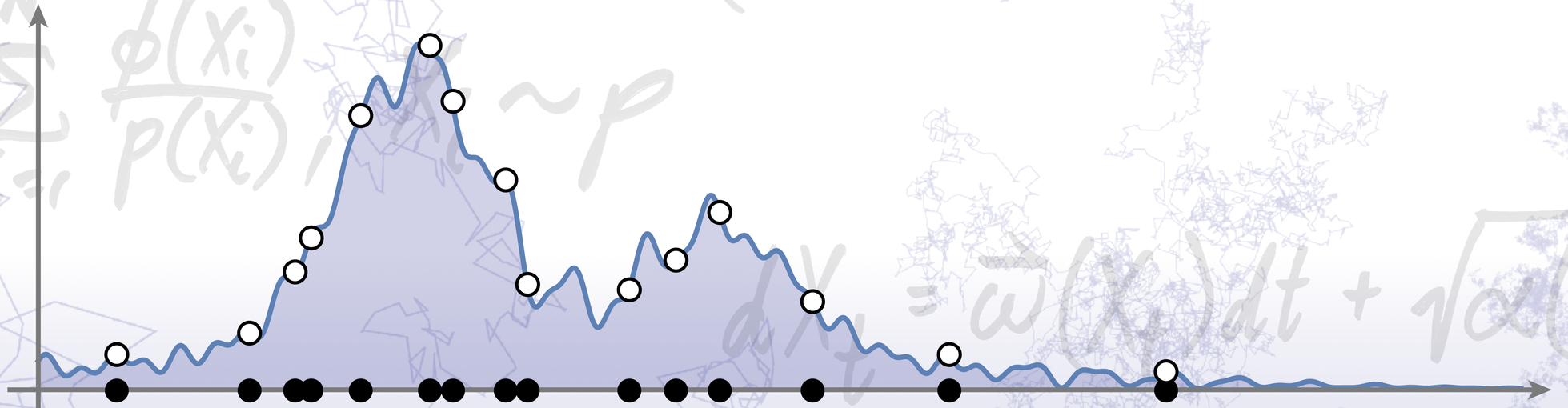
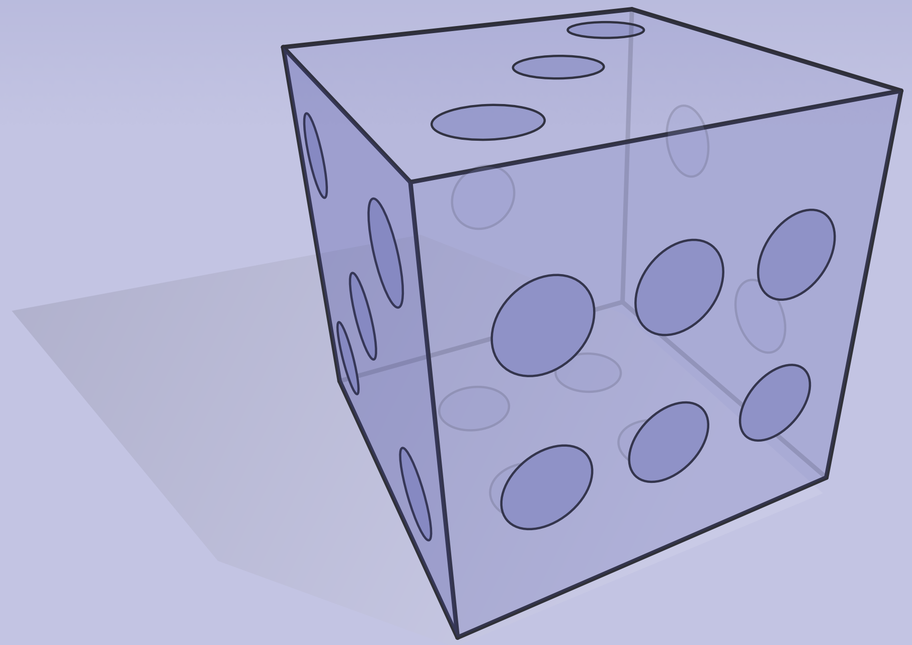
Monte Carlo Methods and Applications



Carnegie Mellon University | 21-387 / 15-327 / 15-627 / 15-860 | Fall 2023

Lecture 5

Variance Reduction

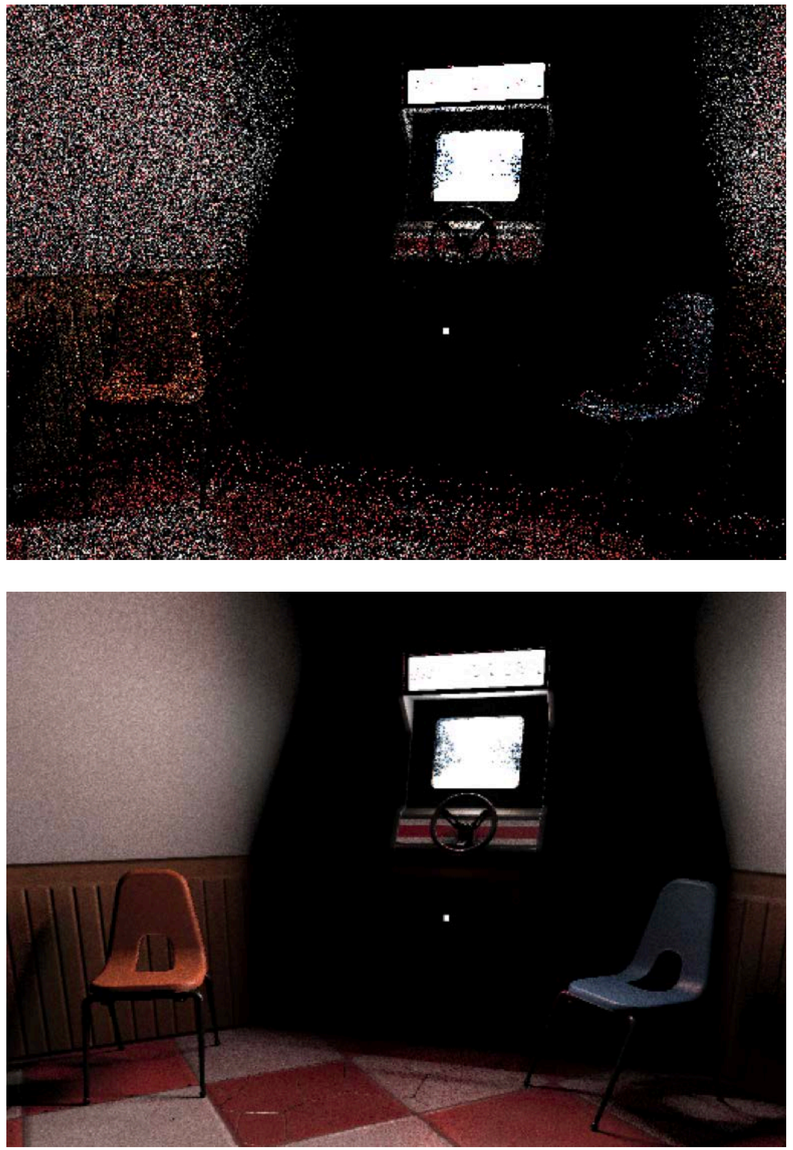


Monte Carlo Methods and Applications

Carnegie Mellon University | 21-387 / 15-327 / 15-627 / 15-860 | Fall 2023

Variance Reduction—Overview

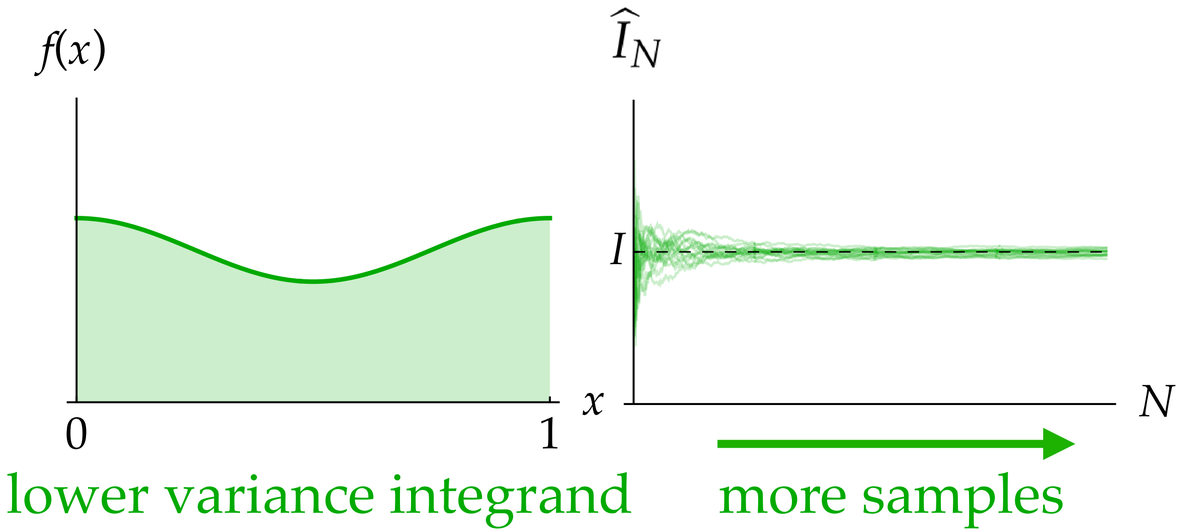
- **So far:** "basic" Monte Carlo estimator
  - draw N samples uniformly at random
- Error is $O(1/\sqrt{N})$ *asymptotically*—but constants can be big ("noisy" estimates!)
- **Today:** how can we do better?
- Basic idea: squeeze more out of each sample via *variance reduction*
- Constants matter! (Especially if you only have time to take a few samples…)
  - In practice: 1,000+ x reduction in error

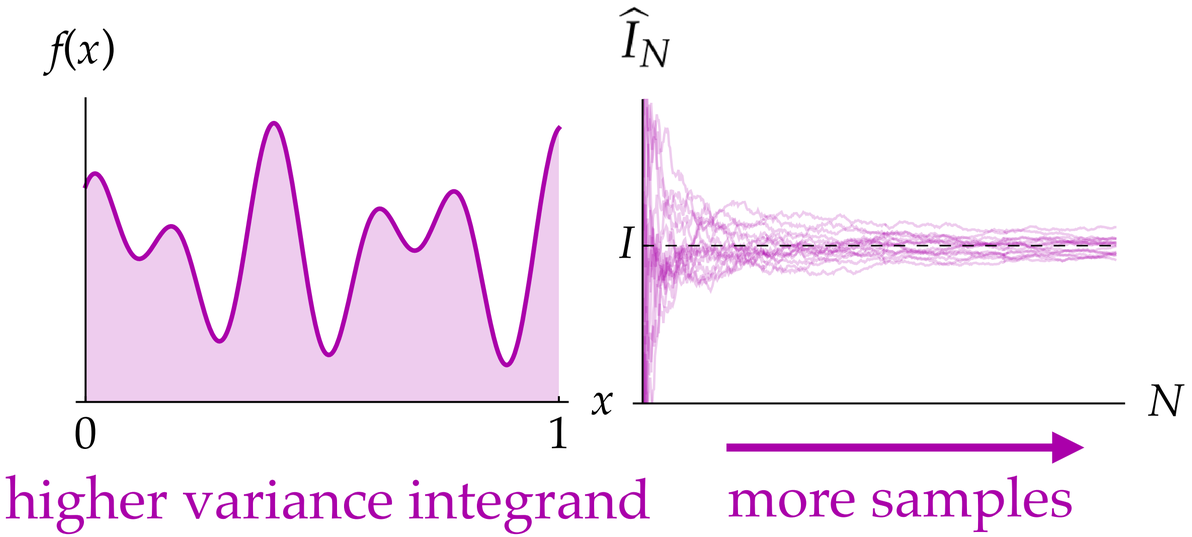


Recap: Efficiency of Monte Carlo

- Suppose we have a fixed time budget (e.g., 1s). How can we reduce error?
- Recall our error analysis of basic Monte Carlo method
- **Main conclusion:** error σ depends on:
  - variance of integrand $V[f]$
  - number of samples $(1/\sqrt{N})$

$$\sigma(\widehat{I}_N) = \sqrt{V[f]/N}$$





Make It Go Faster

- To do better, have to make at least one of two factors smaller:
  - $(1/N)$ **more samples per second** $\Longrightarrow$ computer science at large (software engineering, parallel computing, computer architecture…)
  - $(V[f])$ **less error for equal time** $\Longrightarrow$ *variance reduction*

$$\boxed{\sigma(\widehat{I}_N) = \sqrt{V[f]/N}}$$

V = weight of dog

N = speed of turtle

Some wisdom on making it go faster…

"Computation costs so much less than human effort that we ordinarily require large efficiency gains to offset the time spent programming up a variance reduction."

—Art Owen

"Premature optimization is the root of all evil."

—Donald Knuth

Variance of the Integrand

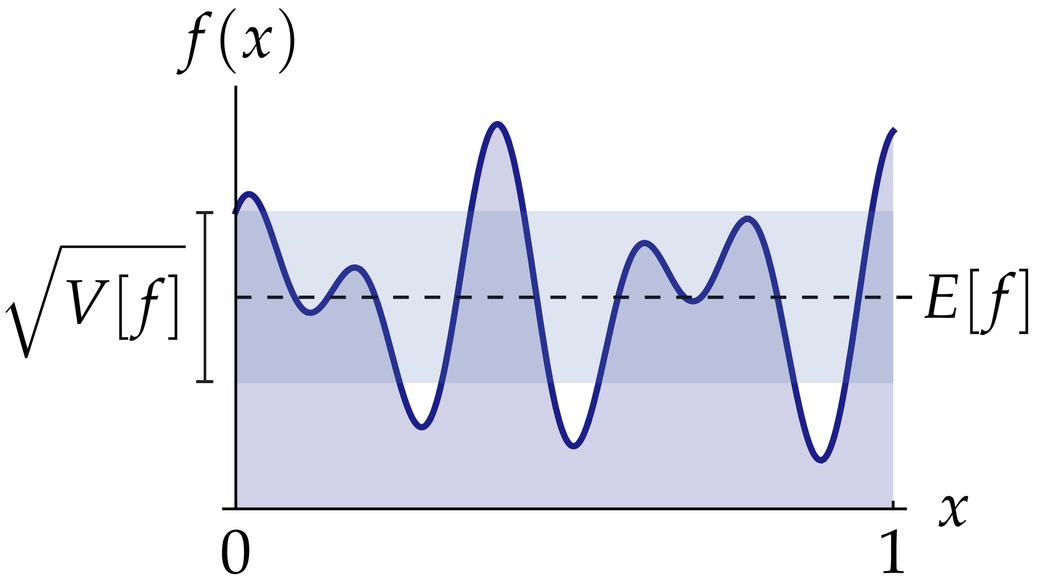
- By the "variance of the integrand" $f : \Omega \to \mathbb{R}$ we mean variance of random variable $Y := f(X)$, where X is uniform random variable on Ω
- **Q:** How can we make the variance smaller?
- **A:** We can't.
  - for a given integrand f, $V[f]$ is fixed!
- Instead, find an **equivalent integration problem with a lower-variance integrand**
  - describes basically *all* variance reduction methods (if viewed the right way…)

expected value $E[f]$

$$\frac{1}{|\Omega|} \int_\Omega f(x)\, dx$$

variance $V[f]$

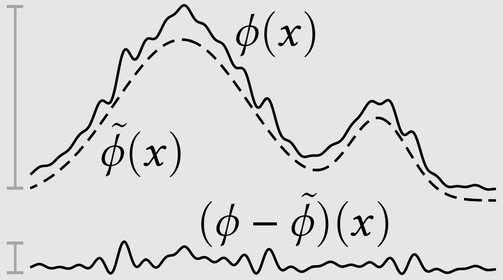
$$\frac{1}{|\Omega|} \int_\Omega (E[f] - f(x))^2\, dx$$



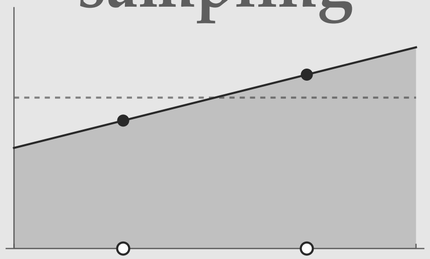
Taxonomy of Acceleration Strategies

There's a whole zoo of Monte Carlo acceleration strategies:

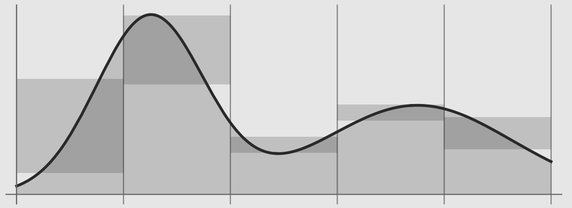
control variates



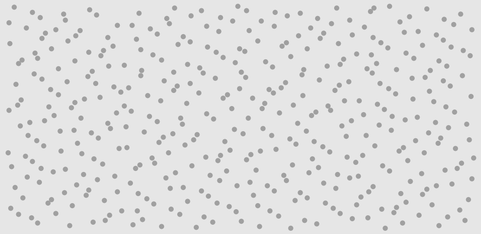
antithetic sampling



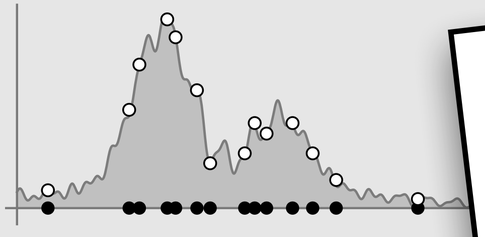
stratified sampling



quasi Monte Carlo



importance sampling



All can be viewed as replacing integrand with *lower-variance* integrand.

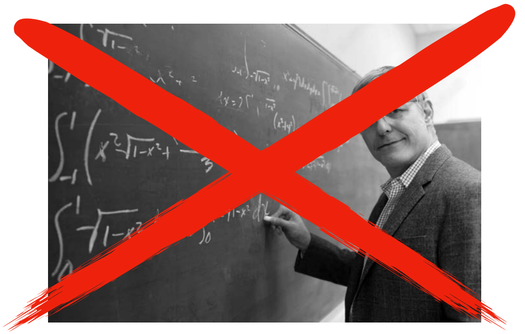
Also: application-specific strategies

A Warning About Variance Reduction

- **WARNING:** Not all variance "reduction" strategies are guaranteed to reduce variance
  - sometimes it can become (much) *worse!*
  - sometimes it becomes only a little better—but more costly to evaluate
- Have to think about when a technique is appropriate
- Will try to highlight which techniques do/don't have universal guarantees

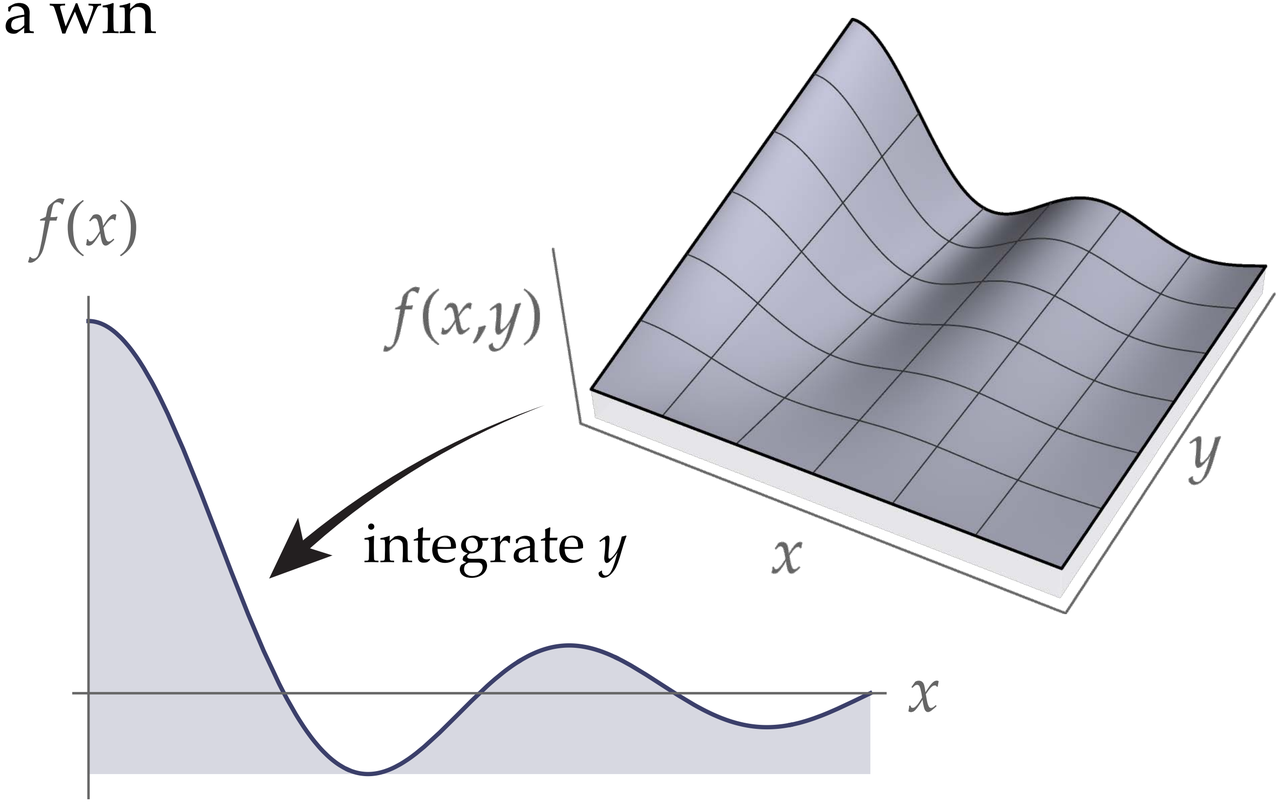
Integrate What You Can Integrate

- Before we get to variance reduction, also worth saying: *don't throw closed-form integrals completely out the window…*
- Poked fun at them earlier—but if you can easily reduce the dimension of your integral, it's a win
- E.g., consider $f(x, y) := y(\sin(x)/x)$
  - can't integrate $\sin(x)/x$ in closed form
  - can <u>easily</u> integrate y with respect to y
  - 1D estimate more accurate for same N
- Sometimes called *"use of expected values", "conditioning", "conditioned MC"*



$$\int_0^1 \frac{\sin(x)}{x}\, dx$$

Nope.



Sample Variance

Suppose we're debugging an implementation of Monte Carlo.
How do we check if we're converging at the expected rate?

