1 1 1 1 1 1 1 1 1 0 1 0
(0, 1)
(2, 3)
(4, 5)
(6, 7)
(8, 9)
(10, 11)
(12, 13)
(14, 15)
(16, 17)
(18, 19)
(20, 21)
(22, 23)
(24, 25)
(26, 27)


(a) Sobol'
(b) Faure-Lemieux
(c) ZeroTwo
(d) Padded Sobol' (0123)
(e) Cascaded Sobol'
(f) Ours

Experiment design

Good Samples

Bad Samples

Experiment design

Good Samples

Bad Samples

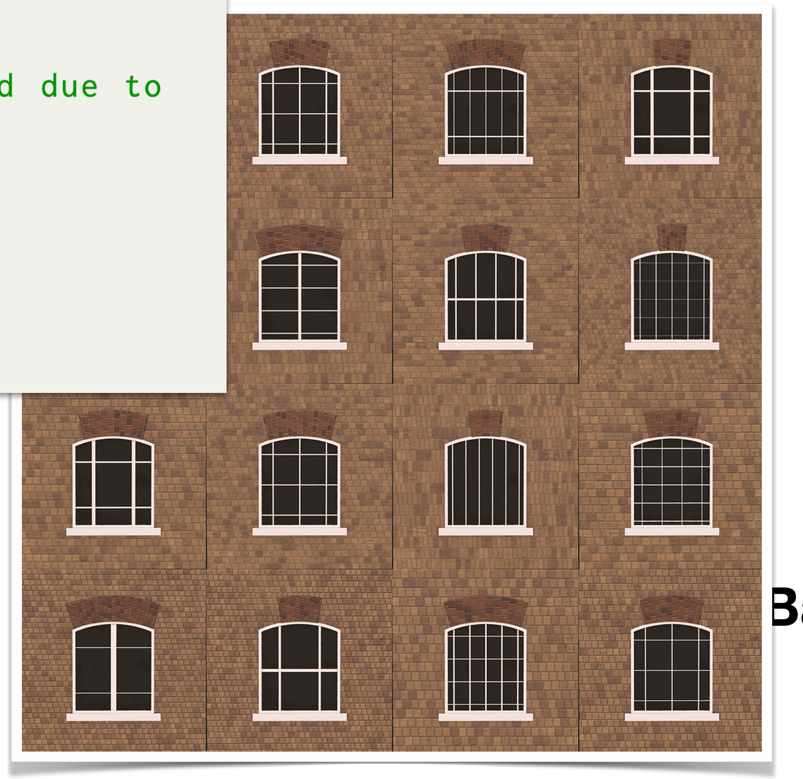
Experiment design

Good Samples

Bad Samples

Experiment design

```
s=4
p=3
m=17
# particular stratifications were skipped due to
# lack of solutions
from 3 to 5 stratified 0 1 2 3
from 7 to 9 stratified 0 1 2 3
from 11 to 13 stratified 0 1 2 3
from 15 stratified 0 1 2 3
weak 1 net 0 1 2 3
```



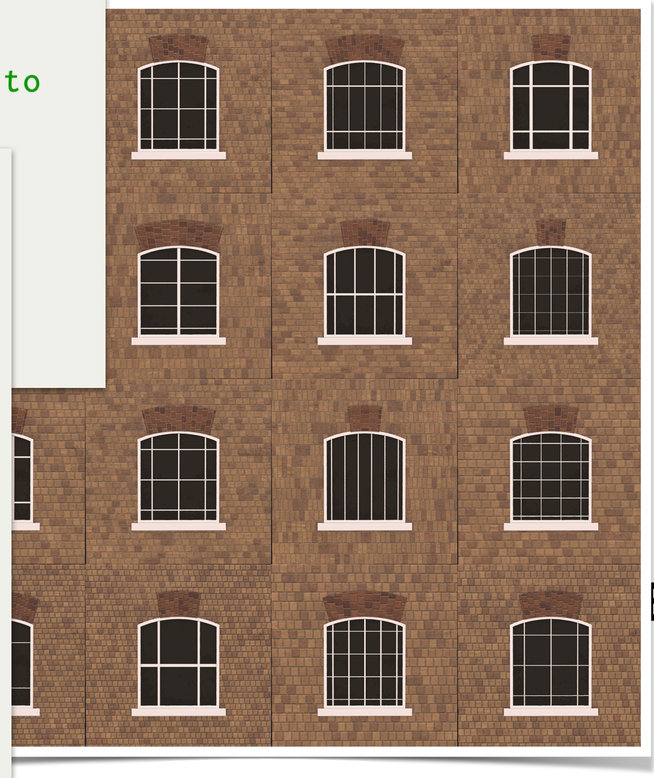
Good Samples

Bad Samples

Experiment design

```
s=4
p=3
m=17
# particular stratifications were skipped due to
# lack of solutions
from 3 to 5 stratified 0 1
from 7 to 9 stratified 0 1
from 11 to 13 stratified 0
from 15 stratified 0 1 2 3
weak 1 net 0 1 2 3
```

```
#Generic-proj-LDS
s=6
p=3
m=10
net 0 1
net 1 2
net 2 3
net 3 4
net 4 5
weak 1 net 0 2
weak 1 net 0 3
weak 1 net 0 4
weak 1 net 0 5
weak 1 net 1 3
weak 1 net 1 4
weak 1 net 1 5
weak 1 net 2 4
weak 1 net 2 5
weak 1 net 3 5
```



Good Sam

Bad Samples

Experiment design

```
s=4
p=3
m=17
# particular stratifications were
# lack of solutions
from 3 to 5 stratified 0 1
from 7 to 9 stratified 0 1
from 11 to 13 stratified 0
from 15 stratified 0 1 2 3
weak 1 net 0 1 2 3
```

```
#Gener
s=6
p=3
m=10
net 0
net 1
net 2
net 3
net 4 5
weak 1 net 0 2
weak 1 net 0 3
weak 1 net 0 4
weak 1 net 0 5
weak 1 net 1 3
weak 1 net 1 4
weak 1 net 1 5
weak 1 net 2 4
weak 1 net 2 5
weak 1 net 3 5
```

```
#Generic-OA
s=9
p=3
m=10
from 3 stratified 0 1 2
from 3 stratified 1 2 3
from 3 stratified 2 3 4
from 3 stratified 3 4 5
from 3 stratified 4 5 6
from 3 stratified 5 6 7
from 3 stratified 6 7 8
from 5 weak 1 stratified 0 1 2 3 4 5 6 7 8
```

Good Sam

Bad Samples

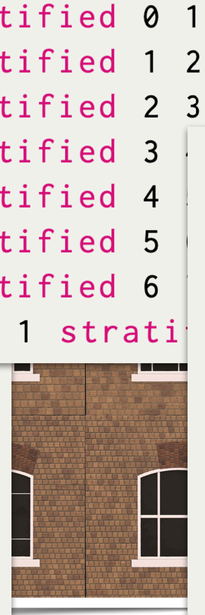
Experiment design

```
s=4
p=3
m=17
# particular stratifications were
# lack of solutions
from 3 to 5 stratified 0 1
from 7 to 9 stratified 0 1
from 11 to 13 stratified 0
from 15 stratified 0 1 2 3
weak 1 net 0 1 2 3
```

```
#Gener
s=6
p=3
m=10
net 0
net 1
net 2
net 3
net 4 5
weak 1 net 0 2
weak 1 net 0 3
weak 1 net 0 4
weak 1 net 0 5
weak 1 net 1 3
weak 1 net 1 4
weak 1 net 1 5
weak 1 net 2 4
weak 1 net 2 5
weak 1 net 3 5
```

```
#Generic-OA
s=9
p=3
m=10
from 3 stratified 0 1 2
from 3 stratified 1 2 3
from 3 stratified 2 3 4
from 3 stratified 3
from 3 stratified 4
from 3 stratified 5
from 3 stratified 6
from 5 weak 1 strati
```

```
#Mixed
s=10
p=3
m=10
net 0 1
net 1 2
net 2 3
net 3 4
net 4 5
from 3 stratified 0 1 2
from 4 stratified 1 2 3
from 3 stratified 2 3 4
from 4 stratified 3 4 5
from 4 to 6 stratified 0 1 2 3 4
from 4 to 6 stratified 1 2 3 4 5
```



Good Sam **es**

Experiment design

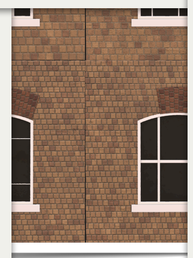
```
s=4
p=3
m=17
# particular stratifications were
# lack of solutions
from 3 to 5 stratified 0 1
from 7 to 9 stratified 0 1
from 11 to 13 stratified 0
from 15 stratified 0 1 2 3
weak 1 net 0 1 2 3
```

```
#Gener
s=6
p=3
m=10
net 0
net 1
net 2
net 3
net 4 5
weak 1 net 0 2
weak 1 net 0 3
weak 1 net 0 4
weak 1 net 0 5
weak 1 net 1 3
weak 1 net 1 4
weak 1 net 1 5
weak 1 net 2 4
weak 1 net 2 5
weak 1 net 3 5
```

```
#Generic-OA
s=9
p=3
m=10
from 3 stratified 0 1 2
from 3 stratified 1 2 3
from 3 stratified 2 3 4
from 3 stratified 3
from 3 stratified 4
from 3 stratified 5
from 3 stratified 6
from 5 weak 1 strati
```

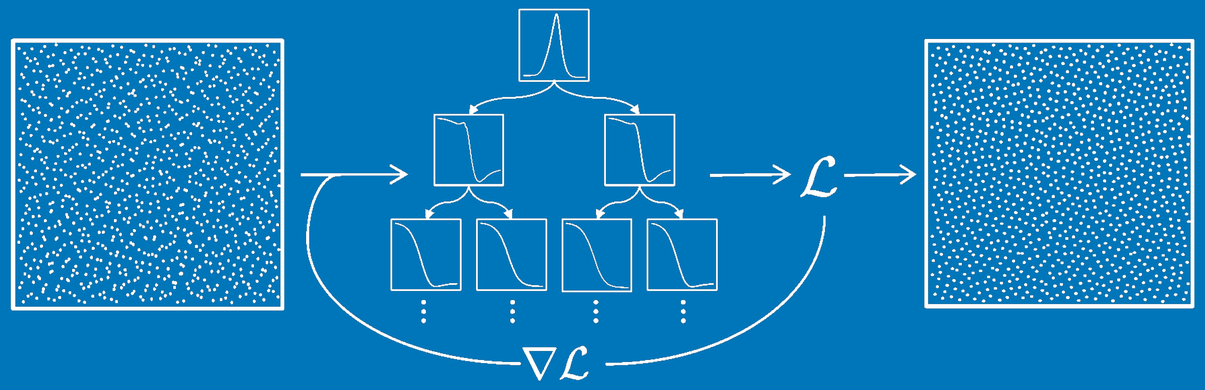
```
s=7
p=2
m=5
# (Brick-Amount-X, Brick-Amount-Y)
net 0 1
# (Window-Brace-Amount-X, Window-Brace-Width-X,
#  Window-Brace-Amount-Y, Window-Brace-Width-Y)
weak 1 net 3 4 5 6
# Overall uniformity, including Brick-Lintel-Wi
weak 2 net 0 1 2 3 4 5 6
```

```
#Mixed
s=10
p=3
m=10
net 0 1
net 1 2
net 2 3
net 3 4
net 4 5
from 3 stratified 0 1 2
from 4 stratified 1 2 3
from 3 stratified 2 3 4
from 4 stratified 3 4 5
from 4 to 6 stratified 0 1 2 3 4
from 4 to 6 stratified 1 2 3 4 5
```

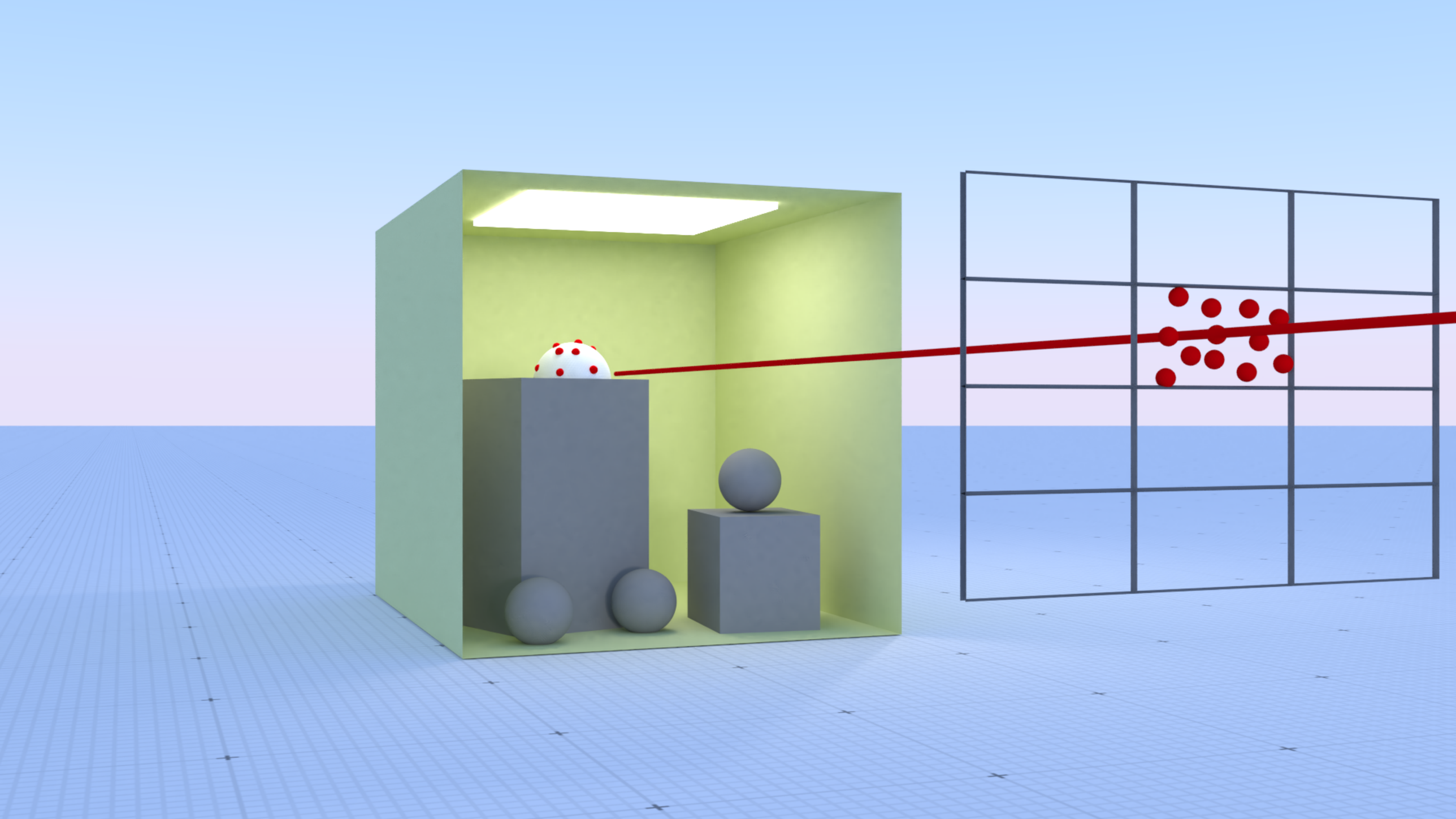


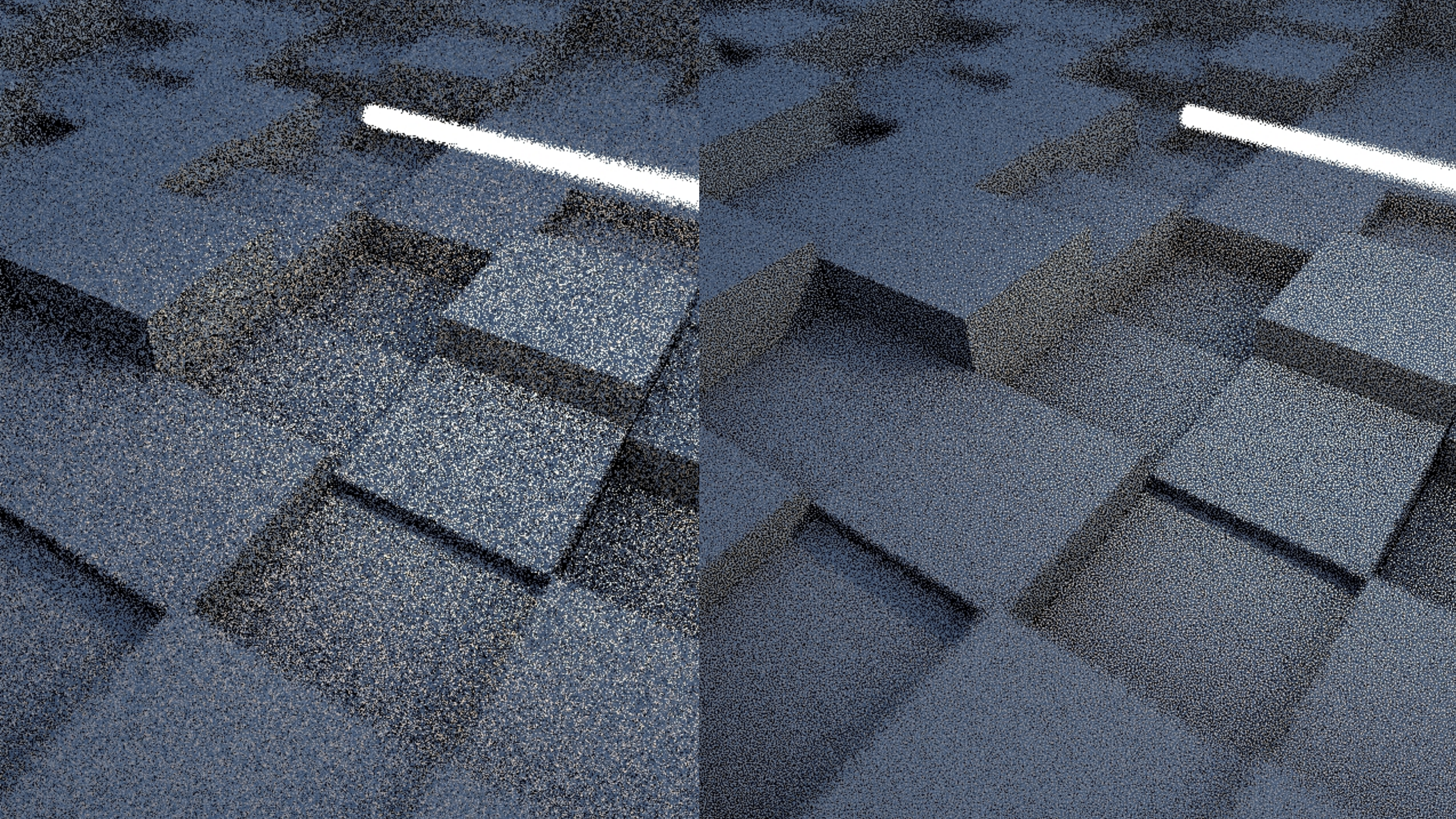
Good Sam **es**

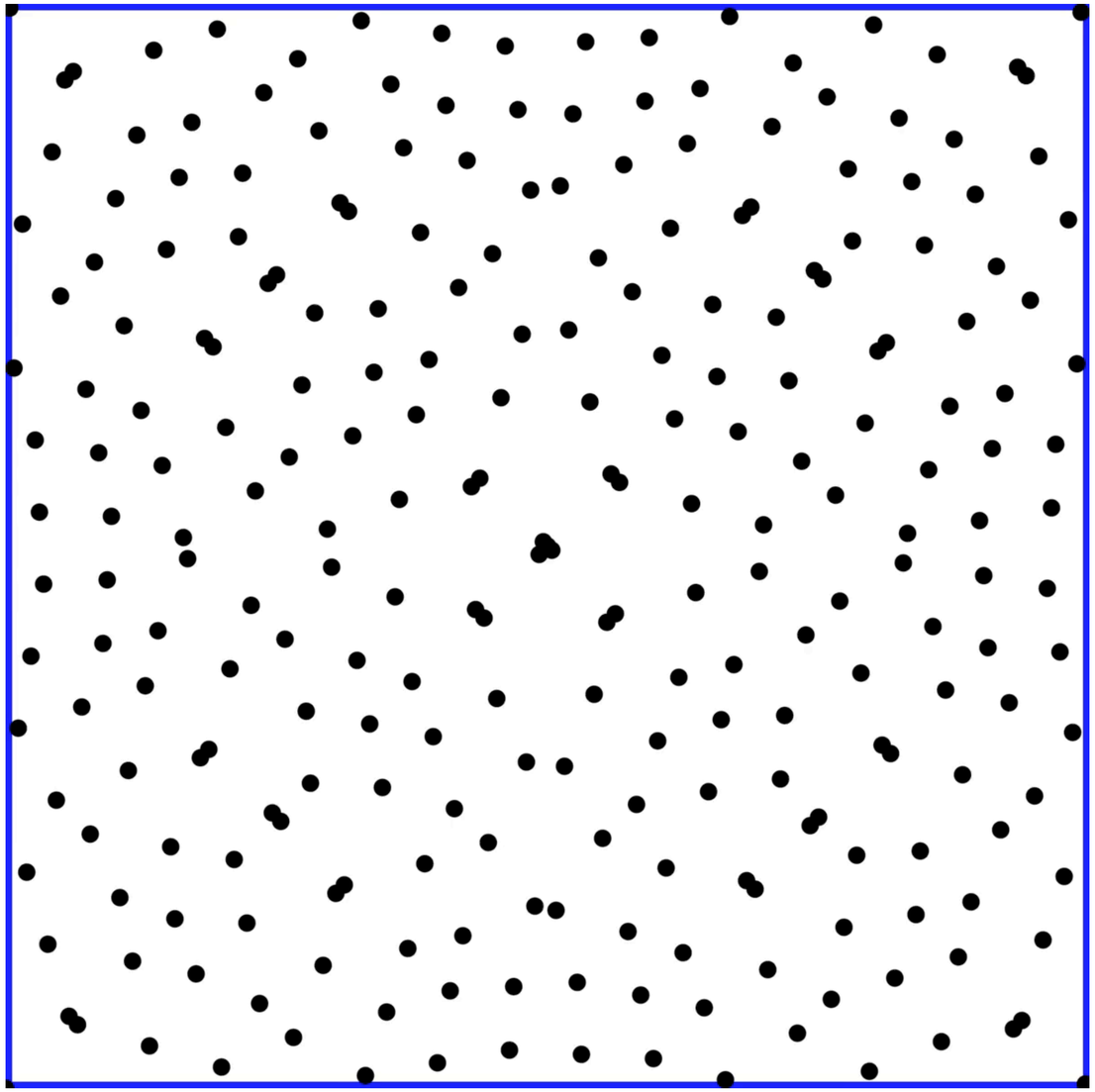
Low discrepancy scrambling



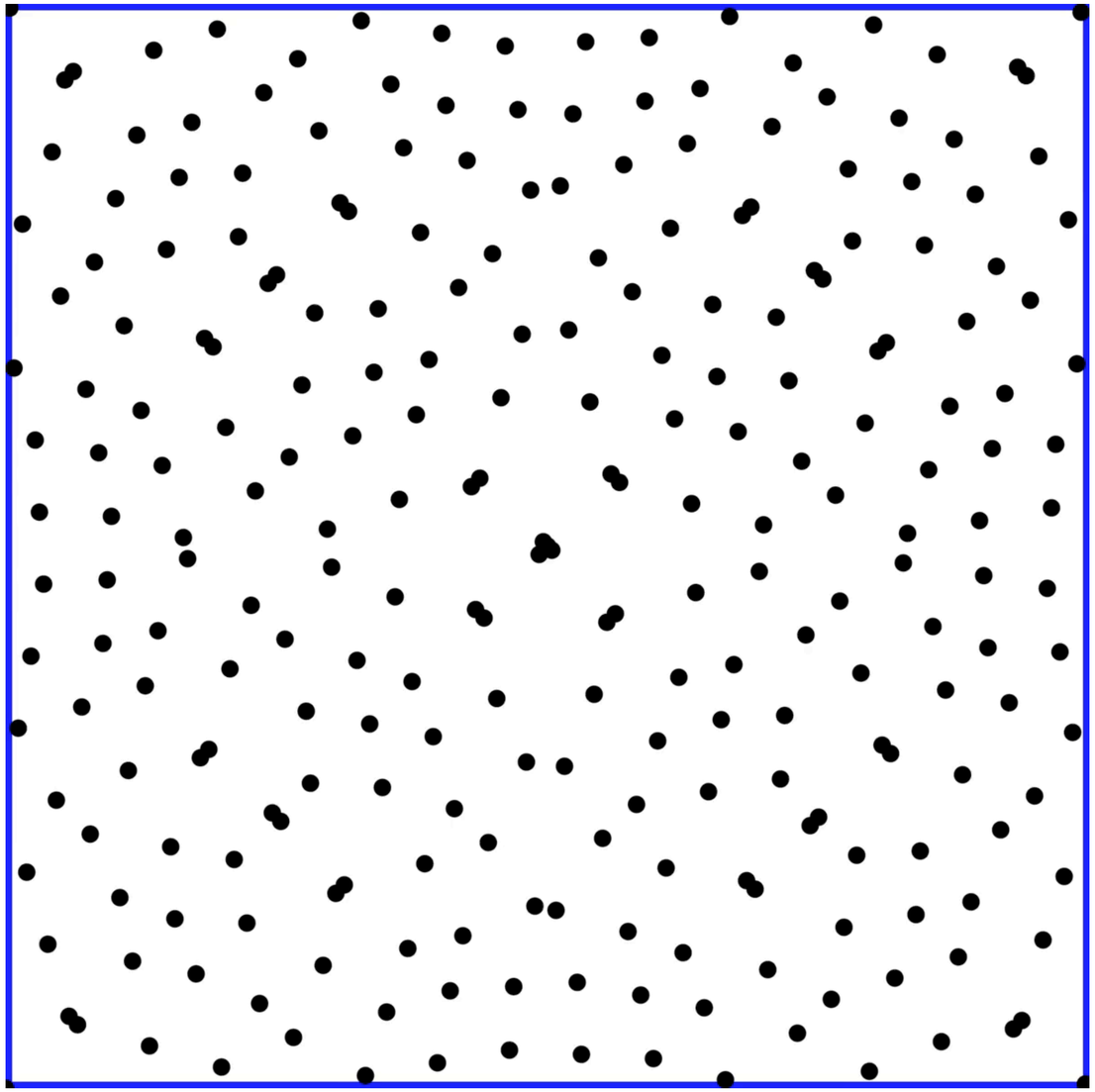
Bastien Doignies, D.C., Nicolas Bonneel, Julie Digne, Jean-Claude Iehl, Victor Ostromoukhov
ACM Transactions on Graphics December 2024





