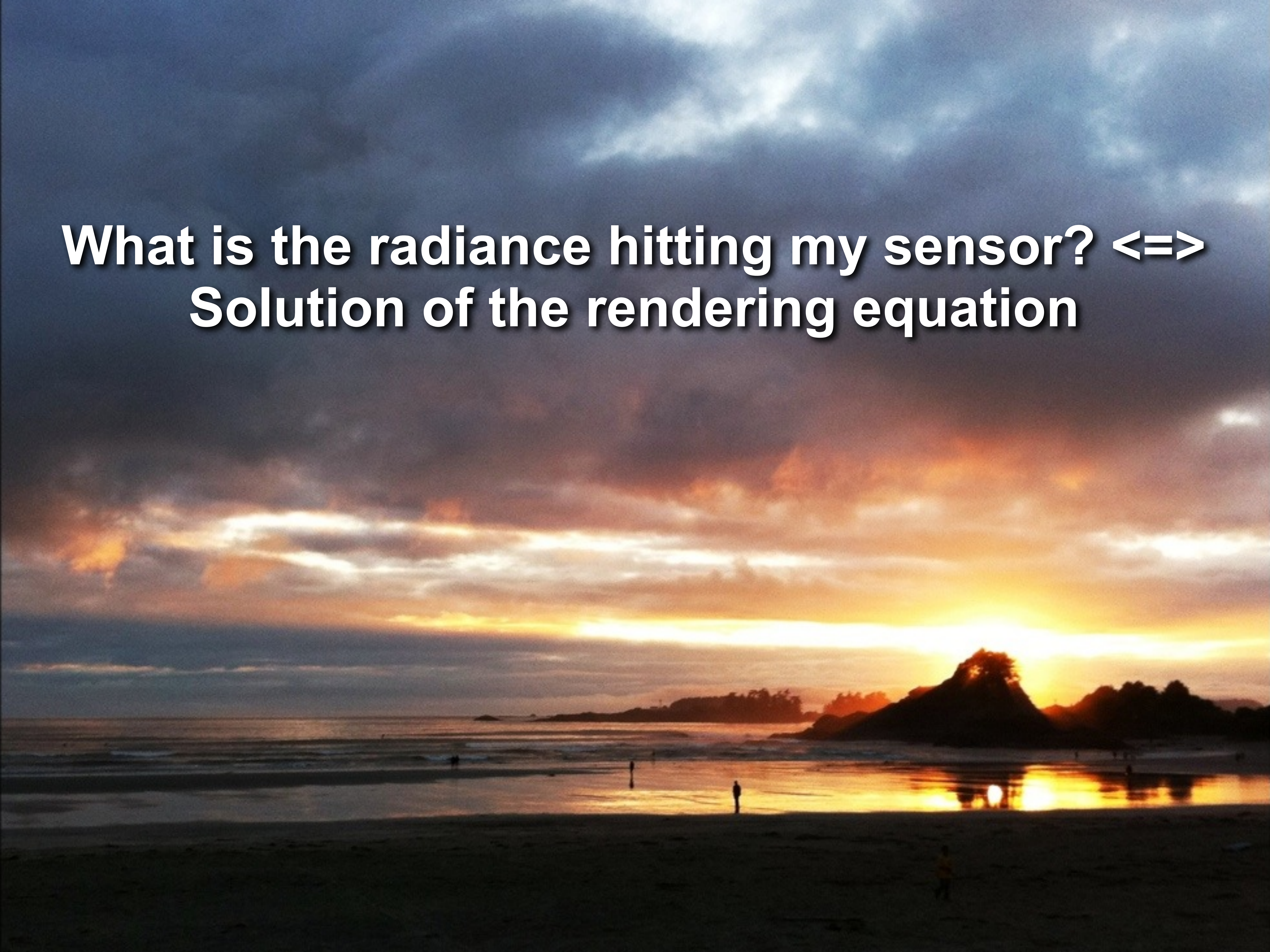
Monte Carlo Integration and Importance Sampling I

CS-E5520 Spring 2023
Jaakko Lehtinen
with many slides from Frédo Durand

Monte Carlo Integration and Importance Sampling I

CS-E5520 Spring 2023
Jaakko Lehtinen
with many slides from Frédo Durand

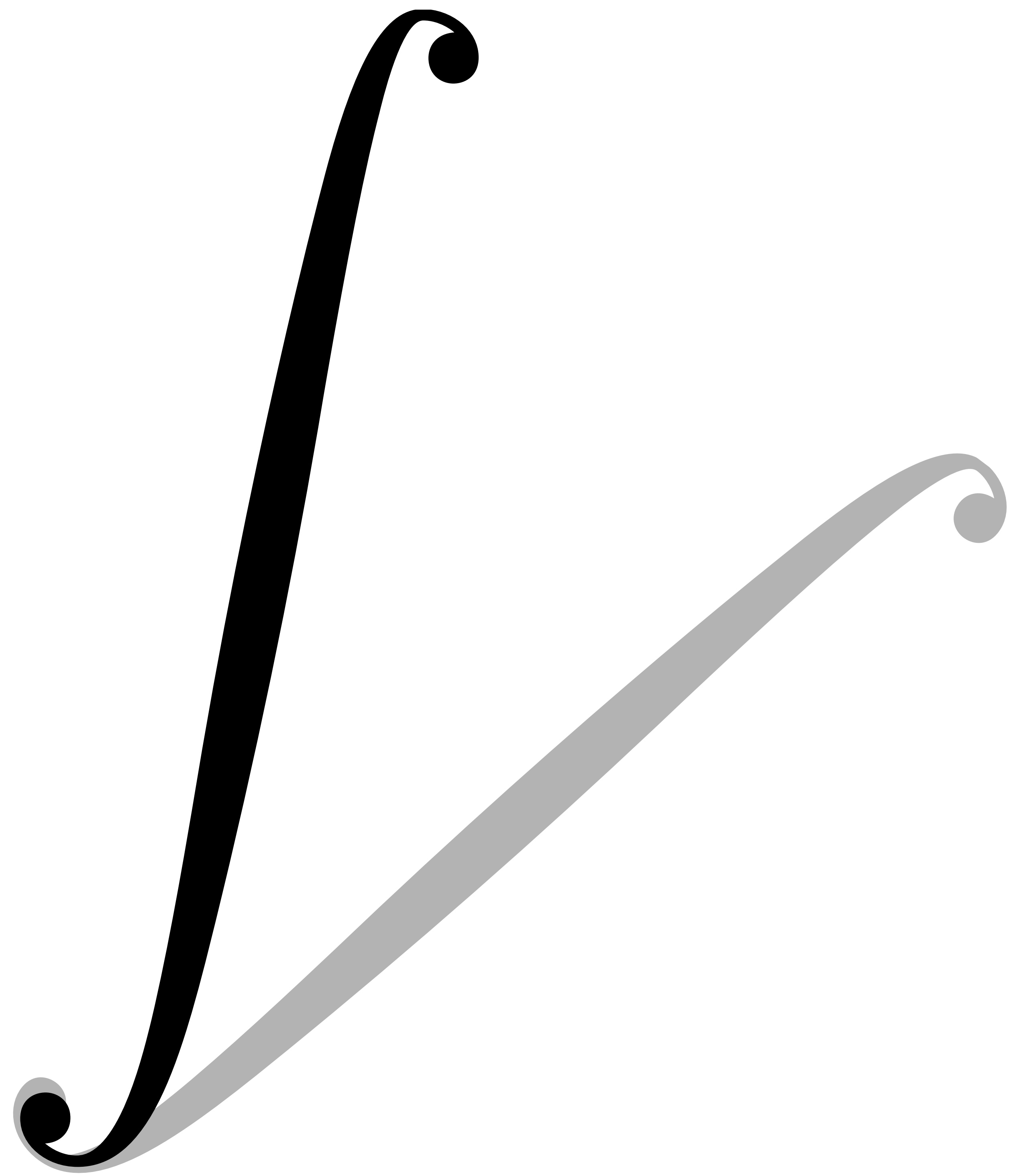


What is the radiance hitting my sensor? <=> Solution of the rendering equation

Today

- Intro to Monte Carlo integration
  - Basics
  - Importance Sampling

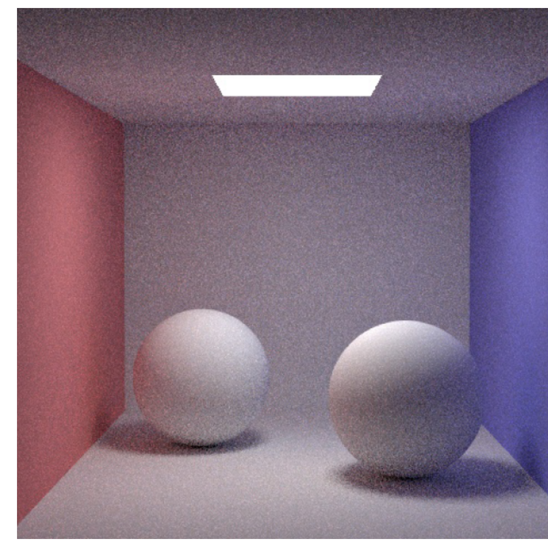
Integrals are Everywhere

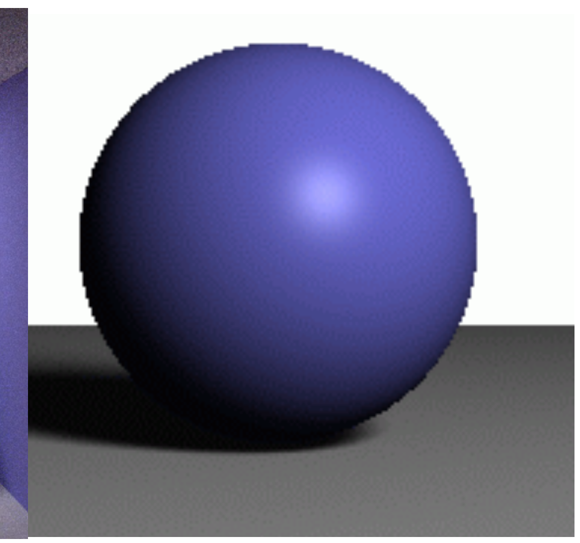


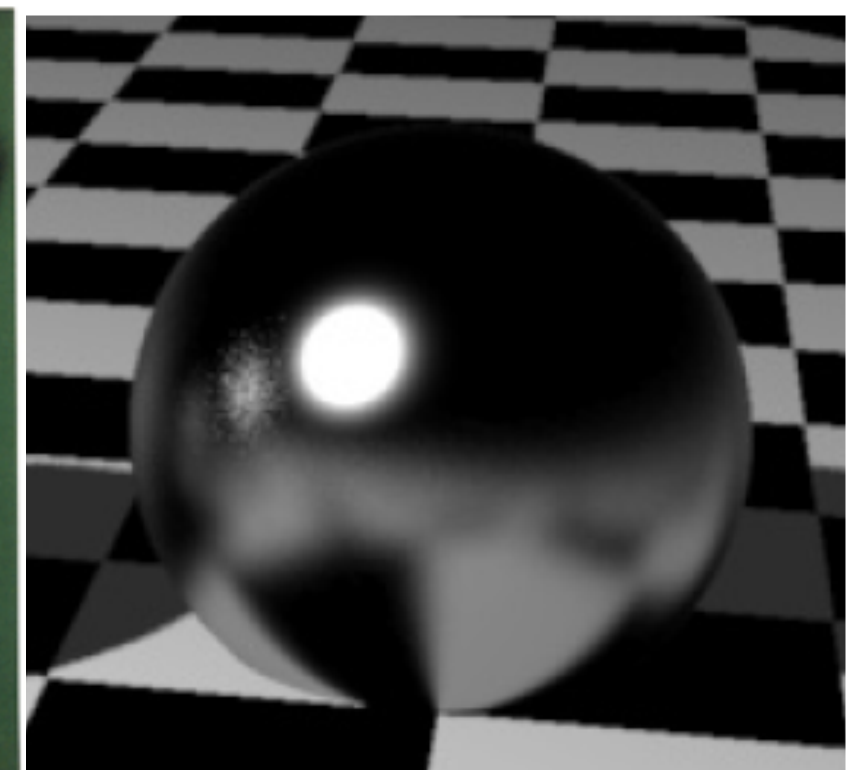
For Example...

$$\int\int\int\int\int L(x,y,t,u,v)\,dx\,dy\,dt\,du\,dv$$

- Pixel: antialiasing
- Light sources: Soft shadows
- Lens: Depth of field
- Time: Motion blur
- BRDF: glossy reflection
- Hemisphere: indirect lighting

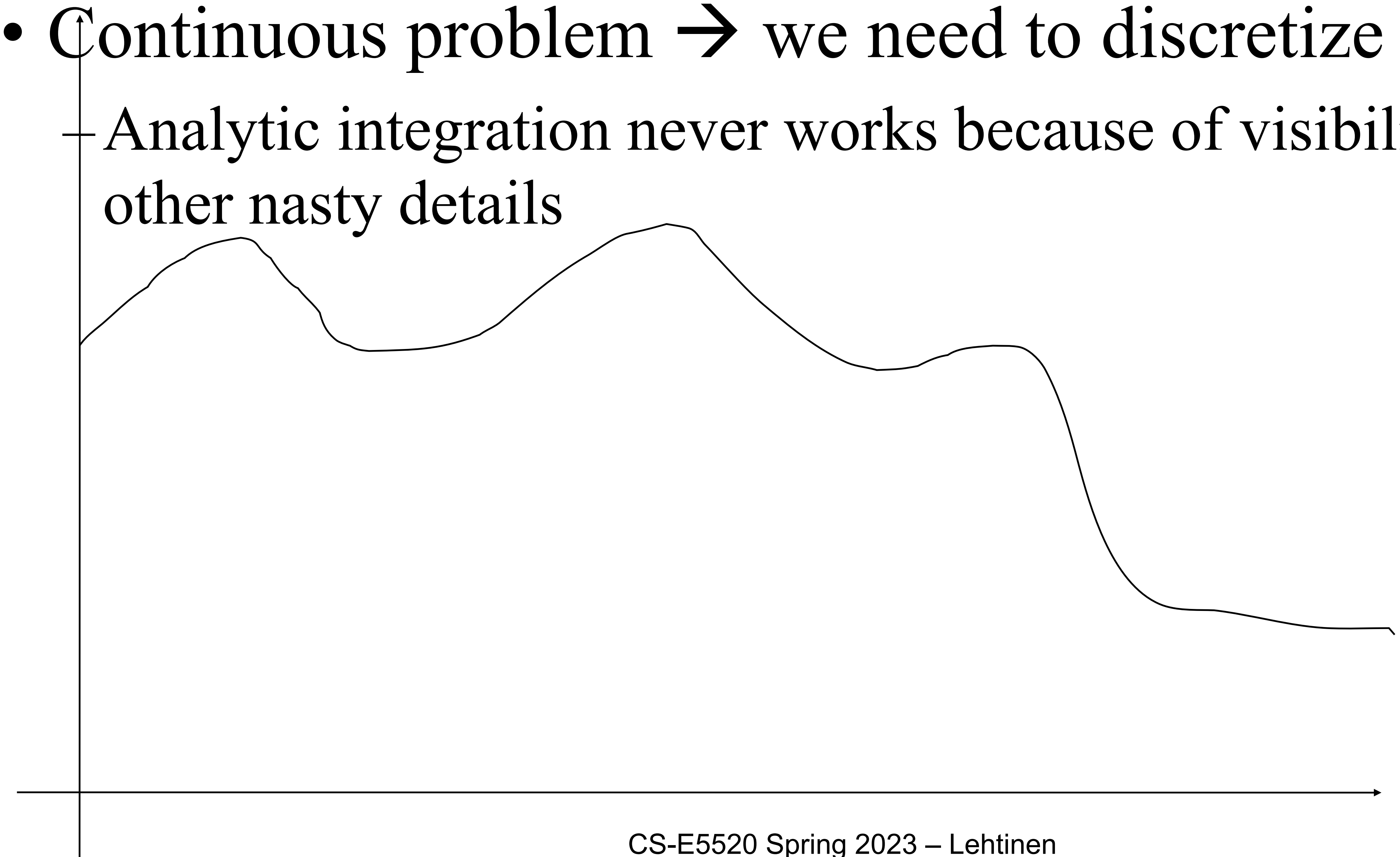






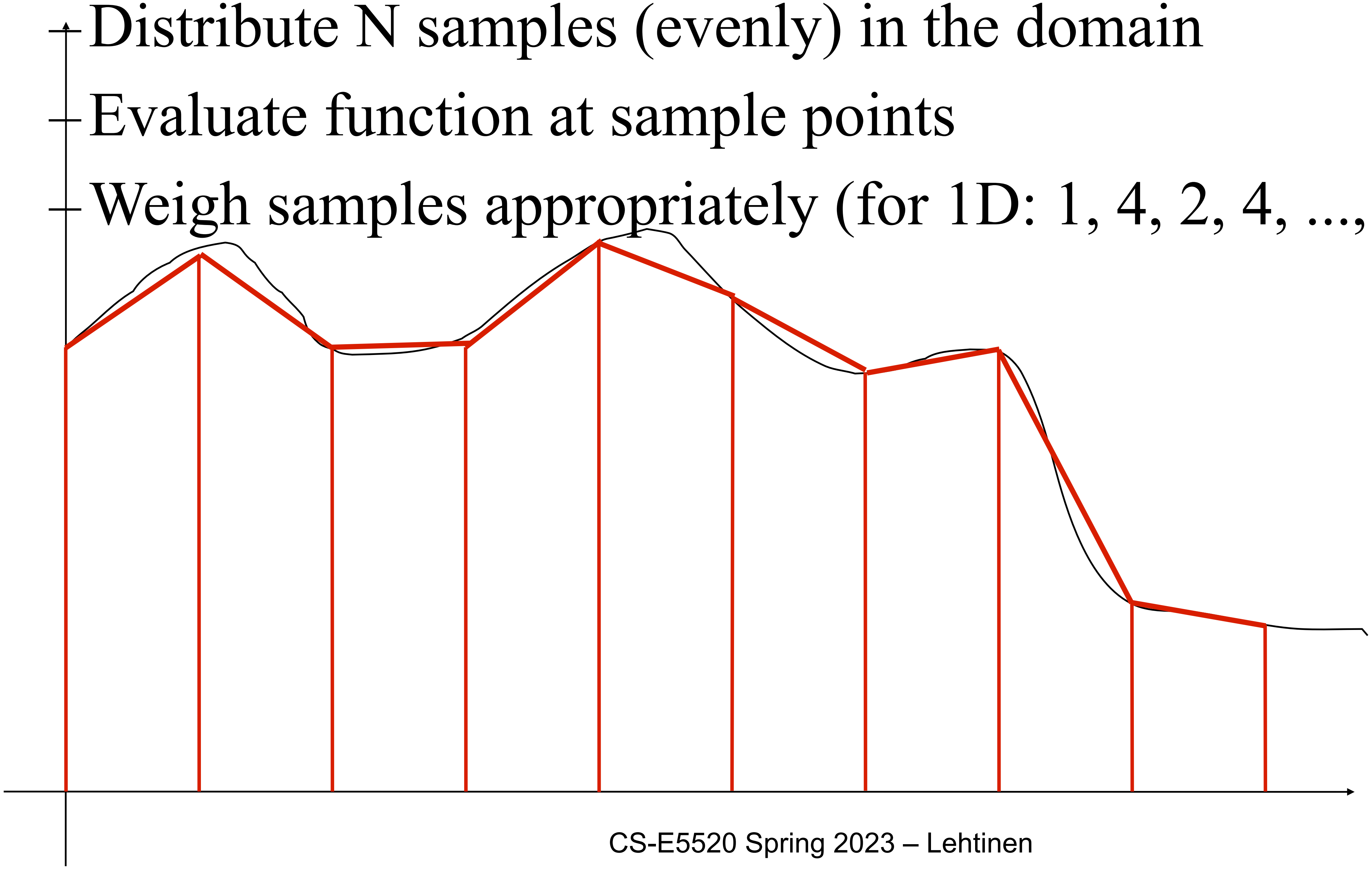
Numerical Integration

- Compute integral of arbitrary function
  - e.g. integral over area light source, over hemisphere, etc.
- Continuous problem → we need to discretize
  - Analytic integration never works because of visibility and other nasty details