First compute **sample mean** $\widehat{\mu}$ (i.e., usual Monte Carlo estimate)

Then compute **sample variance** of M independent estimates Y_i :

$$\frac{1}{M}\sum_{i=1}^{M}(Y_i-\widehat{\mu})^2$$

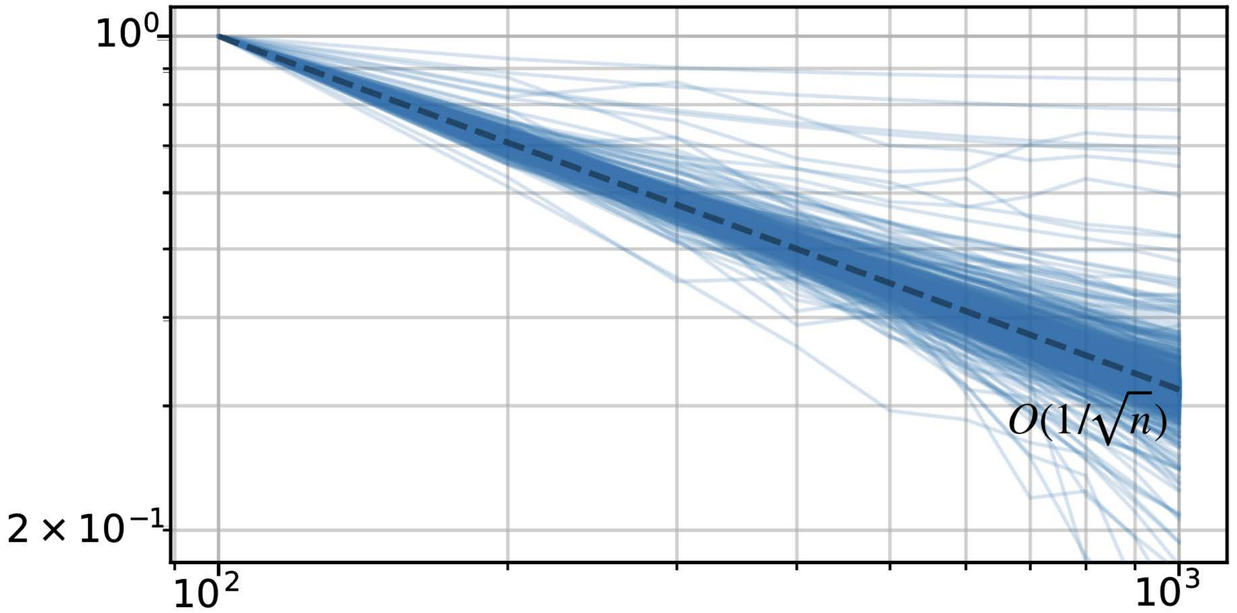
…or is it…

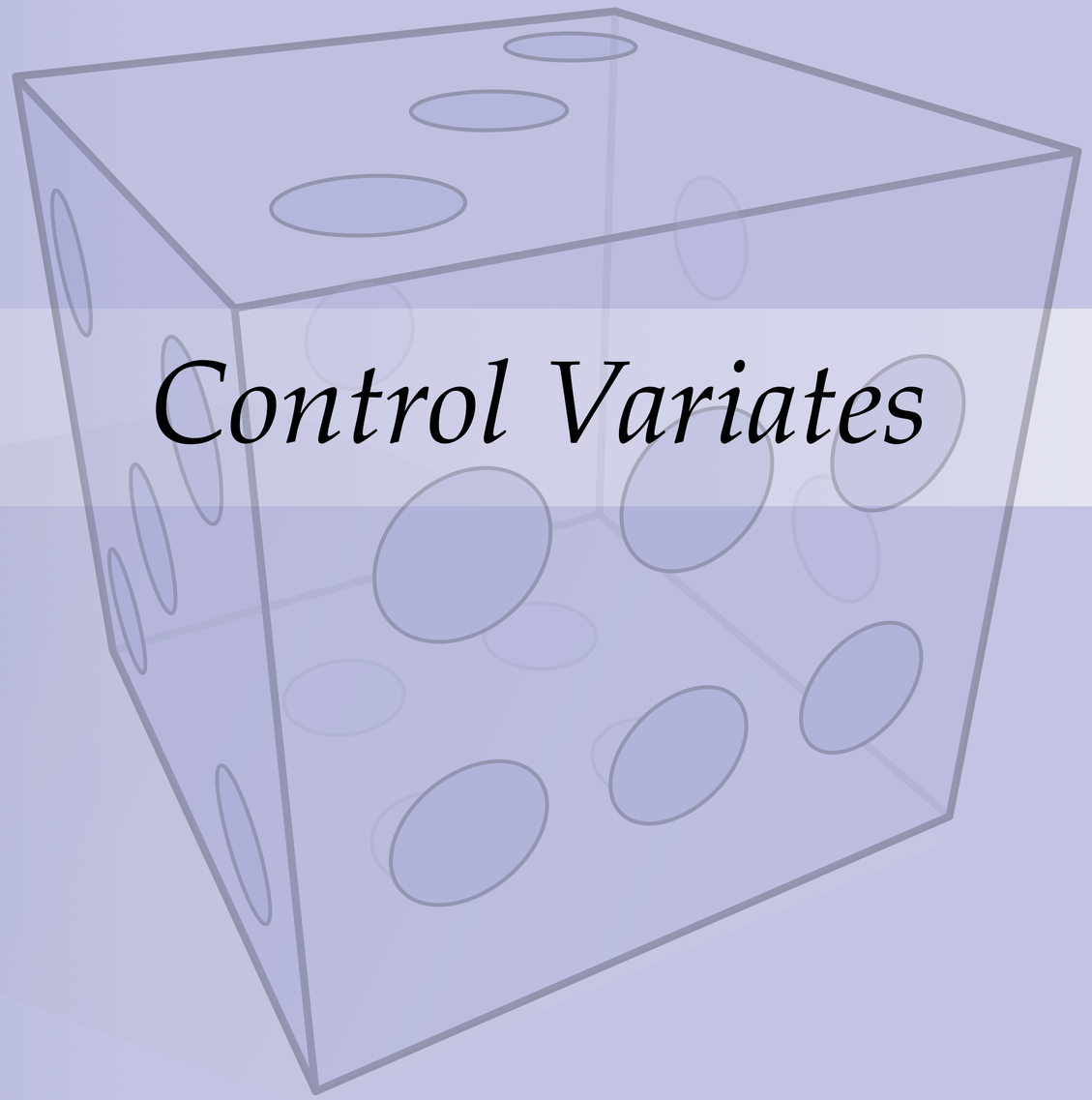
$$\frac{1}{M-1}\sum_{i=1}^{M}(Y_i-\widehat{\mu})^2$$

Nice little exercise: show that the expression on the left is a **biased** estimate of the true variance, and the expression on the right is **unbiased.**

Plotting Error

- When debugging, also extremely helpful to have clear visualization
- Almost always good idea to use a **log-log** plot
  - I broke this rule on many of the earlier slides! 😅
- Compare log-log plot of variance to line with expected slope of –1 (corresponding to $O(1/N)$ convergence)

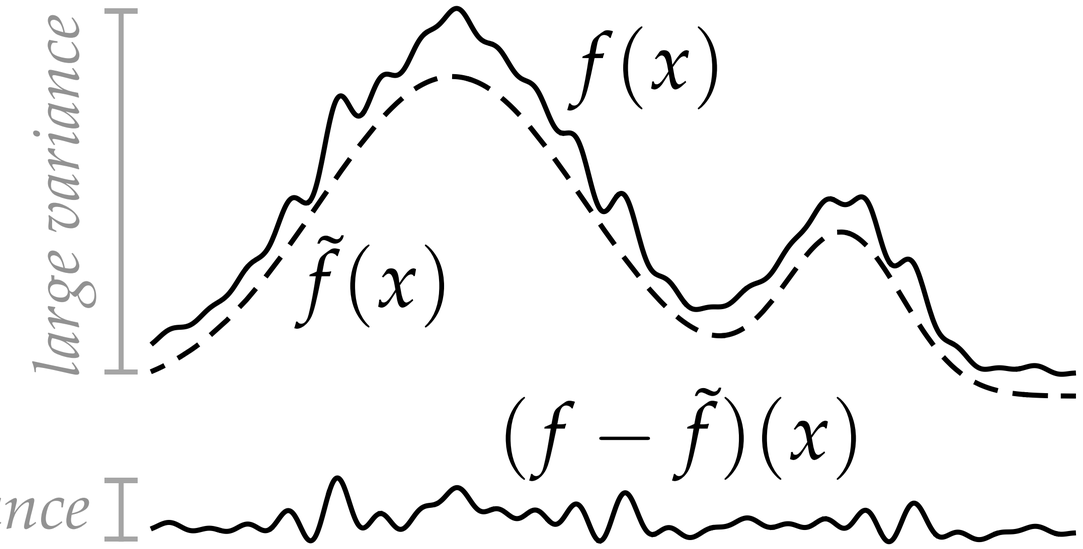




Control Variates

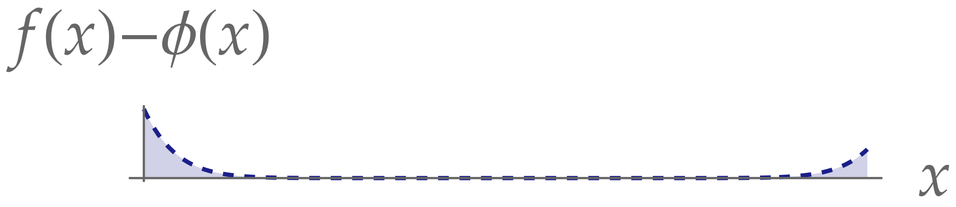
Control Variates

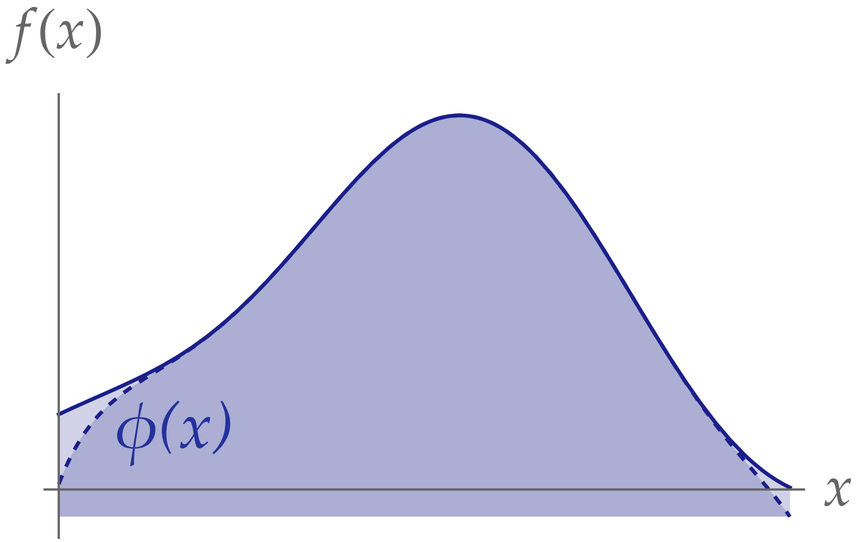
- How do we reduce variance of integrand?
- Conceptually easy place to start: **control variates**
- Basic idea: decompose integrand as $f(x) = f_0(x) + \tilde{f}(x)$ where
  - $f_0(x)$ can easily be integrated (e.g., in closed form)
  - $\tilde{f}(x)$ has smaller variance than $f(x)$
- Then estimate $\tilde{f}(x)$ using Monte Carlo

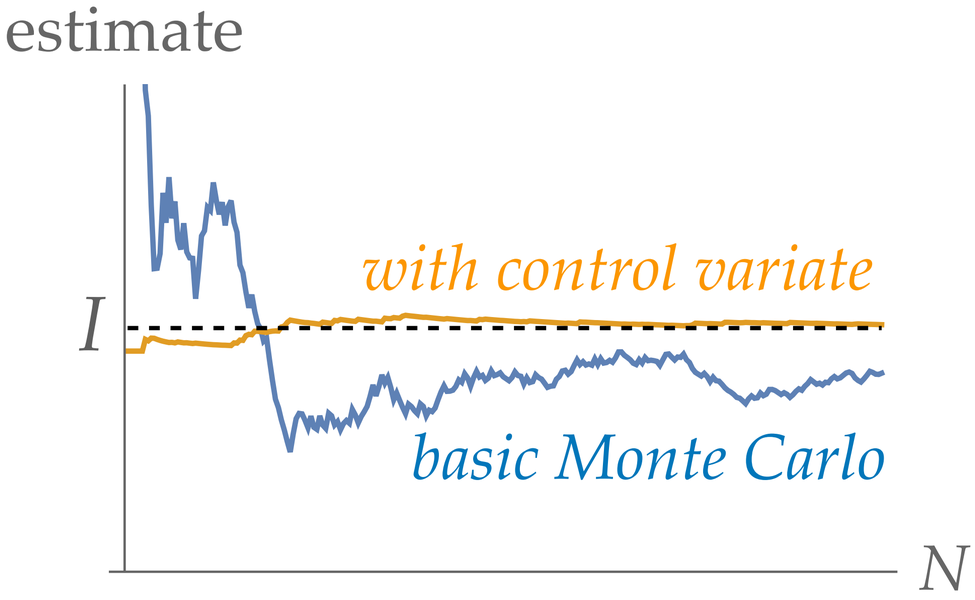


Control Variates Example: Polynomial Fit

- Consider a function $f(x)$ where we only have "black box" access (i.e., can evaluate at any point, but don't know its explicit form)
- How can we subtract off an *easily integrable* part?
- **Idea:**
  - sample at $(k+1)$ points
  - fit a degree-k polynomial $\phi(x)$ (easy to integrate)
  - use Monte Carlo to estimate $\tilde{f}(x) := f(x) - \phi(x)$
- No guarantee this strategy <u>always</u> works well…

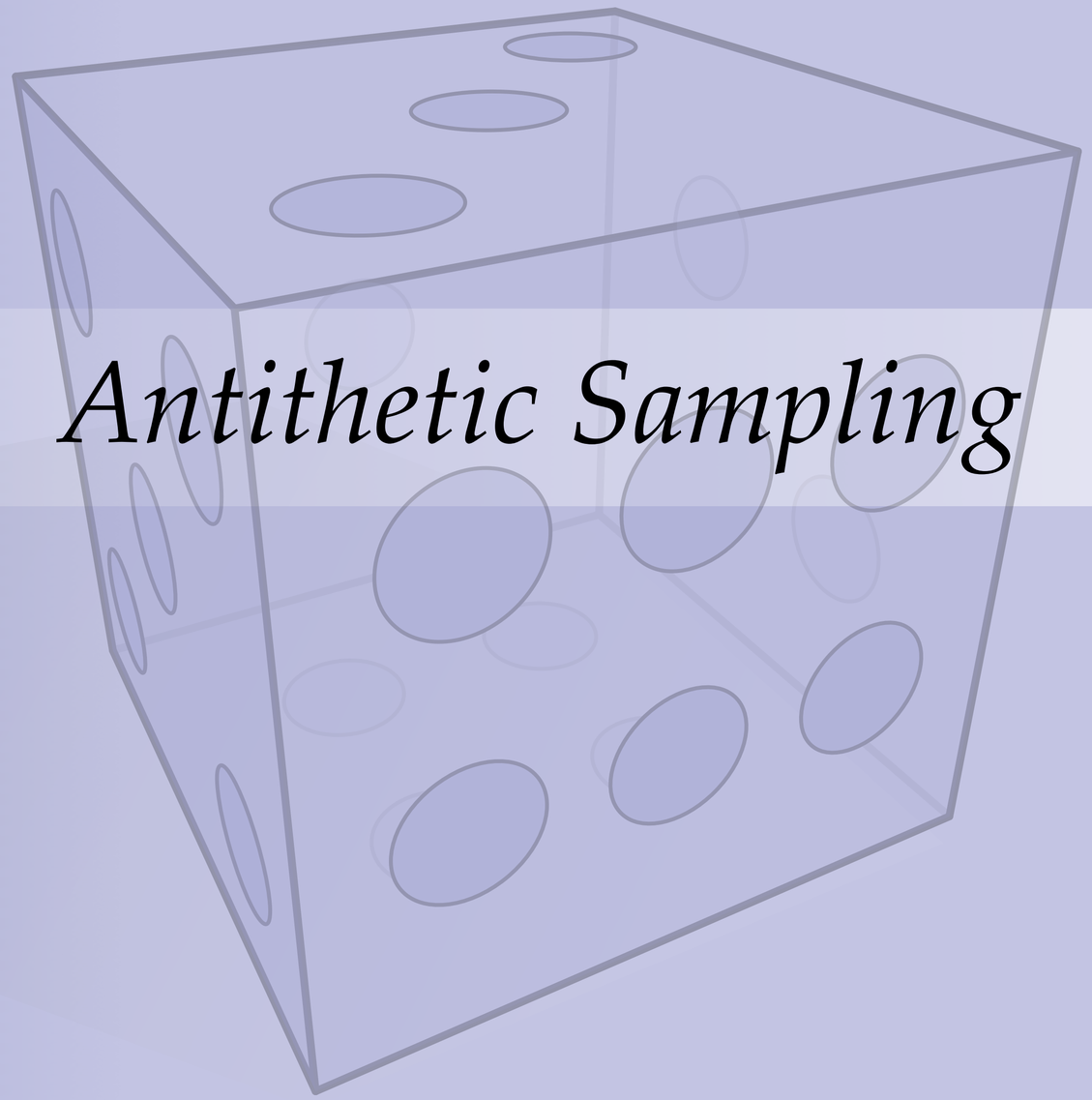






Multi-level Monte Carlo

- Don't forget that evaluating integrand $f(x)$ can be *expensive!*
- Suppose we have progressively more accurate—but more expensive—way to approximate the integrand $f(x)$
  - sequence of functions $f_0(x), \ldots, f_L(x)$ approximating $f(x)$ as $L \to \infty$
  - corresponding integrals are $I_k := |\Omega| E[f_k]$
- Then have telescoping sum $I_L = I_0 + \sum_{k=1}^{L} I_k - I_{k-1}$
  - each term in the sum is effectively a control variate
  - terms have lower & lower variance; can use fewer & fewer samples
- **Notoriously tricky to tune parameters for MLMC!**



Antithetic Sampling

Antithetic Sampling

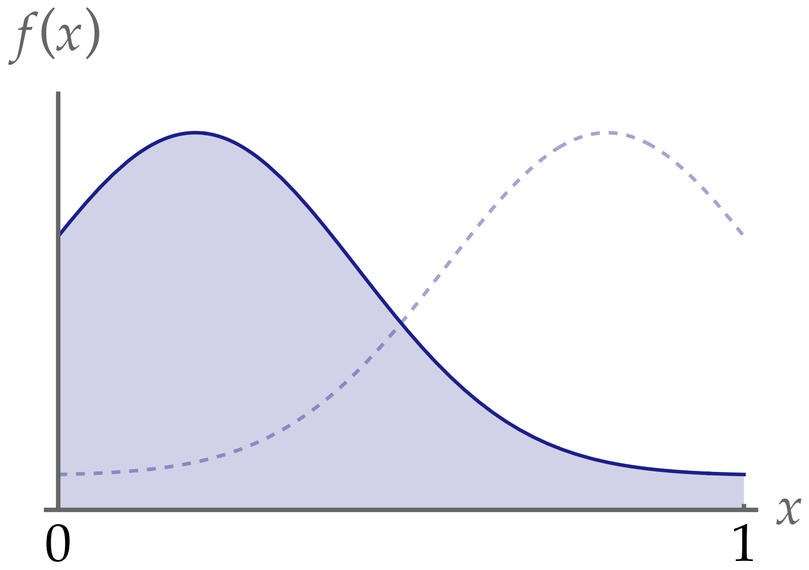
- So far assumed we always want to take <u>independent</u> random samples
- Basic justification: for independent samples,
  $V[(X_1 + X_2)/2] = (V[X_1] + V[X_2])/2 = V[X]/2.$
  – hence, more independent samples guarantees lower variance.
- But for correlated samples, we have
  $V[X + Y] = V[X] + V[Y] + 2\mathrm{Cov}[X, Y]$
- So, suppose we get "greedy": if we allow samples to be correlated, might we get *even lower* variance (via negative covariance)?

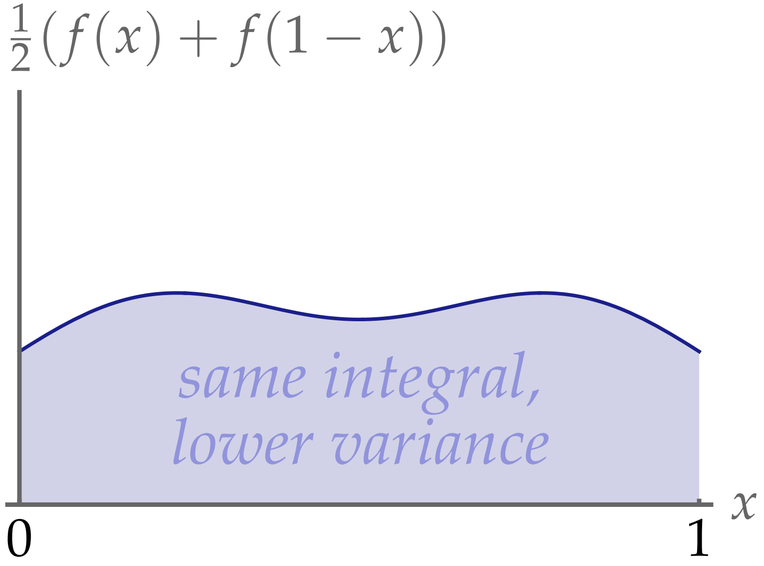
Covariance & Correlation

- Given random variables X, Y (not necessarily independent), the **covariance** quantifies the amount of correlation
- Specifically, $\text{Cov}[X, Y] := E[(X - E[X])(Y - E[Y])]$
  - If X, Y both have zero mean, this is just $E[XY]$
  - Large positive/negative covariance mean X, Y are correlated/anti-correlated
- The **correlation** is a normalized, unitless version of covariance
  - $\text{Corr}[X, Y] := \text{Cov}[X, Y]/(\sigma_X \sigma_Y)$

Antithetic Sampling

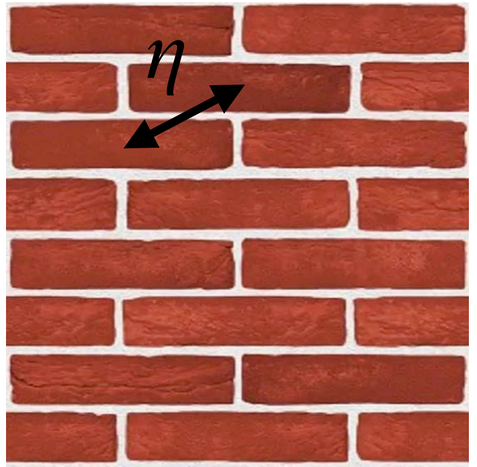
- To reduce variance, must therefore take samples that are *anti-correlated*
- **Rough strategy:** take samples at "equal and opposite points"
- **Intuition:** sample values above mean will cancel those below mean
- E.g., consider function $f : [0,1] \to \mathbb{R}$
  - sample in pairs $f(X_i), f(1 - X_i)$
  - estimator is now $\frac{1}{2N} \sum_{k=1}^{N} f(X_k) + f(1 - X_k)$
  - same as integrating $f'(x) := (f(x) + f(1 - x))/2$, which has same integral as $f(x)$
- No guarantee variance is *always* lower!

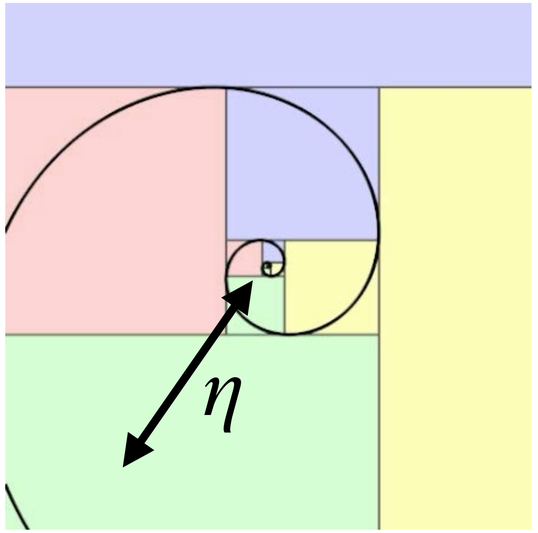


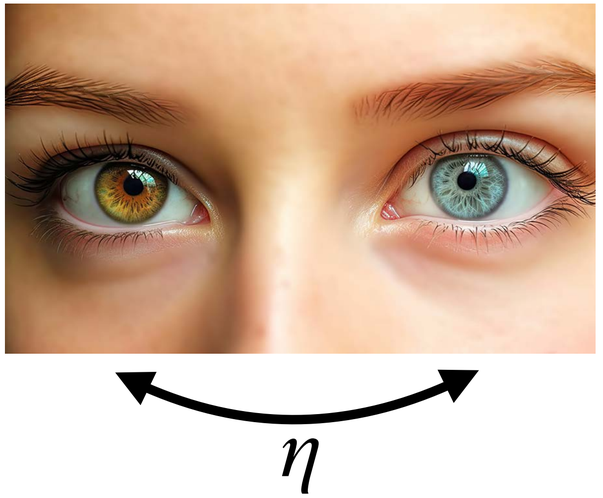


Antithetic Estimator—General Form

- More generally, suppose we have some *symmetry* of the domain Ω, i.e., a map $\eta : \Omega \to \Omega$. Let $\tilde{x} := \eta(x)$ denote the "opposite point."
- For any given Monte Carlo estimator $\widehat{I}_N$ for a function $f : \Omega \to \mathbb{R}$, the corresponding **antithetic estimator** applies the same estimator $\widehat{I}_{N/2}$ to the function $g := (f + f \circ \eta)/2$.