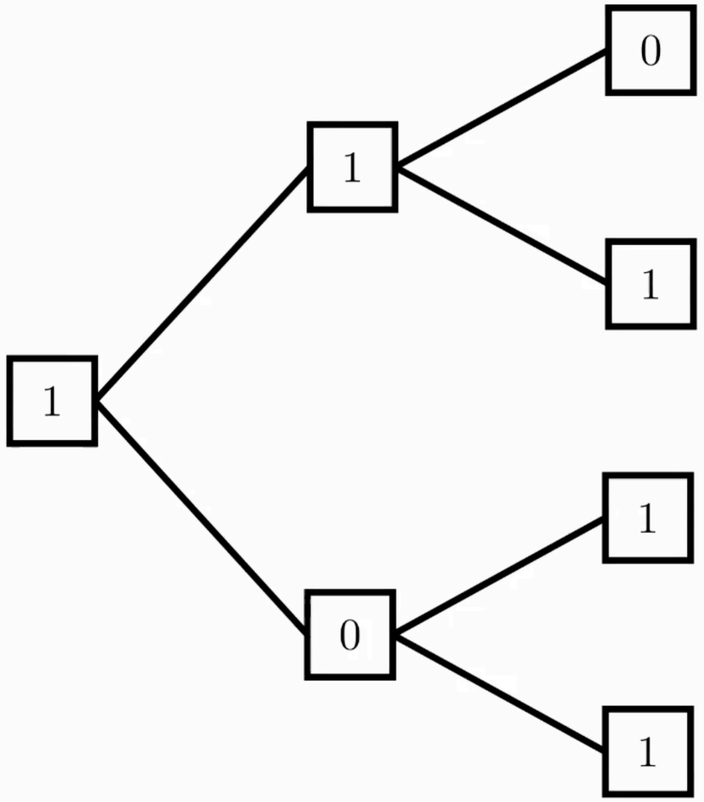


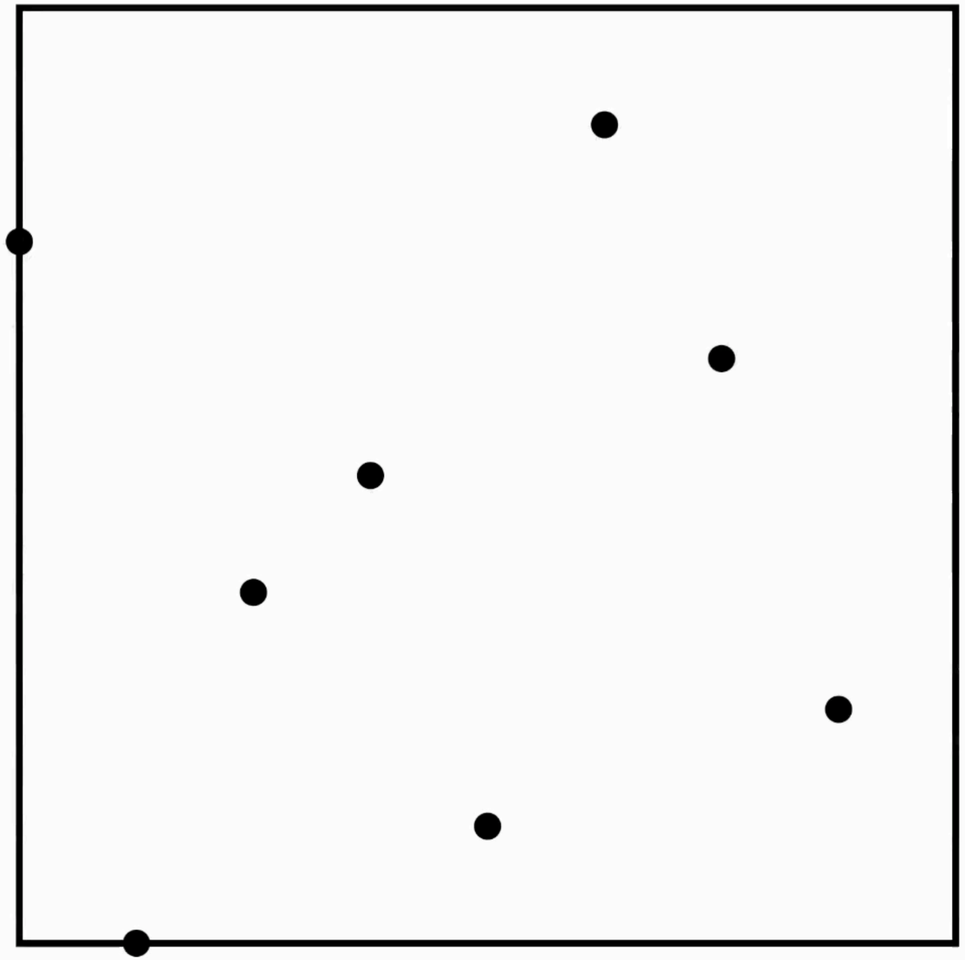
Per dimension scrambling preserving the (t,m,s)-net properties [Owen, 97]



Per dimension scrambling preserving the (t,m,s)-net properties [Owen, 97]

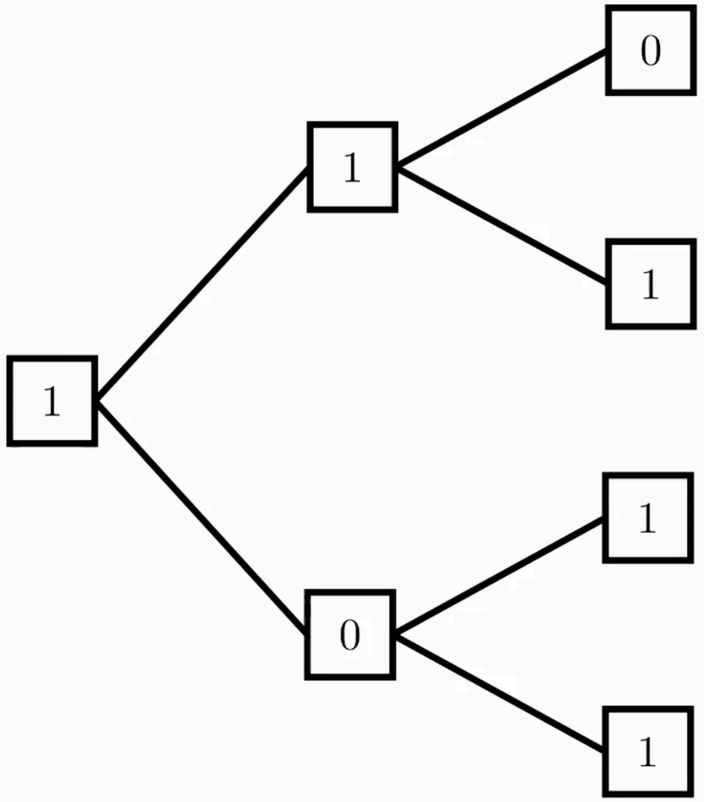
Differentiable Owen's scrambling

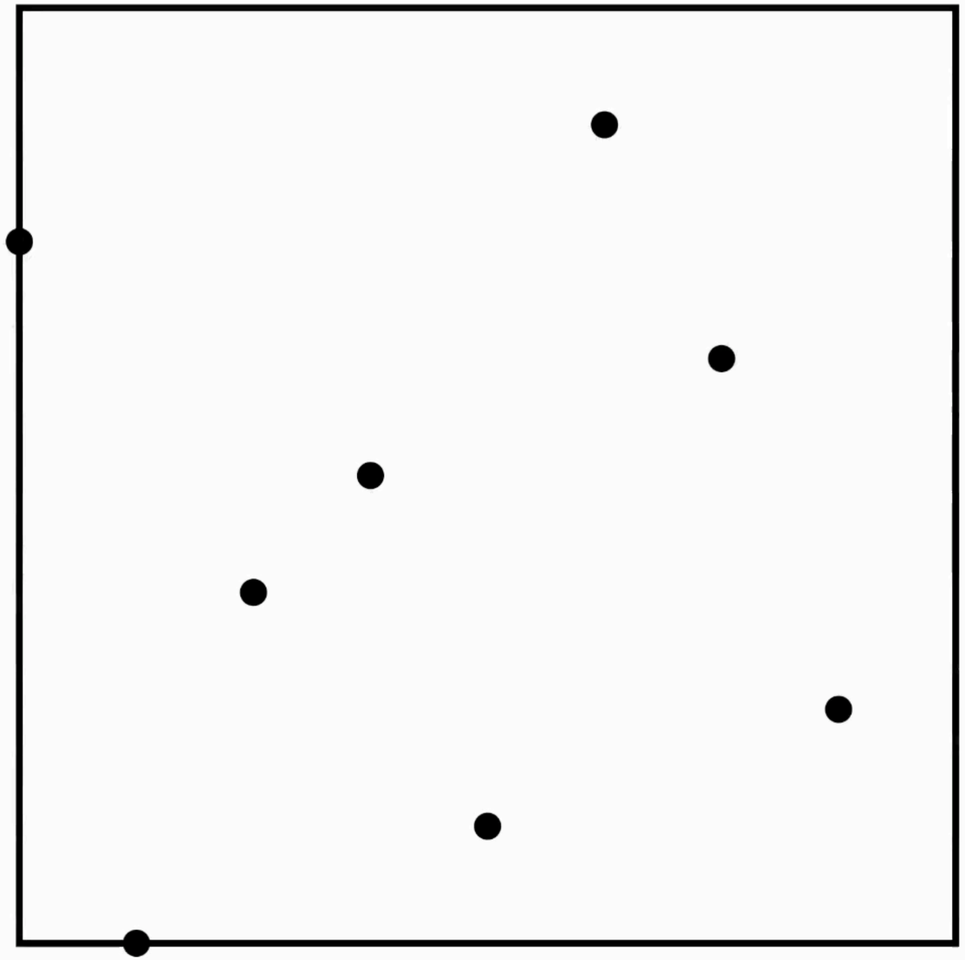




$$\sum b_i 2^{-i} \;\rightarrow\; \sum \pi_{b_1,\cdots,b_{i-1}}(b_i) 2^{-i}$$

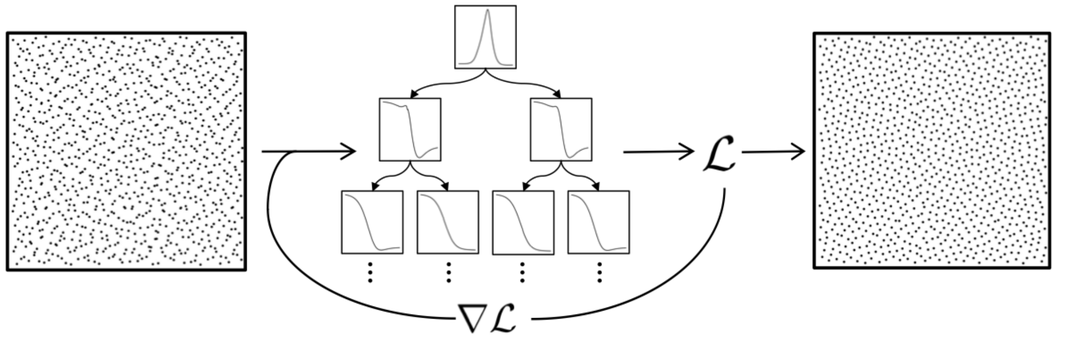
Differentiable Owen's scrambling





$$\sum b_i 2^{-i} \;\rightarrow\; \sum \pi_{b_1,\cdots,b_{i-1}}(b_i) 2^{-i}$$

Contributions



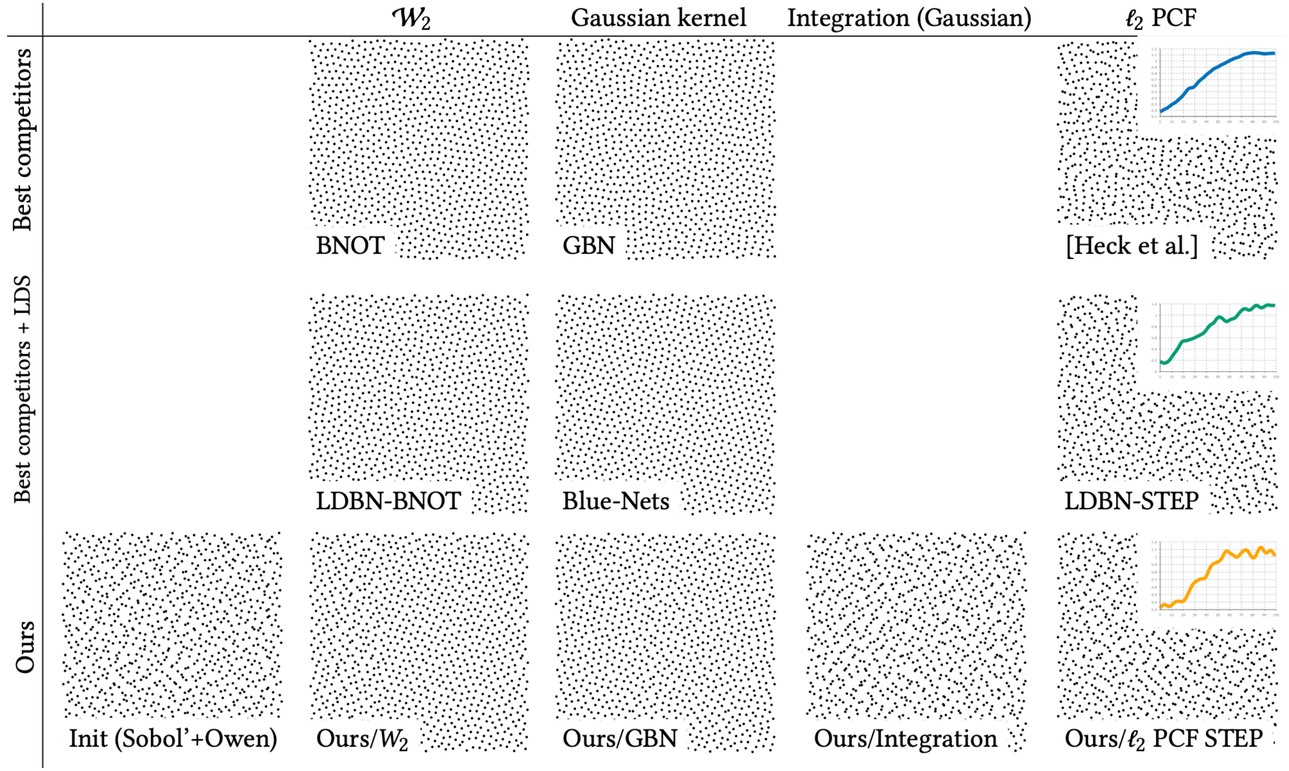
- One can differentiate Owen's tree
- Optimize for various loss functions staying low discrepancy

$$\mathcal{W}_2(X) = \inf_T \sum_i \int_{T^{-1}(x_i)} \|x - x_i\|^2 \mathrm{dx}$$

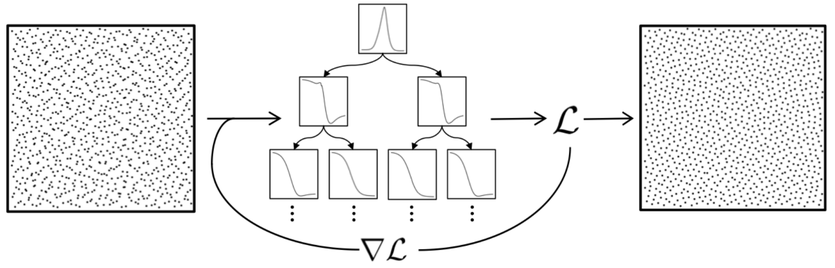
$$\mathcal{G}(X) = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{n} e^{-\|x_i - x_j\|^2/(2\sigma'^2)}$$

$$\mathcal{I}(X) = \frac{1}{K} \sum_{k=1}^{K} \left| \int g_k(x)dx - \frac{1}{n} \sum_{i=1}^{n} g_k(x_i) \right|^2$$

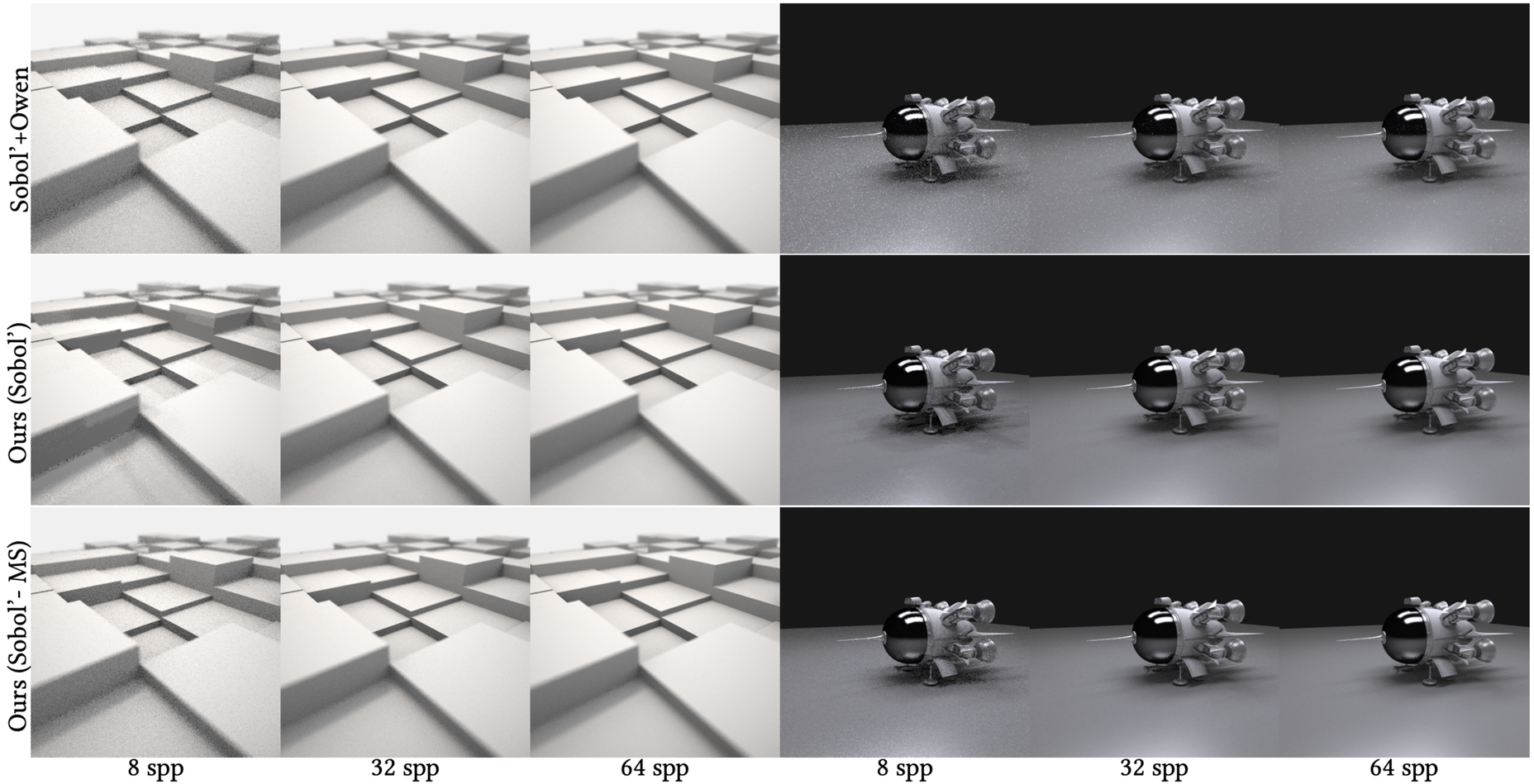
$$\mathcal{P}(X) = \int_{r=r_{\min}}^{r_{\max}} |\mathrm{PCF}(X,r) - \widetilde{\mathrm{PCF}}(r)|^2 \mathrm{dr}$$



Contributions



- nD low discrepancy sequence with projective blue noise



Loss: $\mathcal{W}_2(\text{proj}_{x^0x^1}(X)) + \mathcal{W}_2(\text{proj}_{x^2x^3}(X)) + \mathcal{W}_2(\text{proj}_{x^4x^5}(X))$

Conclusion

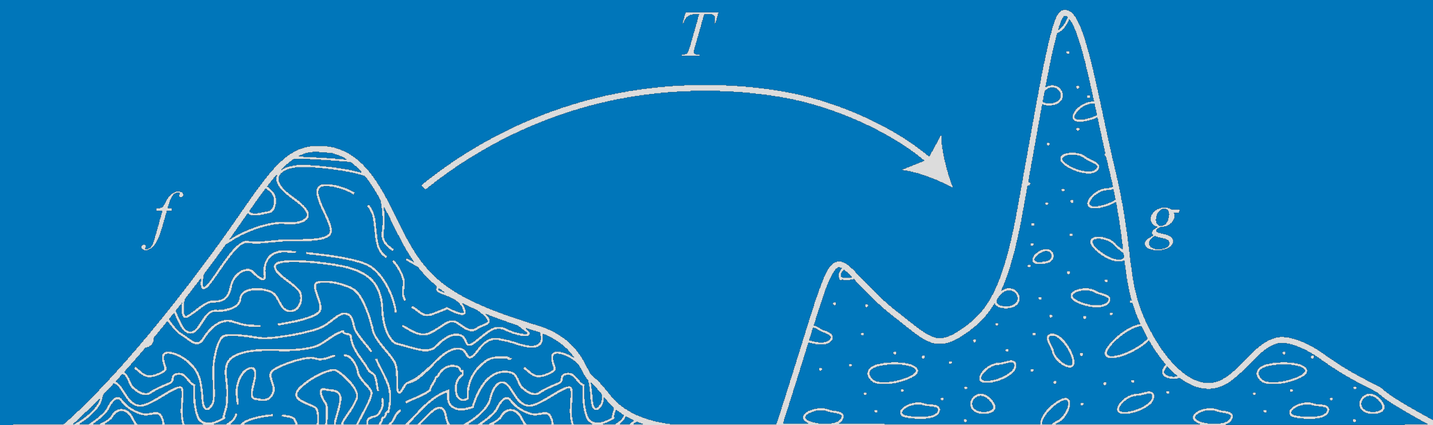
- **Algebraic construction**
  - one small (boolean) matrix per dimension
  - Highly optimized matrix/vector multiplication in GF(2)
  - Strong uniformity guarantees

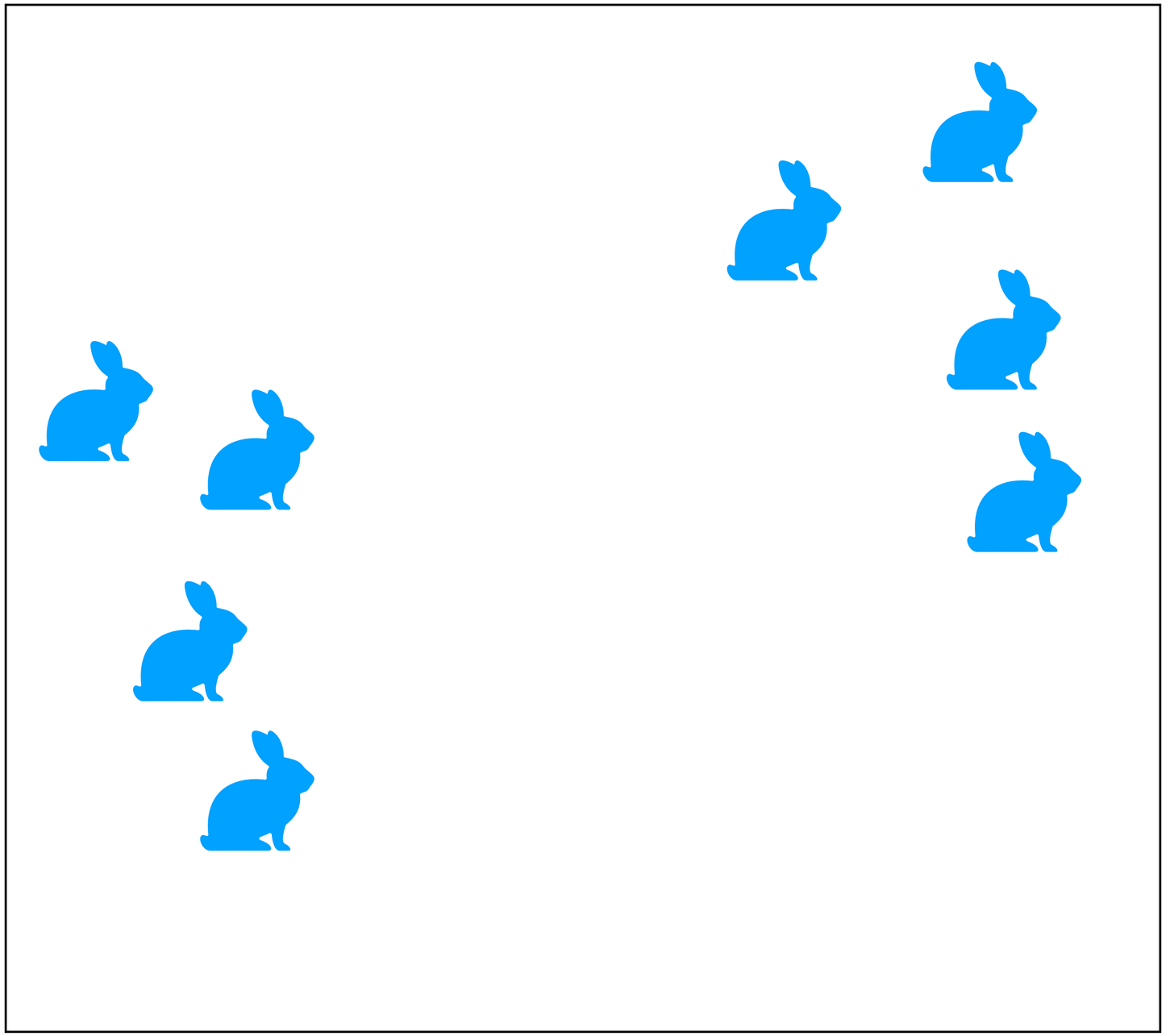
- **Low discrepancy point set optimization with meaningful gradients**
  - Eg. LDS + W_2 = 🎉
  - Projective sampling
  - Works in arbitrary dimensions, with any (differentiable) loss function

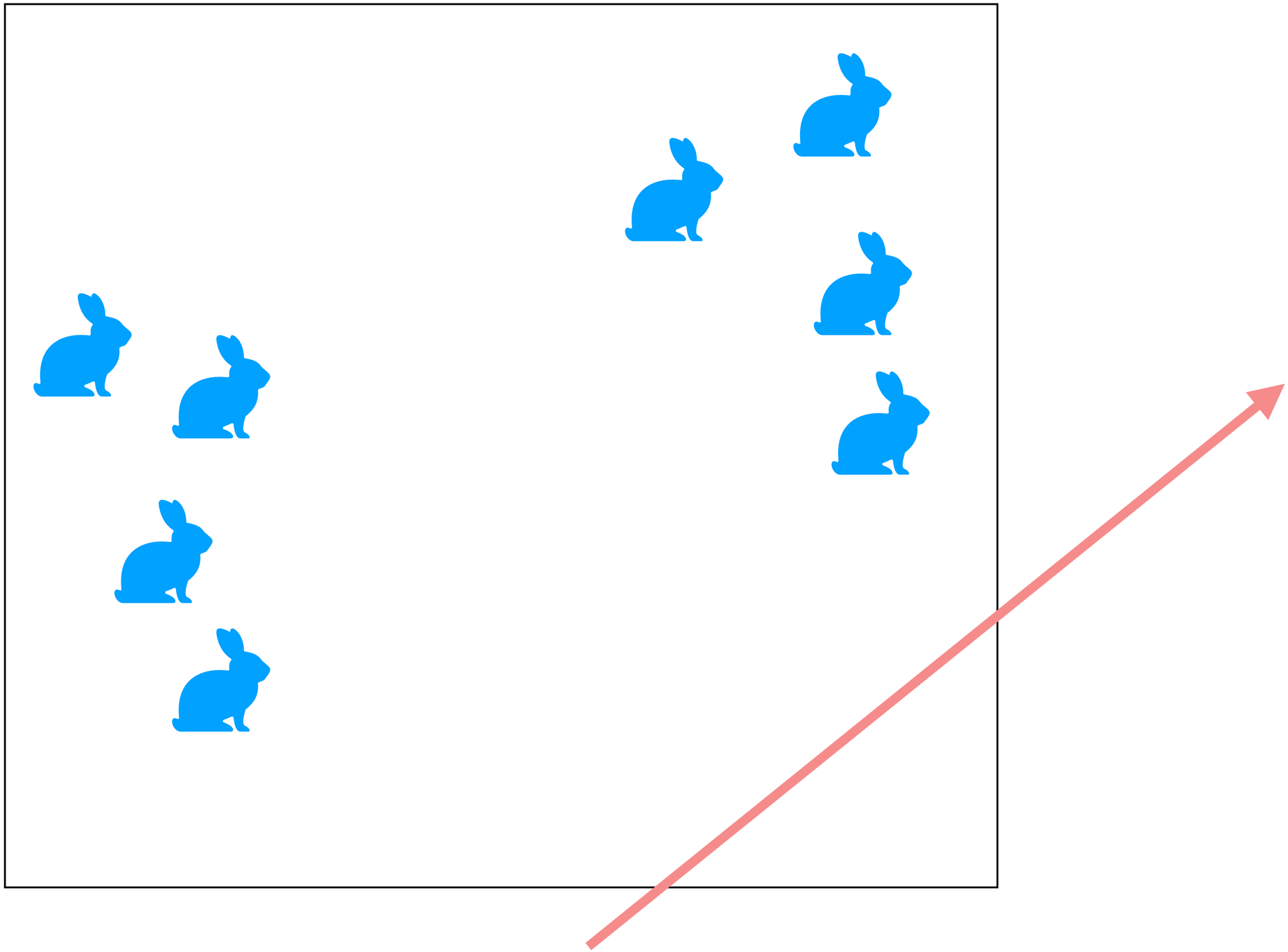
Non-Euclidean Sliced Optimal Transport Sampling