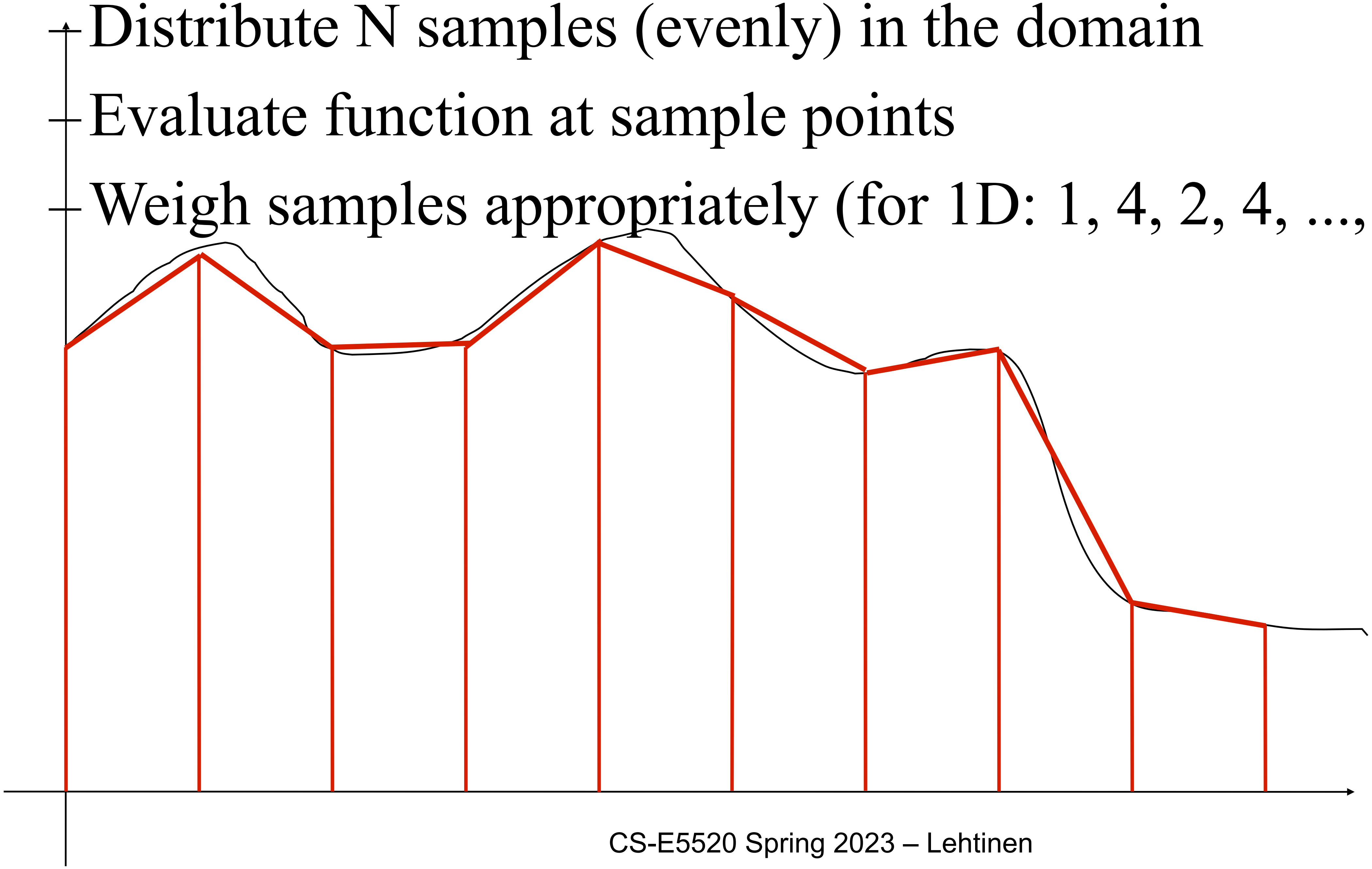
Numerical Integration

- You know trapezoid, Simpson's rule, etc. from your first engineering math class
  - Distribute N samples (evenly) in the domain
  - Evaluate function at sample points
  - Weigh samples appropriately (for 1D: 1, 4, 2, 4, ..., 2, 4, 1)



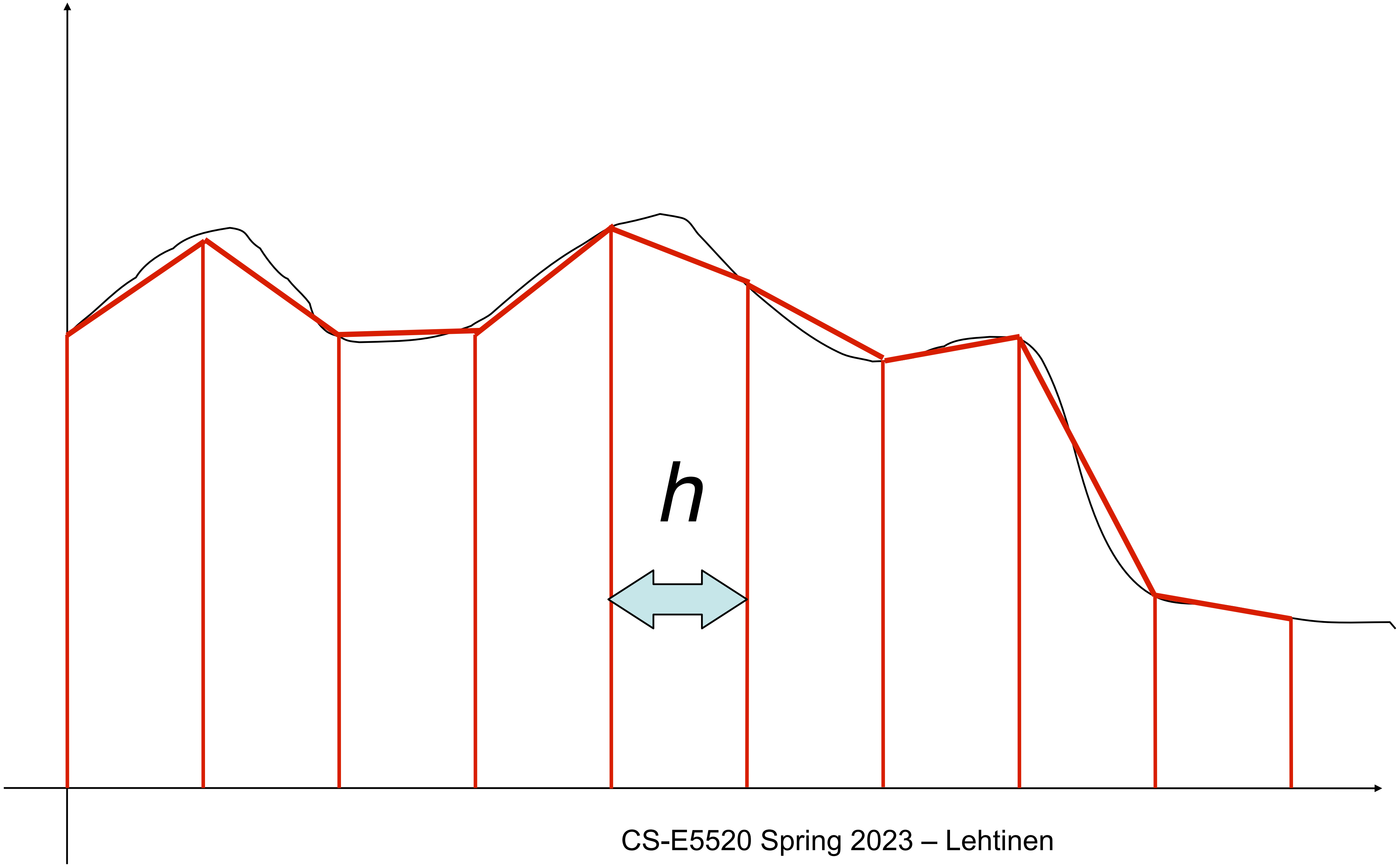
Why is This Bad?

- You know trapezoid, Simpson's rule, etc. from your first engineering math class
  - Distribute N samples (evenly) in the domain
  - Evaluate function at sample points
  - Weigh samples appropriately (for 1D: 1, 4, 2, 4, ..., 2, 4, 1)



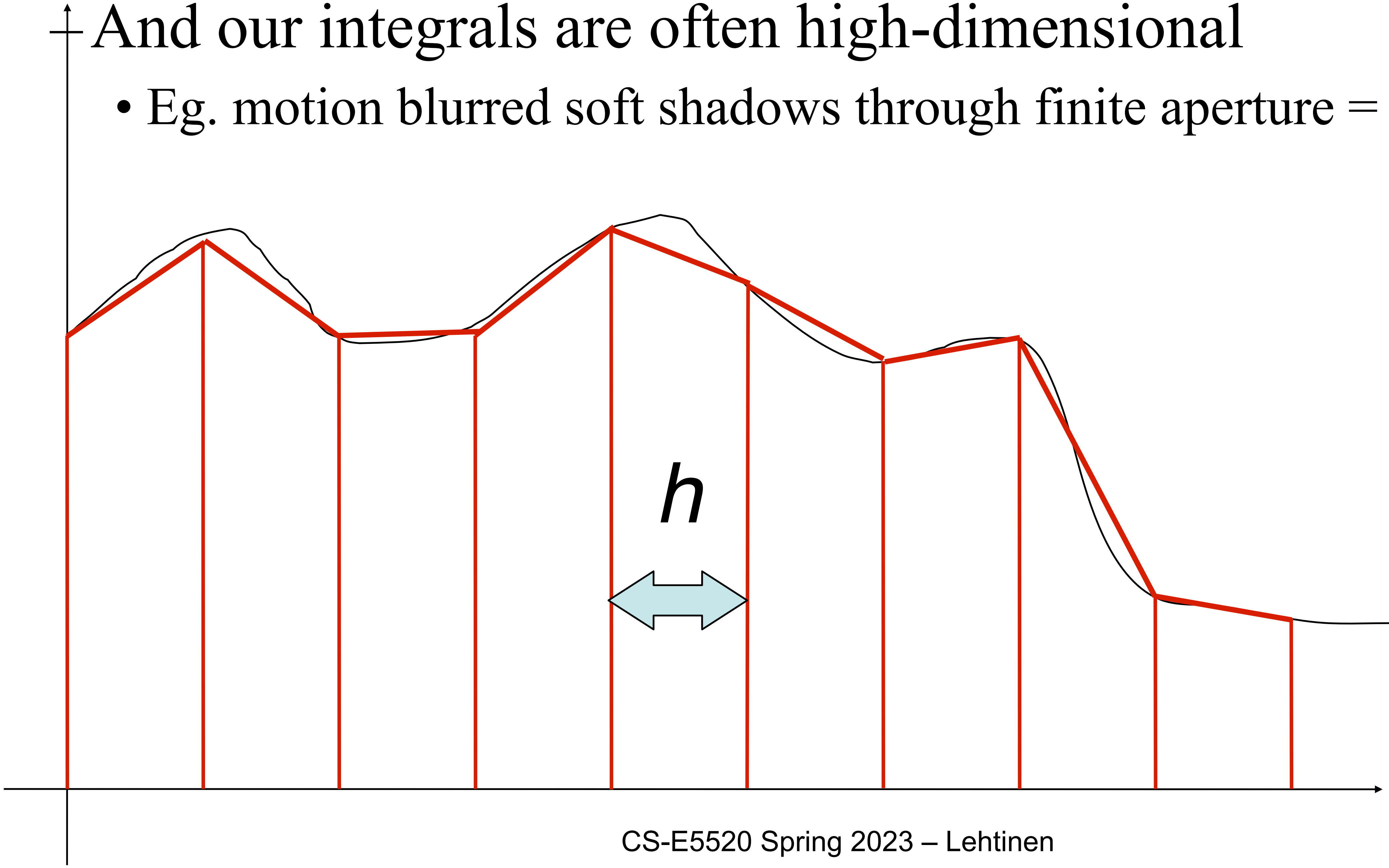
Why is This Bad?

- Error scales with (some power of) grid spacing h



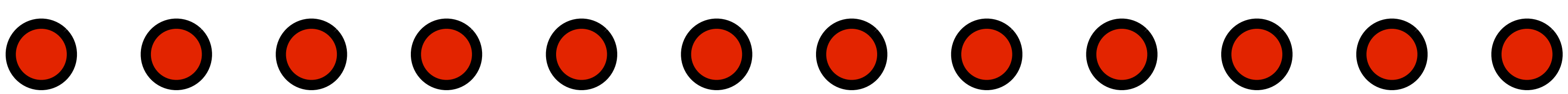
Why is This Bad?

- Error scales with (some power of) grid spacing h
- Bad things happen when dimension grows..
  - And our integrals are often high-dimensional
    - Eg. motion blurred soft shadows through finite aperture = 7D!

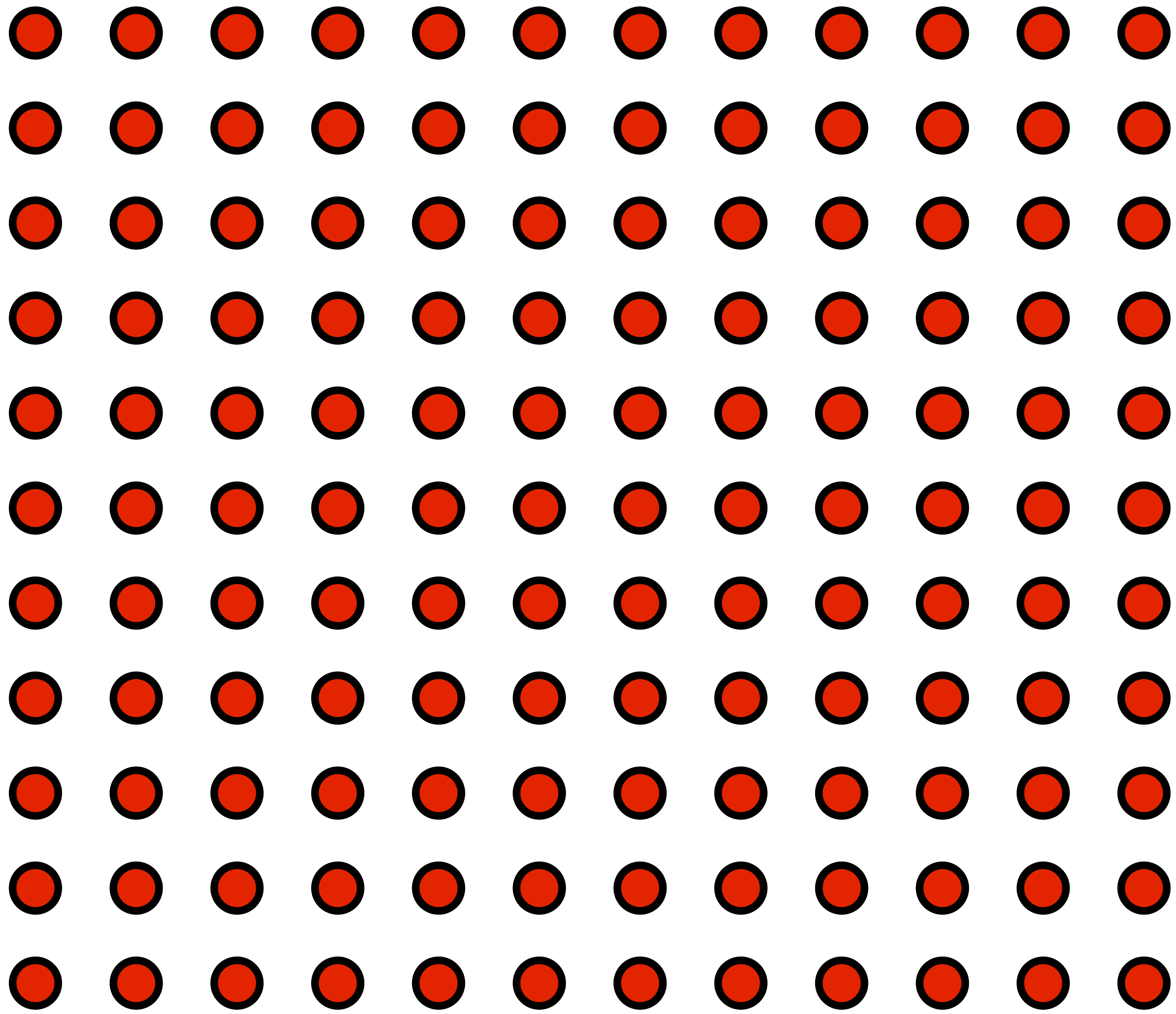


Constant spacing, 1D

n

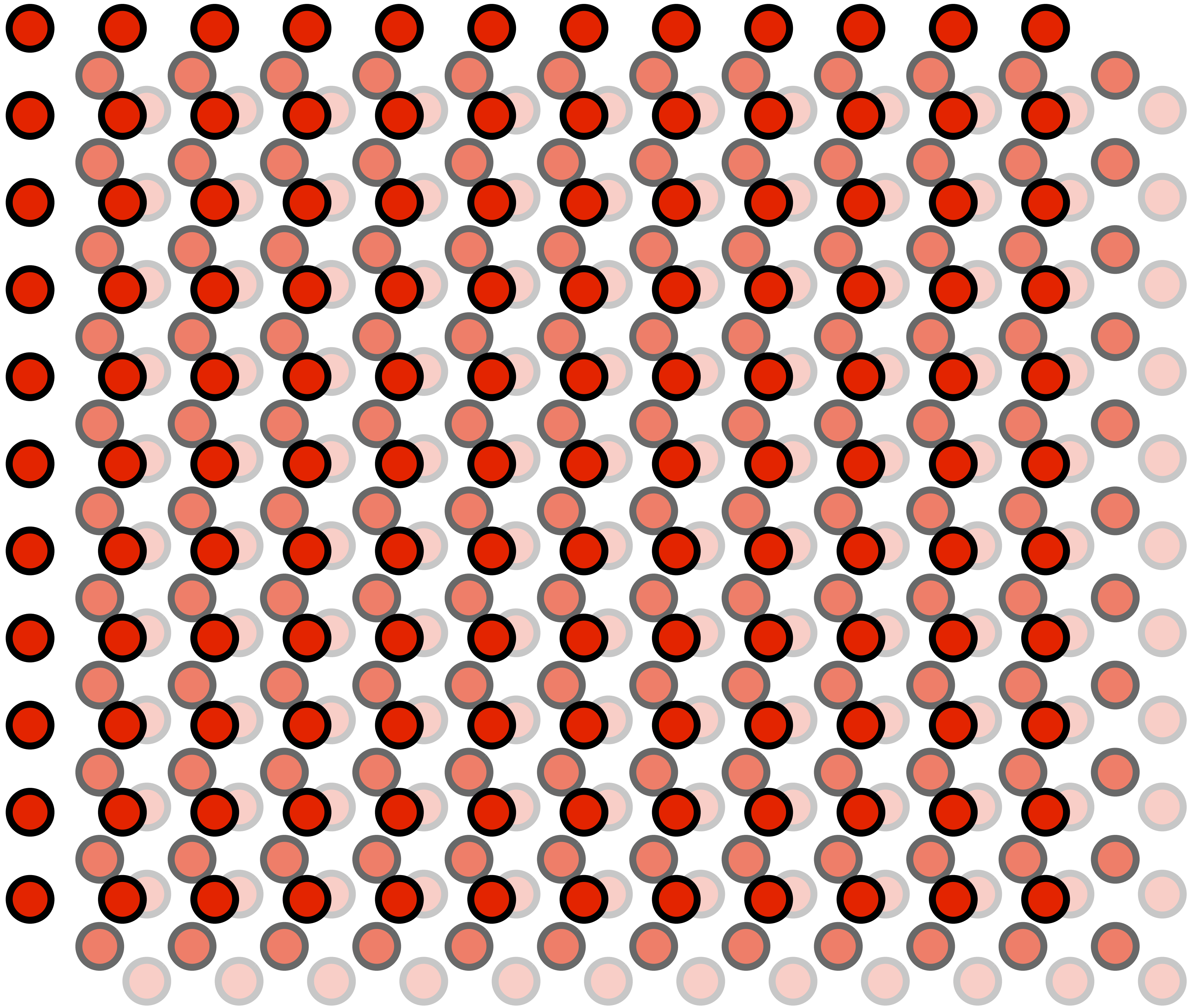


2D (yikes!)