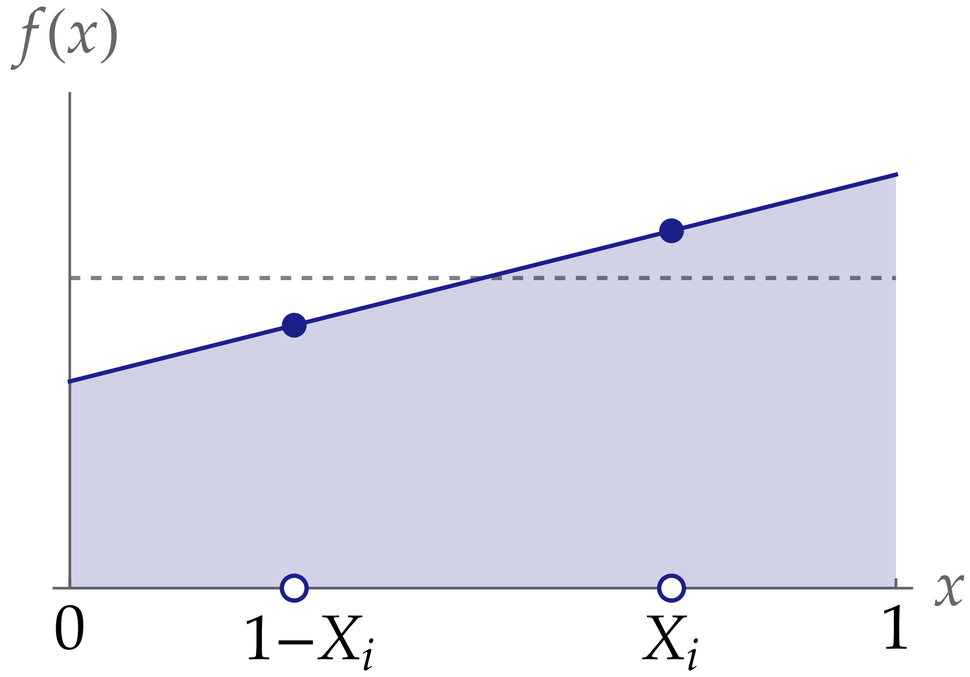






Example: Affine Functions

- Affine functions are an idealized case for antithetic sampling
- E.g., $f(x) := ax + b$ for $0 \leq x \leq 1$, $\eta(x) = 1 - x$
- Antithetic strategy: sample both X_i and $1 - X_i$
- Always gives <u>exact</u> integral, for any $X_i \in [0,1]$:



$$\frac{f(X_i) + f(1 - X_i)}{2} = \frac{(aX_i + b) + a(1 - X_i) + b}{2} = \frac{a}{2} + b$$

$$\int_0^1 ax + b \, dx = \left[\tfrac{1}{2}ax^2 + bx\right]_0^1 = \frac{a}{2} + b$$

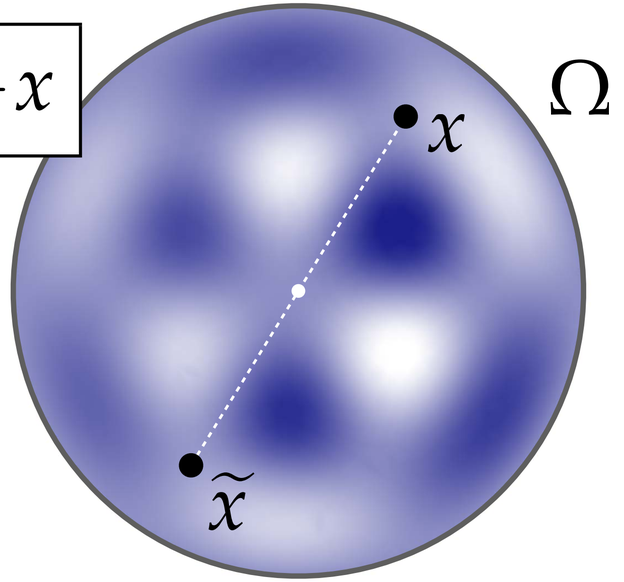
Intuition: over- and under-estimates cancel

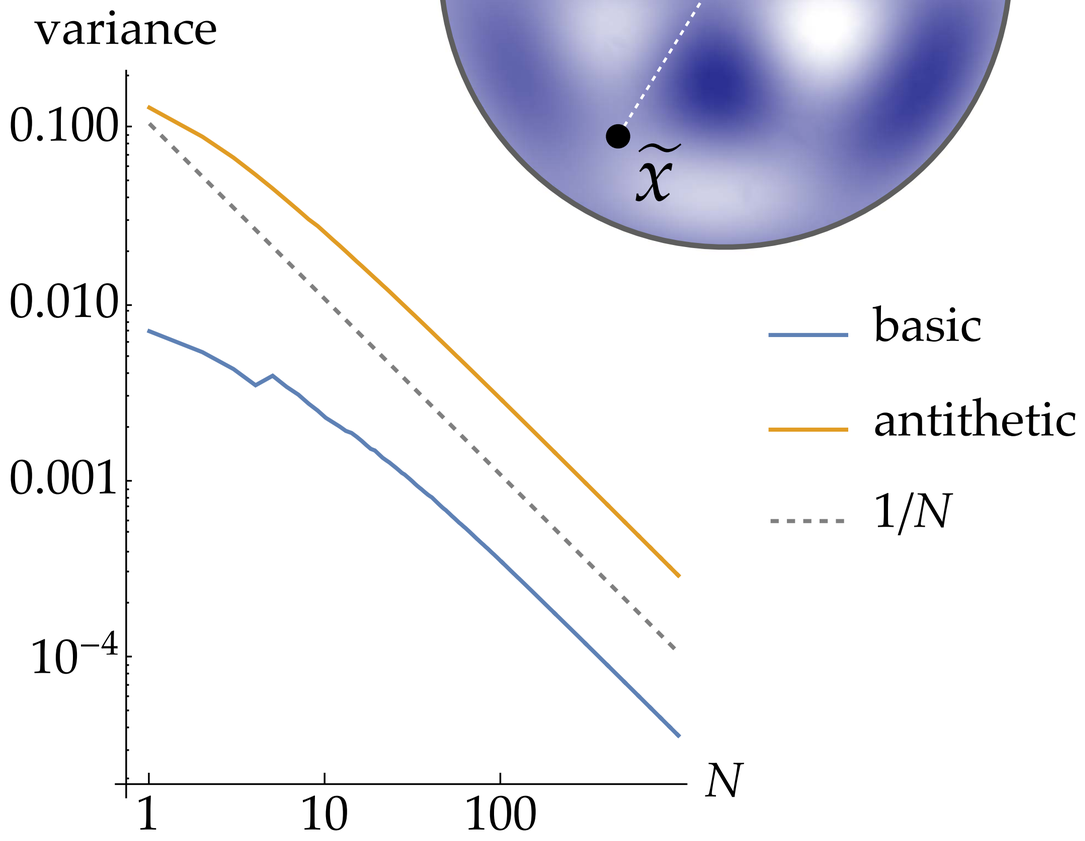
Antithetic Sampling—Example

- Let's try this out on some more generic function $f(x)$ given by sum of harmonics on the unit disk $\Omega := \{x \in \mathbb{R}^2 : |x| \leq 1\}$, using the *antipodal map* $\eta(x) = -x$.

- Get same <u>rate</u> of convergence (variance goes like $1/N$), but the <u>constant</u> is much better in this case (about **10x better!**)

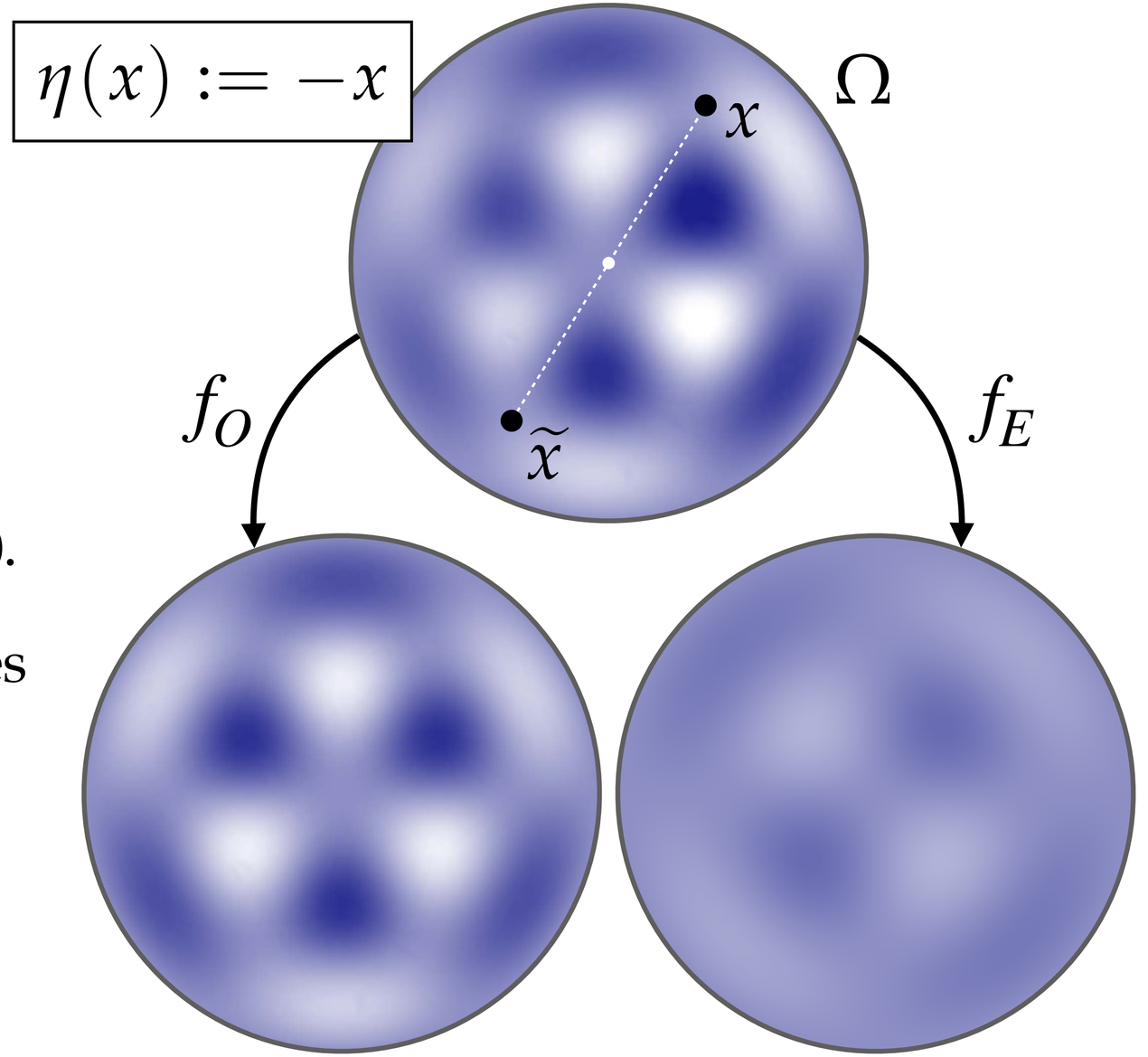
$\eta(x) := -x$





Even/Odd Decomposition

- To understand error behavior, can decompose <u>any</u> integrand $f(x)$ into
  - even part $f_E(x) := (f(x) + f(\tilde{x}))/2$
  - odd part $f_O(x) := (f(x) - f(\tilde{x}))/2$
- Easy to show: <u>even</u> part *increases* variance (up to 2x); <u>odd</u> part *decreases* it (down to 0).
- **Game:** try to find a symmetry η that makes $f(x)$ as "odd as possible."



Even/Odd Decomposition — Analysis

Variance of antithetic estimator $\hat{I}^a_N := \frac{1}{N}\sum_{i=1}^{N/2} f(X_i) + f(\widetilde{X}_i)$ given by

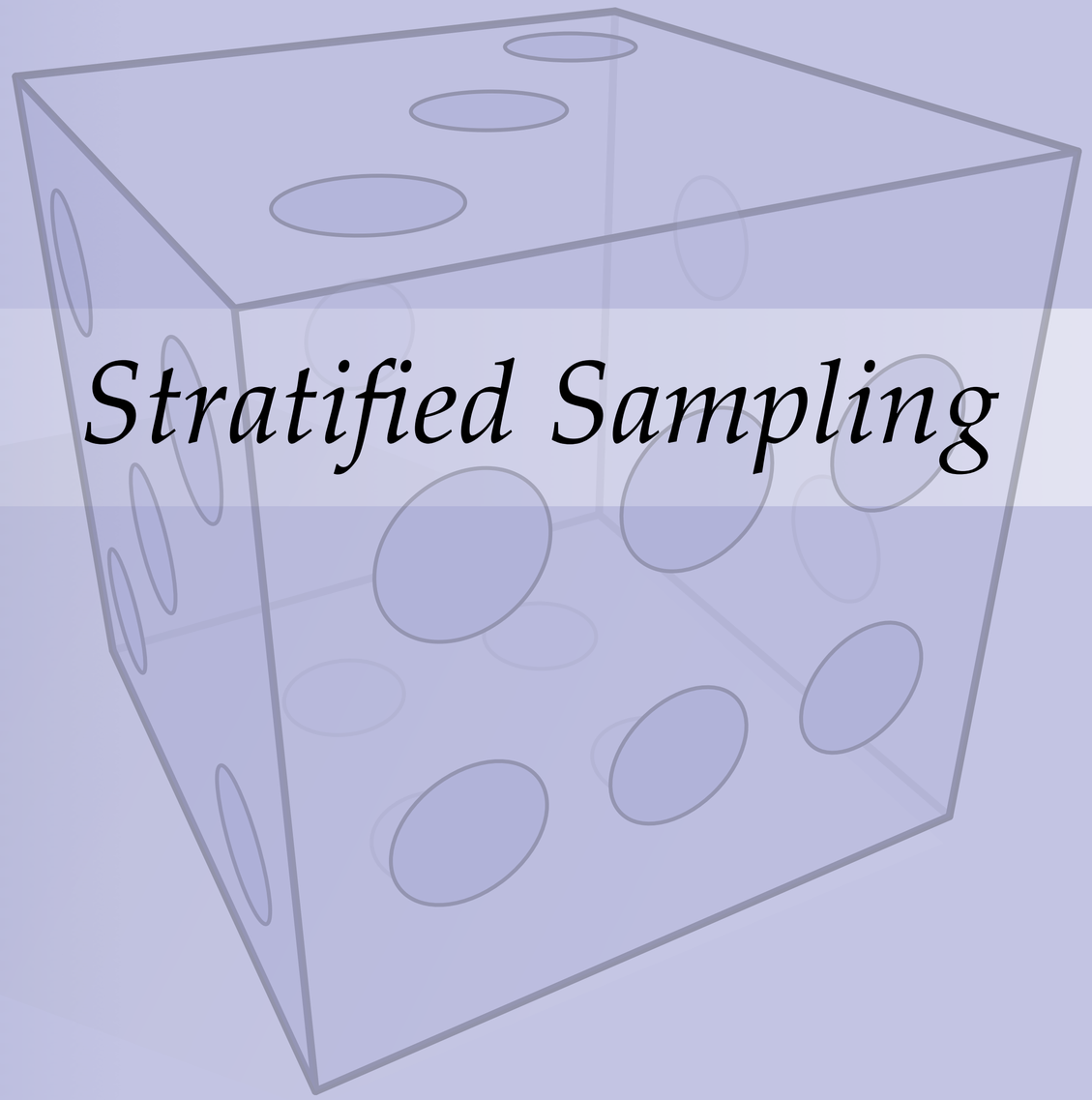
$$V[\hat{I}^a_N] = \frac{N/2}{N^2} V[f(X_i) + f(\widetilde{X}_i)] = \quad \textit{(summands are i.i.d.)}$$

$$\frac{1}{2N}\left(V[f(x)] + V[f(\widetilde{x})] + 2\mathrm{Cov}[f(x), f(\widetilde{x})]\right) = \quad \textit{(definition of variance, linearity of expectation)}$$

$$\frac{V[f(x)]}{N}(1 + \mathrm{Corr}[f(x), f(\widetilde{x})]) \quad \textit{(}V[f(x)] = V[f(\tilde{x})]\textit{, definition of correlation)}$$

$$\in [-1, 1]$$

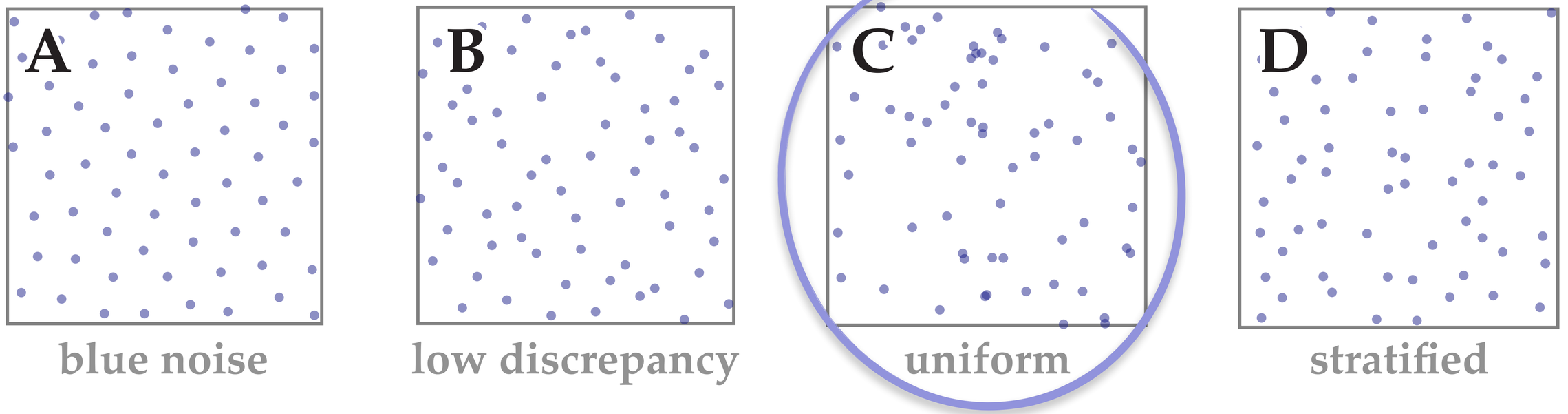
When $f(x)$ is <u>even</u>, **double** the variance; when <u>odd</u>, get **zero** variance.



Stratified Sampling

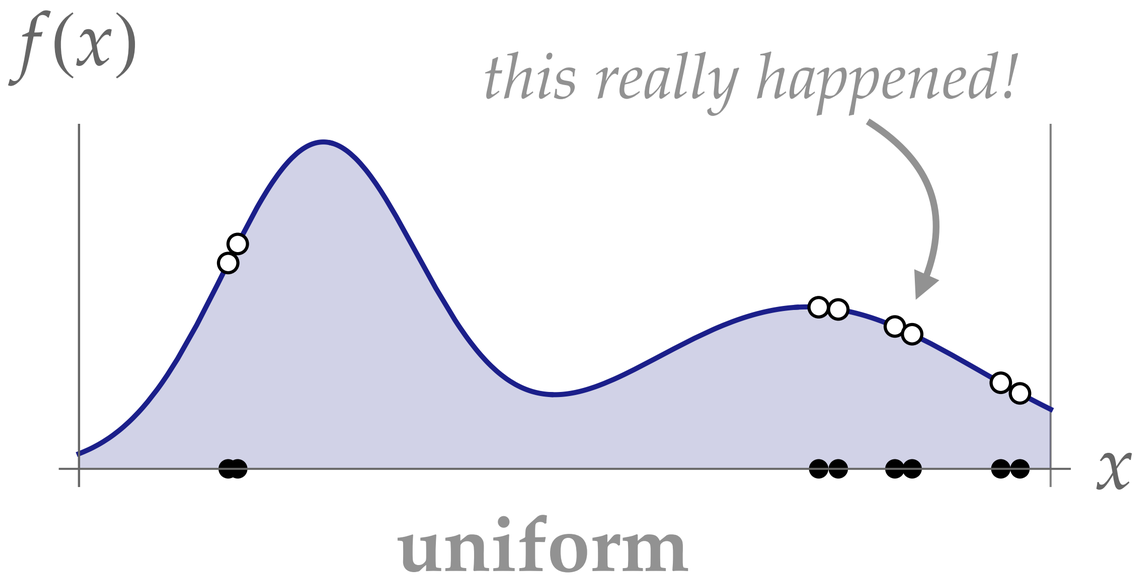
Which Points Are "Random"?

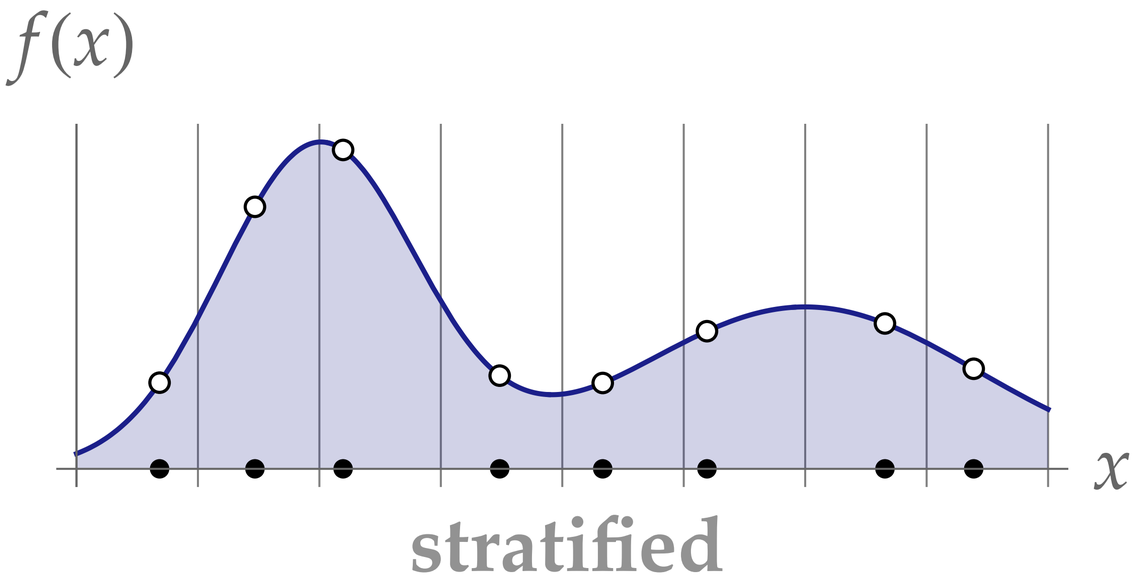
- Basic Monte Carlo estimator uses points sampled uniformly at random
- **Q:** Which of the point sets below do you think are **uniformly** random?
  - *i.e.*, probability of picking sample from a region is proportional to area
- **A:** Uniformly sampled points can actually "clump up" quite a bit!