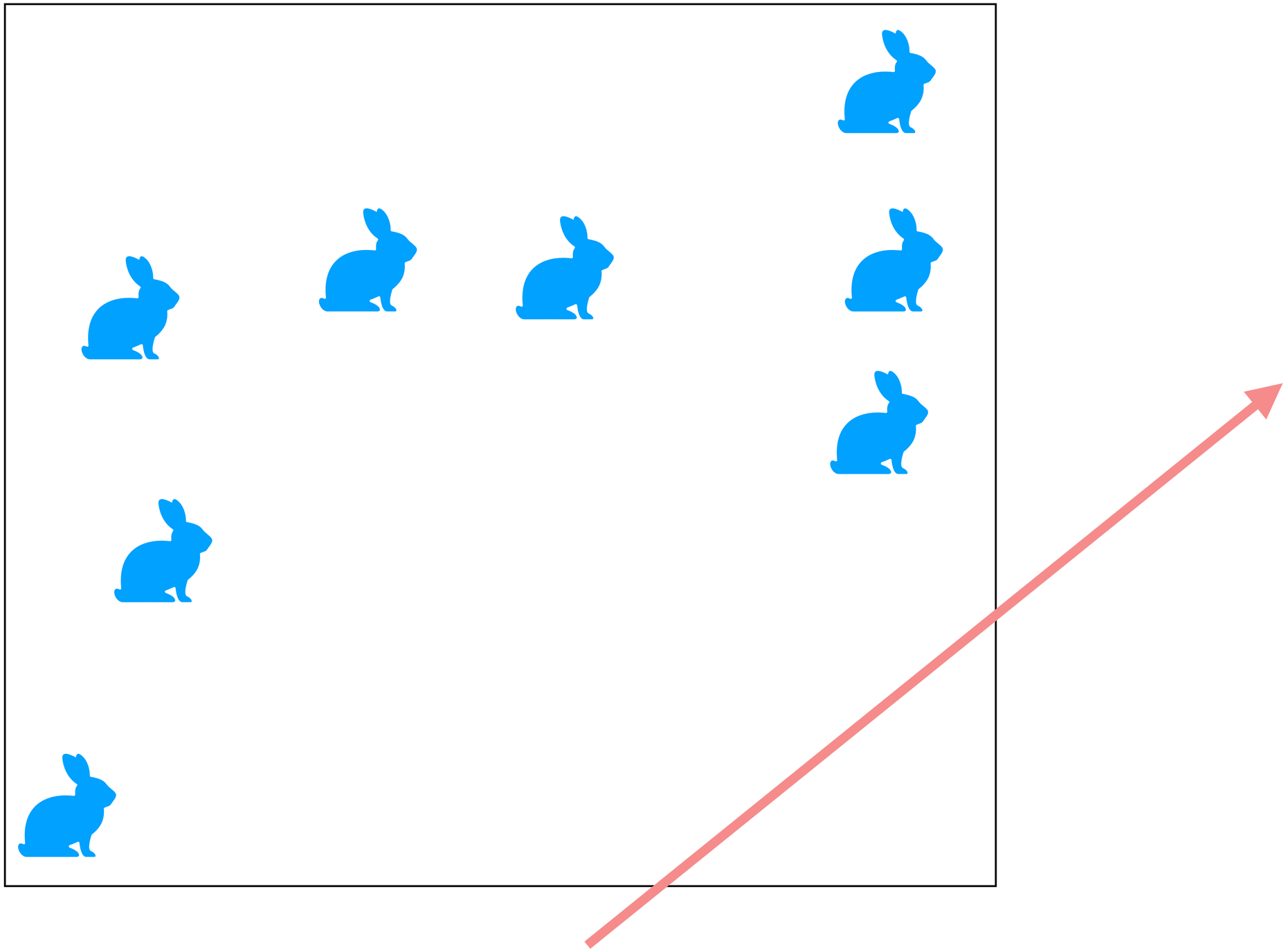
Loïs Paulin, Nicolas Bonneel, D. C., Jean-Claude Iehl, Antoine Webanck,
Mathieu Desbrun, Victor Ostromoukhov
ACM Transactions on Graphics (Proceedings of SIGGRAPH), July, 2020

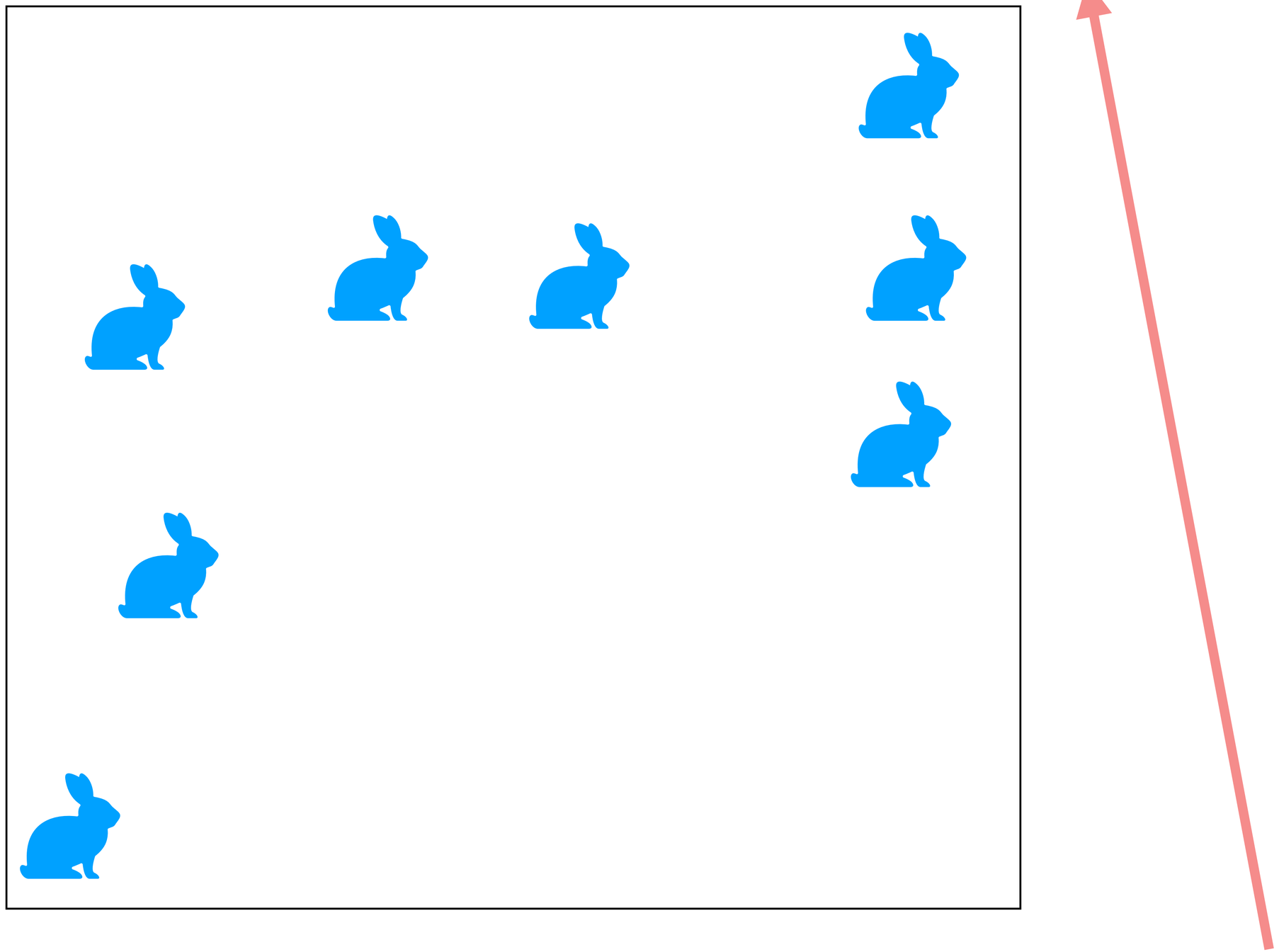
Baptiste Genest, Nicolas Courty, D. C.
Computer Graphics Forum (Proceedings of Eurographics),
April, 2024

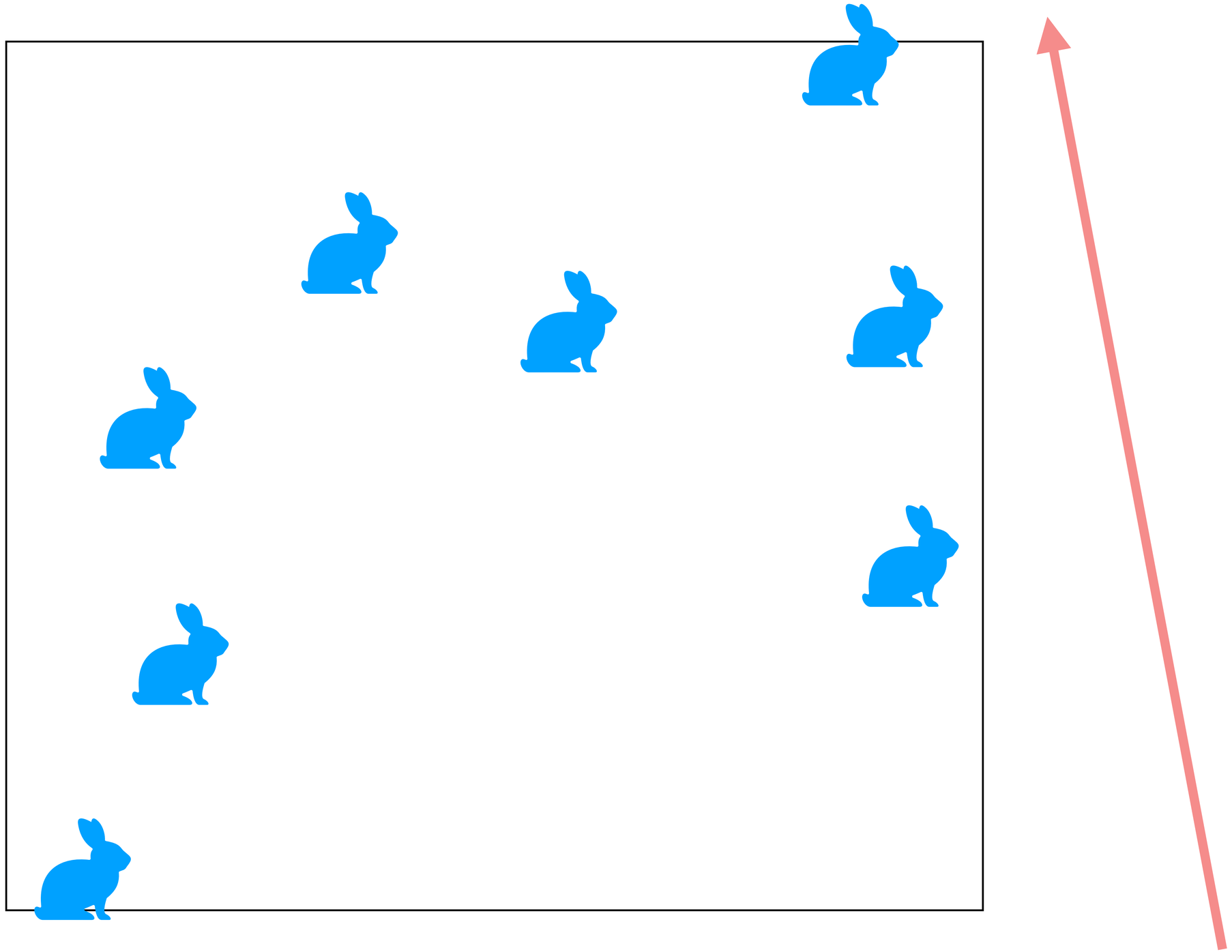


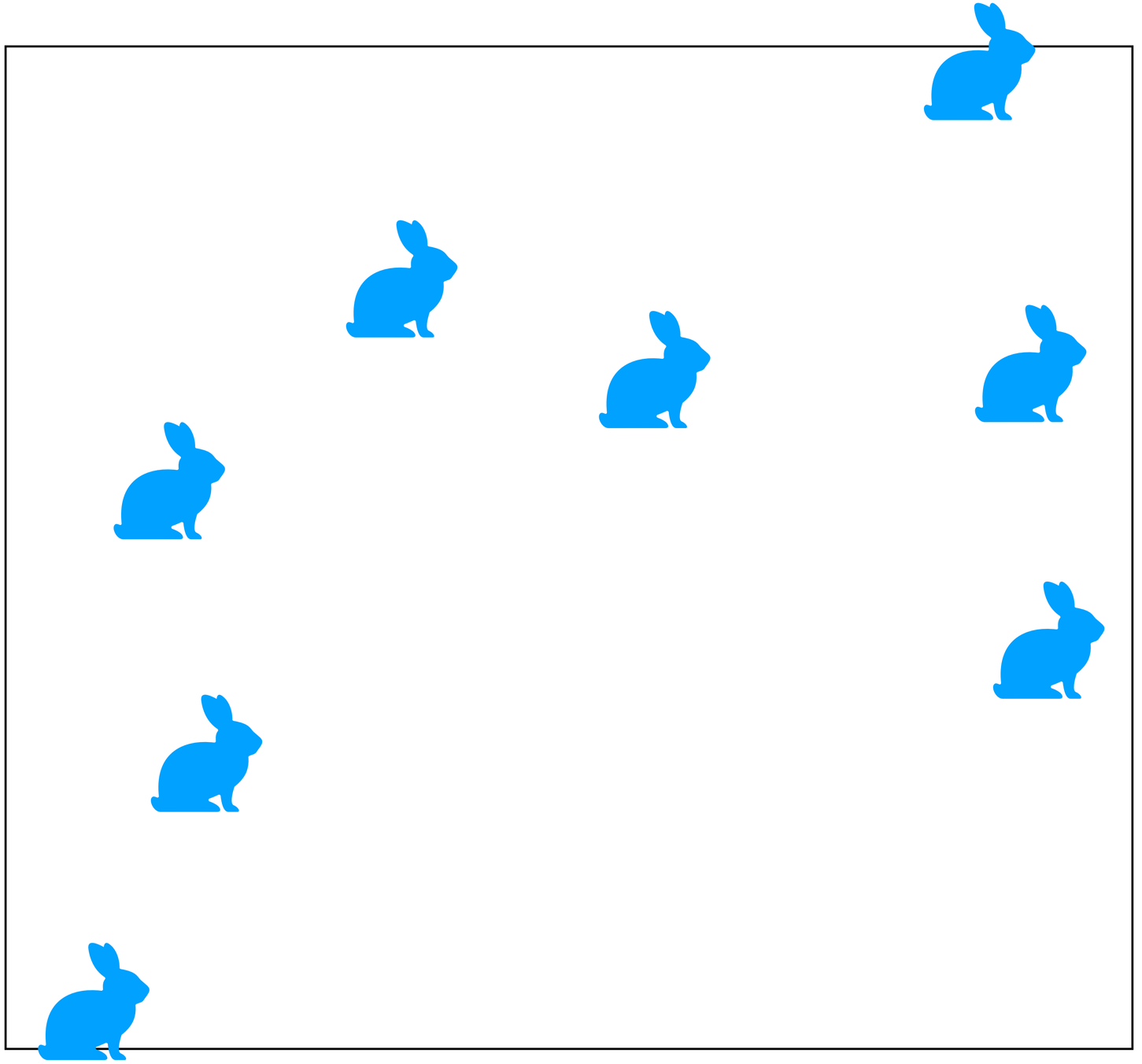


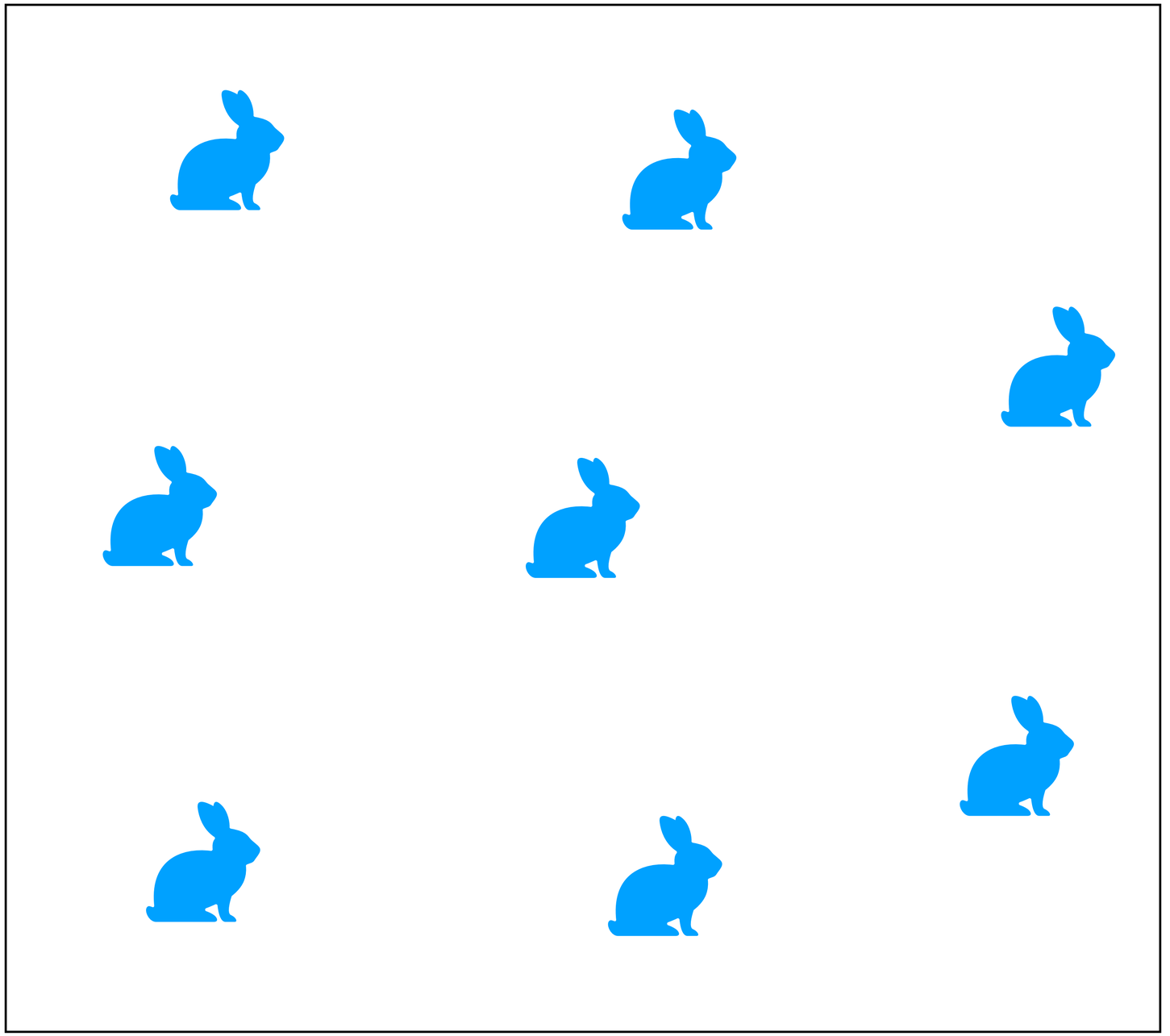






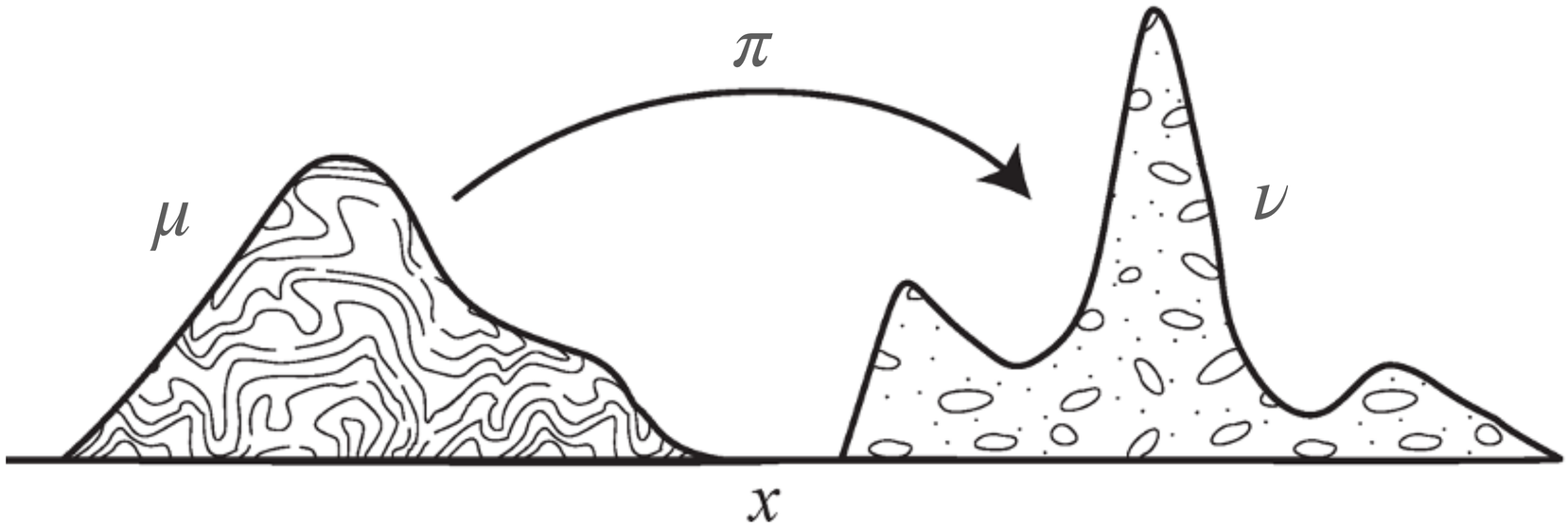






1d sliced advection ≈ stochastic gradient descent on W_{SOT}

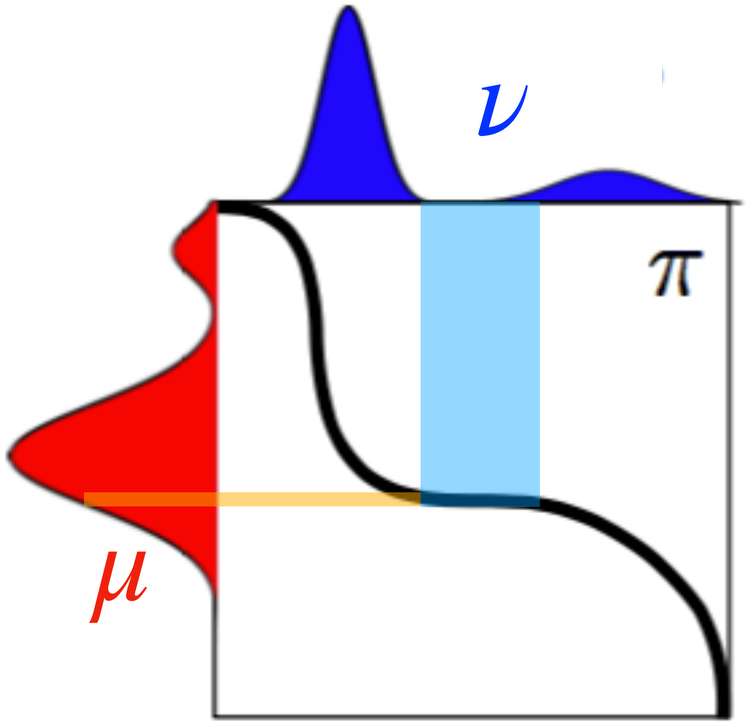
Core idea



Generic measures: discrete measures (sum of Diracs), histograms, continuous pdf….

Wasserstein distance

$$W_p(\mu,\nu) := \left(\inf_{\pi \in \Gamma(\mu,\nu)} \int_{X \times Y} c(x,y) d\pi(x,y) \right)^{1/p}$$

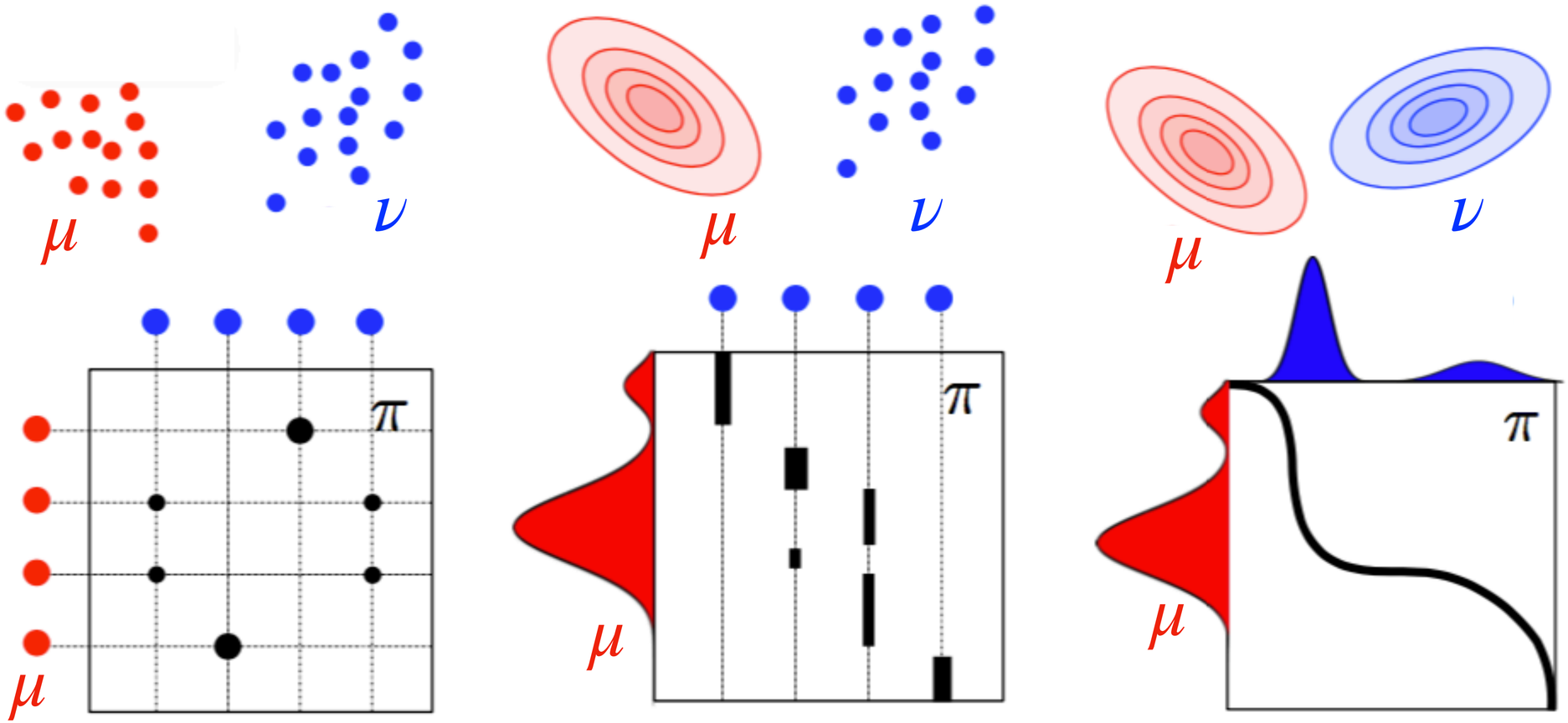


[Monge, Kantorovich, ...]

$$c(x,y) := \|x-y\|_p^p$$

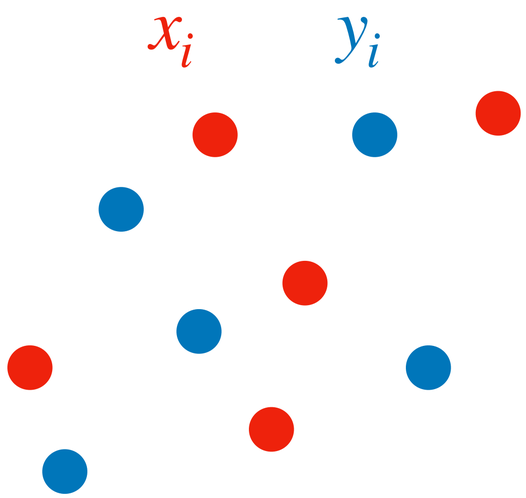
$$\pi \in \Gamma(\mu,\nu) \Leftrightarrow \begin{cases} \int_Y d\pi(x,y) = d\mu(x) \\ \int_X d\pi(x,y) = d\nu(y) \\ \pi(x,y) \geq 0 \end{cases}$$

Generic tool

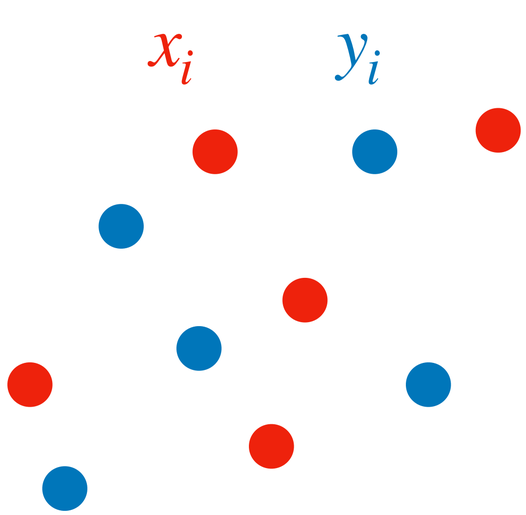


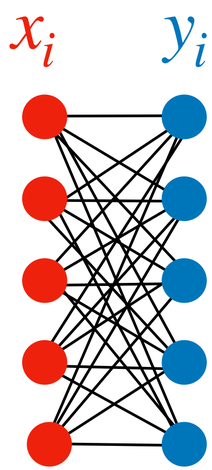
Many numerical solutions: linear programming, Hungarian algorithm, Network simplex, Sinkhorn like entropic regularization, semi-discrete from Power diagrams, PDE / flow approach, sliced...