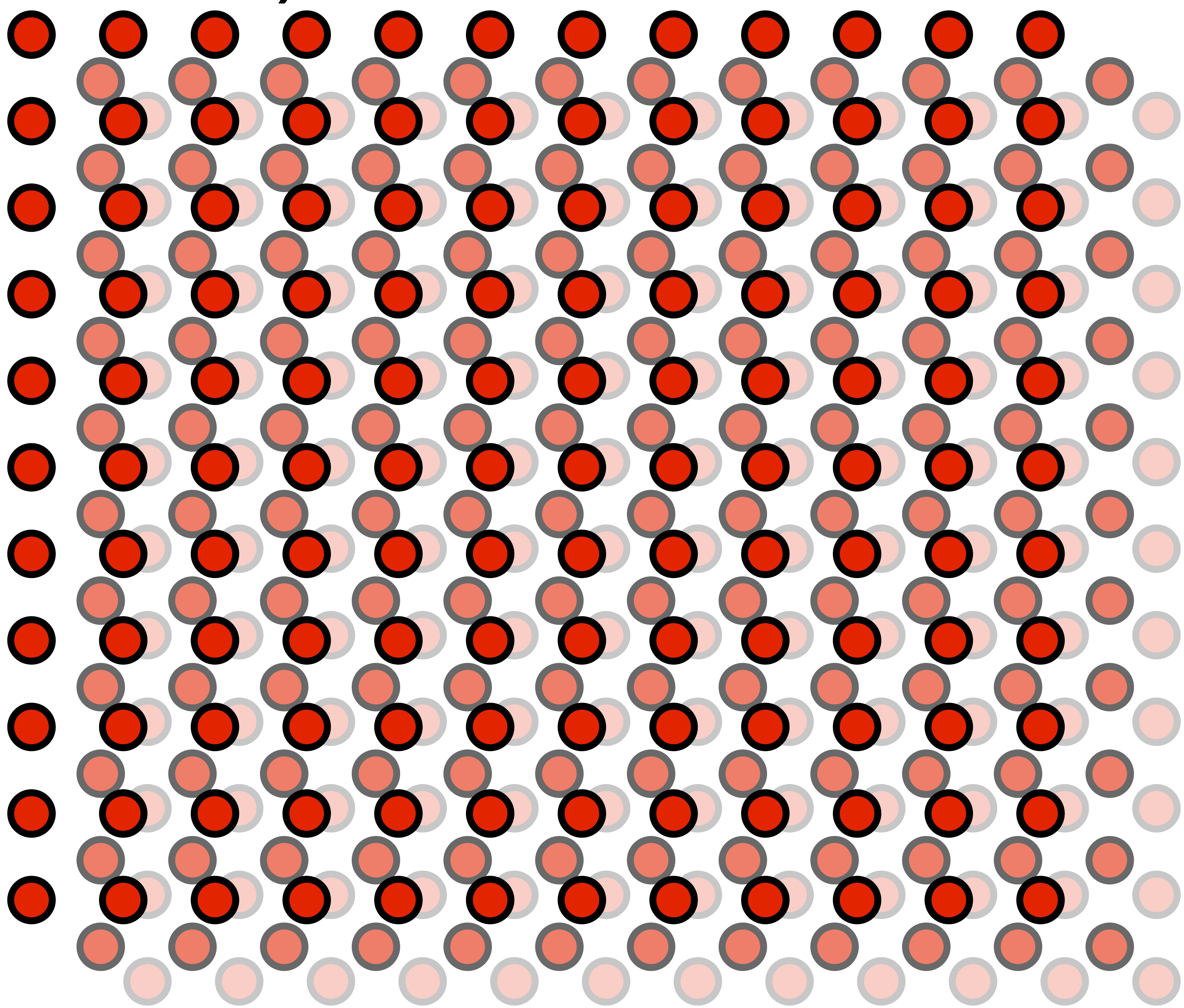
$$n^2$$

3D (YIKES!)



n^3

3D (YIKES!)

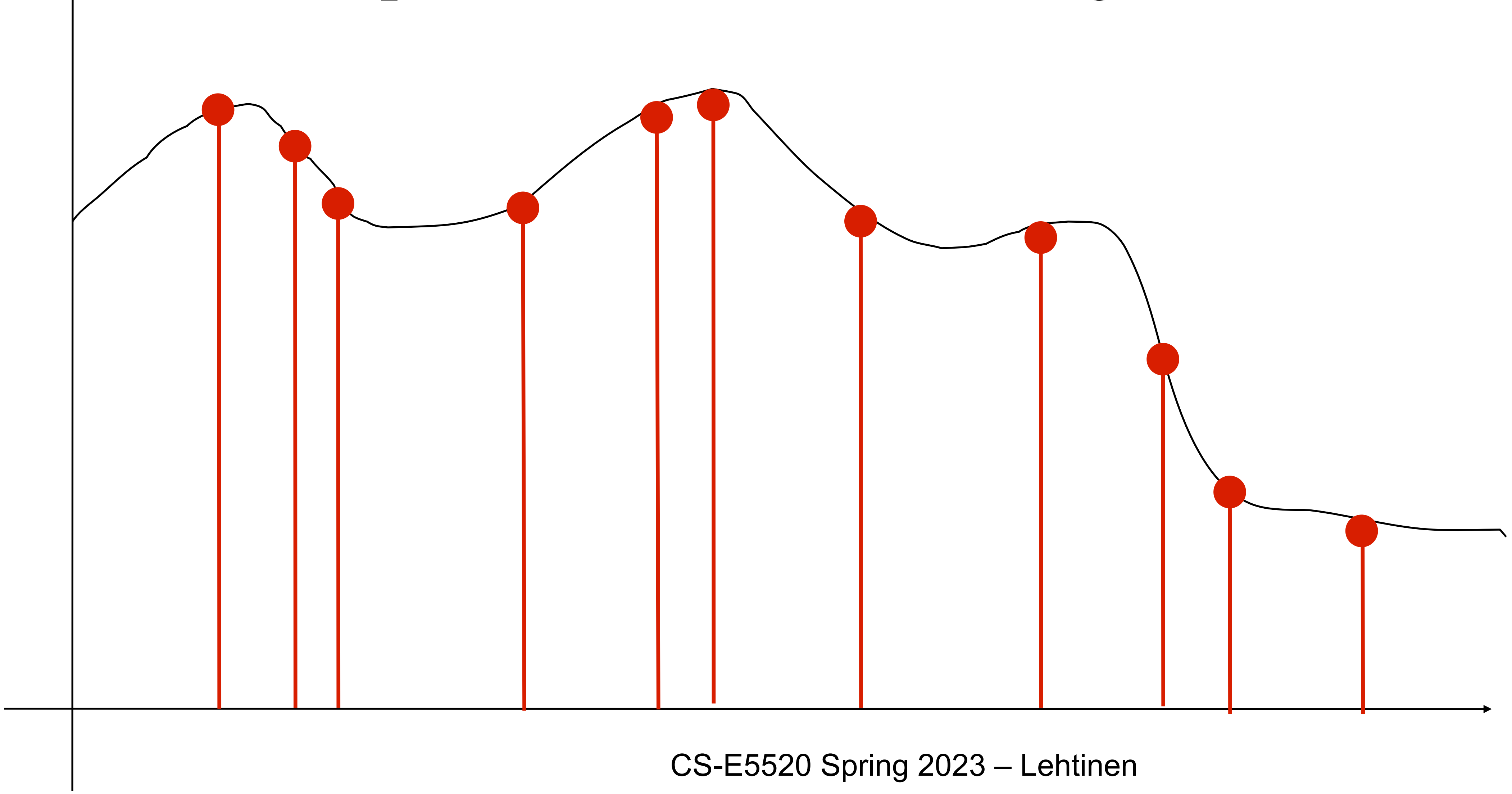


n^3

4D... you get the picture

Monte Carlo Integration

- Monte Carlo integration: use random samples and compute average
  - We don't keep track of spacing between samples
  - But we hope it will be 1/N on average



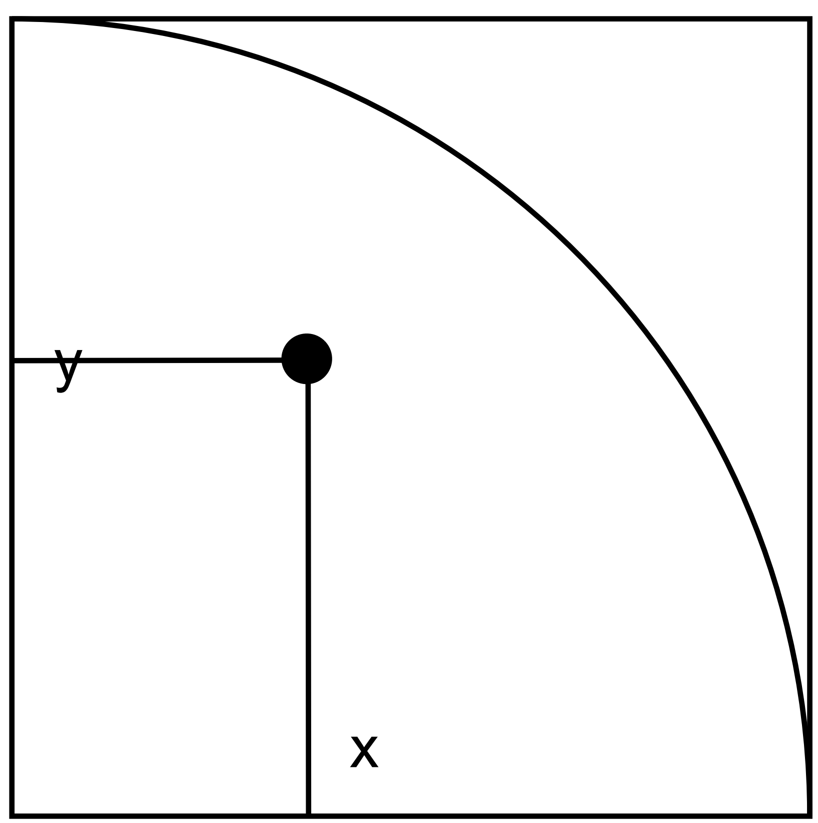
Naive Monte Carlo Integration

$$\int_S f(x)\,\mathrm{d}x \approx \frac{\mathrm{Vol}(S)}{N}\sum_{i=1}^{N} f(x_i)$$

- S is the integration domain
  - Vol(S) is the volume (measure) of S (1D: length, 2D: area, ...)
- $\{x_i\}$ are *independent, uniform* random points in S
- That's right: integral is average of *f* multiplied by size of domain
  - We estimate the average by random sampling
  - E.g. for hemisphere Vol(S) = 2pi

Naive Monte Carlo Computation of π

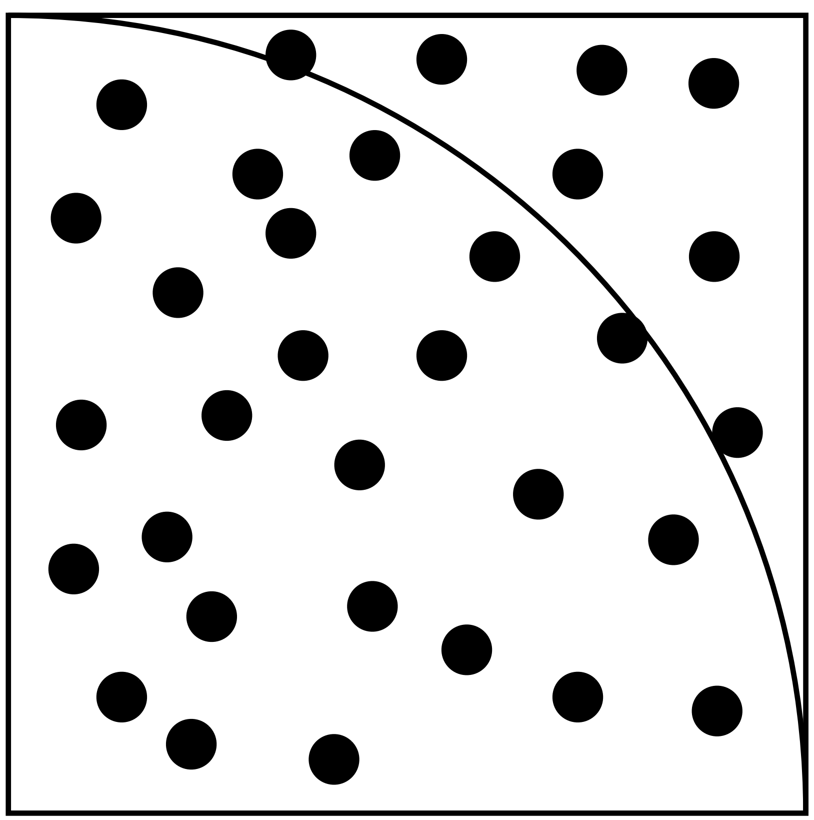
- Take a square
- Take a random point (x,y) in the square
- Test if it is inside the ¼ disc ($x^2+y^2 < 1$)
- The probability is $\pi /4$



Integral of the function that is one inside the circle, zero outside

Naive Monte Carlo Computation of π

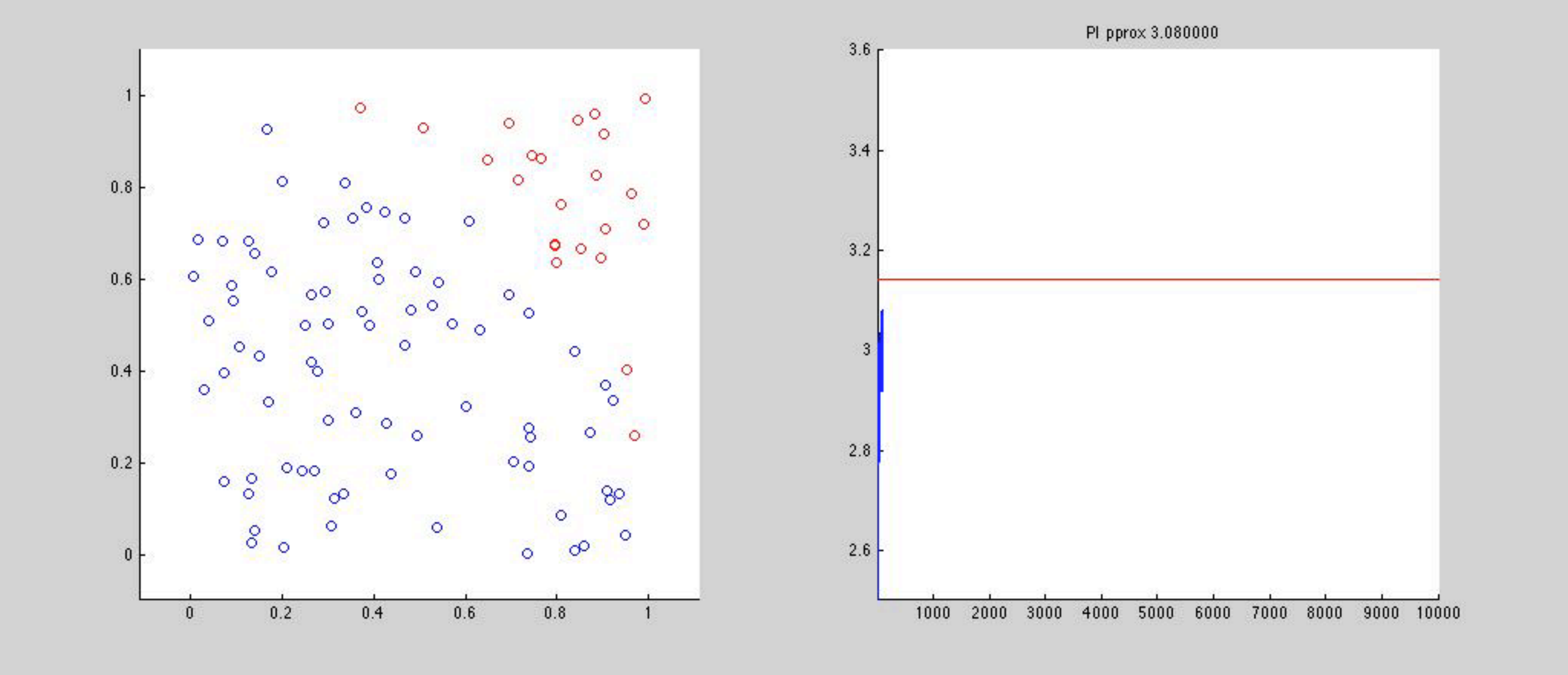
- The probability is π /4
- Count the inside ratio n = # inside / total # trials
- $\pi \approx n * 4$
- The error depends on the number or trials



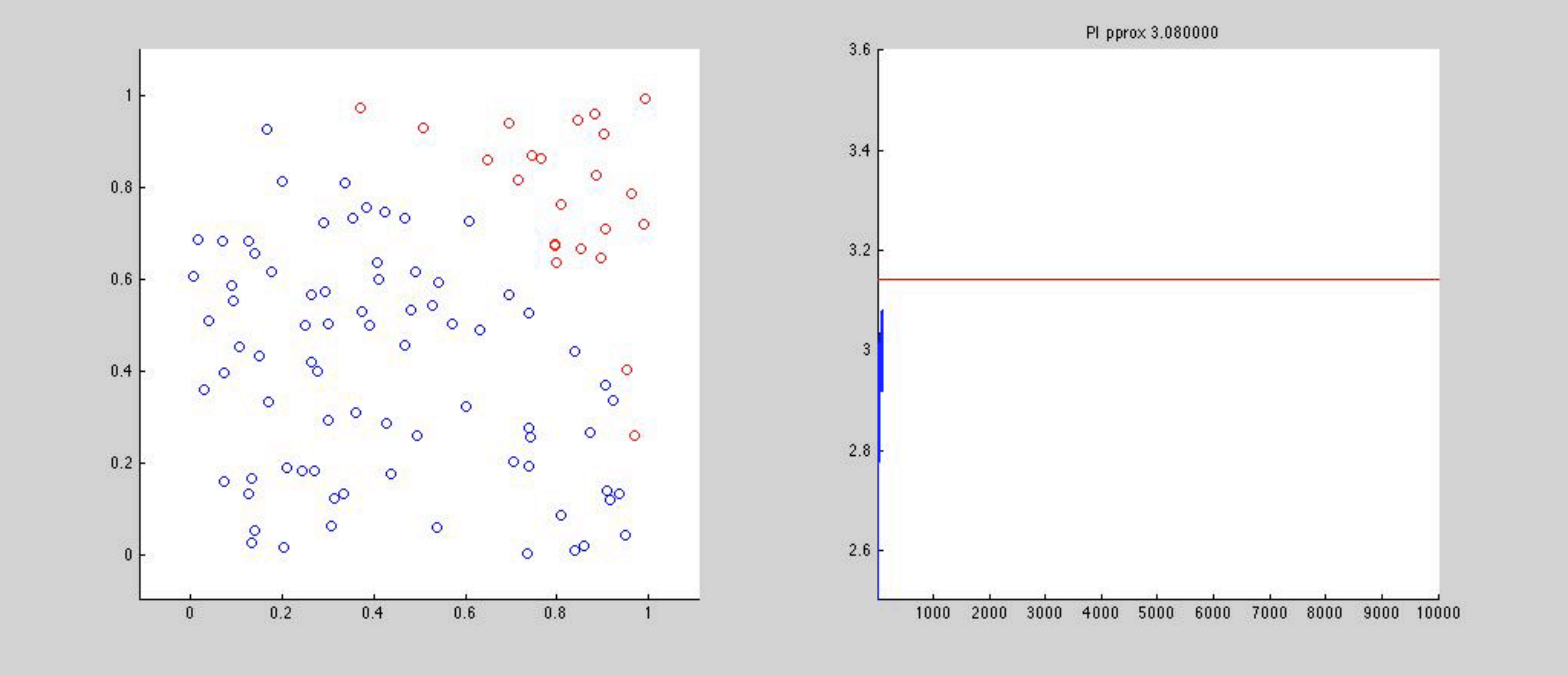
Demo

```
def piMC(n):
    success = 0
    for i in range(n):
        x=random.random()
        y=random.random()
        if x*x+y*y<1: success = success+1
    return 4.0*float(success)/float(n)
```

Matlab Demo



Matlab Demo



Why Not Use Simpson Integration?

- You're right, Monte Carlo is not very efficient for computing π

- So *when* is it useful? High dimensions!
  - Asymptotic convergence rate is independent of dimension!
  - For d dimensions, Simpson requires N^d domains (!!!)
    - Similar explosion for other quadratures (Gaussian, etc.)
  - You saw this visually a little earlier

Asymptotic convergence rate =
the relationship of error to number of samples n *when n is large*

Random Variables Recap

- You know this from your basic probability classes
  - Gentle reminder follows..