Stratified Sampling

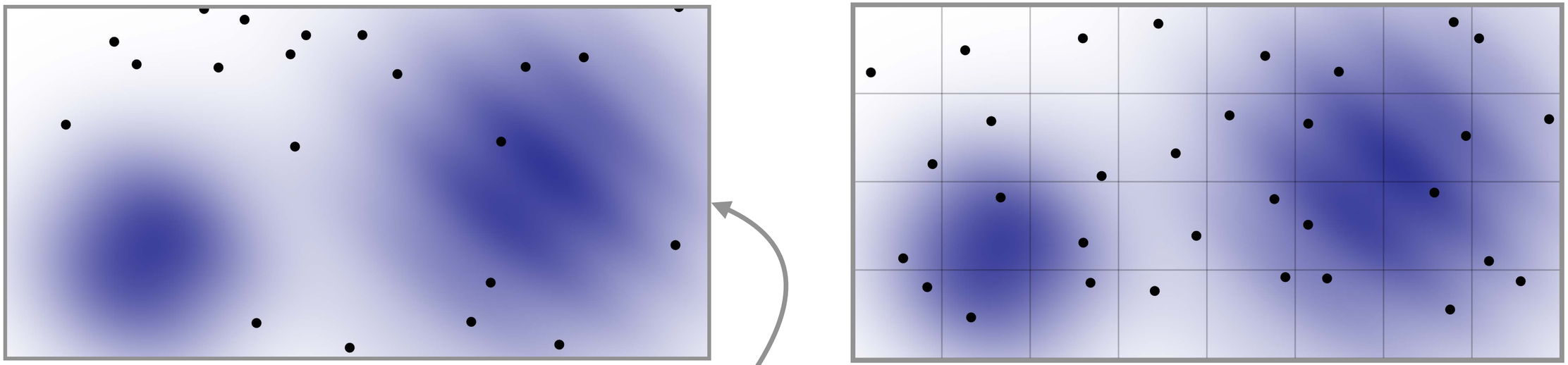
- **Stratified sampling** spreads points out more evenly over domain
  - partition domain into pieces (“strata”); draw a fixed number of samples uniformly at random from each piece
  - less likely to miss important features of integrand, *and* smaller variance (typically) within each stratum





Stratified Sampling

- **Stratified sampling** spreads points out more evenly over domain
  - partition domain into pieces (“strata”); draw a fixed number of samples uniformly at random from each piece
  - less likely to miss important features of integrand, *and* smaller variance (typically) within each stratum

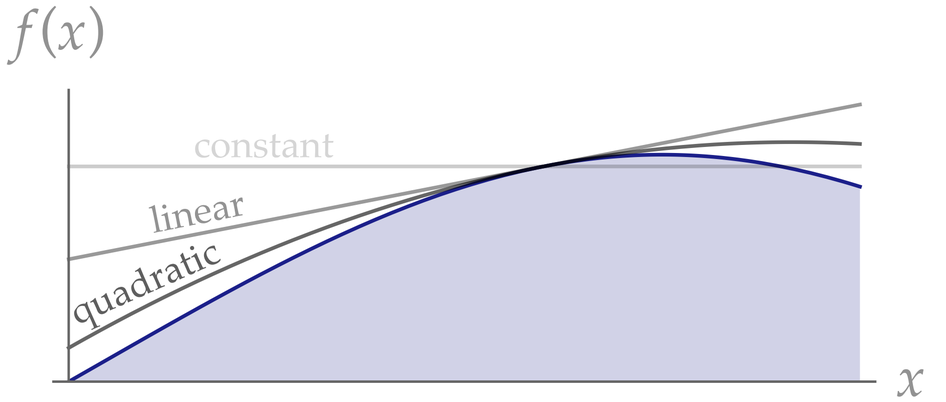


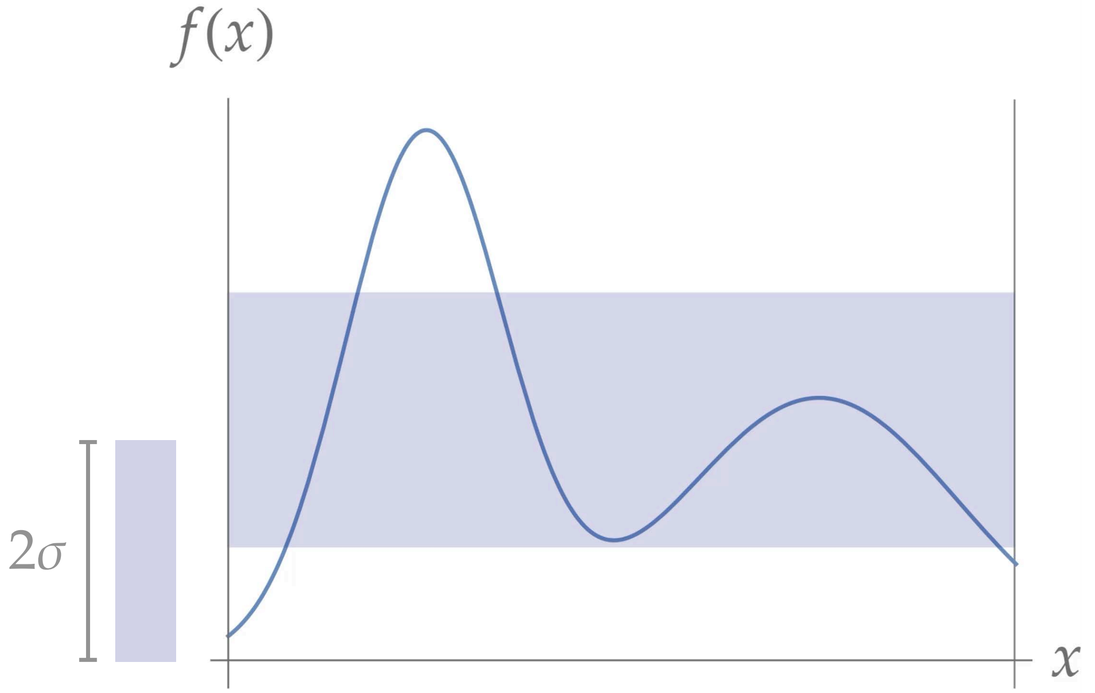
Stratified Sampling—Intuition

- Why does stratified sampling tend to reduce variance?
- **(Taylor series)** "Nice" functions (smooth, analytic, …), look nearly linear "up close"
- The smaller strata become, the closer we come to integrating *linear* functions over very *short* intervals

⇒ Nearly *constant* in each stratum

⇒ variance goes to <u>zero</u> in each stratum!





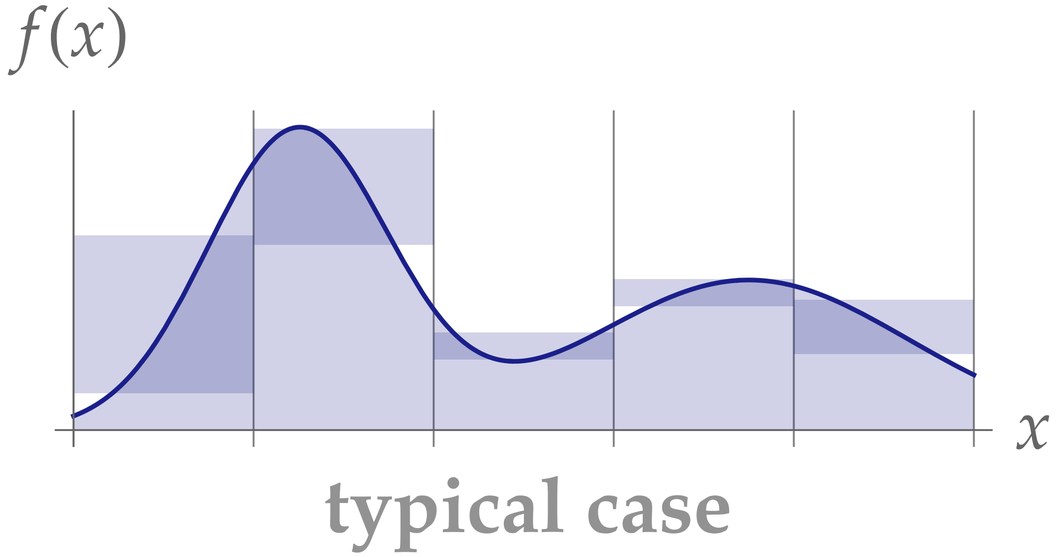
Stratified Sampling Cannot Increase Variance

- Not hard to show that stratified sampling can never increase variance

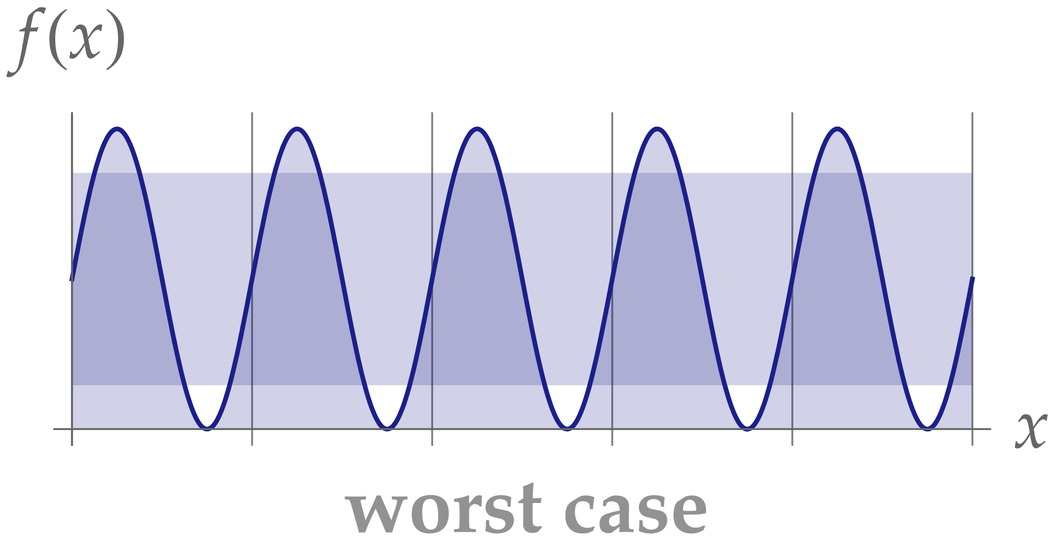
  – *won't do it here, because it's a homework exercise!* :-)

- **Basic intuition:** impossible for function to vary more within any given strata than it does over the whole domain

- Typically *decreases* variance—unless you get really unlucky!



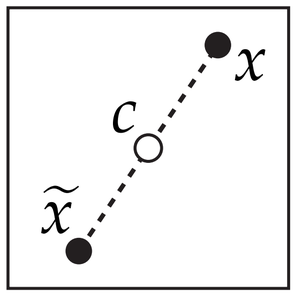
typical case



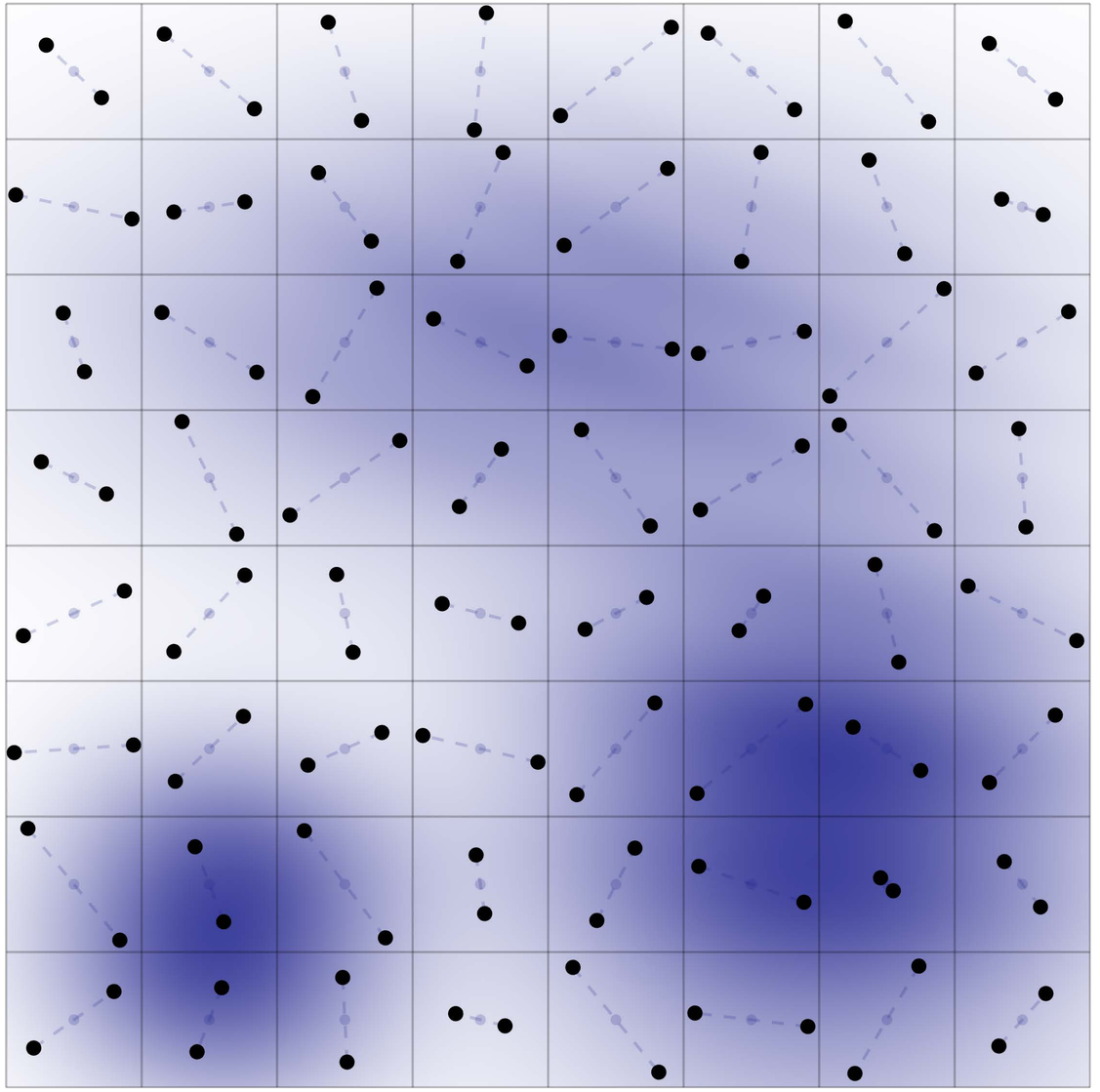
worst case

Stratification + Antithetic Sampling

- For small enough strata, smooth function is “roughly linear” in each region
- Linear functions are ideal case for antithetic sampling
- So, combine the two: take two samples in each stratum, reflected across center

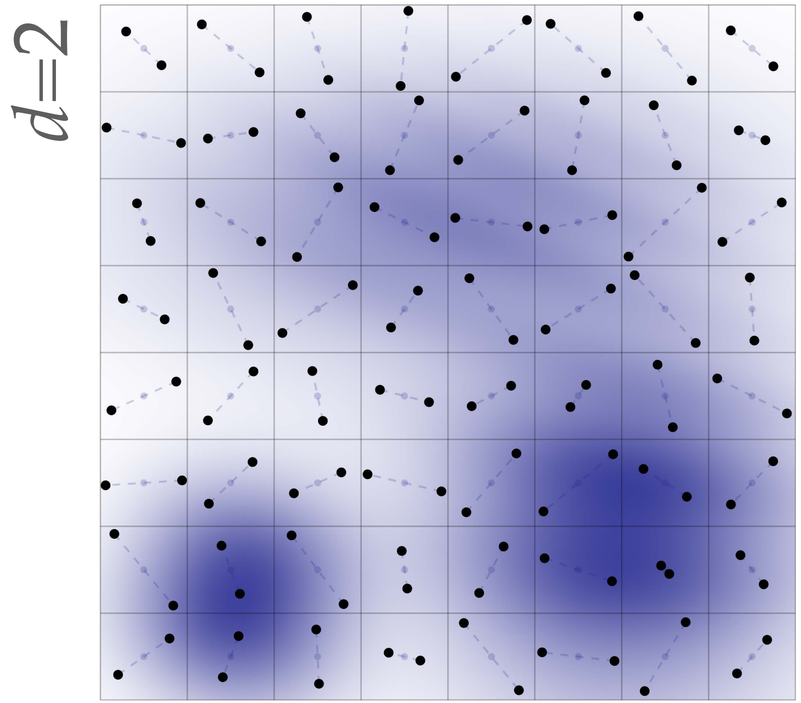


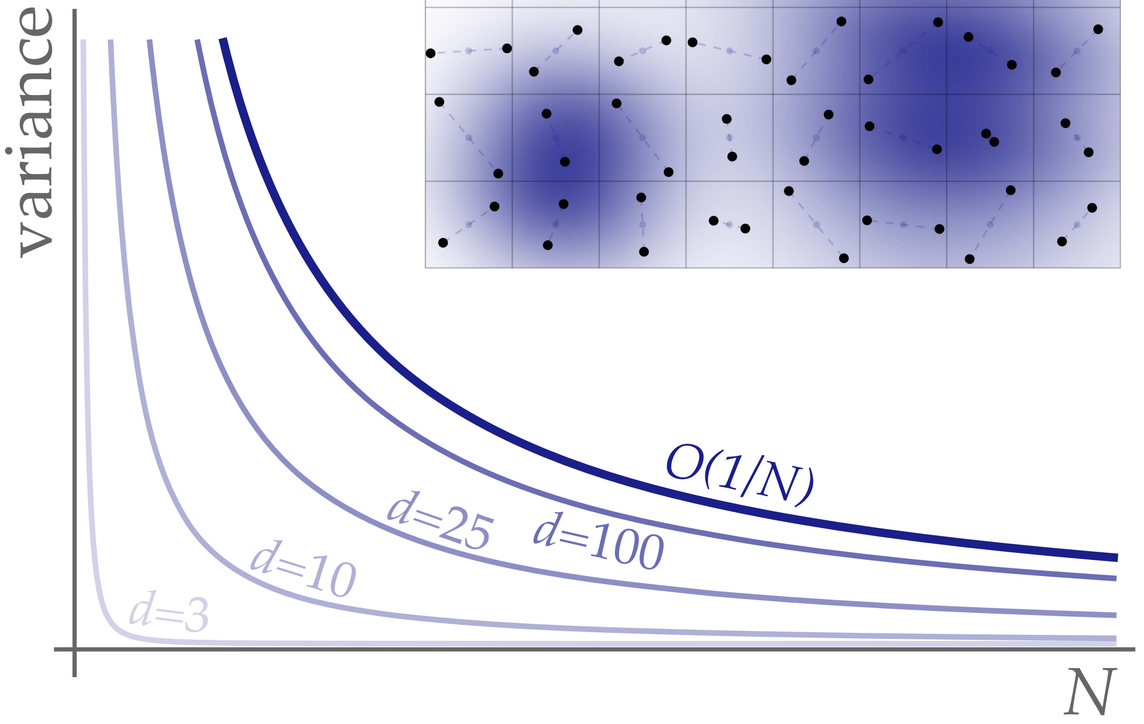
$$\tilde{x} = c - (x - c)$$



Stratification + Antithetic Sampling—Analysis

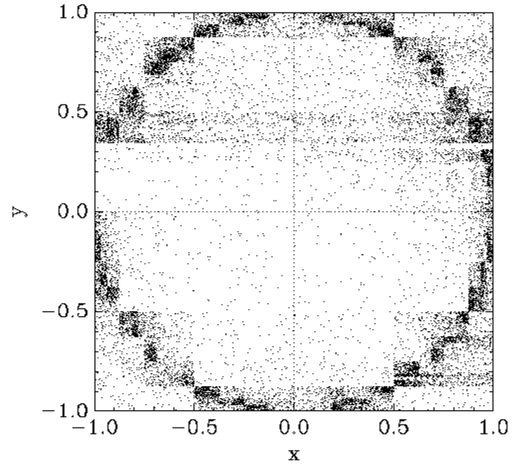
- Consider an integrand $f : [0,1]^d \to \mathbb{R}$, and stratify domain by splitting into (hyper)cubes
- If f is twice-differentiable, can show that variance of stratification + antithetic sampling is $O(n^{-1-d/4})$
  - *much* better than basic Monte Carlo in low-medium dimensions
  - see Owen §10.2 for a proof





Recursive Stratified Sampling

- Regularly-spaced strata may not do a good job of partitioning into lower-variance integrands
- **Recursive Stratified Sampling**: subdivide strata *adaptively*:
  - in each region, take k samples
  - compute the sample variance $\hat{\sigma}^2$
  - if $\hat{\sigma}/\sqrt{N}$ greater than error tolerance ε, split
  - stop when error is $< \varepsilon$, or max depth reached

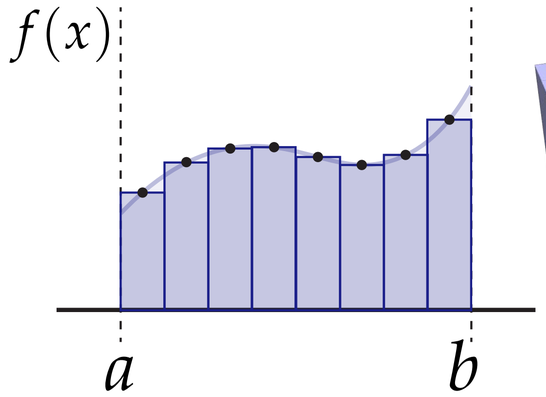


- Improvements: split along just one axis, pick best split (MISER), …
  - keep overhead in mind—may not help if evaluating $f(x)$ is super cheap
- With any adaptive scheme: *might stop too early!*

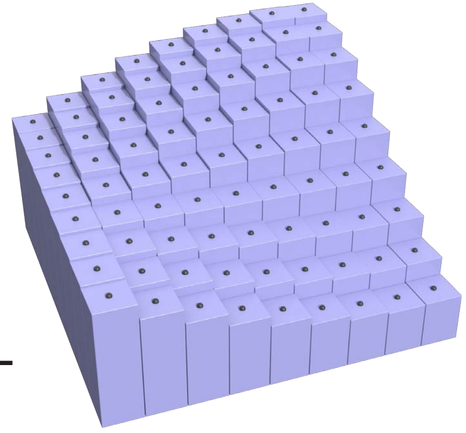
Stratified Sampling in Many Dimensions

- Naïve generalization to d dimensions: break into d-dimensional strata (e.g., d-dimensional hypercubes)
- **Q:** What's the fundamental problem with this approach?
- **A:** As d increases, face the curse of dimensionality again!
  - e.g., for 1 sample/stratum, essentially just a "jittered midpoint rule"

