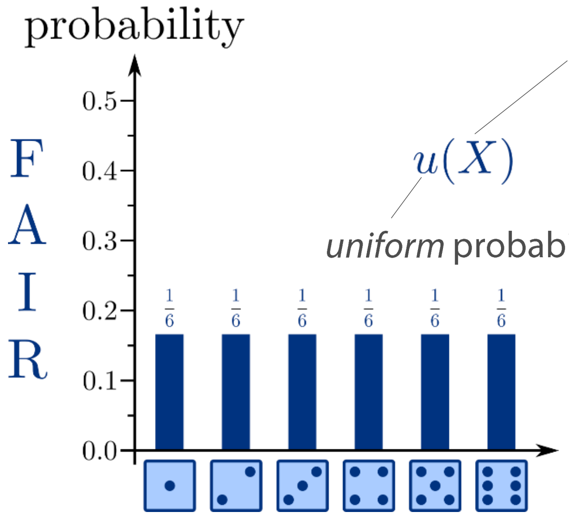
Importance Sampling

Importance Sampling: idea

[Example from *https://www.youtube.com/watch?v=pAbQHKr_Zqo*]

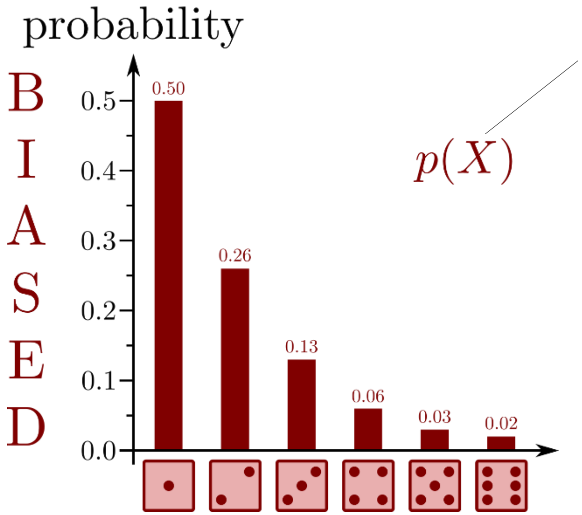
- *Rolling one dice: "fair" vs "biased"*



$u(X)$

random variable with values in $\{0, 1, 2, 3, 4, 5, 6\}$

uniform probability distribution



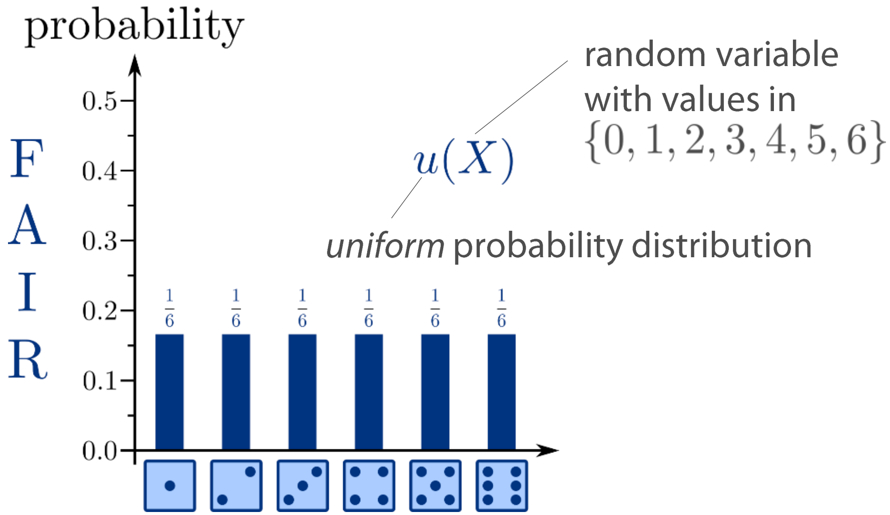
$p(X)$

random variable with values in $\{0, 1, 2, 3, 4, 5, 6\}$

Importance Sampling: idea

[Example from *https://www.youtube.com/watch?v=pAbQHKr_Zqo*]

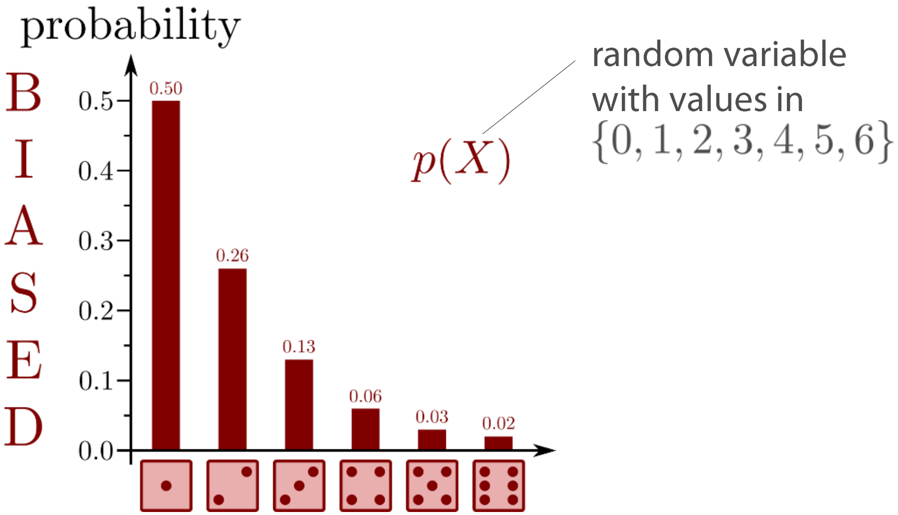
- *Rolling one dice: "fair" vs "biased"*



random variable with values in $\{0,1,2,3,4,5,6\}$

$u(X)$

uniform probability distribution



random variable with values in $\{0,1,2,3,4,5,6\}$

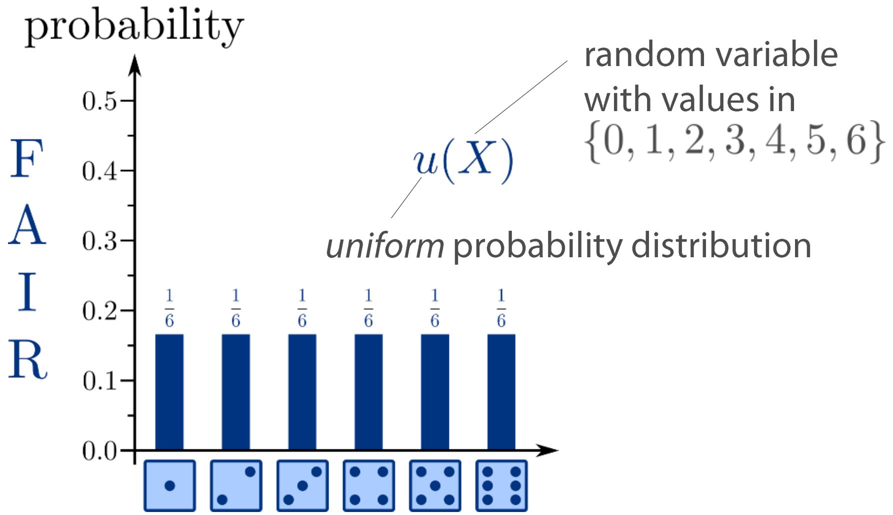
$p(X)$

What is the expected outcome, respectively?