Kantorovich discrete OT = Linear program

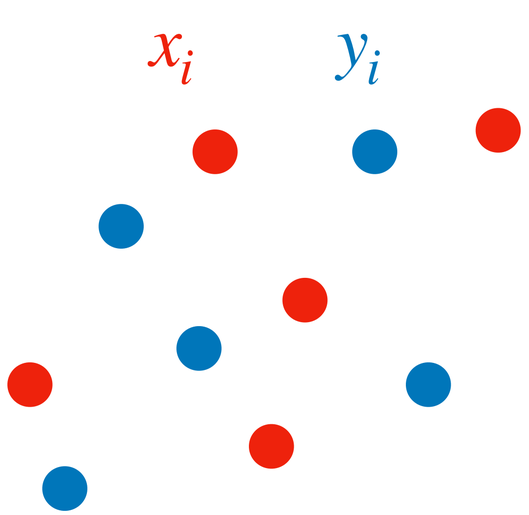


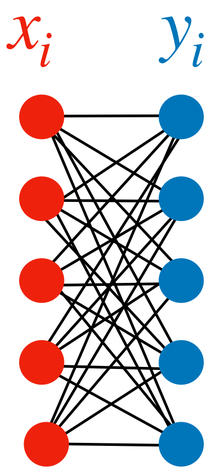
Kantorovich discrete OT = Linear program





Kantorovich discrete OT = Linear program





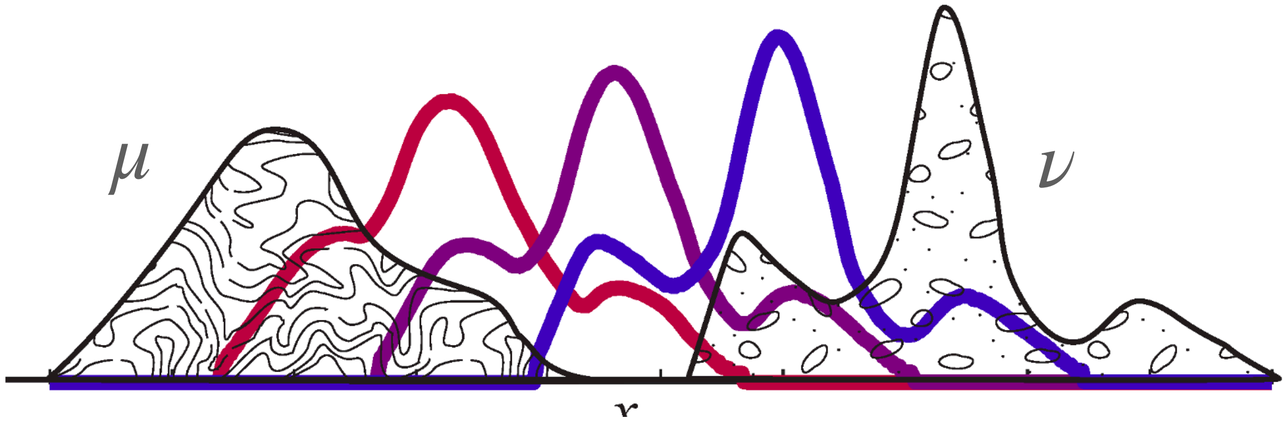
$$W(f_0, f_1) = \min_{\mathbf{P}} \sum_{i,j} c(x_i, y_j) P_{i,j} \quad (1)$$

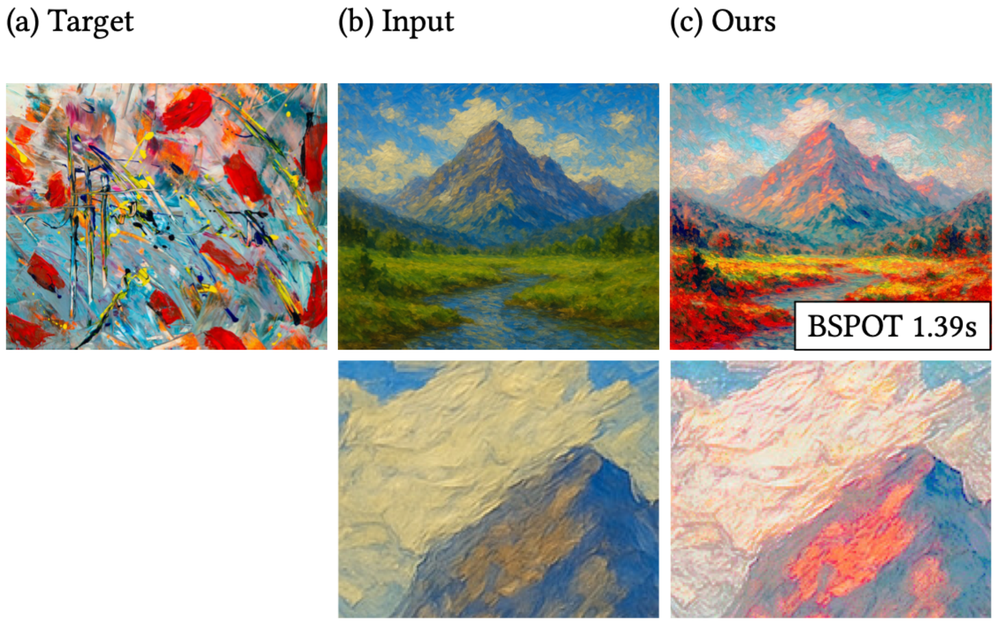
$$\text{s.t.} P_{i,j} \geq 0, \forall i, j \quad (2)$$

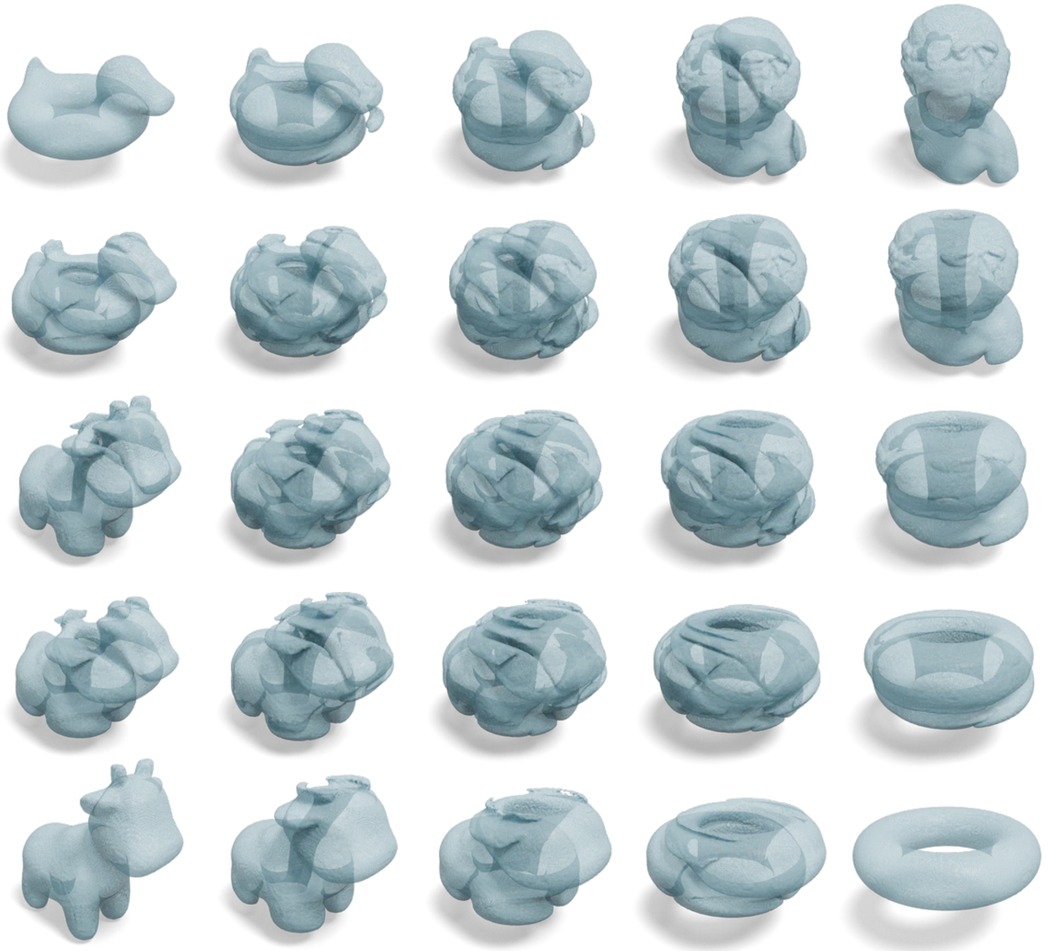
$$\sum_{j=1}^{m} P_{i,j} = f_0(x_i), \forall i \quad (3)$$

$$\sum_{i=1}^{n} P_{i,j} = f_1(x_j), \forall j \quad (4)$$

Interpolation and Wasserstein barycenters







$$\nu^* = argmin_\nu \sum_k \lambda_k W_2^2(\nu, \mu_k)$$

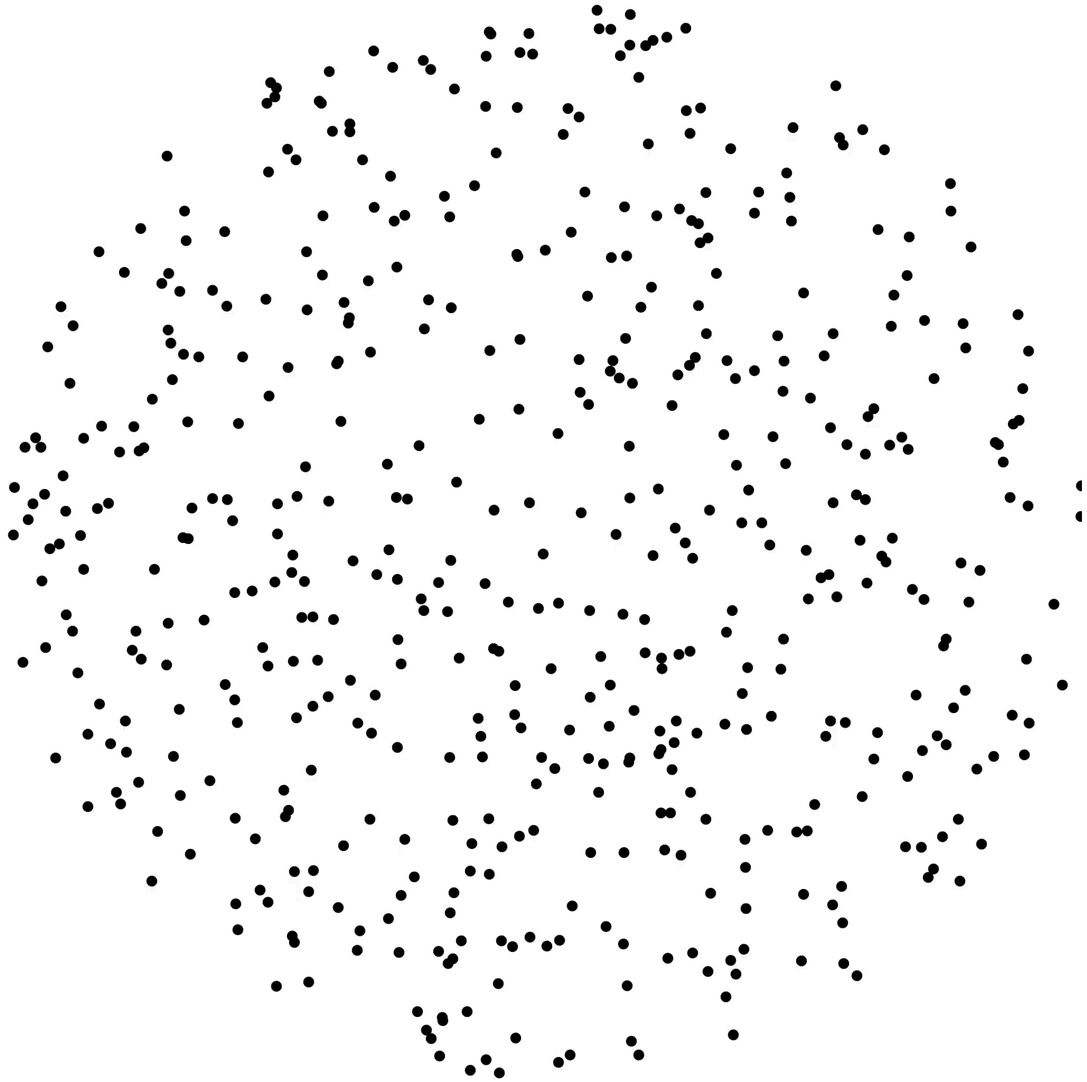
Sampling & OT ?

[Kantorovich-Rubinstein 58]

$$\left| \int f dx - \frac{1}{n} \sum f(x_i) \right| \leq W_1(X_n, \phi) \, \mathrm{Lip}(f)$$

Problem: Find X_n such that W_1 is minimal

$\Rightarrow$ Gradient based descent from $\nabla W(X_n, \phi)$!



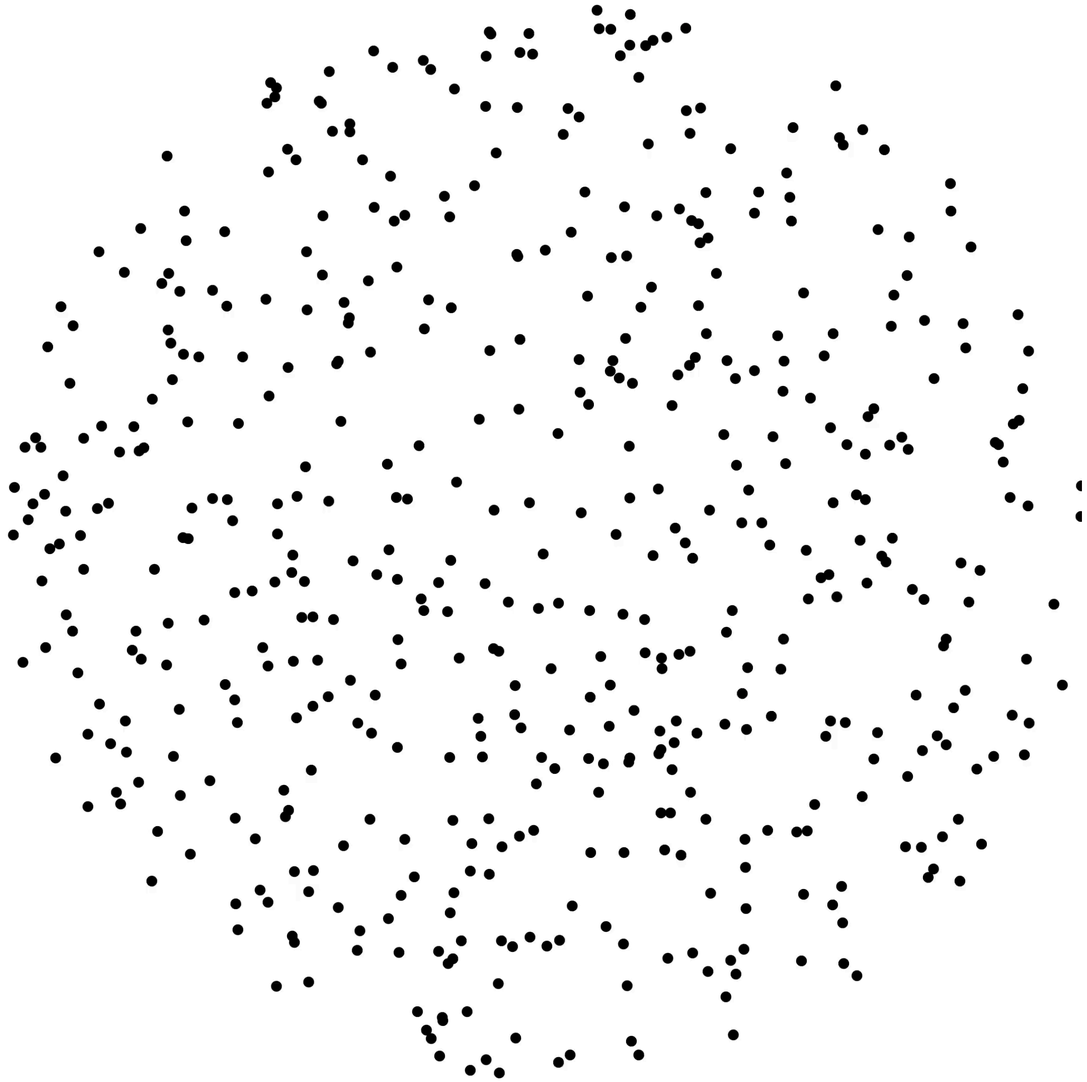
Sampling & OT ?

[Kantorovich-Rubinstein 58]

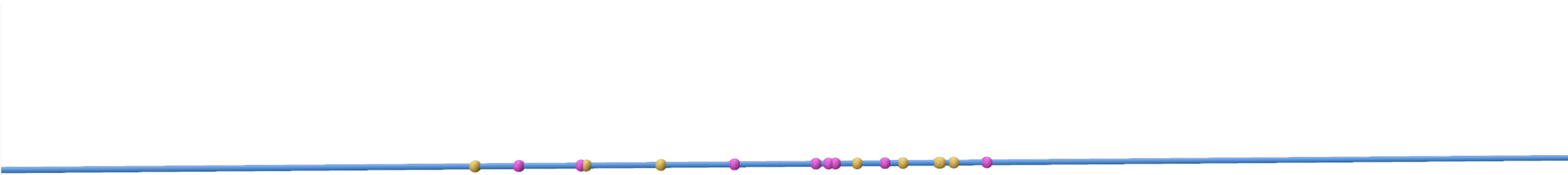
$$\left| \int f dx - \frac{1}{n} \sum f(x_i) \right| \leq W_1(X_n, \phi) \, \mathrm{Lip}(f)$$

Problem: Find X_n such that W_1 is minimal

$\Rightarrow$ Gradient based descent from $\nabla W(X_n, \phi)$!



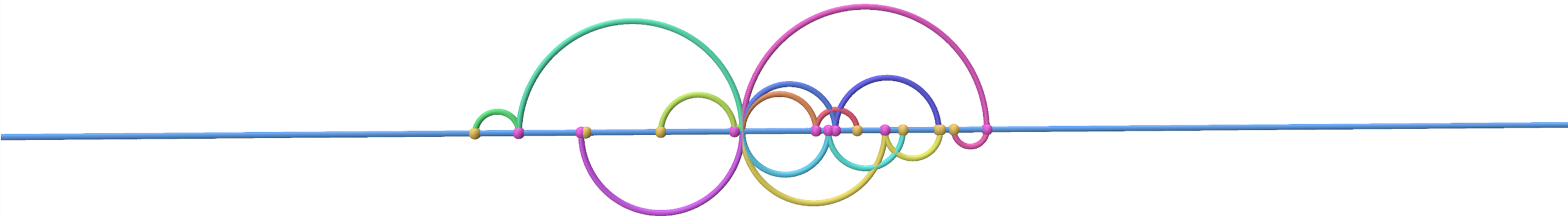
Sliced Optimal Transport



- 1D case: sorting + advection $\Rightarrow O(n \log n)$

[Pitié et al 2006, Rabin et al 2011, Bonnotte 2013, Bonneel et al 2015, Paulin et al 2020]

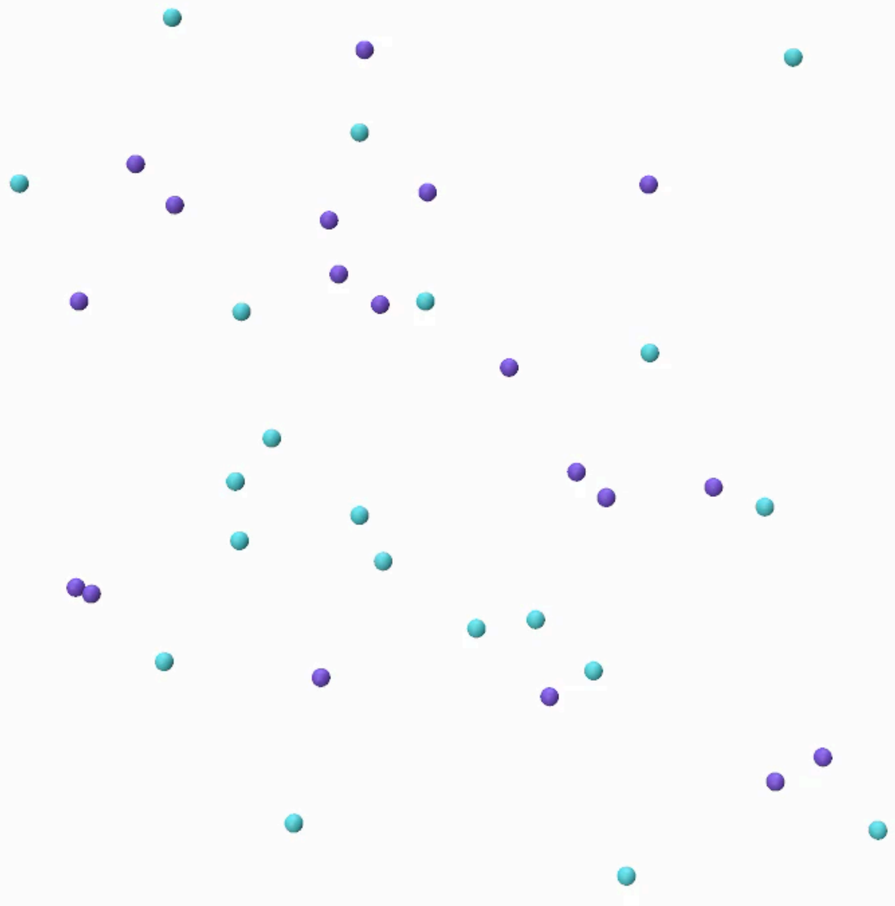
Sliced Optimal Transport



- 1D case: sorting + advection $\Rightarrow O(n \log n)$

[Pitié et al 2006, Rabin et al 2011, Bonnotte 2013, Bonneel et al 2015, Paulin et al 2020]

Sliced Optimal Transport

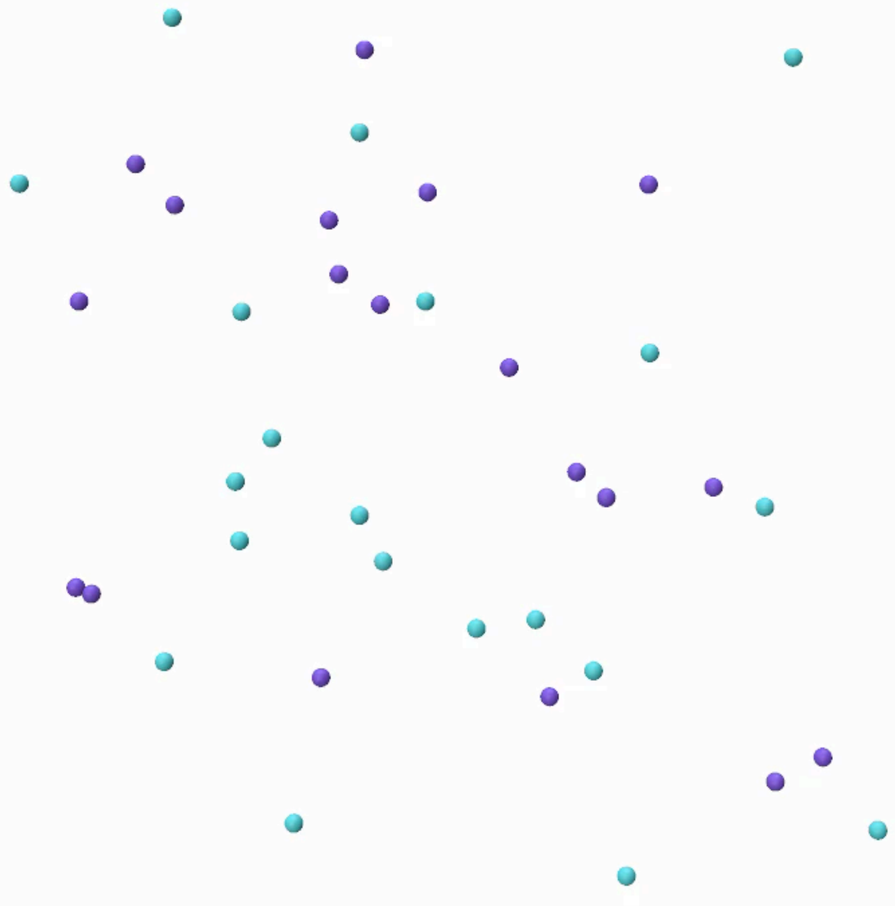


$$\frac{1}{L}\sum_{i}^{L}\mathcal{W}(P^{\theta_i}_{\#}\mu, P^{\theta_i}_{\#}\nu) \underset{L\to\infty}{\longrightarrow} \int_{\mathbb{S}}\mathcal{W}(P^{\theta}_{\#}\mu, P^{\theta}_{\#}\nu)d\theta = SW(\mu,\nu)$$

$$\approx \mathcal{W}(\mu,\nu)$$

[Pitié et al 2006, Rabin et al 2011, Bonnotte 2013, Bonneel et al 2015, Paulin et al 2020]

Sliced Optimal Transport

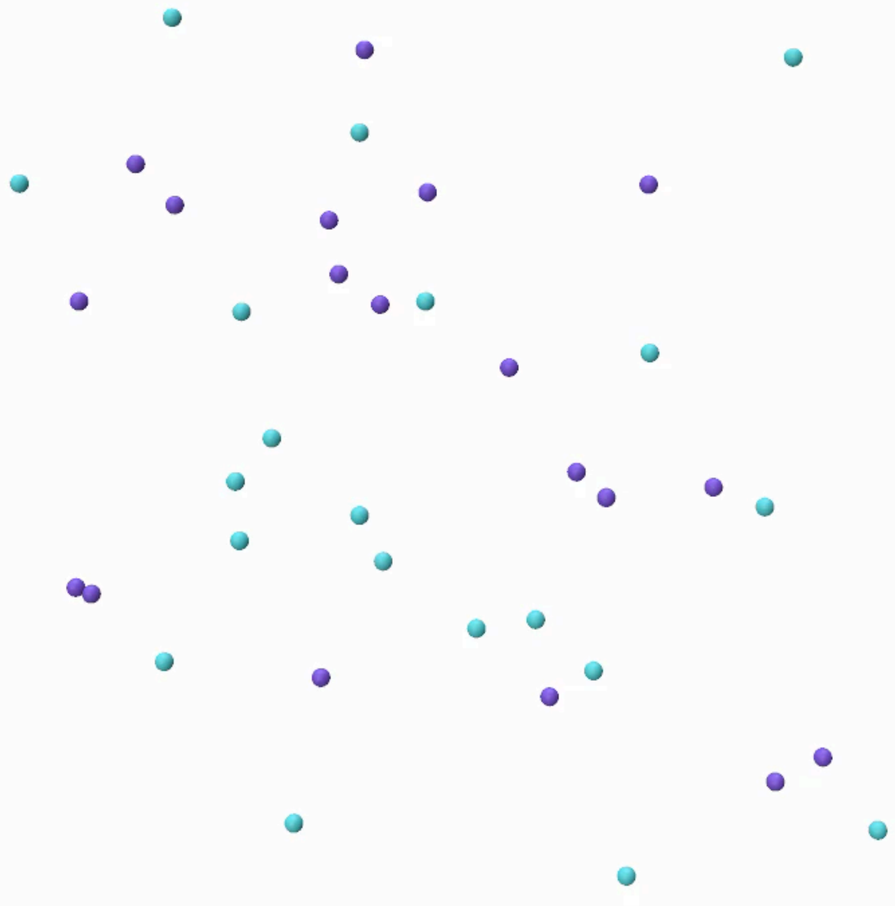


$$\frac{1}{L}\sum_{i}^{L}\mathcal{W}(P^{\theta_i}_{\#}\mu, P^{\theta_i}_{\#}\nu) \underset{L\to\infty}{\longrightarrow} \int_{\mathbb{S}}\mathcal{W}(P^{\theta}_{\#}\mu, P^{\theta}_{\#}\nu)d\theta = SW(\mu,\nu)$$

$$\approx \mathcal{W}(\mu,\nu)$$

[Pitié et al 2006, Rabin et al 2011, Bonnotte 2013, Bonneel et al 2015, Paulin et al 2020]

