Monte Carlo integration

15-468, 15-668, 15-868
Physics-based Rendering
Spring 2025, Lecture 8

http://graphics.cs.cmu.edu/courses/15-468

Course announcements

- Programming assignment 2 will be posted on Friday.

Overview of today's lecture

- Leftover from BRDFs.
- Monte Carlo integration.
- Sampling techniques.
- Importance sampling.
- Ambient occlusion.

Slide credits

Most of these slides were directly adapted from:

- Wojciech Jarosz (Dartmouth).

Numerical Integration - Motivation

For very, *very* simple integrals, we can compute the solution analytically

$$\int_0^1 \frac{1}{3}x^2 \, \mathrm{d}x = \Big[x^3 \Big]_0^1 = 1$$

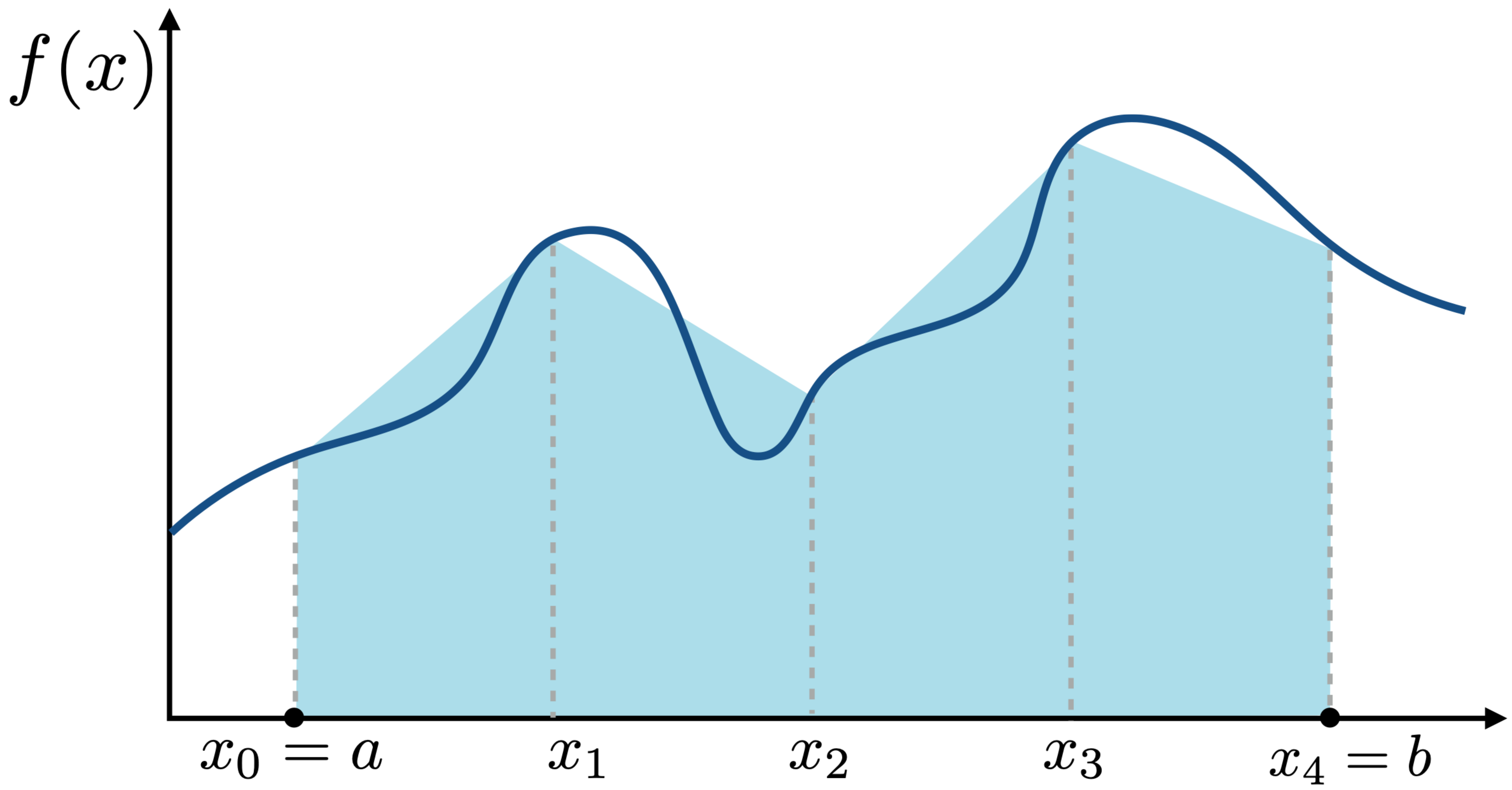
But ours are a bit more complicated:

$$L_r(\mathbf{x}, \vec{\omega}_r) = \int_{H^2} f_r(\mathbf{x}, \vec{\omega}_i, \vec{\omega}_r) L_i(\mathbf{x}, \vec{\omega}_i) \cos\theta_i \, \mathrm{d}\vec{\omega}_i$$

Typical quadrature: Trapezoid rule

Approximate integral of $f(x)$ by assuming function is piecewise linear

For equal length segments: $h = \frac{b-a}{n-1}$

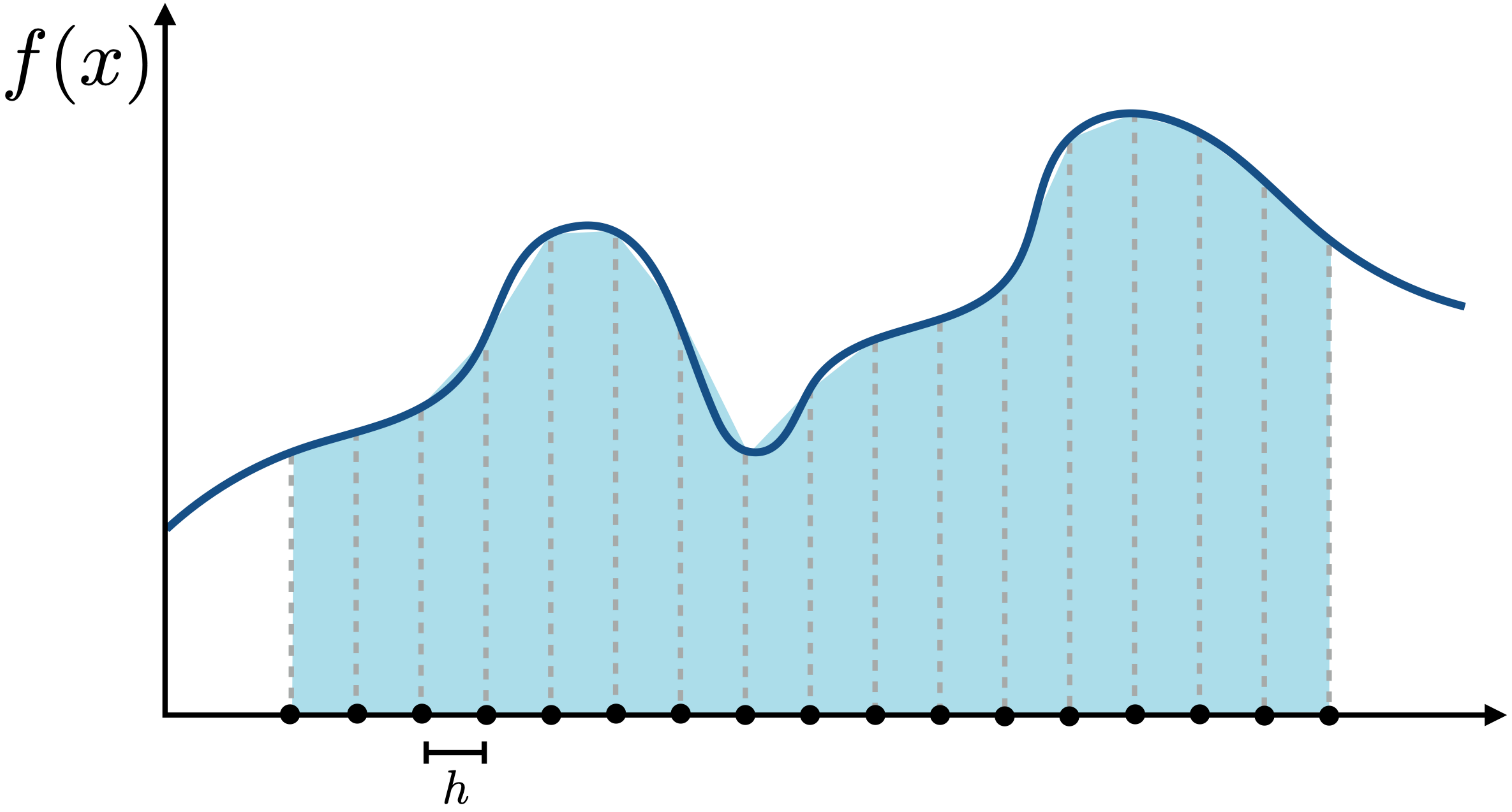


Typical quadrature: Trapezoid rule

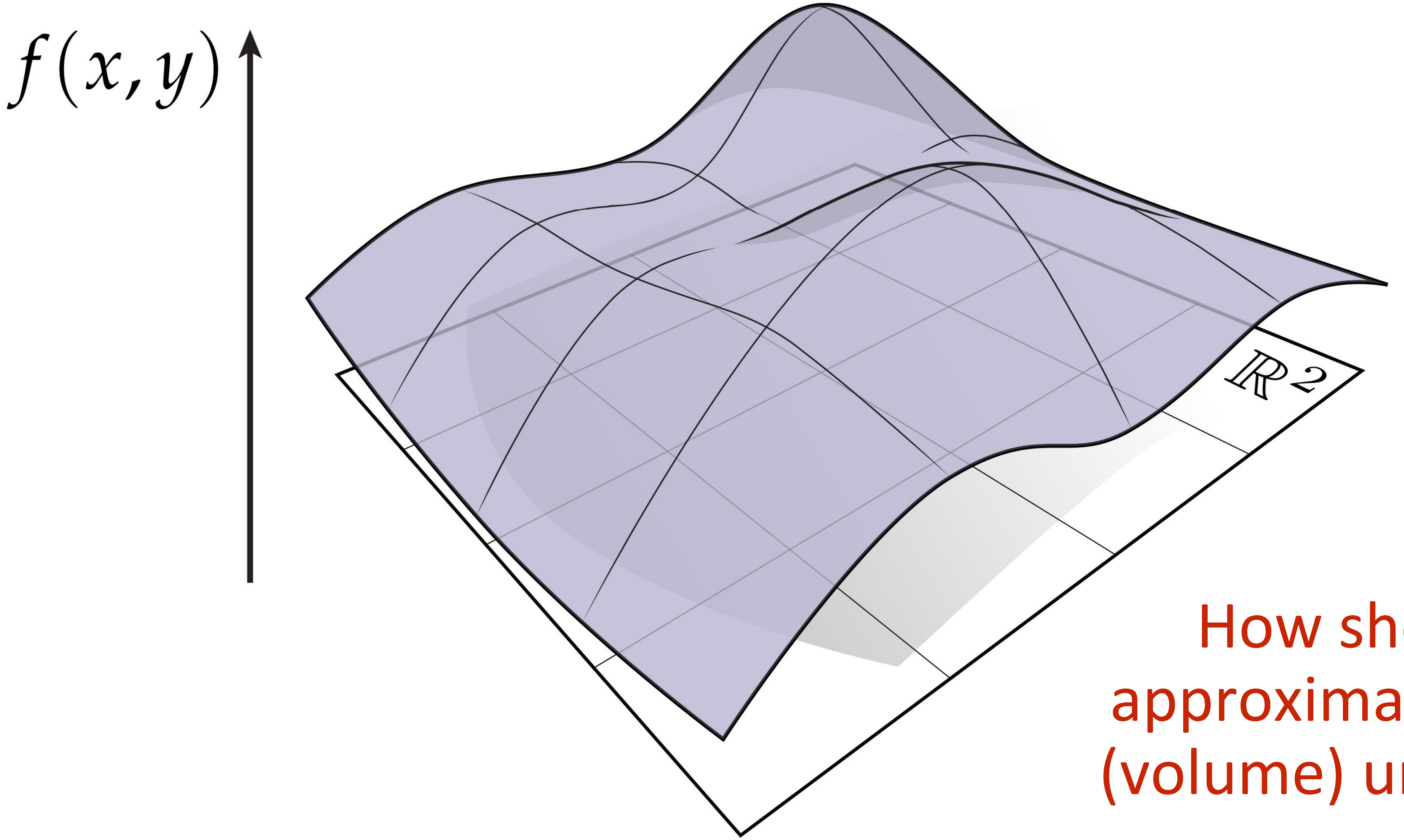
Consider cost and accuracy as $n \to \infty$ (or $h \to 0$)

Work: $O(n)$

Error can be shown to be: $$O(h^2) = O\left(\frac{1}{n^2}\right)$$ (for f(x) with continuous second derivative)



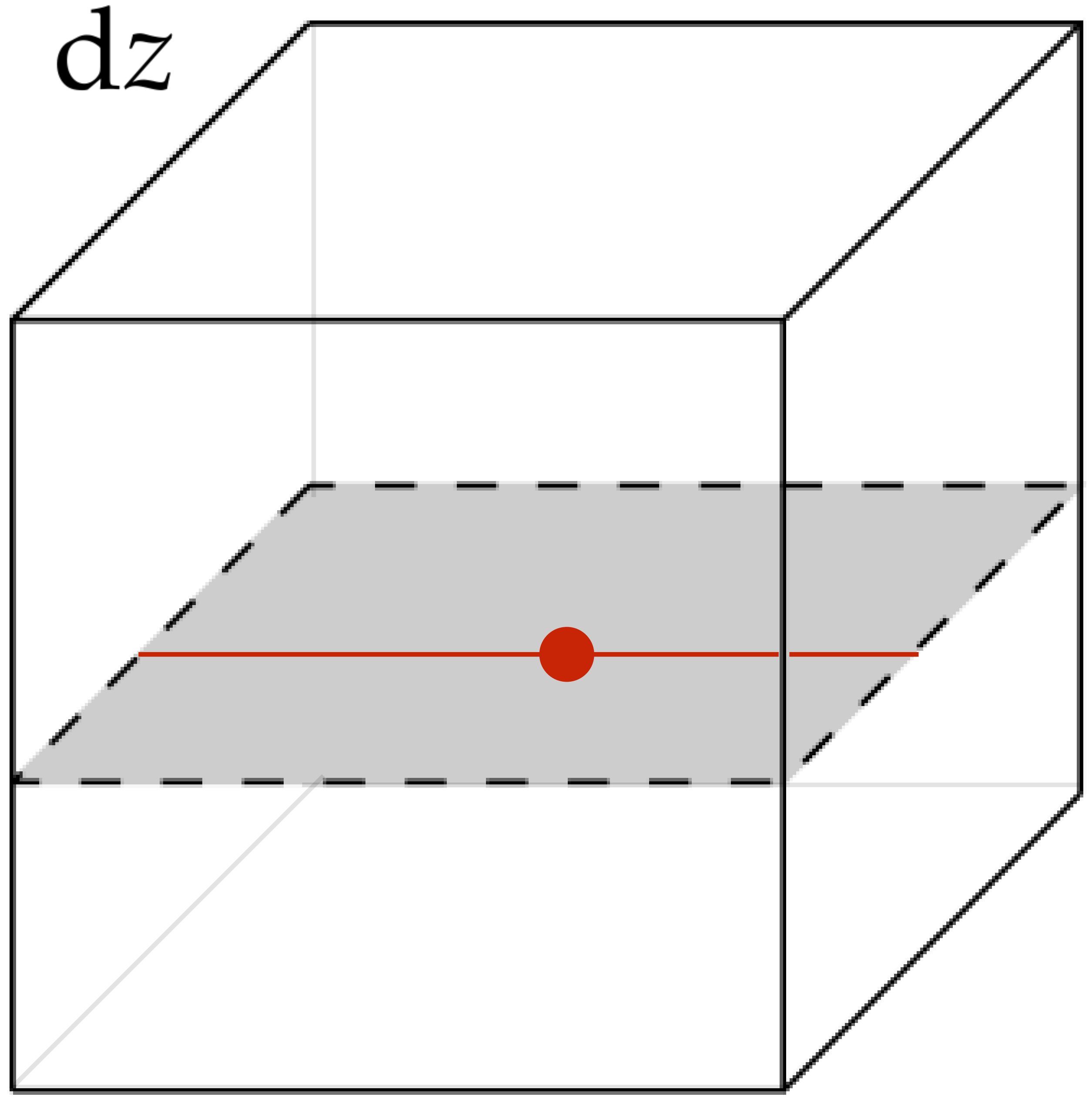
What about a 2D function?



How should we approximate the area (volume) underneath?

Multidimensional integrals & Fubini's theorem

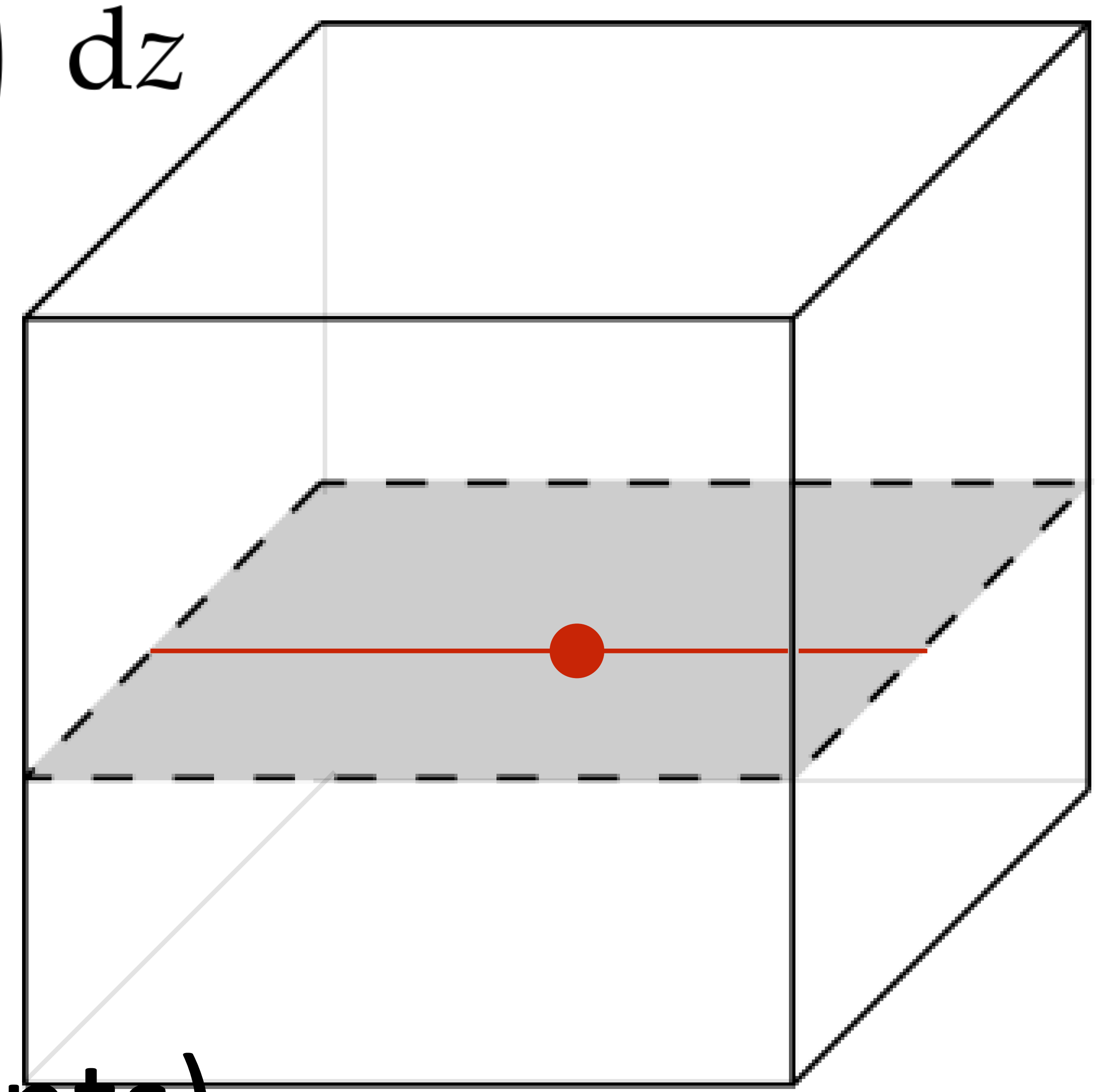
$$\int_{X\times Y\times Z} f(x,y,z)\mathrm{d}(\mathrm{x},\mathrm{y},\mathrm{z}) = \int_X \left(\int_Y \left(\int_Z f(x,y,z)\,\mathrm{d}x\right)\,\mathrm{d}y\right)\,\mathrm{d}z$$



Apply the trapezoid rule repeatedly

Multidimensional integrals & Fubini's theorem

$$\int_{X\times Y\times Z} f(x,y,z)\mathrm{d}(\mathrm{x},\mathrm{y},\mathrm{z}) = \int_X \left(\int_Y \left(\int_Z f(x,y,z)\,\mathrm{d}x\right)\,\mathrm{d}y\right)\,\mathrm{d}z$$



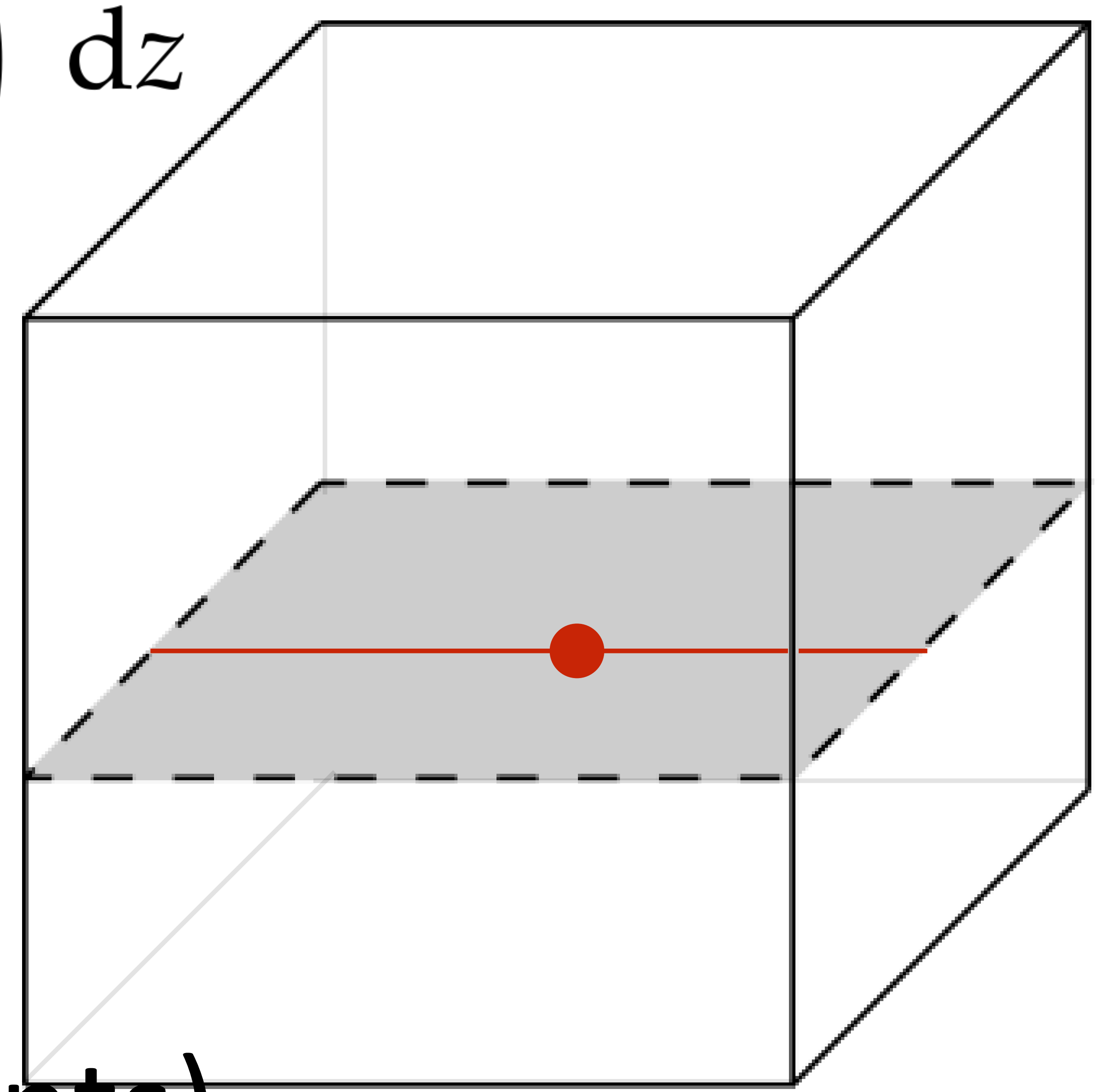
Apply the trapezoid rule repeatedly

Can show that:

- Errors add, so error still: $O(h^2)$
- But work is now: $O(n^2)$ (n x n set of measurements)

Multidimensional integrals & Fubini's theorem

$$\int_{X\times Y\times Z} f(x,y,z)\mathrm{d}(\mathrm{x},\mathrm{y},\mathrm{z}) = \int_X \left(\int_Y \left(\int_Z f(x,y,z)\,\mathrm{d}x\right)\,\mathrm{d}y\right)\,\mathrm{d}z$$



Apply the trapezoid rule repeatedly

Can show that:

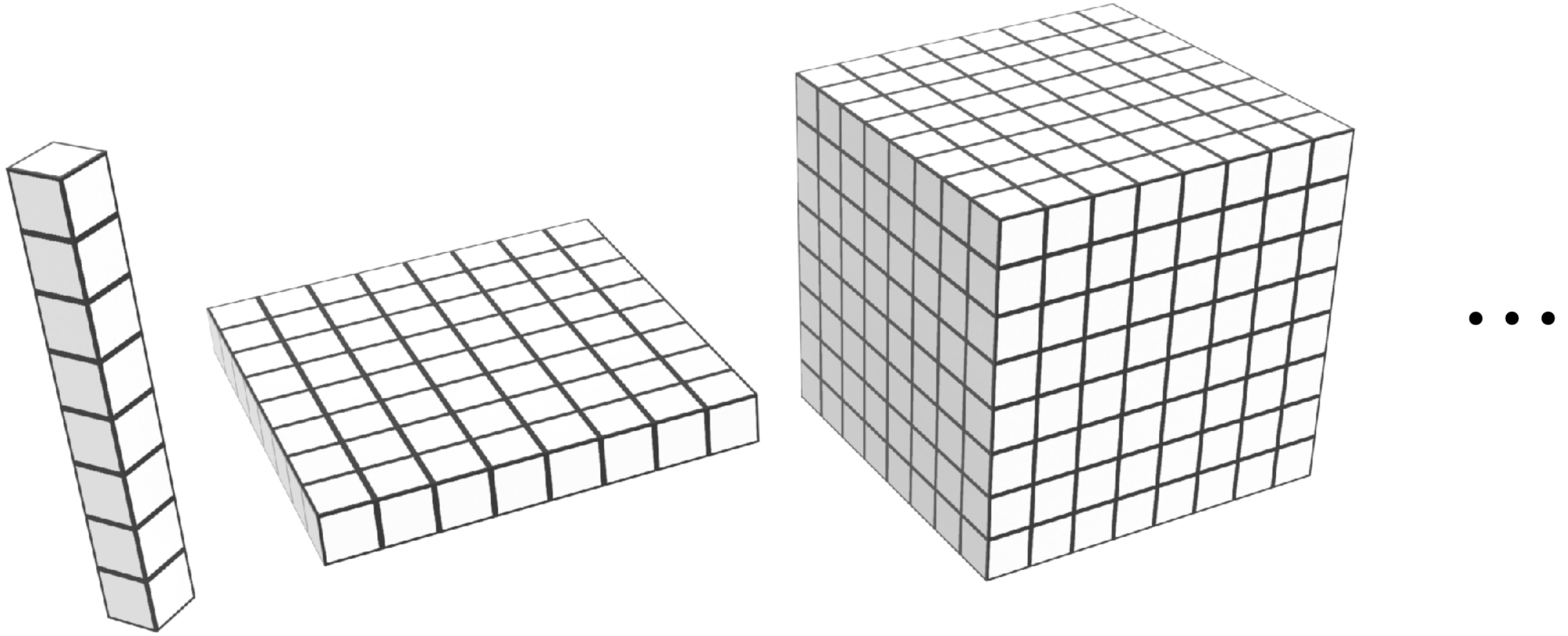
- Errors add, so error still: $O(h^2)$
- But work is now: $O(n^2)$ (n x n set of measurements)

Must perform **much** more work in 2D to get same error bound!

Curse of Dimensionality

How much does it cost to apply the trapezoid rule as we go up in dimension?

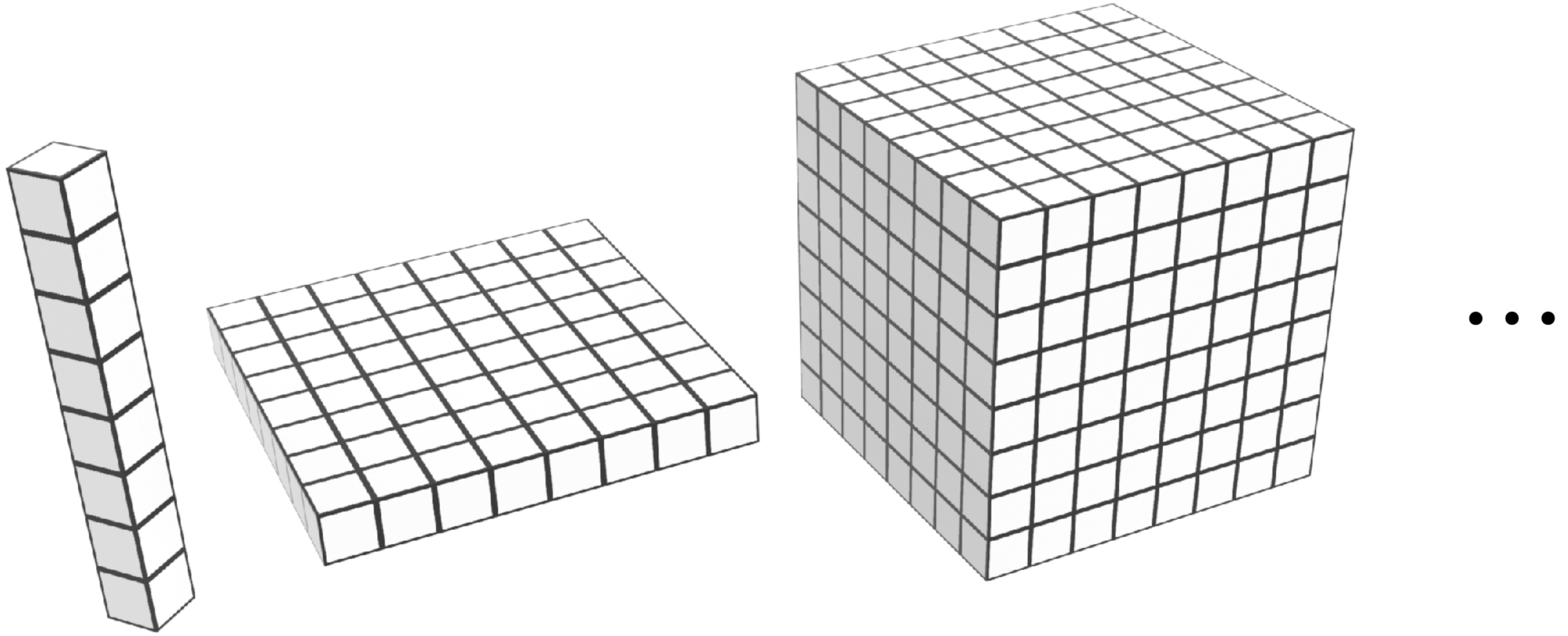
- 1D: $O(n)$
- 2D: $O(n^2)$
- …
- kD: $O(n^k)$



Curse of Dimensionality

How much does it cost to apply the trapezoid rule as we go up in dimension?

- 1D: $O(n)$
- 2D: $O(n^2)$
- …
- kD: $O(n^k)$



Deterministic quadrature does not scale to higher dimensions!