Importance Sampling: idea

[Example from *https://www.youtube.com/watch?v=pAbQHKr_Zqo*]

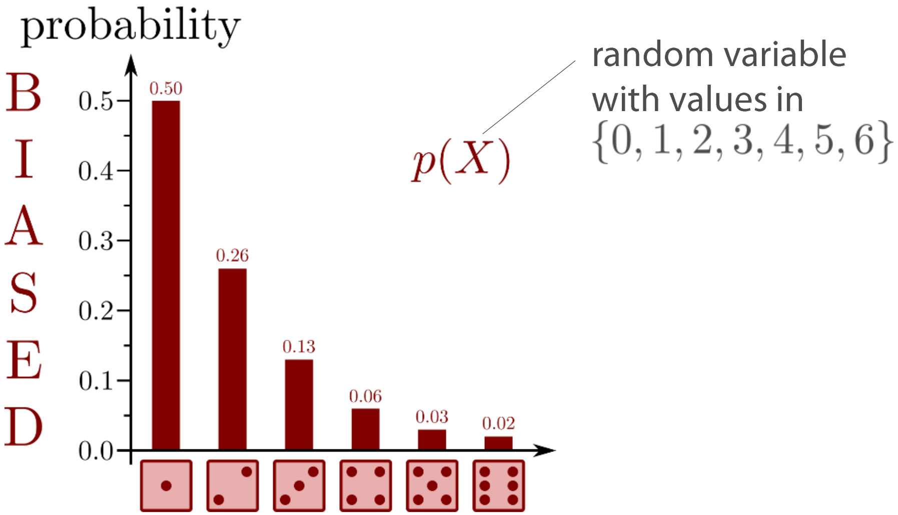
- *Rolling one dice: “fair” vs “biased”*



$u(X)$

random variable with values in $\{0, 1, 2, 3, 4, 5, 6\}$

uniform probability distribution



$p(X)$

random variable with values in $\{0, 1, 2, 3, 4, 5, 6\}$

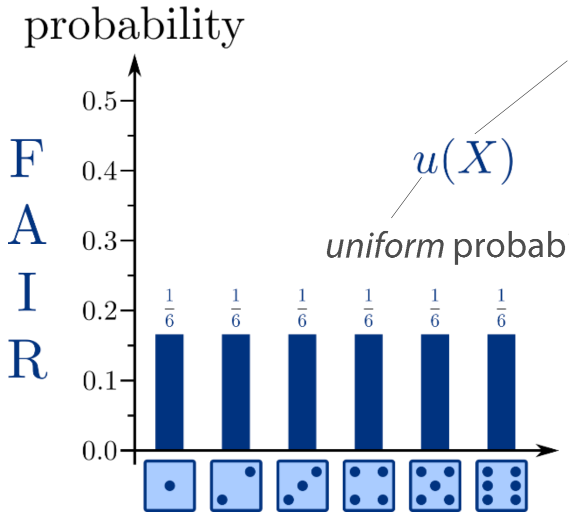
What is the expected outcome, respectively?

$$\mathbb{E}_u[X] := \sum_{x=1}^{6} x u(X = x) = \sum_{x=1}^{6} x \frac{1}{6} = \frac{21}{6} = 3.5$$

Importance Sampling: idea

- *Rolling one dice: “fair” vs “biased”*

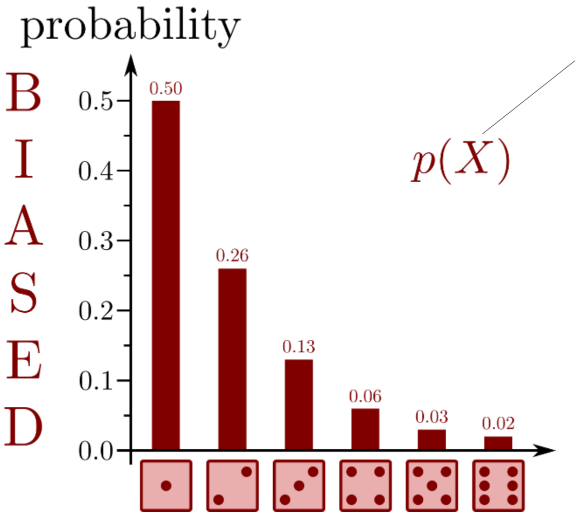
[Example from *https://www.youtube.com/watch?v=pAbQHKr_Zqo*]



random variable with values in $\{0, 1, 2, 3, 4, 5, 6\}$

uniform probability distribution

$$\mathbb{E}_u[X] = 3.5$$



random variable with values in $\{0, 1, 2, 3, 4, 5, 6\}$

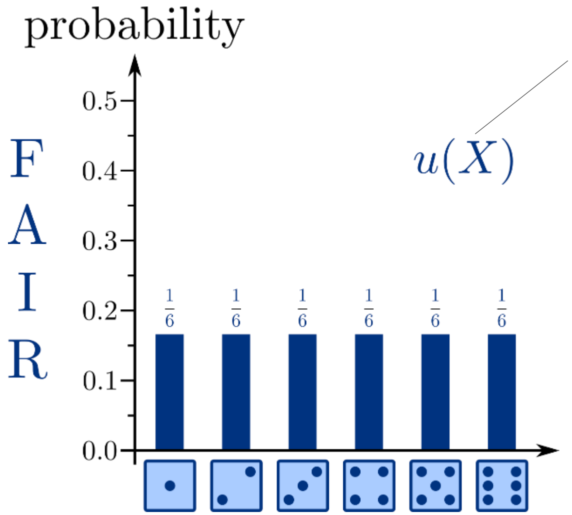
What is the expected outcome, respectively?

$$\mathbb{E}_p[X] := \sum_{x=1}^{6} x p(X = x) = 0.5 + 0.52 + 0.39 + 0.24 + 0.15 + 0.12 = 1.92$$

Importance Sampling: idea

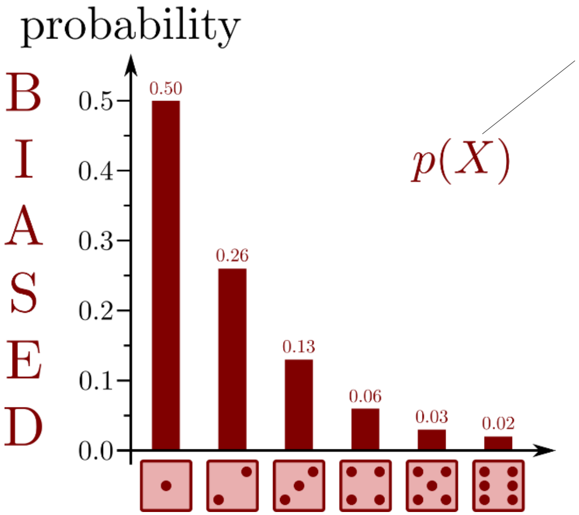
[Example from *https://www.youtube.com/watch?v=pAbQHKr_Zqo*]