Sliced Optimal Transport



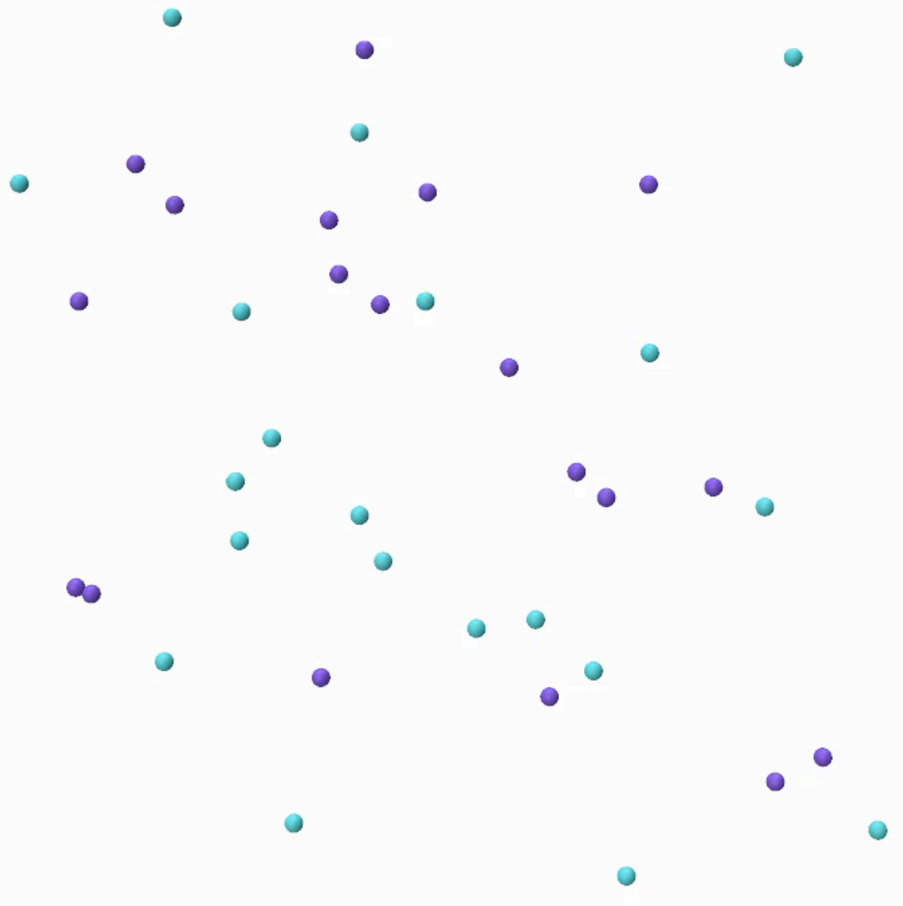
$$\frac{1}{L}\sum_{i}^{L}\mathcal{W}(P^{\theta_i}_{\#}\mu, P^{\theta_i}_{\#}\nu) \underset{L\to\infty}{\longrightarrow} \int_{\mathbb{S}}\mathcal{W}(P^{\theta}_{\#}\mu, P^{\theta}_{\#}\nu)d\theta = SW(\mu,\nu)$$

$$\approx \mathcal{W}(\mu,\nu)$$

$$\Delta_n \leq C_s\, SW(X_n, 1_\Omega)^{\frac{1}{s+1}}\ \mathrm{Lip}(f)$$

[Pitié et al 2006, Rabin et al 2011, Bonnotte 2013, Bonneel et al 2015, Paulin et al 2020]

Sliced Optimal Transport



$$\frac{1}{L}\sum_{i}^{L}\mathcal{W}(P^{\theta_i}_{\#}\mu, P^{\theta_i}_{\#}\nu) \underset{L\to\infty}{\longrightarrow} \int_{\mathbb{S}}\mathcal{W}(P^{\theta}_{\#}\mu, P^{\theta}_{\#}\nu)d\theta = SW(\mu,\nu)$$

$$\approx \mathcal{W}(\mu,\nu)$$

$$\Delta_n \leq C_s\, SW(X_n, 1_\Omega)^{\frac{1}{s+1}}\ \mathrm{Lip}(f)$$

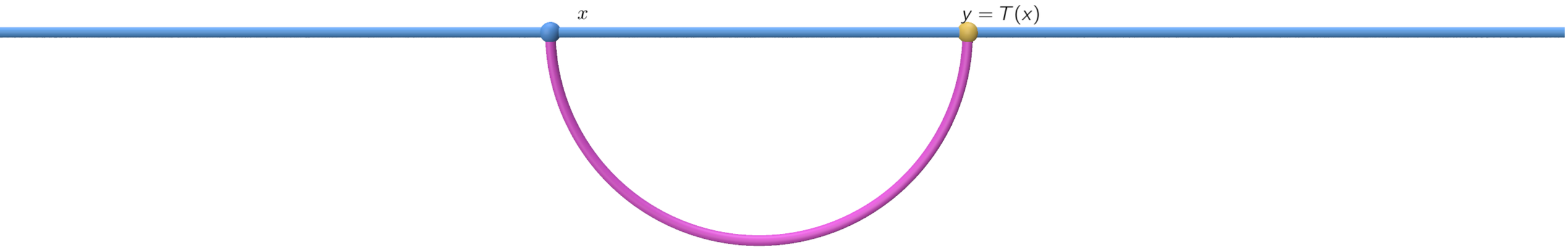
⇒Gradient descent on SW

[Pitié et al 2006, Rabin et al 2011, Bonnotte 2013, Bonneel et al 2015, Paulin et al 2020]

Sliced Optimal Transport Sampling

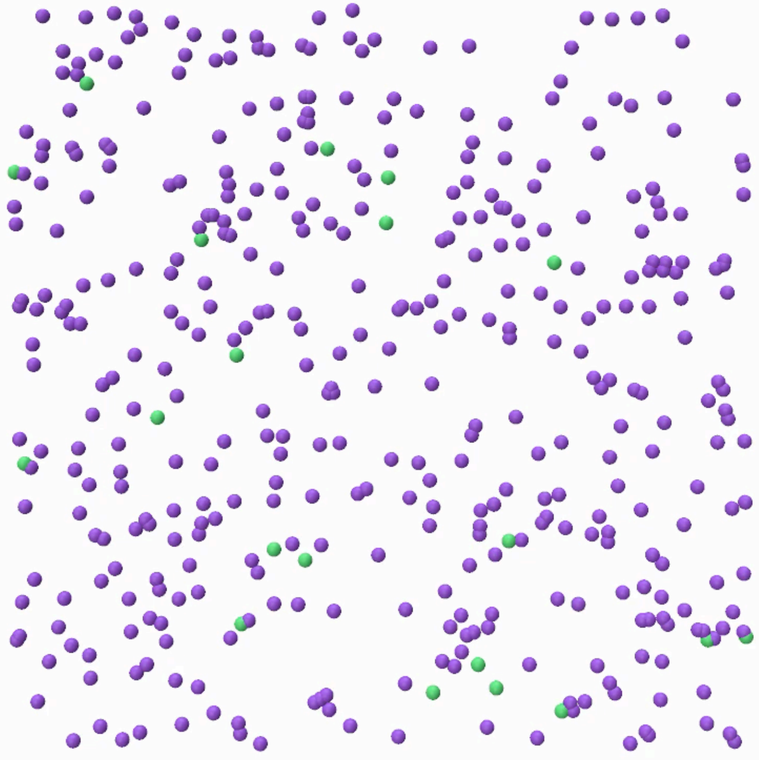
Stochastic Gradient Descent on $\mu \mapsto SW(\mu, \mathcal{U})$

$$SW^{\theta}(\delta_x, \delta_y) = \frac{1}{2}(x-y)^2 \implies \nabla_{x_i} SW^{\theta} = x_i - T^{\theta}(x_i)$$



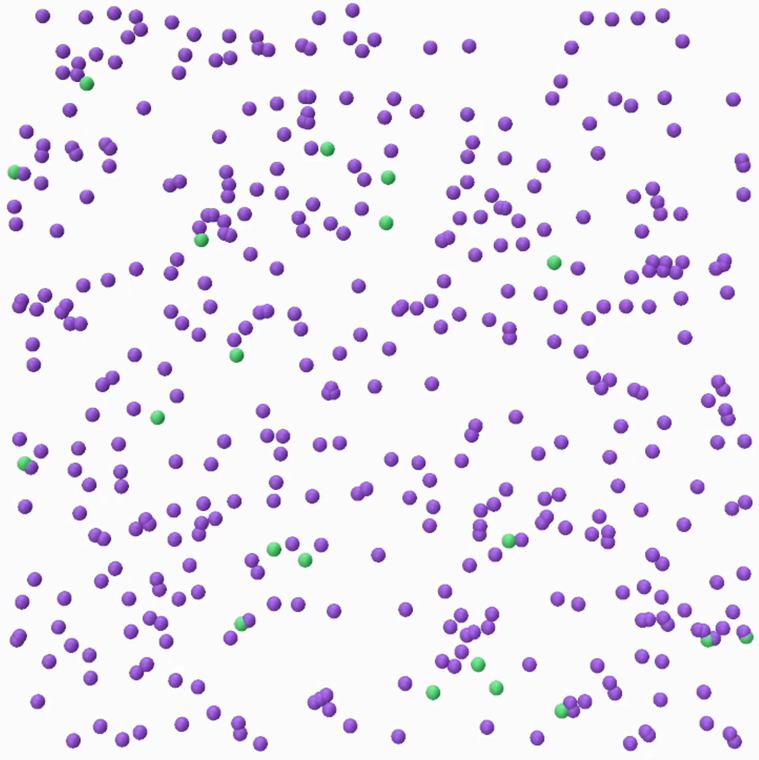
Sliced Optimal Transport Sampling

Stochastic Gradient Descent on $\mu \mapsto SW(\mu, \mathcal{U})$

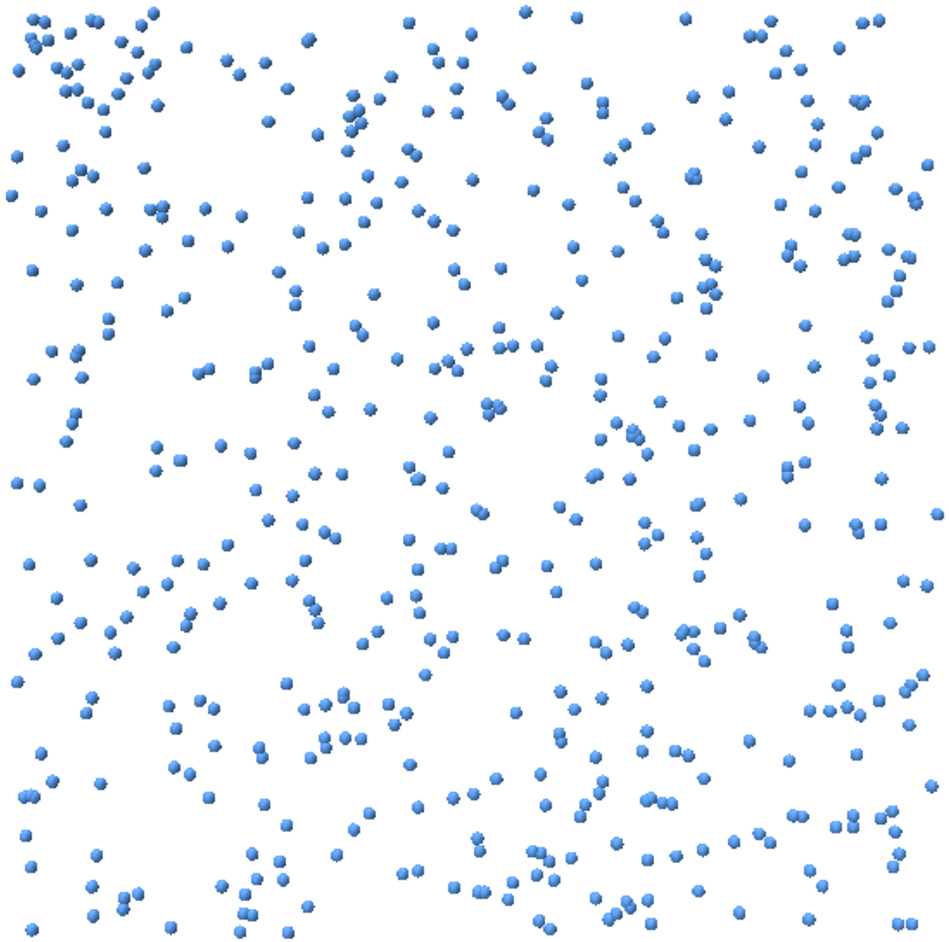


Sliced Optimal Transport Sampling

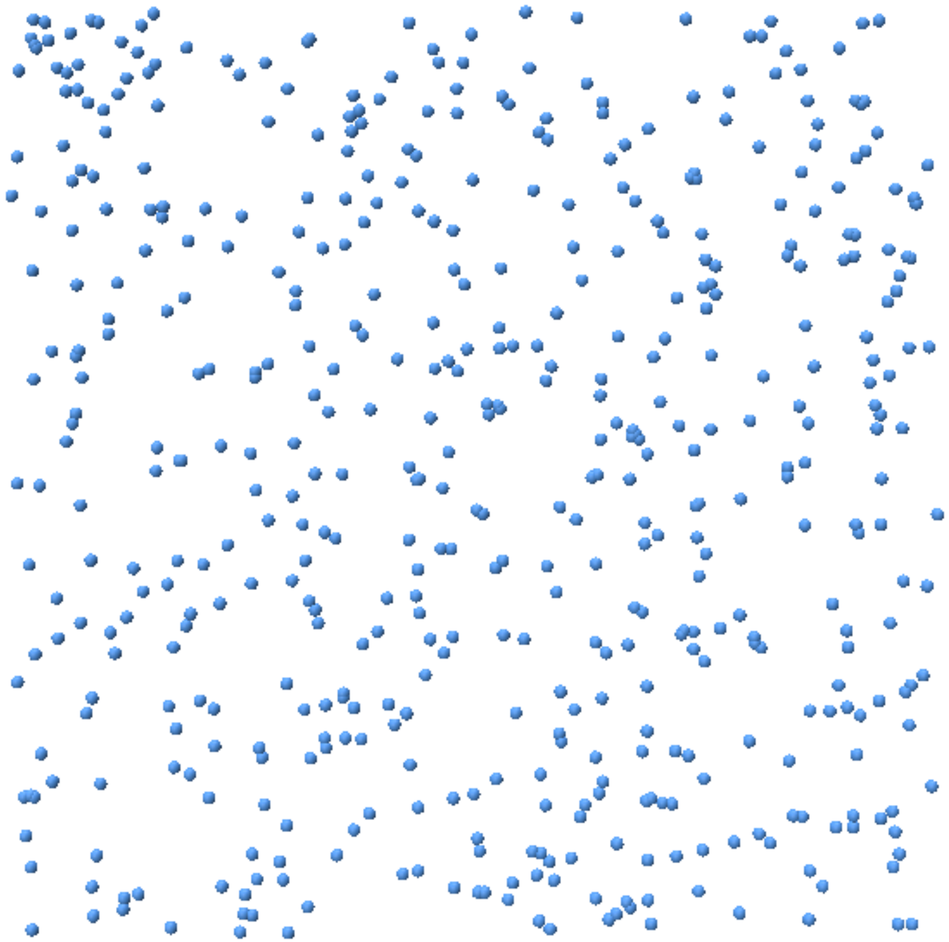
Stochastic Gradient Descent on $\mu \mapsto SW(\mu, \mathcal{U})$

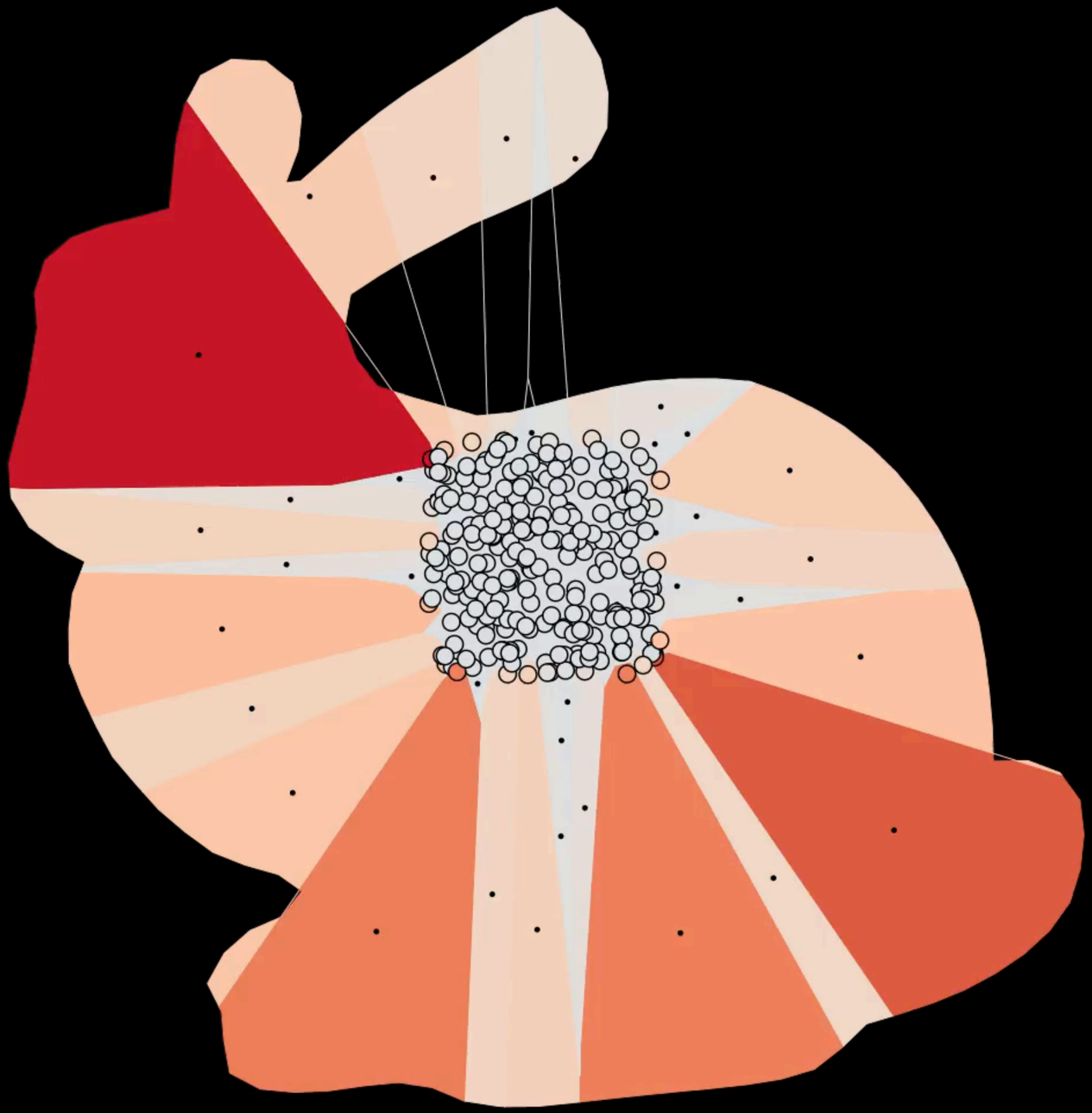


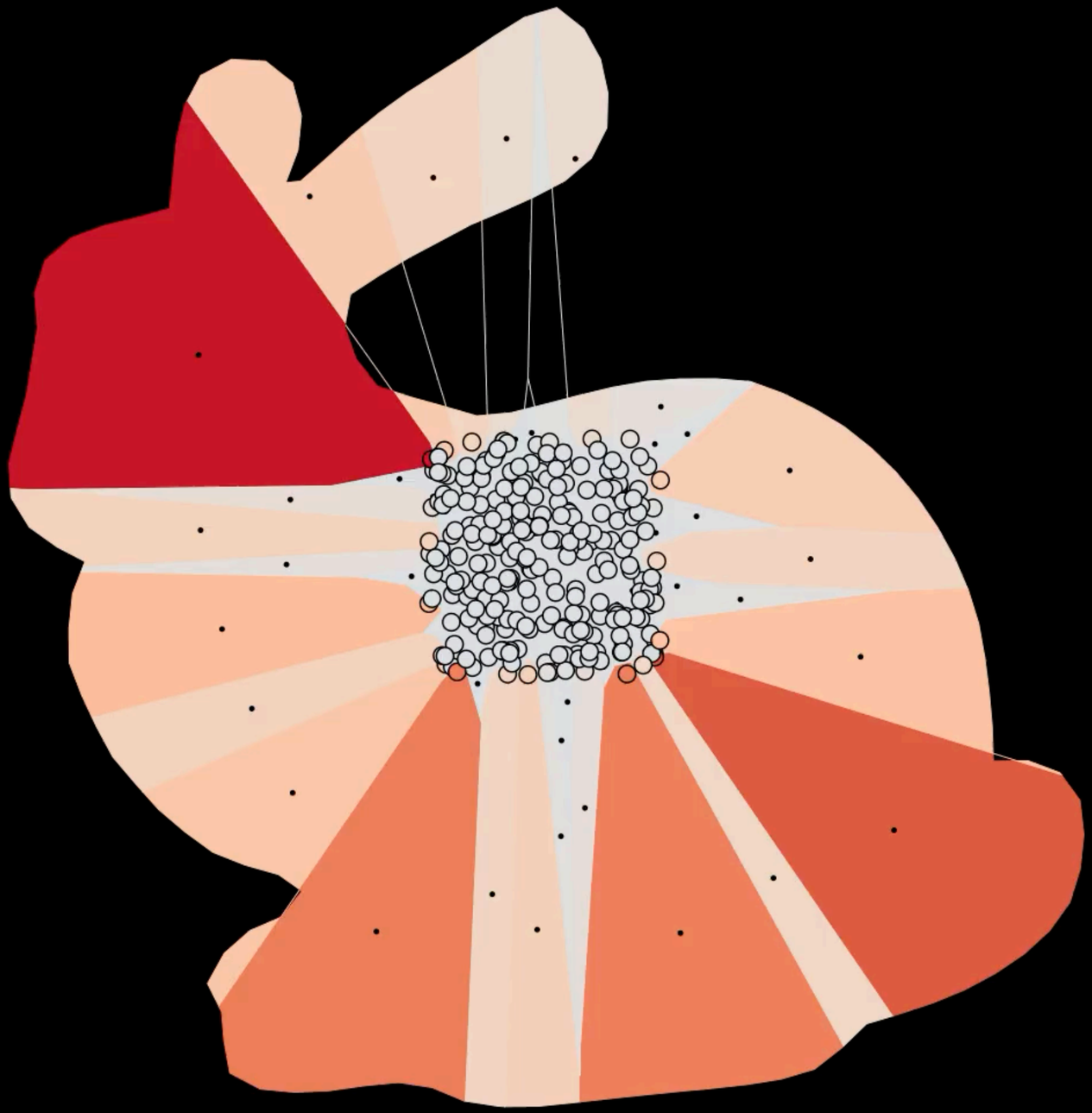
Sliced Optimal Transport Sampling



Sliced Optimal Transport Sampling





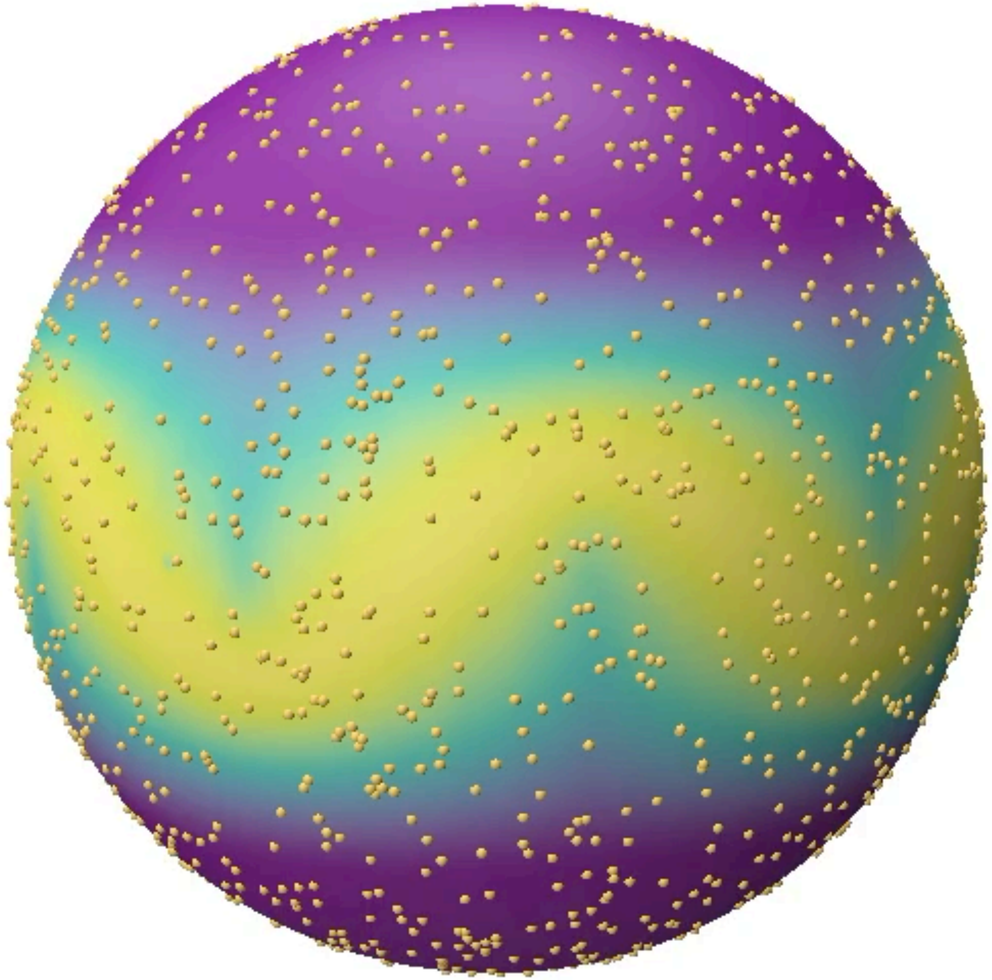


OT in CG / IP (just one example)