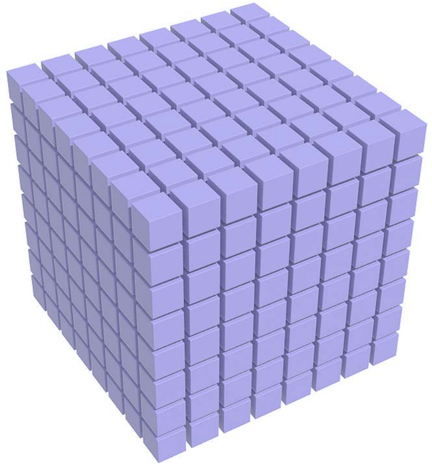
THE CURSE OF DIMENSIONALITY



$O(n)$



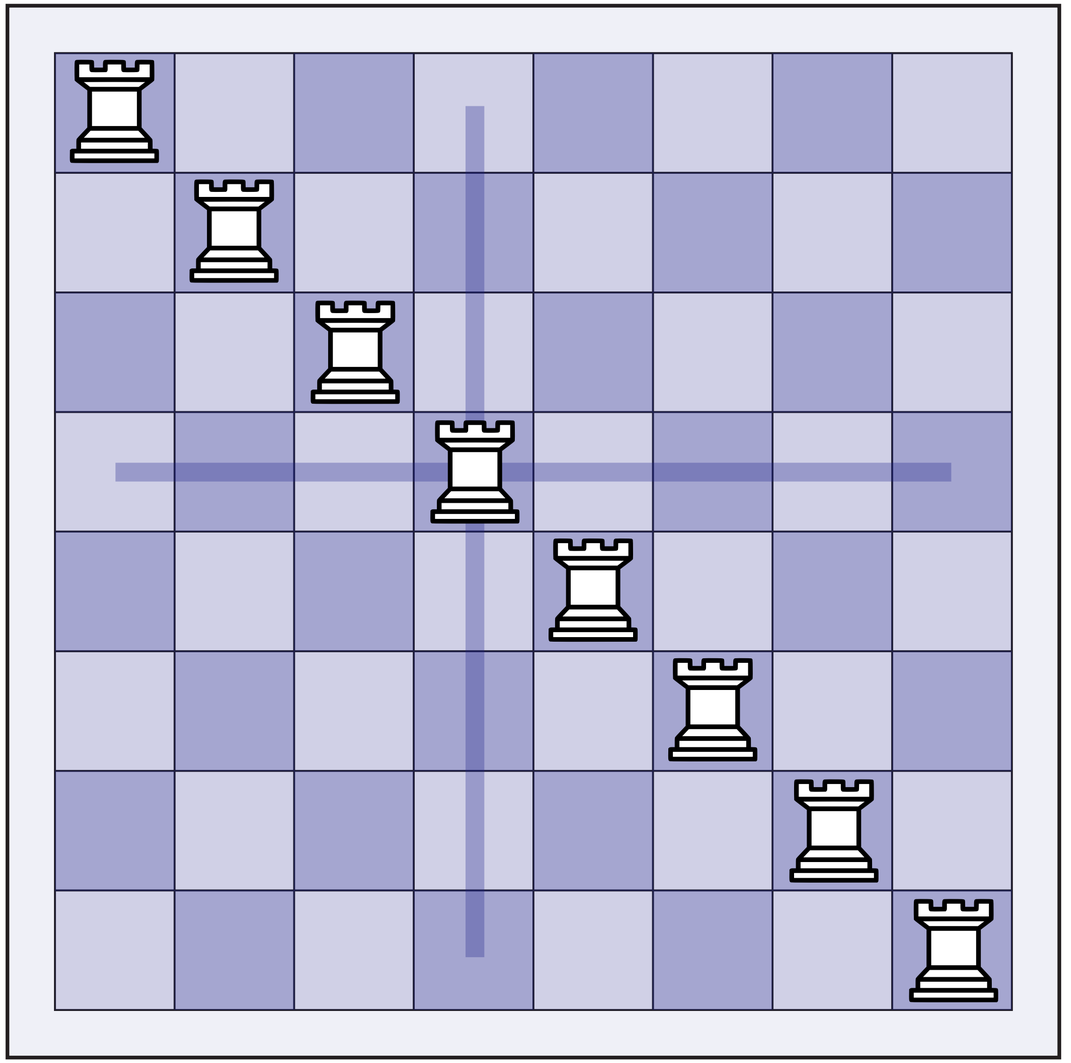
$O(n^2)$



$O(n^3) \cdots O(n^d)$

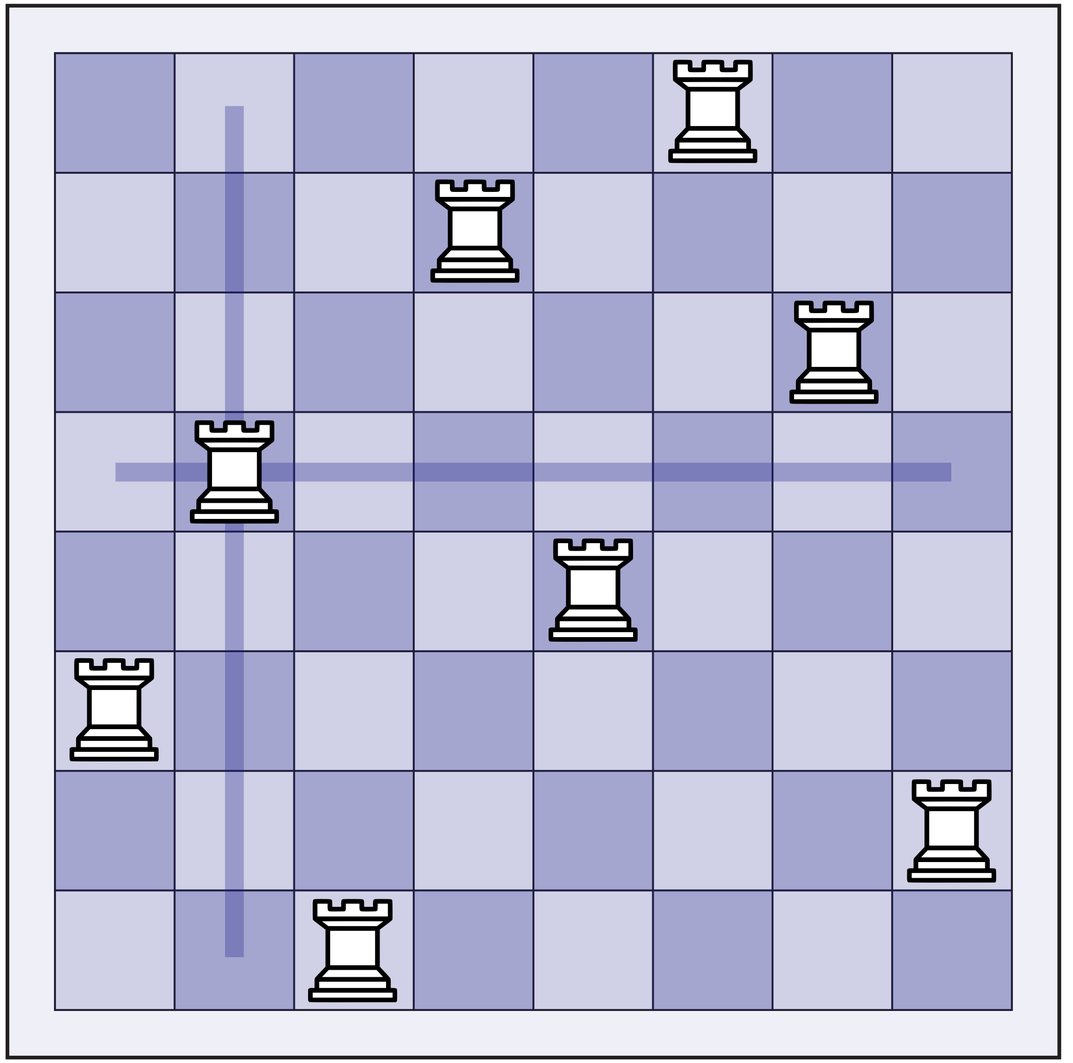
Latin Hypercube Sampling

- More clever solution in high dimensions: **Latin hypercube sampling**
- Pick *n* samples that are *simultaneously* stratified across each dimension
- Related to *n-rooks problem*: simultaneously place *n* rooks on a chess board so no rook threatens another
  - trivial solution: place along diagonal
  - apply permutation to get other solutions



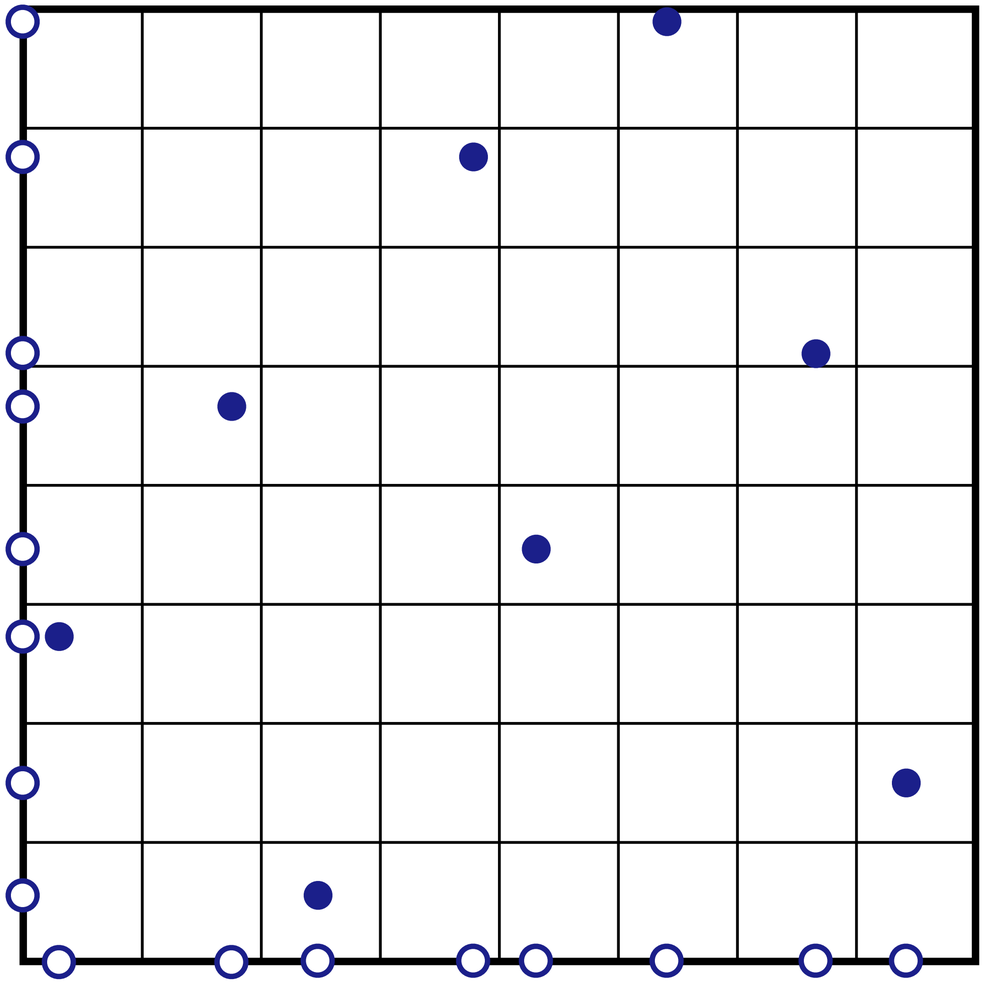
Latin Hypercube Sampling

- More clever solution in high dimensions: **Latin hypercube sampling**
- Pick *n* samples that are *simultaneously* stratified across each dimension
- Related to *n-rooks problem*: simultaneously place *n* rooks on a chess board so no rook threatens another
  - trivial solution: place along diagonal
  - apply permutation to get other solutions



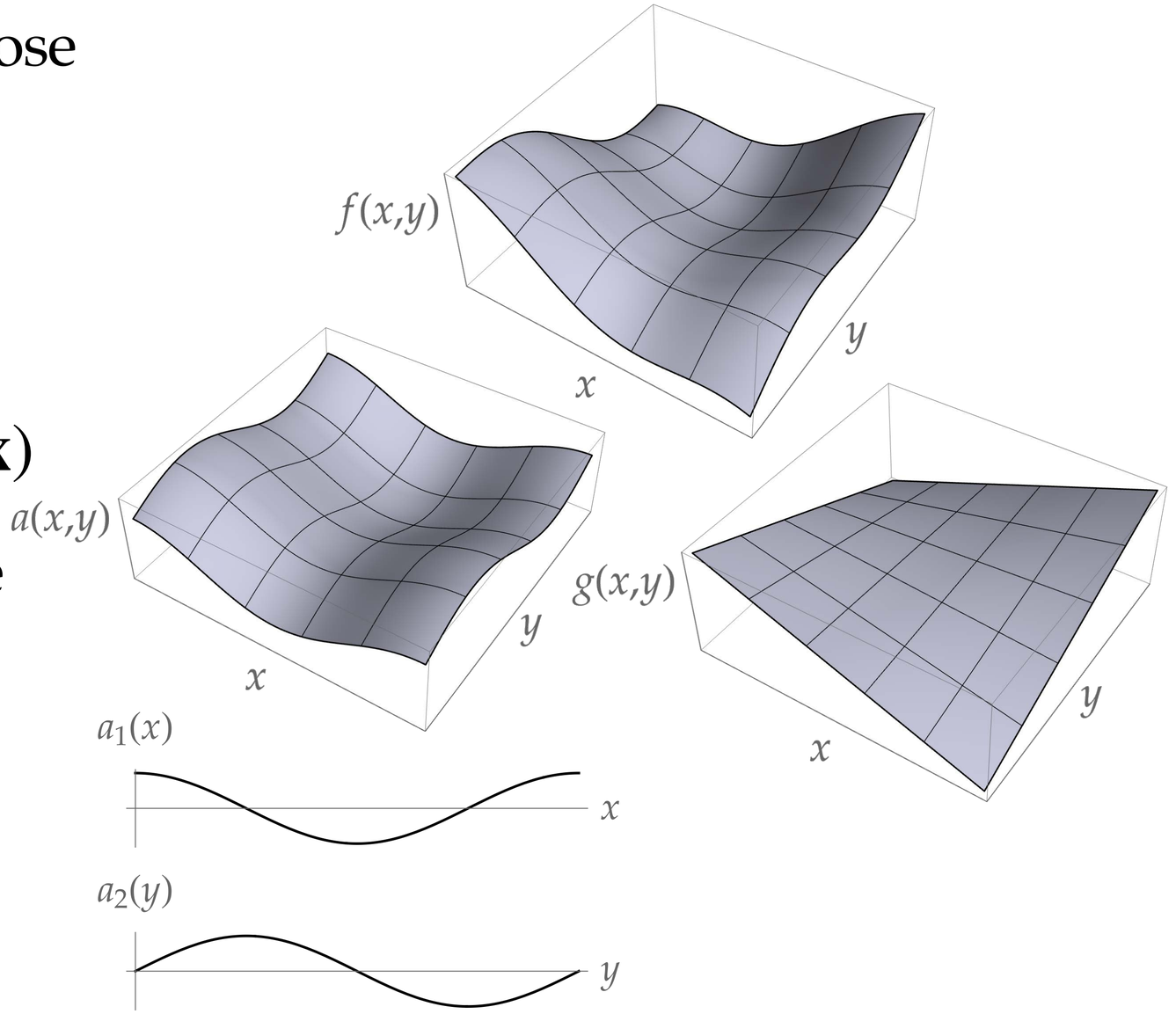
Latin Hypercube Sampling

- More clever solution in high dimensions: **Latin hypercube sampling**
- Pick *n* samples that are *simultaneously* stratified across each dimension
- Related to *n-rooks problem*: simultaneously place *n* rooks on a chess board so no rook threatens another
  - trivial solution: place along diagonal
  - apply permutation to get other solutions
- Latin hypercube: perturb each sample within stratum



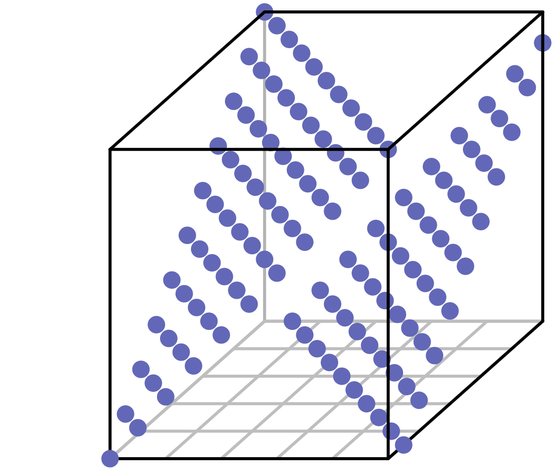
Latin Hypercube Sampling—Analysis

- Suppose in d dimensions we decompose integrand as $f(\mathbf{x}) = a(\mathbf{x}) + g(\mathbf{x})$ into
  - **additive part** $a(\mathbf{x}) = a_1(x_1) + \cdots + a_d(x_d)$
  - **non-additive part** $g(\mathbf{x}) = f(\mathbf{x}) - a(\mathbf{x})$
- Then asymptotically, Latin hypercube sampling is <u>unaffected by variance of the additive part</u>
  - see Owen §10.3, Proposition 10.1

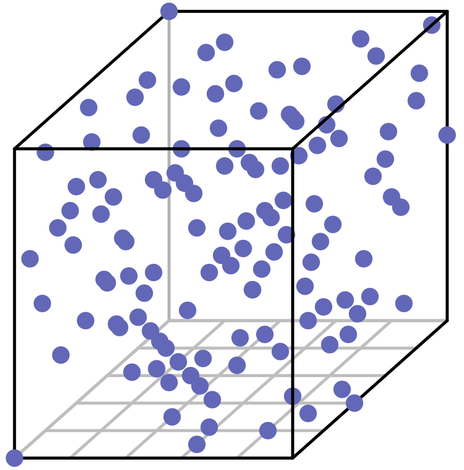


Orthogonal Array Sampling

- Latin hypercube sampling ensures all 1D projections are stratified
- Can take this idea further using an **orthogonal array**, where all m-dimensional projections are stratified, for $m \geq 2$
- Standard constructions (e.g., *Bose orthogonal array*) are akin to placing all rooks on the diagonal; by randomizing, we get good samples for Monte Carlo
- See Owen §10.4 for further discussion



orthogonal array (m=2, d=3)

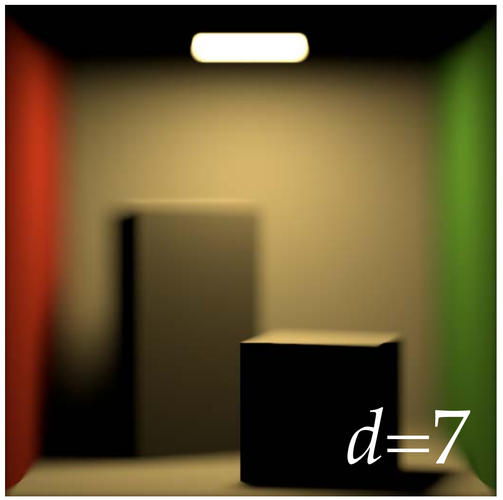


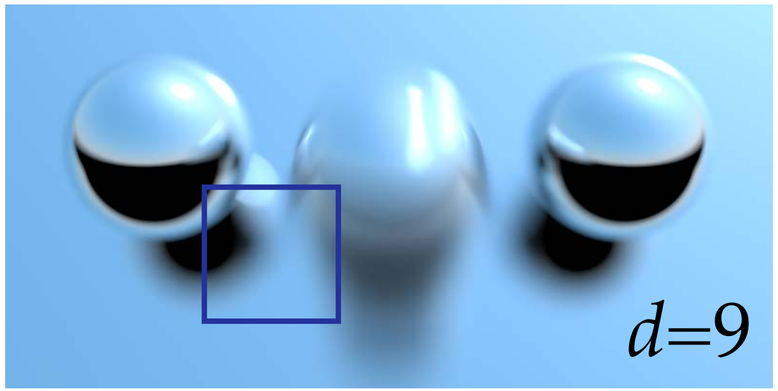
randomized orthogonal array

(images adapted from Owen, *"Monte Carlo Theory, Methods and Examples"*)

Application— Photorealistic Rendering

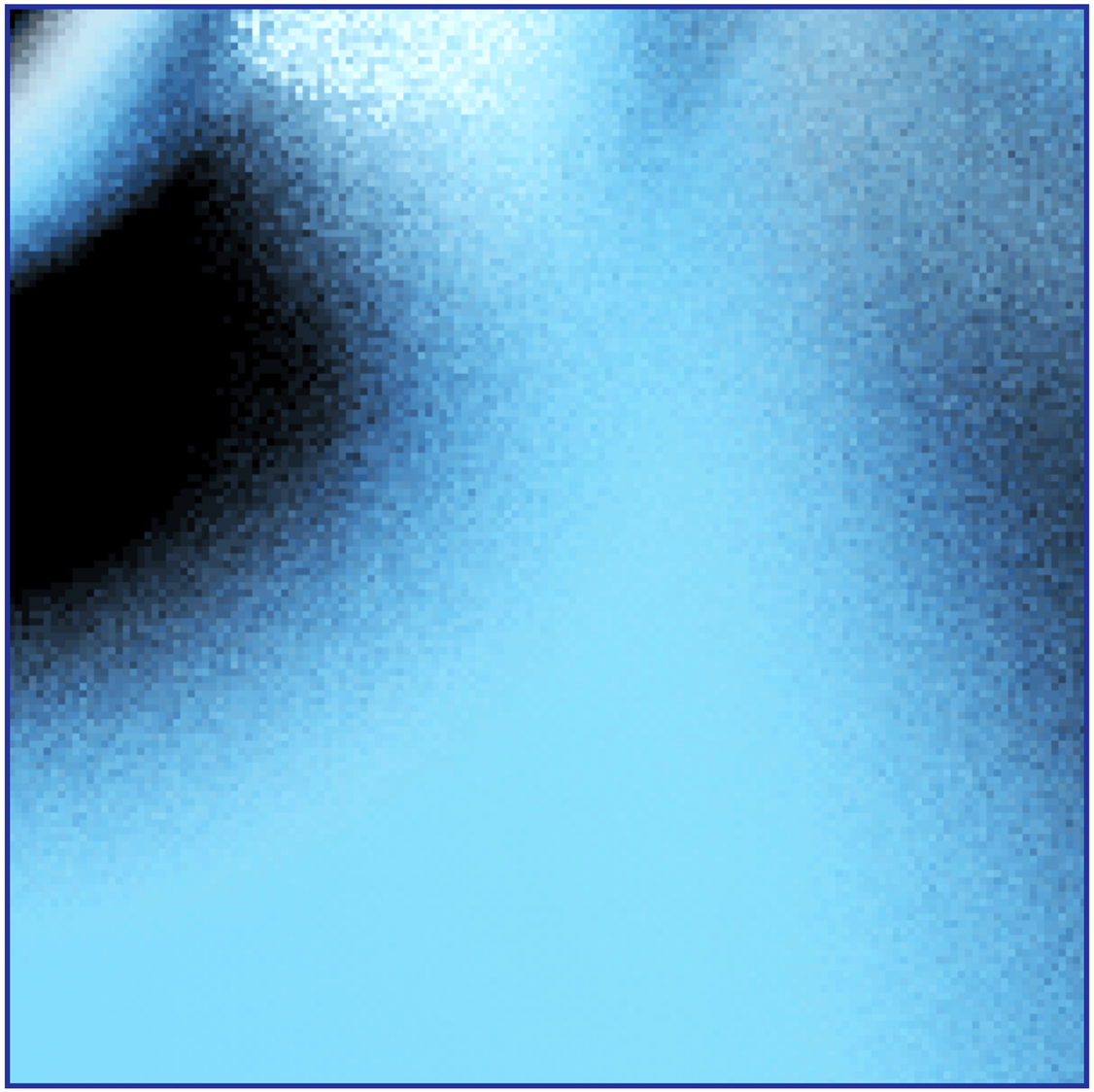
To determine illumination, must integrate over temporal & many spatial dimensions (motion blur, depth of field, bounces of light…)





Jarosz et al, *"Orthogonal Array Sampling for Monte Carlo Rendering"* (2019)

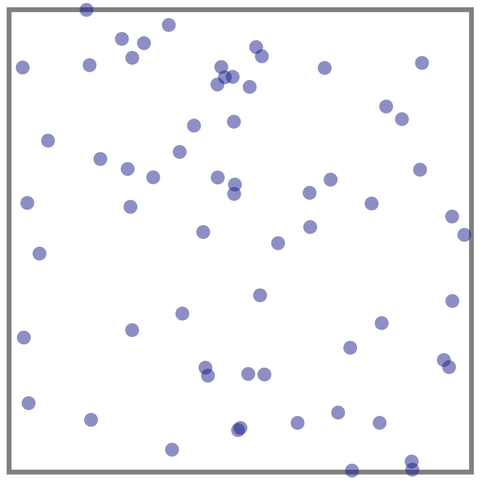
pairwise stratification vs. orthogonal array



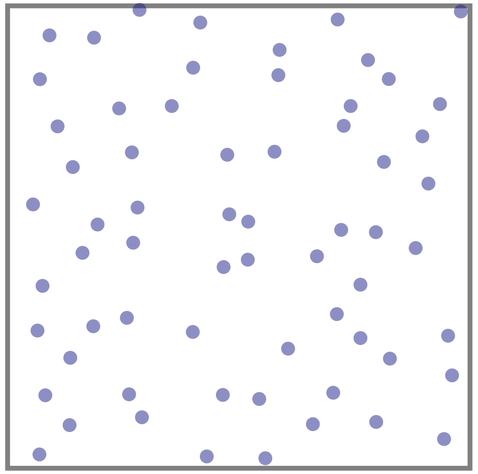
Quasi Monte Carlo

Quasi Monte Carlo (QMC)