- *Rolling one dice: "fair" vs "biased"*



random variable with values in $\{0, 1, 2, 3, 4, 5, 6\}$

$$\mathbb{E}_u[X] = 3.5$$



random variable with values in $\{0, 1, 2, 3, 4, 5, 6\}$

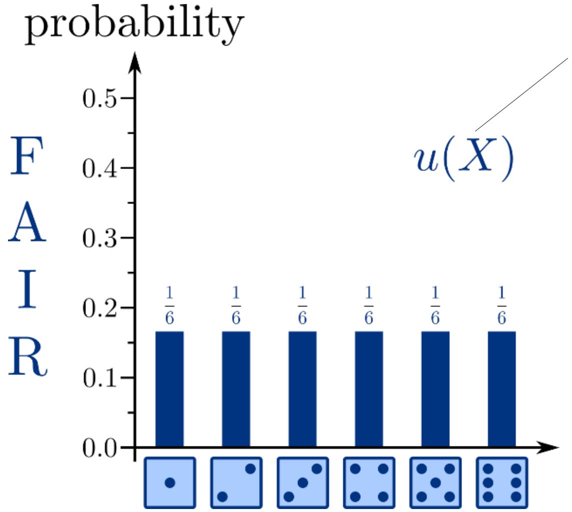
$$\mathbb{E}_p[X] = 1.92$$

- *What if* $p(X)$ *is* ***unknown**?*

Importance Sampling: idea

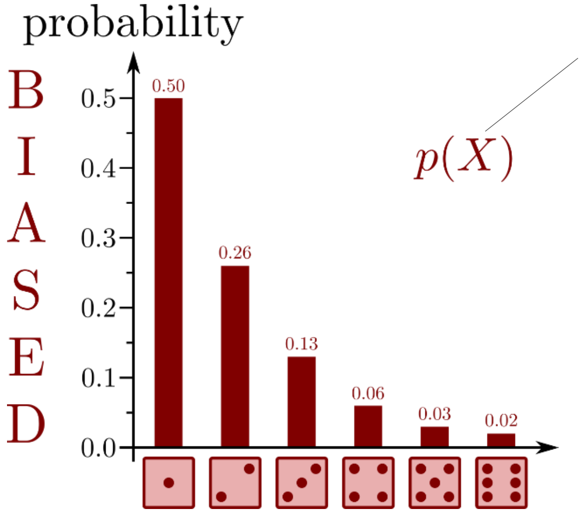
[Example from *https://www.youtube.com/watch?v=pAbQHKr_Zqo*]

- *Rolling one dice: "fair" vs "biased"*



random variable with values in $\{0, 1, 2, 3, 4, 5, 6\}$

$\mathbb{E}_u[X] = 3.5$



random variable with values in $\{0, 1, 2, 3, 4, 5, 6\}$

$\mathbb{E}_p[X] = 1.92$

- *What if $p(X)$ is <u>unknown</u>?*

compute the average outcome of N rolls of the biased dice

$$\mathbb{E}_p[X] \approx \frac{1}{N} \sum_{i=1}^{N} X_i^{(p)}$$

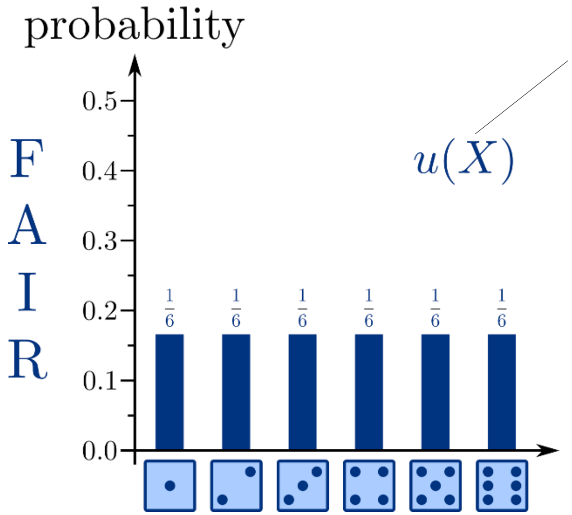
outcome of the i^{th} roll (with probability p)

Monte Carlo method

Importance Sampling: idea

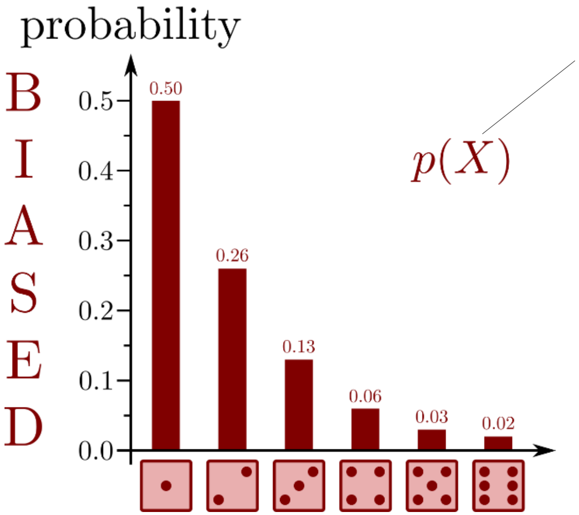
[Example from *https://www.youtube.com/watch?v=pAbQHKr_Zqo*]

- *Rolling one dice: “fair” vs “biased”*



random variable with values in $\{0, 1, 2, 3, 4, 5, 6\}$

$$\mathbb{E}_u[X] = 3.5$$



random variable with values in $\{0, 1, 2, 3, 4, 5, 6\}$

$$\mathbb{E}_p[X] = 1.92$$

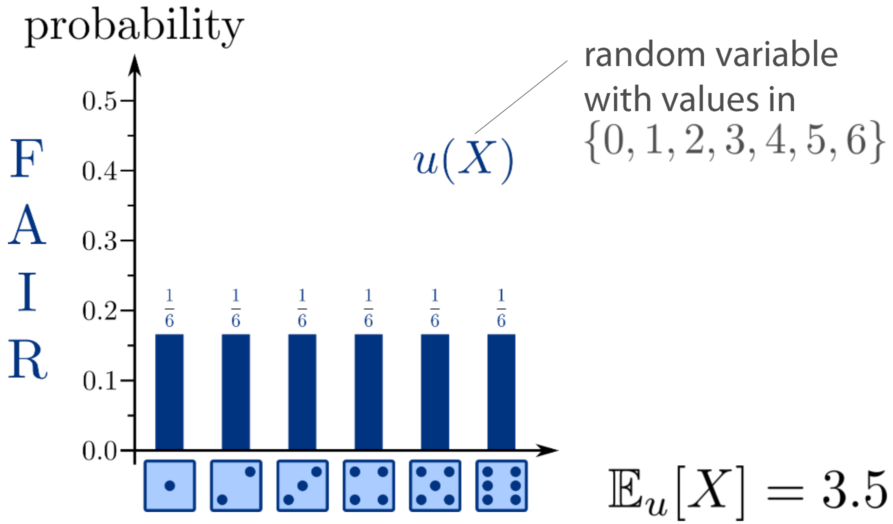
- *What if <u>sampling</u> from $p(X)$ is <u>impossible</u> ?*

$$\mathbb{E}_p[X] := \sum_{x=1}^{6} x \underbrace{p(X=x)}_{p(x)} = \sum_{x=1}^{6} x \frac{p(x)}{u(x)} u(x) = \mathbb{E}_u\left[X \frac{p(X)}{u(X)}\right]$$