Need a fundamentally different approach…

Monte Carlo Integration

Monte Carlo vs Las Vegas

Random variation creeps into the results

Always gives the correct answer, e.g., a randomized sorting algorithm

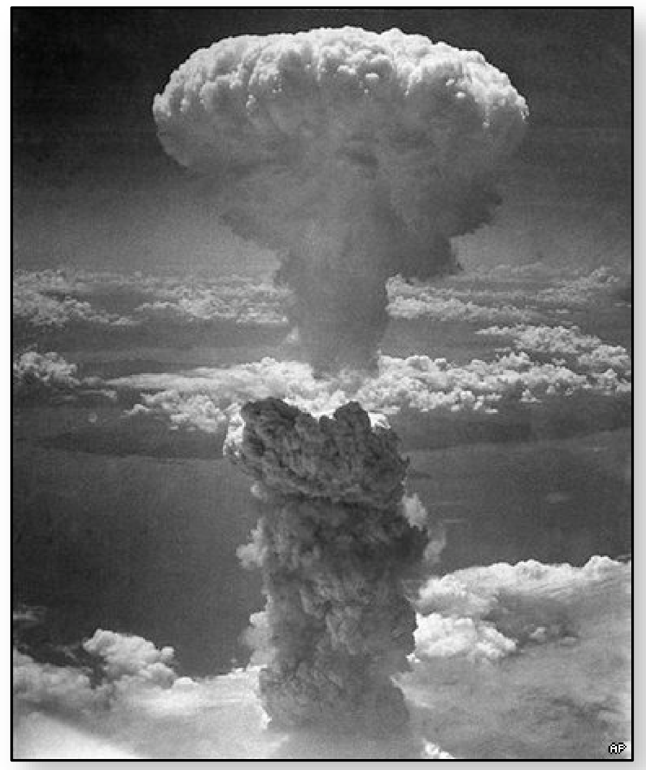
Monte Carlo History

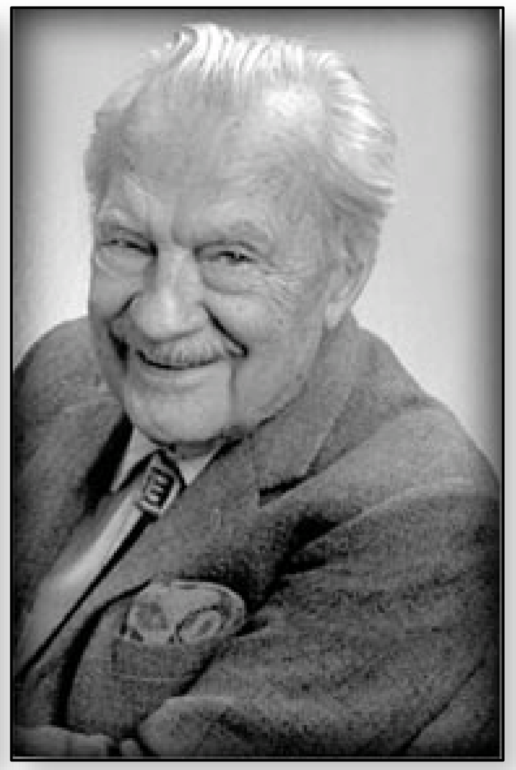
Use random numbers to solve numerical problems

Early use during development of atomic bomb

Von Neumann, Ulam, Metropolis

Named after the casino in Monte Carlo






Playing Solitaire




…

Lose Win Win Lose

What's the chance of winning with a properly shuffled deck?

Playing Solitaire

$$P_n = \frac{1}{n} \sum_{i=1}^{n} \begin{cases} 1, & \text{game } i \text{ is won,} \\ 0, & \text{game } i \text{ is lost} \end{cases}$$

$$P = \lim_{n \to \infty} P_n$$

Monte Carlo Integration

Estimate value of integral using ***random*** sampling of function

- Value of estimate depends on random samples used
- But algorithm gives the correct value "on average"

Monte Carlo Integration Advantages

Only requires function to be evaluated at random points on its domain

- Applicable to functions with discontinuities, functions that are impossible to integrate directly

Error is independent of dimensionality of integral!

- $O(n^{-0.5})$

Review: random variables

X: **random variable**. Represents a distribution of potential outcomes. Assigns a value of each outcome.

Two types: discrete vs. continuous

Discrete Random Variables

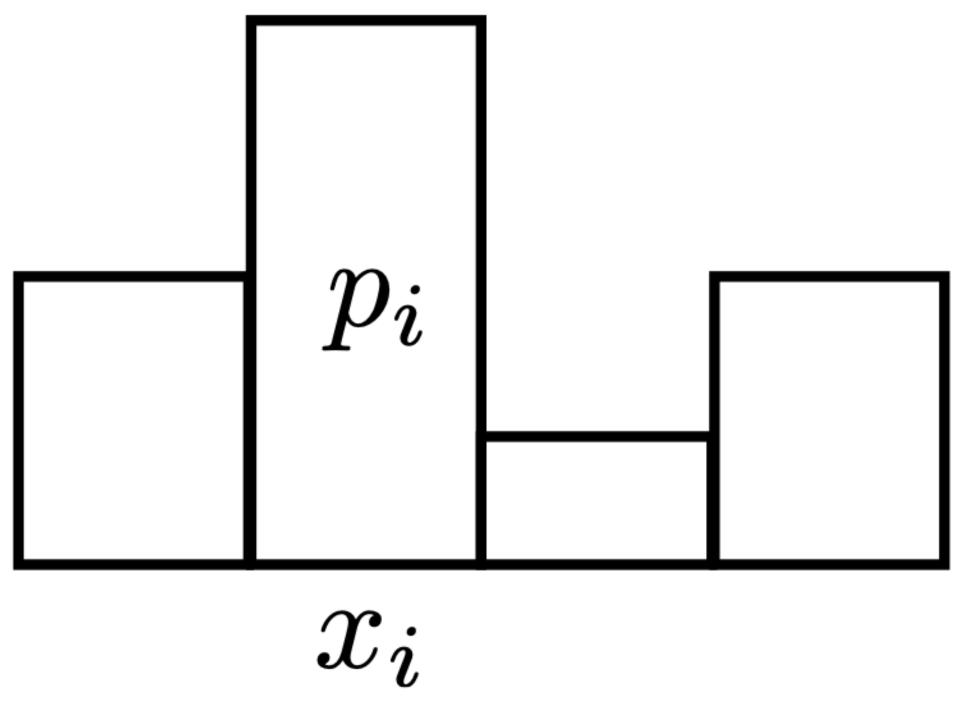
Discrete Random Variable: countable set of outcomes

Discrete Random Variables

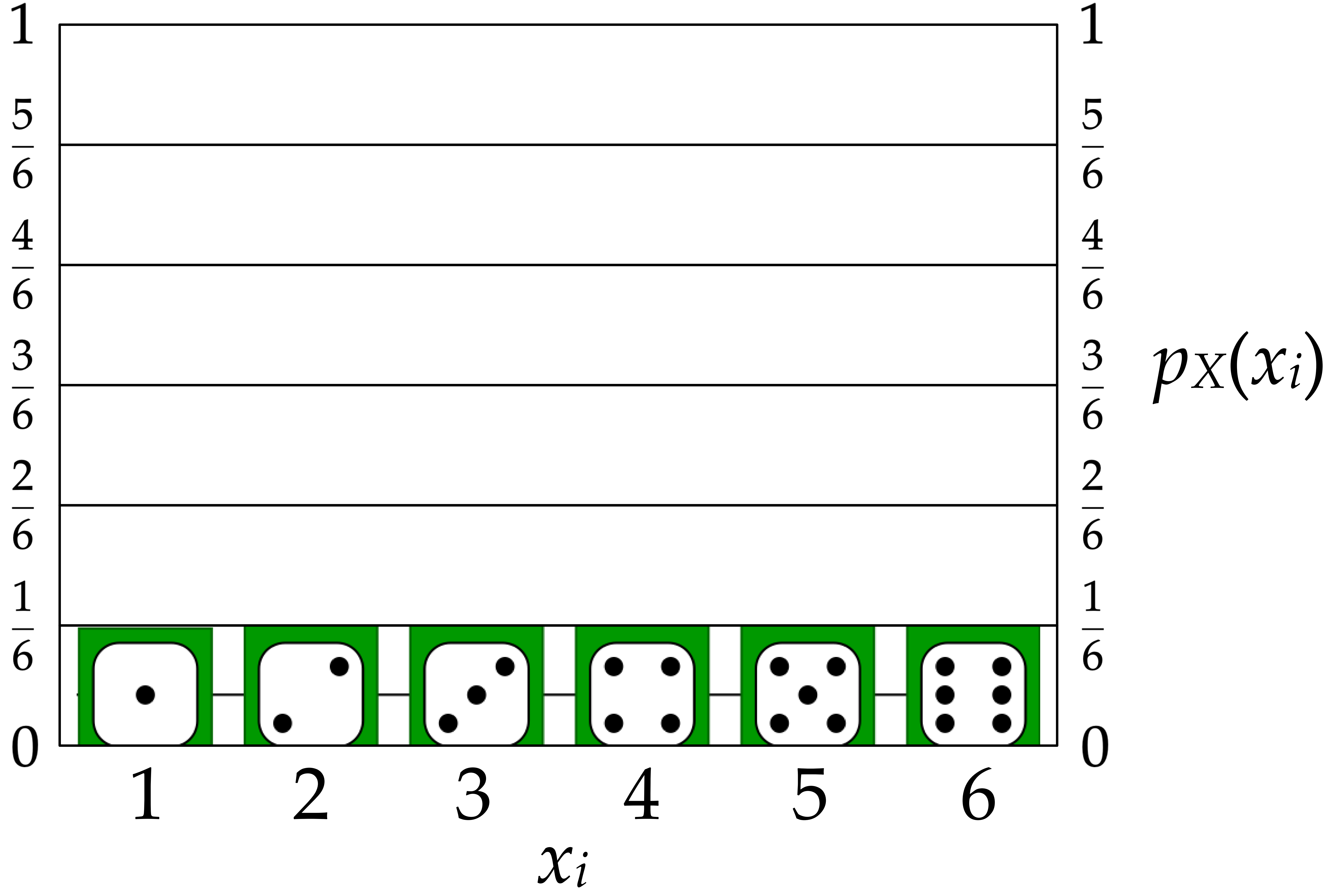
Discrete Random Variable: countable set of outcomes

Probability mass function (pmf) of X:

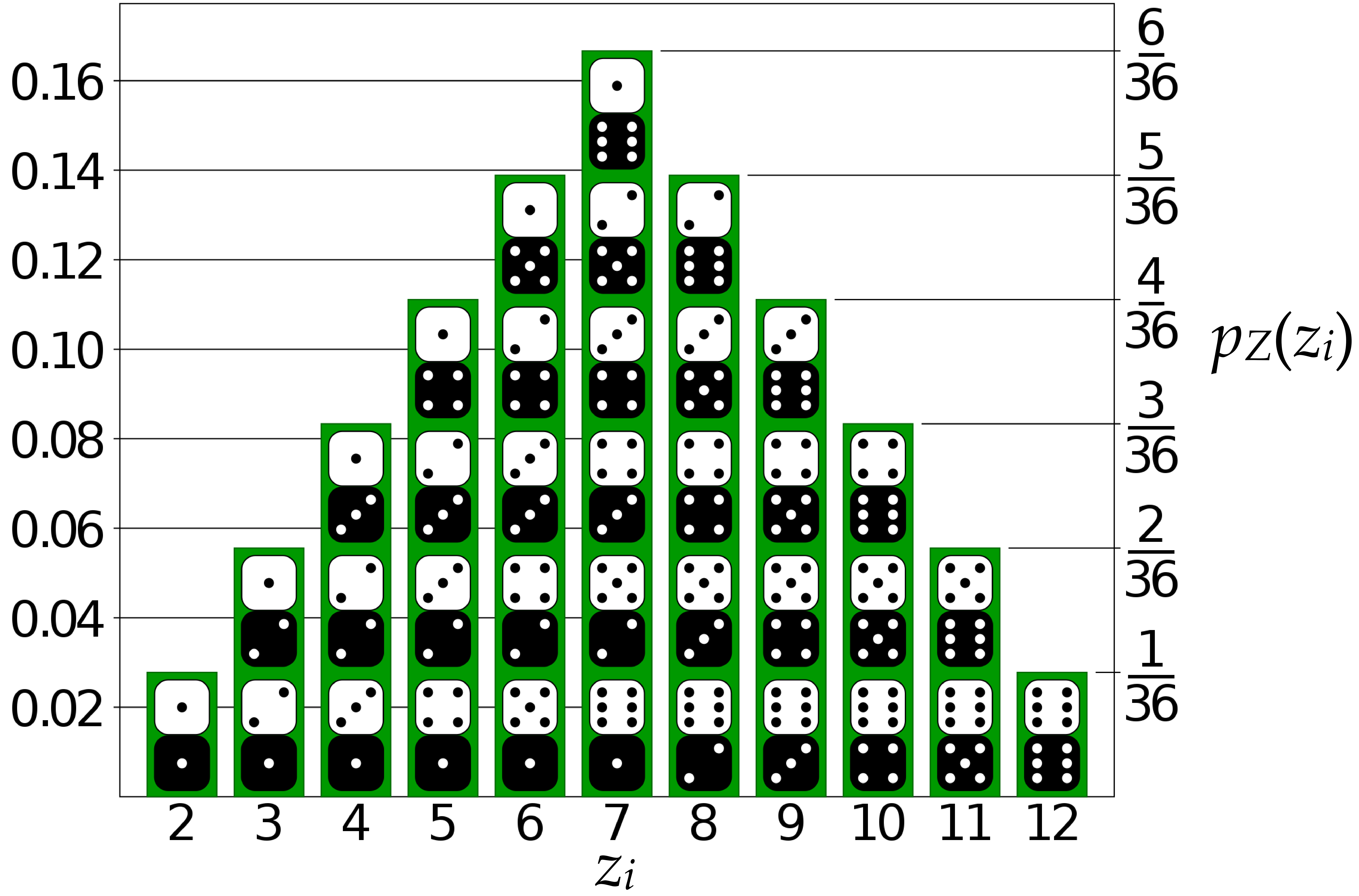
- $p_X(x_i) = P(X = x_i)$, or simply $p_i = p(x_i) = P(X = x_i)$
- $p(x_i) \geq 0$
- Sums to one: $\sum_a p(a) = 1$



Probability mass function



Probability mass function

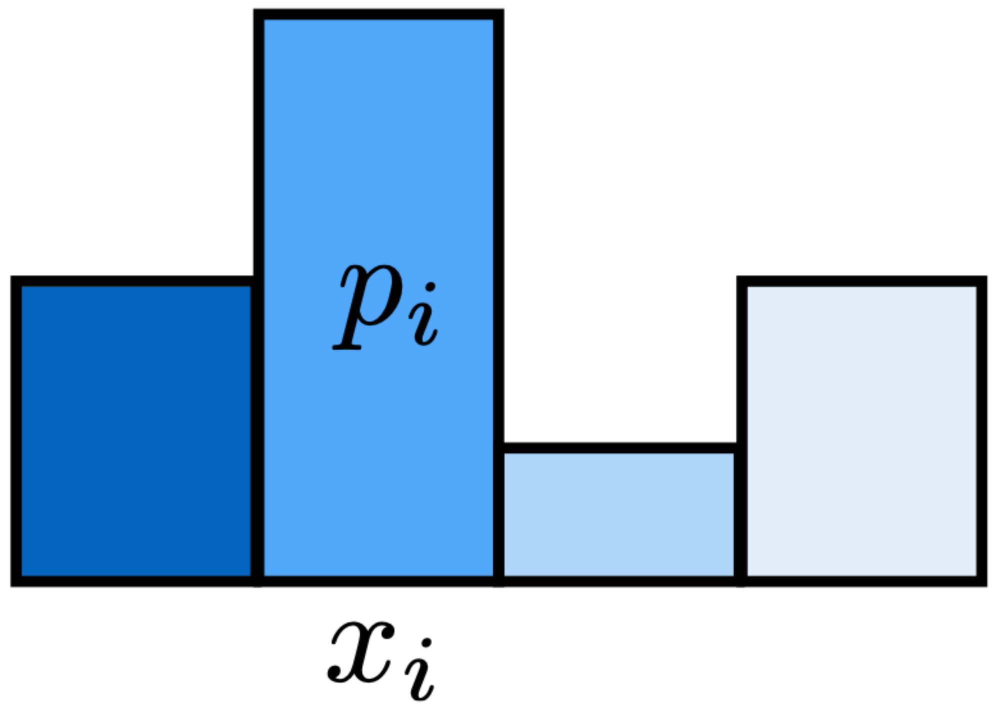


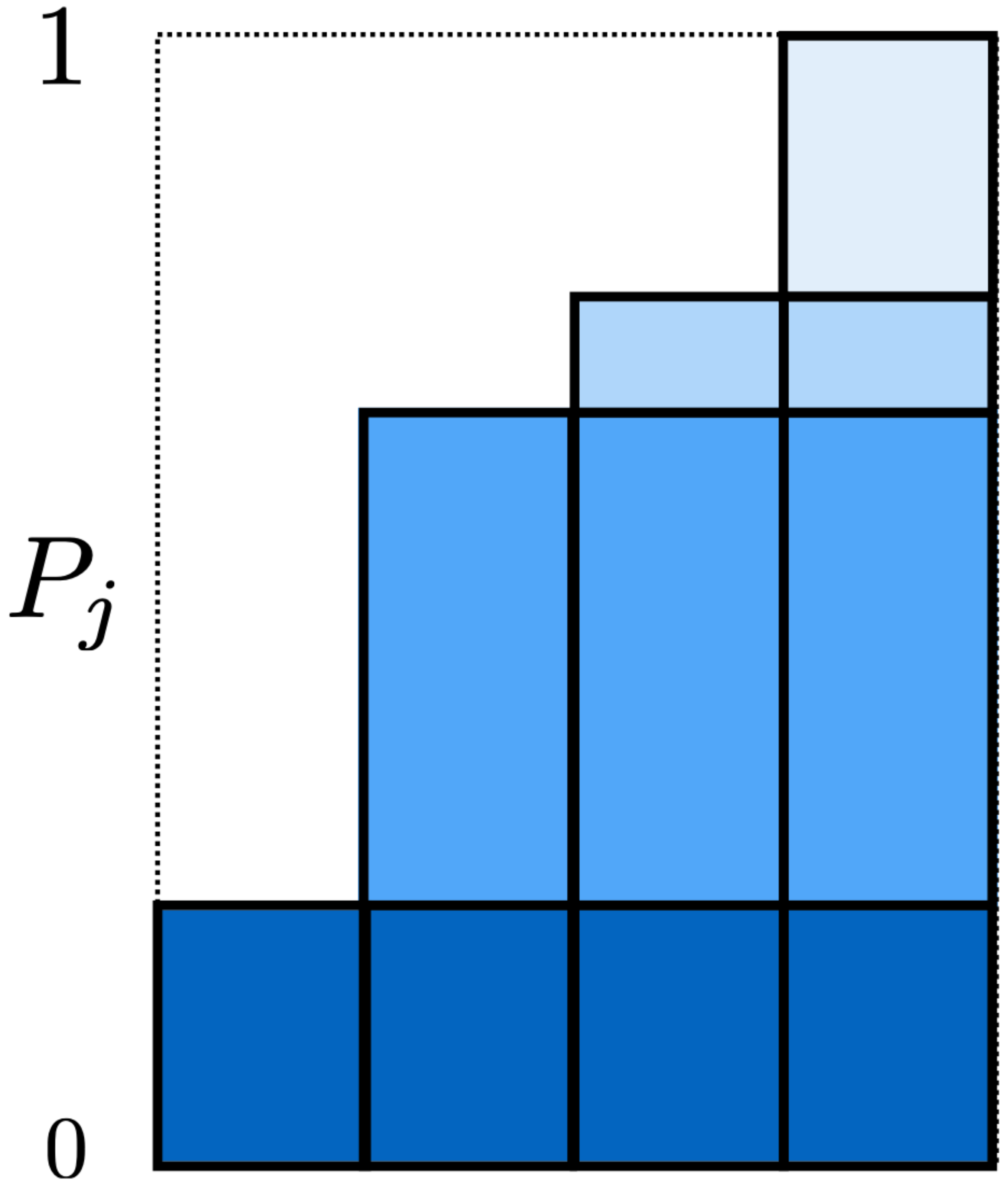
Cumulative distribution function (CDF)

Cumulative pmf: $P(j) = \sum_{i=1}^{j} p(i)$

where: $0 \leq P(i) \leq 1$

$P_n = 1$





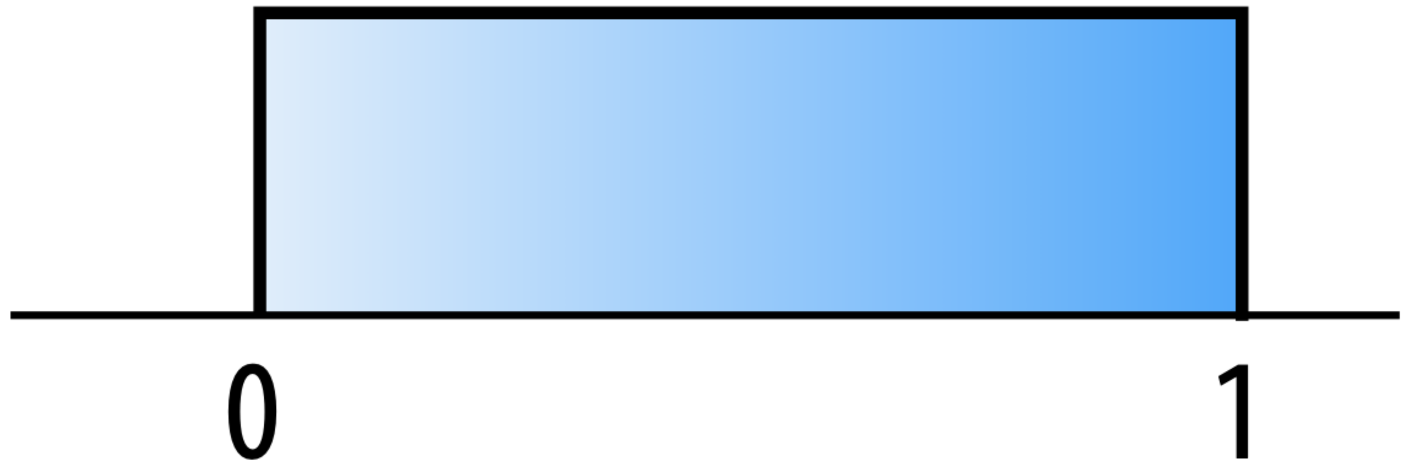
Continuous Random Variables

Probability density function (pdf) of X: $p(x)$

- $p(x) \geq 0$
- No restriction that $p(x) < 1$ (Not a probability!)

Uniform distribution

(for random variable X defined on [0,1] domain)



Continuous Random Variables

Probability density function (pdf) of X: $p(x)$

- $p(x) \geq 0$
- No restriction that $p(x) < 1$ (Not a probability!)

Cumulative distribution function (cdf): $P(x)$

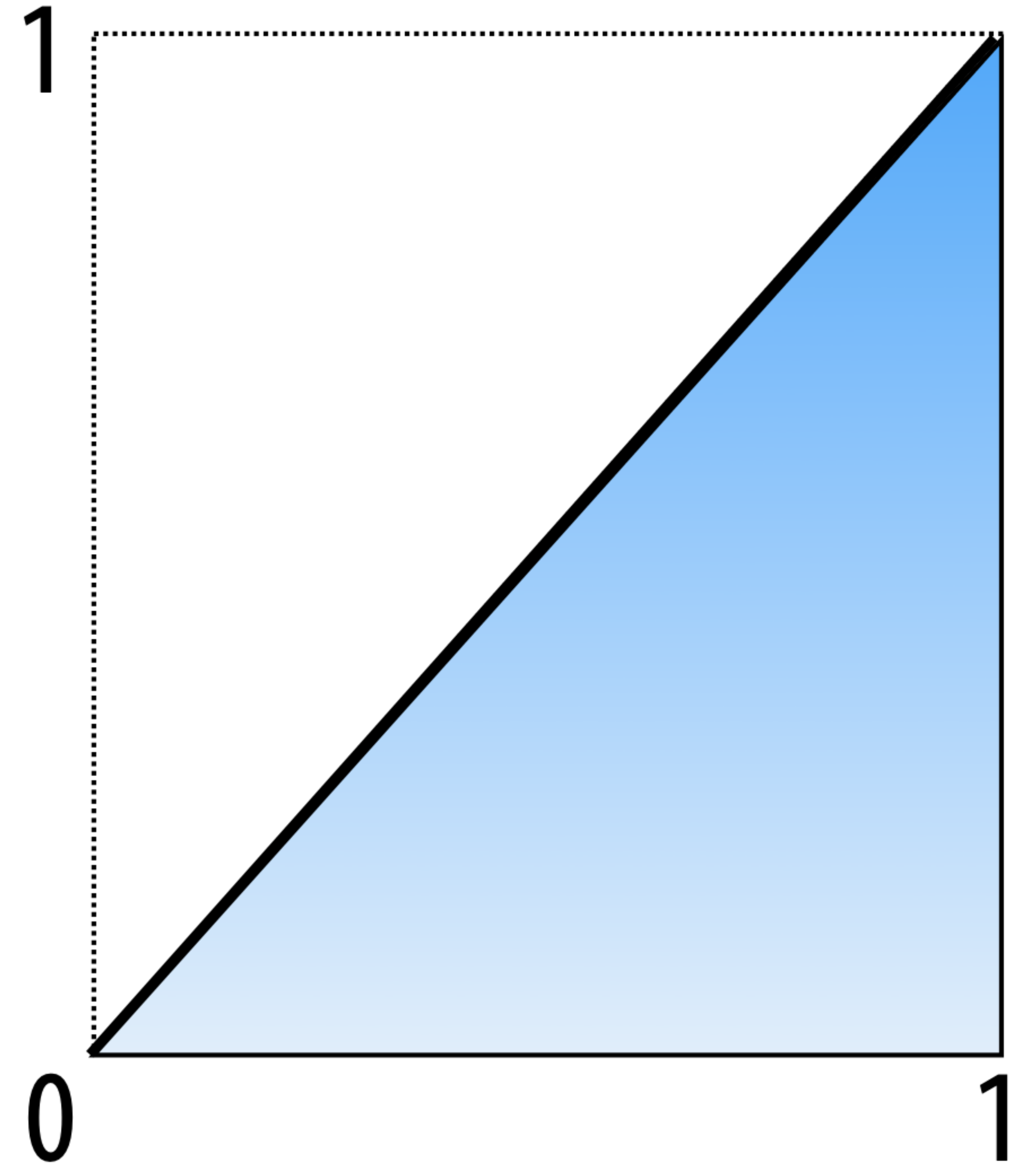
$$P(x) = \int_0^x p(x')\, \mathrm{d}x'$$

$$P(x) = \Pr(X < x)$$

$$\Pr(a \leq X \leq b) = \int_a^b p(x')\, \mathrm{d}x' = P(b) - P(a)$$

Uniform distribution

(for random variable X defined on [0,1] domain)



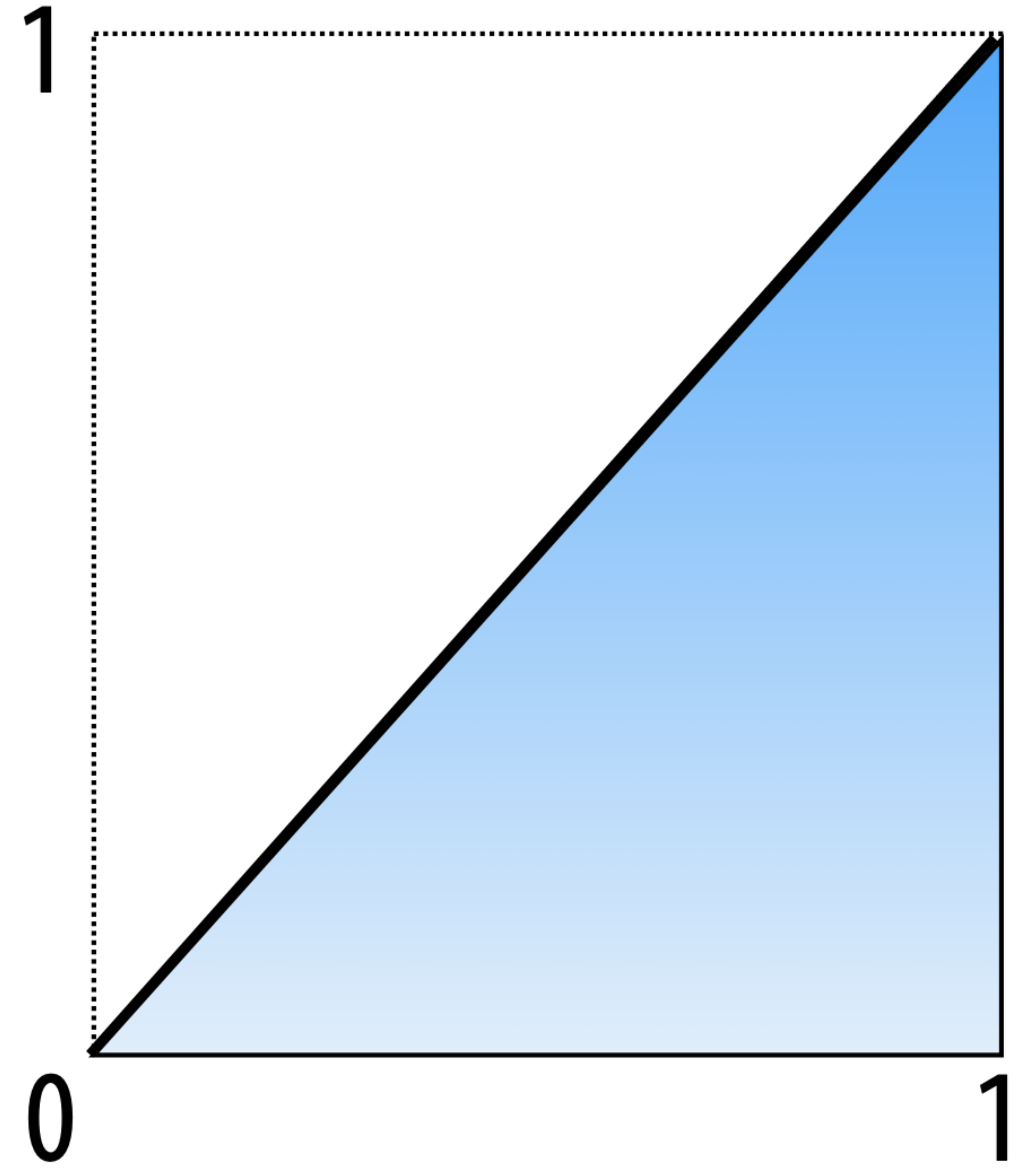
Continuous Random Variables

Canonical uniform random variable

$$p(x) = \begin{cases} 1 & x \in [0,1], \\ 0 & \text{otherwise.} \end{cases}$$

Uniform distribution

(for random variable X defined on [0,1] domain)



Ingredient: Uniform variates

Need: realizations of a uniformly distributed variable on the interval [0.0, 1.0]