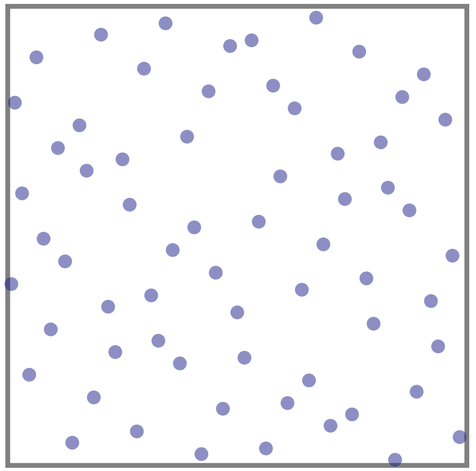
- Already saw uniform random isn't ideal (points "clump up")
- Used *stratification* to make sure we have at least some samples in each region of domain
- Can take this even further: forget about random sampling altogether, and replace with deterministic **low-discrepancy sequence**



uniform



stratified



low discrepancy

Quasi Monte Carlo (QMC)

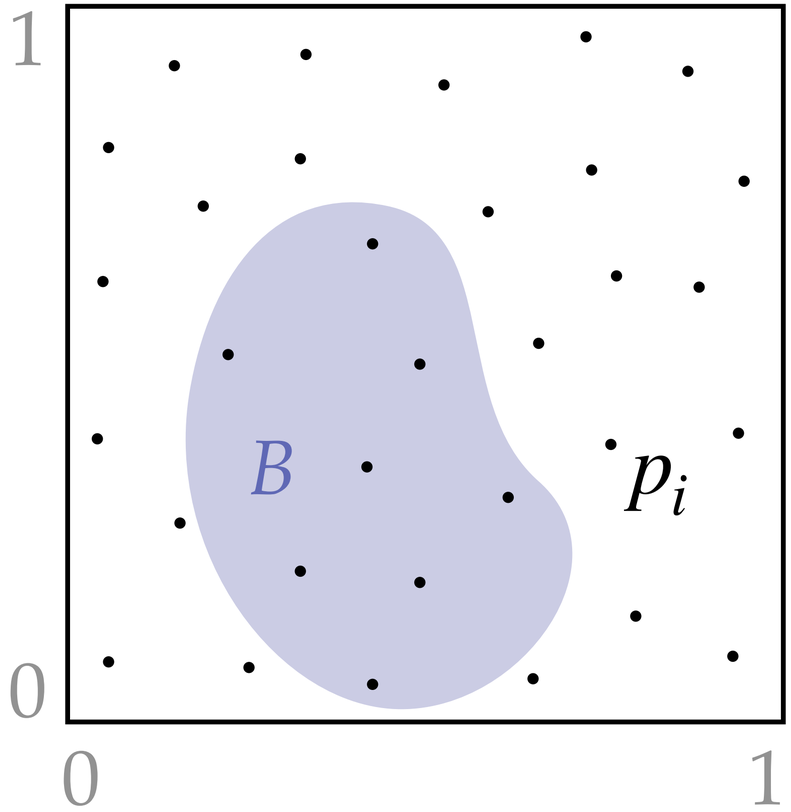
- **Intuition:** low-discrepancy points $p_1, \ldots, p_N$ are spread out "evenly"
- Estimators otherwise look the same, e.g., $(|\Omega|/N)\sum_{k=1}^{N} f(p_k)$
  - Or, can use in conjunction with any other variance reduction strategy
- QMC can achieve <u>asymptotically</u> faster convergence than plain MC
  - not without some drawbacks (will discuss later)

Discrepancy of Point Sets

- What makes a set of sample points $P_N := \{p_1, \ldots, p_N\}$ "good"?
- One feature we might like: <u>number</u> of points covered by a "random" region is proportional to the <u>area</u> of the region.
- Several ways to formalize—e.g., for points P in unit cube $C_d := [0,1]^d$,

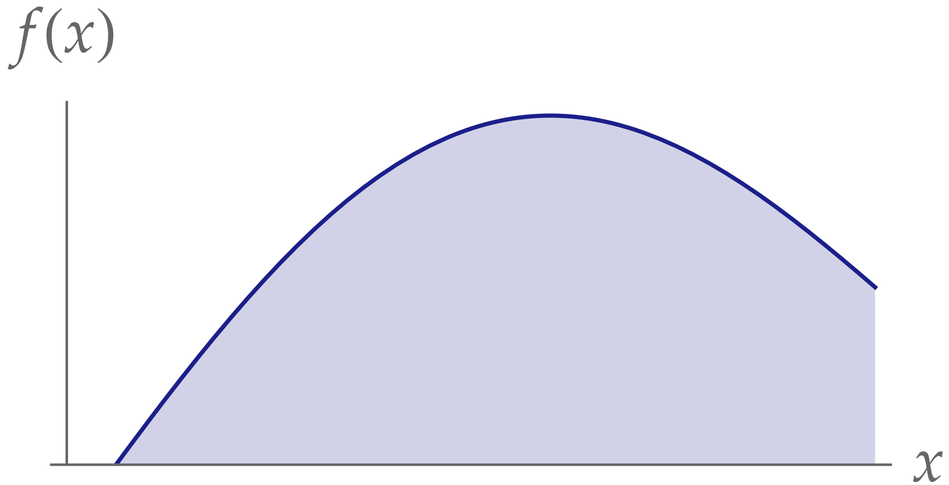
  $$D(P_N) := \sup_{B \in J} \left| \frac{B \cap P_N}{N} - \mathrm{vol}(B) \right|$$ for some family of subsets J of C_d

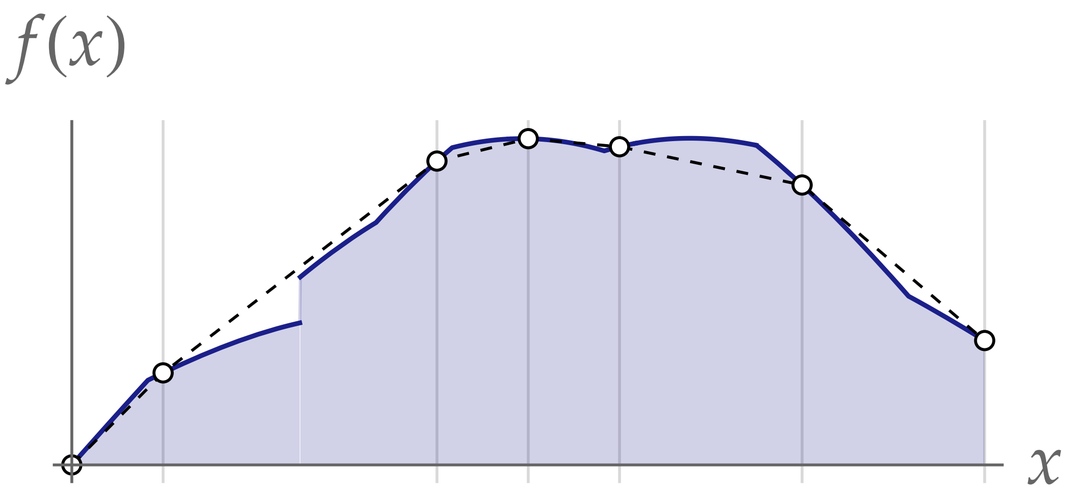
  – e.g., J could be all intervals, half intervals, …



Total Variation

- The **total variation** of a function captures, well, the total amount the function "varies" over its domain!
- For a differentiable function $f : [a, b] \to \mathbb{R}$, $TV(f) := \int_a^b |f'(x)|\ dx$
- In general: $TV(f) := \sup_{P[a,b]} \sum_{k=0}^{|P|-1} |f(x_{i+1} - f(x_i)|$, where $P[a, b]$ are partitions of [a,b]
- **Note:** *total variation* is not the same as *variance*!
  - harder to compute (need derivatives, or search over partitions)





Koksma–Hlawka Inequality

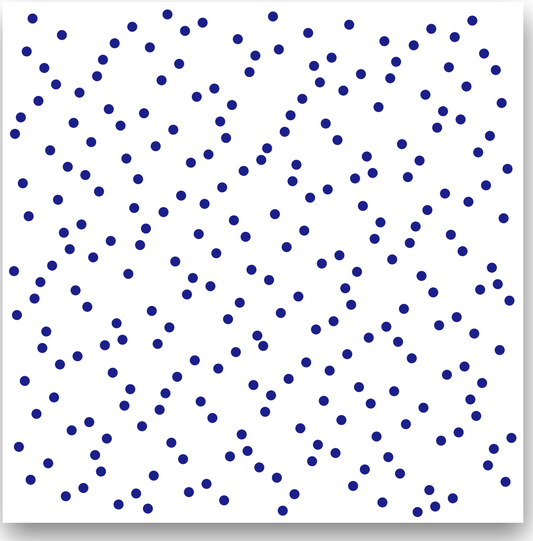
- Can bound the error of quasi-Monte Carlo estimator in terms of discrepancy of point set $D(P_k)$ and total variation of integrand $TV(f)$.
- E.g., *Koksma-Hlawka inequality*:

$$\left| \int_{\Omega} f(x)\ dx - \frac{|\Omega|}{N} \sum_{k=1}^{N} f(p_k) \right| \leq D(P_N) TV(f)$$

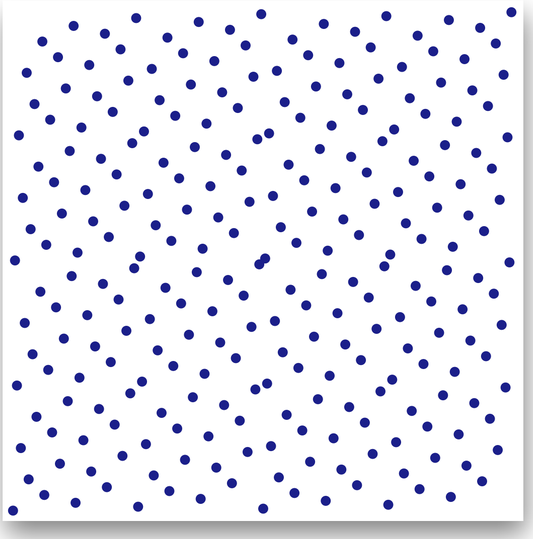
- Can in turn show QMC has error $O((\log N)^d / N)$
- In high dimensions, this means QMC converges much faster than MC: more like $O(1/N)$ than $O(1/\sqrt{N})$.

Low-Discrepancy Sequences

- Many established low-discrepancy sequences, with different properties
- **Halton sequence** — can generate progressively, higher discrepancy than some alternatives
- **Hammersley points** — must know how many you want up-front, but lower discrepancy
- Will talk about *how* to generate such sequences in next lecture (connected to pseudorandom number generation)



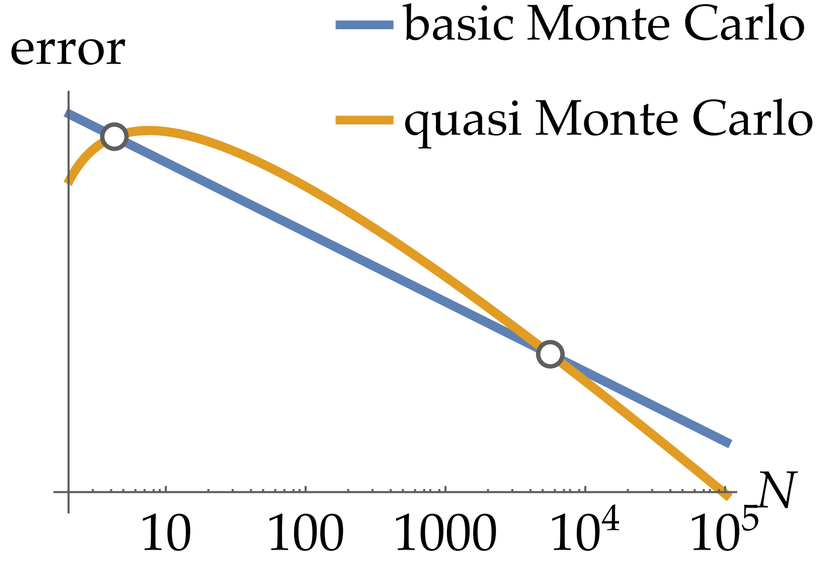
Halton



Hammersley

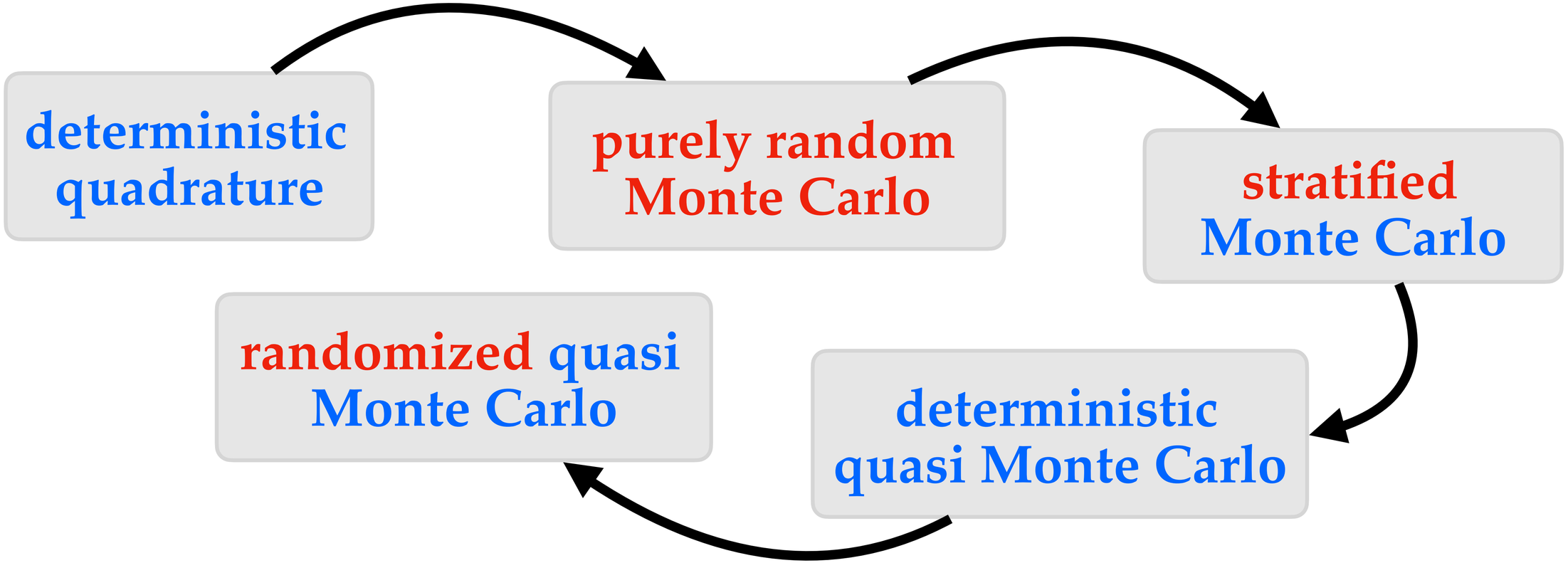
QMC—Pros & Cons

- Harder to quantify error
  - basic Monte Carlo: $\sigma/\sqrt{N}$ (easy to estimate)
  - QMC: total variation times discrepancy (hard to estimate)
- Better *asymptotic* convergence than ordinary MC
  - …but still takes a long time (large N) to beat MC in higher dimensions
  - *randomized* QMC (RQCM) can help (see Owen, Ch. 17)
  - convergence generally harder to analyze (not just LLN/CLT)
- Can fix some issues of QMC via **randomized quasi Monte Carlo (RQMC)**, i.e., nondeterministic random *"scrambling"*

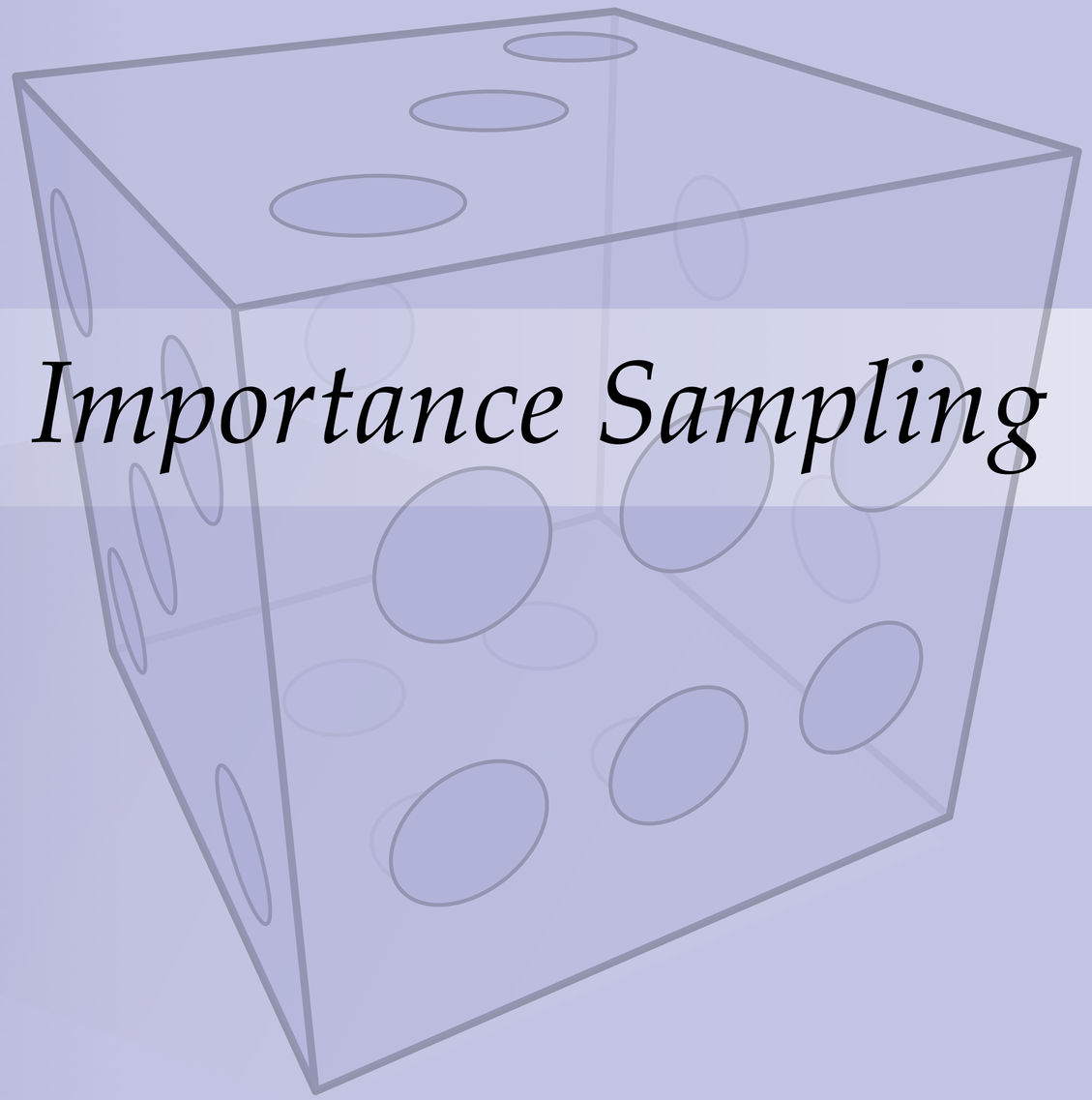


From random to deterministic and back again…

To summarize, have a somewhat funny story:



Clearly there's a tension/balance between deterministic & random sampling!



Importance Sampling

Importance Sampling

- **Importance sampling** replaces basic Monte Carlo estimator $\hat{I}_N$ with *importance sampled estimator* $\hat{I}^p_N$
  - $p : \Omega \to [0,1]$ is *importance density* on domain Ω (integrates to 1)
  - $p(x)$ must be nonzero wherever $f(x) \neq 0$
  - $p(x)$ should be roughly proportional to $f(x)$
- Two interpretations:
  1. **biasing** the sampling
  2. **stretching** the domain

integral

$$I = \int_\Omega f(x)\, dx$$

basic estimator

domain size $|\Omega|$

$$\hat{I}_N := \frac{|\Omega|}{N} \sum_{i=1}^{N} f(X_i), \quad X_i \sim \mathcal{U}_\Omega$$

sample uniformly

Equivalent to basic estimator when $p(x) = 1/|\Omega|$

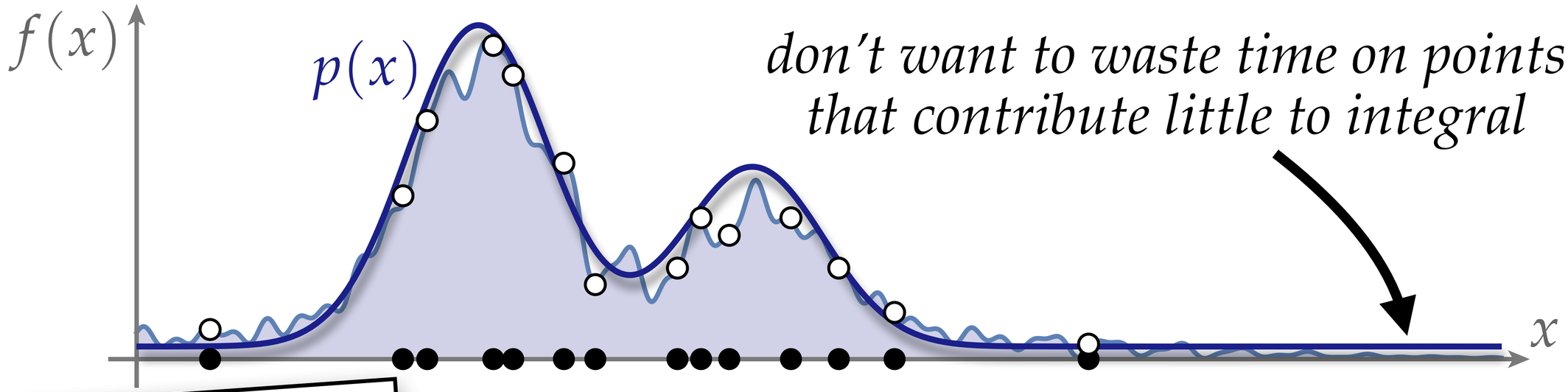
importance sampled estimator

$$\hat{I}^p_N := \frac{1}{N} \sum_{i=1}^{N} \frac{f(X_i)}{p(X_i)}, \quad X_i \sim p$$

sample from p

Importance Sampling—Biasing Viewpoint

Uniform sampling can sample many values close to zero:



don't want to waste time on points that contribute little to integral

Idea: concentrate samples where integrand is large.

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

account for over/ under sampling

"draw more samples where p is large"

$$X_i \sim p$$

Note: "biasing" ≠ biased!

Biasing Viewpoint—Ideal Case

Q: What happens if we draw samples from a distribution $p(x)$ exactly proportional to $f(x)$?

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

(assuming f is nonnegative)

importance sampled estimator

$$\widehat{I}_N^p := \frac{1}{N}\sum_{i=1}^{N}\frac{f(X_i)}{p(X_i)}, \quad X_i \sim p$$

Since p is a probability density function, must integrate to 1:

$$1 = \int_\Omega p(x)\,dx = c\int_\Omega f(x)\,dx \quad \Longleftrightarrow \quad c = 1/I$$

constant of proportionality is the integral we're looking for!