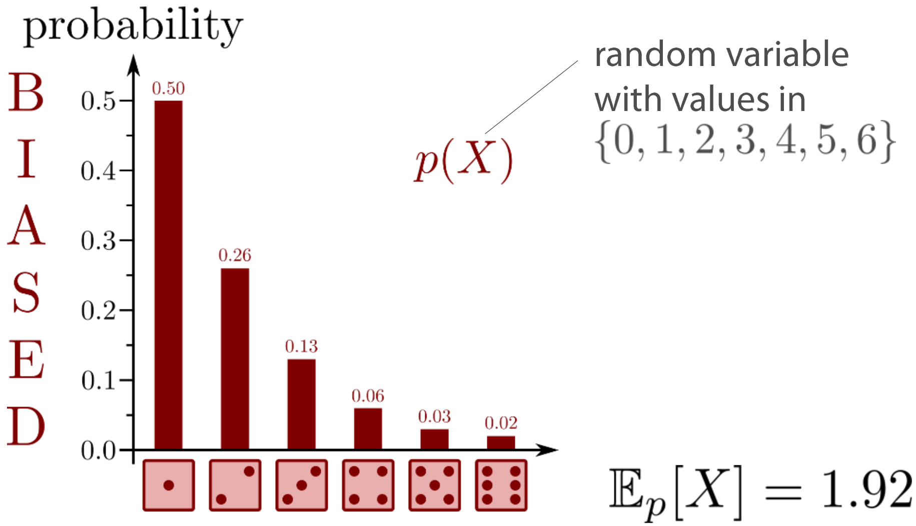
Importance Sampling: idea

[Example from *https://www.youtube.com/watch?v=pAbQHKr_Zqo*]

- *Rolling one dice: "fair" vs "biased"*



$\mathbb{E}_u[X] = 3.5$



$\mathbb{E}_p[X] = 1.92$

- *What if sampling from $p(X)$ is impossible ?*

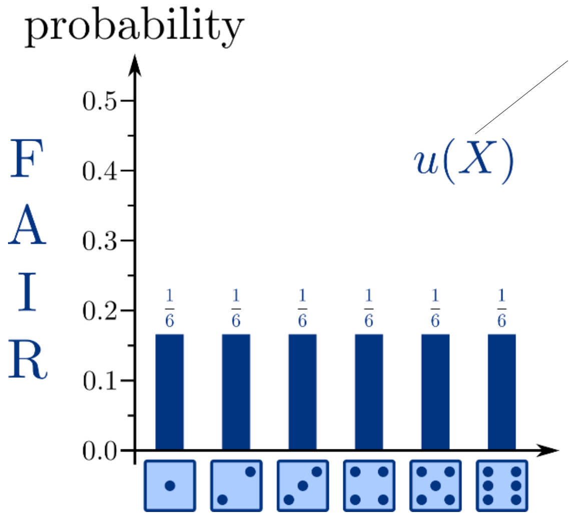
$$\mathbb{E}_p[X] := \sum_{x=1}^{6} x \underbrace{p(X=x)}_{p(x)} = \sum_{x=1}^{6} x \frac{p(x)}{u(x)} u(x) = \mathbb{E}_u\left[X \frac{p(X)}{u(X)}\right]$$

$$\approx \frac{1}{N} \sum_{i=1}^{N} X_i^{(u)} \frac{p(X_i^{(u)})}{u(X_i^{(u)})}$$

Importance Sampling: idea

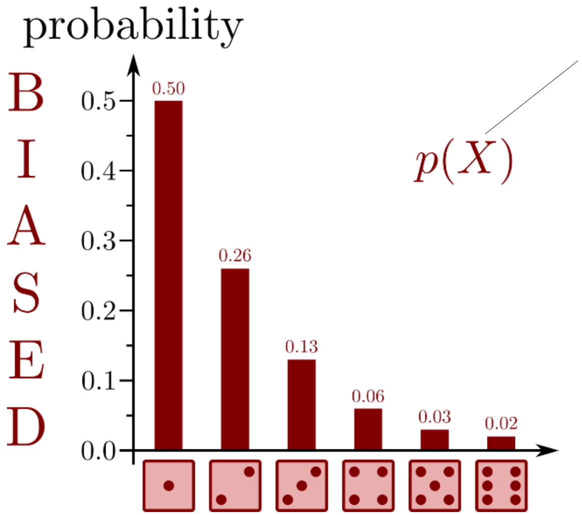
[Example from *https://www.youtube.com/watch?v=pAbQHKr_Zqo*]

- *Rolling one dice: "fair" vs "biased"*



random variable with values in $\{0, 1, 2, 3, 4, 5, 6\}$

$$\mathbb{E}_u[X] = 3.5$$



random variable with values in $\{0, 1, 2, 3, 4, 5, 6\}$

$$\mathbb{E}_p[X] = 1.92$$

- *What if sampling from $p(X)$ is impossible ?*

weight

$$\mathbb{E}_p[X] := \sum_{x=1}^{6} x \underbrace{p(X=x)}_{p(x)} = \sum_{x=1}^{6} x \frac{p(x)}{u(x)} u(x) = \mathbb{E}_u\left[X \boxed{\frac{p(X)}{u(X)}}\right]$$

Importance Sampling

target function

importance distribution

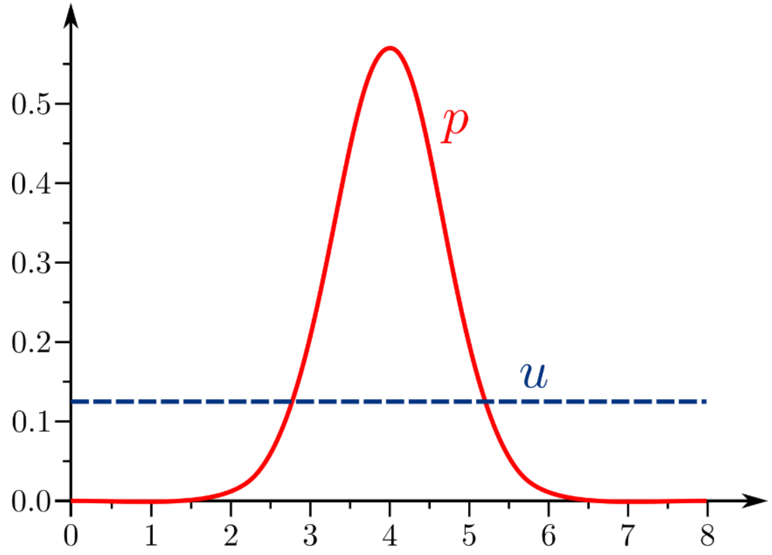
$$\approx \frac{1}{N} \sum_{i=1}^{N} X_i^{(u)} \frac{p(X_i^{(u)})}{u(X_i^{(u)})}$$