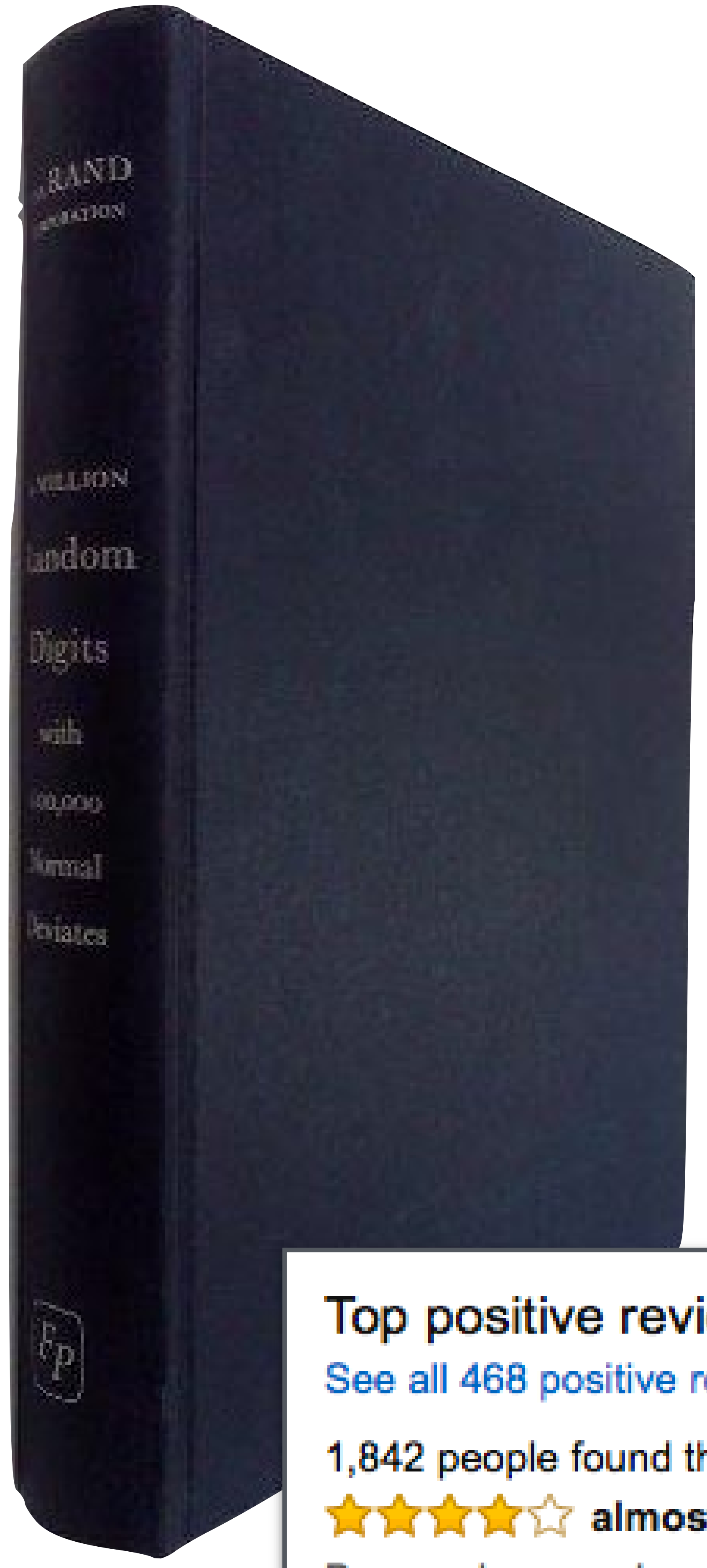
Desired properties:

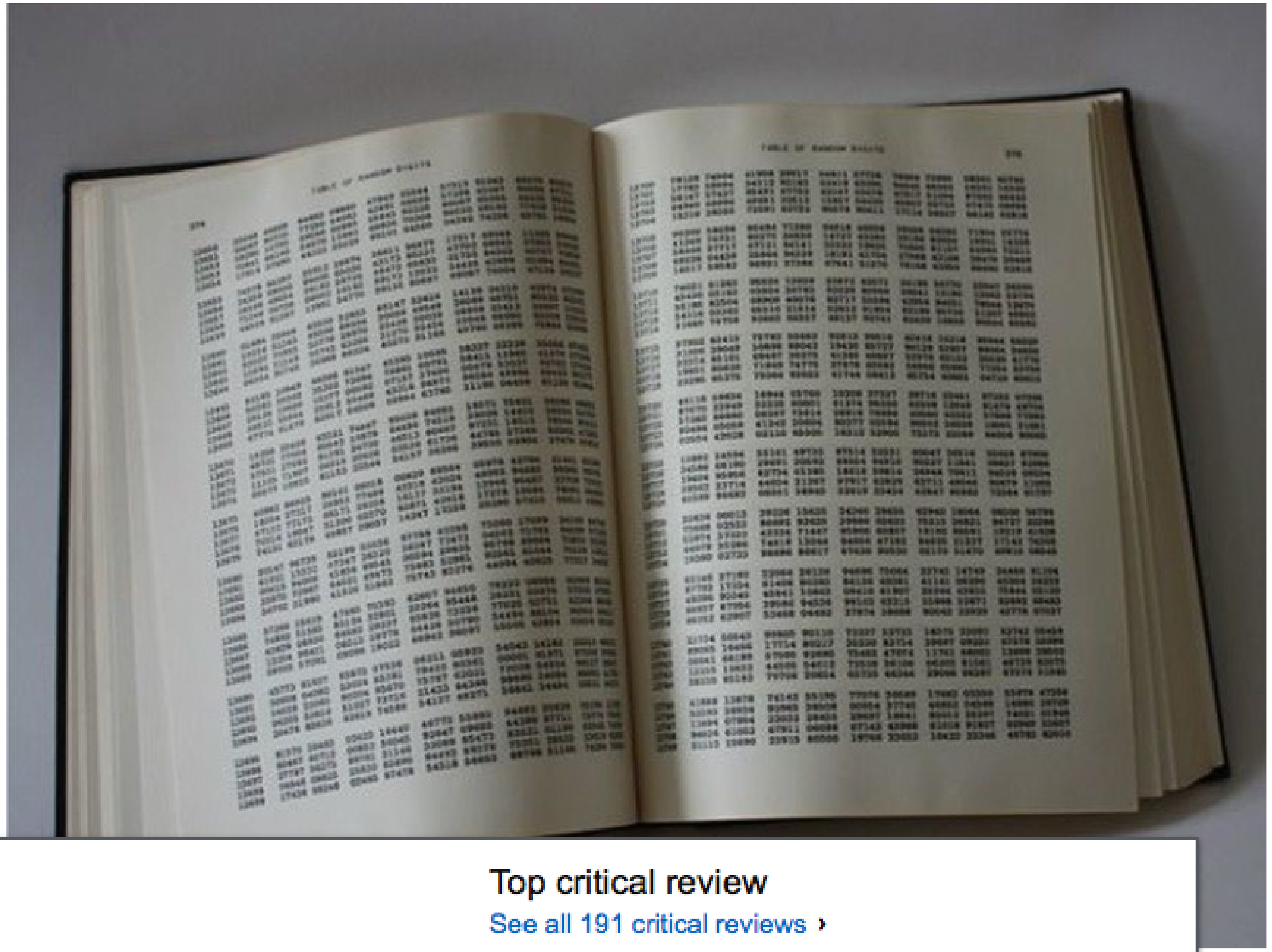
- sequence should pass statistical tests of randomness
- sequence should have a long period
- efficient to compute, requires only little storage
- repeatability: always produce the same sequence (different compilers, operating systems, processors)

Sources of randomness

3.141592653589793238462643383279502884197169399375105820974944592307816406286208998628034825342117067982148086
51328230664709384460955058223172535940812848111745028410270193852110555964462294895493038196442881097566593344
61284756482337867831652712019091456485669234603486104543266482133936072602491412737245870066063155881748815209
20962829254091715364367892590360011330530548820466521384146951941511609433057270365759591953092186117381932611
79310511854807446237996274956735188575272489122793818301194912983367336244065664308602139494639522473719070217
98609437027705392171762931767523846748184676694051320005681271452635608277857713427577896091736371787214684409
01224953430146549585371050792279689258923542019956112129021960864034418159813629774771309960518707211349999998
37297804995105973173281609631859502445945534690830264252230825334468503526193118817101000313783875288658753320
83814206171776691473035982534904287554687311595628638823537875937519577818577805321712268066130019278766111959
09216420198938095257201065485863278865936153381827968230301952035301852968995773622599413891249721775283479131
51557485724245415069595082953311686172785588907509838175463746493931925506040092770167113900984882401285836160
35637076601047101819429555961989467678374494482553797747268471040475346462080466842590694912933136770289891521
047521620569660240580381501935112533824300**35587640247496473263914199272**604269922796782354781636009341721641219
924586315030286182974555706749838505494**58858692699569092721079750930295**532116534498720275596023648066549911988
18347977535663698074265425278625518184**175746728909777727938000816470600**161452491921732172147723501414419735685
4816136115735255213347574184946843852**3323**90739**414**333454776**2416**862518983569485562099219222184272550254256887671
790494601653466804988627232791786085**784**3838279**679**766814541**0095**388378636095068006422512520511739298489608412848
862694560424196528502221066118630674**42**78622039**194**945047123**7137**869609563643719172874677646575739624138908658326
4599581339047802759009946576407895126946839835**259**57098258**2262**0522489407726719478268482601476990902640136394437
4553050682034962524517493996514314298091906592**509**37221696**4615**1570985838741059788595977297549893016175392846813
826868386894277415599185592524595395943104997**2524**68084598**7273**6446958486538367362226260991246080512438843904512
441365497627807977156914359977001296160894416**9486**85558484**0635**3422072225828488648158456028506016842739452267467
67889525213852254995466672782398645659611635**4886**230577456**4980**3559363456817432411251507606947945109659609402522
88797108931456691368672287489405601015033086**1792**868092087**4760**9178249385890097149096759852613655497818931297848
2168299894872265880485756401427047755513237**96414**515237462**3436**45428584**4**4795265867821051141354735739523113427166
102135969536231442952484937187110145765403**59027**9934403742**00731**057853**90**6219838744780847848968332144571386875194
3506430218453191048481005370614680674919**2781911**9793995206**1419663428754**4406437451237181921799983910159195618146
7514269123974894090718649423196156794520**809514**655022523160**38819301420**93762137855956638937787083039069792077346
7221825625996615014215030680384477345492**026054**1466592520149**74428507**3251866600213243408819071048633173464965145
39057962685610055081066587969981635747363**840**52571459102897064**1401**109712062804390397595156771577004203378699360
07230558763176359421873125147120532928191826186125867321579198414848829164470609575270695722091756711672291098
16909152801735067127485832228718352093539657251210835791513698820914442100675103346711031412671113699086585163
98315019701651511685171437657618351556508849099898599823873455283316355076479185358932261854896321329330898570
64204675259070915481416549859461637180270981994309924488957571282890592323326097299712084433573265489382391193
25974636673058360414281388303203824903758985243744170291327656180937734440307074692112019130203303801976211011
00449293215160842444859637669838952286847831235526582131449576857262433441893039686426243410773226978028073189
15441101044682325271620105265227211166039666557309254711055785376346682065310989652691862056476931257058635662
01855810072936065987648611791045334885034611365768675324944166803962657978771855608455296541266540853061434443
18586769751456614068007002378776591344017127494704205622305389945613140711270004078547332699390814546646458807
97270826683063432858785698305235808933065757406795457163775254202114955761581400250126228594130216471550979259
23099079654737612551765675135751782966645477917450112996148903046399471329621073404375189573596145890193897131
11790429782856475032031986915140287080859904801094121472213179476477726224142548545403321571853061422881375850

A Million Random Digits





Top positive review
See all 468 positive reviews ›
1,842 people found this helpful
★★★★☆ **almost perfect**
By a curious reader on October 26, 2006
Such a terrific reference work! But with so many terrific random digits, it's a shame they didn't sort them, to make it easier to find the one you're looking for.

Top critical review
See all 191 critical reviews ›
849 people found this helpful
★★★☆☆ **Wait for the audiobook version**
By R. Rosini on October 19, 2006
While the printed version is good, I would have expected the publisher to have an audiobook version as well. A perfect companion for one's Ipod.

A modern example: PCG32

```c
struct pcg32_random_t { uint64_t state; uint64_t inc; };

uint32_t pcg32_random_r(pcg32_random_t* rng) {
    uint64_t oldstate = rng->state;
   rng->state = oldstate * 6364136223846793005ULL + (rng->inc | 1);
   uint32_t xorshifted = ((oldstate >> 18u) ^ oldstate) >> 27u;
    uint32_t rot = oldstate >> 59u;
    return (xorshifted >> rot) | (xorshifted << ((-rot) & 31));
}
```

[http://www.pcg-random.org/]

Expected value

Intuition: what value does the random variable take, on average?

Expected value

Intuition: what value does the random variable take, on average?

- e.g., consider a fair coin where heads = 1, tails = 0
- Equal probability (1/2 both)
- Expected value is then (1/2) x 1 + (1/2) x 0 = 1/2

Expected value

Intuition: what value does the random variable take, on average?

- e.g., consider a fair coin where heads = 1, tails = 0
- Equal probability (1/2 both)
- Expected value is then (1/2) x 1 + (1/2) x 0 = 1/2

Discrete

expected value of random variable X

number of possible outcomes

$$E[X] = \sum_{i=1}^{n} p_i x_i$$

probability of i-th outcome

value of i-th outcome

Expected value

Intuition: what value does the random variable take, on average?

- e.g., consider a fair coin where heads = 1, tails = 0
- Equal probability (1/2 both)
- Expected value is then (1/2) x 1 + (1/2) x 0 = 1/2

Discrete

expected value of random variable X

number of possible outcomes

$$E[X] = \sum_{i=1}^{n} p_i x_i$$

probability of i-th outcome

value of i-th outcome

Continuous

$$E[X] = \int_{\mathbb{R}} p(x) x \, \mathrm{d}x$$

Expected value

Intuition: what value does the random variable take, on average?

- e.g., consider a fair coin where heads = 1, tails = 0
- Equal probability (1/2 both)
- Expected value is then (1/2) x 1 + (1/2) x 0 = 1/2

Discrete

expected value of random variable X

number of possible outcomes

$$E[X] = \sum_{i=1}^{n} p_i x_i$$

probability of i-th outcome

value of i-th outcome

Continuous

$$E[X] = \int_{\mathbb{R}} p(x) x \, \mathrm{d}x$$

Properties

$$E[X_1 + X_2] =$$

$$E[aX] =$$

Expected value

Intuition: what value does the random variable take, on average?

- e.g., consider a fair coin where heads = 1, tails = 0
- Equal probability (1/2 both)
- Expected value is then (1/2) x 1 + (1/2) x 0 = 1/2

Discrete

expected value of random variable X

number of possible outcomes

$$E[X] = \sum_{i=1}^{n} p_i x_i$$

probability of i-th outcome

value of i-th outcome

Continuous

$$E[X] = \int_{\mathbb{R}} p(x)x \, \mathrm{d}x$$

Properties

$$E[X_1 + X_2] = E[X_1] + E[X_2]$$

$$E[aX] = aE[X]$$

Variance

Intuition: how far are the samples from the average, on average?

Variance

Intuition: how far are the samples from the average, on average?

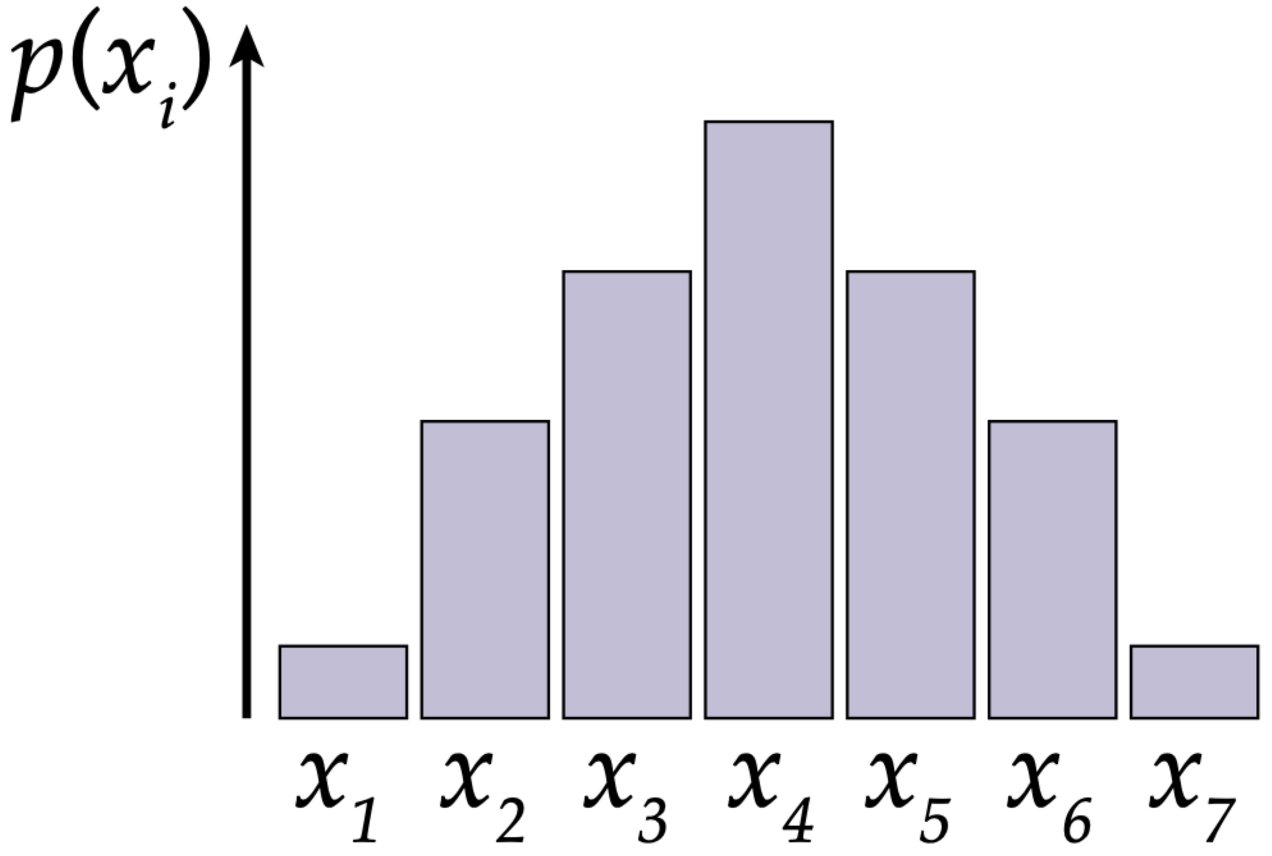
Definition: $$V[X] = E\left[(X - E[X])^2\right]$$

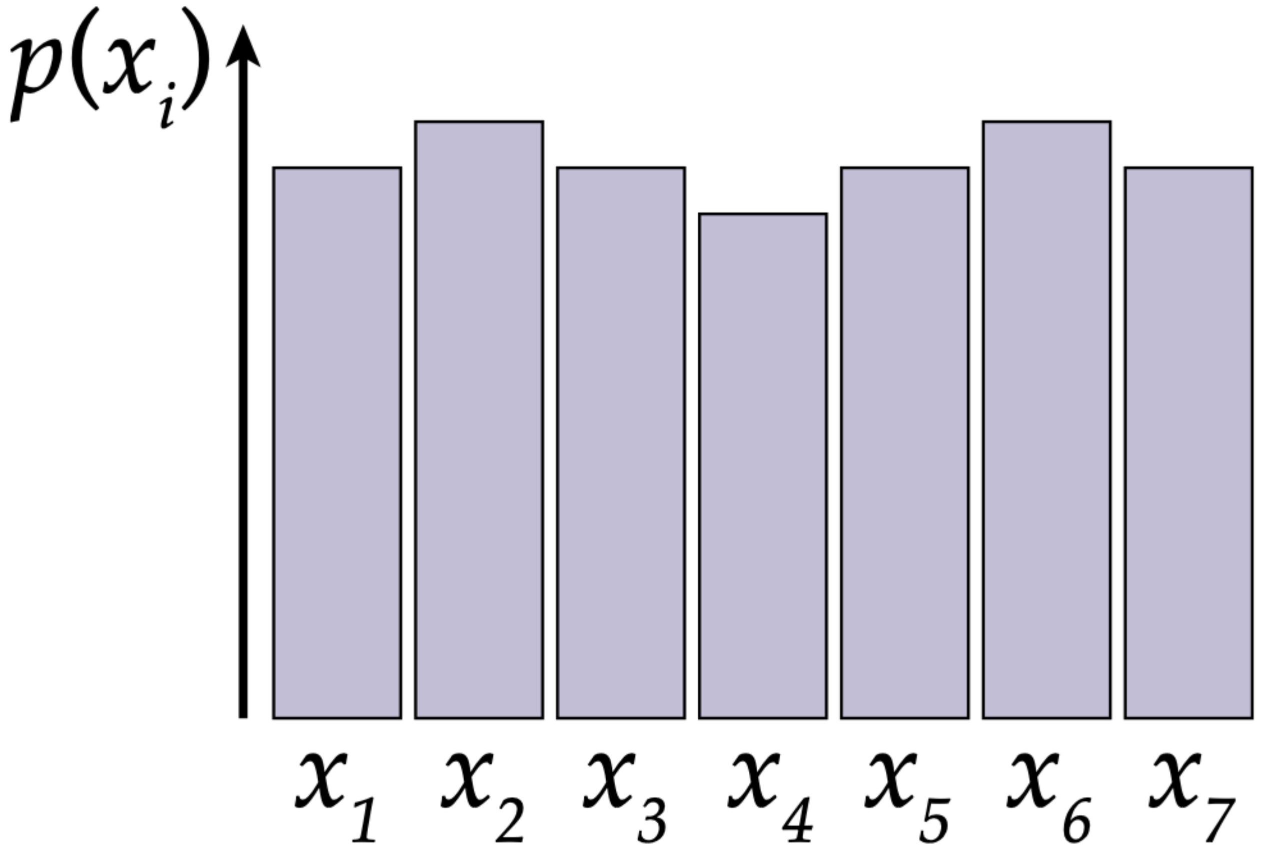
Variance

Intuition: how far are the samples from the average, on average?

Definition: $$V[X] = E\left[(X - E[X])^2\right]$$

Q: Which of these has higher variance?



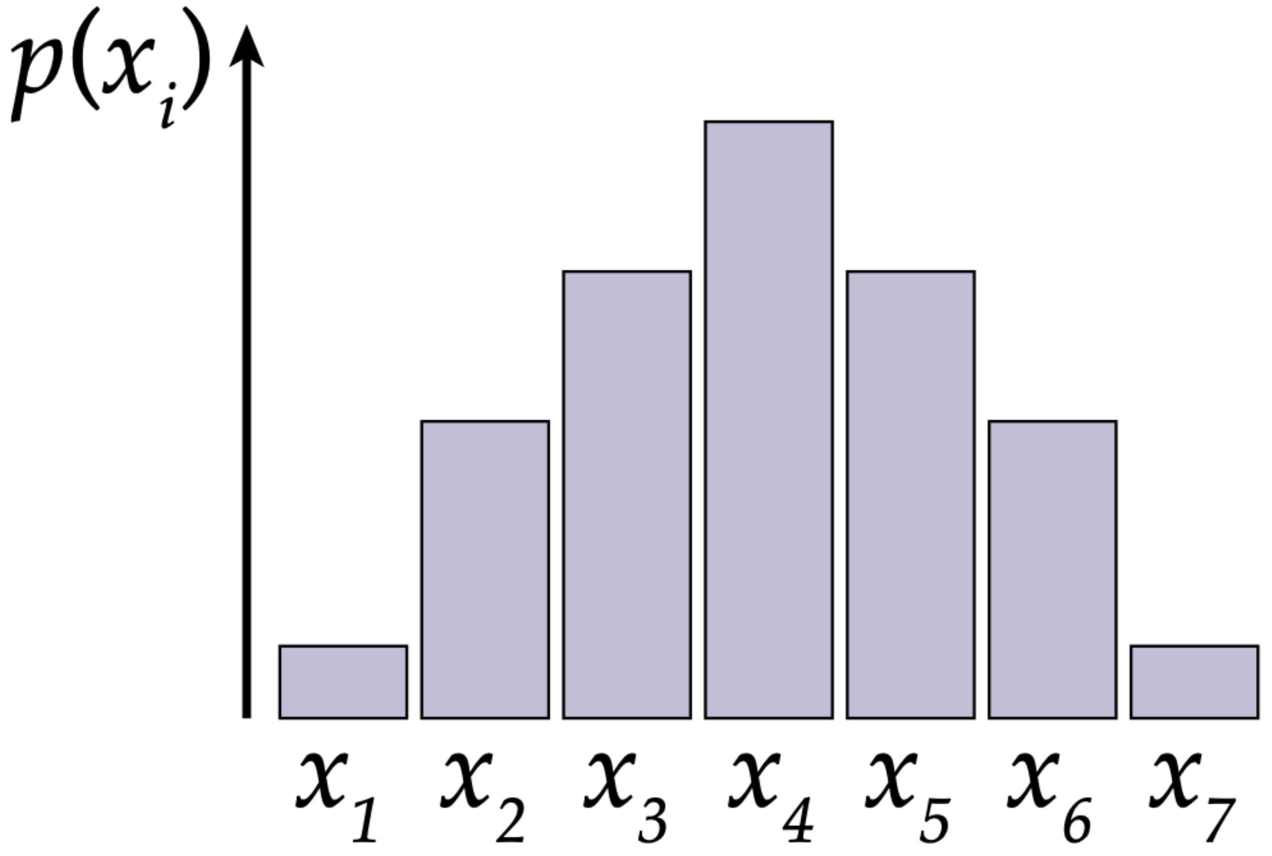


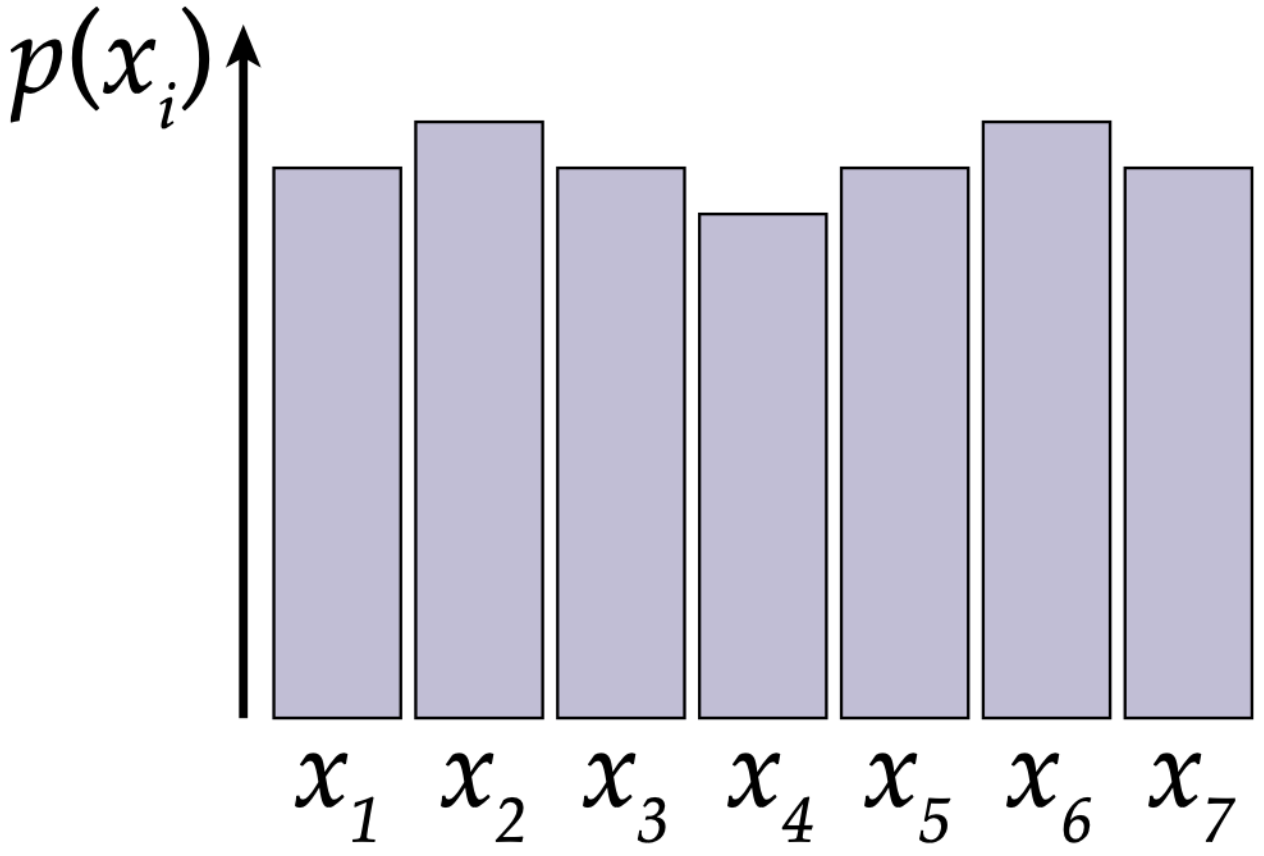
Variance

Intuition: how far are the samples from the average, on average?

Definition: $$V[X] = E\left[(X - E[X])^2\right]$$

Q: Which of these has higher variance?





Properties

$$V[X] =$$

$$V[X_1 + X_2] =$$

$$V[aX] =$$

only if uncorrelated!

Monte Carlo Integration

Motivation: want to compute the integral

$$F = \int_D f(x)\,\mathrm{d}x$$

Could we approximate F by averaging a number of realizations x_i of a random process?

$$\frac{1}{N}\sum_{i=1}^{N} f(x_i)$$

Monte Carlo Integration

$$E\left[\frac{1}{N}\sum_{i=1}^{N} f(X_i)\right] = \frac{1}{N}\sum_{i=1}^{N} E\big[f(X_i)\big]$$

$$= E\big[f(X_i)\big]$$

$$= \int_D f(x)\, p_{X_i}(x)\, \mathrm{d}x$$

Aside: why can we do this?
Law of the unconscious statistician (LOTUS)

(oops, that's not what we wanted!)

Monte Carlo Integration

Motivation: want to compute the integral

$$F = \int_D f(x)\,\mathrm{d}x$$

Solution: Approximate F by averaging realizations of a random variable X, and explicitly accounting for its PDF:

$$F \approx \frac{1}{N} \sum_{i=1}^{N} \frac{f(x_i)}{p(x_i)}$$

Monte Carlo Integration

$$E\left[\frac{1}{N}\sum_{i=1}^{N}\frac{f(X_i)}{p(X_i)}\right] = \frac{1}{N}\sum_{i=1}^{N}E\left[\frac{f(X_i)}{p(X_i)}\right]$$

Monte Carlo integration is correct *on average.*

- This assumes that $p(X_i) \neq 0$ when $f(X_i) \neq 0$.
- This property is called *unbiasedness.*

$$= E\left[\frac{f(X_i)}{p(X_i)}\right]$$

$$= \int_D \frac{f(X_i)}{p(X_i)}p(X_i)\mathrm{d}x$$

$$= \int_D f(X_i)\mathrm{d}x = F$$

Monte Carlo Integration

Requirement (why?)