So, let's plug $p(x) = f(x)/I$ back into our importance sampled estimator…

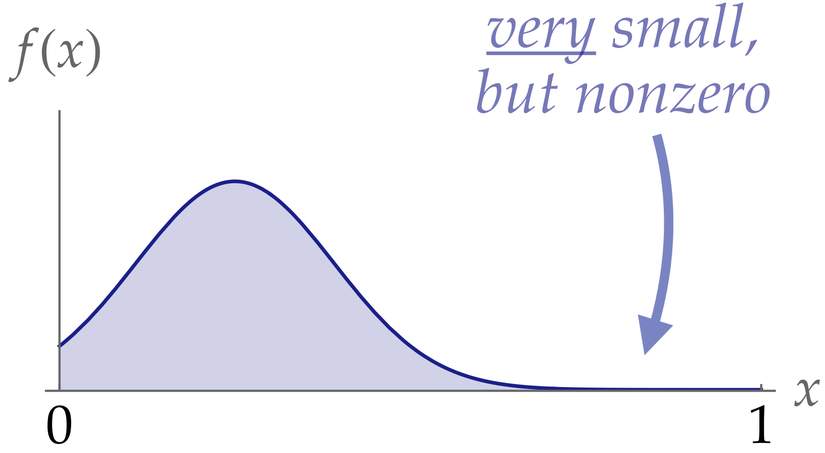
$$\frac{f(X_i)}{p(X_i)} = \frac{f(X_i)}{f(X_i)/I} = I$$

Q: How many samples do we now need to get a good estimate? :-)

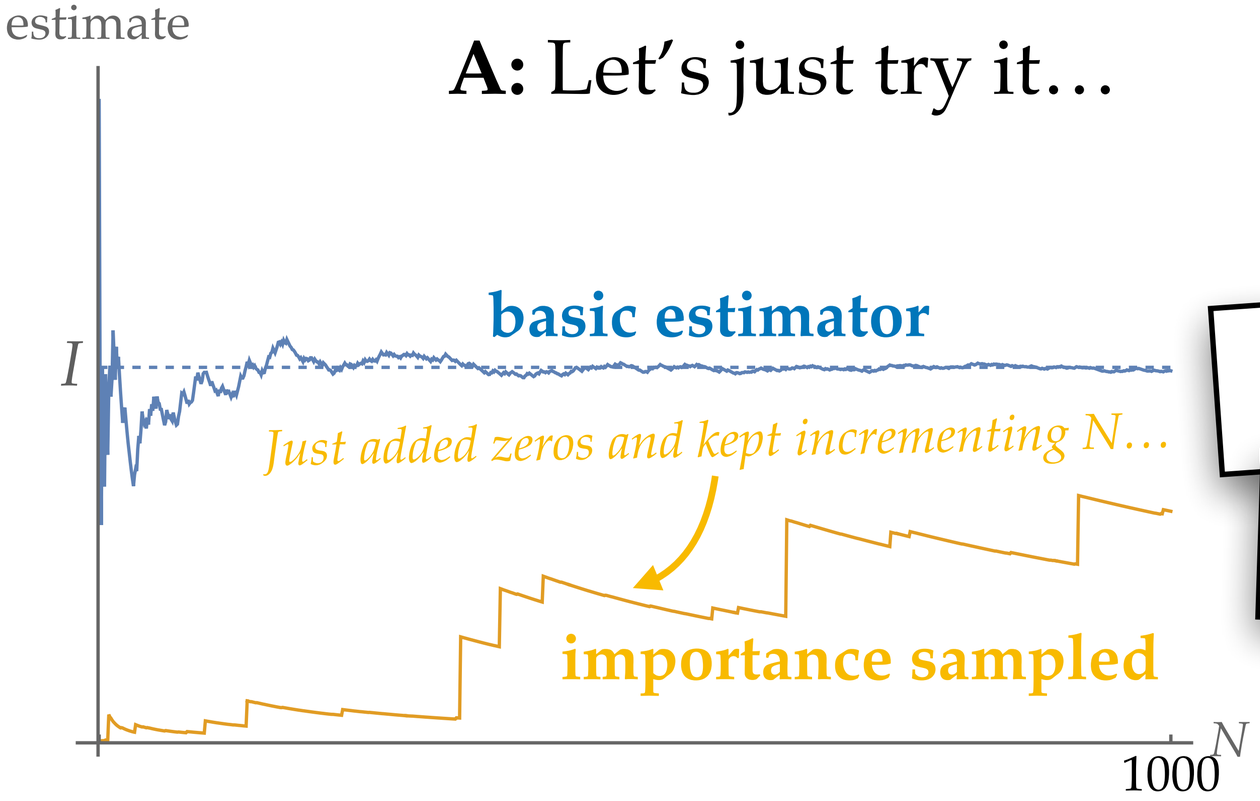
A: Just one! (Immediately get the *exact* integral.)

Biasing Viewpoint—Worst Case

Q: What happens if we instead draw samples from a distribution $p(x)$ that is large where $f(x)$ is small, and vice-versa? E.g., $p(x) = c/f(x)$. *(assuming f is strictly positive)*

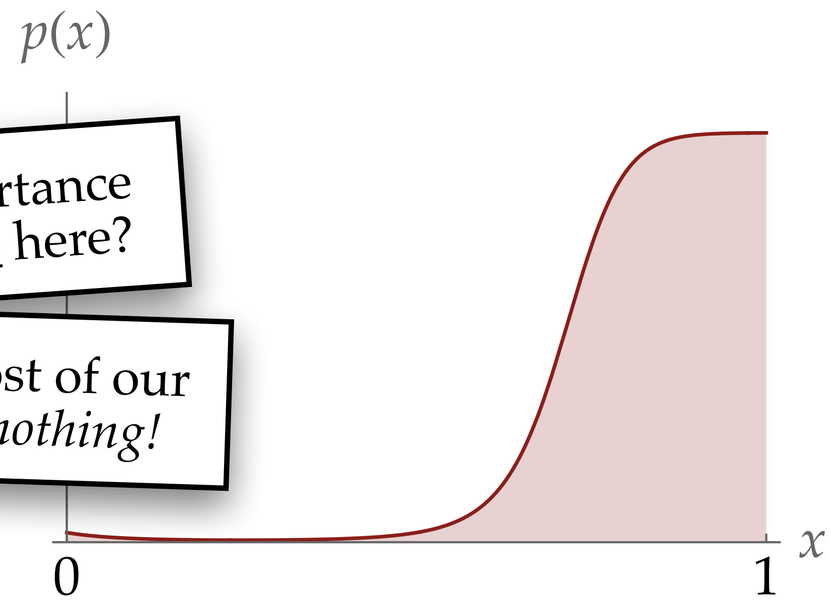


A: Let's just try it…



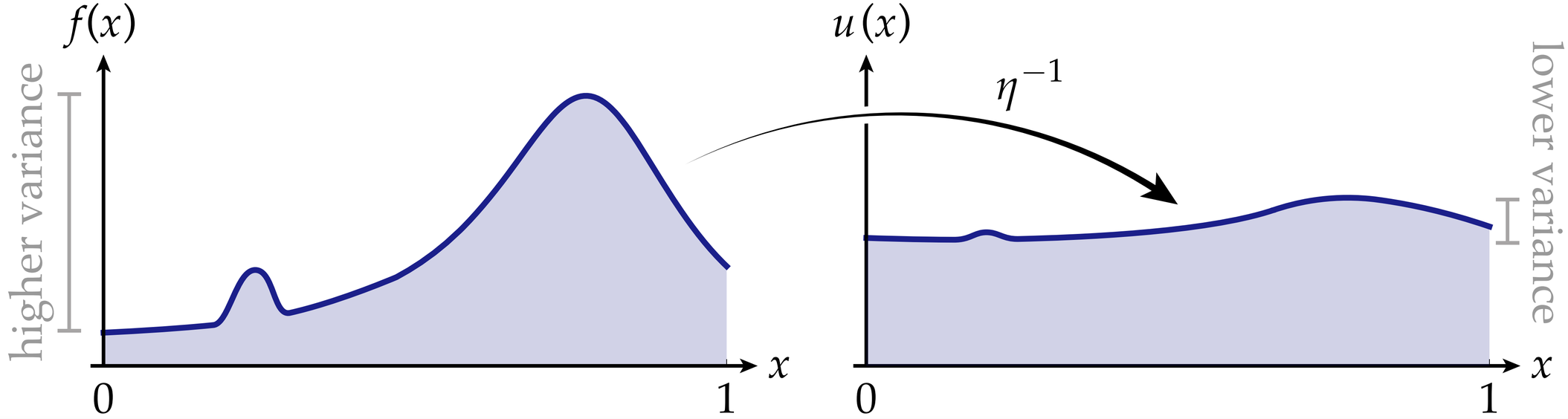
Q: Why is importance sampling worse here?

A: Because most of our samples did *nothing!*



Stretching Viewpoint—Intuition

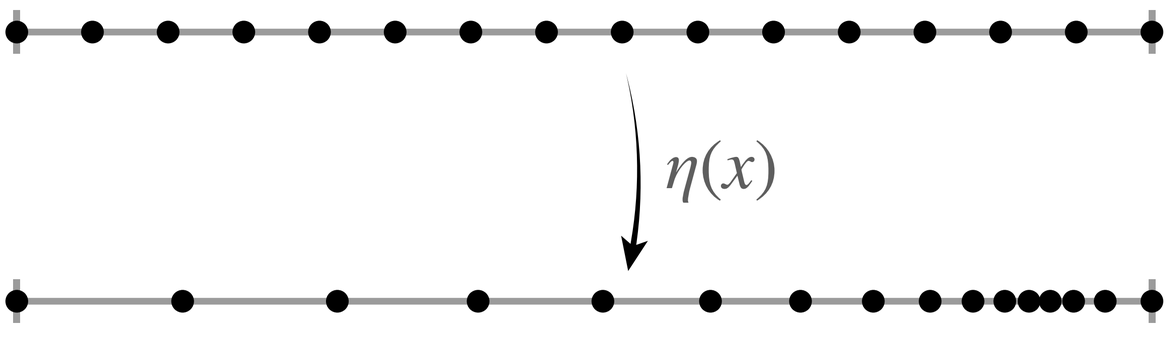
- Alternative interpretation of importance sampling:
  - "squash and stretch" domain so integrand becomes more uniform
  - apply **basic Monte Carlo** to "stretched" integrand
  - same as applying **importance sampled estimator** to original integrand

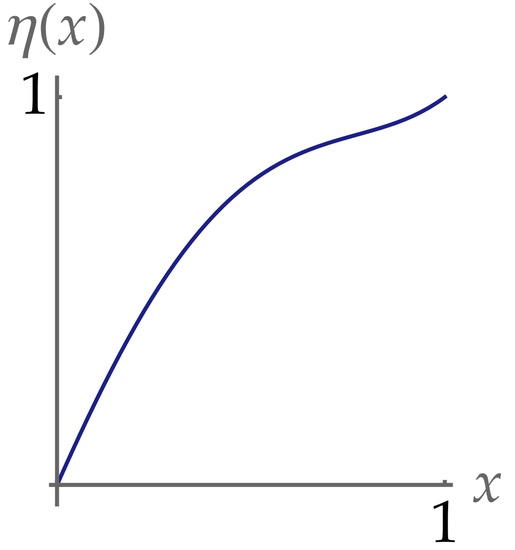


All Monte Carlo acceleration techniques (including importance sampling) can ultimately be viewed as transformations that yield a lower-variance integrand.

Integration under Reparameterization

- Consider any function $u : [0,1] \to \mathbb{R}$
- Let $\eta : [0,1] \to [0,1]$ be a *reparameterization* of the domain (*i.e.*, a monotonically increasing one-to-one function)
- Can compose this reparameterization with the original integrand to get the *pullback* $u^*(x) := u(\eta(x))$
- To get the same integral before/after reparameterization, use $|\eta'(x)|$ to account for "bunching up" of the domain





$$\int_0^1 u(x)\,dx = \int_0^1 u^*(x)|\eta'|(x)\,dx$$

Stretching Viewpoint

Ideal case. Consider the constant function $u(x) := I$, and imagine we could find a reparameterization $\eta : [0,1] \to [0,1]$ such that

$$u(\eta(x))|\eta'(x)| = f(x) \iff \frac{f(x)}{|\eta'(x)|} = u(\eta(x))$$

Since η is a reparameterization, we have

$$\int_0^1 |\eta'(x)|\, dx = \eta(1) - \eta(0) = 1 - 0 = 1$$

Hence, can view $\eta'(x)$ as our probability density $p(x)$, and get

$$\frac{f(x)}{p(x)} = u(\eta(x)) = I$$

More generally: the more uniformly we "stretch out" the integrand, the lower variance becomes.

I.e., if we "perfectly stretch out" our integrand, then we get the exact integral with one sample (as before).

Stretching Viewpoint—General

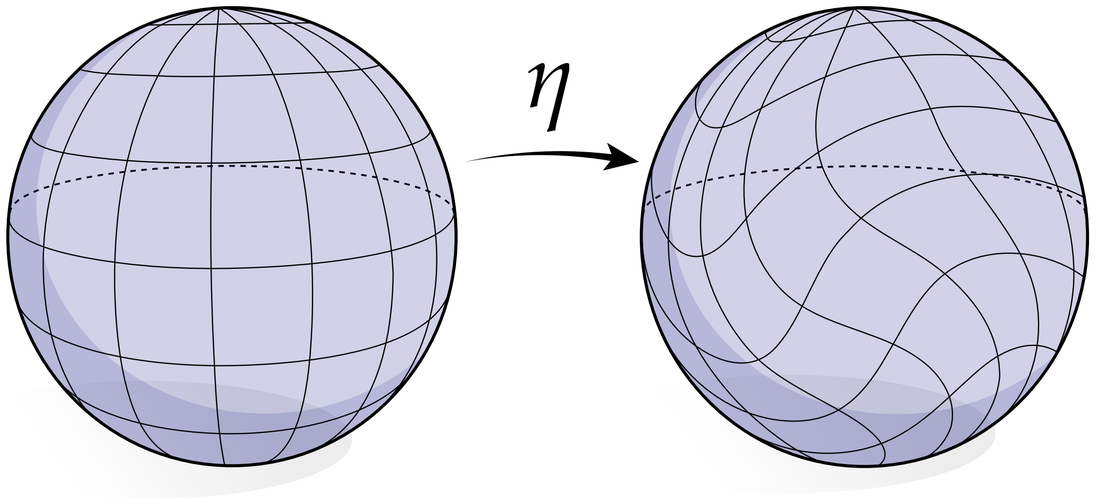
- More generally, for any (manifold) domain Ω, can consider a reparameterization $\eta : \Omega \to \Omega$ given by a differentiable bijection with continuous inverse (*diffeomorphism)*

- Under this reparameterization, we have an equivalence of integrals

$$\int_\Omega u \, dV = \int_\Omega u \circ \eta \, \det(d\eta) dV$$

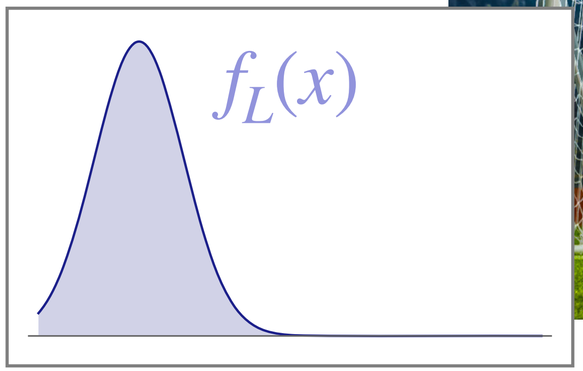
local change in volume

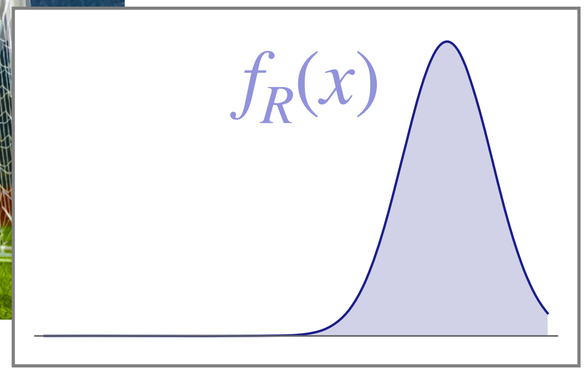
- (The rest of the interpretation is unchanged.)



Combining Importance Sampling Strategies

- In practice, may have more than one importance sampling strategy that seems promising. Which one should you use?
- **<u>Metaphor.</u>** Suppose I'm a soccer goalie, and know my opponent will either shoot left *or* shoot right—but no great way to predict which one
  - want "robust" strategy that works well no matter what happens





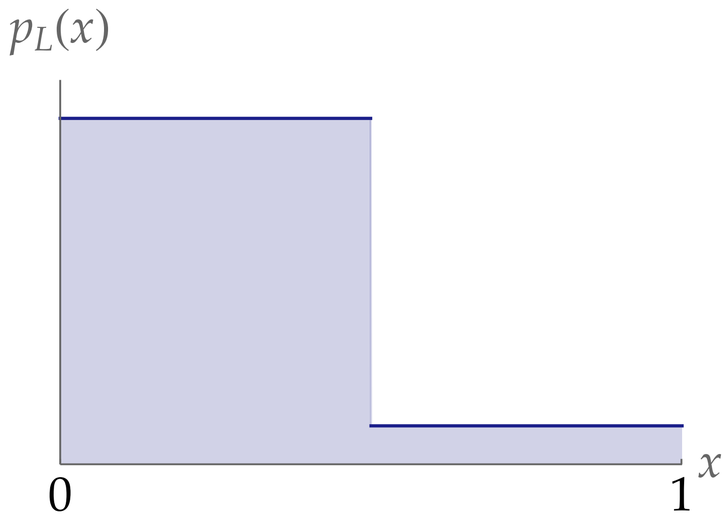
Combining Importance Sampling Strategies

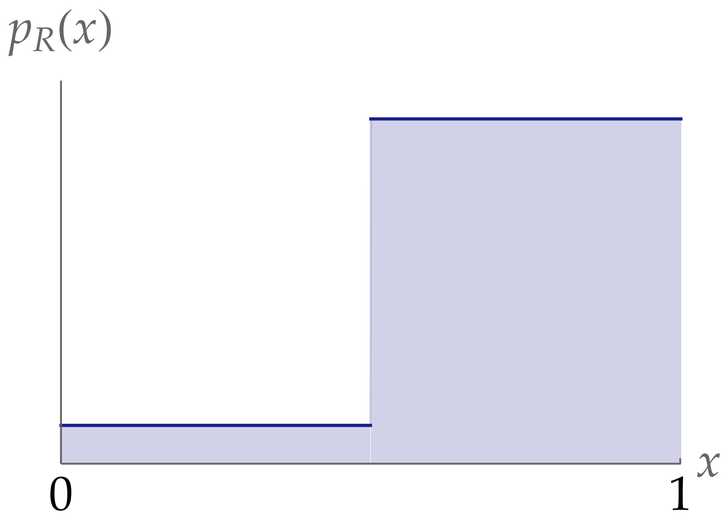
Example. Consider the parameterized family of functions

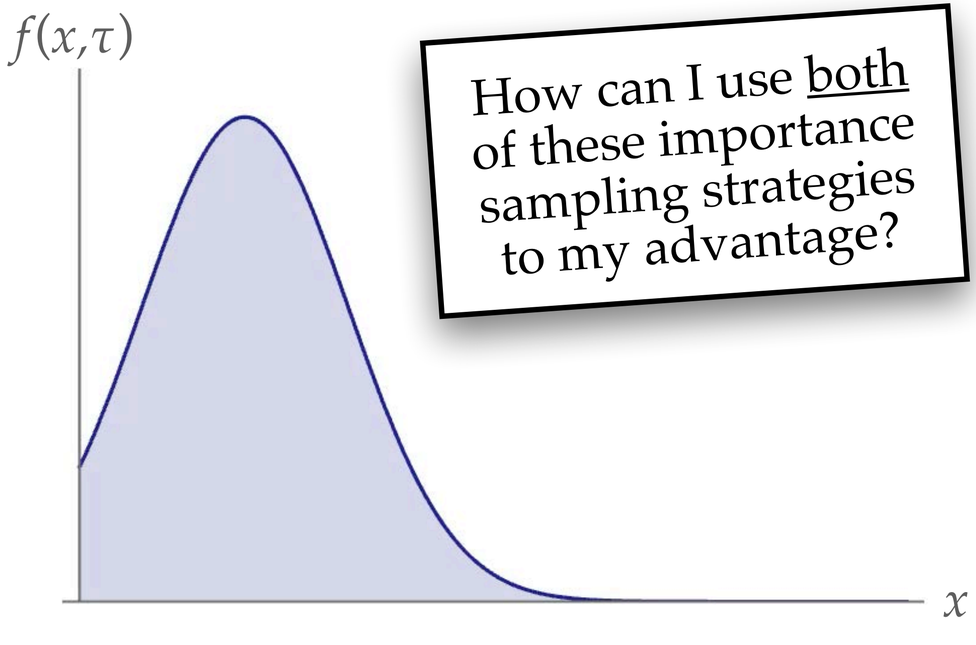
$$f(x,\tau) := (1-\tau)\mathcal{N}_{\frac{1}{5},\frac{1}{8}}(x) + \tau\mathcal{N}_{\frac{4}{5},\frac{1}{8}}(x)$$

where $\mathcal{N}_{\mu,\sigma}$ is a normal distribution with mean μ, standard deviation σ.

Let $p_L(x)$ and $p_R(x)$ be piecewise constant importance densities that roughly approximate $f(x,0)$ and $f(x,1)$, respectively.







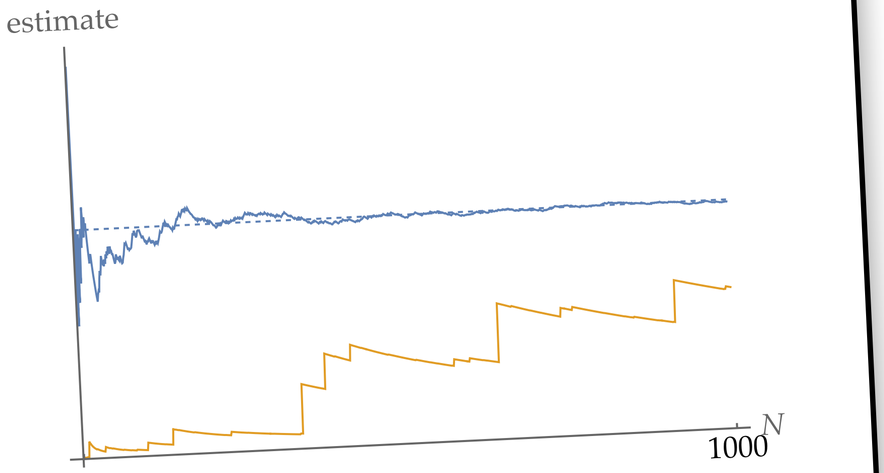
How can I use both of these importance sampling strategies to my advantage?

Combining Strategies—Naïve Approach

- For lack of any better ideas, you might:
  - estimate the integral using $p_L(x) \implies \hat{I}_L$
  - estimate the integral using $p_R(x) \implies \hat{I}_R$
  - take the average of the two estimates
- Unfortunately this naïve approach will be "corrupted" by the worst strategy, since (due to independence)

$$V\left[\frac{\hat{I}_L + \hat{I}_R}{2}\right] = \tfrac{1}{2}V[\hat{I}_L] + \tfrac{1}{2}V[\hat{I}_R]$$

Remember: poor choice of importance density can make things really bad!