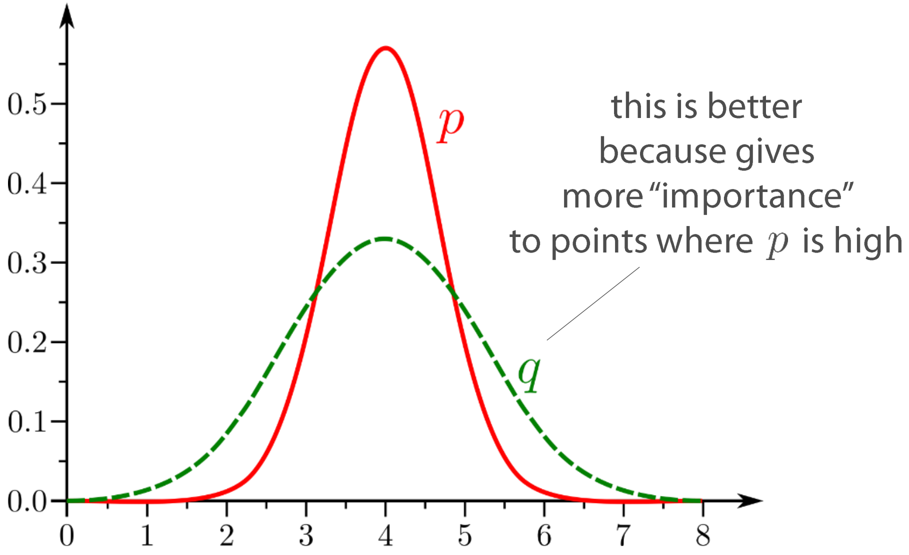
Importance Sampling: applications

- *Computing* ***expectations****:*

$$\mathbb{E}_p[X] = \mathbb{E}_u\left[X\frac{p(X)}{u(X)}\right] \approx \frac{1}{N}\sum_{i=1}^{N} X_i^{(u)}\frac{p(X_i^{(u)})}{u(X_i^{(u)})}$$

Any probability distribution q in place of u can be used, but some are better than others:





this is better
because gives
more "importance"
to points where p is high

Importance Sampling: applications

- *Computing* ***expectations****:*

$$\mathbb{E}_p[X] = \mathbb{E}_q\left[X\frac{p(X)}{q(X)}\right] \approx \frac{1}{N}\sum_{i=1}^{N} X_i^{(q)} \frac{p(X_i^{(q)})}{q(X_i^{(q)})}$$

The target function has not to be normalized:

$$p(x) := \frac{\overset{\text{known}}{f(x)}}{\underset{\text{unknown normalizing constant}}{F}}$$

$$\underset{\text{here } X \text{ is a continuous random variable}}{\mathbb{E}_p[X]} := \int x\frac{f(x)}{F}\mathrm{d}x = \frac{1}{F}\int x\frac{f(x)}{q(x)}q(x)\mathrm{d}x \approx \frac{1}{F}\frac{1}{N}\sum_{i=1}^{N} X_i^{(q)}\frac{f(X_i^{(q)})}{q(X_i^{(q)})}$$

Importance Sampling: applications

- *Computing **expectations**:*

$$\mathbb{E}_p[X] = \mathbb{E}_q\left[X\frac{p(X)}{q(X)}\right] \approx \frac{1}{N}\sum_{i=1}^{N} X_i^{(q)}\frac{p(X_i^{(q)})}{q(X_i^{(q)})}$$

The target function has not to be normalized:

known

$$p(x) := \frac{f(x)}{F}$$

unknown normalizing constant

$$\mathbb{E}_p[X] := \int x\frac{f(x)}{F}\mathrm{d}x = \frac{1}{F}\int x\frac{f(x)}{q(x)}q(x)\mathrm{d}x \approx \frac{1}{F}\frac{1}{N}\sum_{i=1}^{N} X_i^{(q)}\frac{f(X_i^{(q)})}{q(X_i^{(q)})}$$

with

$$F = \int f(x)\mathrm{d}x = \int \frac{f(x)}{q(x)}q(x)\mathrm{d}x \approx \frac{1}{N}\sum_{i=1}^{N}\frac{f(X_i^{(q)})}{q(X_i^{(q)})}$$

Importance Sampling: applications

- *Computing **expectations**:*

$$\mathbb{E}_p[X] = \mathbb{E}_q\left[X\frac{p(X)}{q(X)}\right] \approx \frac{1}{N}\sum_{i=1}^{N} X_i^{(q)}\frac{p(X_i^{(q)})}{q(X_i^{(q)})}$$

- *Computing **integrals**:*

$$F = \int f(x)\mathrm{d}x = \int \frac{f(x)}{q(x)}q(x)\mathrm{d}x \approx \frac{1}{N}\sum_{i=1}^{N}\frac{f(X_i^{(q)})}{q(X_i^{(q)})}$$

Importance Sampling: applications

- *Computing **expectations**:*

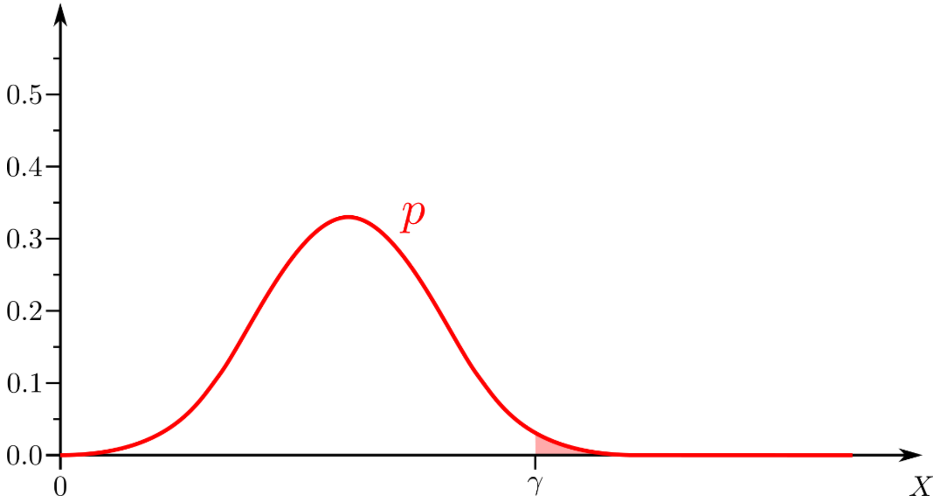
$$\mathbb{E}_p[X] = \mathbb{E}_q\left[X\frac{p(X)}{q(X)}\right] \approx \frac{1}{N}\sum_{i=1}^{N} X_i^{(q)}\frac{p(X_i^{(q)})}{q(X_i^{(q)})}$$

- *Computing **integrals**:*

$$F = \int f(x)\mathrm{d}x = \int \frac{f(x)}{q(x)}q(x)\mathrm{d}x \approx \frac{1}{N}\sum_{i=1}^{N} \frac{f(X_i^{(q)})}{q(X_i^{(q)})}$$

- ***Rare event** estimation:*

$$P(X \geq \gamma) =?$$



Importance Sampling: applications

- *Computing* ***expectations****:*

$$\mathbb{E}_p[X] = \mathbb{E}_q\left[X\frac{p(X)}{q(X)}\right] \approx \frac{1}{N}\sum_{i=1}^{N} X_i^{(q)}\frac{p(X_i^{(q)})}{q(X_i^{(q)})}$$