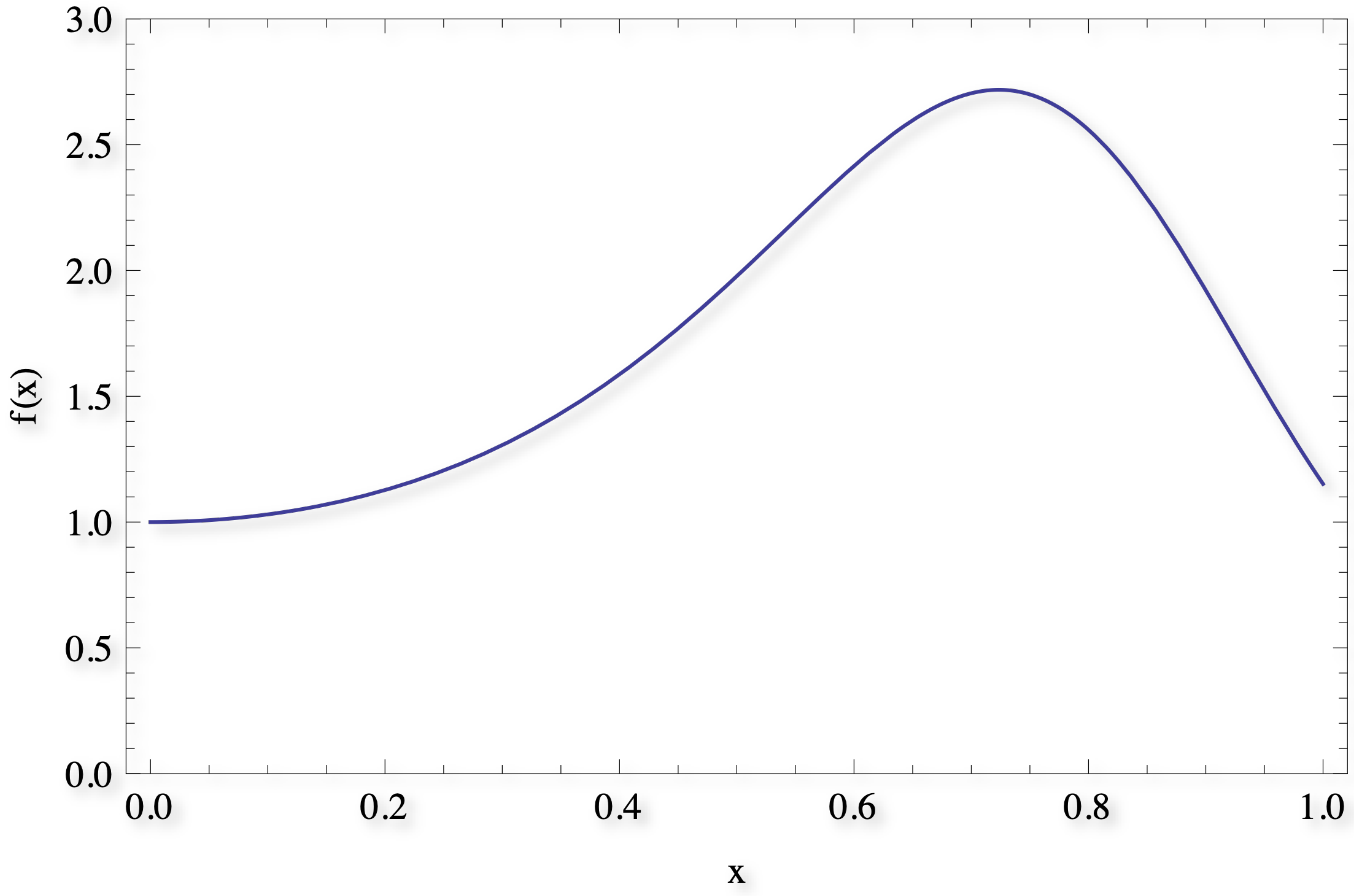
$$f(x) \neq 0 \Rightarrow p(x) > 0$$

Domain D might be: plane, sphere, hemisphere, surface of an object

Reasonable default for $p(x)$: uniform distribution

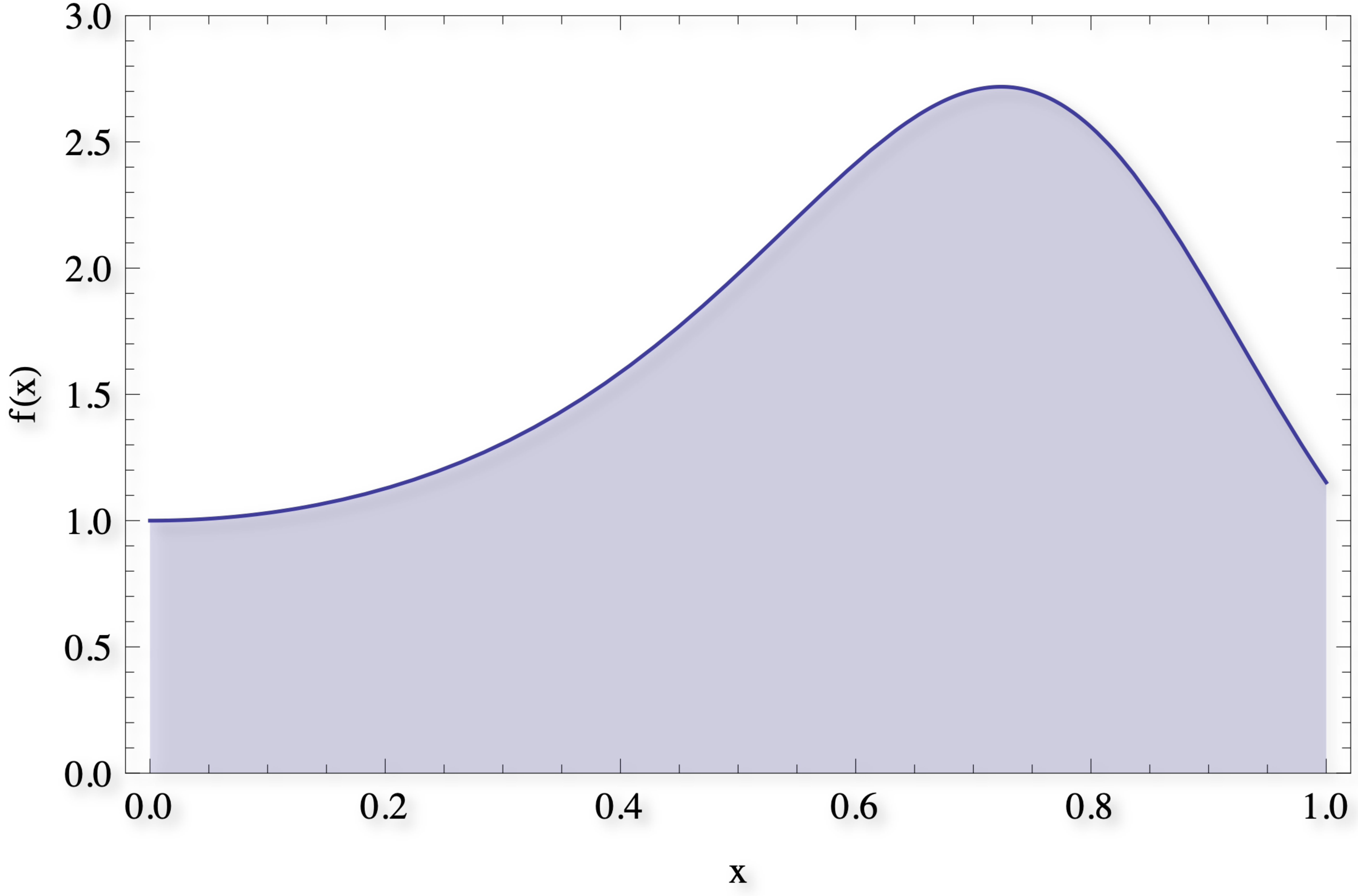
Monte Carlo Integration

$$f(x) = e^{\sin(3x^2)}$$



Monte Carlo Integration

$$F = \int_0^1 e^{\sin(3x^2)}\, dx$$



Monte Carlo Integration

$$F = \int_0^1 e^{\sin(3x^2)}\, dx \approx F_N = \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{p(x_i)} \Rightarrow \frac{1}{N}\sum_{i=1}^{N} f(x_i)$$

```
double integrate(int N)
{
    double x, sum=0.0;
    for (int i = 0; i < N; ++i) {
        x = randf();
        sum += exp(sin(3*x*x));
    }
    return sum / double(N);
}
```

$$p(x_i) = 1$$

Monte Carlo Integration

$$F = \int_{a}^{b} e^{\sin(3x^2)}\, dx \approx F_N = \frac{1}{N} \sum_{i=1}^{N} \frac{f(x_i)}{p(x_i)}$$

```
double integrate(int N, double a, double b)
{
    double x, sum=0.0;
    for (int i = 0; i < N; ++i) {
        x = randf();
        sum += exp(sin(3*x*x));
    }
    return sum / double(N);
}
```

Monte Carlo Integration

$$F = \int_a^b e^{\sin(3x^2)}\,dx \approx F_N = \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{p(x_i)}$$

```
double integrate(int N, double a, double b)
{
    double x, sum=0.0;
    for (int i = 0; i < N; ++i) {
        x = a + randf()*(b-a);
        sum += exp(sin(3*x*x));
    }
    return sum / double(N);
}
```

$$p(x_i) = \frac{1}{b-a}$$

Monte Carlo Integration

$$F = \int_a^b e^{\sin(3x^2)}\,dx \approx F_N = \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{p(x_i)}$$

```
double integrate(int N, double a, double b)
{
   double x, sum=0.0;
   for (int i = 0; i < N; ++i) {
      x = a + randf()*(b-a);
      sum += exp(sin(3*x*x)) / (1/(b-a));
   }
   return sum / double(N);
}
```

$$p(x_i) = \frac{1}{b-a}$$

Monte Carlo Integration

$$f(x) = e^{\sin(3x^2)}$$

N	F_N
1	2.75039
10	1.9893
100	1.79139
1000	1.75146
10000	1.77313
100000	1.77862

True value: 1.760977217585905…

Monte Carlo Integration

$$f(x) = e^{\sin(3x^2)}$$

N	F_N
1	2.75039
10	1.9893
100	[illegible]
1000	1.75146
10000	1.77313
100000	1.77862

True value: 1.76097217585905…

Remarkable thing about this:

Dimension doesn't matter

Monte Carlo Error

$$E[\|F_N - F\|^2] = E[F_N^2 - 2F_N F + F^2]$$

$$= E[F_N^2] - E[2F_N F] + E[F^2]$$

For an *unbiased* estimator, its average error is equal to its variance!

$$= E[F_N^2] - 2E[F_N]F + F^2$$

$$= E[F_N^2] - 2FF + F^2$$

$$= E[F_N^2] - F^2$$

$$= E[F_N^2] - E[F_N]^2 = V[F_N]$$

Monte Carlo error

Variance:

$$V\left[\langle F^N\rangle\right] = V\left[\frac{1}{N}\sum_{i=0}^{N-1}\frac{f(X_i)}{\text{pdf}(X_i)}\right]$$ ← assume uncorrelated samples

$$= \frac{1}{N^2}\sum_{i=0}^{N-1} V\left[\frac{f(X_i)}{\text{pdf}(X_i)}\right]$$

$$= \frac{1}{N^2}\sum_{i=0}^{N-1} V\left[Y_i\right]$$

$$= \frac{1}{N} V\left[Y\right]$$

Monte Carlo error

Variance:

$$V\left[\langle F^N\rangle\right] = V\left[\frac{1}{N}\sum_{i=0}^{N-1}\frac{f(X_i)}{\text{pdf}(X_i)}\right]$$

← assume uncorrelated samples

$$= \frac{1}{N^2}\sum_{i=0}^{N-1} V\left[\frac{f(X_i)}{\text{pdf}(X_i)}\right]$$

$$= \frac{1}{N^2}\sum_{i=0}^{N-1} V\left[Y_i\right]$$

$$= \frac{1}{N} V\left[Y\right]$$

What happens if samples are correlated?

Std. deviation: $\sigma\left[\langle F^N\rangle\right] = \boxed{\frac{1}{\sqrt{N}}\sigma\left[Y\right]}$

- Error scaling is independent of dimensionality!
- Error converges to zero as $N \to \infty$.
- This property is called *consistency.*

Unbiasedness and consistency

Both are desirable, but different, properties of an estimator.

- An estimator can be consistent but not unbiased.

Unbiasedness: You can reduce error by averaging rendered images from independent finite-sample rendering runs. As the number of images grows infinite, the error goes to zero.

Consistency: You can reduce error by increasing the number of samples in a single rendering run. As the number of samples grows infinite, the error goes to zero.

Monte Carlo Methods

Pros

- Flexible
- Easy to implement
- Easily handles complex integrands
- Efficient for high dimensional integrands
- *Unbiased* estimator

Cons

- Variance (noise)
- "Slow" convergence* [but independent of dimension, so it's actually pretty fast at higher dimensions]

$$O(1/\sqrt{N})$$

Monte Carlo Integration Summary

Goal: evaluate integral $$\int_a^b f(x)dx$$

Random variable $$X_i \sim p(x)$$

Monte Carlo Estimator $$F_N = \frac{1}{N}\sum_{i=1}^{N}\frac{f(X_i)}{p(X_i)}$$

Expectation $$E[F_N] = \int_a^b f(x)dx$$

Remaining Agenda

$$F_N = \frac{1}{N} \sum_{i=1}^{N} \frac{f(X_i)}{p(X_i)}$$

Main practical issues:

- How to choose $p(x)$
- How to generate x_i according to $p(x)$

Ambient Occlusion

$$L_r(\mathbf{x}, \vec{\omega}_r) = \int_{H^2} f_r(\mathbf{x}, \vec{\omega}_i, \vec{\omega}_r) L_i(\mathbf{x}, \vec{\omega}_i) \cos\theta_i \, d\vec{\omega}_i$$

Sampling Random Variables

Sampling the function domain:

- Uniform unit interval (0,1)
- Uniform interval (a,b)
- Circle?
- Sphere?
- Hemisphere?
- More complex domains?

Example: uniformly sampling a disk

Uniform probability density on a unit disk

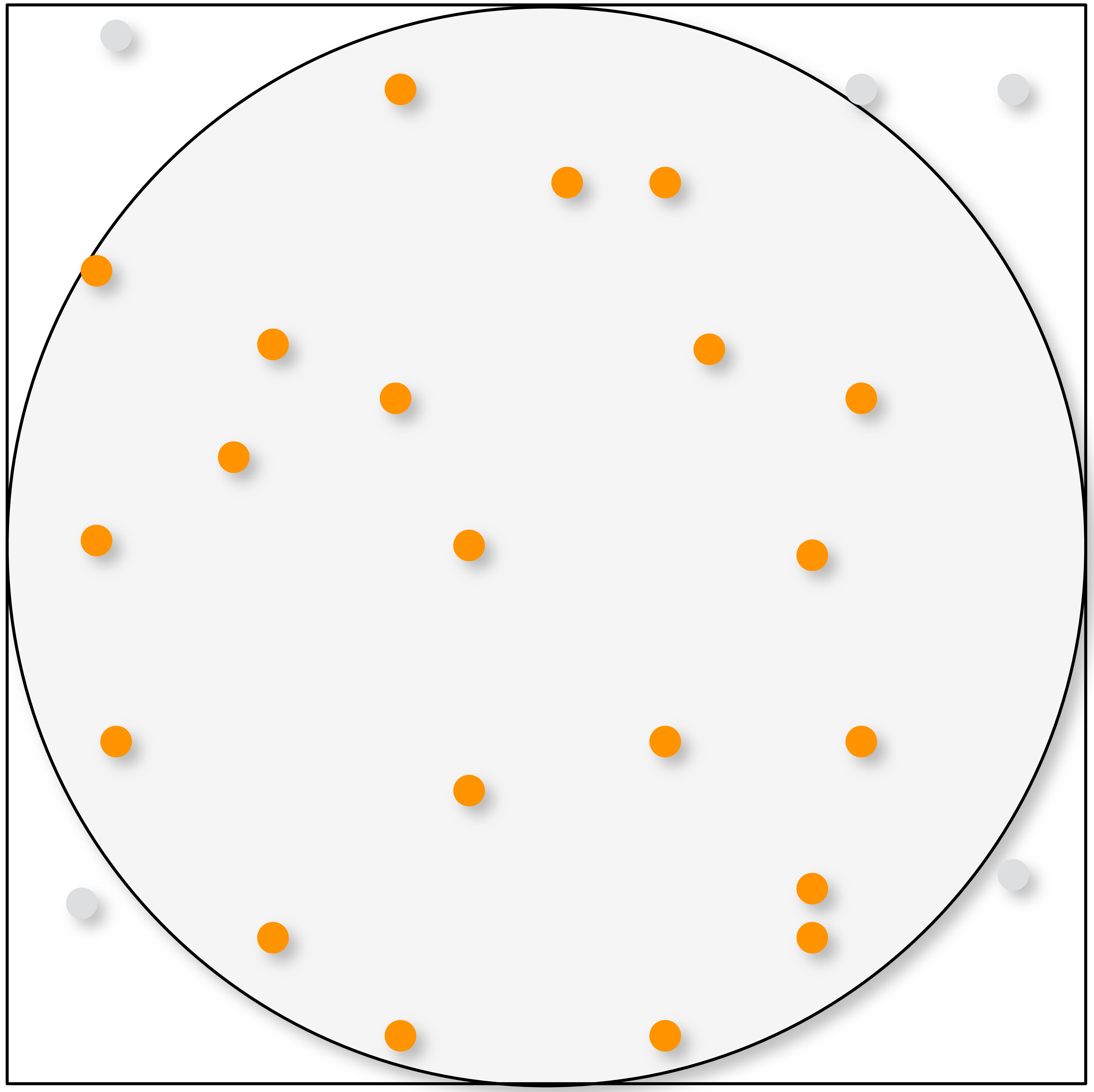
$$p(x,y) = \begin{cases} \frac{1}{\pi} & x^2 + y^2 < 1 \\ 0 & \text{otherwise} \end{cases}$$

Goal: draw samples X_i, Y_i that are distributed as:

$$(X_i, Y_i) \sim p(x,y)$$

Problem: pseudo-random number generator only allows us to draw samples from a canonical uniform distribution

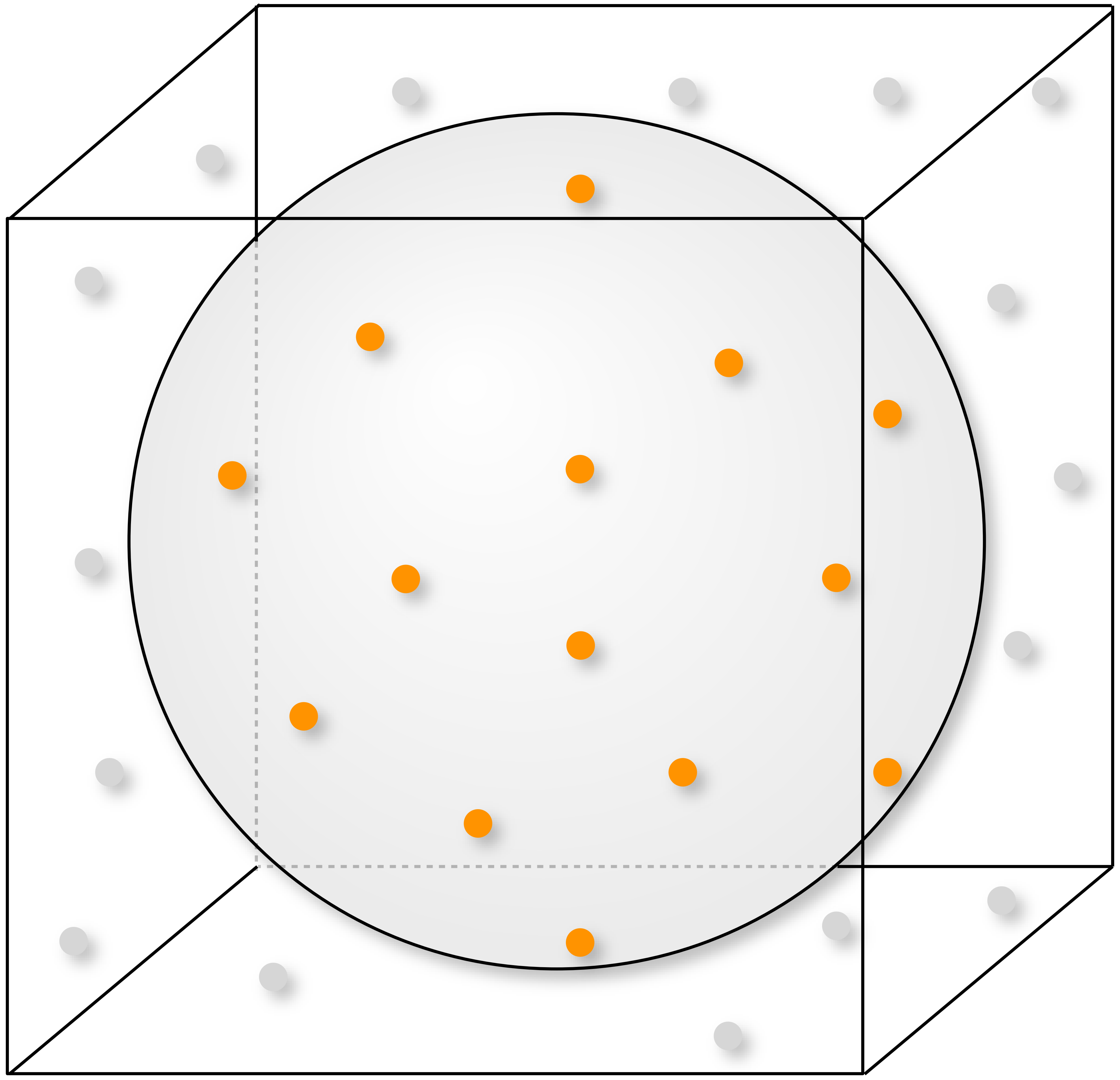
Rejection Sampling in a Disk



```
Vector2 v;
do
{
    v.x = 1-2*randf();
    v.y = 1-2*randf();
} while (dot(v,v) > 1)
```

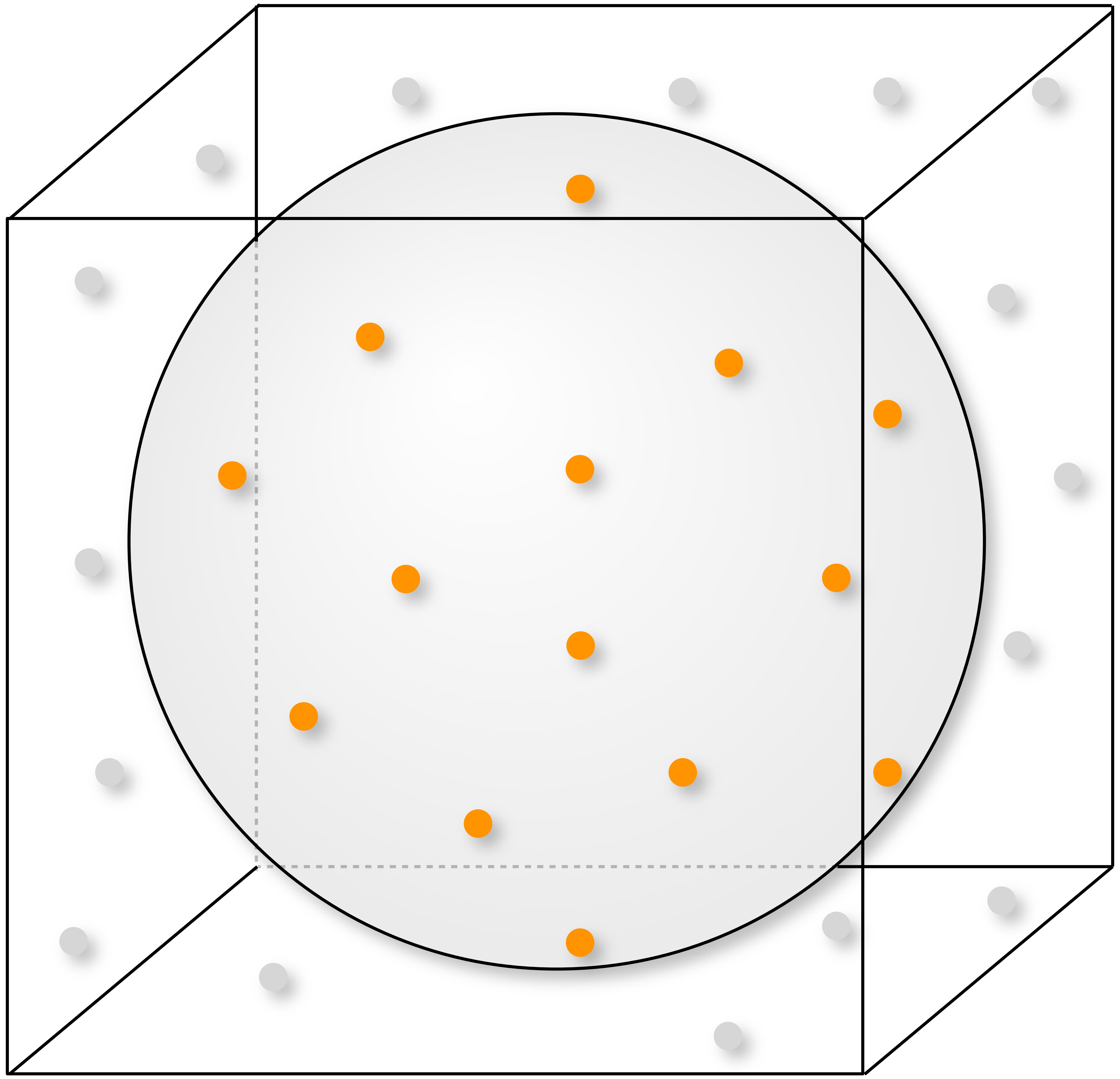
– Similar technique for sampling a sphere

Rejection Sampling in a Sphere



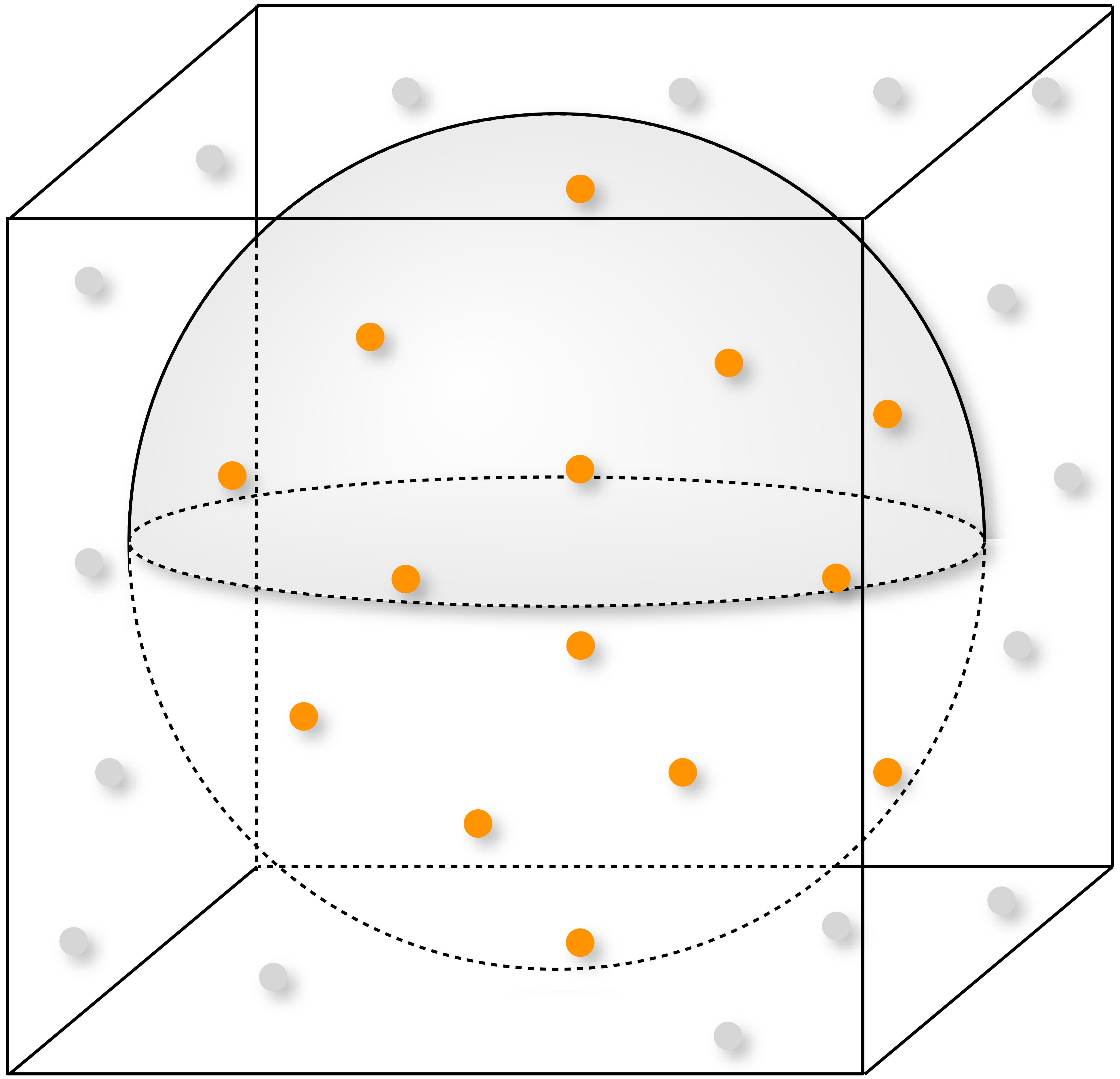
```
Vector3 v;
do
{
    v.x = 1-2*randf();
    v.y = 1-2*randf();
    v.z = 1-2*randf();
} while(dot(v,v) > 1)
```

Rejection Sampling on a Sphere



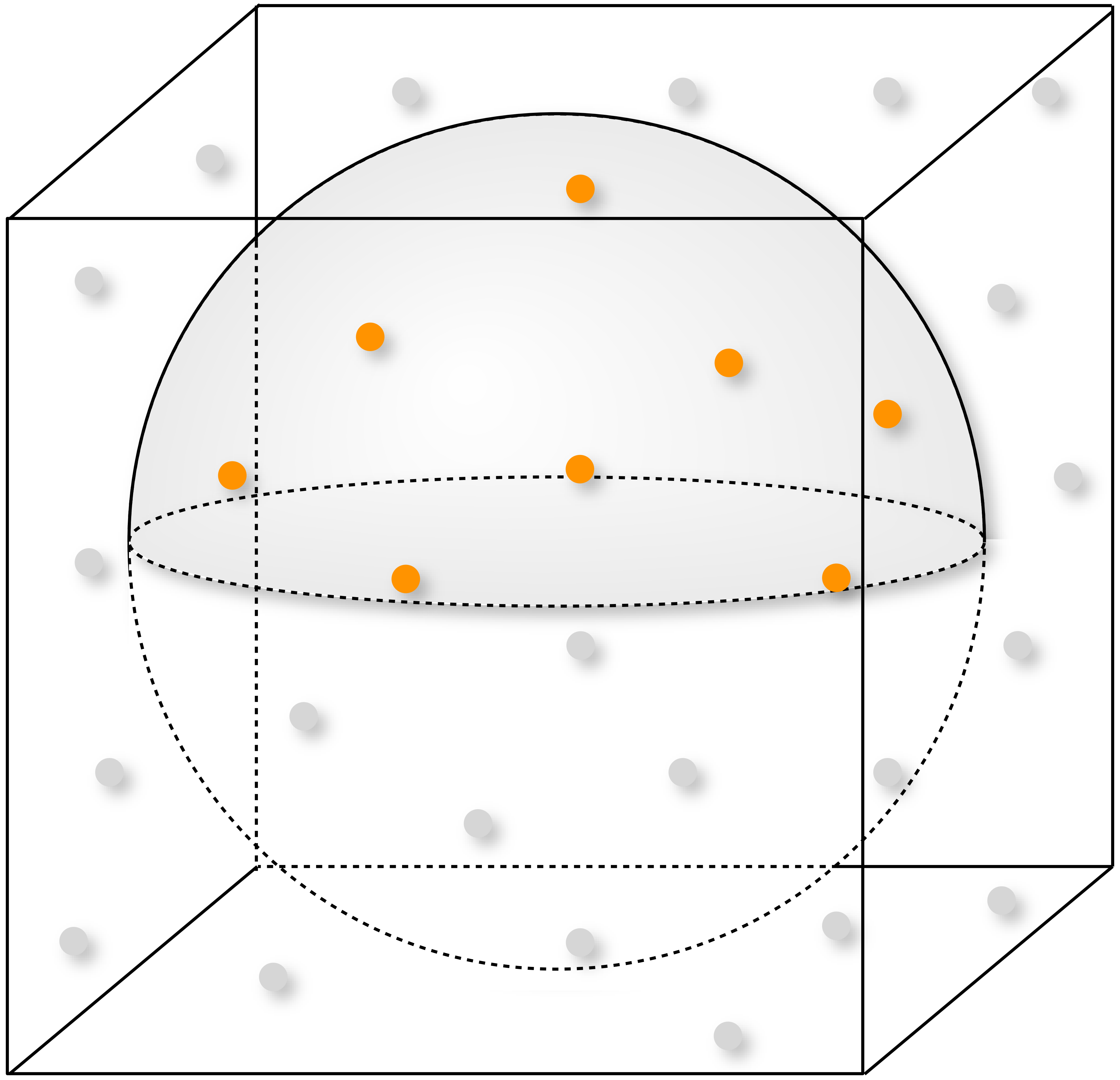
```
Vector3 v;
do
{
    v.x = 1-2*randf();
    v.y = 1-2*randf();
    v.z = 1-2*randf();
} while(dot(v,v) > 1)

// Project onto sphere
v = v/length(v);
```