Random Variables Recap: PDF

- Distribution of random points determined by the Probability Density Function (PDF) $p(x)$
  - Uniform distribution means: each point in the domain equally likely to be picked: $p(x) = 1/\text{Vol}(S)$
  - Why so? PDF must integrate to 1 over S
  - (Uniform distribution is often pretty bad for integration)

Recap: Expected Value (=Average)

- Expected value of a function g under probability distribution p is defined as

$$E\{g(x)\}_p = \int_S g(x)\, p(x)\, \mathrm{d}x$$

- Because p integrates to 1 like a proper PDF should, this is just a weighted average of g over S
  - When p is uniform, this reduces to the usual average

$$\frac{1}{\mathrm{Vol}(S)} \int_S g(x)\, \mathrm{d}x$$

Random Variables Recap: Variance

- Variance is the average (expected) squared deviation from the mean $\mu = E\{X\}_p$

$$\mathrm{Var}(X) = E\{(X-\mu)^2\}_p$$

- Standard deviation is square root of variance

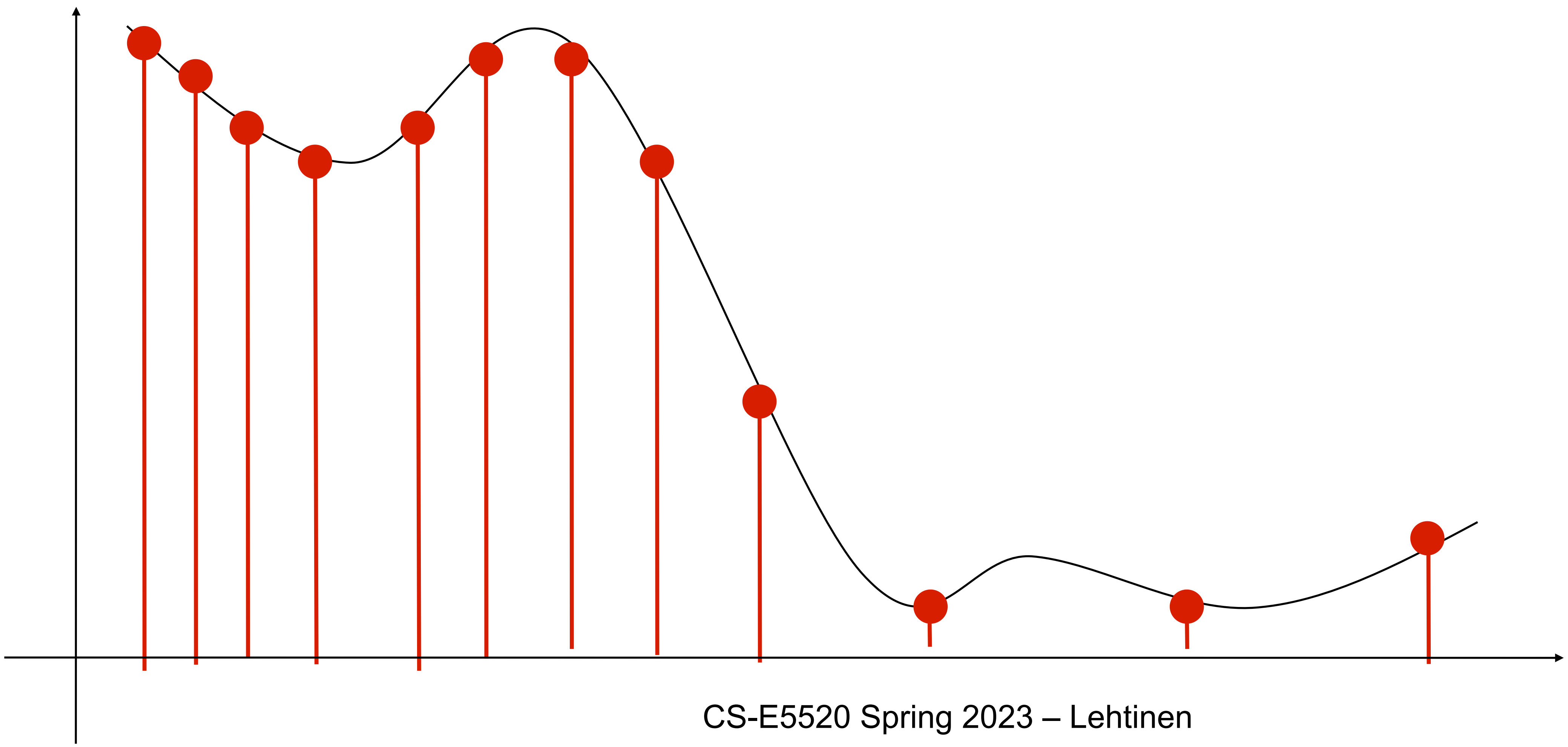
- Note that the PDF p is included in the definition!
  - Also in the computation of the mean

OK, Down to Business Then!

“Importance Sampling”

Sample from non-uniform PDF

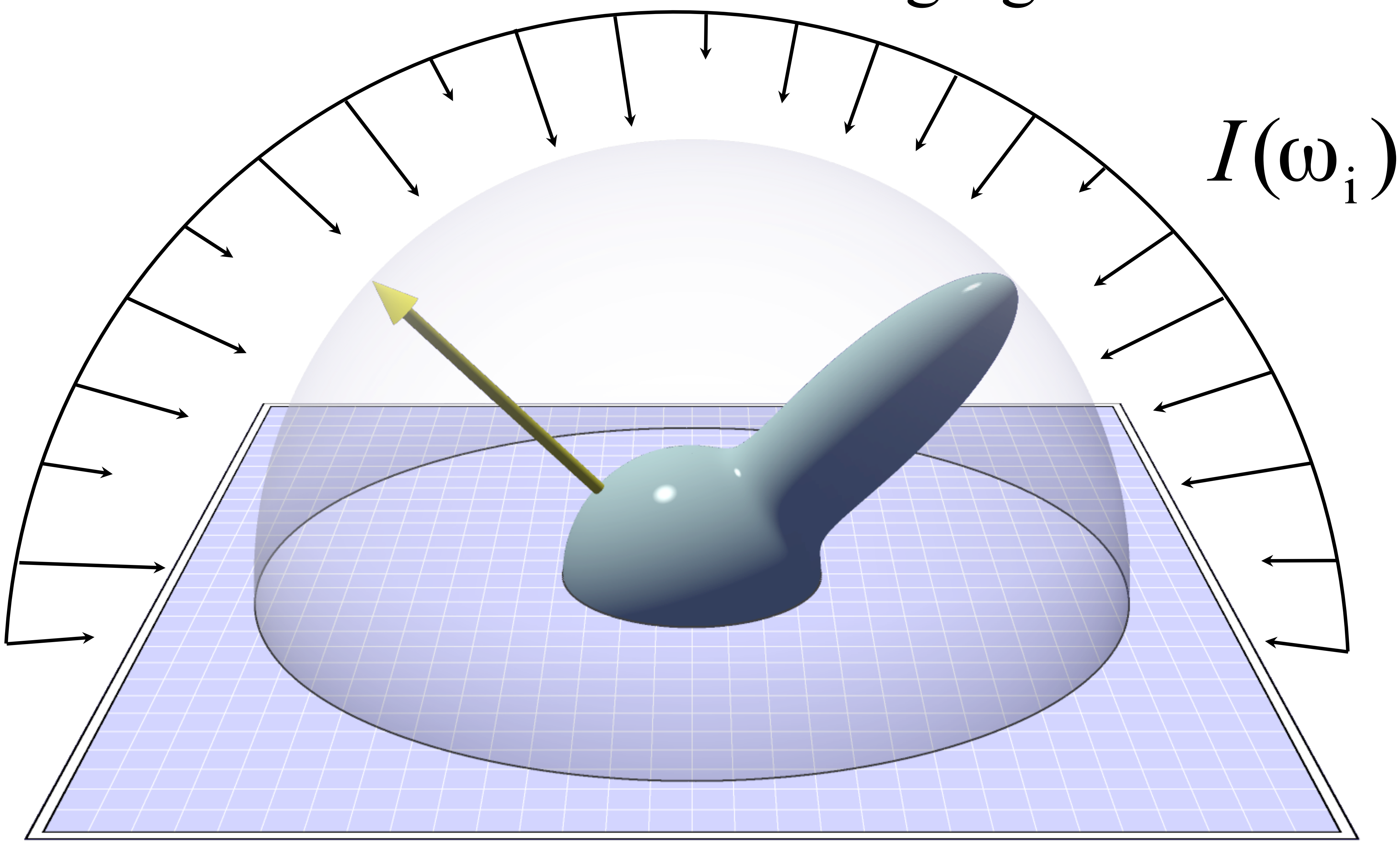
- Intuitive justification: Sample more in places where there are likely to be larger contributions to the integral