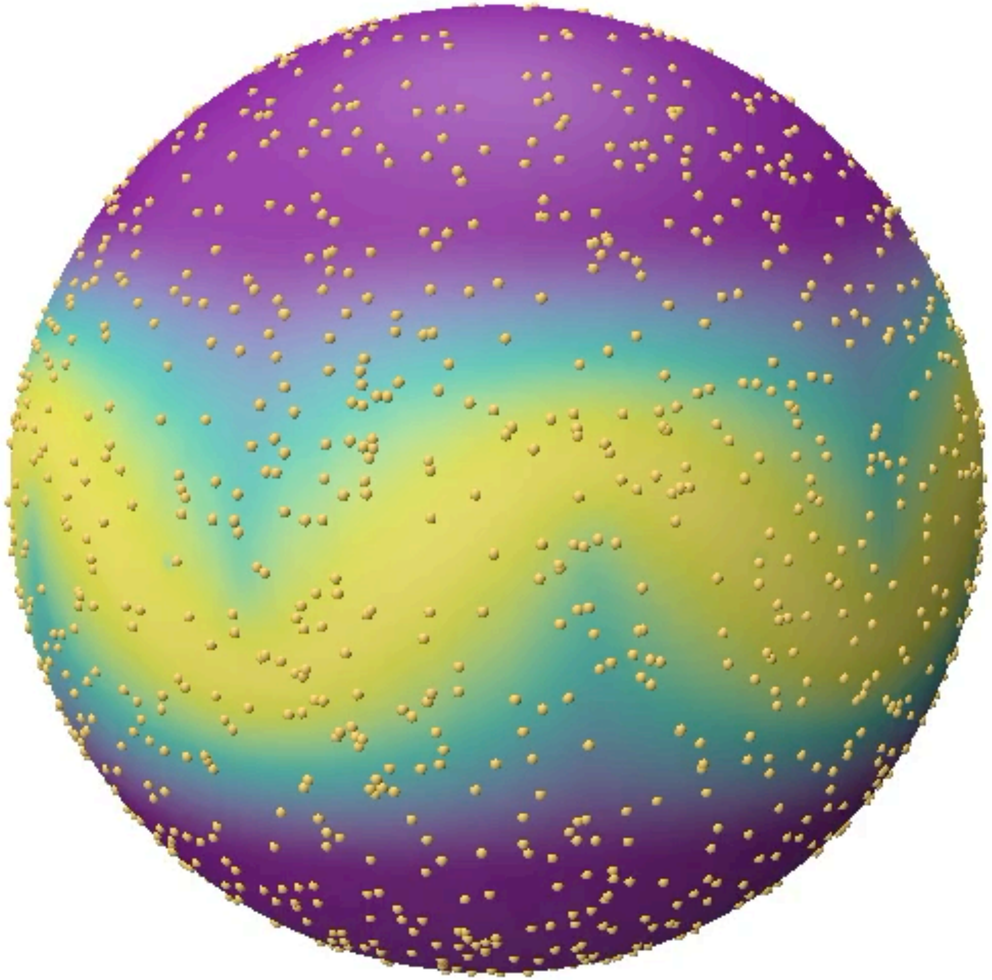


https://dcoeurjo.github.io/OTColorTransfer/

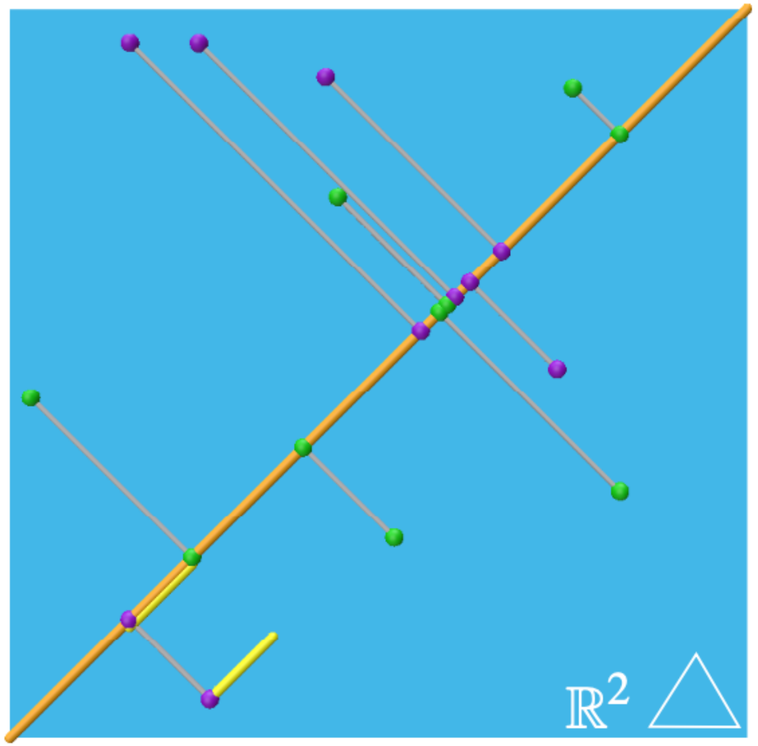
Non-Euclidean Sliced OT Sampling

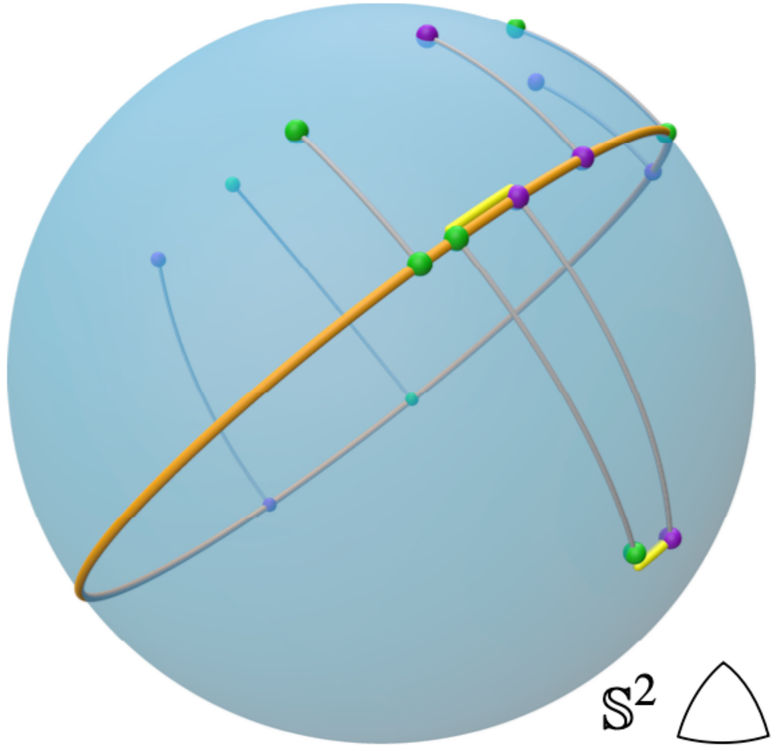


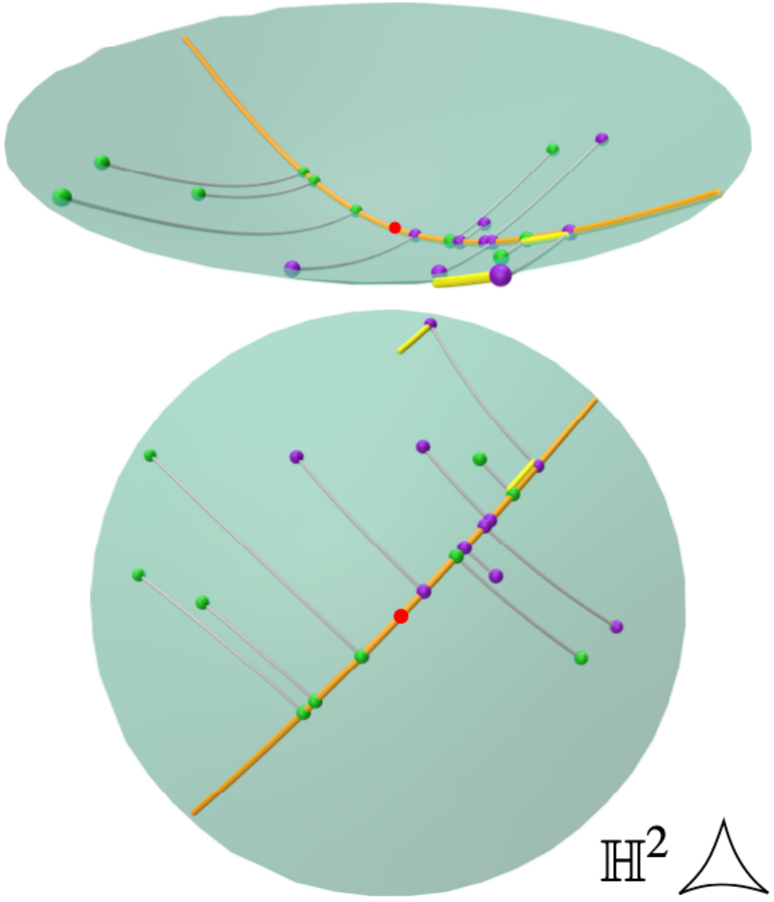
Non-Euclidean Sliced OT Sampling



Constant curvature manifolds ($\mathbb{R}^d$, $\mathbb{S}^d$, $\mathbb{H}^d$)







Ingredients

- Geodesic slices and projection
- Solving discrete 1D Wasserstein problem
- Group actions for the translation of samples
- Motions along geodesics from Exp / Log maps

$\mathbb{R}^d$

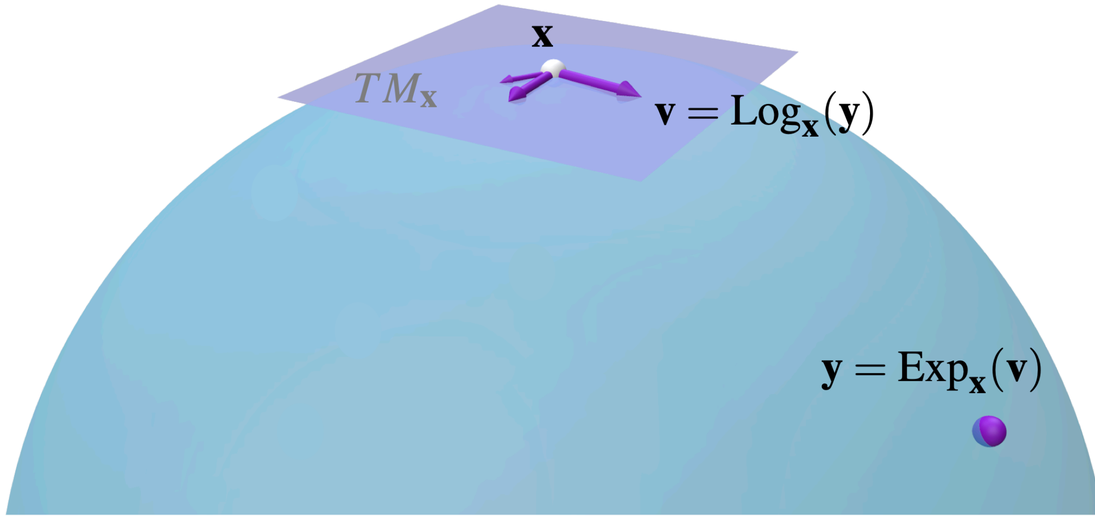
$$x_i^{n+1} = x_i^n - \tau \nabla_{x_i^n} SW$$

$$\nabla_{x_i} SW^\theta = T^\theta(x_i) - x_i$$

$\mathscr{M}$

$$x_i^{n+1} = \mathrm{Exp}_{x_i^n}(-\tau \nabla_{x_i^n} SW$$

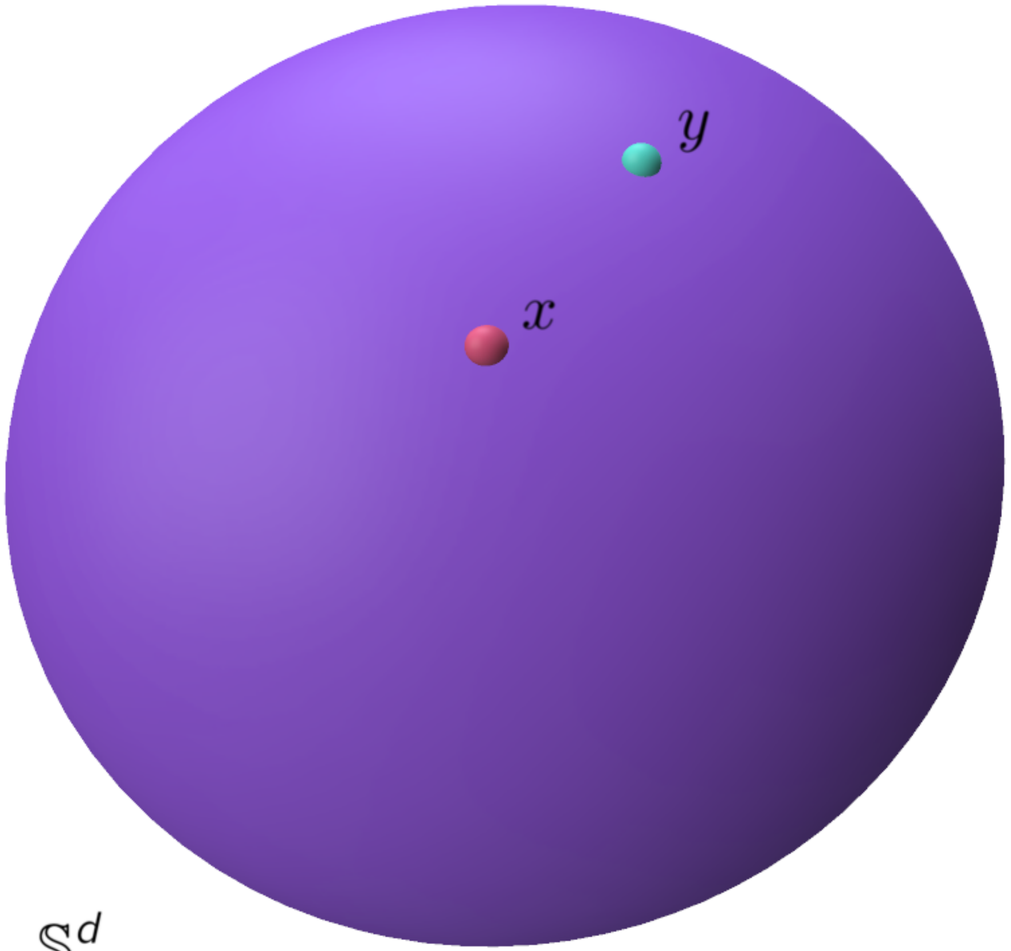
$$\nabla_{x_i} SW^\theta = \mathrm{Log}_{x_i}(T^\theta(x_i))$$



[Bonet et al 2023, Genest et al 2024]

NESOTS

$\mu = \delta_x, \nu = \delta_y$



Algorithm 1: Non Euclidean Sliced Optimal Transport Sampling – NESOTS

Data: The discrete target distribution $\nu = \sum_{i=1}^{m} \delta_{y_i}$, the number of iterations K, the batch size L, the gradient descent step γ

Result: The discrete distribution $\mu^{(K)}$ after K iterations.

$\mu^{(0)} = \text{SubSample}(\tilde{\nu}, n)$; // Init.

for $j \in [[1, K]]$ **do**

  parallel for $l \in [[1, L]]$ **do** // Batch

    $\tilde{\nu} = \text{SubSample}(\tilde{\nu}, n)$; // Sec. 3.6

    $\theta = \text{RandomSlice}()$; // Sec. 3.1

    $\tilde{\nu}_\theta = P^\theta\left(\tilde{\nu}^l\right)$; // Sec. 3.1

    $\mu_\theta = P^\theta\left(\mu^{(j)}\right)$; // Sec. 3.1

    $T = \text{Solve1DOT}(\mu_\theta, \tilde{\nu}_\theta)$; // Sec. 3.2

    for $i \in [[1, n]]$ **do**

      $\mathbf{g} = \Gamma_\theta\left(P^\theta\left(\mathbf{x}_i^{(j)}\right), T\left(P^\theta\left(\mathbf{x}_i^{(j)}\right)\right)\right)$; // Sec. 3.3

      $\mathbf{d}_i^l = \text{Log}_{\mathbf{x}_i^{(j)}}\left(\mathbf{g}\left(\mathbf{x}_i^{(j)}\right)\right)$; // Sec. 3.4

    end

  end

  parallel for $i \in [[1, n]]$ **do**

    $\mathbf{d}_i = \text{GeoMed}\left(\{\mathbf{d}_i^l\}_L\right)$; // Sec. 3.7

    $\mathbf{x}_i^{(j+1)} = \text{Exp}_{\mathbf{x}_i^{(j)}}\left(\gamma \mathbf{d}_i\right)$; // Sec. 3.5

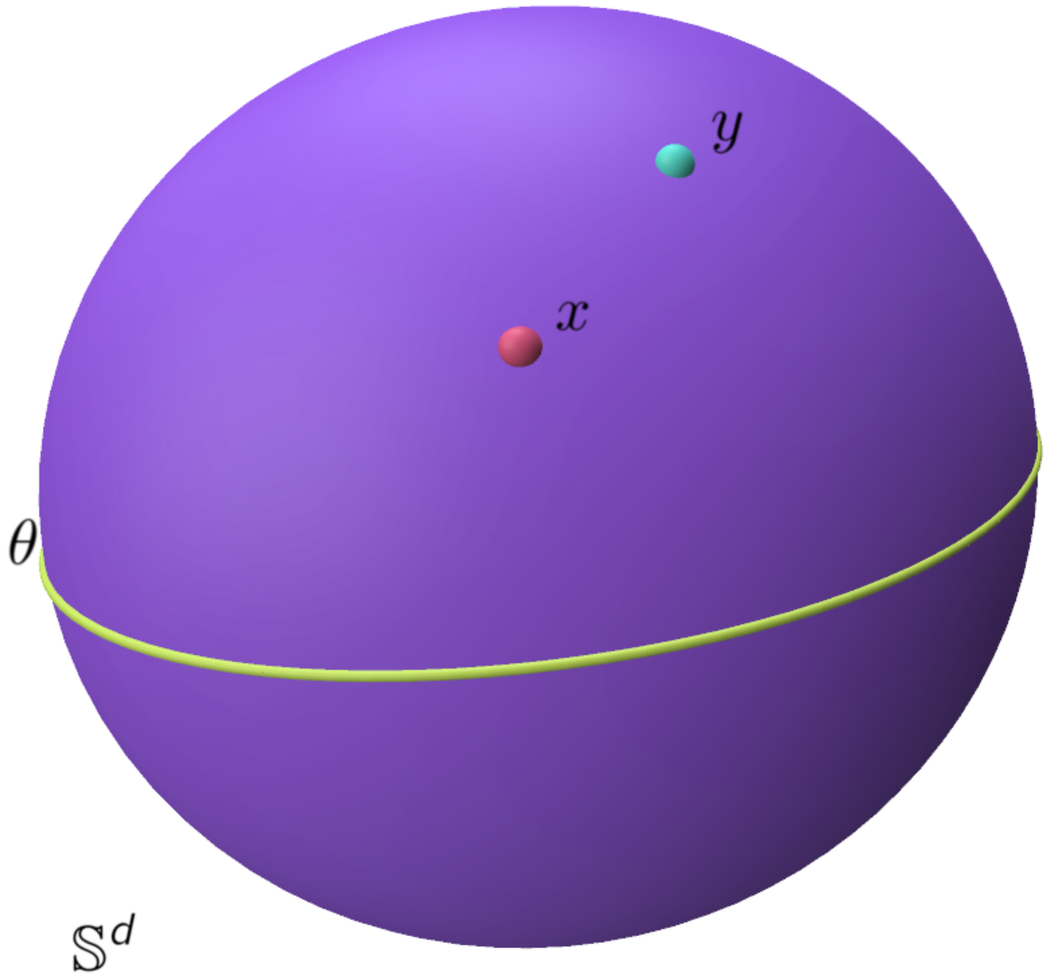
  end

end

return $\mu^{(K)} = \sum_{i=1}^{m} \delta_{\mathbf{x}_i^{(K)}}$

NESOTS

$$\mu = \delta_x, \nu = \delta_y$$



Algorithm 1: Non Euclidean Sliced Optimal Transport Sampling – NESOTS

Data: The discrete target distribution $\nu = \sum_{i=1}^{m} \delta_{y_i}$, the number of iterations K, the batch size L, the gradient descent step γ

Result: The discrete distribution $\mu^{(K)}$ after K iterations.

$\mu^{(0)} = \text{SubSample}(\tilde{\nu}, n)$; // Init.

for $j \in [[1, K]]$ **do**

  parallel for $l \in [[1, L]]$ **do** // Batch

    $\tilde{\nu} = \text{SubSample}(\tilde{\nu}, n)$; // Sec. 3.6

    $\theta = \text{RandomSlice}()$; // Sec. 3.1

    $\tilde{\nu}_\theta = P^\theta\left(\tilde{\nu}^l\right)$; // Sec. 3.1

    $\mu_\theta = P^\theta\left(\mu^{(j)}\right)$; // Sec. 3.1

    $T = \text{Solve1DOT}(\mu_\theta, \tilde{\nu}_\theta)$; // Sec. 3.2

    for $i \in [[1, n]]$ **do**

      $\mathbf{g} = \Gamma_\theta\left(P^\theta\left(\mathbf{x}_i^{(j)}\right), T\left(P^\theta\left(\mathbf{x}_i^{(j)}\right)\right)\right)$; // Sec. 3.3

      $\mathbf{d}_i^l = \text{Log}_{\mathbf{x}_i^{(j)}}\left(\mathbf{g}\left(\mathbf{x}_i^{(j)}\right)\right)$; // Sec. 3.4

    end

  end

  parallel for $i \in [[1, n]]$ **do**

    $\mathbf{d}_i = \text{GeoMed}\left(\{\mathbf{d}_i^l\}_L\right)$; // Sec. 3.7

    $\mathbf{x}_i^{(j+1)} = \text{Exp}_{\mathbf{x}_i^{(j)}}\left(\gamma \mathbf{d}_i\right)$; // Sec. 3.5

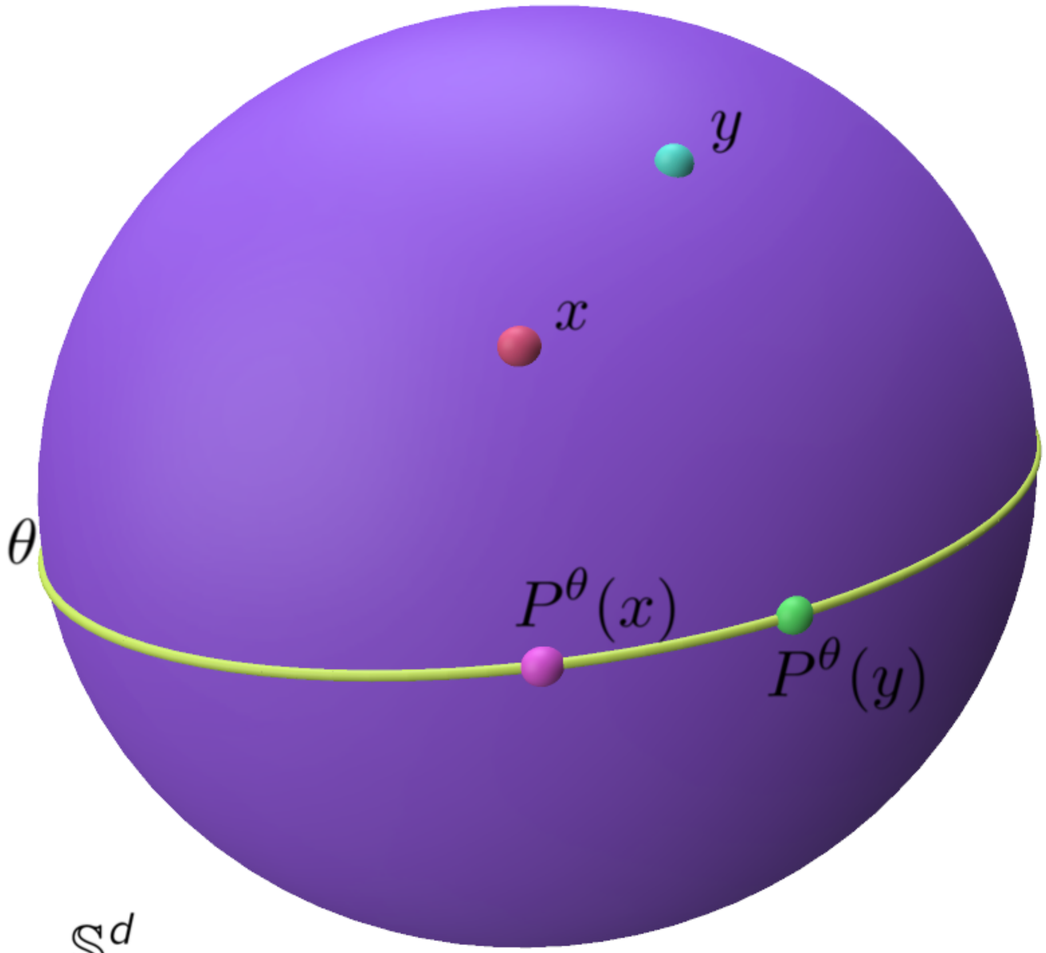
  end

end

return $\mu^{(K)} = \sum_{i=1}^{m} \delta_{\mathbf{x}_i^{(K)}}$

NESOTS

$\mu = \delta_x, \nu = \delta_y$



Algorithm 1: Non Euclidean Sliced Optimal Transport Sampling – NESOTS

Data: The discrete target distribution $\nu = \sum_{i=1}^{m} \delta_{y_i}$, the number of iterations K, the batch size L, the gradient descent step γ

Result: The discrete distribution $\mu^{(K)}$ after K iterations.

$\mu^{(0)} = \text{SubSample}(\tilde{\nu}, n)$; // Init.

for $j \in [[1, K]]$ **do**

  parallel for $l \in [[1, L]]$ **do** // Batch

    $\tilde{\nu} = \text{SubSample}(\tilde{\nu}, n)$; // Sec. 3.6

    $\theta = \text{RandomSlice}()$; // Sec. 3.1

    $\tilde{\nu}_\theta = P^\theta\left(\tilde{\nu}^l\right)$; // Sec. 3.1

    $\mu_\theta = P^\theta\left(\mu^{(j)}\right)$; // Sec. 3.1

    $T = \text{Solve1DOT}(\mu_\theta, \tilde{\nu}_\theta)$; // Sec. 3.2

    for $i \in [[1, n]]$ **do**

      $\mathbf{g} = \Gamma_\theta\left(P^\theta\left(\mathbf{x}_i^{(j)}\right), T\left(P^\theta\left(\mathbf{x}_i^{(j)}\right)\right)\right)$; // Sec. 3.3

      $\mathbf{d}_i^l = \text{Log}_{\mathbf{x}_i^{(j)}}\left(\mathbf{g}\left(\mathbf{x}_i^{(j)}\right)\right)$; // Sec. 3.4

    end

  end

  parallel for $i \in [[1, n]]$ **do**

    $\mathbf{d}_i = \text{GeoMed}\left(\{\mathbf{d}_i^l\}_L\right)$; // Sec. 3.7

    $\mathbf{x}_i^{(j+1)} = \text{Exp}_{\mathbf{x}_i^{(j)}}\left(\gamma \mathbf{d}_i\right)$; // Sec. 3.5

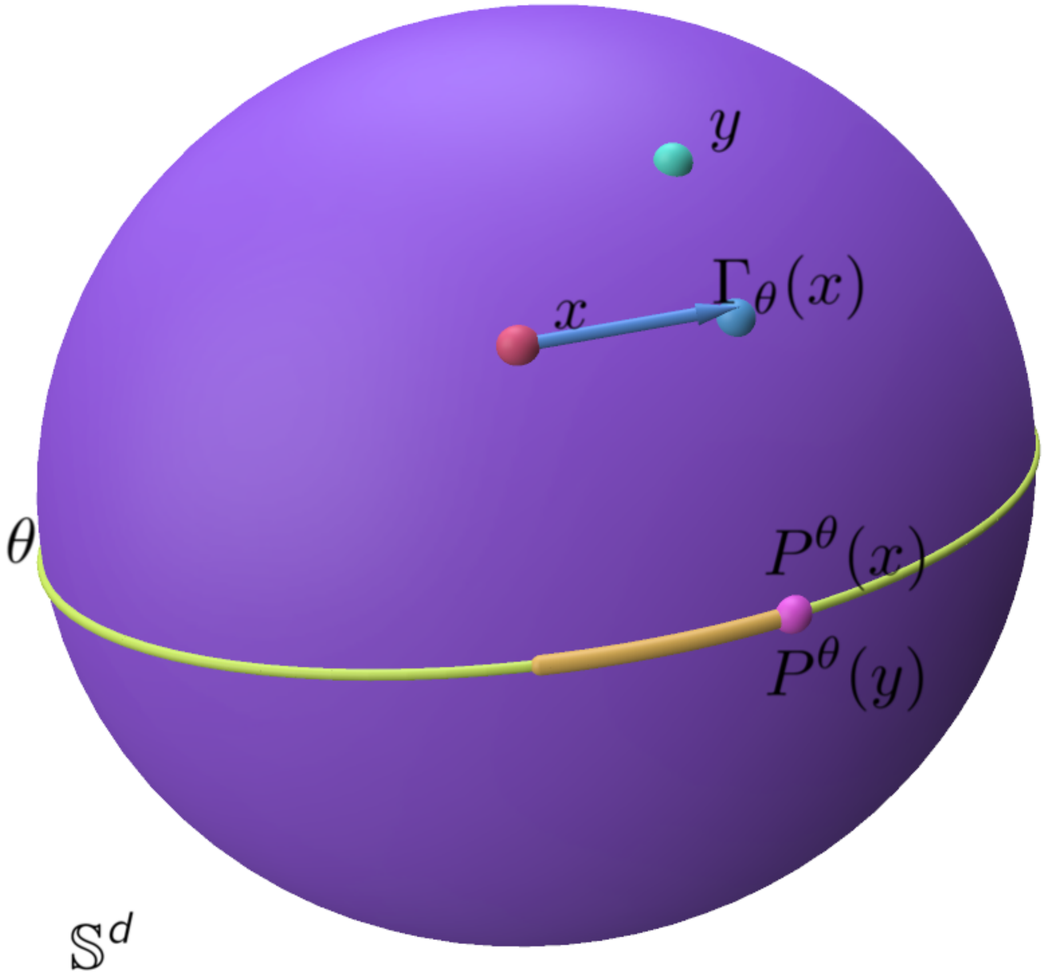
  end

end

return $\mu^{(K)} = \sum_{i=1}^{m} \delta_{\mathbf{x}_i^{(K)}}$

NESOTS

$$\mu = \delta_x, \nu = \delta_y$$



Algorithm 1: Non Euclidean Sliced Optimal Transport Sampling – NESOTS

Data: The discrete target distribution $\nu = \sum_{i=1}^{m} \delta_{y_i}$, the number of iterations K, the batch size L, the gradient descent step γ

Result: The discrete distribution $\mu^{(K)}$ after K iterations.