Rejection Sampling a Hemisphere



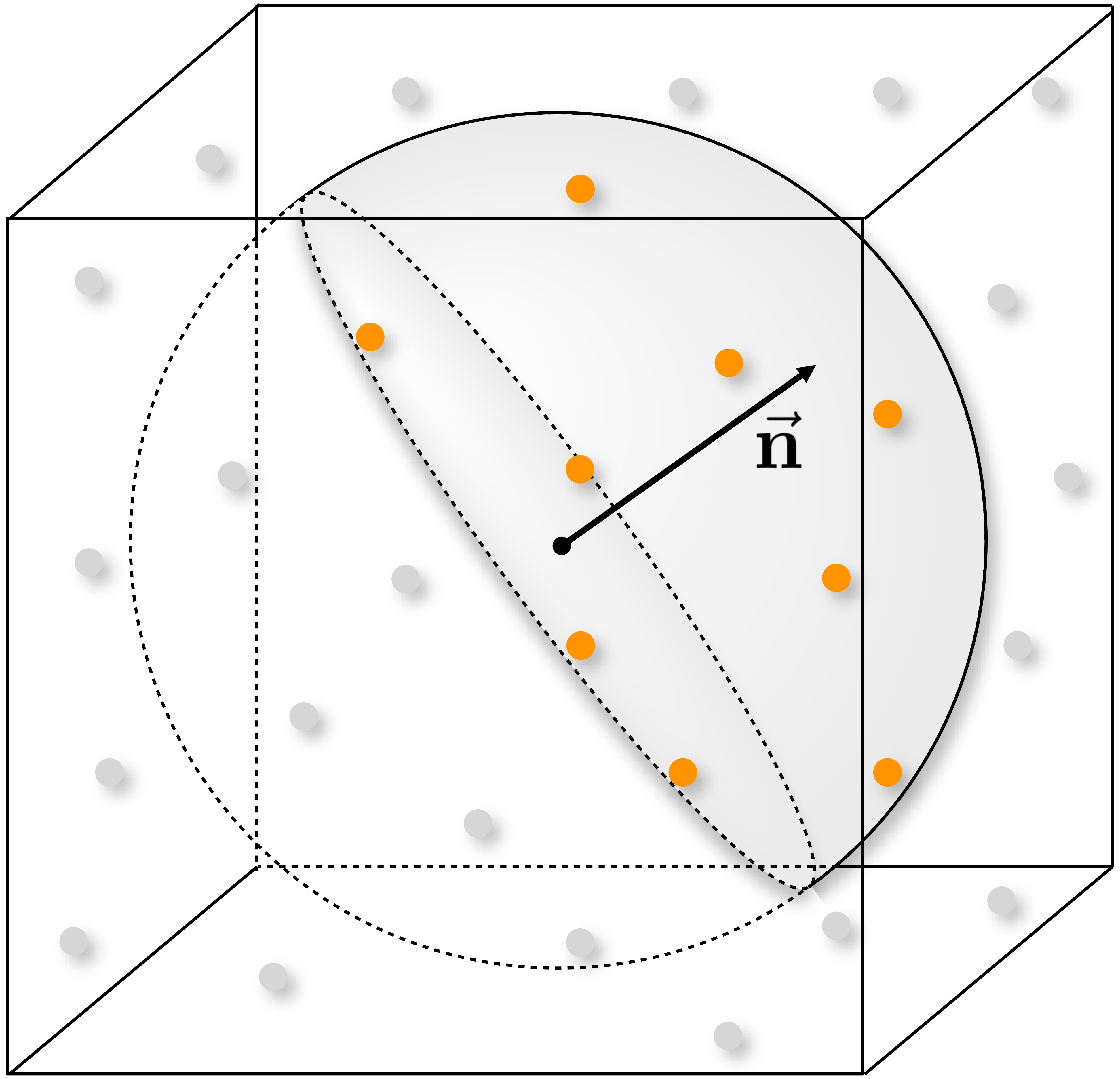
```
Vector3 v;
do
{
    v.x = 1-2*randf();
    v.y = 1-2*randf();
    v.z = 1-2*randf();
} while(dot(v,v) > 1)
```

Rejection Sampling a Hemisphere



```
Vector3 v;
do
{
   v.x = 1-2*randf();
   v.y = 1-2*randf();
   v.z = 1-2*randf();
} while(dot(v,v) > 1 ||
        v.z < 0)
```

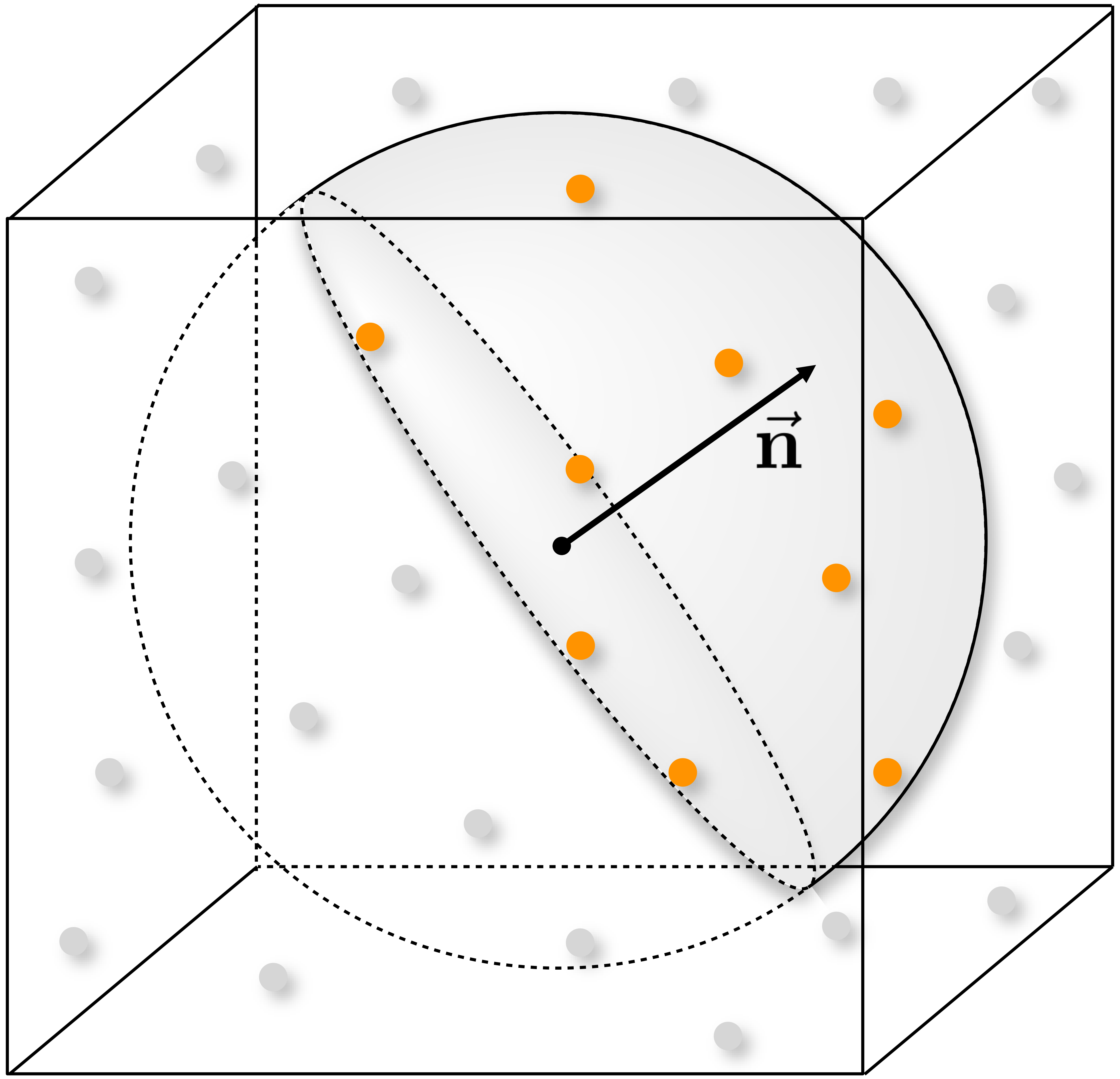
Rejection Sampling a Hemisphere




```
Vector3 v;
do
{
    v.x = 1-2*randf();
    v.y = 1-2*randf();
    v.z = 1-2*randf();
} while(dot(v,v) > 1 ||
          v.z < 0)
```

- Arbitrary orientation?

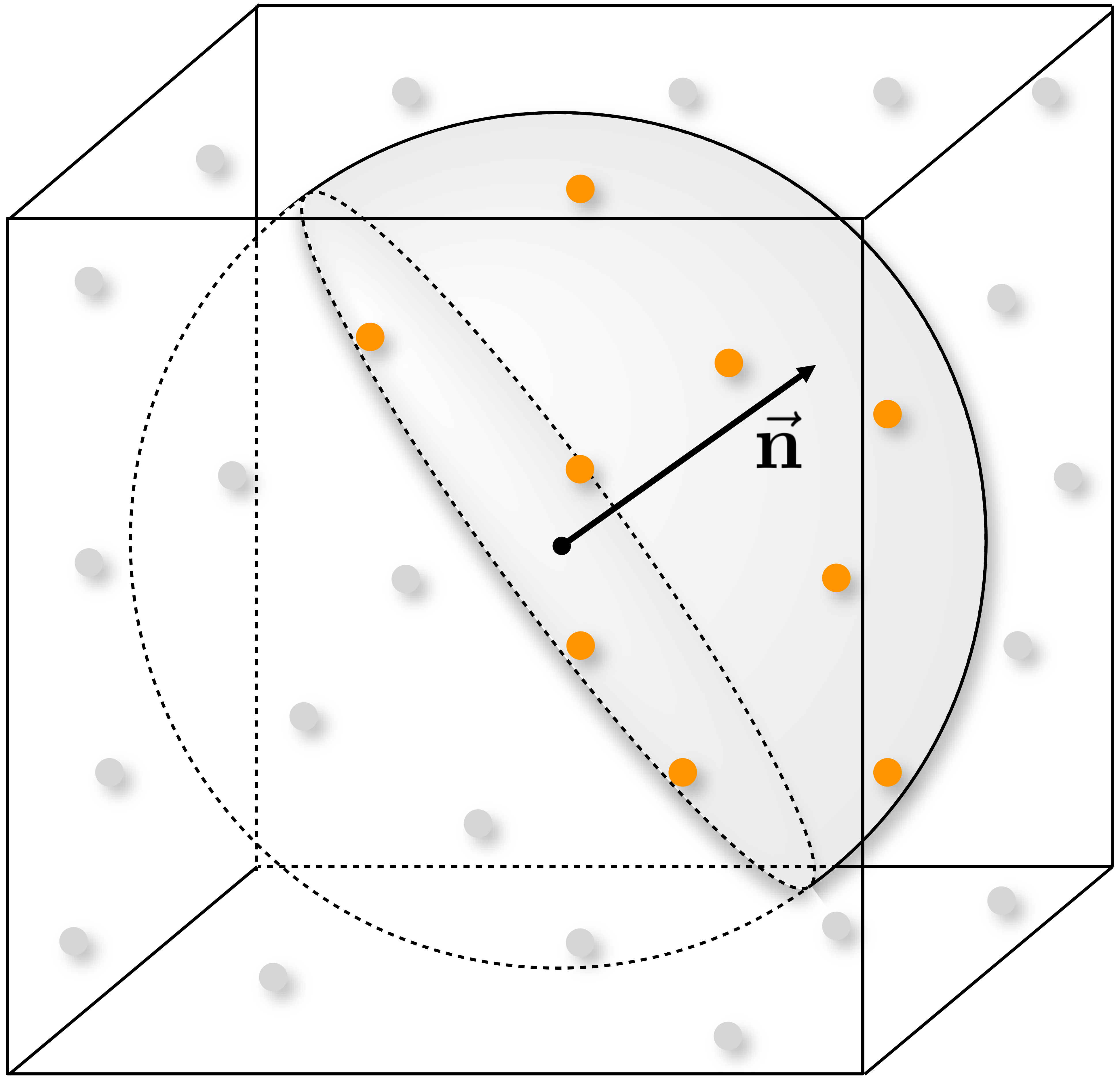
Rejection Sampling a Hemisphere




```
Vector3 v;
do
{
    v.x = 1-2*randf();
    v.y = 1-2*randf();
    v.z = 1-2*randf();
} while(dot(v,v) > 1 ||
        dot(v,n) < 0)
```

- Arbitrary orientation?

Rejection Sampling a Hemisphere



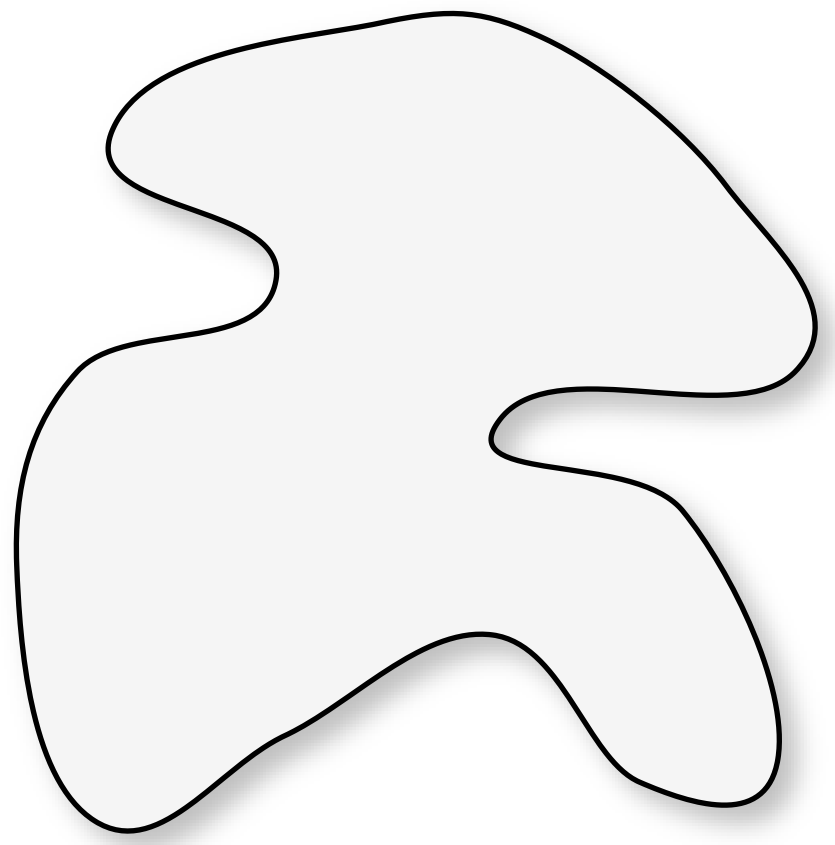
- Or, just generate in canonical orientation, and then rotate

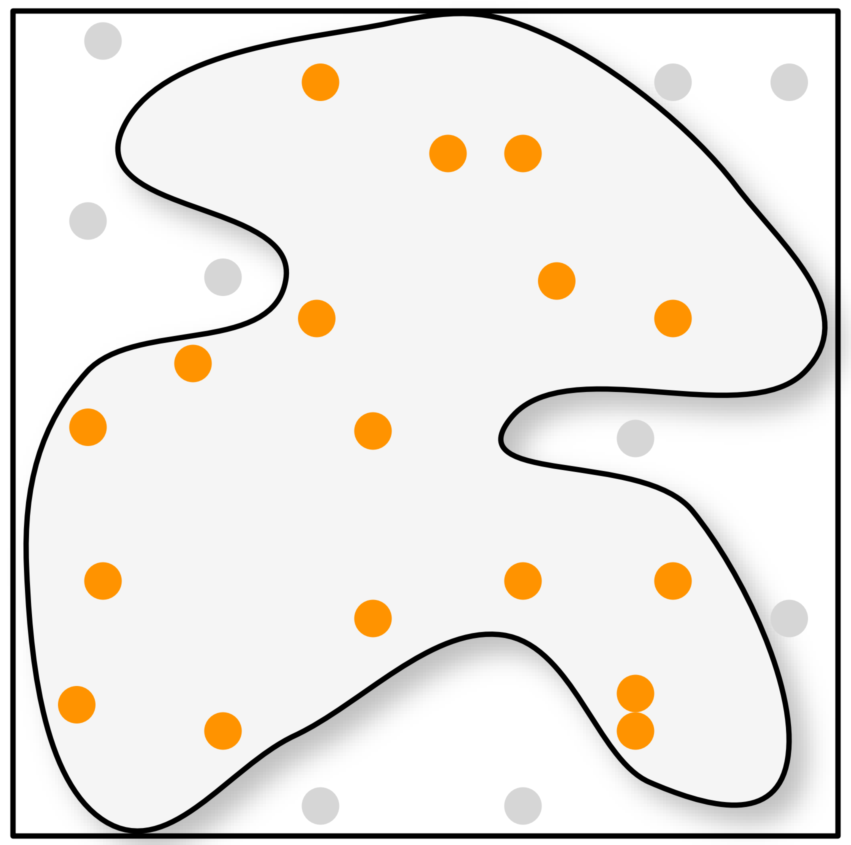
Rejection Sampling

More complex shapes

Pros:

- Flexible





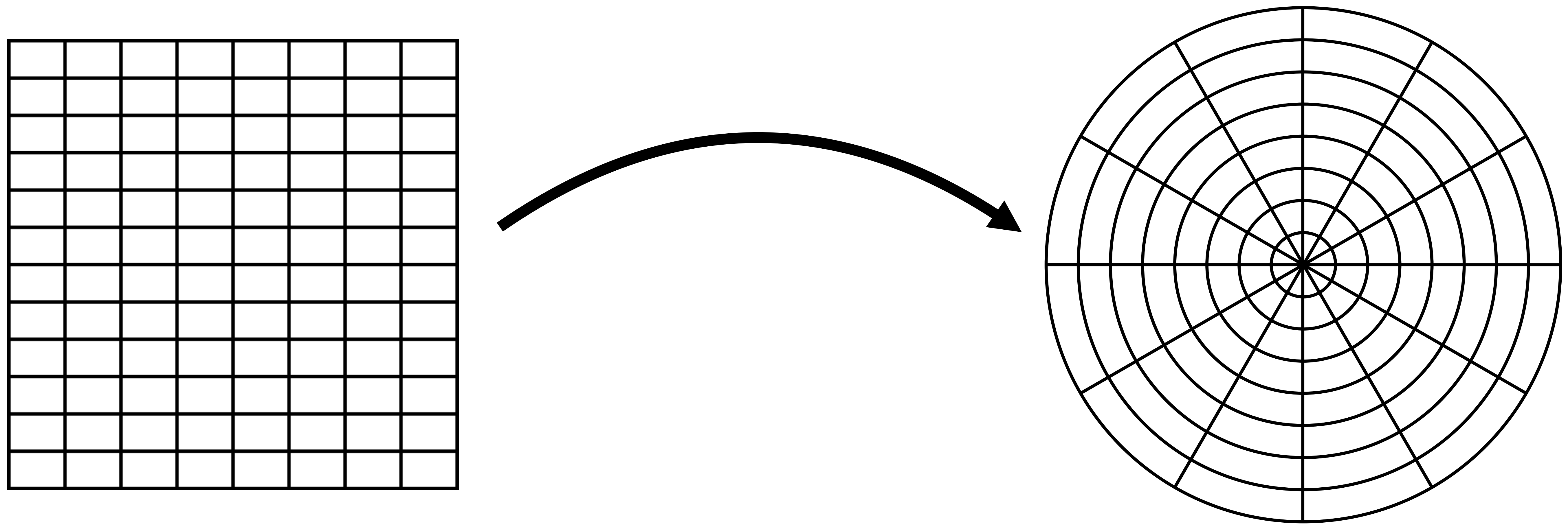
Cons:

- Inefficient
- Difficult/impossible to combine with stratification or quasi-Monte Carlo

Directly sampling a disk?

Idea: transform samples to polar coordinates:

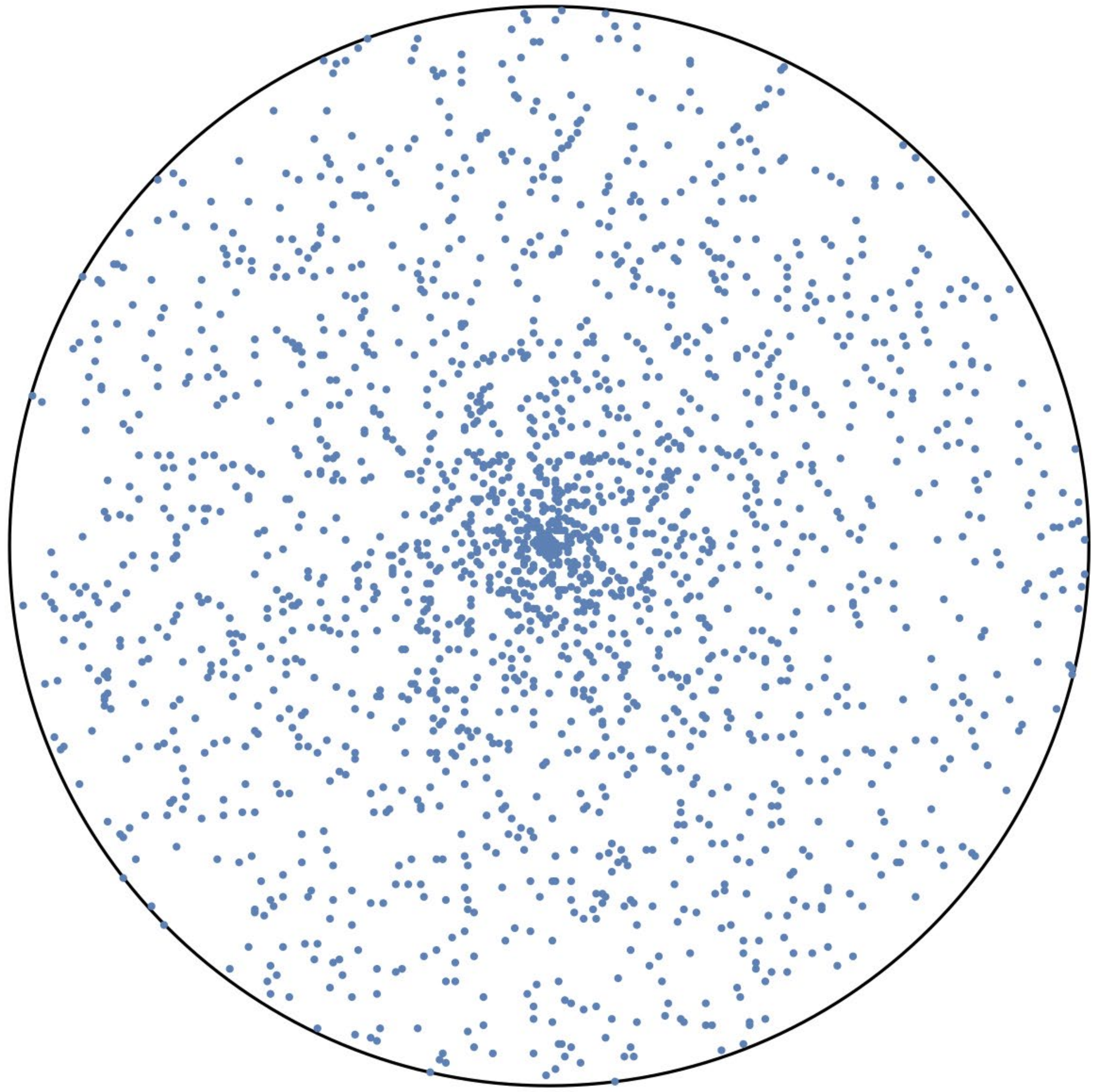
- pick two uniform random variables ξ_1, ξ_2
- select point at (r, ϕ) with $r = \xi_1$ and $\phi = 2\pi\xi_2$
- This algorithm **does not** produce the desired uniform sampling of the disk. Why?



not equi-areal

Wrong!

Samples are uniform in (θ, r), but non-uniform in (x,y)!



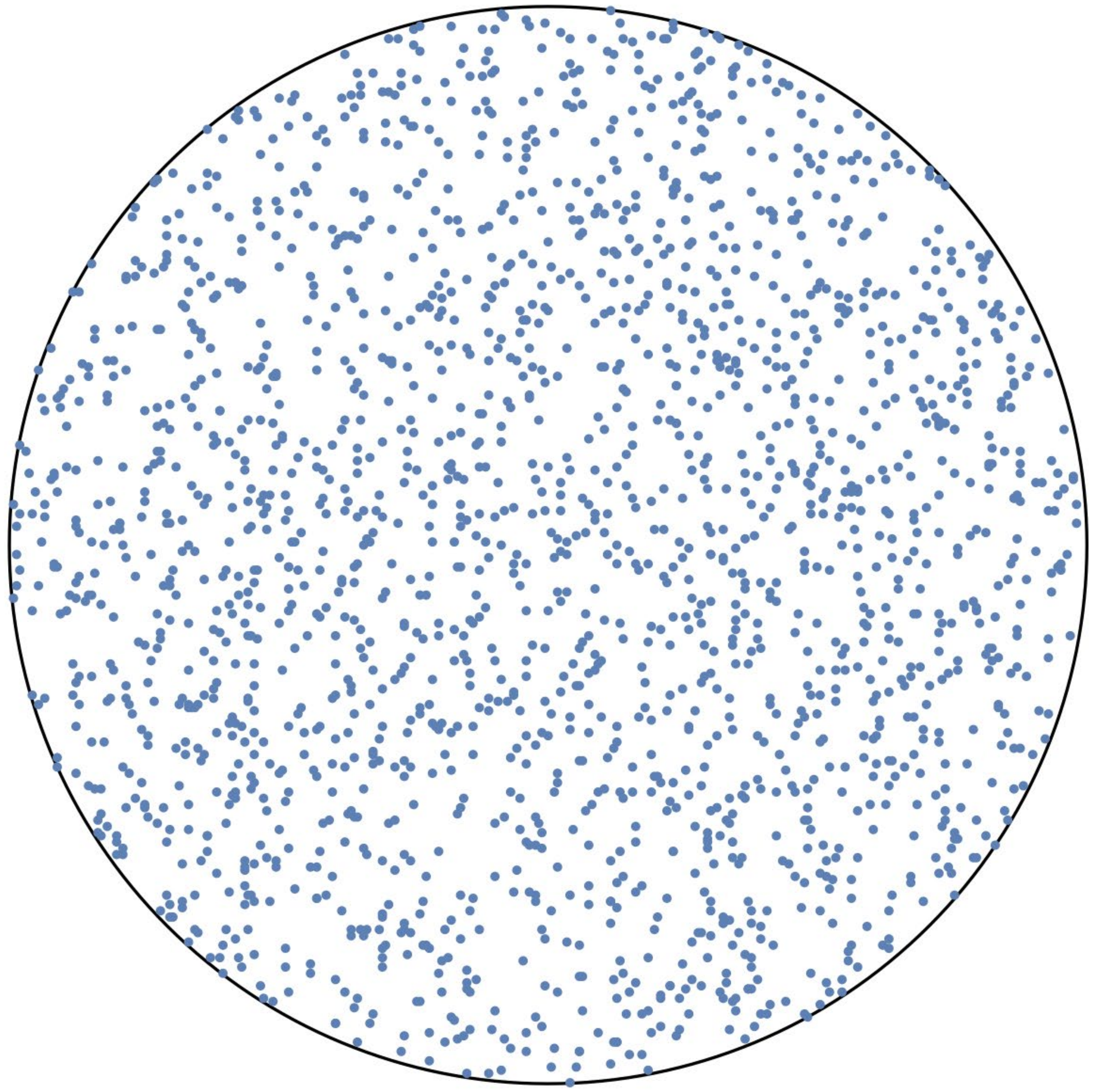
$$\theta = 2\pi\xi_1$$

$$r = \xi_2$$

This can be corrected by choosing r non-uniformly!

Right!

Samples are non-uniform in (θ, r), but uniform in (x,y)!



$$\theta = 2\pi\xi_1$$

$$r = \sqrt{\xi_2}$$

Transforming Between Distributions

Given a random variable $X_i \sim p(x)$

$Y_i = T(X_i)$ is also a random variable

- but what is its probability density?

$$p_y(y) = p_y(T(x)) = \frac{p_x(x)}{|J_T(x)|}$$

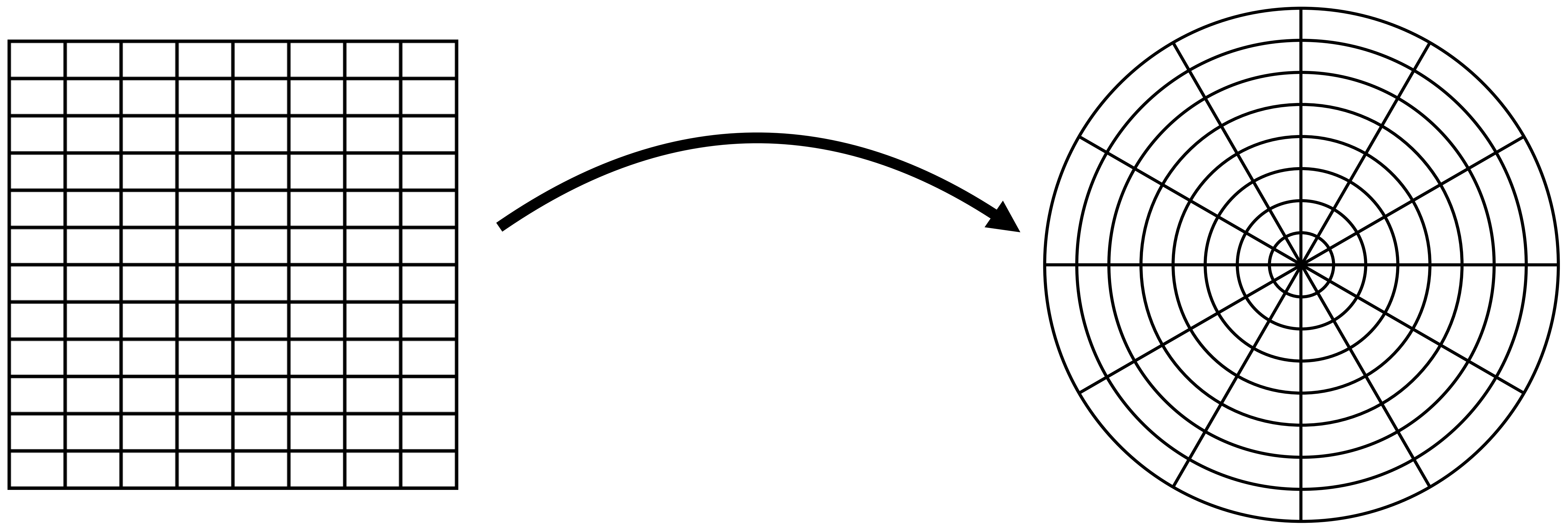
- where $|J_T(x)|$ is the absolute value of the determinant of the Jacobian of T

Polar coordinate parameterization

$$T(r, \phi) \mapsto \begin{bmatrix} r\cos\phi \\ r\sin\phi \end{bmatrix}$$

$$J_T(r,\phi) = \begin{bmatrix} \frac{\partial T_x}{\partial r} & \frac{\partial T_x}{\partial \phi} \\ \frac{\partial T_y}{\partial r} & \frac{\partial T_y}{\partial \phi} \end{bmatrix} = \begin{bmatrix} \cos\phi & -r\sin\phi \\ \sin\phi & r\cos\phi \end{bmatrix}$$

$$|\det J_T(r,\phi)| = r$$



Account for parameterization

Desired distribution on target domain

$$p(x,y) = \begin{cases} \frac{1}{\pi}, & x^2+y^2<1 \\ 0, & \text{otherwise} \end{cases}$$

If we sample in spherical coordinates:

$$\overbrace{p(x,y)}^{\text{target domain}} = p(T(r,\phi)) = \frac{\overbrace{p(r,\phi)}^{\text{sampling domain}}}{|\det J_T(r,\phi)|}$$

Thus, need this distribution on source domain:

$$p(r,\phi) = \underbrace{p(T(r,\phi))}_{=1/\pi} \cdot \underbrace{|\det J_T(r,\phi)|}_{=r} = \frac{r}{\pi}$$

Sampling 2D Distributions

Draw samples (X, Y) from a 2D distribution $p(x, y)$

If $p(x, y)$ is separable, i.e., $p(x, y) = p(x)\, p(y)$, we can independently sample $p(x)$, and $p(y)$

Otherwise, compute the marginal density function:

$$p(x) = \int p(x,y)\, dy$$

and, the conditional density:

$$p(y \mid x) = \frac{p(x,y)}{p(x)}$$

Procedure: first sample $X_i \sim p(x)$, then $Y_i \sim p(y \mid X_i)$

Account for parameterization

Thus: need this distribution on source domain

$$p(r,\phi) = \underbrace{p(T(r,\phi))}_{=1/\pi} \cdot \underbrace{|\det J_T(r,\phi)|}_{=r} = \frac{r}{\pi}$$