Example: Glossy Reflection

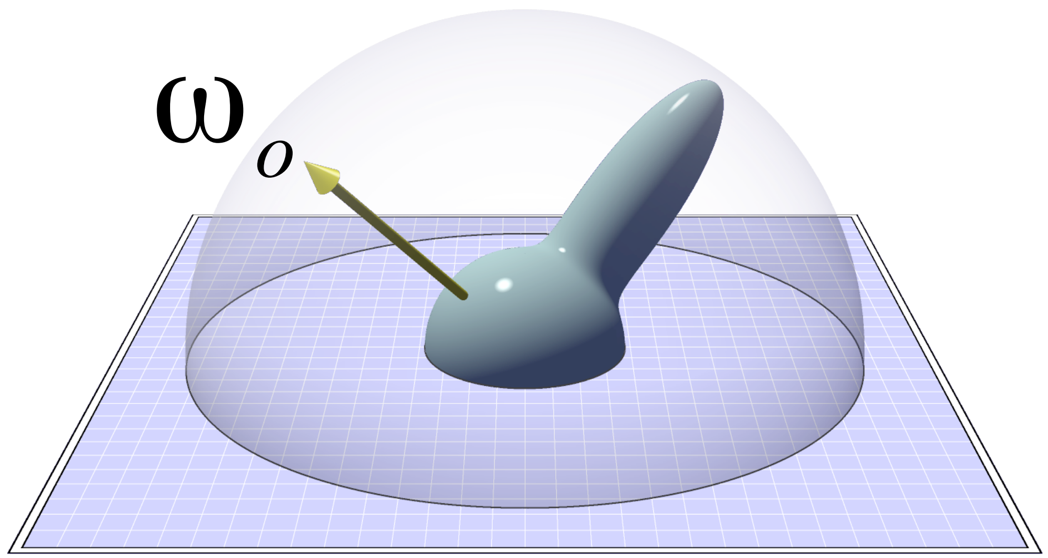
Slide courtesy of Jason Lawrence

- Integral over hemisphere
- BRDF times cosine times incoming light



Sampling a BRDF

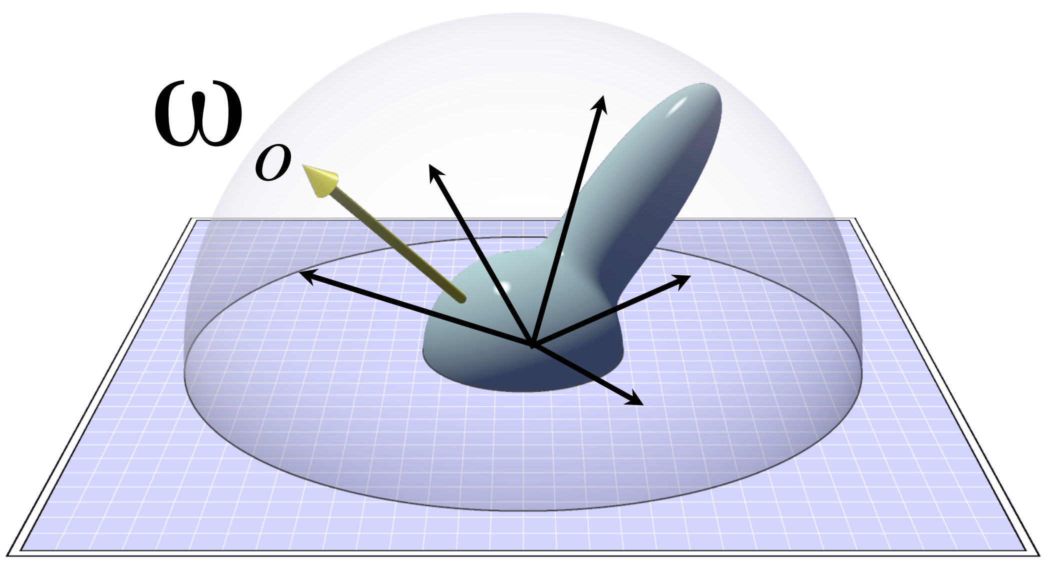
Slide courtesy of Jason Lawrence



Sampling a BRDF

Slide courtesy of Jason Lawrence

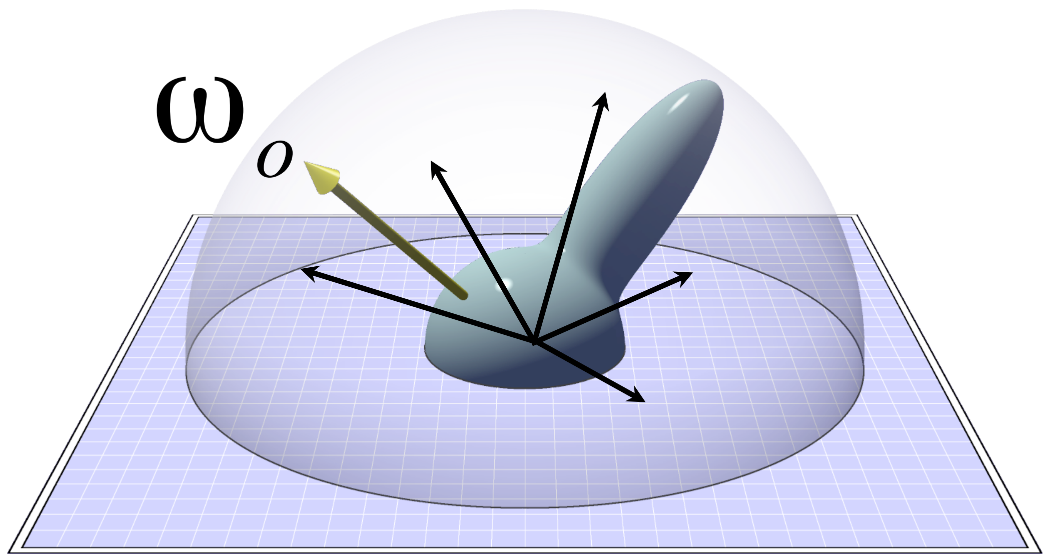
$U(\omega_i)$



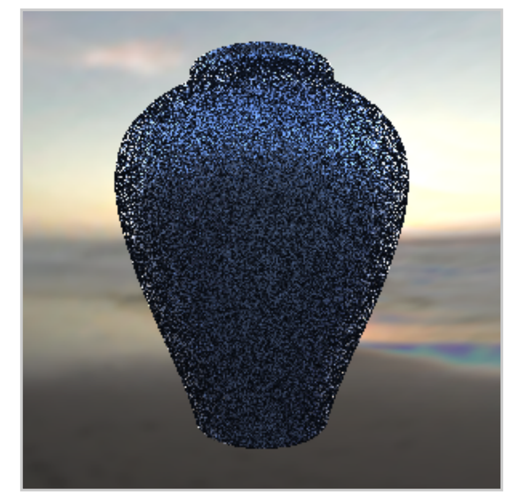
Sampling a BRDF

Slide courtesy of Jason Lawrence

$U(\omega_i)$



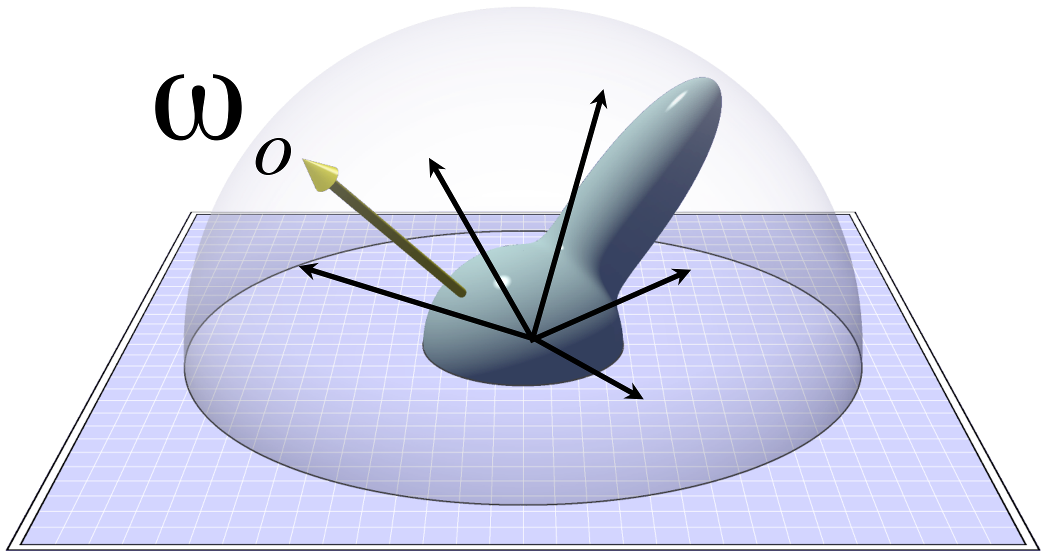
5 Samples/Pixel



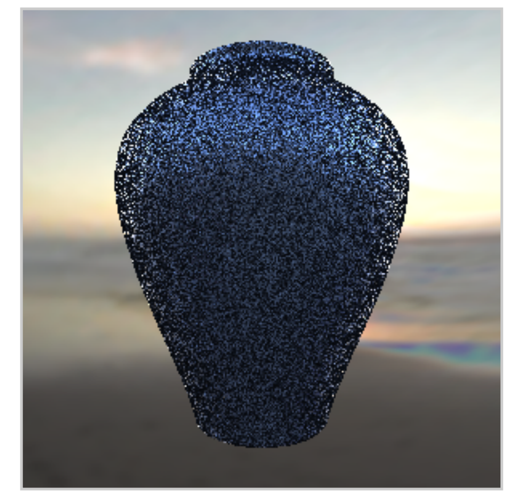
Sampling a BRDF

Slide courtesy of Jason Lawrence

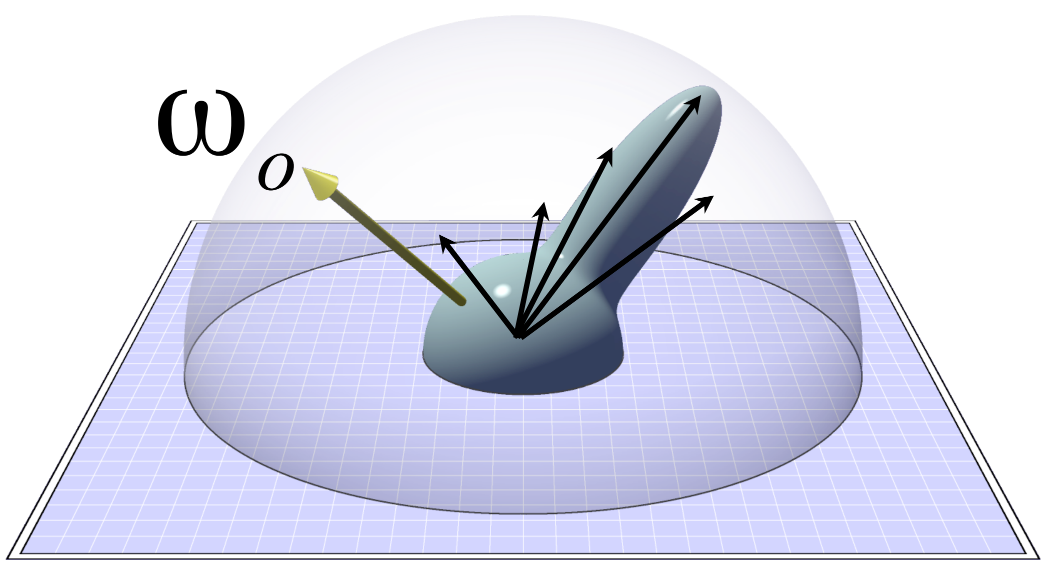
$U(\omega_i)$



5 Samples/Pixel



$P(\omega_i)$

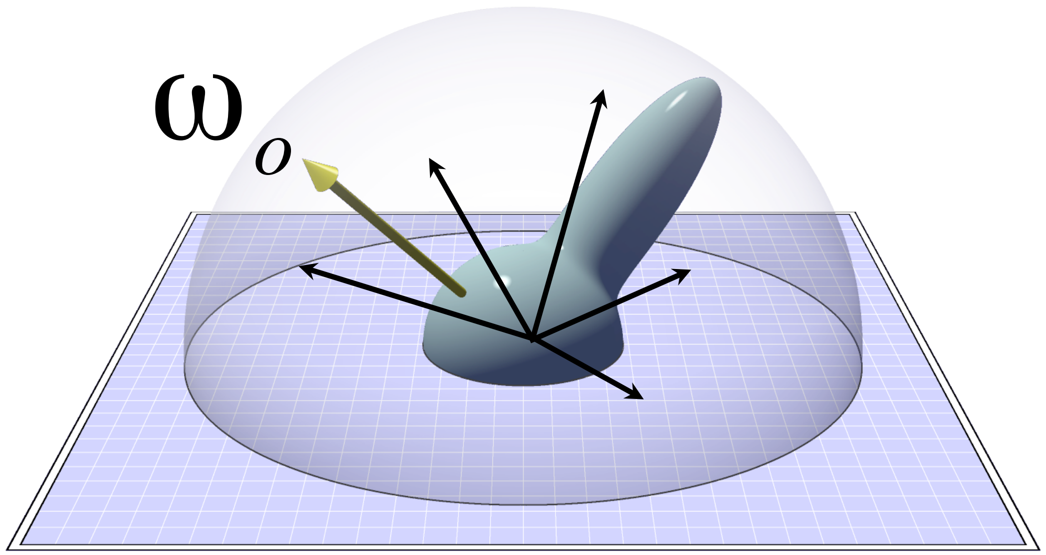


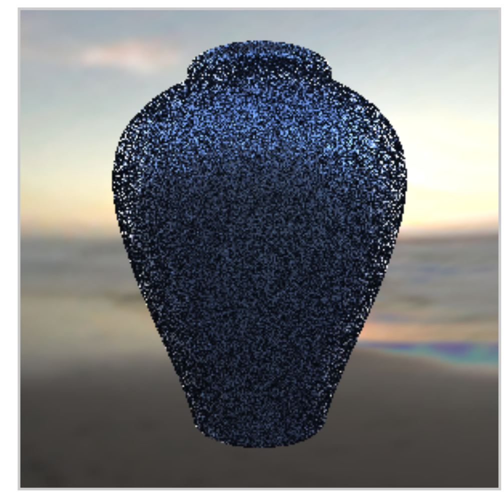
Sampling a BRDF

Slide courtesy of Jason Lawrence

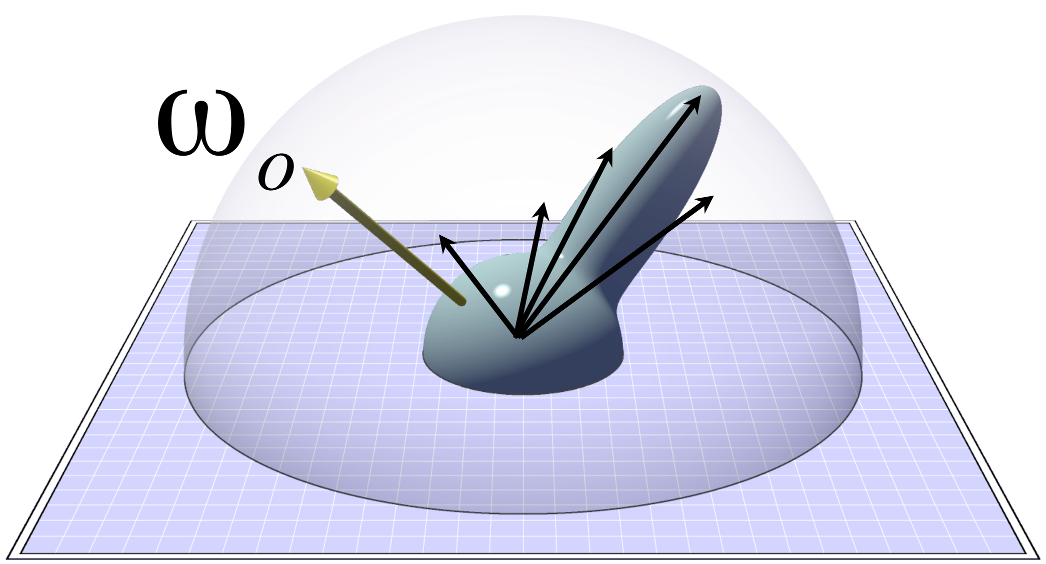
5 Samples/Pixel

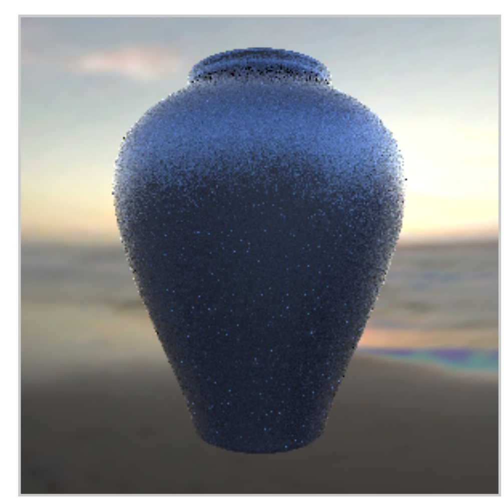
$U(\omega_i)$





$P(\omega_i)$

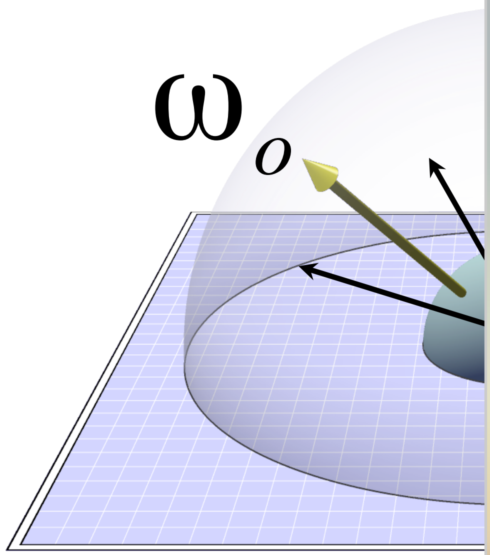




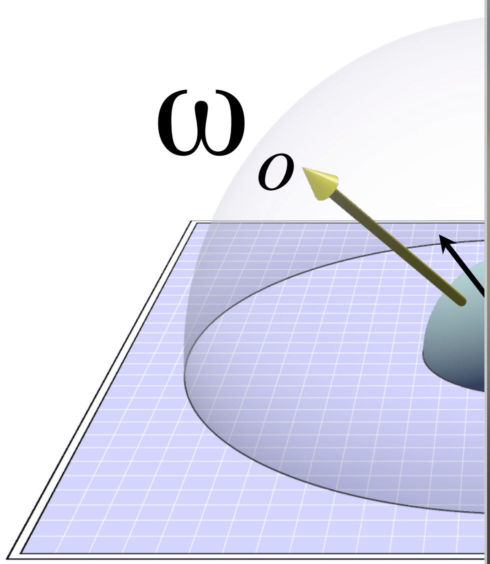
Sampling a BRDF

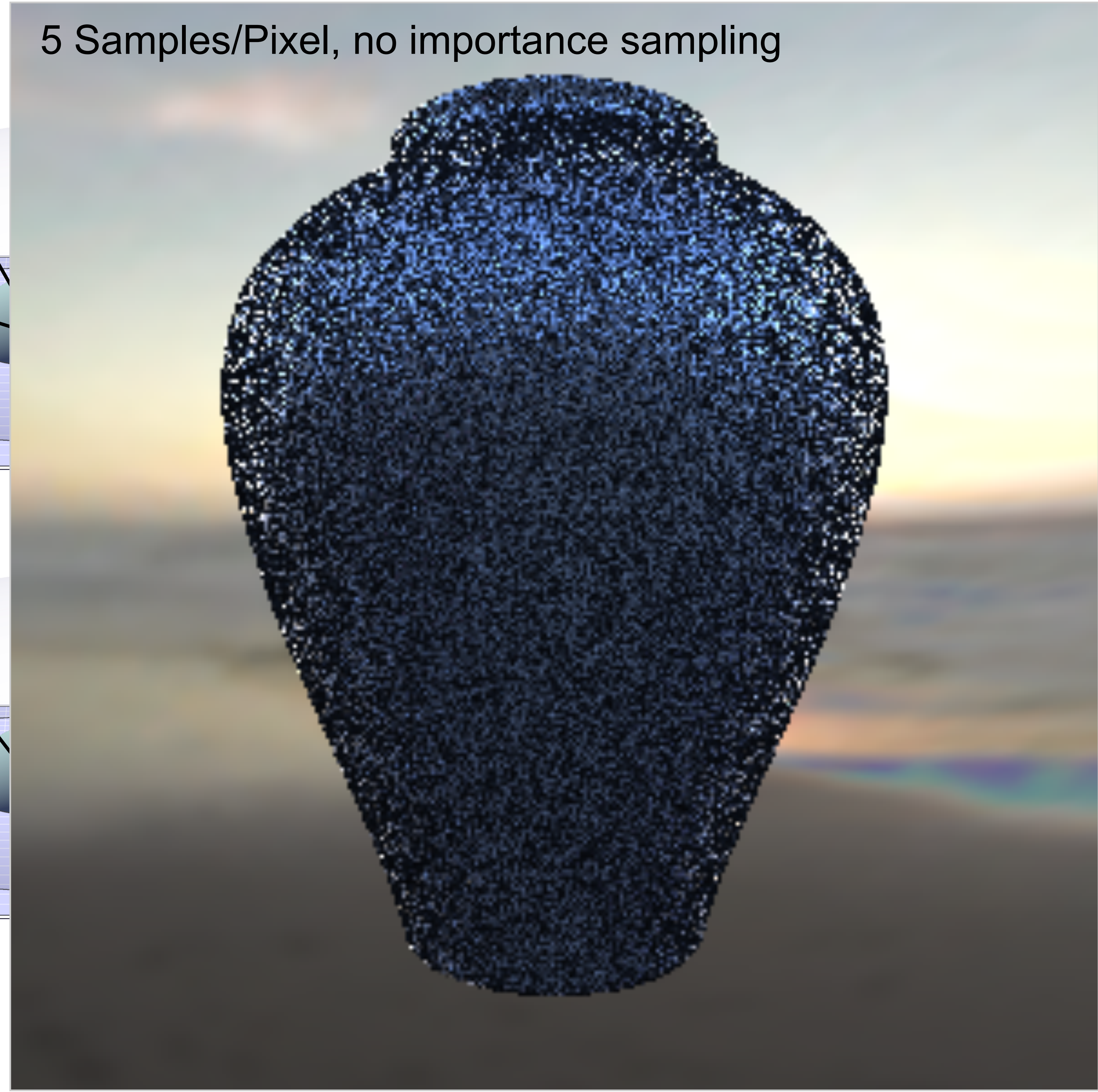
Slide modified from Jason Lawrence's

$U(\omega_i)$



$P(\omega_i)$



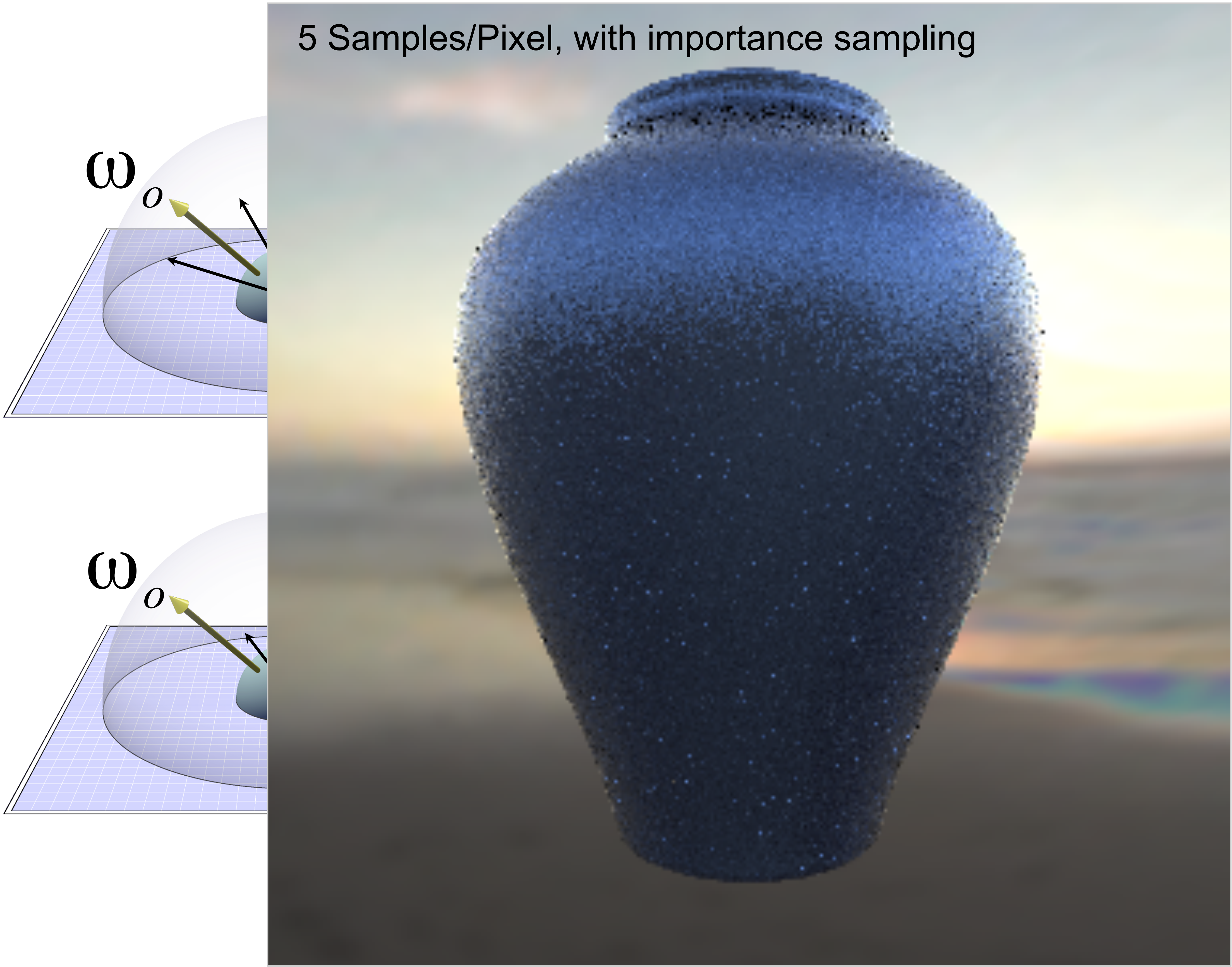


Sampling a BRDF

Slide modified from Jason Lawrence's

$U(\omega_i)$

$P(\omega_i)$

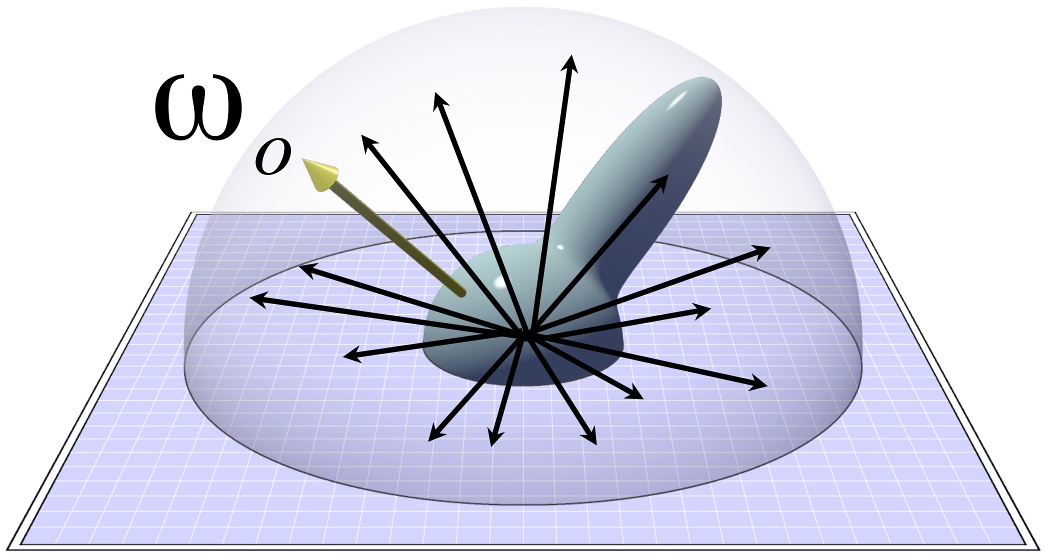