$\mu^{(0)} = \text{SubSample}(\tilde{\nu}, n)$; // Init.

for $j \in [[1, K]]$ **do**

  parallel for $l \in [[1, L]]$ **do** // Batch

    $\tilde{\nu} = \text{SubSample}(\tilde{\nu}, n)$; // Sec. 3.6

    $\theta = \text{RandomSlice}()$; // Sec. 3.1

    $\tilde{\nu}_\theta = P^\theta\left(\tilde{\nu}^l\right)$; // Sec. 3.1

    $\mu_\theta = P^\theta\left(\mu^{(j)}\right)$; // Sec. 3.1

    $T = \text{Solve1DOT}(\mu_\theta, \tilde{\nu}_\theta)$; // Sec. 3.2

    for $i \in [[1, n]]$ **do**

      $\mathbf{g} = \Gamma_\theta\left(P^\theta\left(\mathbf{x}_i^{(j)}\right), T\left(P^\theta\left(\mathbf{x}_i^{(j)}\right)\right)\right)$; // Sec. 3.3

      $\mathbf{d}_i^l = \text{Log}_{\mathbf{x}_i^{(j)}}\left(\mathbf{g}\left(\mathbf{x}_i^{(j)}\right)\right)$; // Sec. 3.4

    end

  end

  parallel for $i \in [[1, n]]$ **do**

    $\mathbf{d}_i = \text{GeoMed}\left(\{\mathbf{d}_i^l\}_L\right)$; // Sec. 3.7

    $\mathbf{x}_i^{(j+1)} = \text{Exp}_{\mathbf{x}_i^{(j)}}\left(\gamma \mathbf{d}_i\right)$; // Sec. 3.5

  end

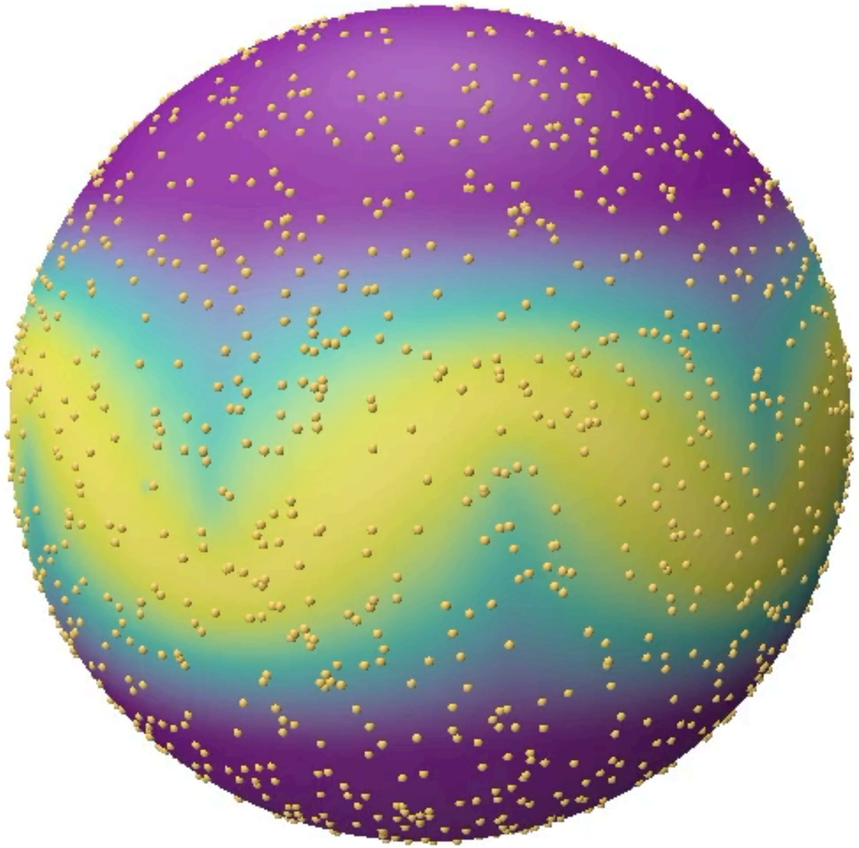
end

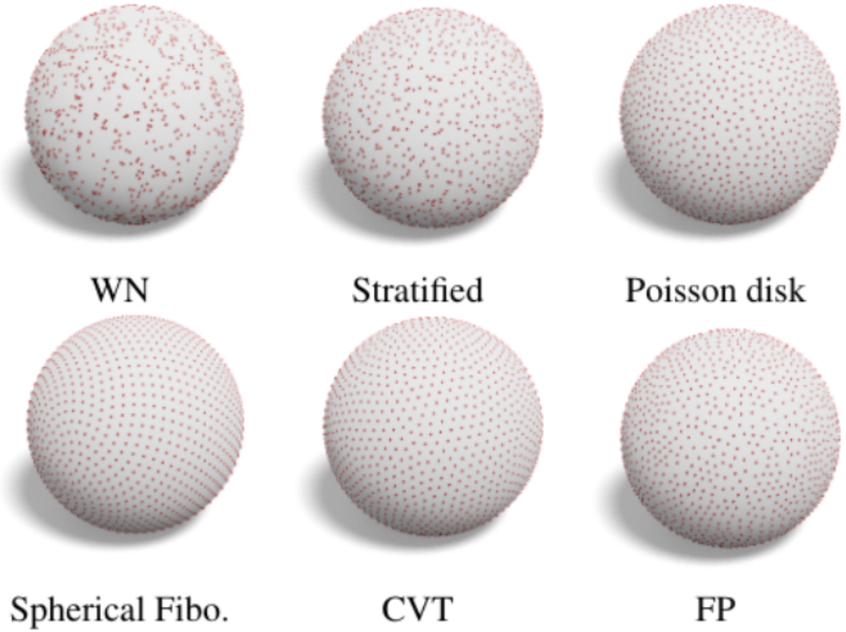
return $\mu^{(K)} = \sum_{i=1}^{m} \delta_{\mathbf{x}_i^{(K)}}$

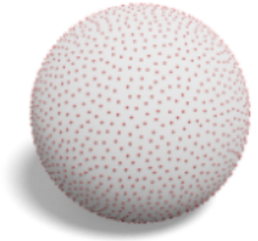
NESOTS

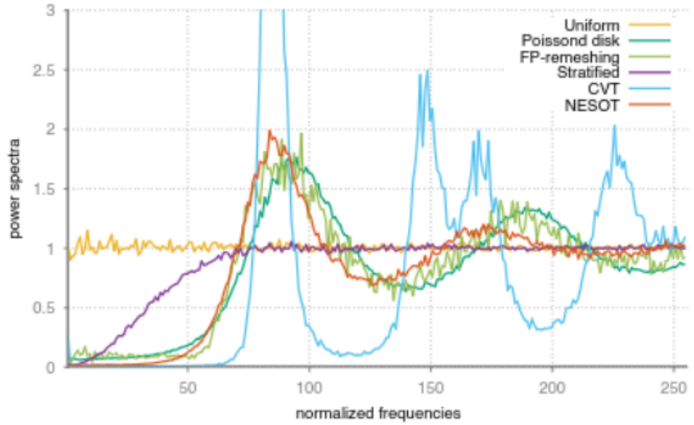
	$\mathbb{R}^d$	$\mathbb{S}^d$	$\mathbb{H}^d$ (Lorentz Model)
$\mathbf{Exp}_x(v)$	$x + v$	$\cos(\|\|v\|\|)x + \sin(\|\|v\|\|)\frac{v}{\|\|v\|\|}$	$\cosh(\|\|v\|\|_L)x + \cosh(\|\|v\|\|_L)\frac{v}{\|\|v\|\|}$
$\mathbf{Log}_x(y)$	$y - x$	$\frac{\Pi_{TM_x}(y-x)}{\|\|\Pi_{TM_x}(y-x)\|\|} d(x, y)$	$\frac{\operatorname{arccosh}(-\langle x,y\rangle_L)}{\sqrt{\langle x,y\rangle_L^2-1}}(y + \langle x, y\rangle_L x)$
$\mathbf{P}^\theta(x)$	$\theta\langle\theta, x\rangle$	$\frac{\Pi^\theta(x)}{\|\|\Pi^\theta(x)\|\|}$	$\frac{\Pi^\theta(x)}{\sqrt{-\langle\Pi^\theta(x),\Pi^\theta(x)\rangle_L}}$

Results in $\mathbb{S}^d$

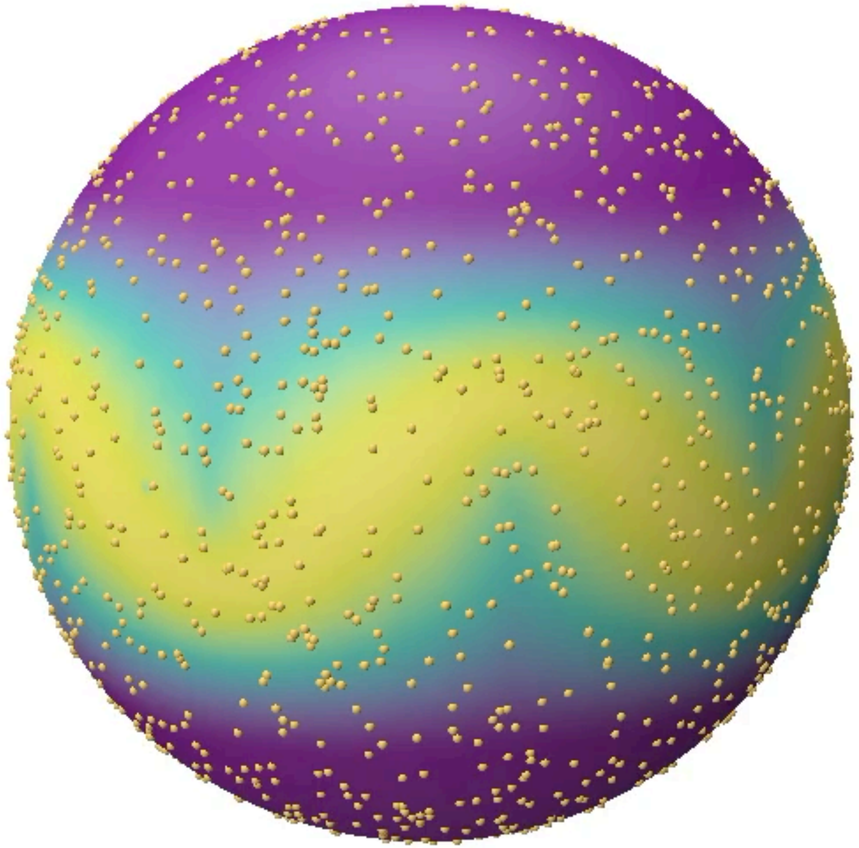


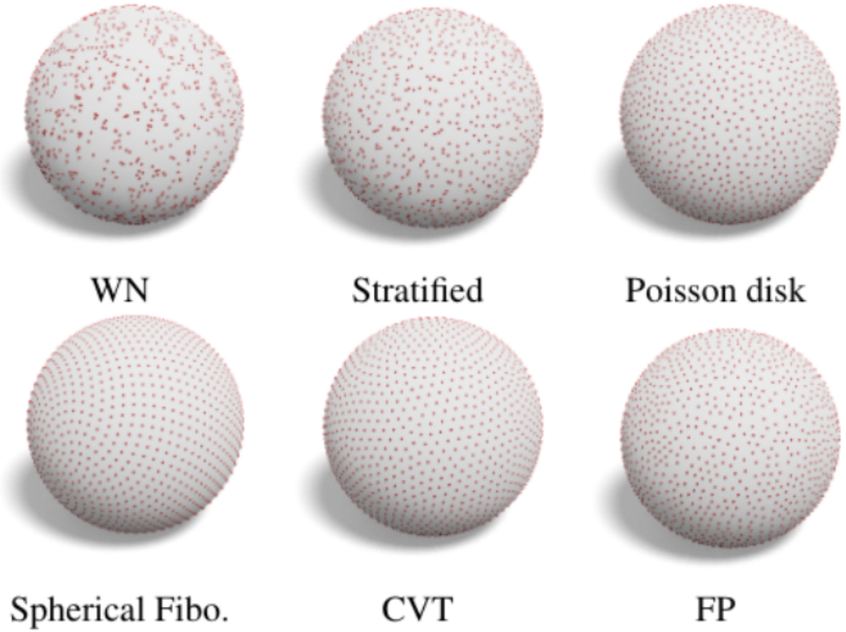


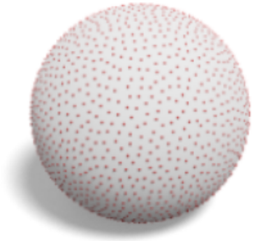


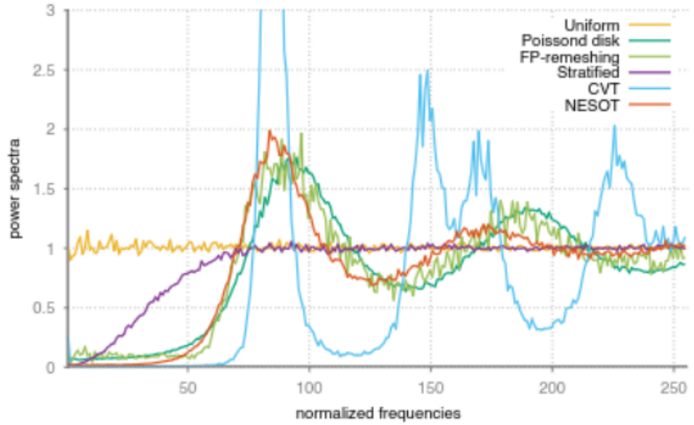


Results in $\mathbb{S}^d$



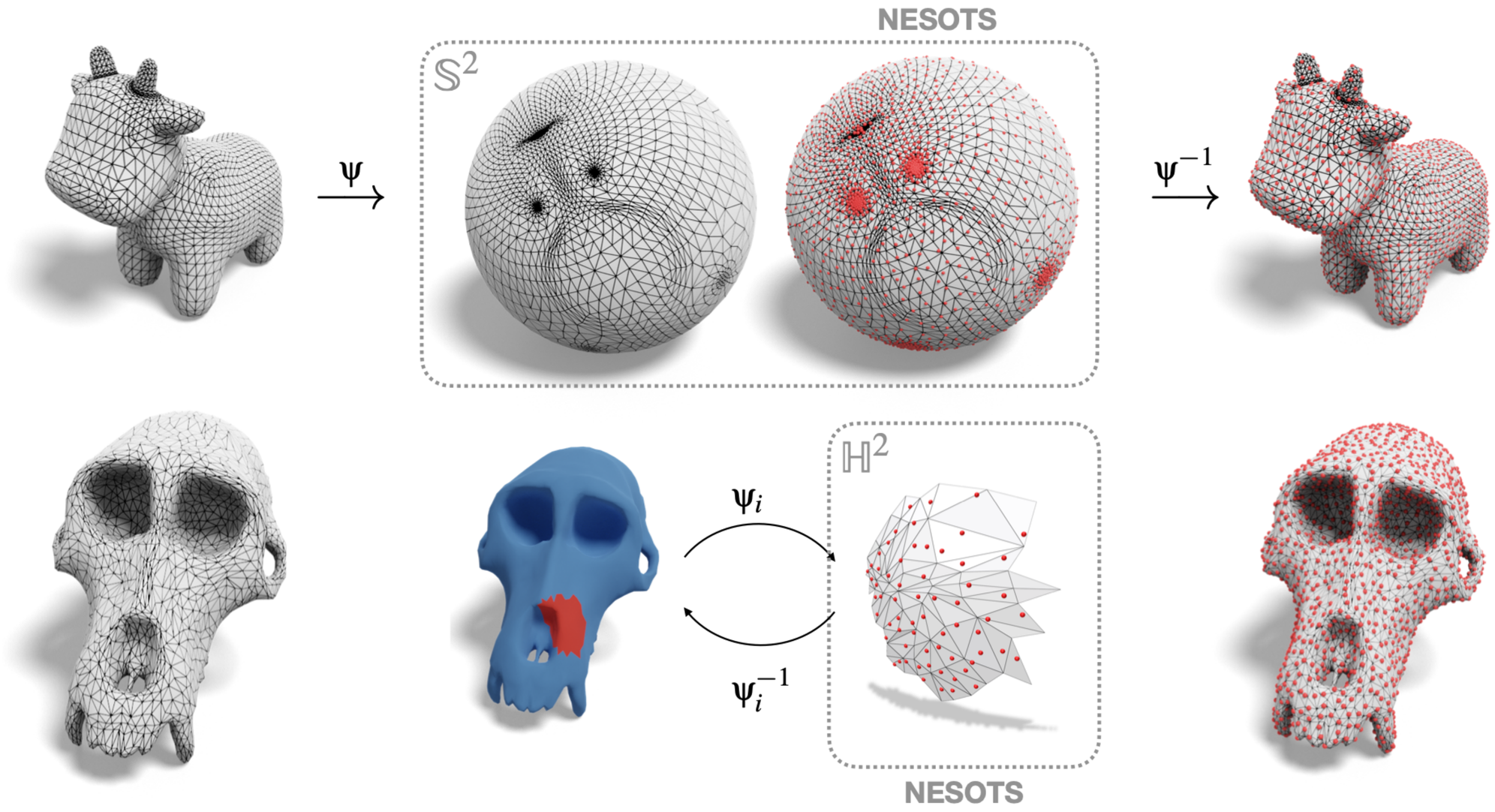




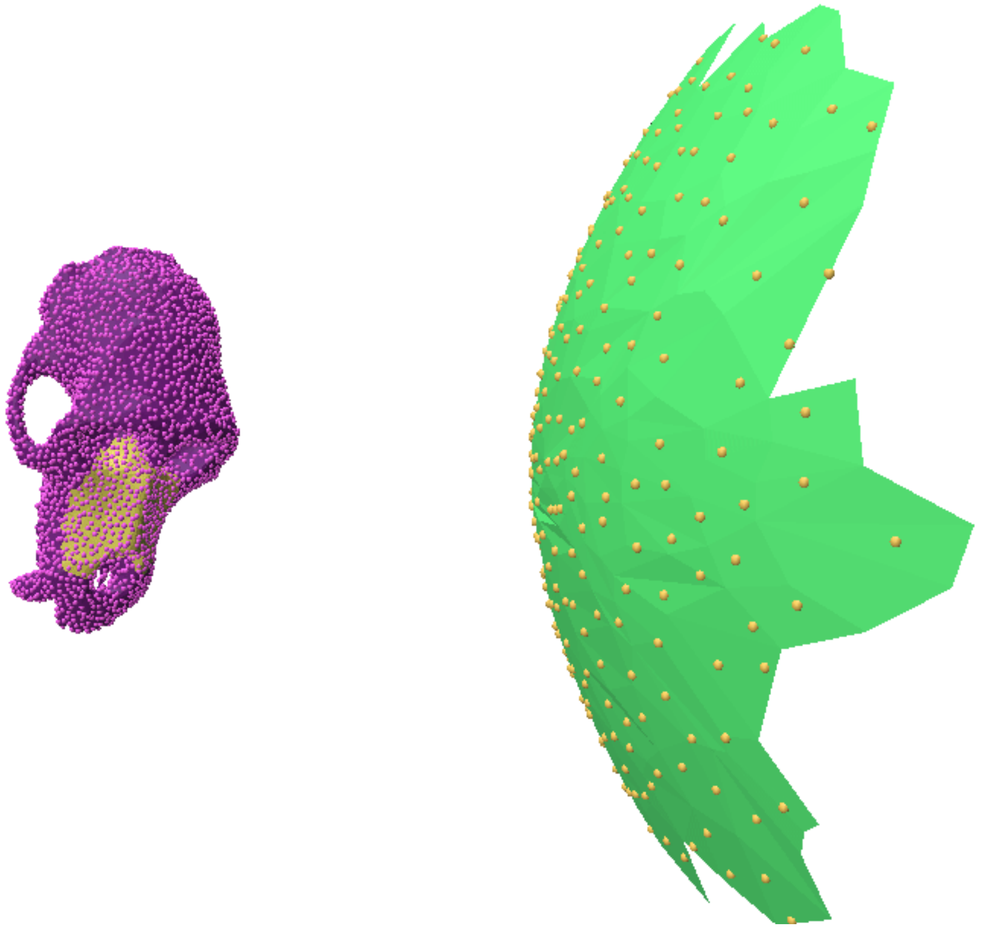


BN sampling of meshes

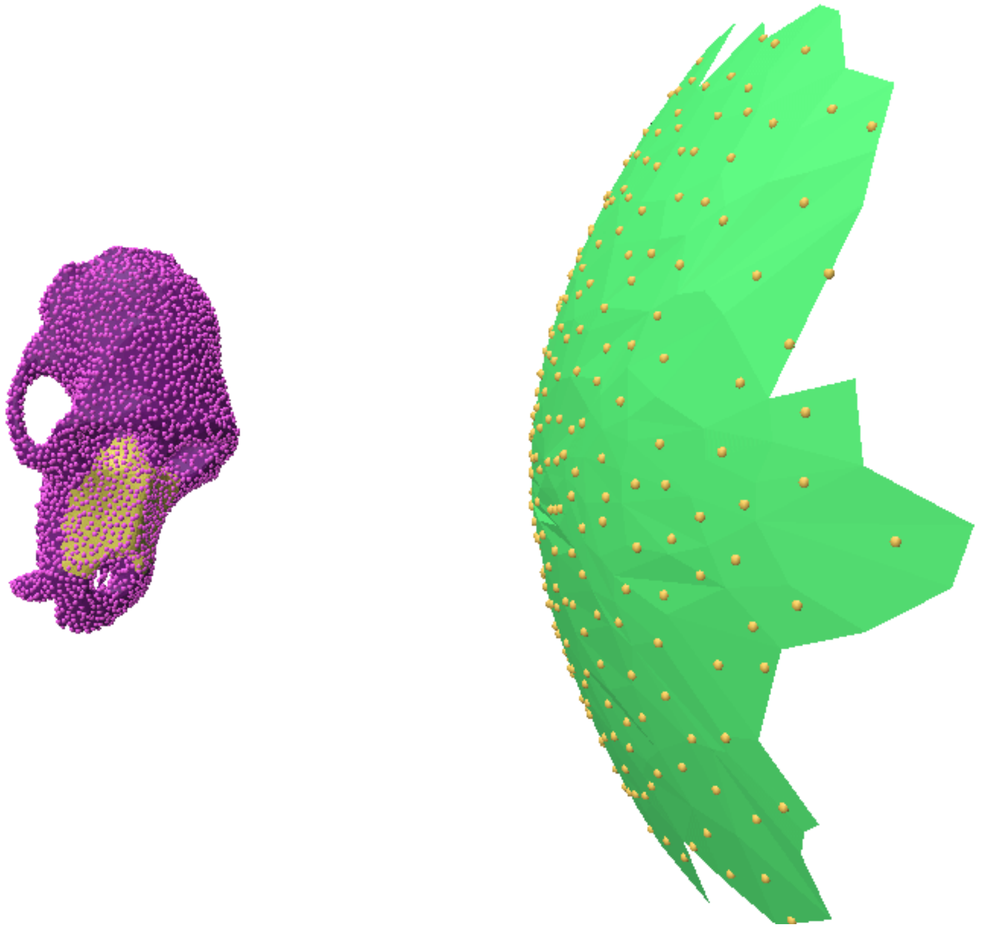
Conformal maps and uniformization theorem

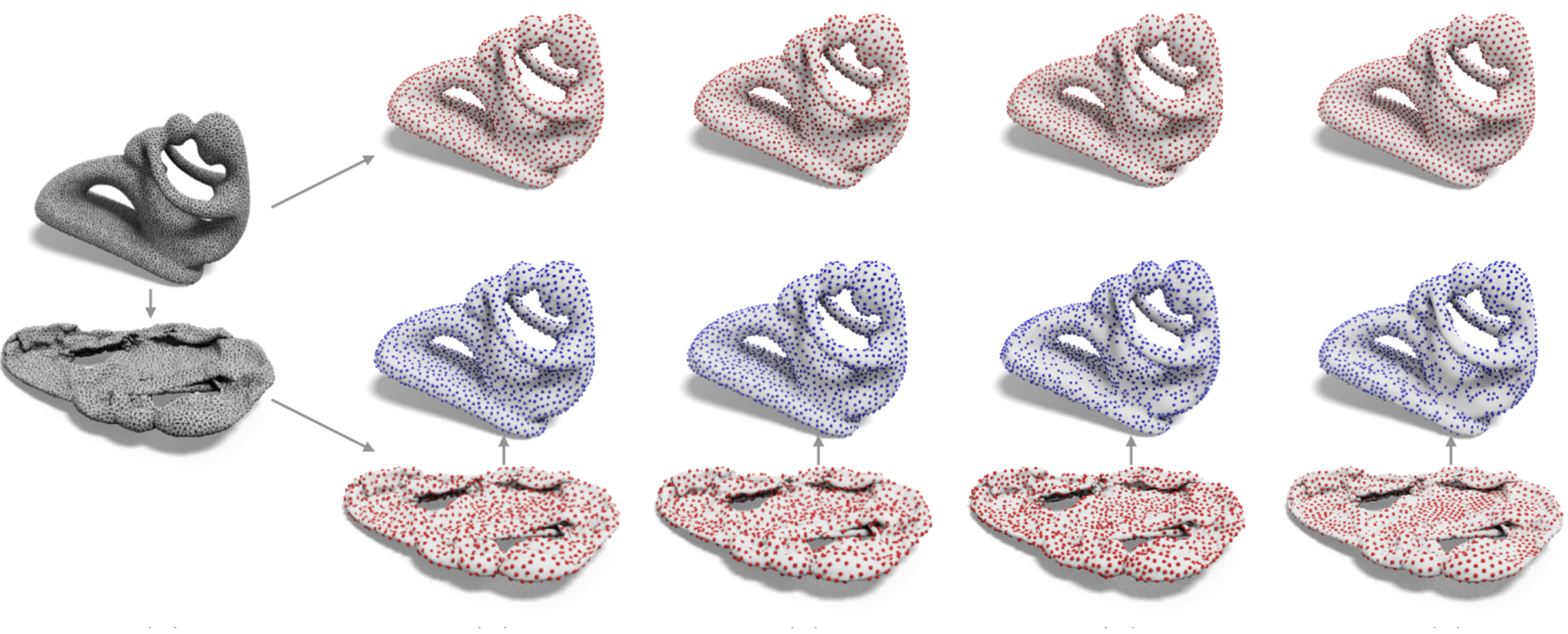


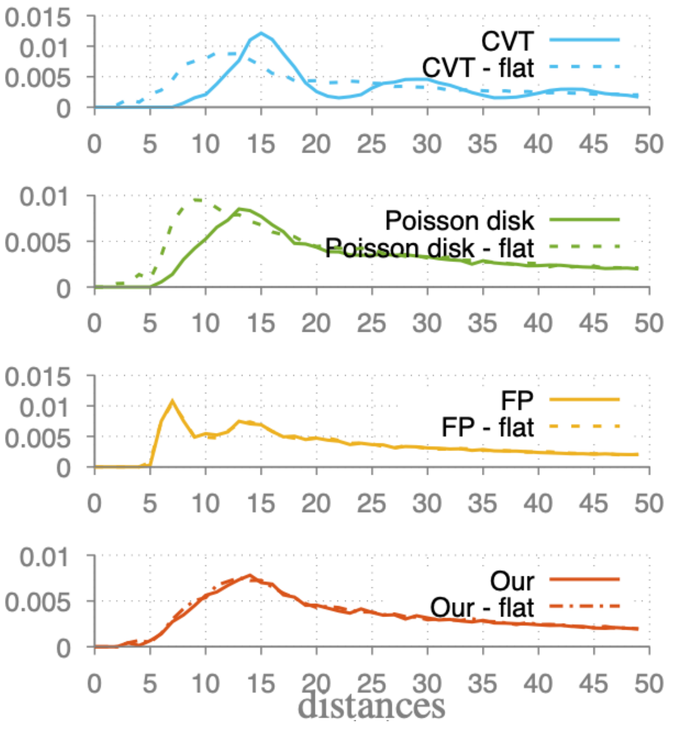
Local optimization on $\mathbb{H}^d$

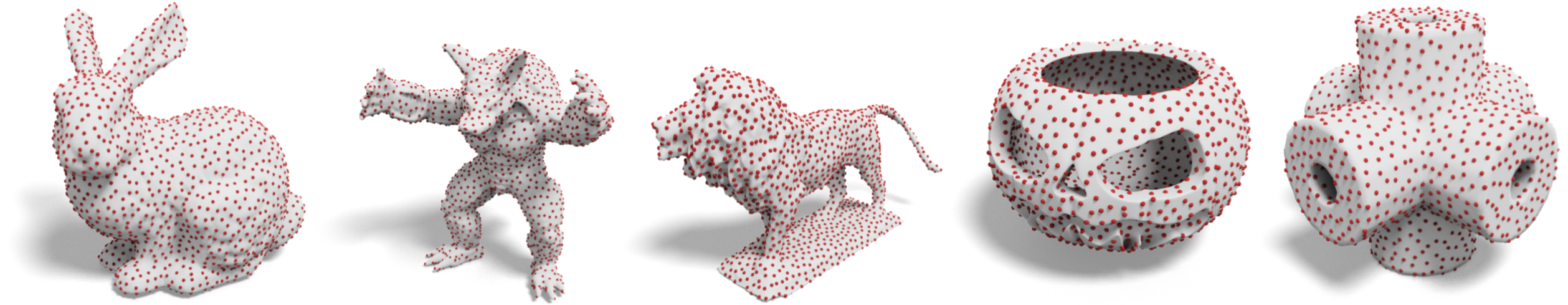


Local optimization on $\mathbb{H}^d$

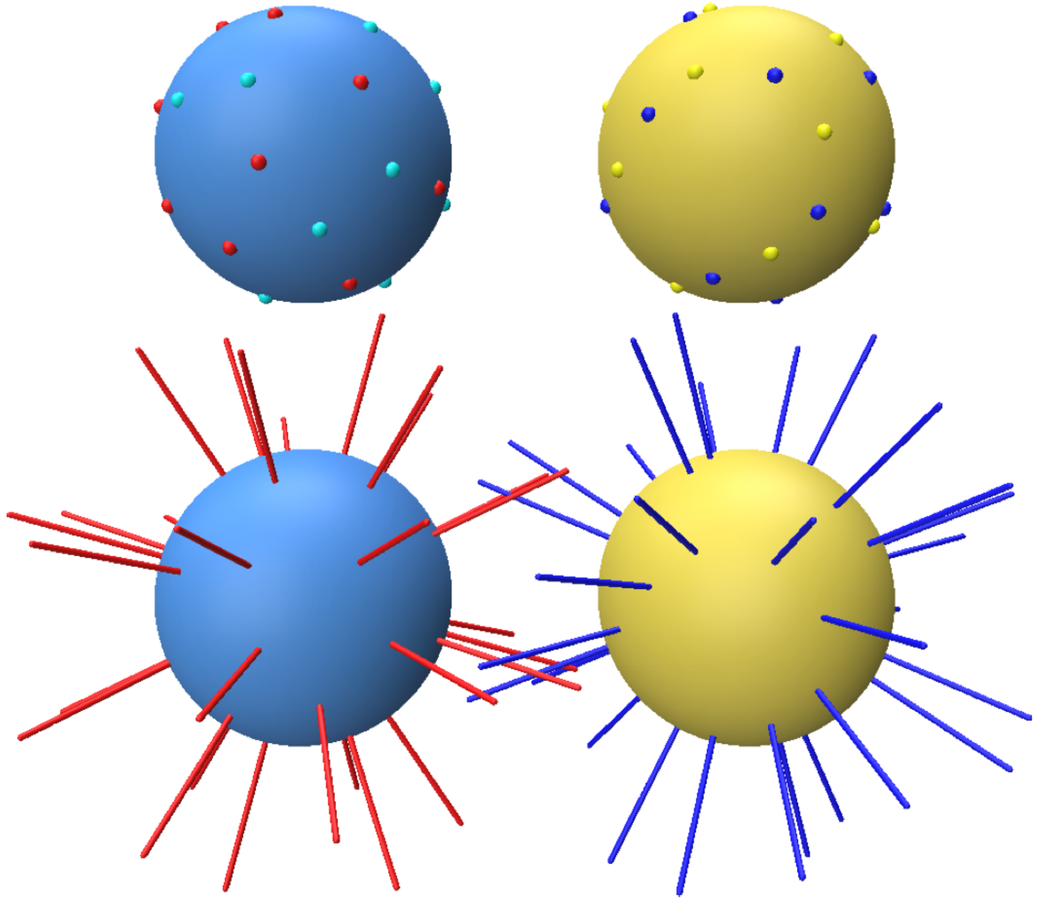




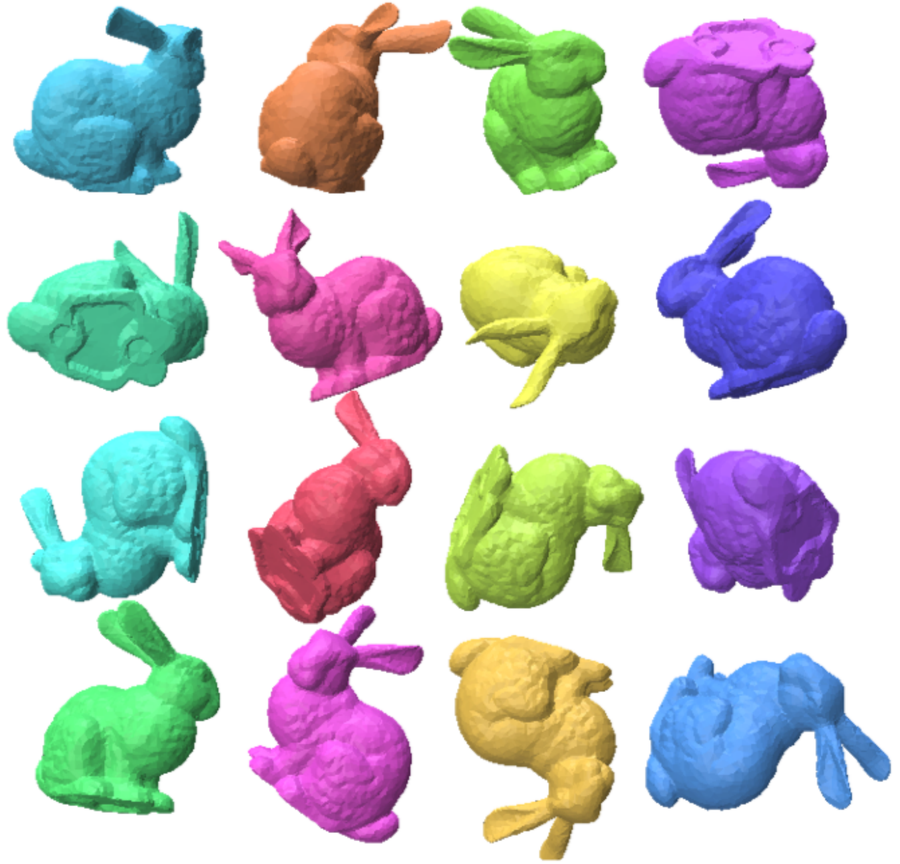

0.015
0.01
0.005
0
CVT
CVT - flat
0 5 10 15 20 25 30 35 40 45 50
0.01
0.005
0
Poisson disk
Poisson disk - flat
0 5 10 15 20 25 30 35 40 45 50
0.015
0.01
0.005
0
FP
FP - flat
0 5 10 15 20 25 30 35 40 45 50
0.01
0.005
0
Our
Our - flat
0 5 10 15 20 25 30 35 40 45 50
distances