Step 1: generate φ proportional to

$$p_1(\phi) = \frac{1}{2\pi} \quad (\phi \in [0, 2\pi])$$

Step 2: generate r proportional to

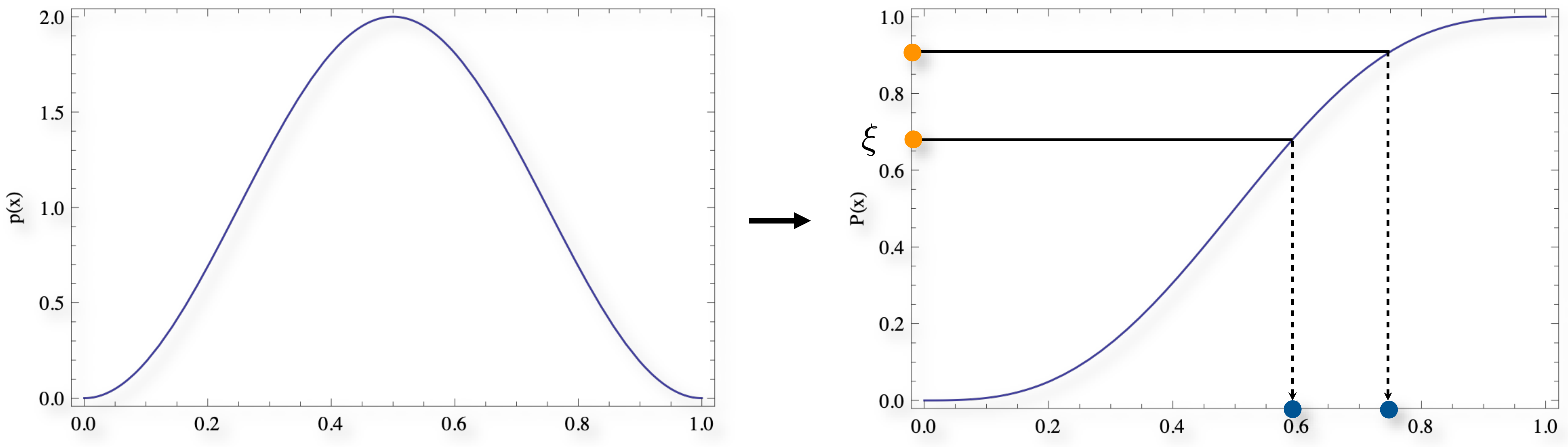
$$p_2(r) \propto r = 2r \quad (r \in [0, 1])$$

Constant PDF in φ, linearly increasing PDF in r

Sampling arbitrary distributions

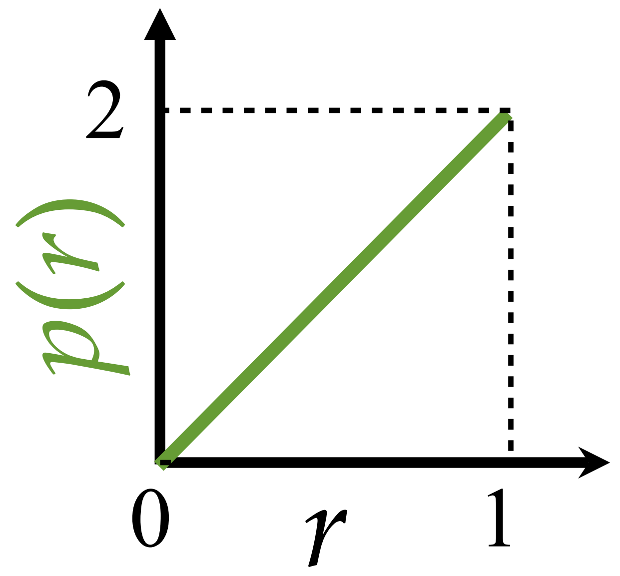
The inversion method:

1. Compute the CDF $P(x) = \int_0^x p(x')\, dx'$
2. Compute its inverse $P^{-1}(y)$
3. Obtain a uniformly distributed random number ξ
4. Compute $X_i = P^{-1}(\xi)$



Sampling a linear ramp

Goal: sample with PDF: $$p(r) = 2r$$



Step 1: $$P(r) = r^2$$

Step 2: $$P^{-1}(y) = \sqrt{y}$$

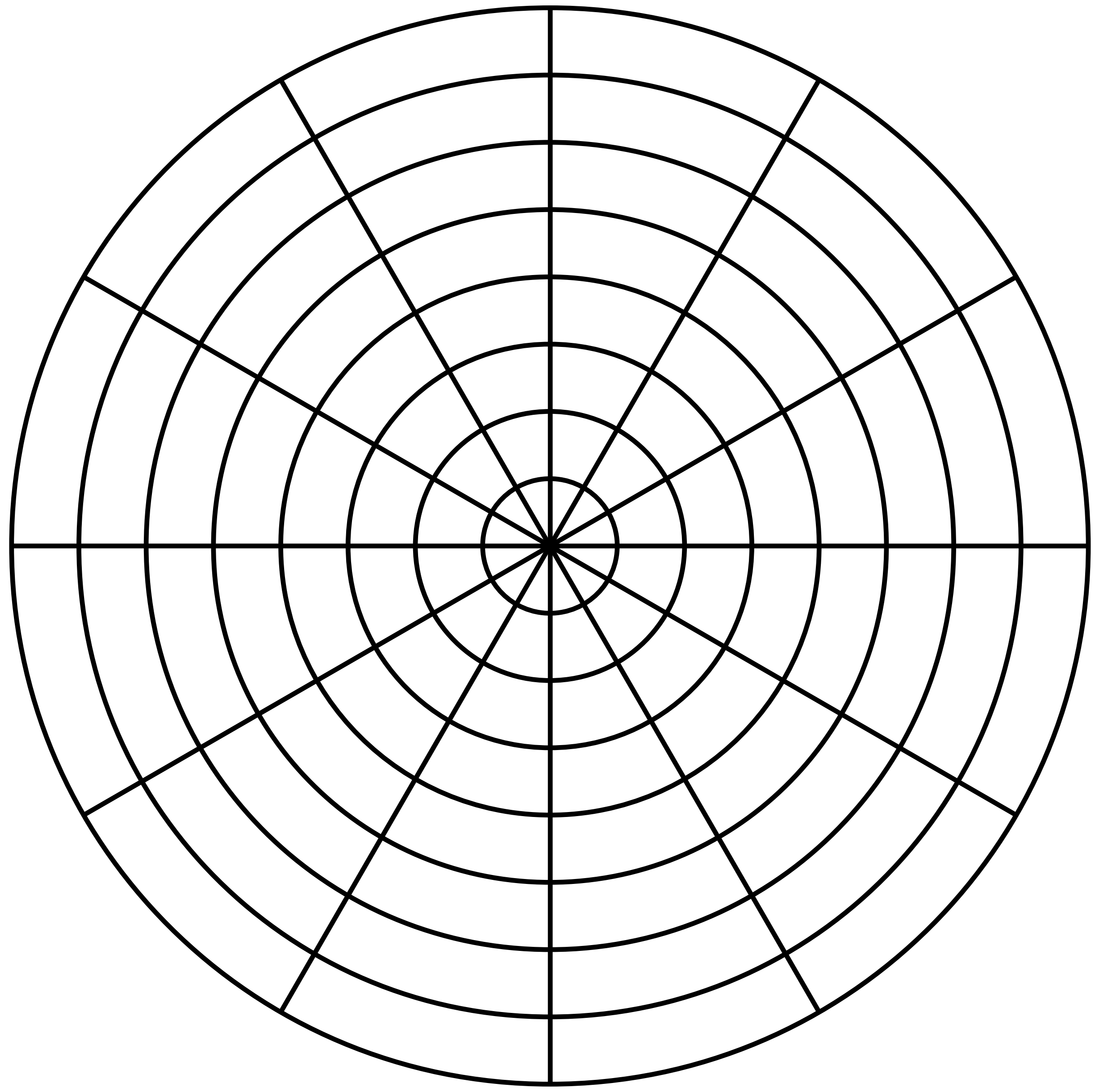
Step 3: $$r_i = \sqrt{\xi}$$

Uniformly Sampling a Disk

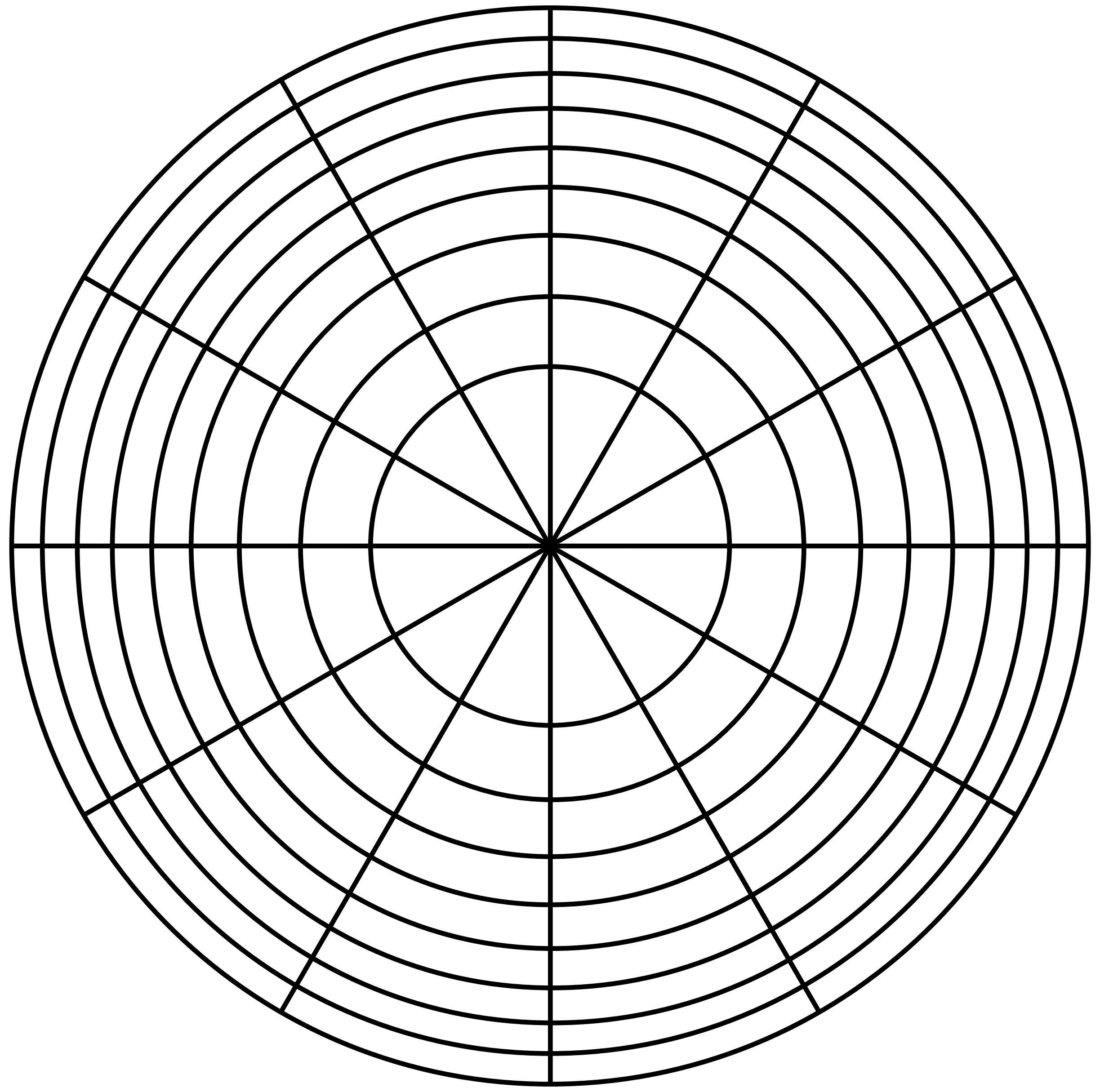
Pick two uniform random variables ξ_1, ξ_2

Sample in polar coordinates with:

$$(r, \phi) = (\xi_1,\ 2\pi\xi_2) \qquad (r, \phi) = \left(\sqrt{\xi_1},\ 2\pi\xi_2\right)$$



not equi-area



equi-area

Recipe

1. Express the desired distribution in a convenient coordinate system
2. Account for distortion by coordinate system
   - Requires computing the determinant of the Jacobian
3. Compute marginal and conditional 1D PDFs
4. Sample 1D PDFs using the inversion method

Directly Sampling on a Sphere

Can we use this?

Given a random variable $X_i \sim p(x)$

$Y_i = T(X_i)$ is also a random variable

- but what is its probability density?

$$p_y(y) = p_y(T(x)) = \frac{p_x(x)}{|J_T(x)|}$$

- where $|J_T(x)|$ is the absolute value of the determinant of the Jacobian of T

Directly Sampling on a Sphere

Different transformation rule:

$$p_x(\boldsymbol{x}(u,v)) = \frac{p_{(u,v)}(u,v)}{\|\boldsymbol{x}_u(u,v) \times \boldsymbol{x}_v(u,v)\|}$$

Where does this come from?

- Expression for differential area (e.g., as in area integral):

$$\mathrm{d}A(\boldsymbol{x}) = \|\boldsymbol{x}_u(u,v) \times \boldsymbol{x}_v(u,v)\| \mathrm{d}u\mathrm{d}v$$

Directly Sampling on a Sphere

Pick two uniform random variables ξ_1, ξ_2

Idea: select point at (θ, φ) with $\theta = \pi\xi_1$ and $\varphi = 2\pi\xi_2$

- **Problem**: not uniform with respect to surface area!

Correct solution: $\theta = \cos^{-1}(2\xi_1 - 1)$ and $\varphi = 2\pi\xi_2$

Algorithm

$$
\begin{aligned}
\theta &= \cos^{-1}(2\xi_1 - 1) \\
\phi &= 2\pi\xi_2 \\
\vec{\omega}_x &= \sin\theta\cos\phi \\
\vec{\omega}_y &= \sin\theta\sin\phi \\
\vec{\omega}_z &= \cos\theta
\end{aligned}
$$

→

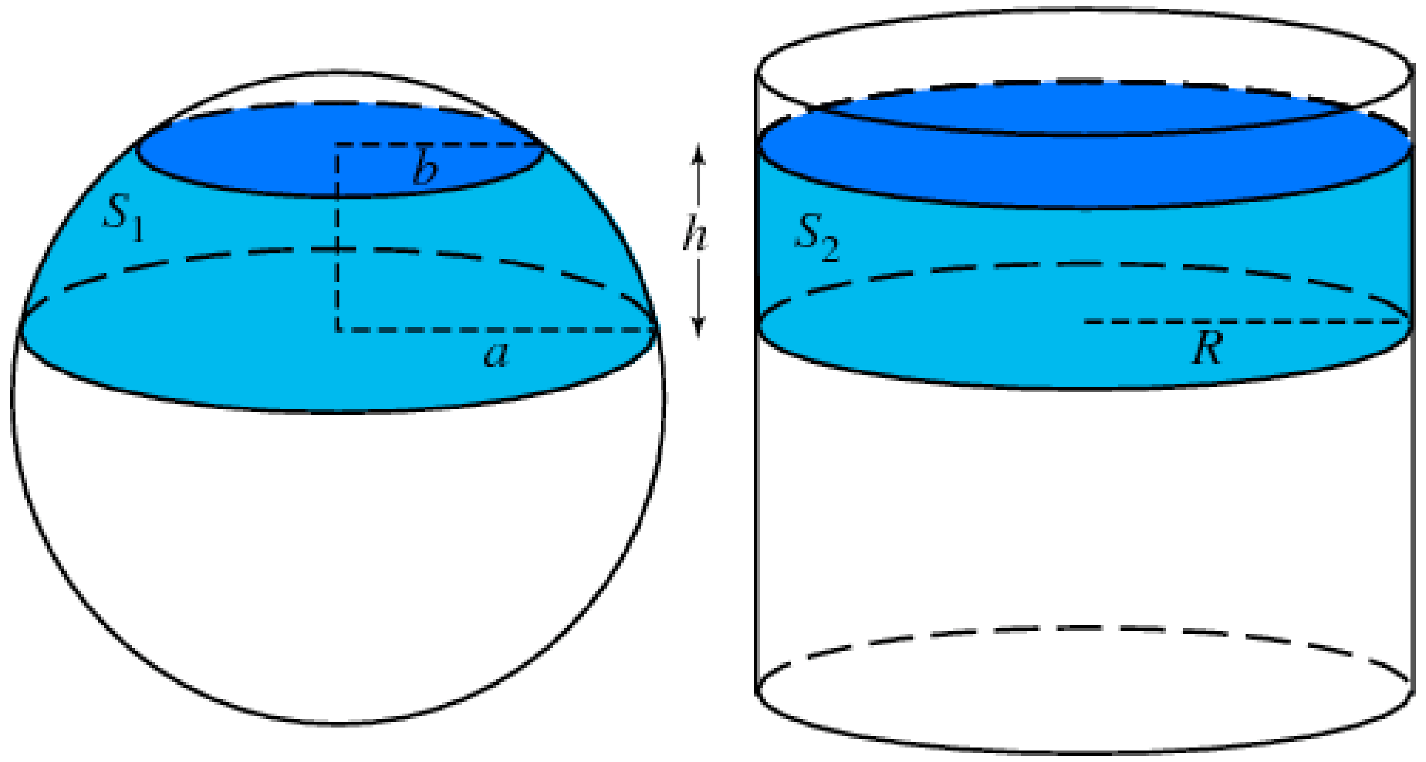
Better

$$
\begin{aligned}
\vec{\omega}_z &= 2\xi_1 - 1 \\
r &= \sqrt{1 - \vec{\omega}_z^2} \\
\phi &= 2\pi\xi_2 \\
\vec{\omega}_x &= r\cos\phi \\
\vec{\omega}_y &= r\sin\phi
\end{aligned}
$$

Archimedes' Hat-Box Theorem

The surface area of a sphere between any two horizontal planes is equal to the corresponding area on the circumscribing cylinder.

- i.e.: uniform areas on a cylinder map to uniform areas on a sphere
- What is $|J_T|$ for cylindrical mapping?



Weisstein, Eric W. "Archimedes' Hat-Box Theorem." From MathWorld--A Wolfram Web Resource. http://mathworld.wolfram.com/ArchimedesHat-BoxTheorem.html

Archimedes' Hat-Box Theorem

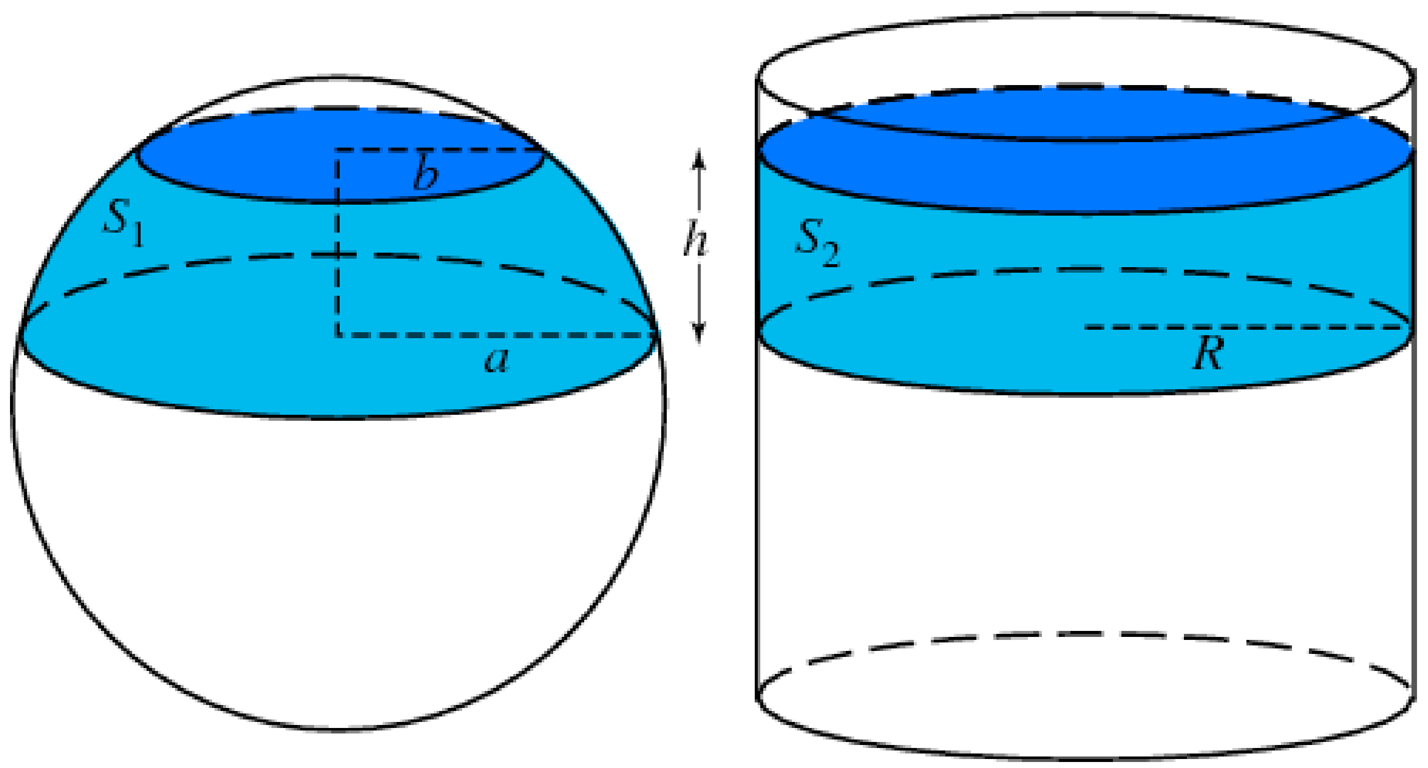
The surface area of a sphere between any two horizontal planes is equal to the corresponding area on the circumscribing cylinder.

- i.e.: uniform areas on a cylinder map to uniform areas on a sphere

- What is $|J_T|$ for cylindrical mapping?

$$\vec{\omega}_z = 2\xi_1 - 1$$
$$r = \sqrt{1 - \vec{\omega}_z^2}$$
$$\phi = 2\pi\xi_2$$
$$\vec{\omega}_x = r\cos\phi$$
$$\vec{\omega}_y = r\sin\phi$$

- point on unit cylinder
- projection onto sphere



Weisstein, Eric W. "Archimedes' Hat-Box Theorem." From MathWorld--A Wolfram Web Resource. http://mathworld.wolfram.com/ArchimedesHat-BoxTheorem.html

Directly Sampling a Hemisphere

Just like a sphere

Use Hat-Box theorem with shorter cylinder

More Random Sampling

Other useful sampling domains:

- triangles
- 1- or 2-D discrete PDFs (e.g. environment maps)

Much more!

Sampling Various Distributions

Target space	Density	Domain	Transformation
Radius R disk	$p(r,\theta) = \frac{1}{\pi R^2}$	$\theta \in [0, 2\pi]$ $r \in [0, R]$	$\theta = 2\pi u$ $r = R\sqrt{v}$
Sector of radius R disk	$p(r,\theta) = \frac{2}{(\theta_2 - \theta_1)(r_2^2 - r_1^2)}$	$\theta \in [\theta_1, \theta_2]$ $r \in [r_1, r_2]$	$\theta = \theta_1 + u(\theta_2 - \theta_1)$ $r = \sqrt{r_1^2 + v(r_2^2 - r_1^2)}$
Phong density exponent n	$p(\theta,\phi) = \frac{n+1}{2\pi}\cos^n\theta$	$\theta \in \left[0, \frac{\pi}{2}\right]$ $\phi \in [0, 2\pi]$	$\theta = \arccos\left((1-u)^{1/(n+1)}\right)$ $\phi = 2\pi v$
Separated triangle filter	$p(x,y)(1-\lvert x\rvert)(1-\lvert y\rvert)$	$x \in [-1, 1]$ $y \in [-1, 1]$	$x = \begin{cases} 1 - \sqrt{2(1-u)} & \text{if } u \geq 0.5 \\ -1 + \sqrt{2u} & \text{if } u < 0.5 \end{cases}$ $y = \begin{cases} 1 - \sqrt{2(1-v)} & \text{if } v \geq 0.5 \\ -1 + \sqrt{2v} & \text{if } v < 0.5 \end{cases}$
Triangle with vertices a_0, a_1, a_2	$p(a) = \frac{1}{\text{area}}$	$s \in [0, 1]$ $t \in [0, 1-s]$	$s = 1 - \sqrt{1-u}$ $t = (1-s)v$ $a = a_0 + s(a_1 - a_0) + t(a_2 - a_0)$
Surface of unit sphere	$p(\theta,\phi) = \frac{1}{4\pi}$	$\theta \in [0, \pi]$ $\phi \in [0, 2\pi]$	$\theta = \arccos(1 - 2u)$ $\phi = 2\pi v$
Sector on surface of unit sphere	$p(\theta,\phi) = \frac{1}{(\phi_2 - \phi_1)(\cos\theta_1 - \cos\theta_2)}$	$\theta \in [\theta_1, \theta_2]$ $\phi \in [\phi_1, \phi_2]$	$\theta = \arccos[\cos\theta_1 + u(\cos\theta_2 - \cos\theta_1)]$ $\phi = \phi_1 + v(\phi_2 - \phi_1)$
Interior of radius R sphere	$p = \frac{3}{4\pi R^3}$	$\theta \in [0, \pi]$ $\phi \in [0, 2\pi]$ $R \in [0, R]$	$\theta = \arccos(1 - 2u)$ $\phi = 2\pi v$ $r = w^{1/3}R$

[a] The symbols u, v, and w represent instances of uniformly distributed random variables ranging over [0, 1].

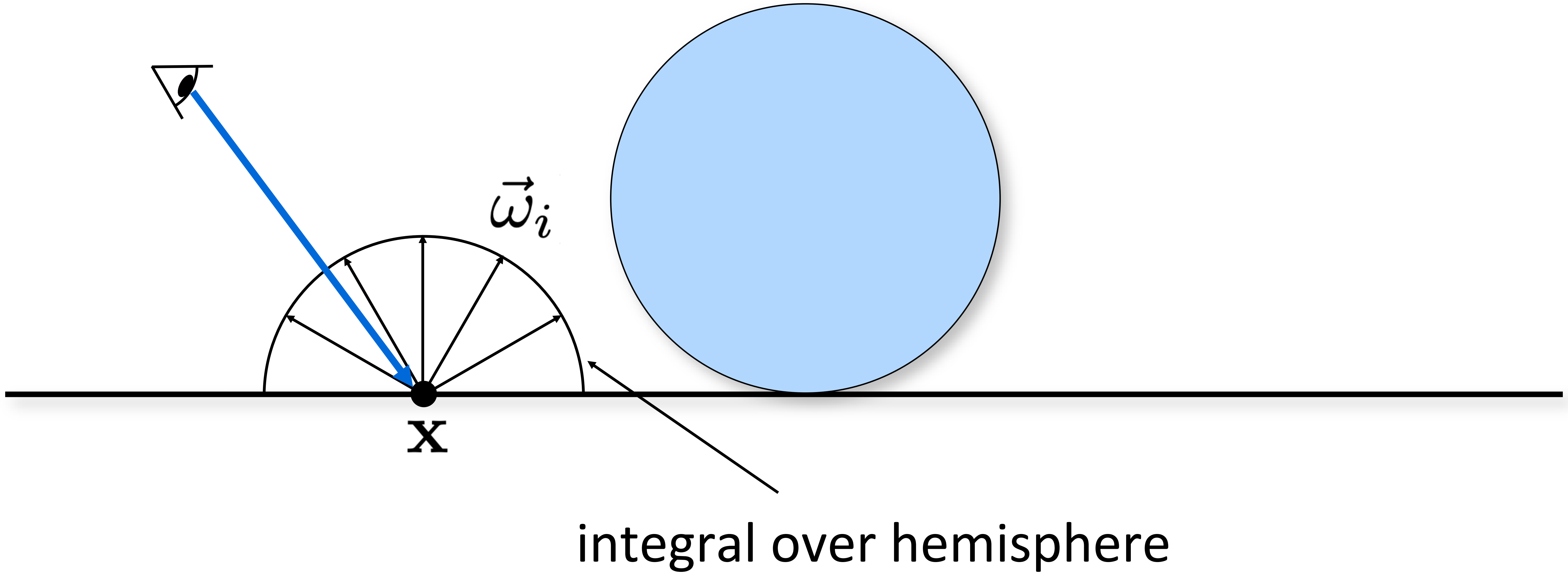
from: Peter Shirley. "Nonuniform random point sets via warping." Graphics Gems III, 1992.