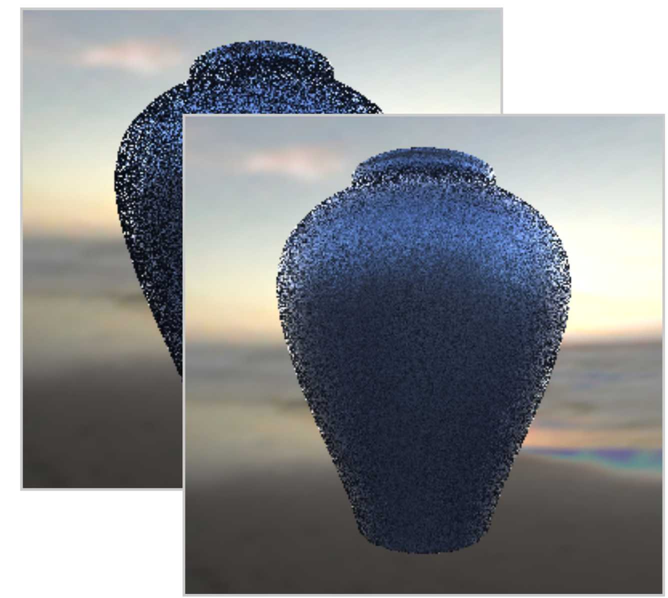
Sampling a BRDF

Slide courtesy of Jason Lawrence

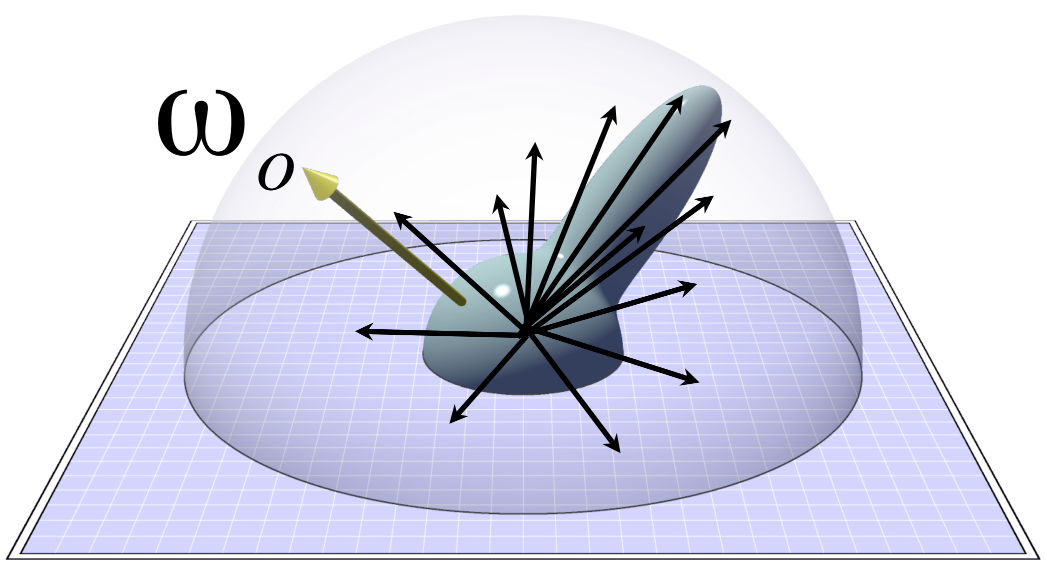
25 Samples/Pixel

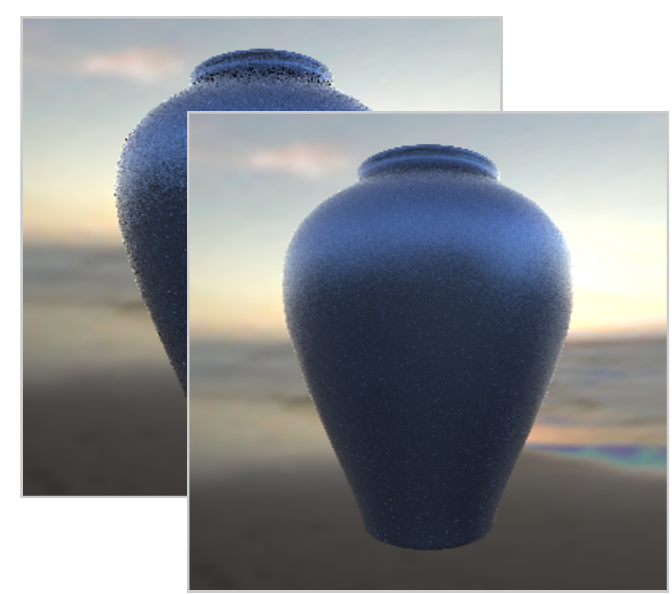
$U(\omega_i)$





$P(\omega_i)$



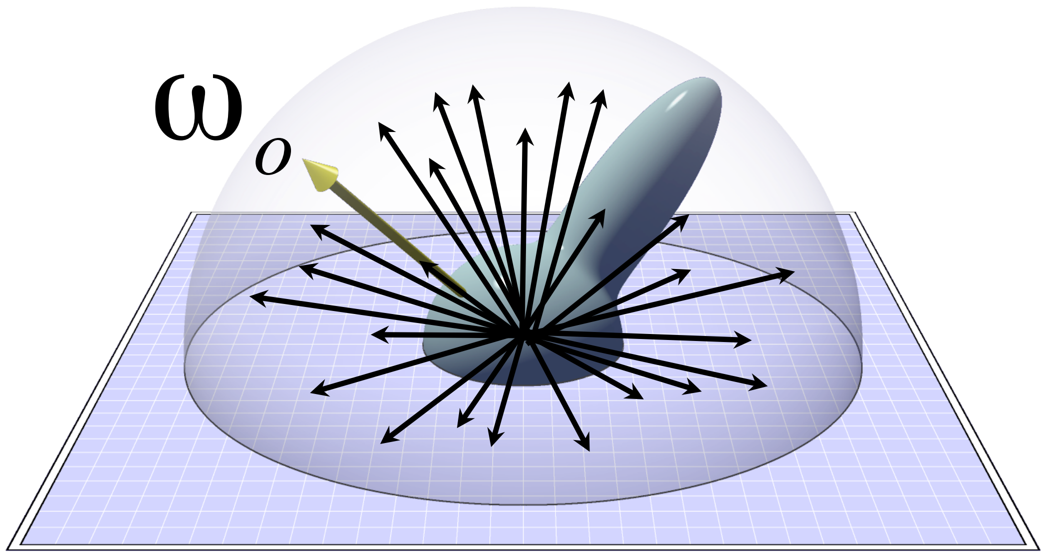


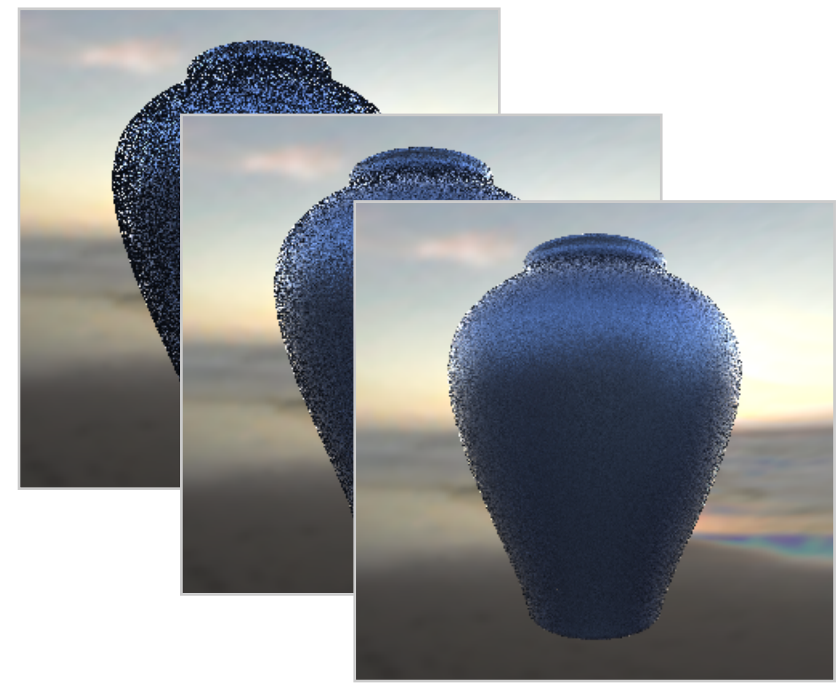
Sampling a BRDF

Slide courtesy of Jason Lawrence

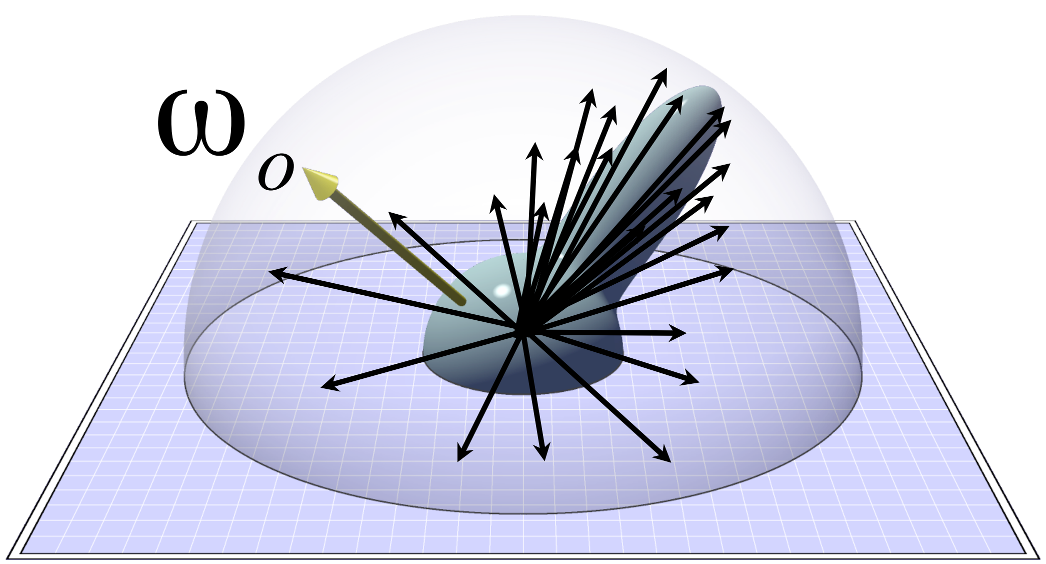
75 Samples/Pixel

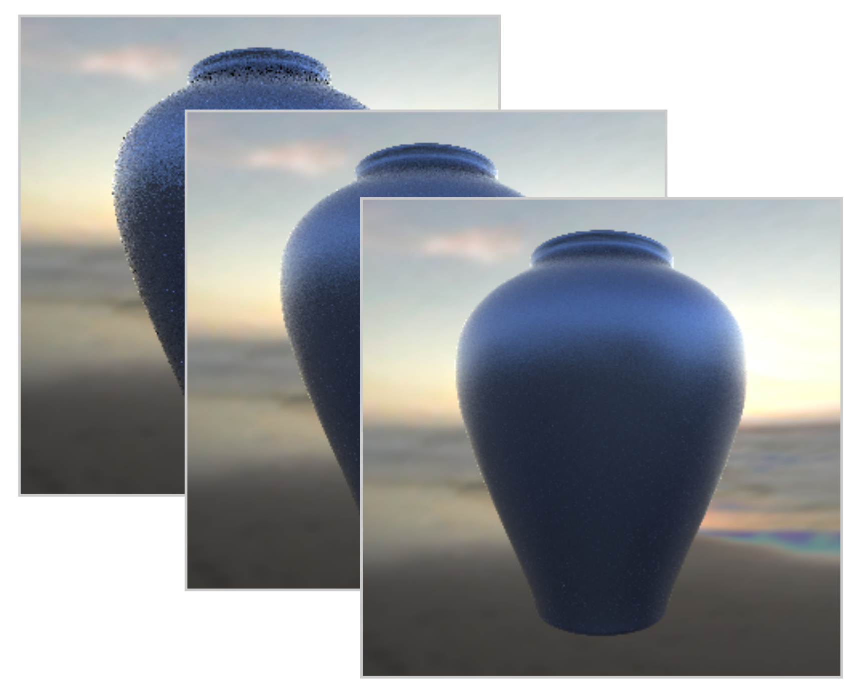
$U(\omega_i)$





$P(\omega_i)$

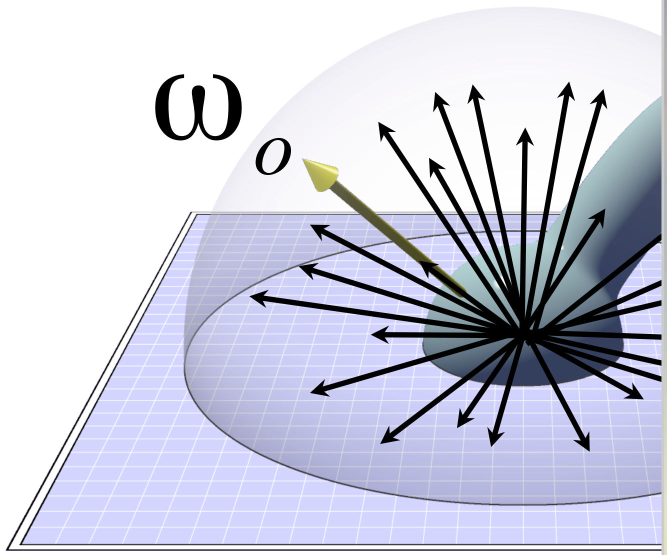




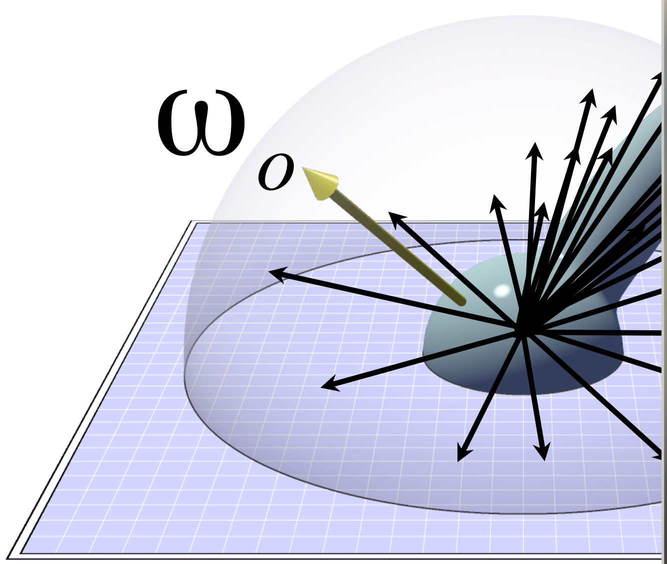
Sampling a BRDF

Slide modified from Jason Lawrence's

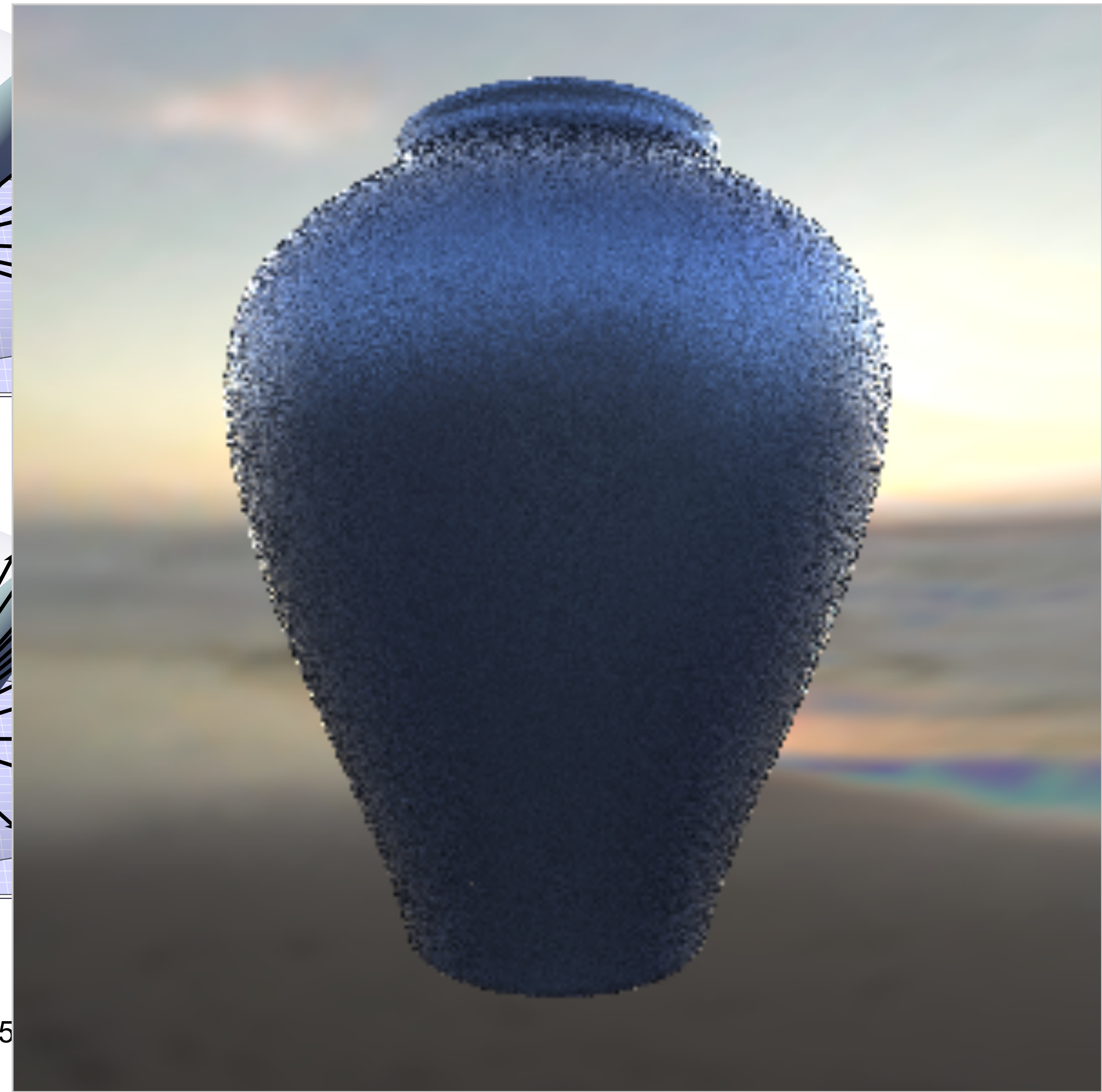
$U(\omega_i)$



$P(\omega_i)$



75 Samples/Pixel, no importance sampling

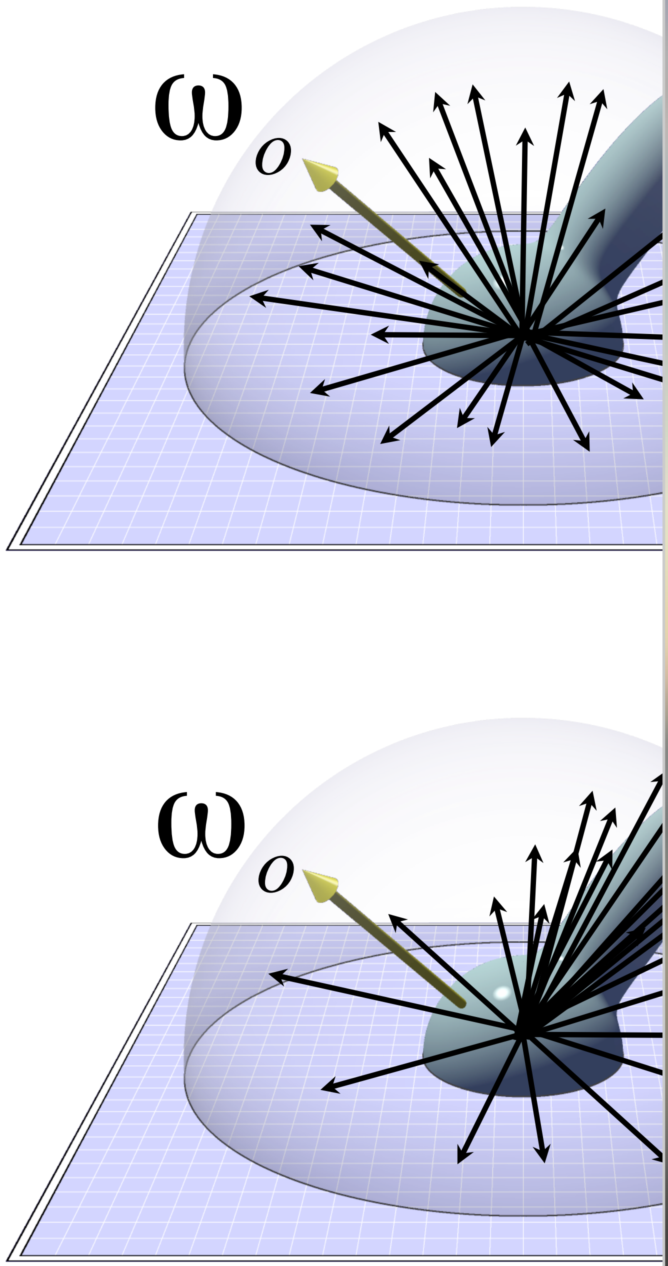


Sampling a BRDF

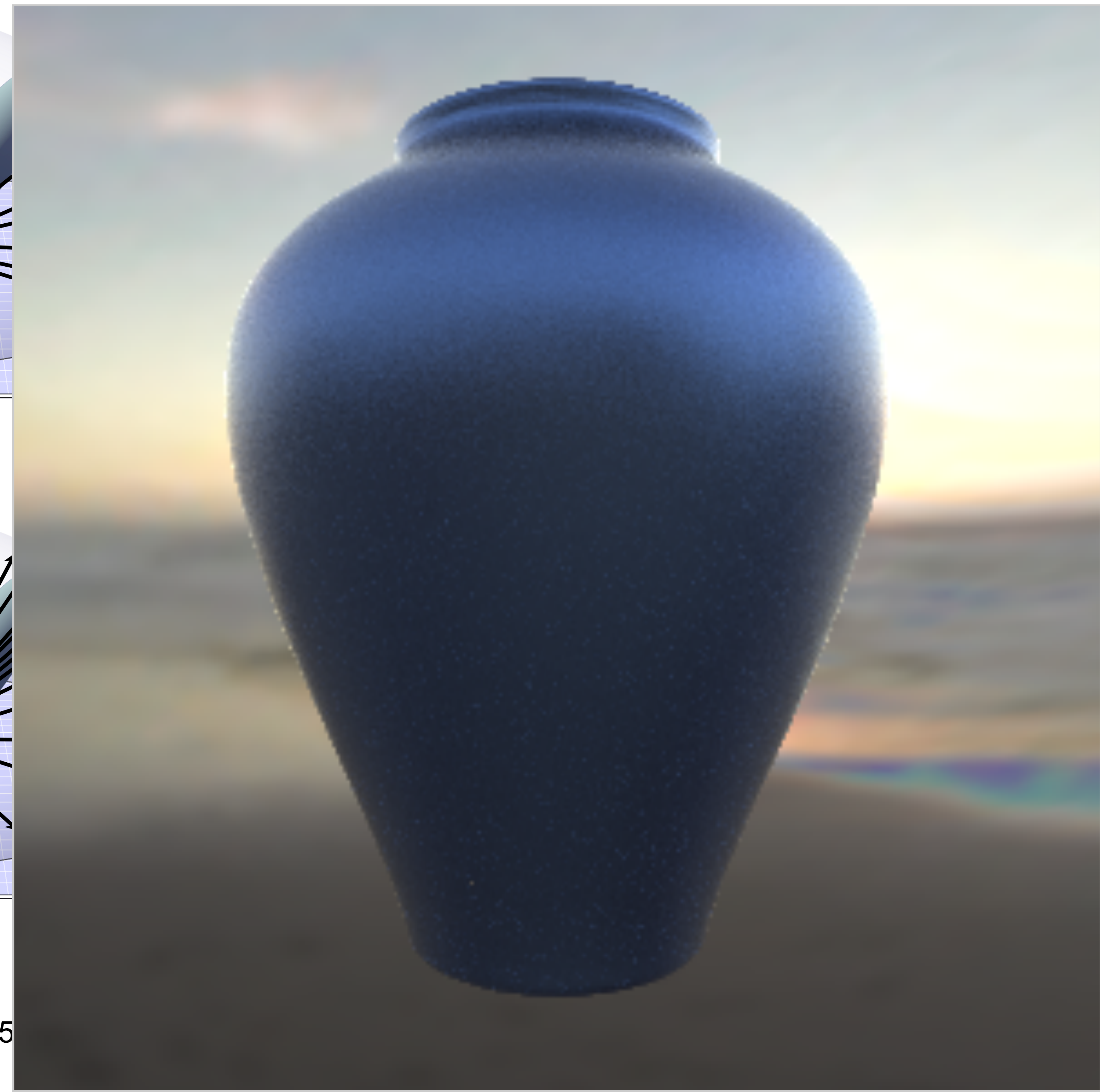
Slide modified from Jason Lawrence's

$U(\omega_i)$

$P(\omega_i)$



75 Samples/Pixel, with importance sampling



How does that work?

- Sample density changes over domain S ~ p(x) is not a constant any more

How does that work?

- Sample density changes over domain S ~ p(x) is not a constant any more
- So let's drop the uniform PDF requirement and rewrite:

$$\boxed{\int_S f(x)\,\mathrm{d}x = \int_S \frac{f(x)}{p(x)}\,p(x)\,\mathrm{d}x}$$

- **Important**! p(x) must be nonzero where f(x) is nonzero!