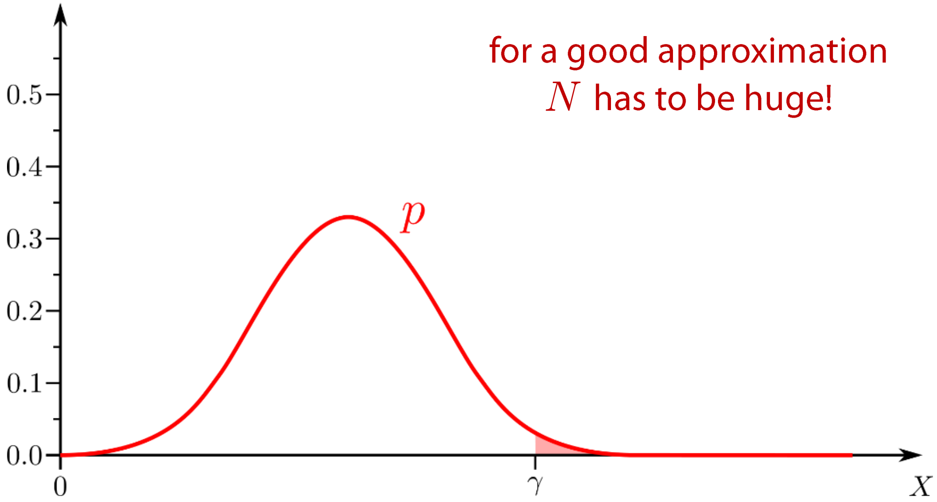
- *Computing* ***integrals****:*

$$F = \int f(x)\mathrm{d}x = \int \frac{f(x)}{q(x)}q(x)\mathrm{d}x \approx \frac{1}{N}\sum_{i=1}^{N}\frac{f(X_i^{(q)})}{q(X_i^{(q)})}$$

- ***Rare event*** *estimation:*

$$P(X \geq \gamma) = \int_{\gamma}^{\infty} xp(x)\mathrm{d}x$$

$$\approx \frac{1}{N}\#\left\{X_i^{(p)} \geq \gamma\right\}_{i=1}^{N}$$



for a good approximation
N has to be huge!

Importance Sampling: applications

- *Computing **expectations**:*

$$\mathbb{E}_p[X] = \mathbb{E}_q\left[X\frac{p(X)}{q(X)}\right] \approx \frac{1}{N}\sum_{i=1}^{N} X_i^{(q)}\frac{p(X_i^{(q)})}{q(X_i^{(q)})}$$

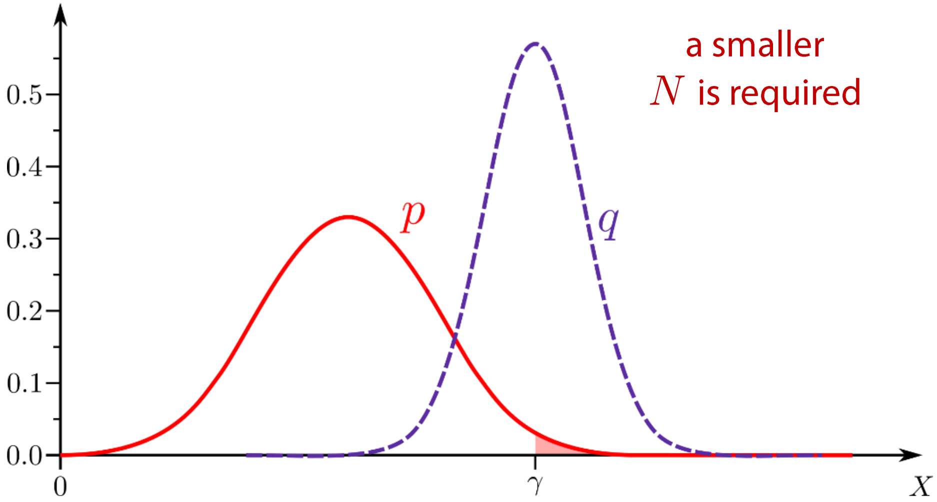
- *Computing **integrals**:*

$$F = \int f(x)\mathrm{d}x = \int \frac{f(x)}{q(x)}q(x)\mathrm{d}x \approx \frac{1}{N}\sum_{i=1}^{N}\frac{f(X_i^{(q)})}{q(X_i^{(q)})}$$

- ***Rare event** estimation:*

$$P(X \geq \gamma) = \int_{\gamma}^{\infty} x\frac{p(x)}{q(x)}q(x)\mathrm{d}x$$

$$\approx \frac{1}{N}\#\left\{X_i^{(q)}\frac{p(X_i^{(q)})}{q(X_i^{(q)})} \geq \gamma\right\}_{i=1}^{N}$$



a smaller N is required

Importance Sampling: applications

- Computing **expectations**:

$$\mathbb{E}_p[\boldsymbol{X}] = \mathbb{E}_q\left[\boldsymbol{X}\frac{p(\boldsymbol{X})}{q(\boldsymbol{X})}\right] \approx \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{X}_i^{(q)}\frac{p(\boldsymbol{X}_i^{(q)})}{q(\boldsymbol{X}_i^{(q)})}$$

- Computing **integrals**:

$$F = \int f(\boldsymbol{x})\mathrm{d}\boldsymbol{x} = \int \frac{f(\boldsymbol{x})}{q(\boldsymbol{x})}q(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \approx \frac{1}{N}\sum_{i=1}^{N}\frac{f(\boldsymbol{X}_i^{(q)})}{q(\boldsymbol{X}_i^{(q)})}$$

- ***Rare event*** *estimation:*

$$P(\boldsymbol{X} \geq \boldsymbol{\gamma}) = \int_{\boldsymbol{X}\geq\boldsymbol{\gamma}} \boldsymbol{x}\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}q(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \approx \frac{1}{N}\#\left\{\boldsymbol{X}_i^{(q)}\frac{p(\boldsymbol{X}_i^{(q)})}{q(\boldsymbol{X}_i^{(q)})} \geq \boldsymbol{\gamma}\right\}_{i=1}^{N}$$

Everything works for a <u>multivariate</u> random variable $\boldsymbol{X}$ too!

Importance Sampling: applications

- *Computing **expectations**:*

$$\mathbb{E}_p[\boldsymbol{X}] = \mathbb{E}_q\left[\boldsymbol{X}\frac{p(\boldsymbol{X})}{q(\boldsymbol{X})}\right] \approx \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{X}_i^{(q)} \boxed{\frac{p(\boldsymbol{X}_i^{(q)})}{q(\boldsymbol{X}_i^{(q)})}}$$

Weights transform $\boldsymbol{X}$ into a new random variable

$$\boldsymbol{Y} := \boldsymbol{X}\frac{p(\boldsymbol{X})}{q(\boldsymbol{X})}$$

- *Computing **integrals**:*

$$F = \int f(\boldsymbol{x})\mathrm{d}\boldsymbol{x} = \int \frac{f(\boldsymbol{x})}{q(\boldsymbol{x})}q(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \approx \frac{1}{N}\sum_{i=1}^{N} \frac{f(\boldsymbol{X}_i^{(q)})}{q(\boldsymbol{X}_i^{(q)})}$$

- ***Rare event** estimation:*

$$P(\boldsymbol{X} \geq \gamma) = \int_{\boldsymbol{X}\geq\gamma} \boldsymbol{x}\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}q(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \approx \frac{1}{N}\#\left\{\boldsymbol{X}_i^{(q)}\boxed{\frac{p(\boldsymbol{X}_i^{(q)})}{q(\boldsymbol{X}_i^{(q)})}} \geq \gamma\right\}_{i=1}^{N}$$

Everything works for a multivariate random variable $\boldsymbol{X}$ too!