Ambient Occlusion

Consider diffuse objects illuminated by an ambient overcast sky

$$L_r(\mathbf{x},\vec{\omega}_r) = \int_{H^2} f_r(\mathbf{x},\vec{\omega}_i,\vec{\omega}_r)\, L_i(\mathbf{x},\vec{\omega}_i)\cos\theta_i\, d\vec{\omega}_i$$

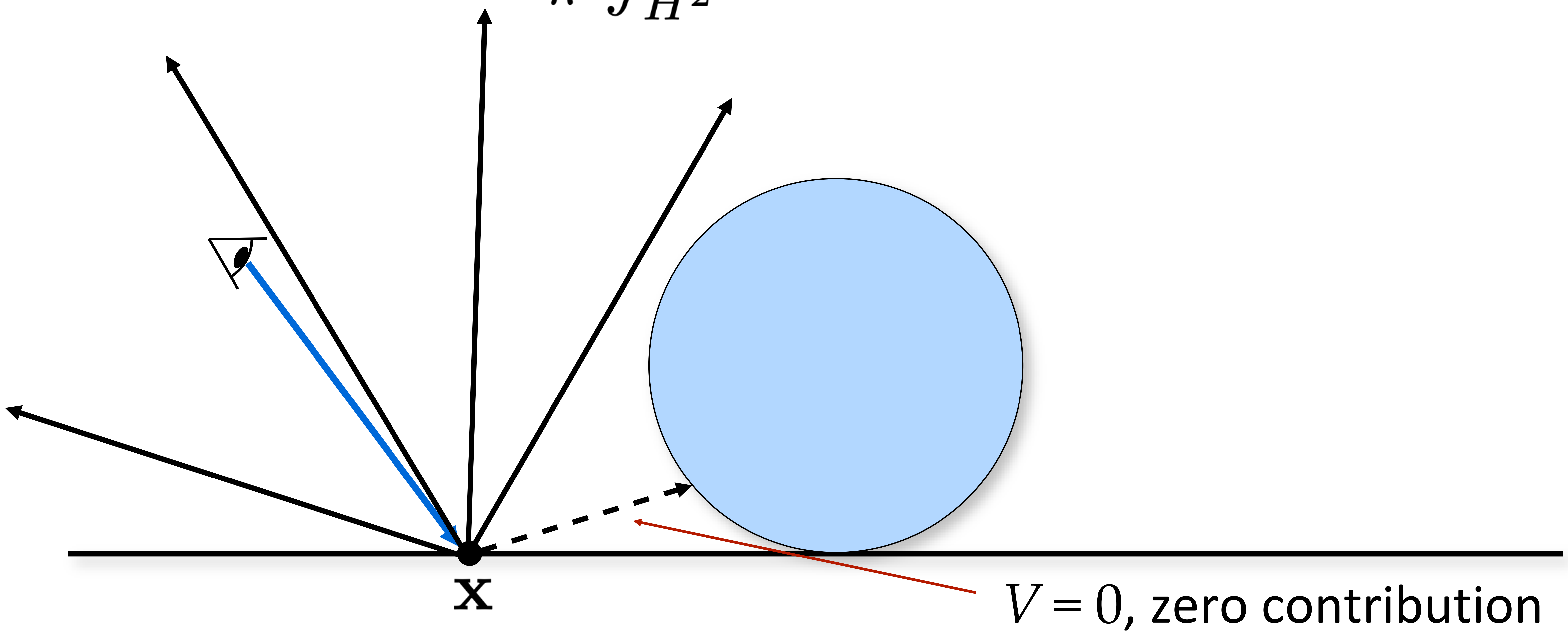
$$L_r(\mathbf{x}) = \frac{\rho}{\pi}\int_{H^2} V(\mathbf{x},\vec{\omega}_i)\cos\theta_i\, d\vec{\omega}_i$$



Ambient Occlusion

Consider diffuse objects illuminated by an ambient overcast sky

$$L_r(\mathbf{x}) = \frac{\rho}{\pi} \int_{H^2} V(\mathbf{x}, \vec{\omega}_i) \cos\theta_i \, d\vec{\omega}_i$$



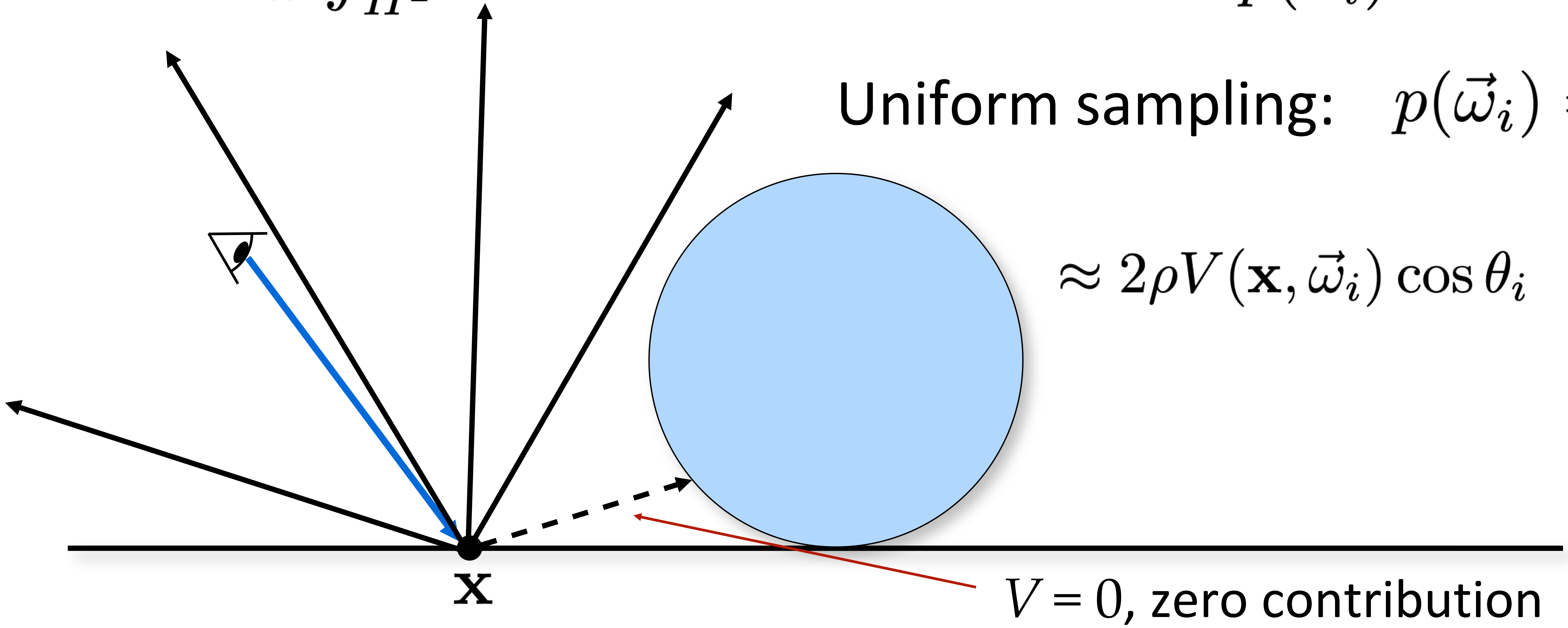
Ambient Occlusion

Consider diffuse objects illuminated by an ambient overcast sky

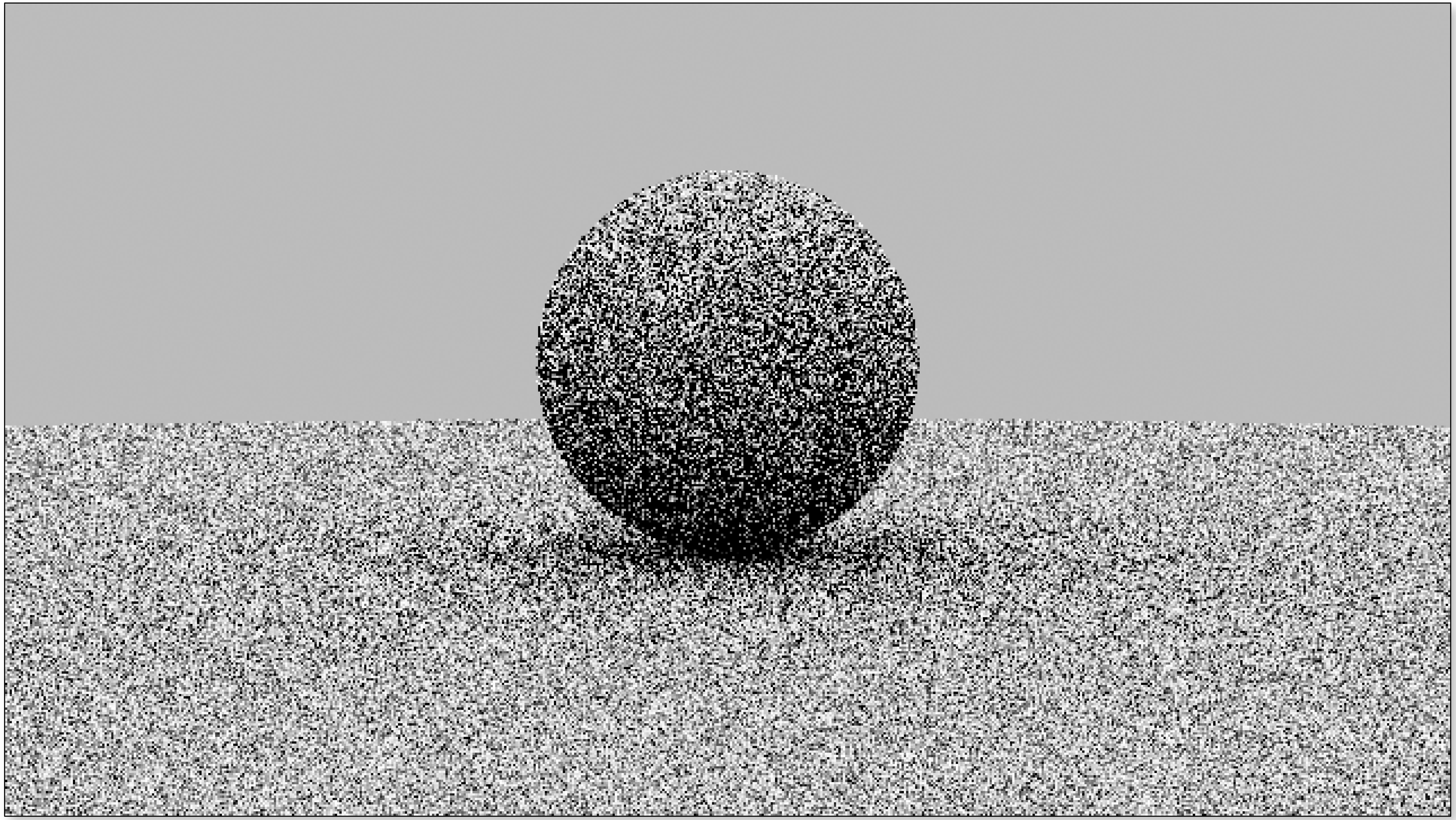
$$L_r(\mathbf{x}) = \frac{\rho}{\pi} \int_{H^2} V(\mathbf{x}, \vec{\omega}_i) \cos\theta_i \, d\vec{\omega}_i \approx \frac{\rho}{\pi} \frac{V(\mathbf{x}, \vec{\omega}_i) \cos\theta_i}{p(\vec{\omega}_i)}$$

Uniform sampling: $p(\vec{\omega}_i) = \frac{1}{2\pi}$

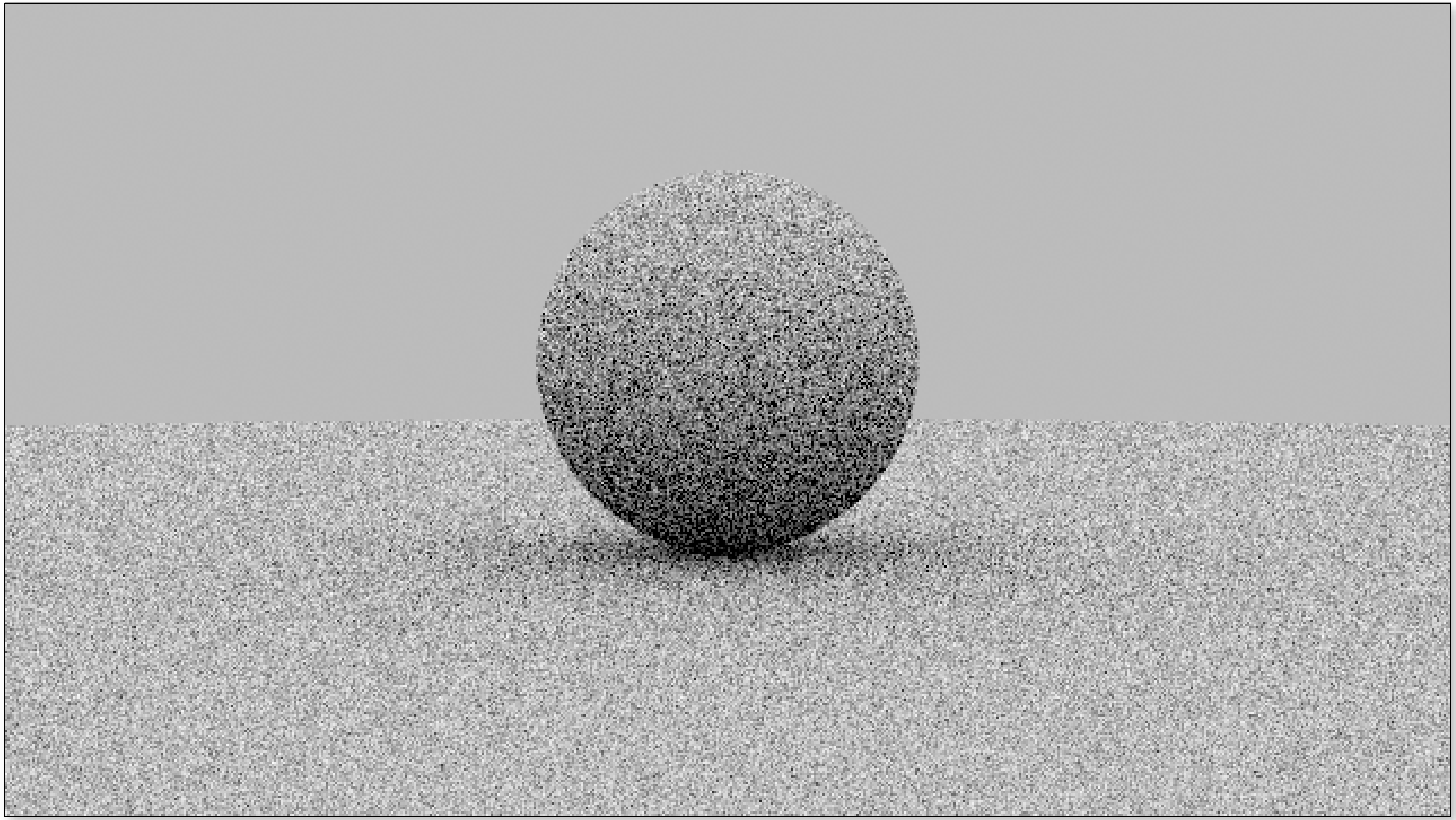
$$\approx 2\rho V(\mathbf{x}, \vec{\omega}_i) \cos\theta_i$$



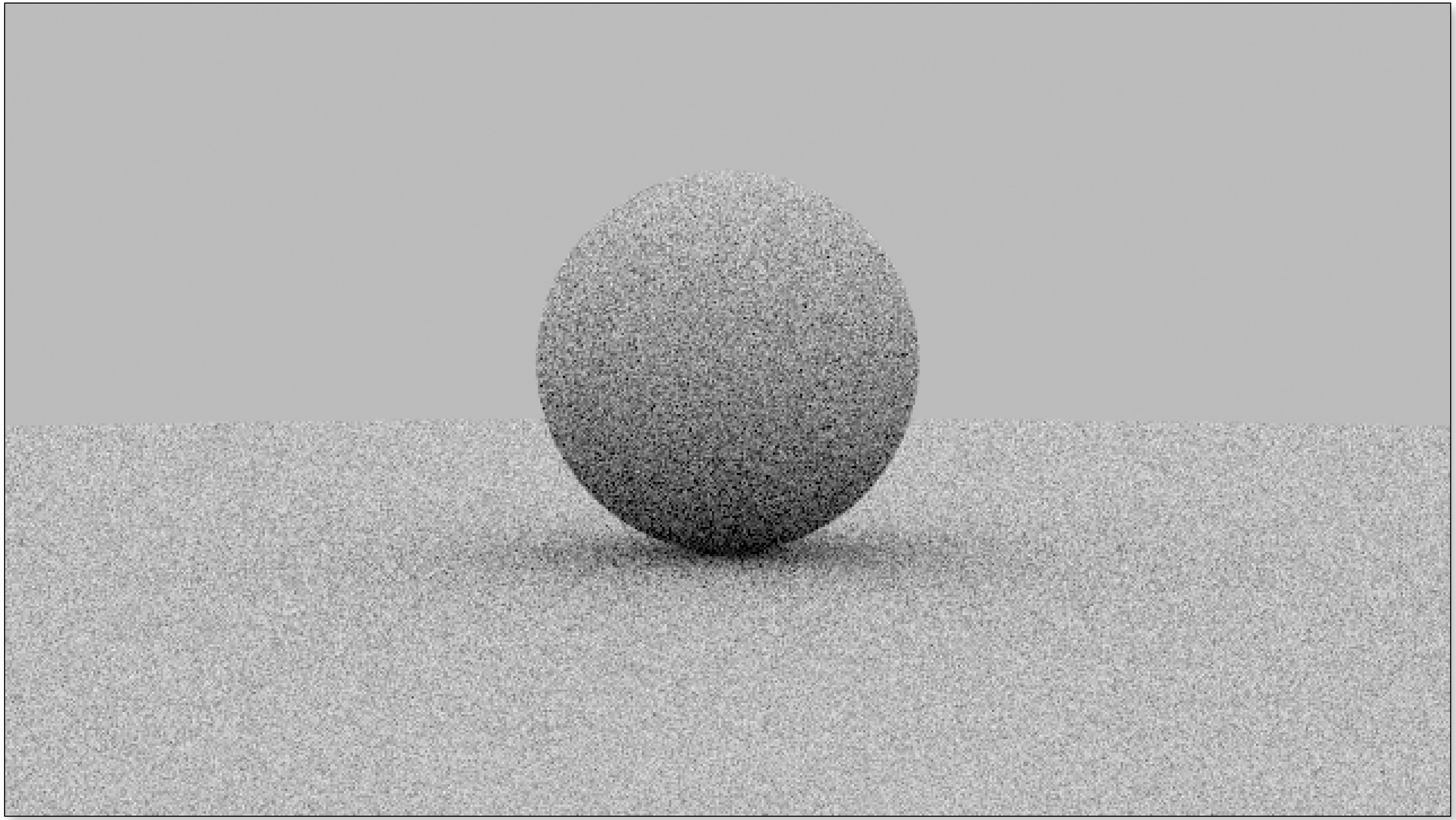
Hemispherical Sampling (1 Sample)



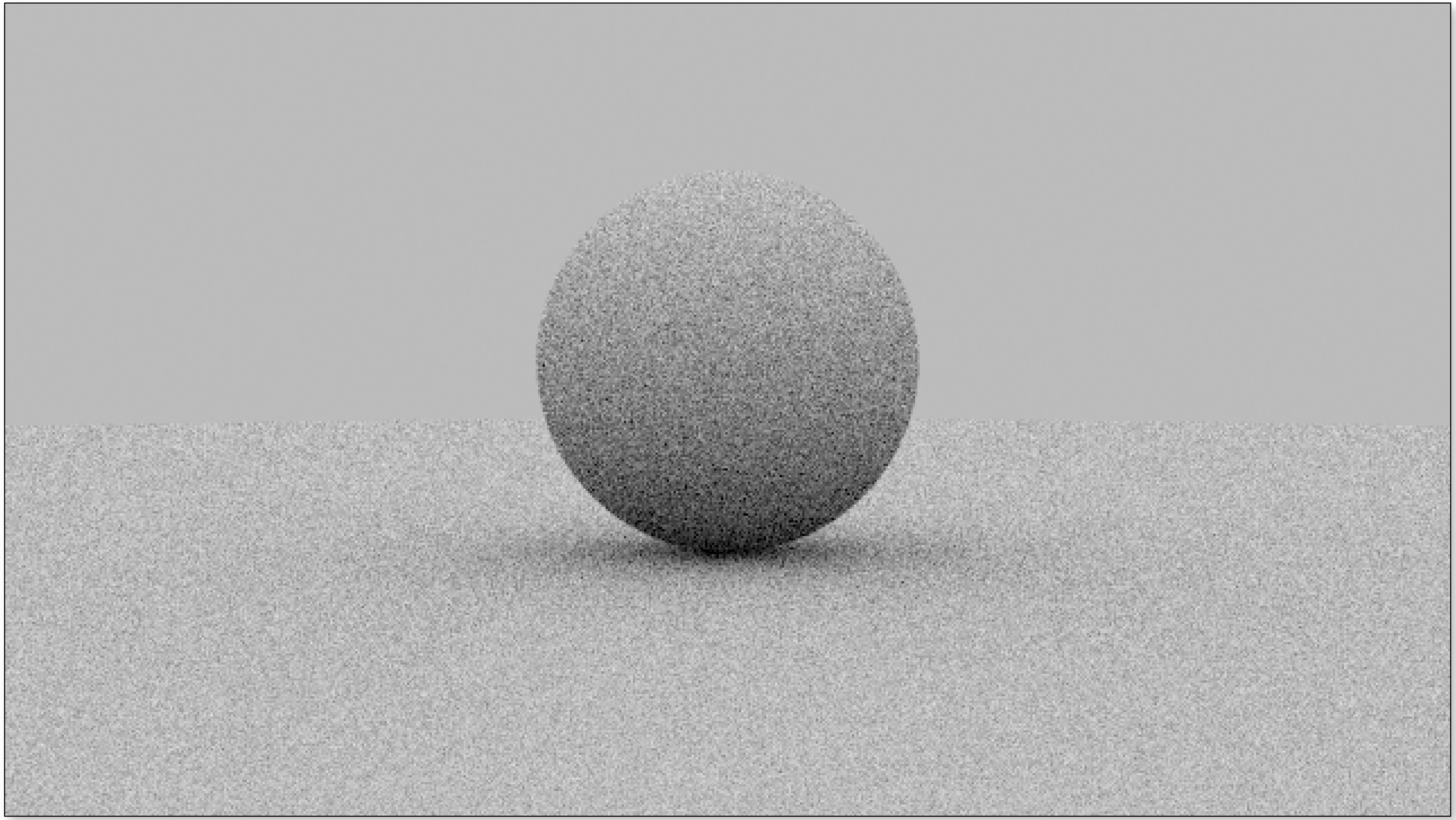
Hemispherical Sampling (4 Samples)



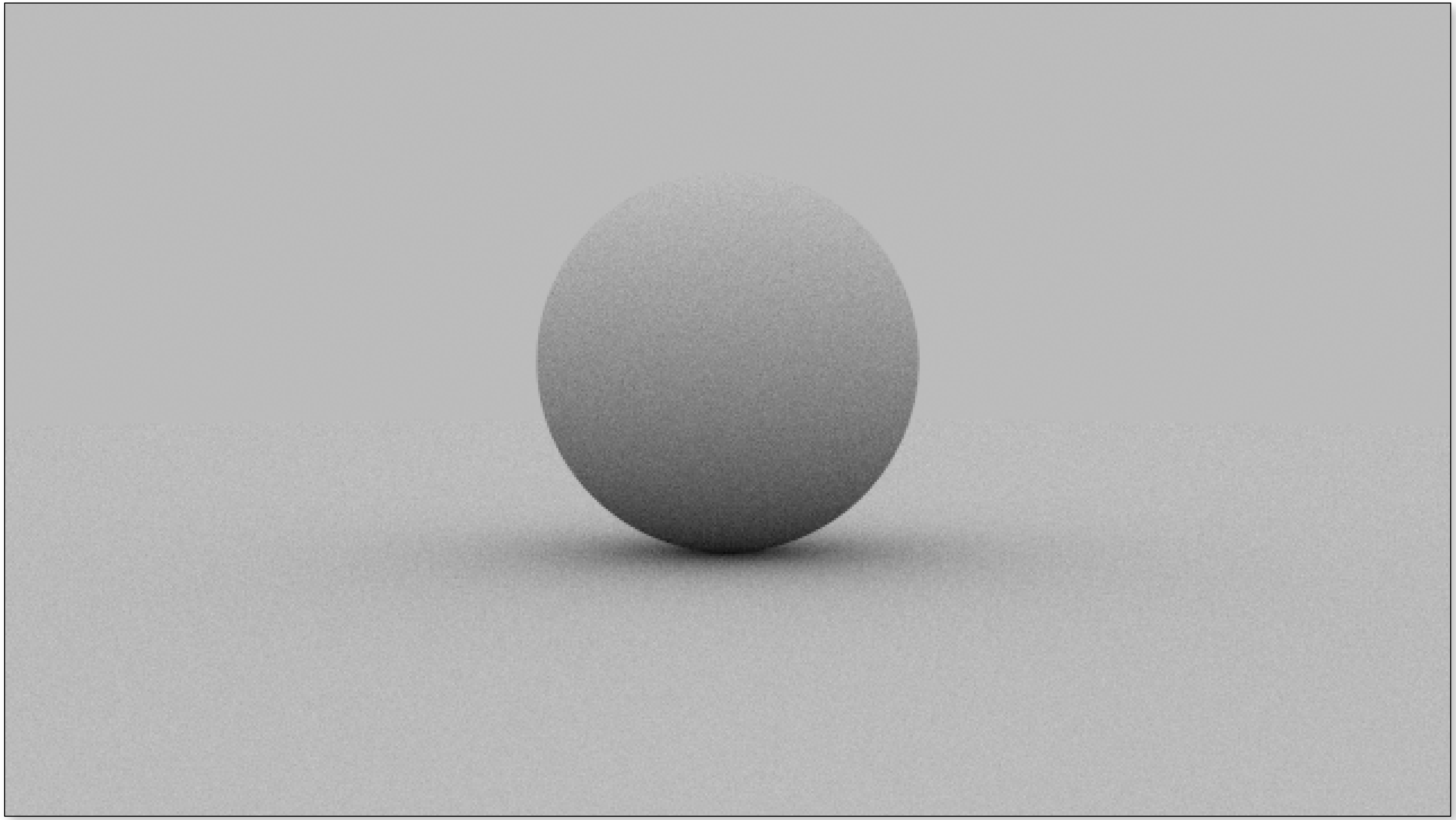
Hemispherical Sampling (9 Samples)



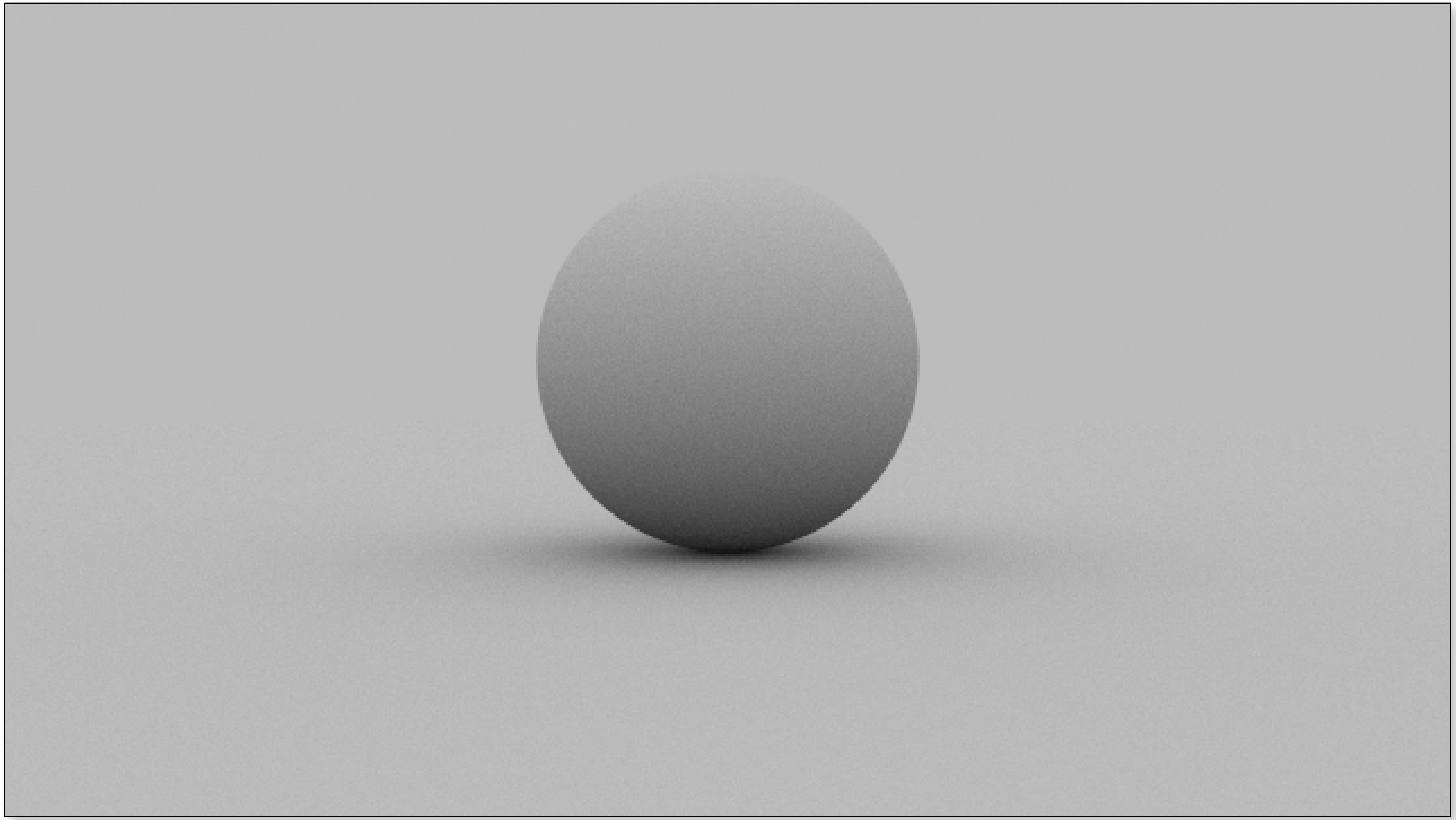
Hemispherical Sampling (16 Samples)



Hemispherical Sampling (256 Samples)

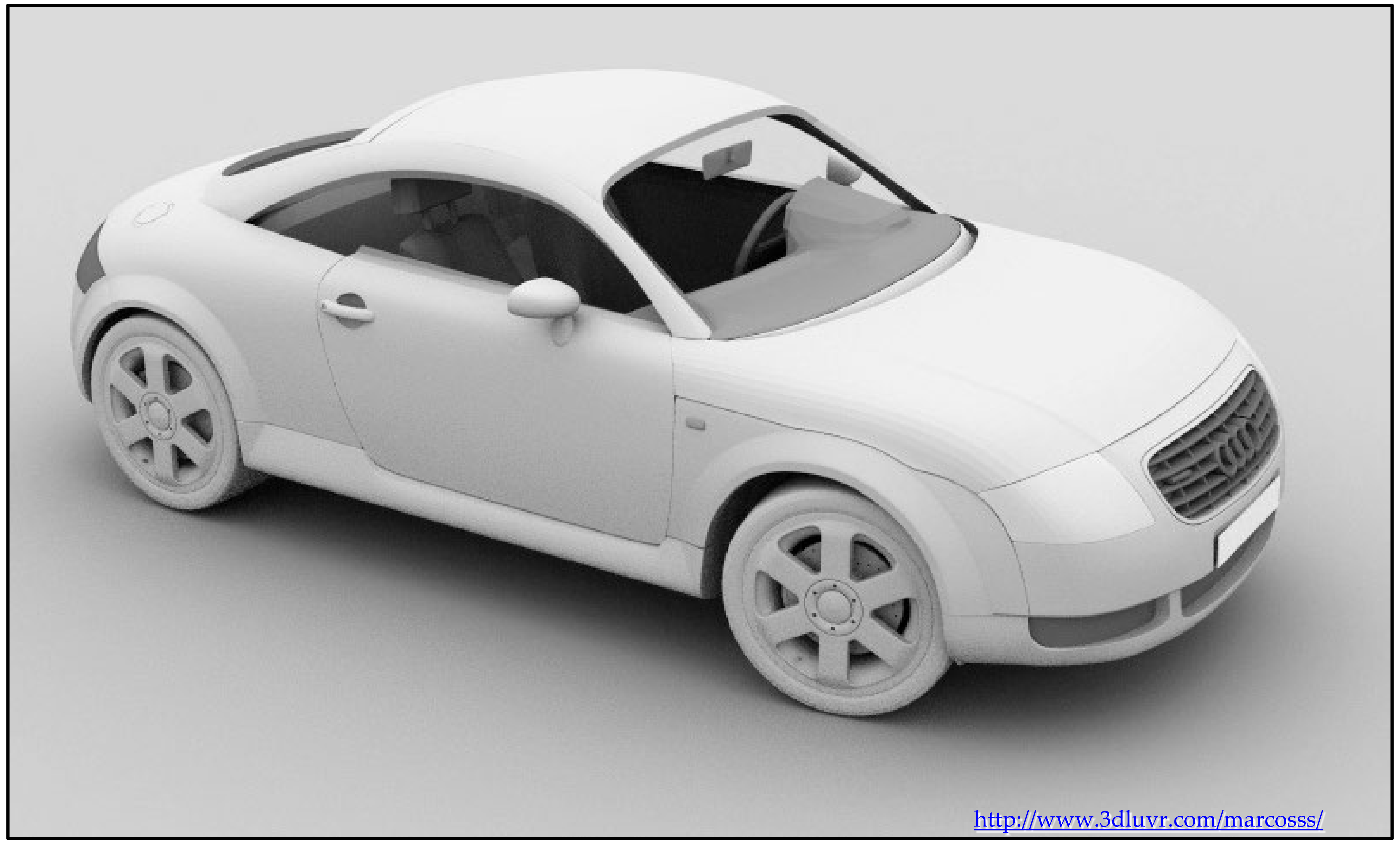


Hemispherical Sampling (1024 Samples)

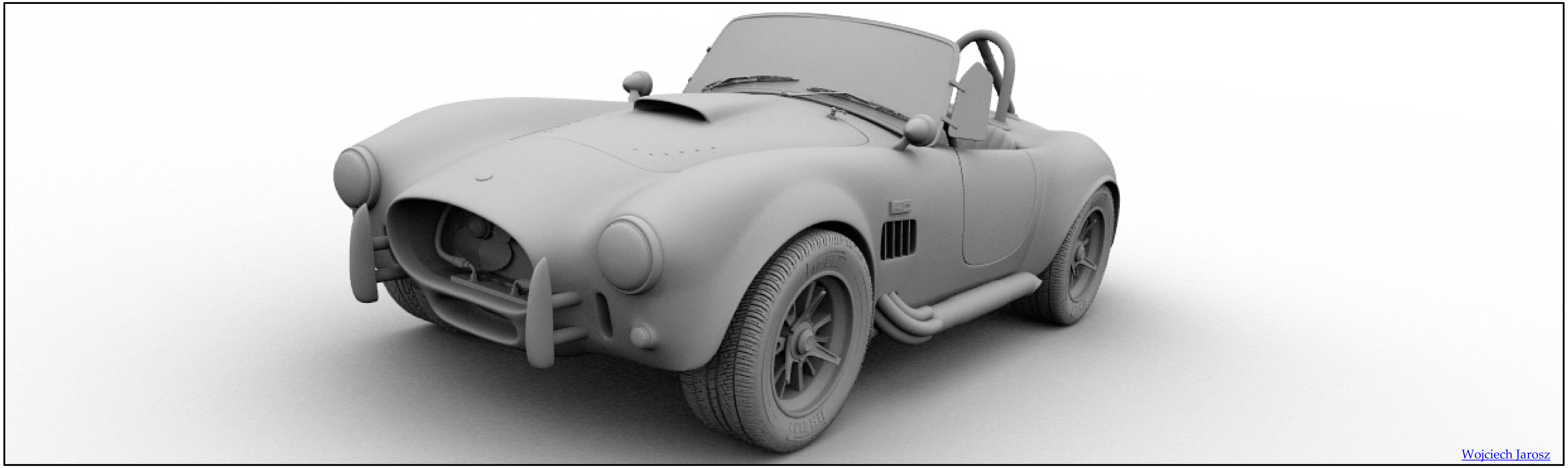


Ambient Occlusion

http://www.3dluvr.com/marcosss/



http://www.3dluvr.com/marcosss/



Wojciech Jarosz

Strategies for reducing variance

The standard MC estimator:

$$F = \int_{\mu(x)} f(x) \, \mathrm{d}\mu(x)$$

$$\langle F^N \rangle = \frac{1}{N} \sum_{i=0}^{N-1} \frac{f(X_i)}{\mathrm{pdf}(X_i)}$$

$$\sigma\left[\langle F^N \rangle\right] = \frac{1}{\sqrt{N}} \sigma\left[Y\right]$$

How do we reduce the variance of Y?