Multiple Importance Sampling

Multiple importance sampling (MIS) combines several strategies, while remaining <u>provably</u> close to optimal

multi-sample estimator

number of strategies s — *samples for kth strategy* N_k — *ith sample drawn with kth strategy* $X_{k,i}$

$$\widehat{I}_{\text{MIS}} := \sum_{k=1}^{s} \frac{1}{N_k} \sum_{i=1}^{N_k} w_k(X_{k,i}) \frac{f(X_{k,i})}{p_k(X_{k,i})}$$

w_k: *weights*; p_k: *density for kth strategy*

balance heuristic

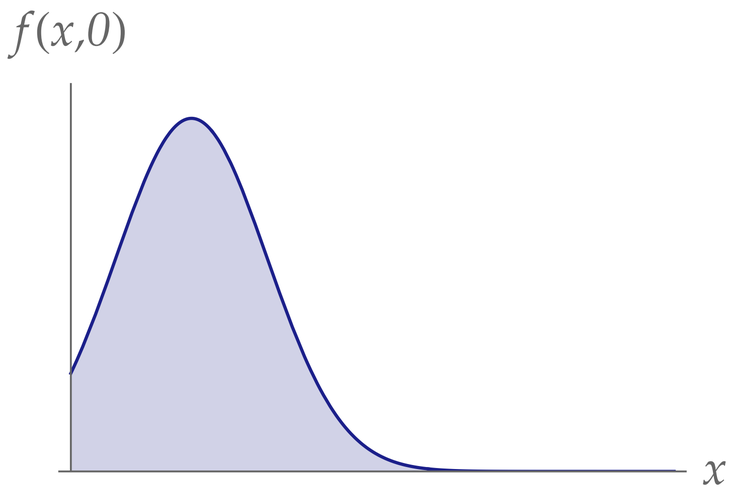
$$w_k(x) := \frac{N_k p_k(x)}{\sum_{l=1}^{s} N_l p_l(x)}$$

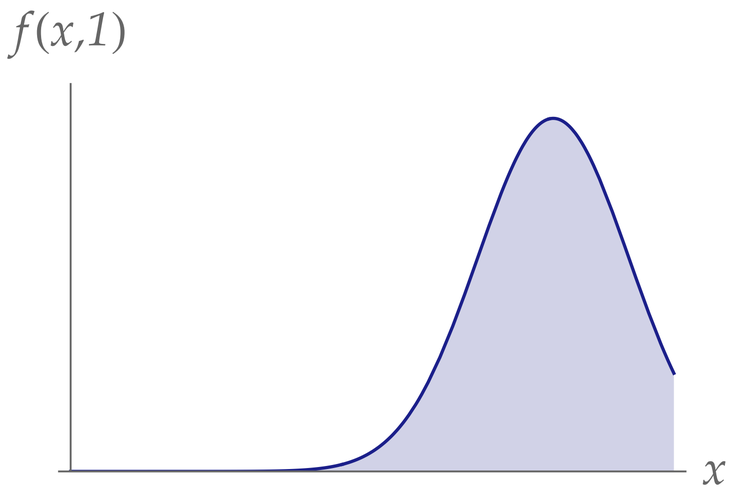
Theorem. (Veach 1997, §9.2.2) The variance of the balance heuristic $\widehat{I}_{BH}$ estimator cannot be significantly worse than any other MIS estimator:

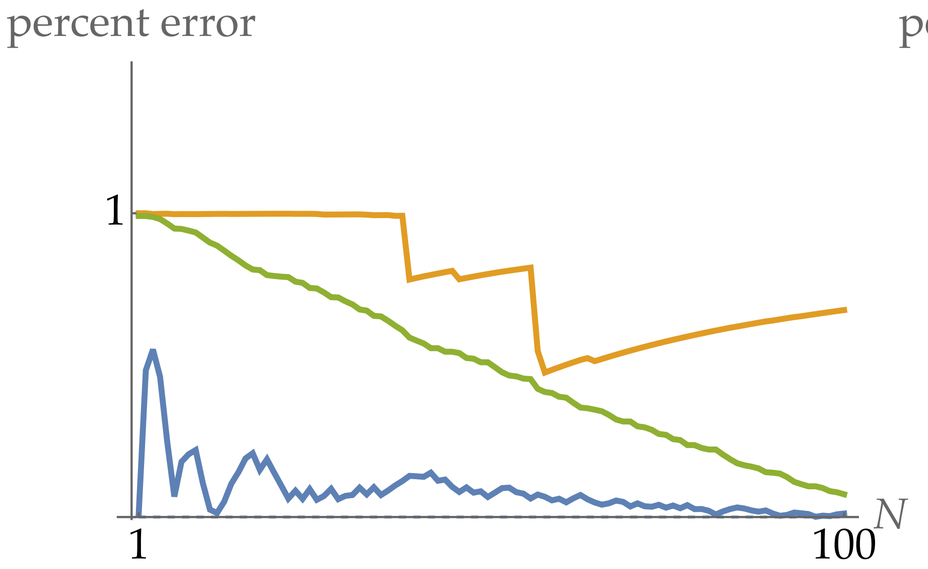
$$V[\widehat{I}_{BH}] - V[\widehat{I}_{MIS}] \leq \left(\frac{1}{\min_k n_k} - \frac{1}{\sum_k n_k} \right) I^2$$

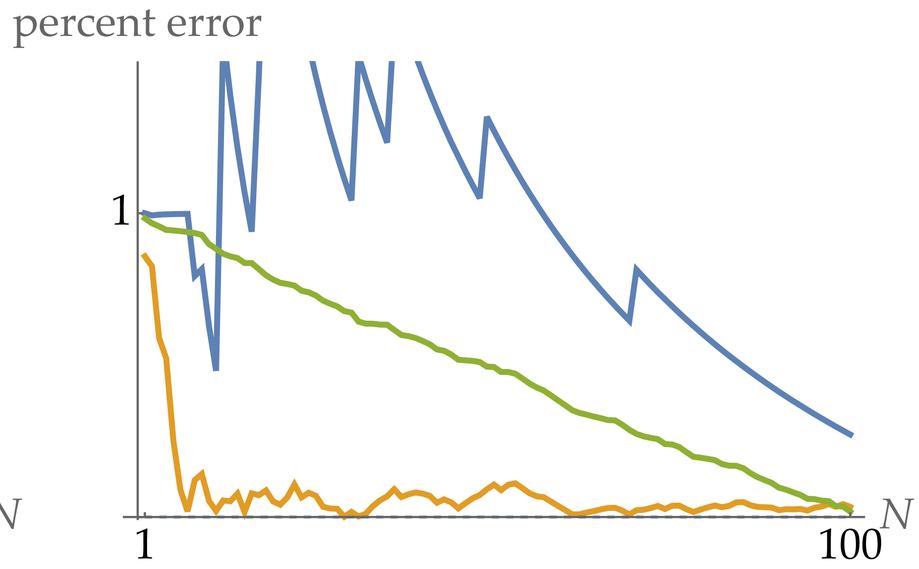
Multiple Importance Sampling—Example

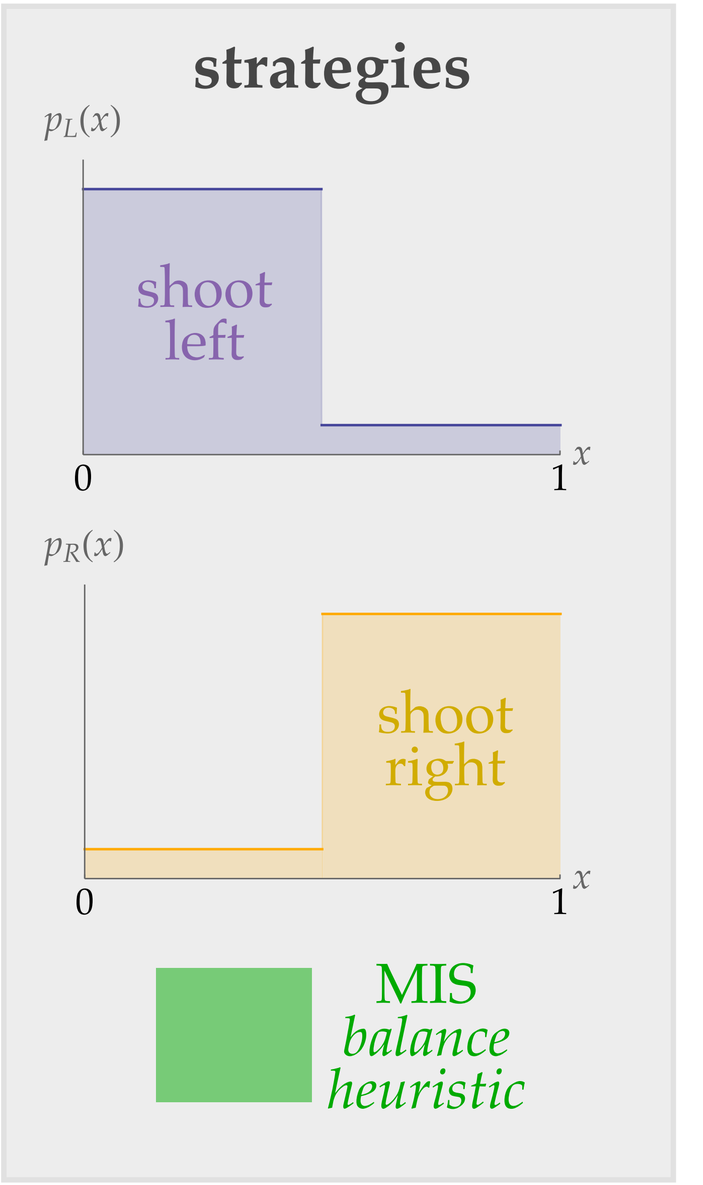
Automatically get best of both worlds:





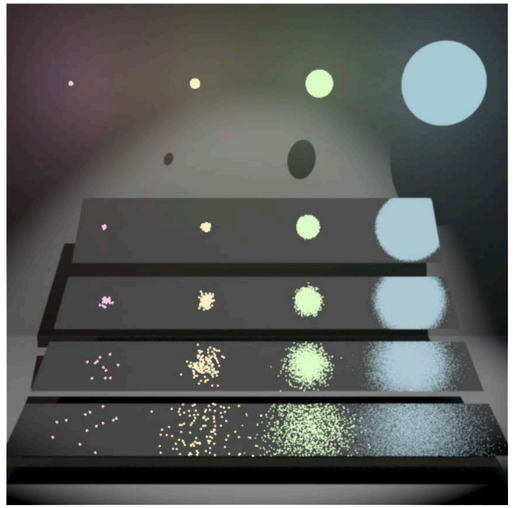




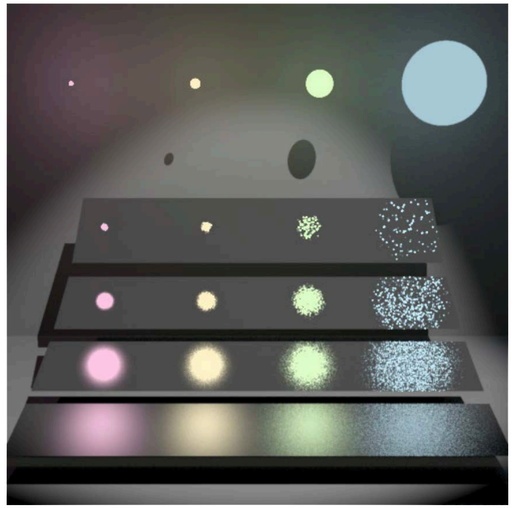


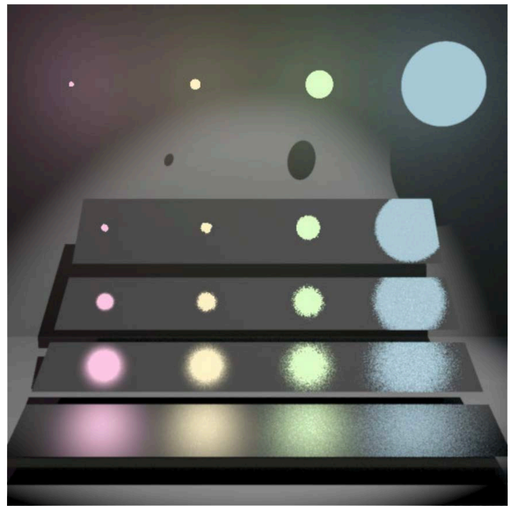
Multiple Importance Sampling—Rendering

sample lights



sample materials



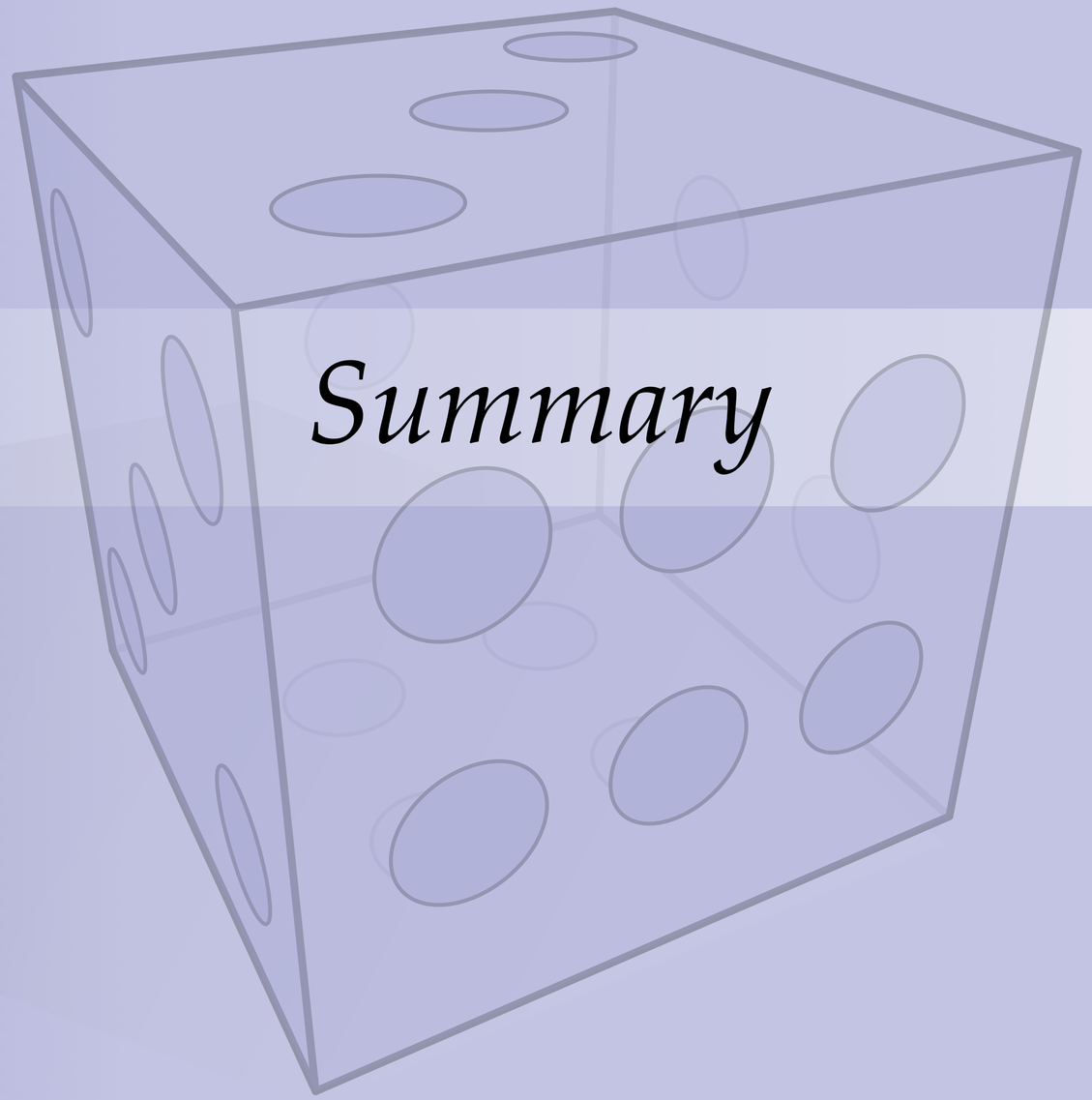


multiple importance sampling

"Optimally combining sampling techniques for Monte Carlo rendering"
Veach & Guibas (SIGGRAPH 1995)

Importance Sampling & Markov Chain Monte Carlo

- Earlier we said that "if only" we could draw samples proportional to $f(x)$, we could dramatically reduce the variance of our estimator
- There is a technique to do this, called Markov chain Monte Carlo (MCMC)!
  - will talk about it extensively later in the semester
- So… why don't we just use that?
  - No free lunch: MCMC doesn't give you $p(x) \sim f(x)$ *immediately*
  - Need to take a fairly large number of samples (mixing/startup bias)
  - In many integration problems, need to integrate many different integrands with just a few samples each
  - In general: no one silver bullet! But MCMC will be "the only hope" for some problems…



Summary

Summary

- Basic Monte Carlo is simple & dependable, but slow
- Lots of little things we can do to speed it up:
  - **always:** stratify samples, or replace random with low-discrepancy samples (QMC)
  - **sometimes:** control variates, antithetic sampling, importance sampling
- Death by 1000 paper cuts: each strategy might get us small factor improvement; product of many small factors can be huge improvement!
- Conceptually, all strategies aim to "flatten out" integrand, in different ways
- MCMC will give us a whole different take on (importance) sampling

Unified Picture of Acceleration Strategies?

All Monte Carlo acceleration techniques (even importance sampling) ultimately about finding transformations that yield lower-variance integrand:

control variates

subtract piece with known integral to get "flatter" integrand

antithetic sampling

sum transformed copies to get "flatter" function with same integral

stratified sampling

consider subdomains over which pieces of the integrand are "flatter"

quasi Monte Carlo

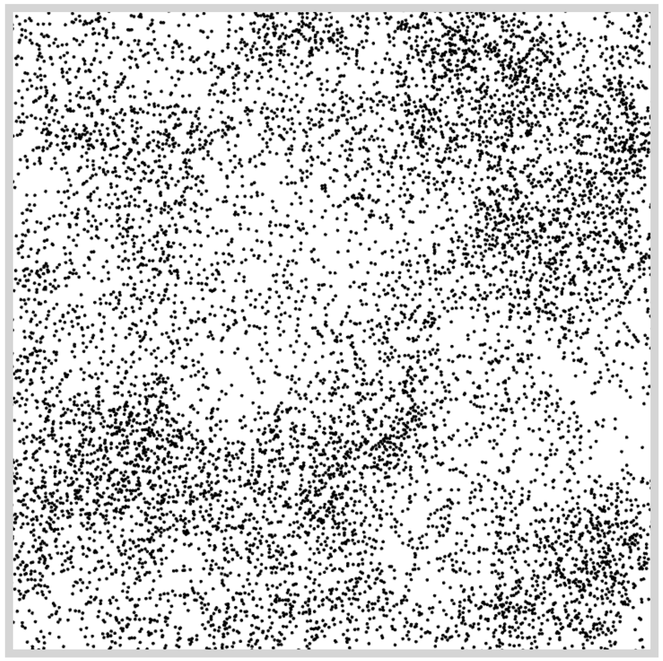
essentially a means to achieve stratification (hence "flatter" sub-integrands)

importance sampling

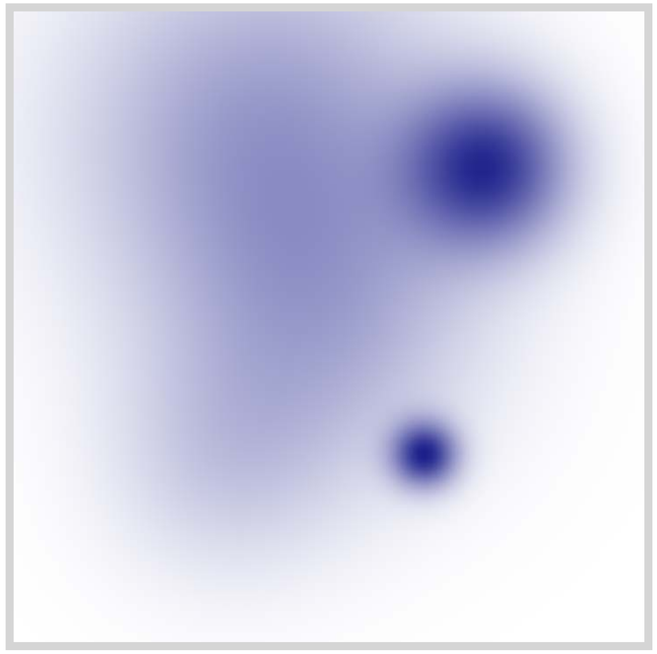
warp domain so that stretched-out integrand is flatter

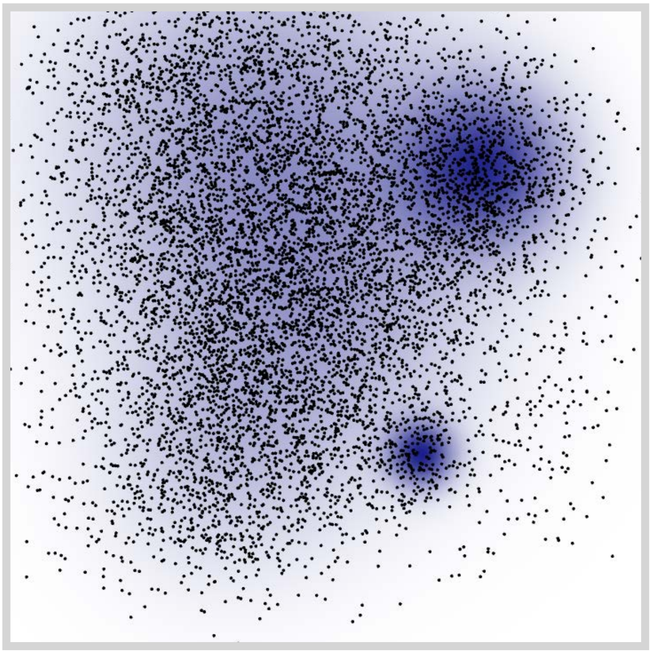
Food for future thought: is MCMC "ideal?"

Future lectures talk about how *Markov chain Monte Carlo* is "**only hope** in high dimensions." But does it provide ideal variance reduction? Still quite "clumpy"…



Metropolis-Hastings uniform





Metropolis-Hastings non uniform

Today's variance reduction strategies are **only hope** in "medium dimensions."

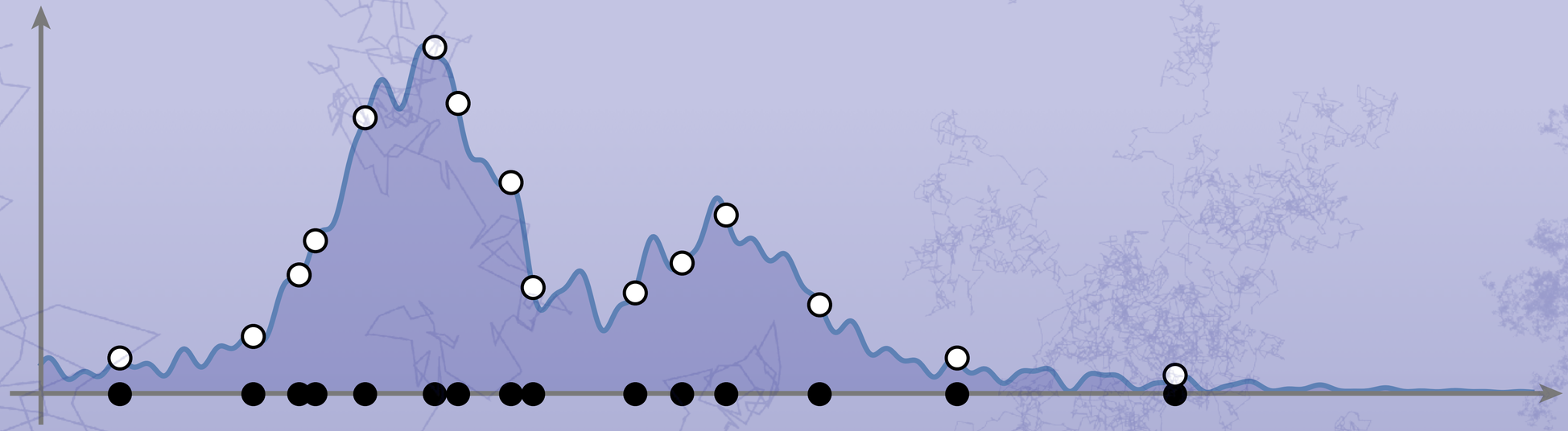
Variance Reduction in "Mixed" Dimensions

- Some of the variance reduction strategies we discussed work best in a reasonably low number of dimensions (e.g., stratification)
- For very high-dimensional functions, can still be quite beneficial to apply variance reduction across just some of the dimensions
  - apply antithetic sampling in coordinates with known symmetries
  - apply stratification in coordinates where "slices" of function are smooth
  - *etc.*

Next Up: Basic Sample Generation

- Many of our strategies for sampling from complicated distributions assume that we already know how to sample from simple distributions
  - e.g., uniform, or normal
- **Next time:** how do we actually draw from these distributions—or even generate random numbers?

Thanks!



Monte Carlo Methods and Applications

Carnegie Mellon University | 21-387 / 15-327 / 15-627 / 15-860 | Fall 2023