Non-Naive MC Integration

- This is (by definition) the expectation of f(x)/p(x):

$$\int_S f(x)\,\mathrm{d}x = \int_S \frac{f(x)}{p(x)}\,p(x)\,\mathrm{d}x$$

$$= E\{\frac{f(x)}{p(x)}\}_p$$

Non-Naive MC Integration

- …and this is how one estimates it numerically

$$\int_S f(x)\,\mathrm{d}x = \int_S \frac{f(x)}{p(x)}\,p(x)\,\mathrm{d}x$$

$$= E\{\frac{f(x)}{p(x)}\}_p$$

The *x_i* are independent random points distributed with density *p(x)*

$$\boxed{\approx \frac{1}{N}\sum_i \frac{f(x_i)}{p(x_i)}}$$

Note that the uniform case reduces to the same because p(x)==1/Vol(S)

This is called *Importance Sampling*

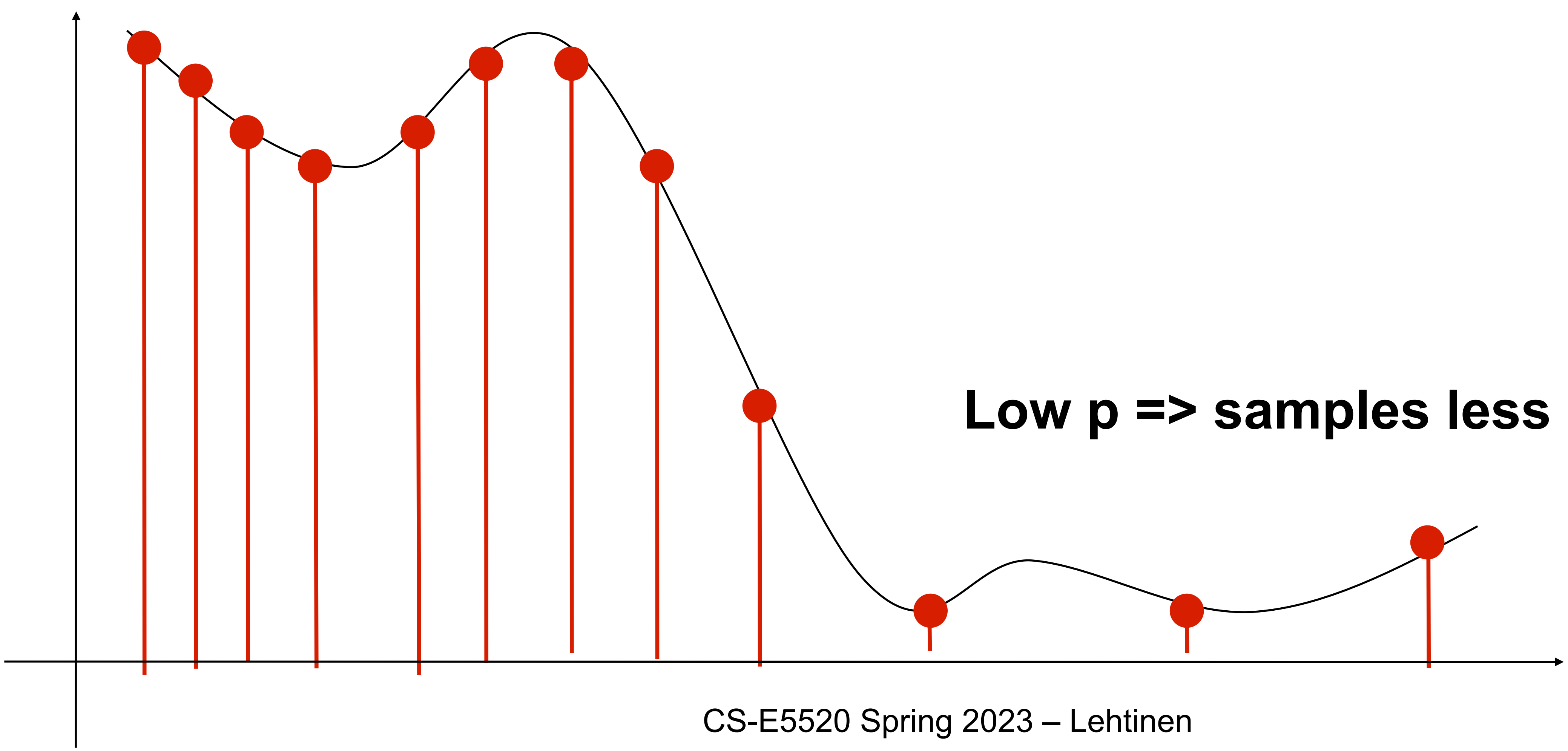
$$\int_S f(x)\,\mathrm{d}x \approx \frac{1}{N}\sum_i \frac{f(x_i)}{p(x_i)}$$

1. Draw random samples distributed with density *p*
2. Evaluate integrand f(x) and p(x) at the samples
3. Average f(x)/p(x)

Let's think about this…

$$\int_S f(x)\,\mathrm{d}x \approx \frac{1}{N}\sum_i \frac{f(x_i)}{p(x_i)}$$

High p => samples more dense

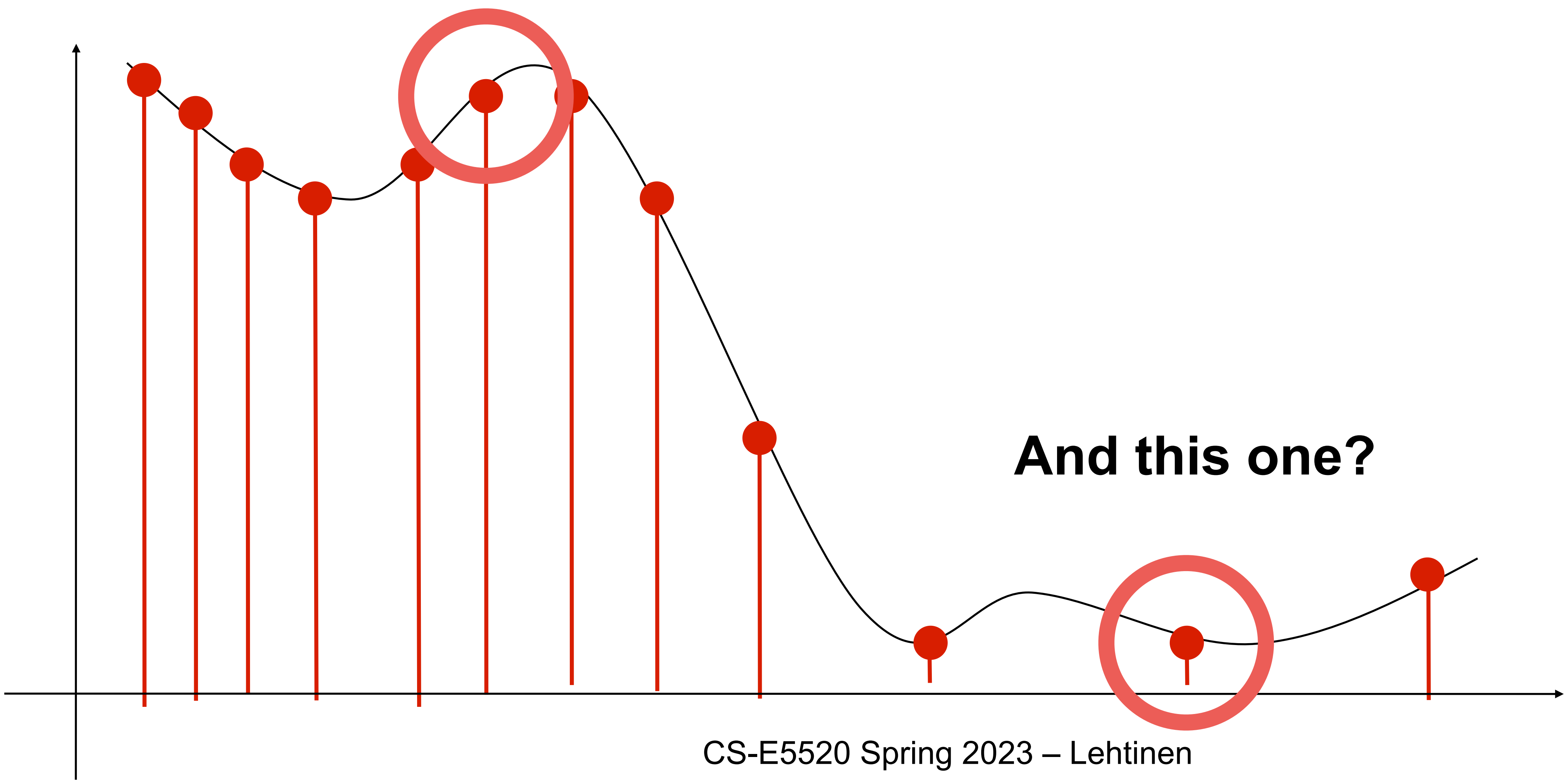


Low p => samples less dense

Let's think about this…

$$\int_S f(x)\,\mathrm{d}x \approx \frac{1}{N}\sum_i \frac{f(x_i)}{p(x_i)}$$

How does this sample contribute to the average?

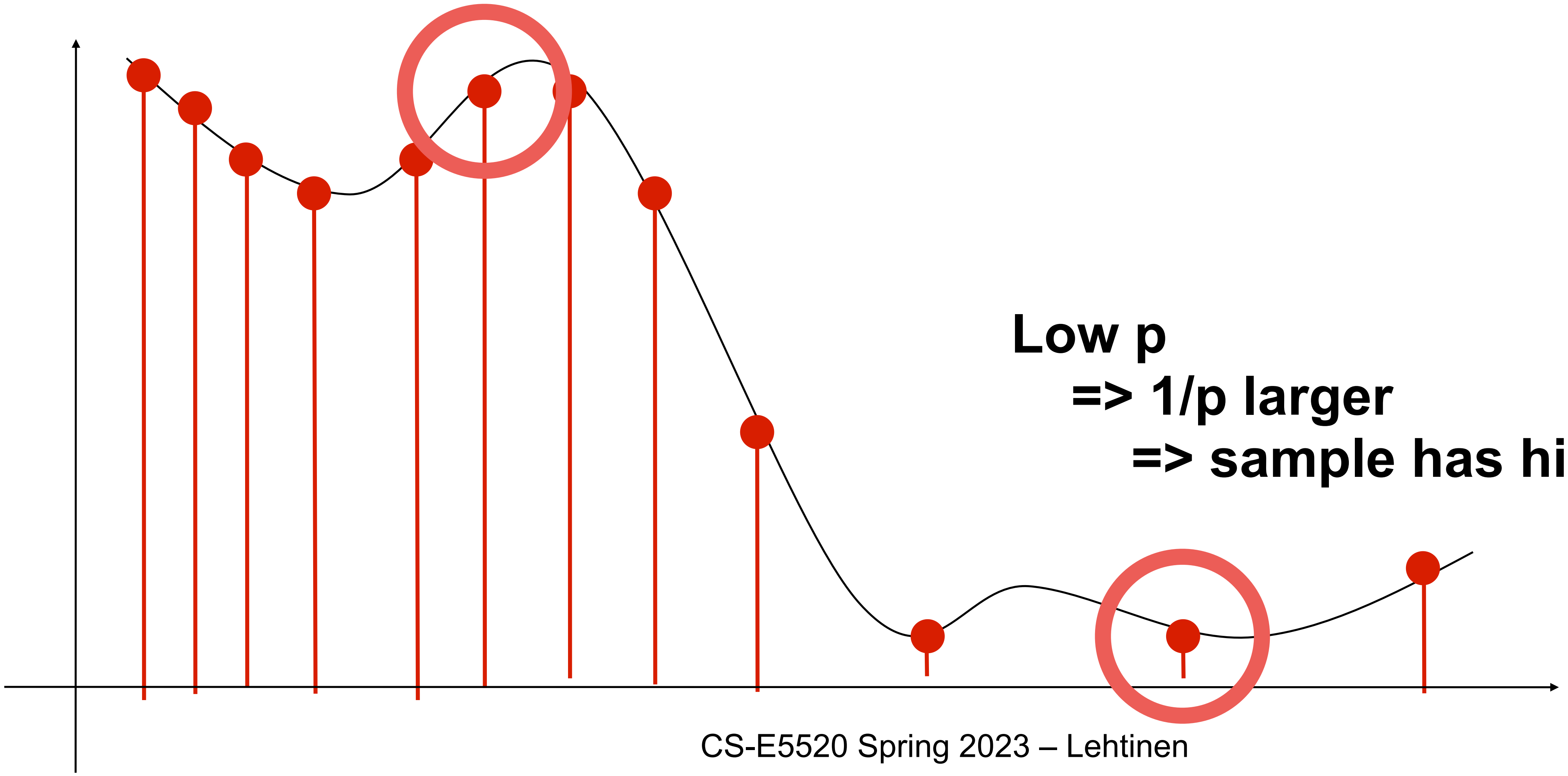


And this one?

Let's think about this…

$$\int_S f(x)\,\mathrm{d}x \approx \frac{1}{N}\sum_i \frac{f(x_i)}{p(x_i)}$$

High p
=> 1/p smaller
=> sample has less weight



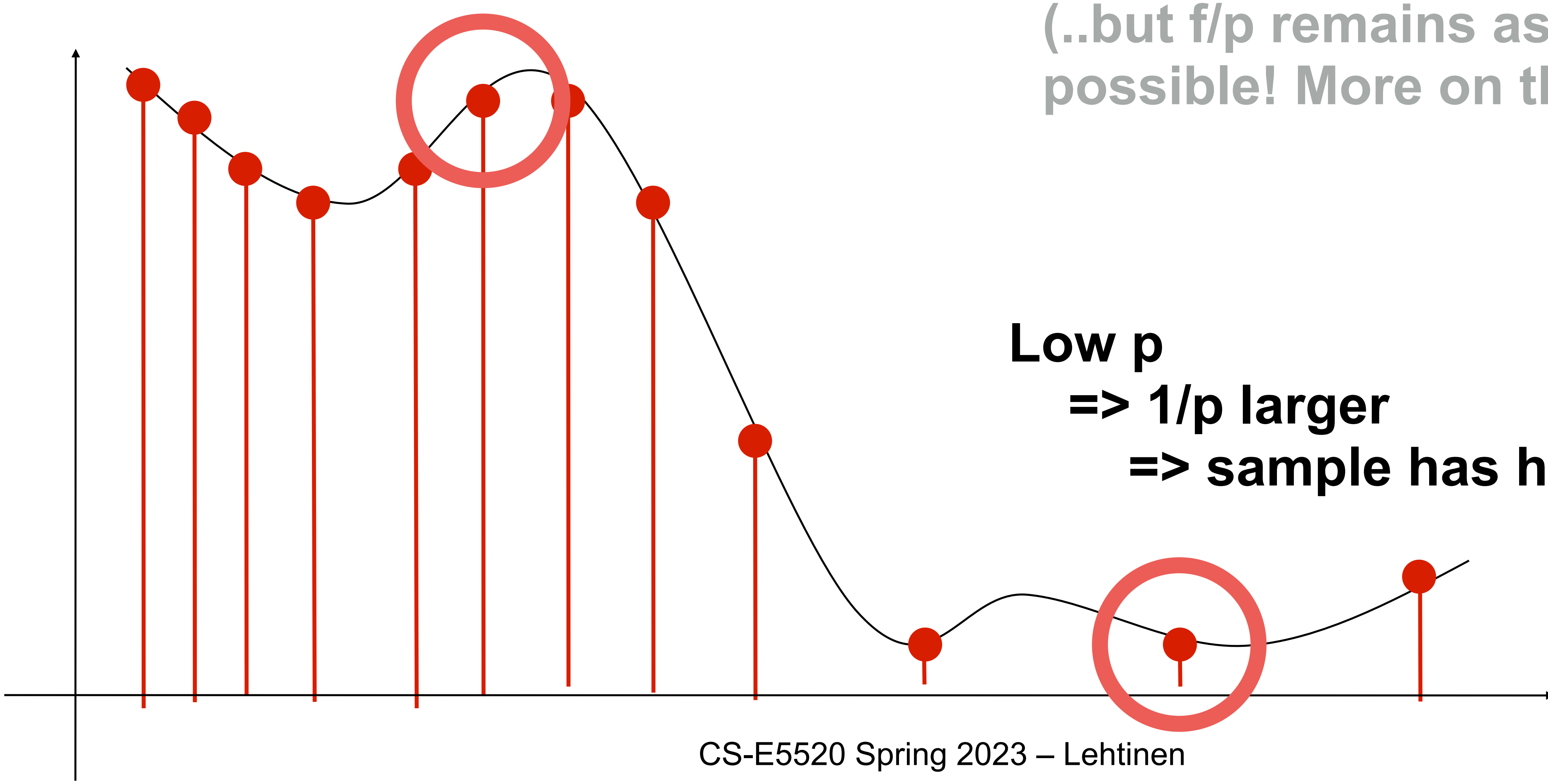
Low p
=> 1/p larger
=> sample has higher weight

Let's think about this…

"If you pick a sample less often, give it more power"

High p
=> 1/p smaller
=> sample has less weight

(..but f/p remains as constant as possible! More on this later.)



Low p
=> 1/p larger
=> sample has higher weight

Monte Carlo Integration Error

$$\int_S f(x)\,\mathrm{d}x \approx \frac{1}{N}\sum_i \frac{f(x_i)}{p(x_i)}$$

- Clearly this is just an approximation!

Monte Carlo Integration Error

$$I \stackrel{\text{def}}{=} \int_S f(x)\,\mathrm{d}x \approx \frac{1}{N}\sum_i \frac{f(x_i)}{p(x_i)} \stackrel{\text{def}}{=} \hat{I}$$

- Clearly this is just an approximation!
  - The value $\hat{I}$ of the estimate is a random variable itself
    - Because we are using random points
  - Error manifests itself as <u>variance</u>, which shows up as **noise**

Monte Carlo Integration Error

$$I \stackrel{\text{def}}{=} \int_S f(x)\,\mathrm{d}x \approx \frac{1}{N}\sum_i \frac{f(x_i)}{p(x_i)} \stackrel{\text{def}}{=} \hat{I}$$

- Clearly this is just an approximation!
  - The value $\hat{I}$ of the estimate is a random variable itself
  - Error manifests itself as <u>variance</u>, which shows up as **noise**
- **Variance of MC integration result $\hat{I}$ is proportional to both 1/N and the variance of *f/p***
  - Avg. error is proportional 1/sqrt(N)
  - To halve error, need 4x samples (!!) (avg. error = sqrt(Var))

Variance of the MC Result

- "Variance of $\hat{I}$ proportional to 1/N and $\boxed{\text{Var(f/p)"}}$

$$\mathrm{Var}(\hat{I}) = \frac{\mathrm{Vol}(S)^2}{N}\,\mathrm{Var(f/p)} = \frac{\mathrm{Vol}(S)^2}{N}\boxed{E\{(\frac{f(x)}{p(x)} - E\{f/p\})^2\}_p}$$

==>

If *f/p* is constant, there is no noise

– In practice: If we use a good PDF, we will have less noise...

What's a Good PDF?

- What if p mimics f perfectly? I.e., let's take

$$p(x) = \frac{f(x)}{\int_S f(x)\,\mathrm{d}x}$$

- This has the same shape as f, but normalized so it integrates to 1
  - Note: need non-negative f for this to work

What's a Good PDF?

- What if p mimics f perfectly? I.e., let's take

$$p(x) = \frac{f(x)}{\int_S f(x)\,\mathrm{d}x}$$

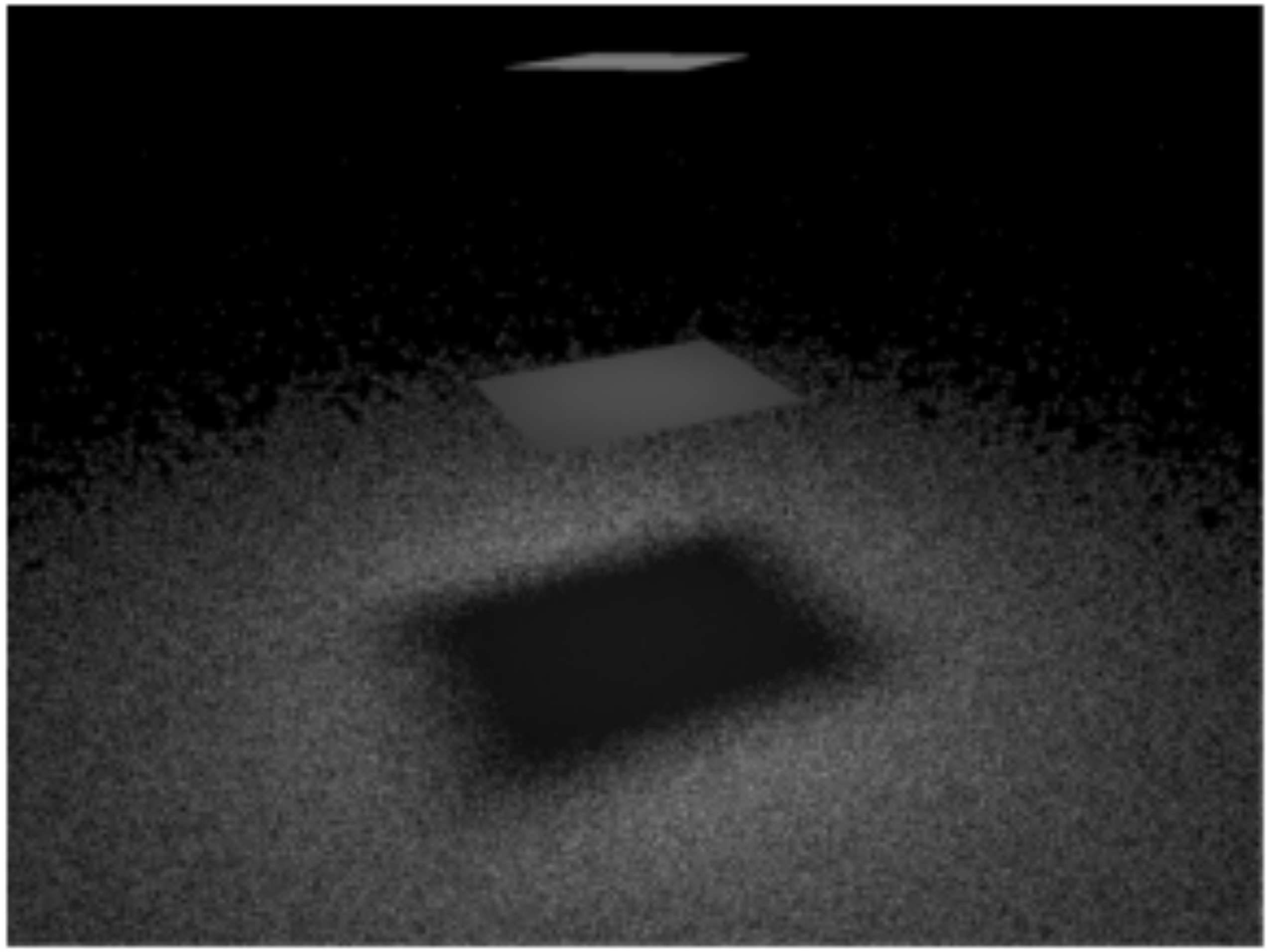
- This has the same shape as f,
but normalized so it integrates to 1
  - Note: need non-negative f for this to work
- But now f/p IS constant and we have no noise at all!
  - Alas: to come up with this p, we need the integral of f, which is what we are trying to compute in the first place :)

What’s a Good PDF?