Importance Sampling: applications

$$\boldsymbol{Y} := \frac{p(\boldsymbol{X})}{q(\boldsymbol{X})}\boldsymbol{X}$$

new variable

variable to sample from

- ***Sampling** from probability distribution* p *:*

1) sample $\boldsymbol{Y}_i$ from an easy probability distribution q

2) apply a suitable transformation $\mathcal{T}$ that acts as the multiplication by $\dfrac{q(\boldsymbol{X})}{p(\boldsymbol{X})}$

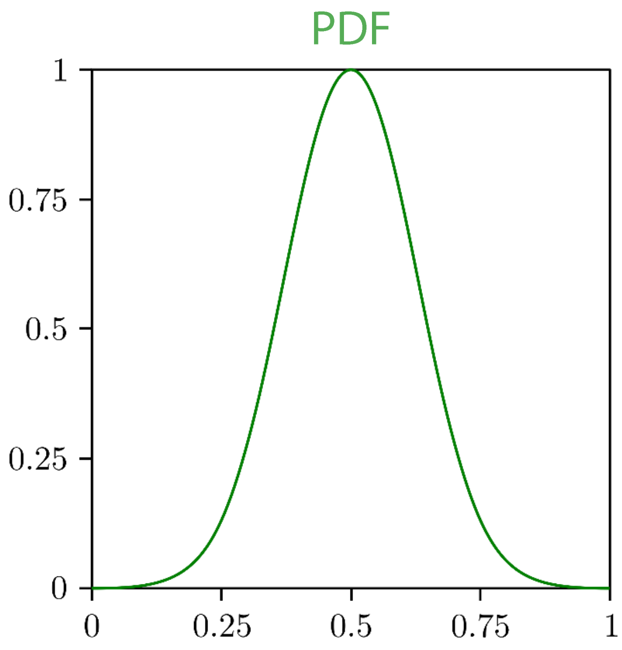
$$\boldsymbol{X}_i := \mathcal{T}(\boldsymbol{Y}_i)$$

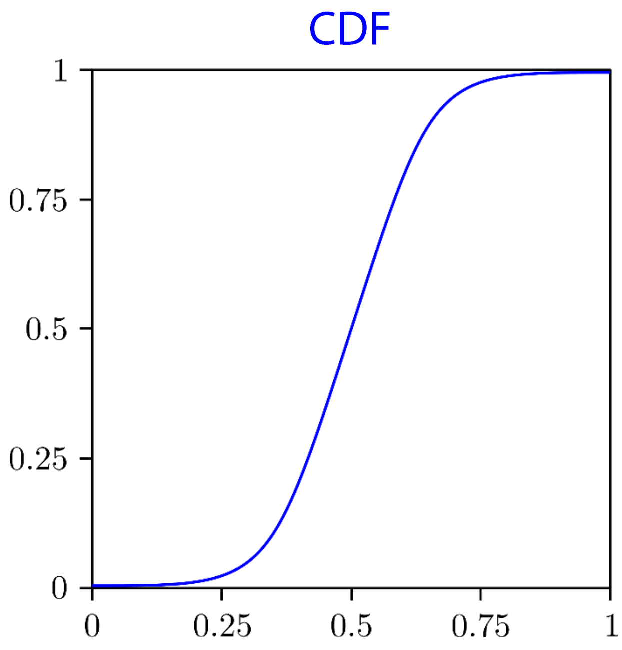
How to choose $\mathcal{T}$ *?*

An aside:
Sampling from inversion

Cumulative distribution function

- ***Cumulative distribution function (CDF)*** *of a probability distribution:*





$$f(x)$$

$$F(x) = \int_0^x f(t)\,\mathrm{d}t$$

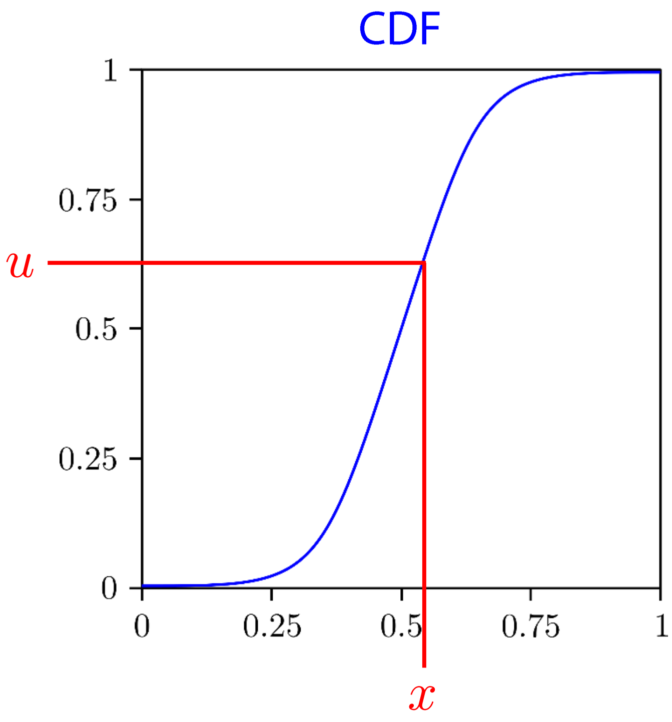
X is a random variable with values in $[0,1]$

probability of $X \leq x$

Inversion Sampling in one dimension

- ***Inversion Sampling*** *for X in $[0,1]$:*

  1) compute $F(x) = \int_0^x f(t)\,\mathrm{d}t$
  2) generate a random number u from the uniform distribution in $[0,1]$
  3) compute $x = F^{-1}(u)$



- x is distributed according to f, with CDF F :

$$P(X \leq x) = P(X \leq F^{-1}(u)) = P(F(X) \leq u) = u = F(x)$$

since $P(U \leq u) = u$ for the *uniformly distributed* random variable $U := F(X)$

Inversion Sampling one dimension at a time

- ***Inversion Sampling** for* $\boldsymbol{X} := [X_1, \ldots, X_d]$ *in* $[0,1]^d$*:*
  - computing $F(\boldsymbol{x}) = \int_0^{x_1} \cdots \int_0^{x_d} f(\boldsymbol{t})\, \mathrm{d}\boldsymbol{t}$ and F^{-1} is often *infeasible*
  - *probability factorization:*

$$P(x_1, \ldots, x_d) = P(x_1) P(x_2 \,|\, x_1) \cdots P(x_d \,|\, x_1, \ldots, x_{d-1})$$

with $P(x_1) = \int_{[0,1]^{d-1}} P(x_1, x_2, \ldots, x_d)\, \mathrm{d}x_2 \ldots \mathrm{d}x_d$

and

$$P(x_k \,|\, x_1, \ldots, x_{k-1}) = \int_{[0,1]^{d-k}} P(x_1, \ldots, x_{k-1}, x_k, x_{k+1}, \ldots, x_d)\, \mathrm{d}x_{k+1} \ldots \mathrm{d}x_d$$