- Importance sampling

Importance sampling

Importance sampling

$$\int f(x)dx \qquad\qquad F_N = \frac{1}{N}\sum_{i=1}^{N}\frac{f(X_i)}{p(X_i)}$$

assume $p(x) = cf(x)$

$$\int p(x)dx = 1 \quad \rightarrow \quad c = \frac{1}{\int f(x)dx}$$

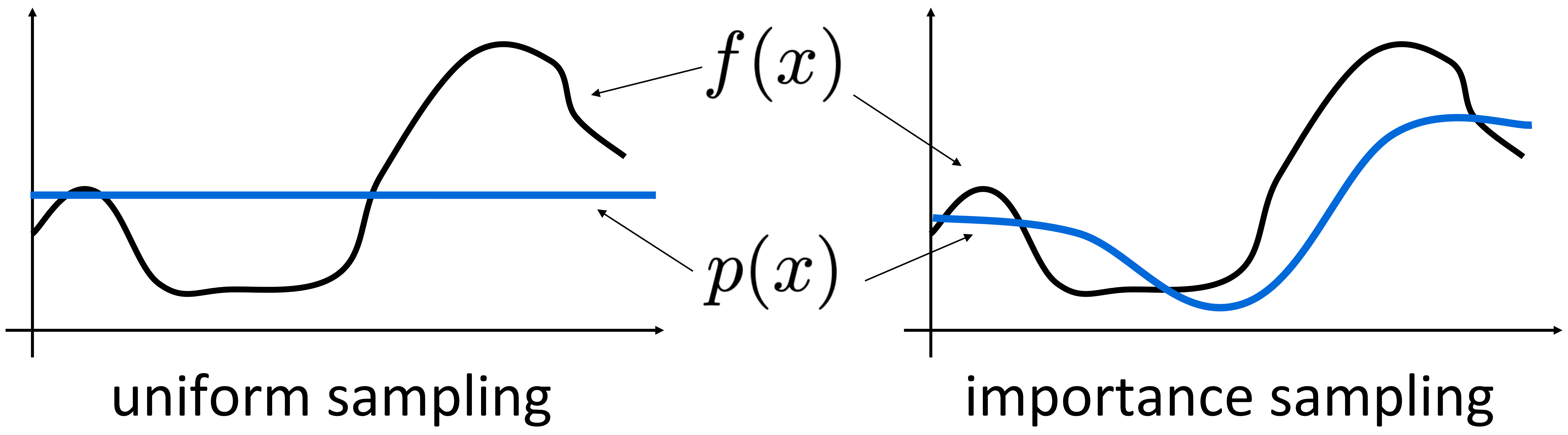
estimator $\frac{f(X_i)}{p(X_i)} = \frac{1}{c} = \int f(x)dx$ zero variance!

Importance sampling

$p(x) = cf(x)$ requires knowledge of the integral we are trying to compute in the first place!

But: If PDF is similar to integrand, variance can be significantly reduced

Common strategy: sample according to part of the integrand



Ambient occlusion

$$L_r(\mathbf{x}) = \frac{\rho}{\pi} \int_{H^2} V(\mathbf{x}, \vec{\omega}_i) \cos\theta_i \, d\vec{\omega}_i$$

What terms can we importance sample?

- incident radiance
- cosine term

Ambient occlusion

$$L_r(\mathbf{x}) = \frac{\rho}{\pi} \int_{H^2} V(\mathbf{x}, \vec{\omega}_i) \boxed{\cos\theta_i} d\vec{\omega}_i$$

What terms can we importance sample?

- incident radiance
- cosine term

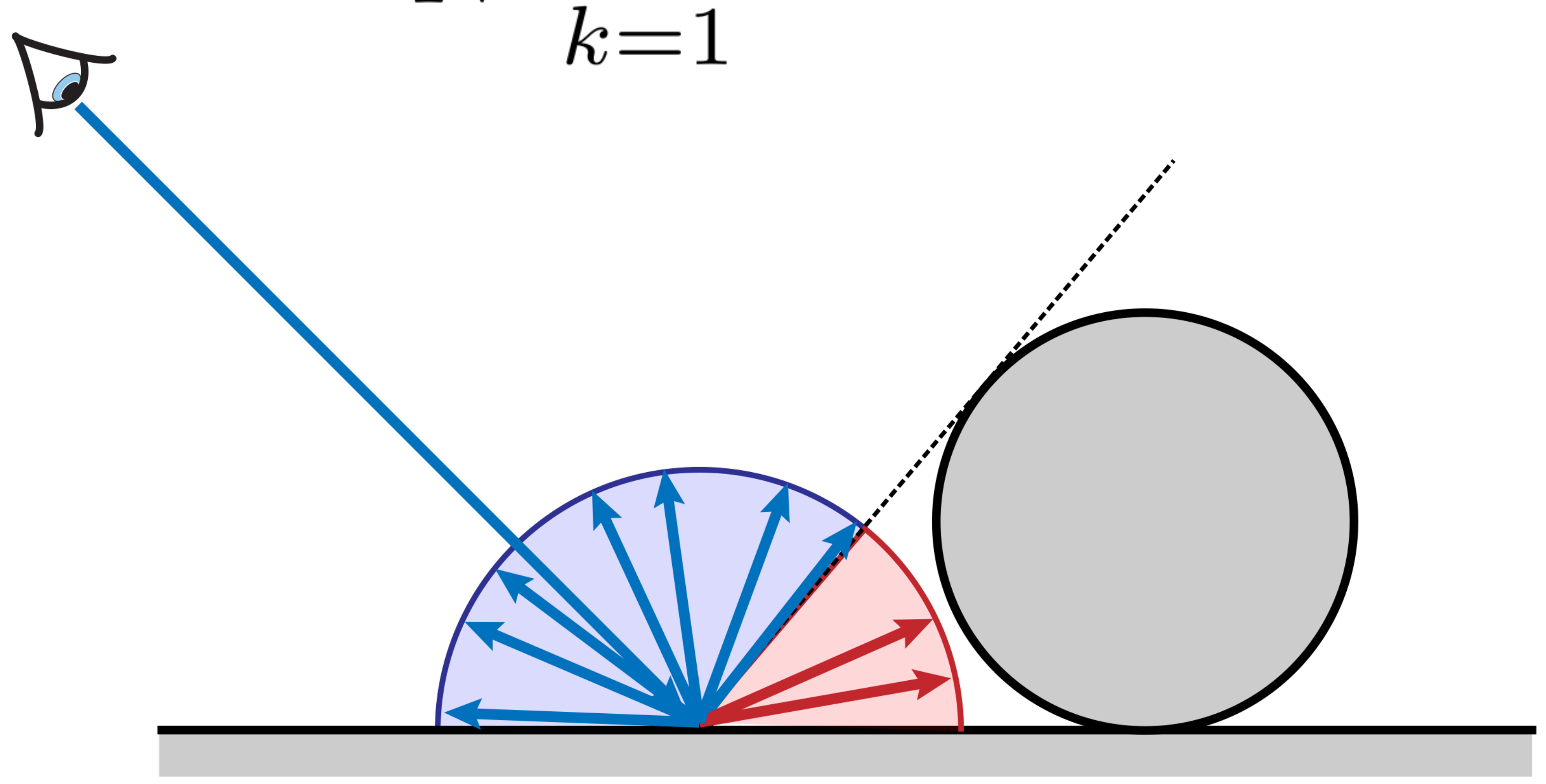
Ambient Occlusion

$$L_r(\mathbf{x}) \approx \frac{\rho}{\pi N} \sum_{k=1}^{N} \frac{V(\mathbf{x}, \vec{\omega}_{i,k}) \cos\theta_{i,k}}{p(\vec{\omega}_{i,k})}$$

Uniform hemispherical sampling

$$p(\vec{\omega}_{i,k}) = 1/2\pi$$

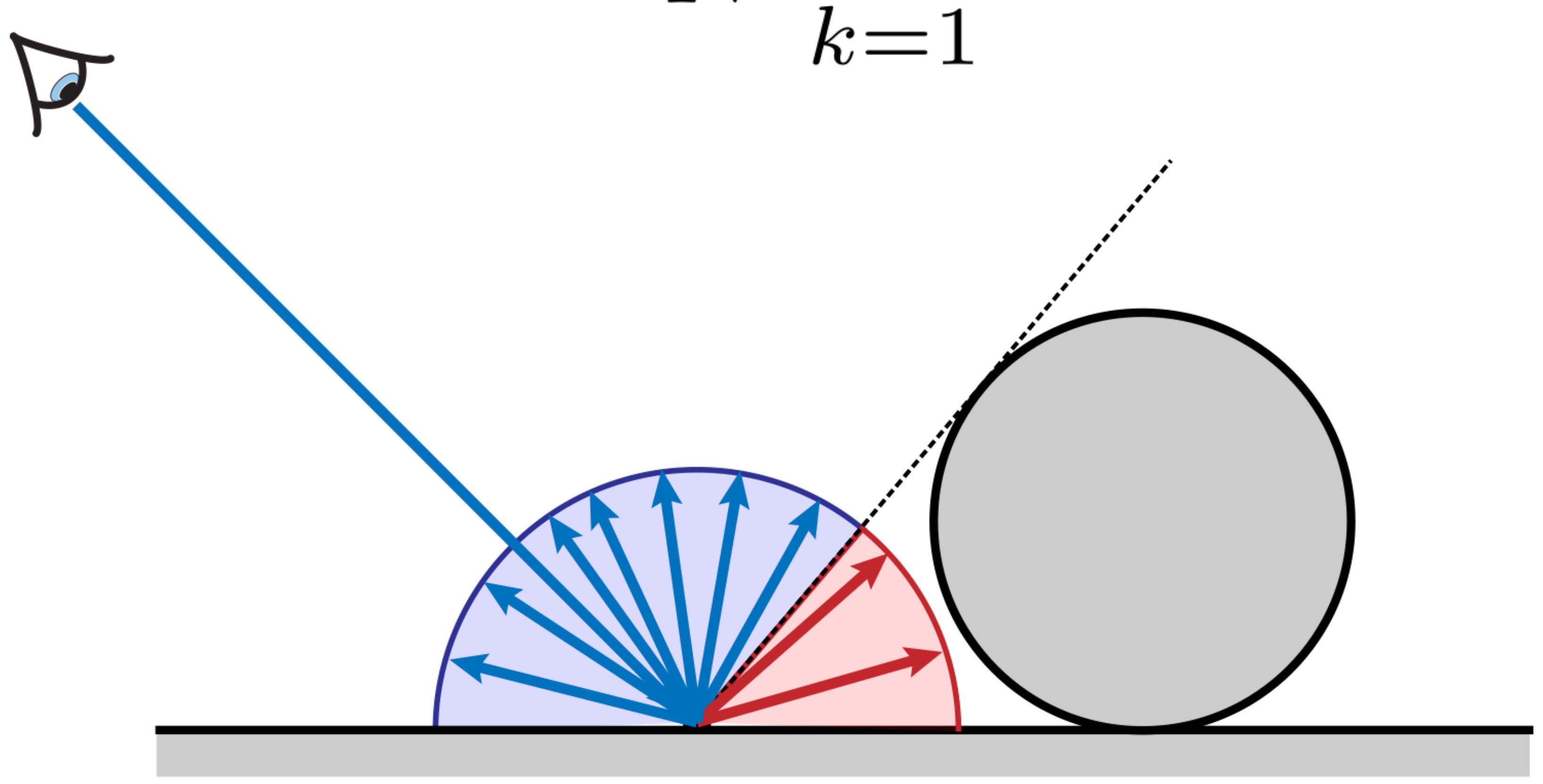
$$L_r(\mathbf{x}) \approx \frac{2\rho}{N} \sum_{k=1}^{N} V(\mathbf{x}, \vec{\omega}_{i,k}) \cos\theta_{i,k}$$



Cosine-weighted importance sampling

$$p(\vec{\omega}_{i,k}) = \cos\theta_{i,k}/\pi$$

$$L_r(\mathbf{x}) \approx \frac{\rho}{N} \sum_{k=1}^{N} V(\mathbf{x}, \vec{\omega}_{i,k})$$

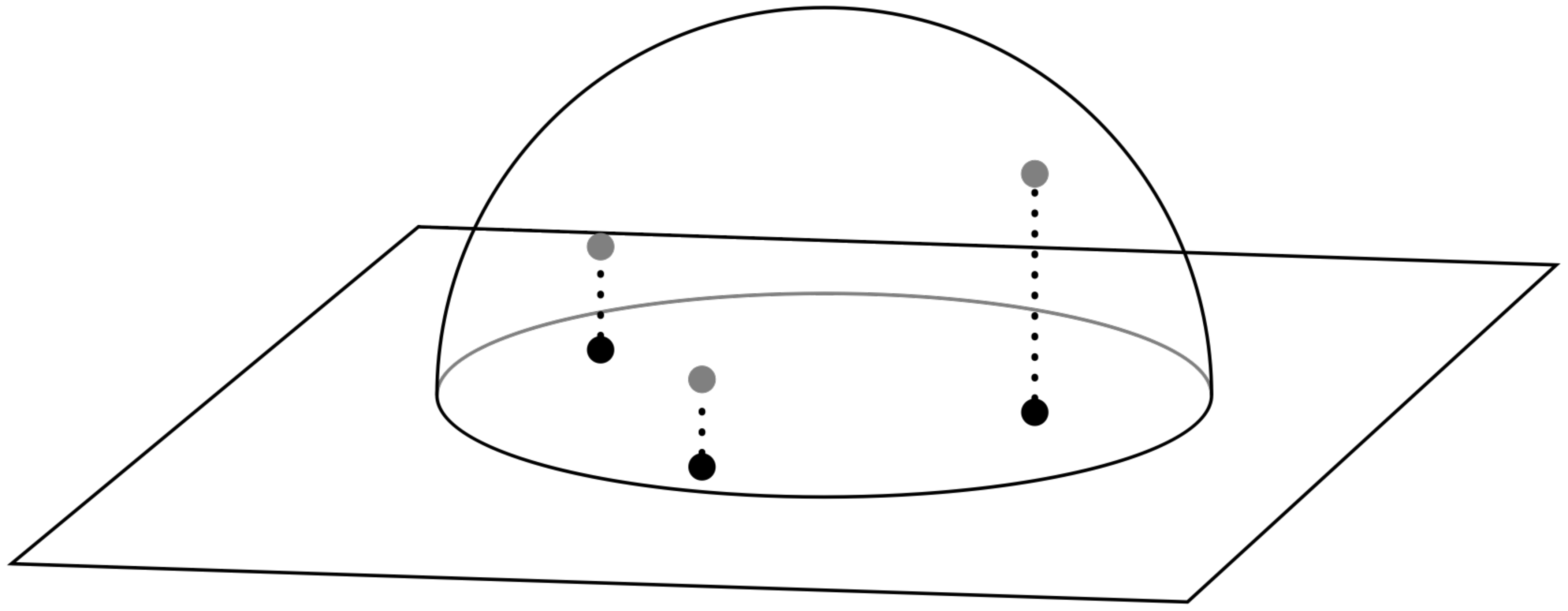


Cosine-weighted Hemispherical Sampling

Could proceed as before: compute marginal and conditional densities, then use inversion method.

It turns out that:

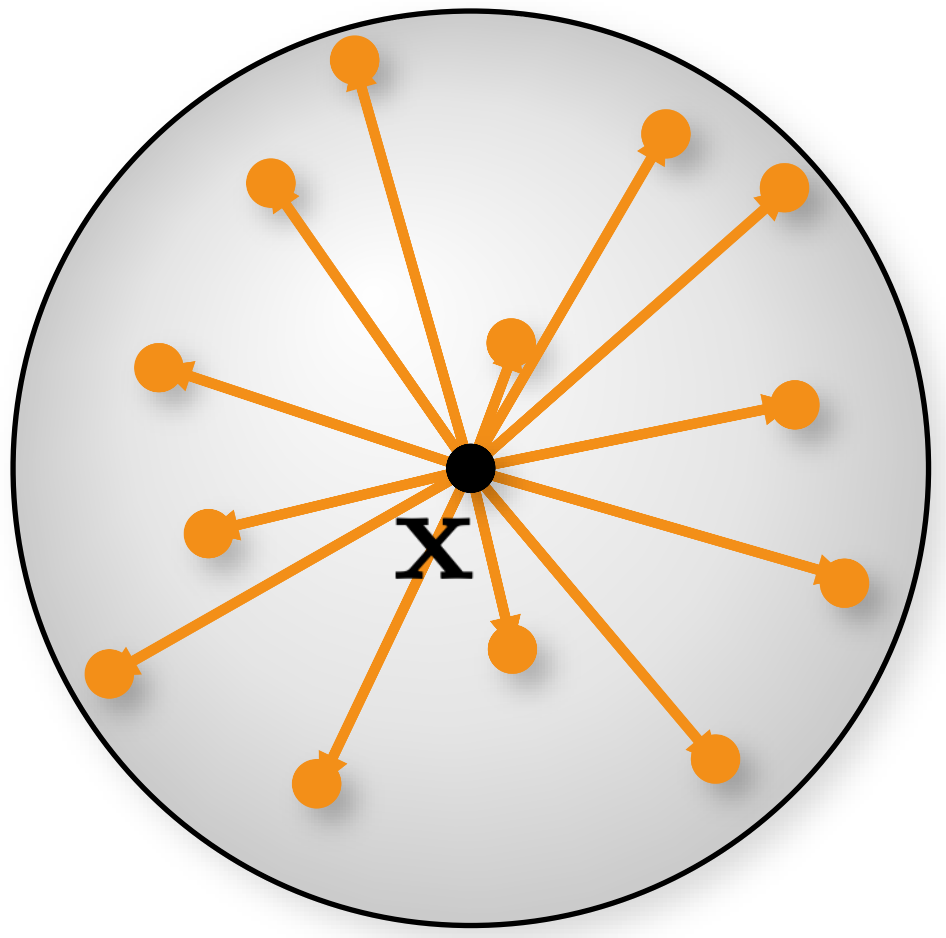
- Generating points uniformly on the disc, and then project these points vertically onto the hemisphere produces the desired distribution.



Cosine-weighted Hemispherical Sampling

Generate points on sphere

(unit directions)



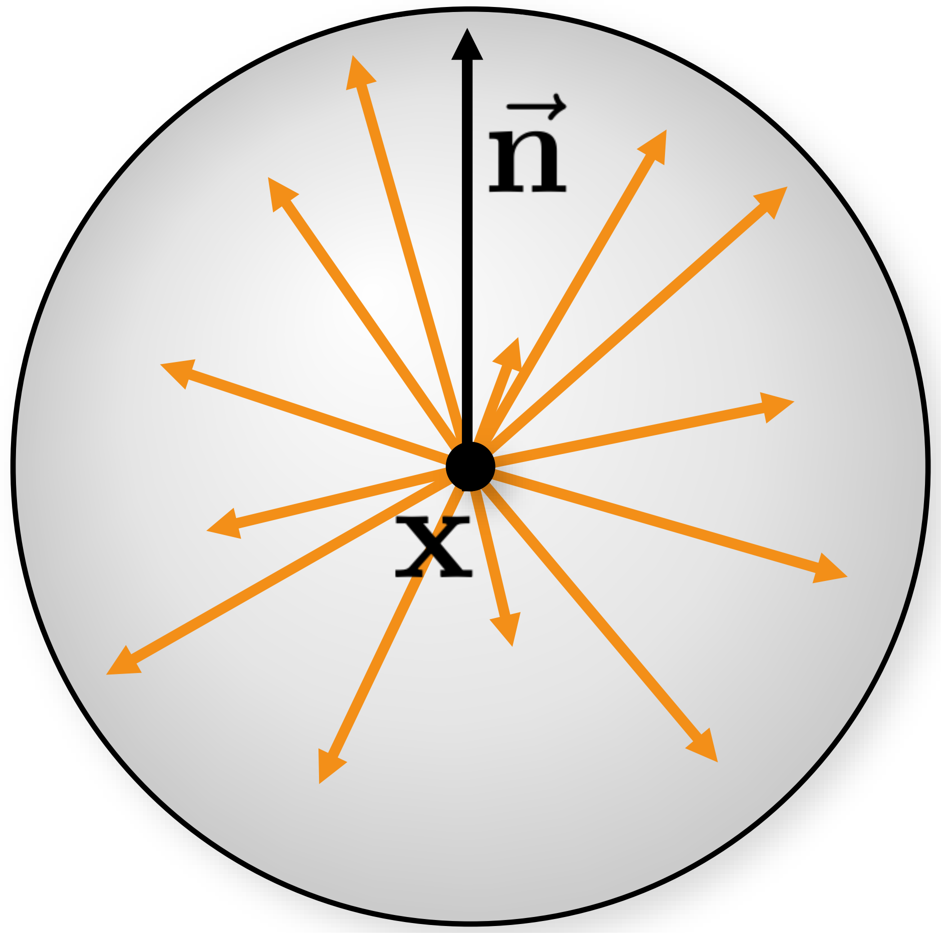
Cosine-weighted Hemispherical Sampling

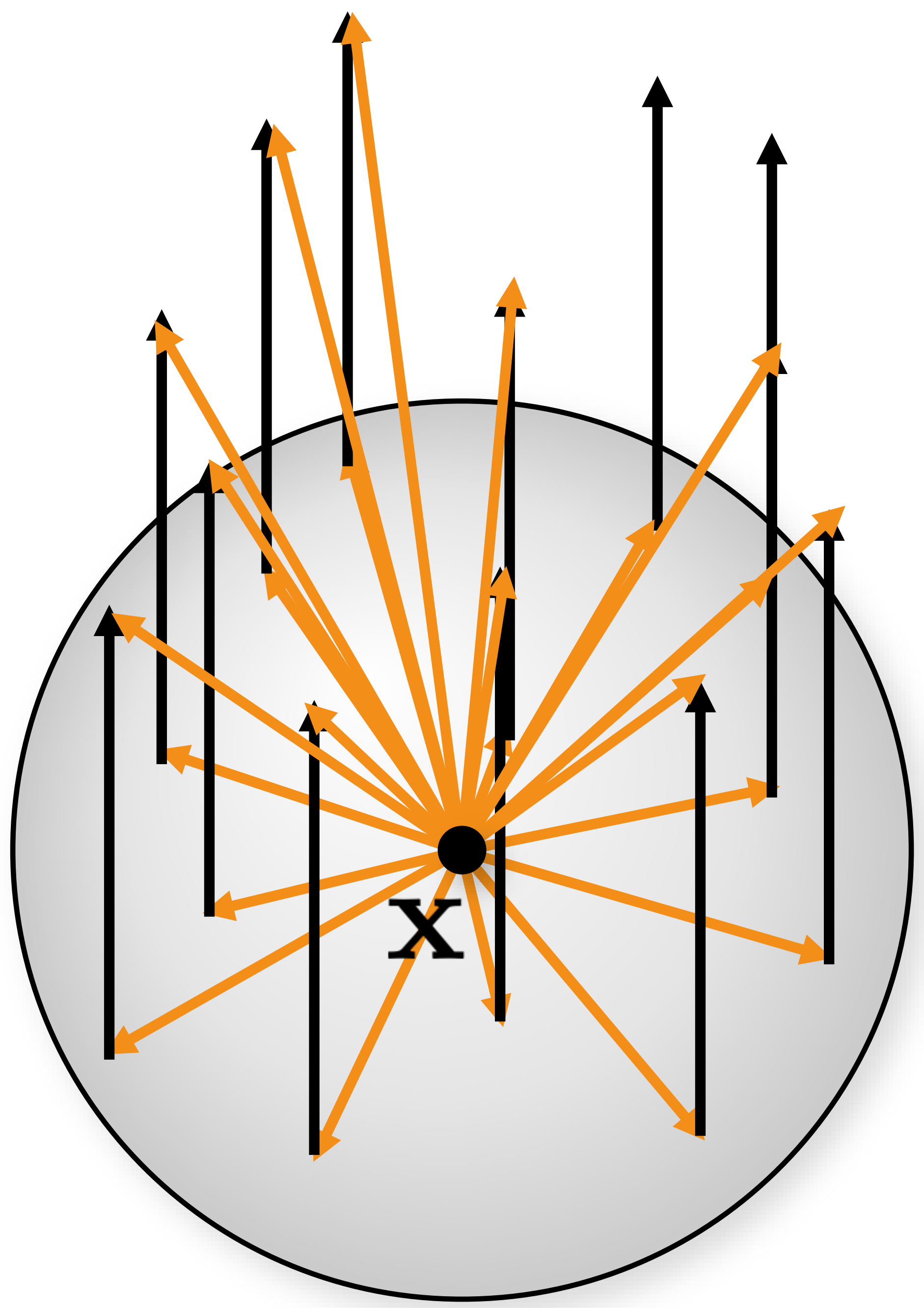
Generate points on sphere

(unit directions)

Add unit normal

unit normal





Cosine-weighted Hemispherical Sampling

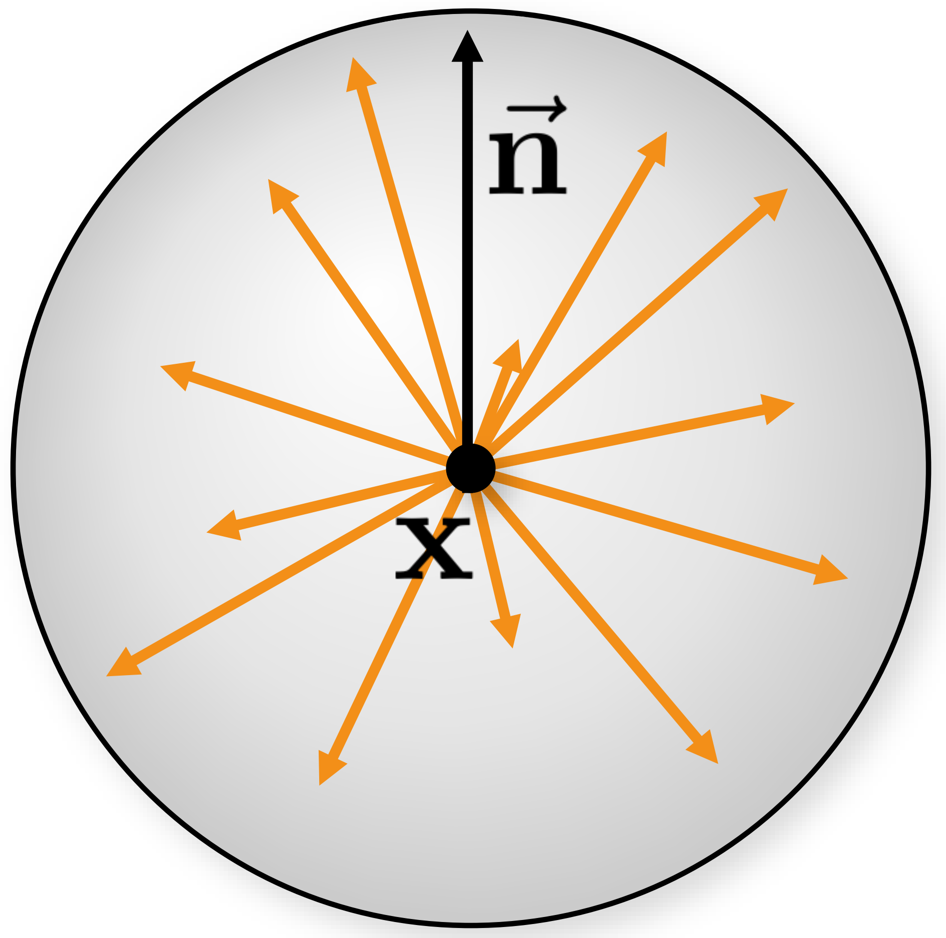
Generate points on sphere

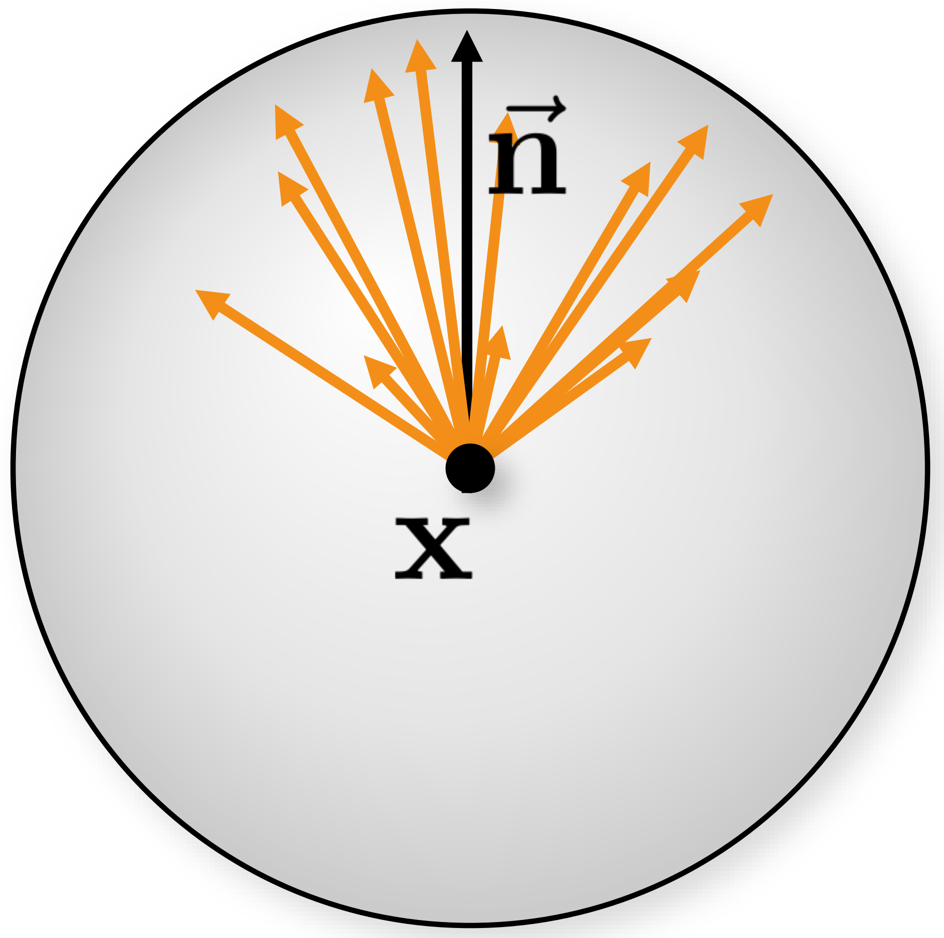
(unit directions)

Add unit normal

normalize

unit normal





Uniform hemispherical sampling

1 sample/pixel

Cosine-weighted importance sampling

Uniform hemispherical sampling

4 sample/pixel

Cosine-weighted importance sampling

Uniform hemispherical sampling

16 sample/pixel

Cosine-weighted importance sampling

Uniform hemispherical sampling

1024 sample/pixel

Cosine-weighted importance sampling

Strategies for reducing variance

The standard MC estimator:

$$F = \int_{\mu(x)} f(x)\, \mathrm{d}\mu(x)$$

$$\langle F^N \rangle = \frac{1}{N} \sum_{i=0}^{N-1} \frac{f(X_i)}{\mathrm{pdf}(X_i)}$$

$$\sigma\left[\langle F^N \rangle\right] = \frac{1}{\sqrt{N}} \sigma\left[Y\right]$$

How do we reduce the variance of Y?

- Importance sampling

Equal-sample versus equal-time comparisons

$$\sigma\left[\langle F^N \rangle\right] = \frac{1}{\sqrt{N}}\sigma\left[Y\right]$$

- Importance sampling improves the $\sigma[Y]$ term
- But an importance sampling technique may be more expensive to run than naive uniform sampling, reducing the N term given fixed runtime.

Cost of an estimator:

$$C = N \cdot T$$

T ← time to draw one sample for a given sampling technique

N ← number of samples

- Equal-sample (fixed N) comparisons can be misleading.
- Equal-time comparisons (fixed total runtime, which is equivalent to fixed cost C) are more representative of performance.
  - At equal time, a naive sampling technique that draws very many bad samples can result in less variance than a sophisticated technique that draws very few great samples.

More Integration Dimensions

Anti-aliasing (image space)

Light visibility (surface of area lights)

Depth-of-field (camera aperture)

Motion blur (time)

Many lights

Multiple bounces of light

Participating media (volume)