Sampling projective domains



Sampling $\mathbb{P}^2$



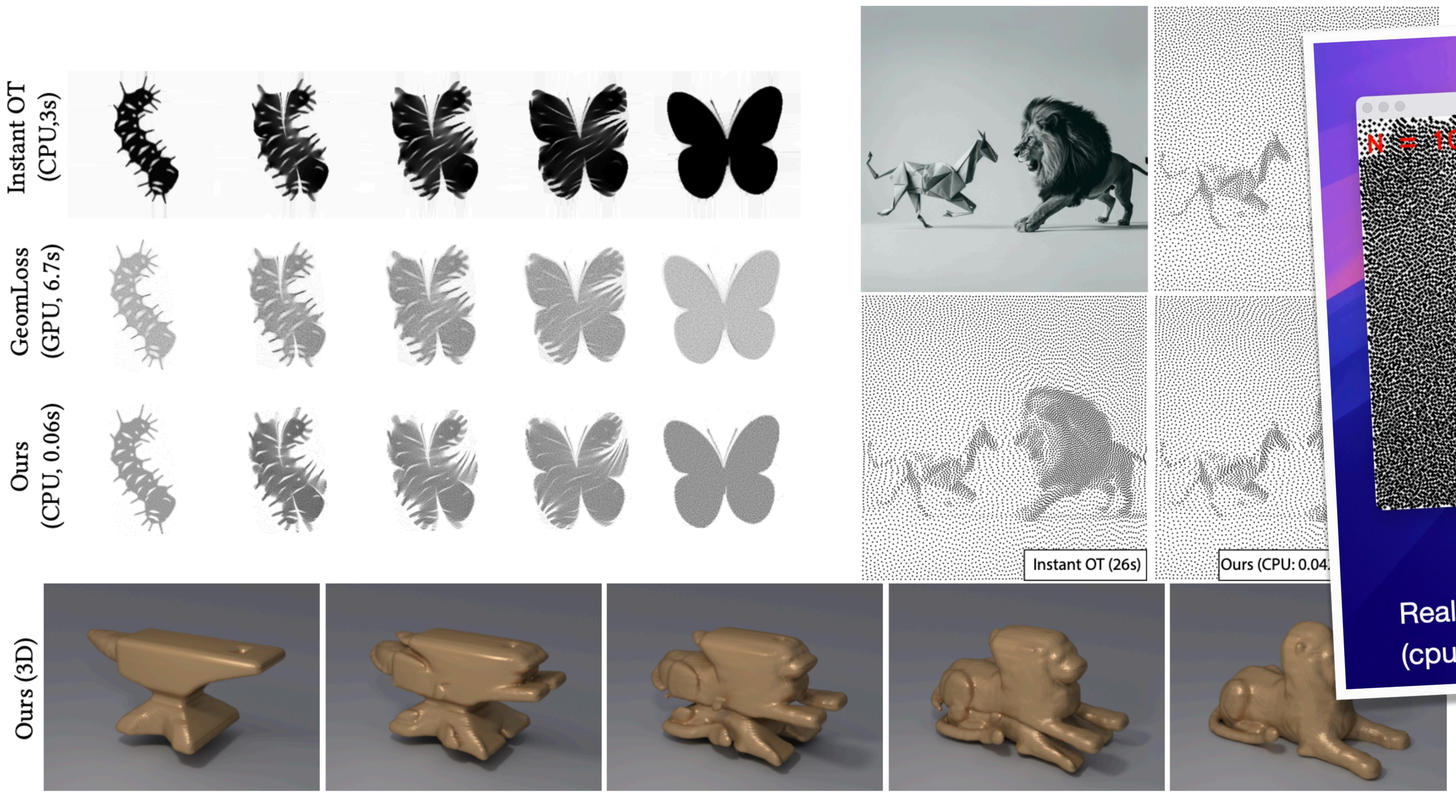
Sampling of unit quaternion ~ $\mathbb{P}^3$

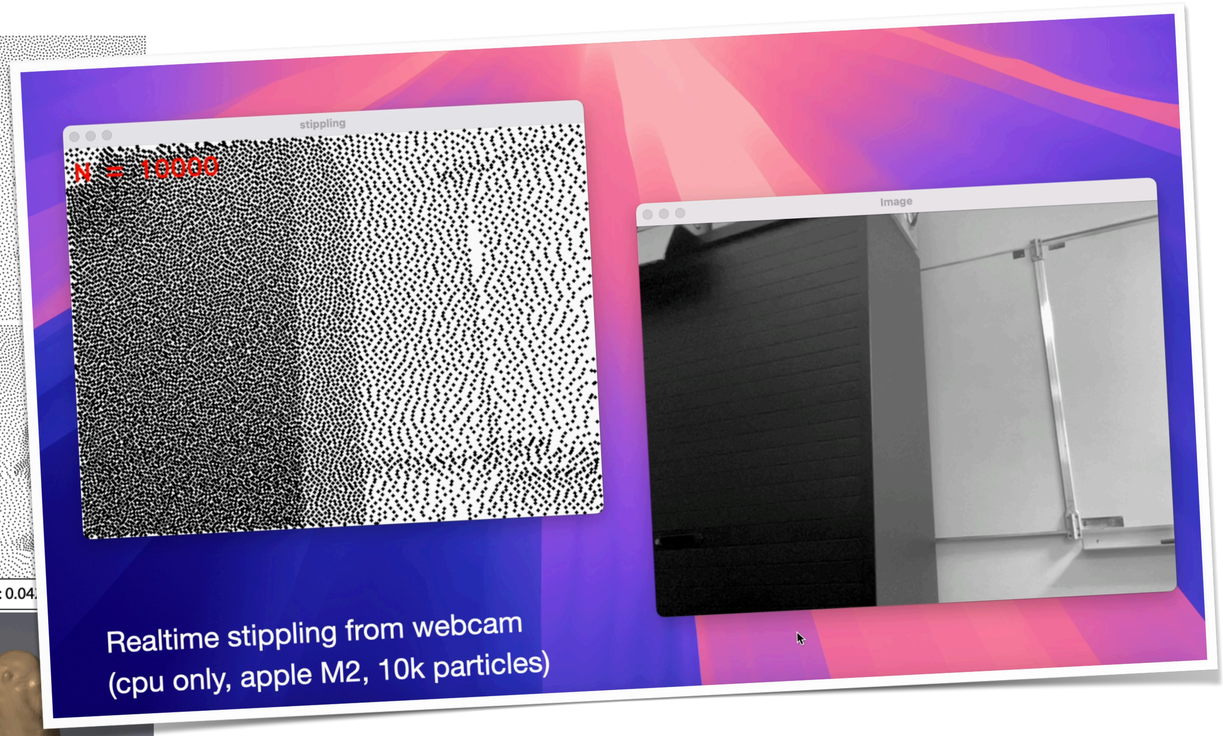
Conclusion

- Generic sliced optimal transport sampling $\mathbb{R}^d$, $\mathbb{S}^d$, $\mathbb{H}^d$, $\mathbb{P}^d$
- Intrinsic sampling of meshes
- Easy to implement

Extras

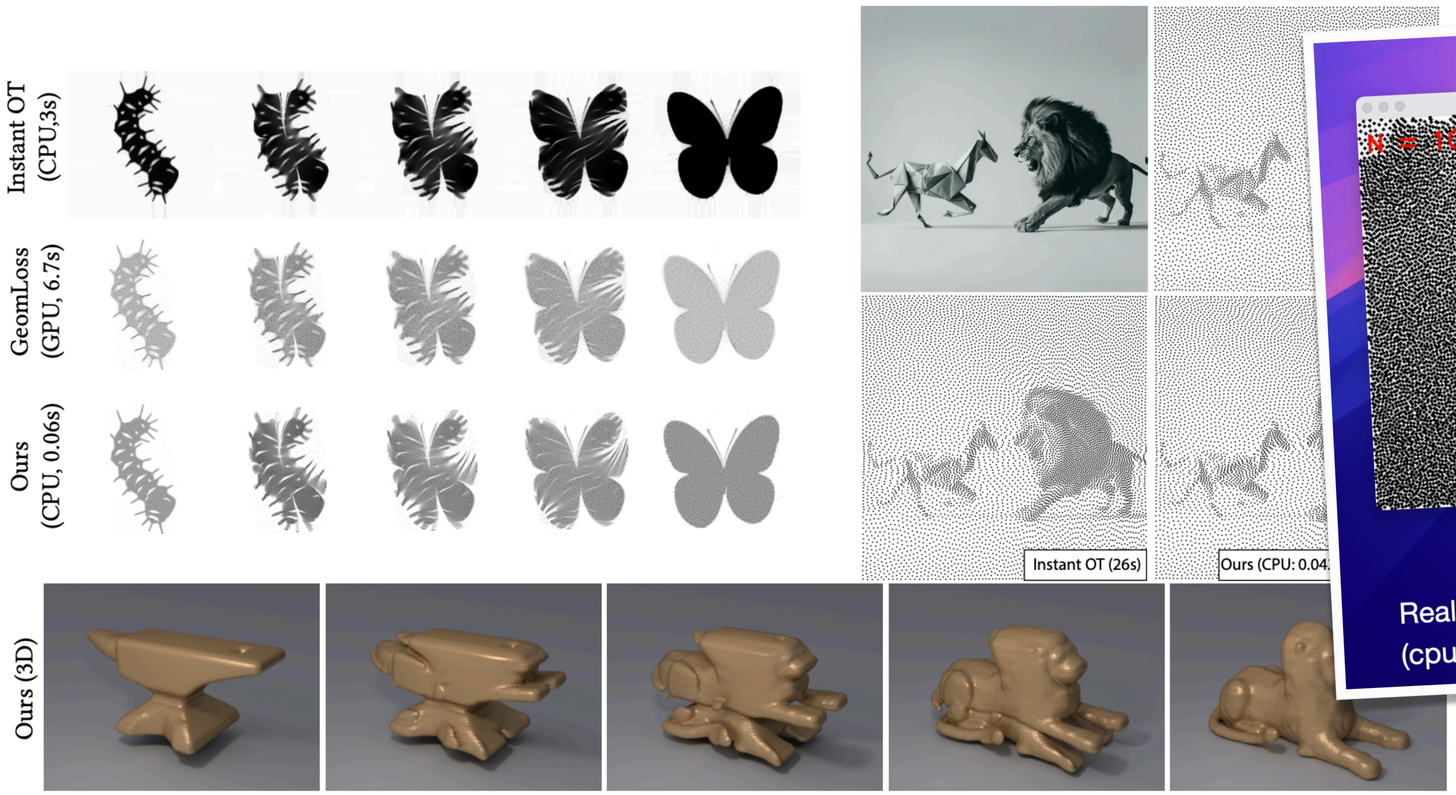
Rectifield Flows

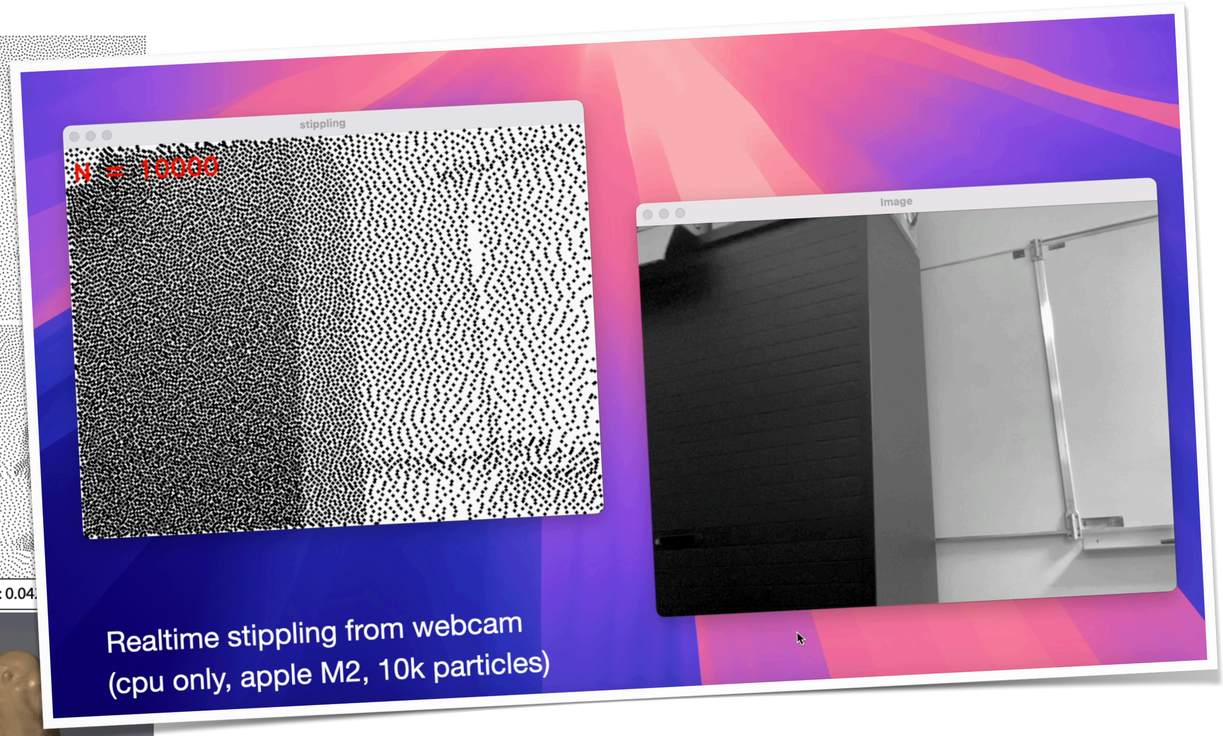




[Khoa Do, David Coeurjolly, Pooran Memari, Nicolas Bonneel
ACM Transactions on Graphics (Proceedings of SIGGRAPH), August, 2025]

Rectifield Flows

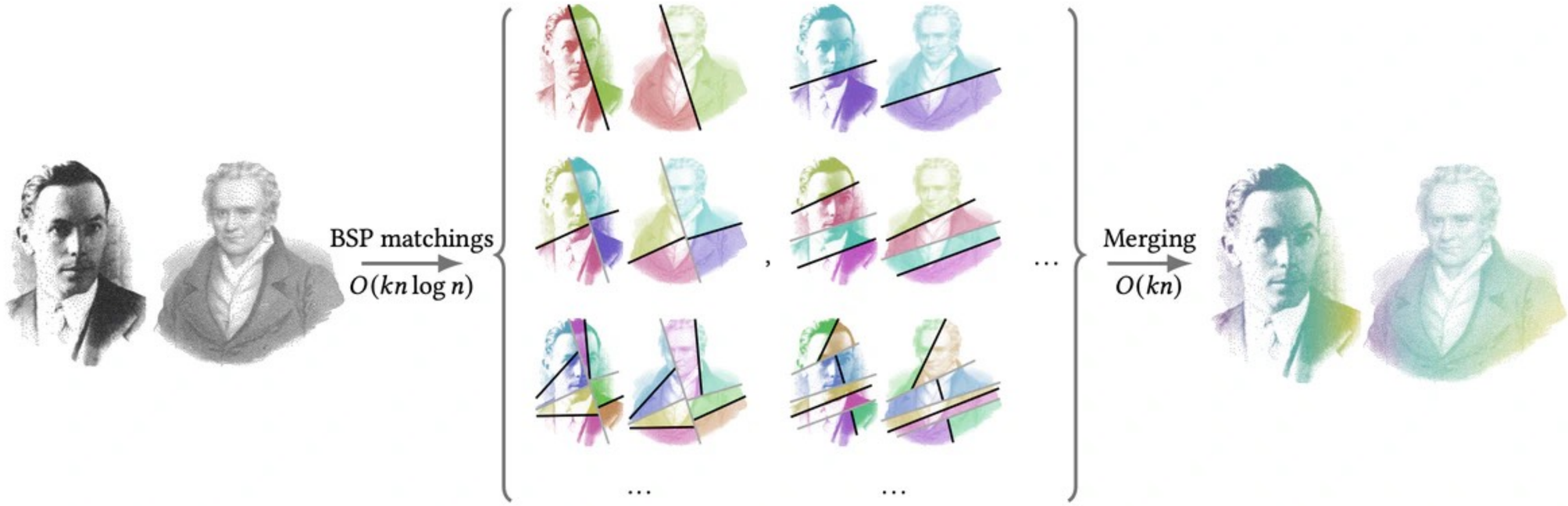




[Khoa Do, David Coeurjolly, Pooran Memari, Nicolas Bonneel
ACM Transactions on Graphics (Proceedings of SIGGRAPH), August, 2025]

BSP-OT

[Baptiste Genest, Nicolas Bonneel, Vincent Nivoliers, David Coeurjolly
ACM Transactions on Graphics (Proceedings of SIGGRAPH ASIA), December, 2025]




Finally!
2.5M pixels
RGB space
L2 metric
9s/bijection


Finally!
2.5M pixels
RGB space
L2 metric
9s/bijection

Conclusion

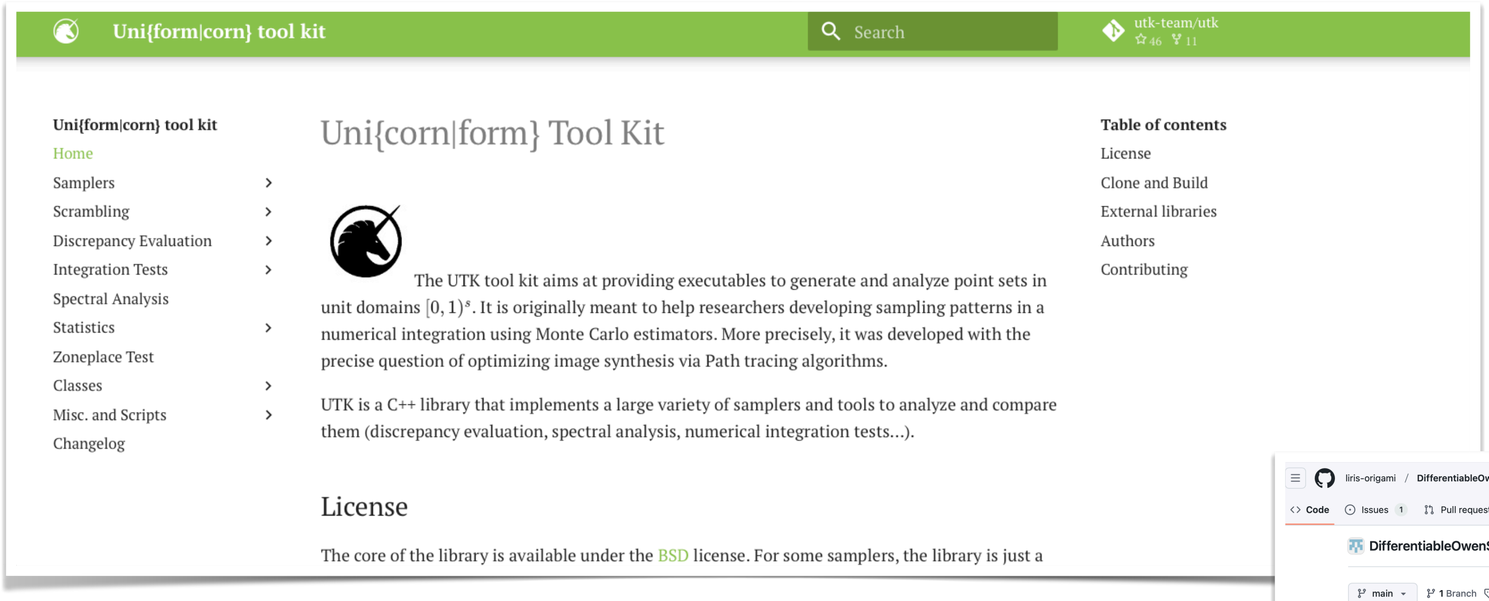
Correlated Samplers in Computer Graphics

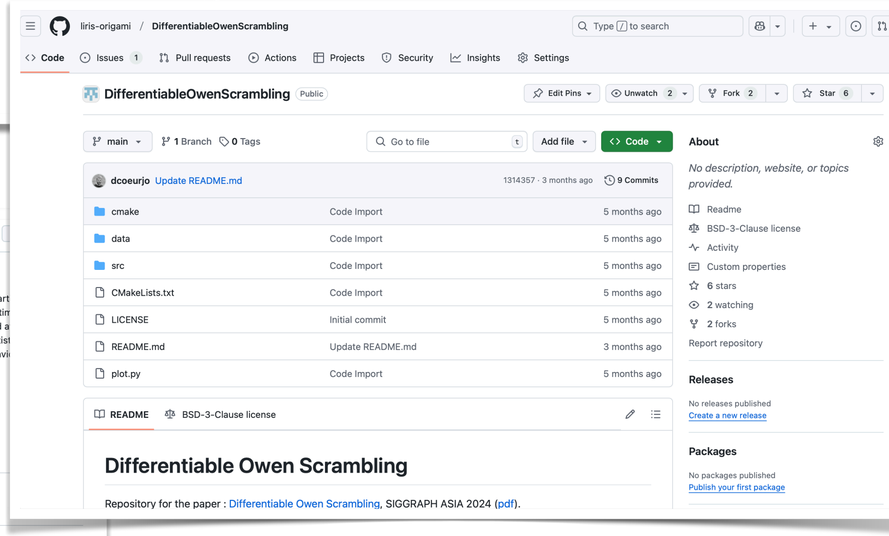
- **Point sampling**
  - **Critical step in many applications**: image rendering, MC integration, signal reconstruction, sampling latent spaces in machine learning, solving PDE using MC techniques…
  - **Many related fields:** arithmetic, applied math, signal processing, optimization…
  - **Practical constraints:** variance reduction in MC rendering, high dimension, fast samplers, controllable subspaces, adaptive

- **Future works:**
  - Still room between (projective) optimized LDS and blue noise properties (for low sample counts)
  - Many possible improvements in MC Rendering
    - By focusing on special classes of integrands
    - By carefully integrating with advanced techniques (canonical vs. actual integration domain, —resampled— importance sampling, splitting, control variates, screen space diffusion...)

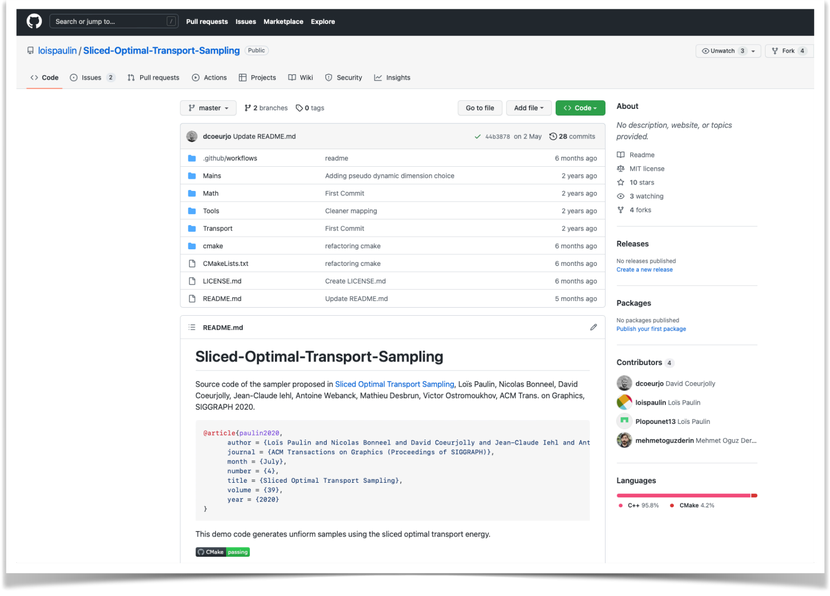
Code

https://utk-team.github.io/utk/

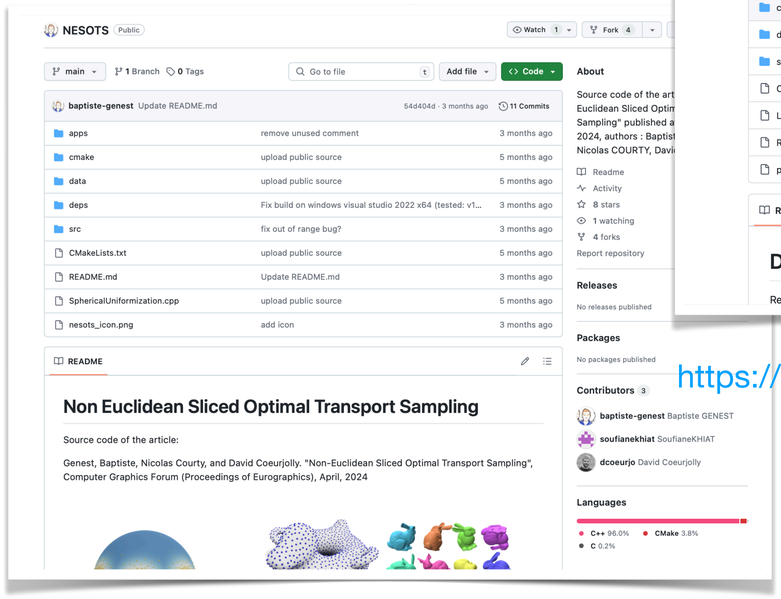




https://github.com/liris-origami/DifferentiableOwenScrambling



https://github.com/loispaulin/Sliced-Optimal-Transport-Sampling



https://github.com/baptiste-genest/NESOTS

Thanks

Khoa Do, Bastien Doignies, Baptiste Genest, Loïs Paulin

Nicolas Bonneel, Nicolas Courty, Mathieu Desbrun, Jean-Claude Iehl,
Alex Keller, Pooran Memari, Vincent Nivoliers, Victor Ostromoukhov