- One that mimics the shape of f, but is easy to sample from
- Because p is in the denominator, should try to avoid cases where p is low and f is high
  - These samples will increase variance a LOT

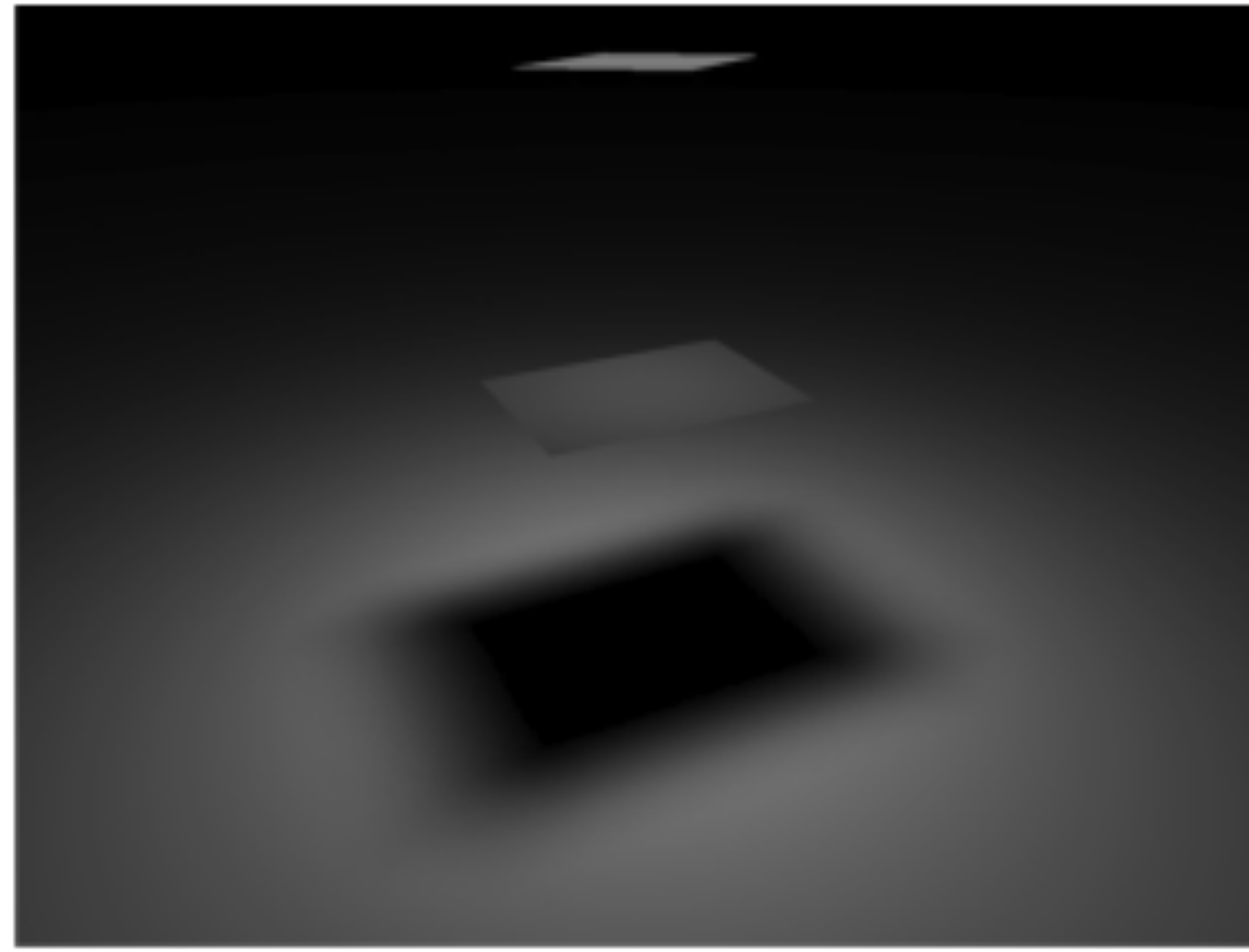
Example of Importance Sampling

This is precisely the difference between sampling directions
vs. sampling light source area for direct illumination (you saw this earlier)



Hemispherical Solid Angle

4 eye rays per pixel
100 rays



Light Source Area

4 eye rays per pixel
100 shadow rays

CS348B Lecture 8

Pat Hanrahan, Spring 2011

Questions?

runes.nu, rendered using Maxwell

Importance Sampling Example

- Remember: computation of irradiance means integrating incident radiance and cosine on hemisphere:

$$E = \int_\Omega L_{\mathrm{in}}(\omega)\ \cos\theta \,\mathrm{d}\omega$$

- We usually can't make assumptions about the lighting, but we *do* know the cosine weighs the samples near the horizon down => makes sense to importance sample with $p(\omega) = \cos\theta/\pi$
  - Why pi? Remember that $\cos\theta$ integrates to pi over hemisphere, so to get a proper PDF must normalize!

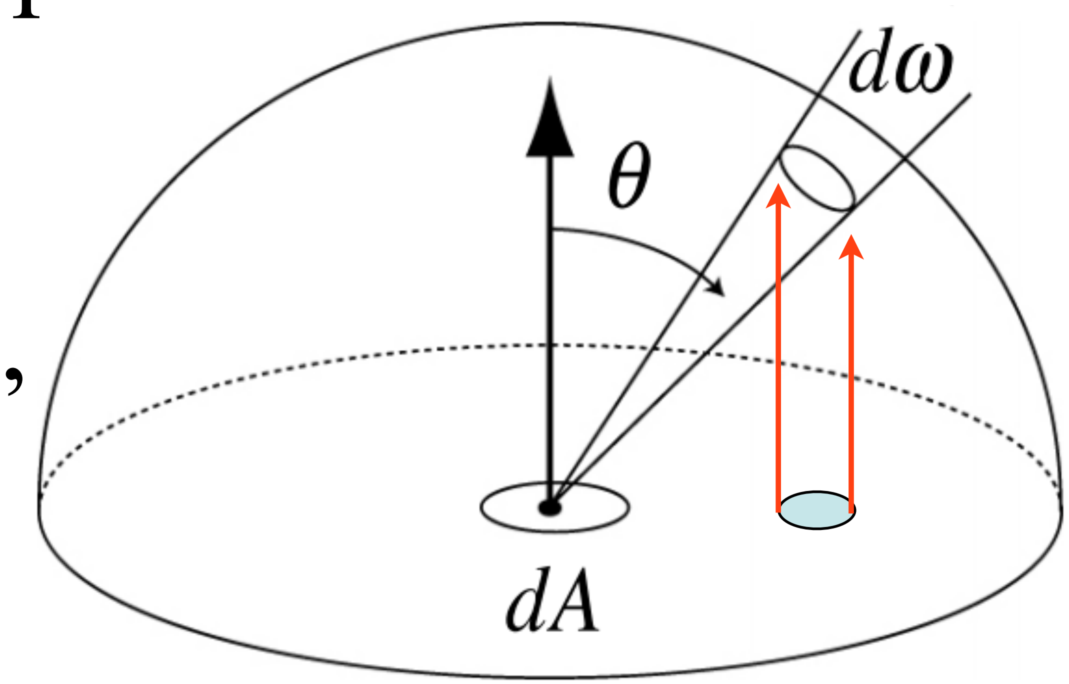
But How? You're Doing This Already

- In your assignment, you're lifting points from the unit disk onto the unit hemisphere, i.e., you're mapping

$$X = x, Y = y, Z(x, y) = \sqrt{1 - x^2 - y^2} \qquad P = (X, Y, Z)$$

- If we have uniform density of points on the disk, i.e., $p(x,y)=1/\pi$, what's the density of points on the hemisphere?

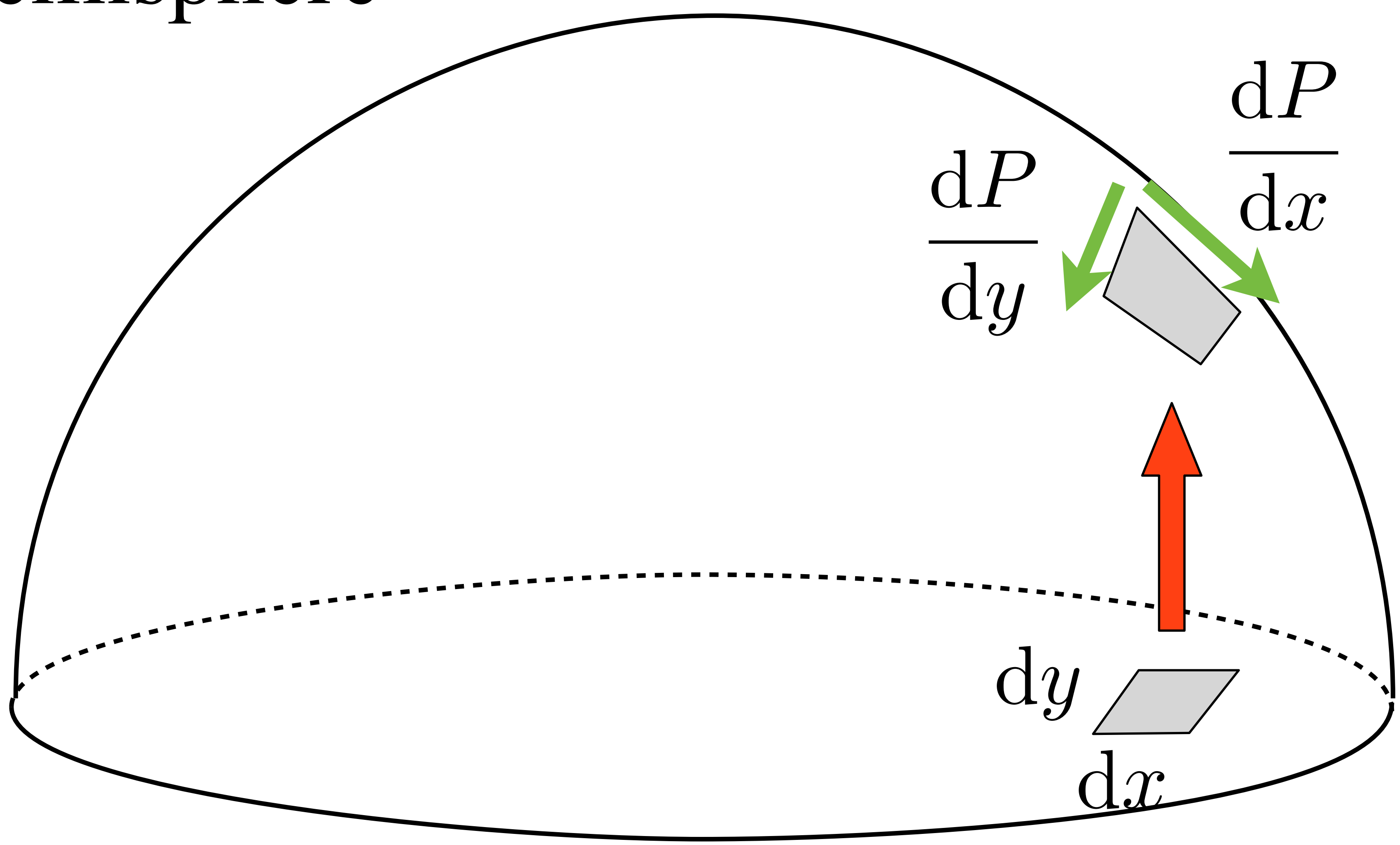
- Instance of "transform sampling"



But How? You're Doing This Already

$$X = x, Y = y, Z(x,y) = \sqrt{1 - x^2 - y^2} \qquad P = (X, Y, Z)$$

- Let's take the infinitesimal square dA = dx*dy and map it to the hemisphere



But How? You're Doing This Already

$$X = x, Y = y, Z(x,y) = \sqrt{1 - x^2 - y^2}$$

- Let's take the infinitesimal square dA = dx*dy and map it to the hemisphere; then, remembering the properties of the cross product, compute its area by

$$\left\| \left(\frac{\partial X}{\partial x}, \frac{\partial Y}{\partial x}, \frac{\partial Z}{\partial x}\right) \times \left(\frac{\partial X}{\partial y}, \frac{\partial Y}{\partial y}, \frac{\partial Z}{\partial y}\right) \right\|$$

$$= \sqrt{\frac{|x|^2}{x^2 + y^2 - 1} + \frac{|y|^2}{x^2 + y^2 - 1} + 1}$$

But...

$$\sqrt{\frac{|x|^2}{x^2+y^2-1}+\frac{|y|^2}{x^2+y^2-1}+1}$$

This equals 1 (why?)

$$=\sqrt{\frac{|x|^2}{|Z|^2}+\frac{|y|^2}{|Z|^2}+\frac{|Z|^2}{|Z|^2}} \qquad = \frac{1}{|Z|}\boxed{\sqrt{|X|^2+|Y|^2+|Z|^2}}$$

$$= 1/Z$$

Ha!

$$\sqrt{\frac{|x|^2}{x^2+y^2-1}+\frac{|y|^2}{x^2+y^2-1}+1}$$

$$=\sqrt{\frac{|x|^2}{|Z|^2}+\frac{|y|^2}{|Z|^2}+\frac{|Z|^2}{|Z|^2}} \quad = \frac{1}{|Z|}\sqrt{|X|^2+|Y|^2+|Z|^2}$$

$$= 1/Z$$

- In polar coordinates, z = $\cos\theta$
- So: a small area on disk gets mapped to one whose area is divided by $\cos\theta$; density is inversely propotional, i.e., $p(\omega) = \cos\theta/\pi$ => samples are cosine-weighted!

Remember: original density on disk is $1/\pi$!

MC Irradiance w/ Cosine Importance

- We'll use the lifting to turn uniform points on the disk onto cosine-distributed points on hemisphere, then

$$E = \int_\Omega L_{\text{in}}(\omega) \cos\theta \, \mathrm{d}\omega \quad \approx \frac{1}{N} \sum_{i=1}^{N} \frac{L_{\text{in}}(\omega_i)}{p(\omega_i)} \cos\theta_i$$

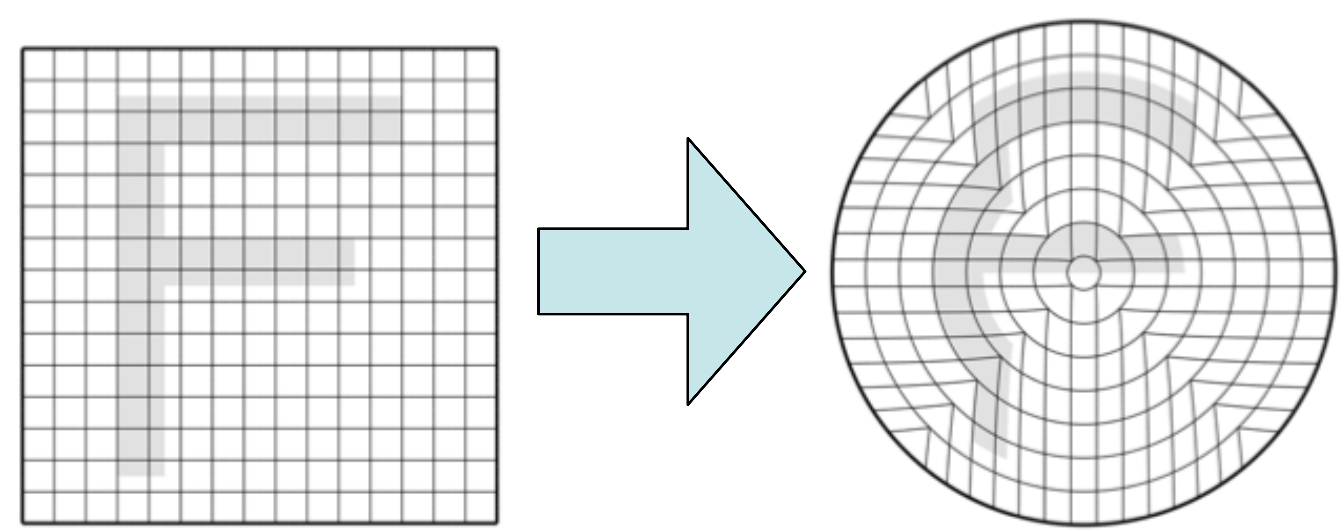
but $p(\omega) = \cos\theta/\pi$, so

$$E \approx \frac{\pi}{N} \sum_{i=1}^{N} L_{\text{in}}(\omega_i)$$

Irradiance is just an average of the incoming radiance when the samples are drawn under the cosine distribution

How to Draw Samples on the Disk?

- You're doing rejection sampling in your assignment
  - I.e., draw uniformly from a larger area (square), reject samples not in the domain (disk)
- Better way is to sample the disk uniformly and continuously map the square to disk
  - Better than rejection sampling, don't need to test and potentially regenerate
  - Also easily allows stratification
  - See Shirley & Chiu 97



Pseudocode

```
Vec3f result;

for i=1:n
   // can implement through rejection or Shirley&Chiu
   Vec2f disk = uniformPointUnitDisk();
   // lift disk point to hemisphere..
   Vec3f Win( disk, sqrt(1.0f - disk.x*disk.x - disk.y*disk.y) );
   // get incoming lighting and add to result
   Vec3f Lin = getRadiance(Win);
   result += Lin;
end

result = result * pi * (1.0f/N);
```

Pseudocode

```
Vec3f result;

for i=1:n
   // can implement through rejection or Shirley&Chiu
   Vec2f disk = uniformPointUnitDisk();
   // lift disk point to hemisphere..
   Vec3f Win( disk, sqrt(1.0f - disk.x*disk.x - disk.y*disk.y) );
   // get incoming lighting and add to result
   Vec3f Lin = getRadiance(Win);
   result += Lin;
end

result = result * pi * (1.0f/N);
```

This is almost a path tracer!
Just missing getRadiance()
and BRDF.

Homework: Phong Lobes

- For a fixed outgoing angle, the specular Phong lobe is

$$f_r(\omega_{\text{in}}) = C(\mathbf{r}(\omega_{\text{out}}) \cdot \omega_{\text{in}})^q$$

- *C* is normalization constant 2π/(q+1) (see Wolfram Alpha), **r** returns the mirror vector, *q* is shininess

- Can you derive a formula for a PDF $p(\omega_{\text{in}})$ that is proportional to the Phong lobe for fixed **r**?
  - Hint: Note that the lobe is radially symmetric around **r** => you can concentrate on a canonical situation, e.g., **r** = (0,0,1)
  - The general case follows by rotation

Abstraction Pays, As Usual

- Because you often need different PDFs, you don't really want to write all the code for picking random points/directions directly in your inner loop
- Instead abstract into two functions
  - 1. one function for generating the points/directions, and
  - *2. another to evaluate the PDF at any given point/direction*

- (Why 2 instead of 1? This comes in handy if you do Multiple Importance Sampling, next slide, where you need to evaluate PDFs also for points drawn from different distributions)

Questions?