Inversion Sampling one dimension at a time

- ***Sequential Sampling*** *for* $\boldsymbol{X} := [X_1, \ldots, X_d]$ *in* $[0, 1]^d$*:*

1) compute $F_1(x) := \int_{[0,1]^{d-1}} f(x, x_2, \ldots, x_d) \, \mathrm{d}x_2 \ldots \mathrm{d}x_d$

univariate (conditional) CDFs

2) generate a random number u_1 from the uniform distribution in $[0, 1]$

3) compute $\hat{x}_1 := F^{-1}(u_1)$

4) compute $F_2(x) := \int_{[0,1]^{d-2}} f(\hat{x}_1, x, x_3, \ldots, x_d) \, \mathrm{d}x_3 \ldots \mathrm{d}x_d$

5) generate a random number u_2 from the uniform distribution in $[0, 1]$

6) compute $\hat{x}_2 := F_2^{-1}(u_2)$

…

$\hat{\boldsymbol{x}} := [\hat{x}_1, \ldots, \hat{x}_d]$ is a sample randomly generated from the PDF $f(\boldsymbol{x})$

Inversion Sampling one dimension at a time

- ***Sequential Sampling*** *for* $\boldsymbol{X} := [X_1, \ldots, X_d]$ *in* $[0,1]^d$*:*
  - computing all the univariate CDFs and inverting them is *expensive*

- *Take-home ideal:*

  the transformation $\mathcal{T}$ required in the ***importance sampling*** should
  - approximate the (inverses of the) *CDFs*
  - be easily *computable*
  - be easily *invertible*

Advanced methods:
Bijectors for Importance Sampling

Importance Sampling: details

- *The transformation $\mathcal{T}$ is a* ***bijector*** *(or* ***normalizing flow****):*

  a 1-to-1 map $\boldsymbol{\mathcal{T}} : \boldsymbol{Y} \to \boldsymbol{X}$

  $$\boldsymbol{y} \mapsto \boldsymbol{x}$$

  sampled with *uniform* probability distribution u

For $\tilde{\boldsymbol{x}} := \boldsymbol{\mathcal{T}}(\tilde{\boldsymbol{y}})$

Dirac delda: 1 if $\boldsymbol{y} = \tilde{\boldsymbol{y}}$, 0 otherwise

$$p(\tilde{\boldsymbol{x}}) = \int p(\tilde{\boldsymbol{x}} \mid \boldsymbol{y})\; u(\boldsymbol{y})\; \mathrm{d}\boldsymbol{y} = \int \delta(\tilde{\boldsymbol{x}} - \boldsymbol{\mathcal{T}}(\boldsymbol{y}))\; u(\boldsymbol{y})\; \mathrm{d}\boldsymbol{y}$$

Importance Sampling: details

- *The transformation $\boldsymbol{\mathcal{T}}$ is a* ***bijector*** *(or* ***normalizing flow****):*

a 1-to-1 map $\boldsymbol{\mathcal{T}} : \boldsymbol{Y} \to \boldsymbol{X}$

$$\boldsymbol{y} \mapsto \boldsymbol{x}$$

sampled with *uniform* probability distribution u

For $\tilde{\boldsymbol{x}} := \boldsymbol{\mathcal{T}}(\tilde{\boldsymbol{y}})$

$$p(\tilde{\boldsymbol{x}}) = \int p(\tilde{\boldsymbol{x}} \mid \boldsymbol{y})\; u(\boldsymbol{y})\; \mathrm{d}\boldsymbol{y} = \int \delta(\tilde{\boldsymbol{x}} - \boldsymbol{\mathcal{T}}(\boldsymbol{y}))\; u(\boldsymbol{y})\; \mathrm{d}\boldsymbol{y}$$

$$= \underbrace{\left|\det\left(\left.\frac{\partial \boldsymbol{\mathcal{T}}(\boldsymbol{y})}{\partial \boldsymbol{y}}\right|_{\boldsymbol{y}=\tilde{\boldsymbol{y}}}\right)\right|}_{\text{Jacobian at } \tilde{\boldsymbol{y}}}{}^{-1} u(\tilde{\boldsymbol{y}})$$

Importance Sampling: details

- *The transformation $\boldsymbol{\mathcal{T}}$ is a* ***bijector*** *(or* ***normalizing flow****):*

  a 1-to-1 map $\boldsymbol{\mathcal{T}} : \boldsymbol{Y} \to \boldsymbol{X}$

  $$\boldsymbol{y} \mapsto \boldsymbol{x}$$

  sampled with *uniform* probability distribution u

  For $\tilde{\boldsymbol{x}} := \boldsymbol{\mathcal{T}}(\tilde{\boldsymbol{y}})$

$$p(\tilde{\boldsymbol{x}}) = \int p(\tilde{\boldsymbol{x}} \mid \boldsymbol{y})\; u(\boldsymbol{y})\; \mathrm{d}\boldsymbol{y} = \int \delta(\tilde{\boldsymbol{x}} - \boldsymbol{\mathcal{T}}(\boldsymbol{y}))\; u(\boldsymbol{y})\; \mathrm{d}\boldsymbol{y}$$

$$= \left| \boxed{\det} \left(\frac{\partial \boldsymbol{\mathcal{T}}(\boldsymbol{y})}{\partial \boldsymbol{y}} \bigg|_{\boldsymbol{y} = \boldsymbol{\mathcal{T}}^{-1}(\tilde{\boldsymbol{x}})} \right) \right|^{-1} u(\boxed{\boldsymbol{\mathcal{T}}^{-1}}(\tilde{\boldsymbol{x}}))$$

these should be easy and fast to compute

i-flow:
Normalizing Flow with Coupling Layers

i-flow

- ***i-flow*** [Gao et al. 2020]
  - the *bijector* is a chain of ***coupling layers***:

    $$\boldsymbol{\mathcal{T}}(\boldsymbol{y}) := \boldsymbol{c}_J(\ldots(\boldsymbol{c}_2(\boldsymbol{c}_1(\boldsymbol{y})))\ldots)$$

    so that

    $$\boldsymbol{\mathcal{T}}^{-1}(\boldsymbol{x}) = \boldsymbol{c}_1^{-1}(\boldsymbol{c}_2^{-1}(\ldots(\boldsymbol{c}_J^{-1}(\boldsymbol{x}))\ldots))$$

    $$\left|\det\left(\frac{\partial\boldsymbol{\mathcal{T}}(\boldsymbol{y})}{\partial\boldsymbol{y}}\right)\right|^{-1} = \prod_{j=1}^{J}\left|\det\left(\frac{\partial\boldsymbol{c}_j(\boldsymbol{y}^{(j)})}{\partial\boldsymbol{y}^{(j)})}\right)\right|^{-1}$$

    with $\boldsymbol{y}^{(1)} := \boldsymbol{y}$ and $\boldsymbol{y}^{(j+1)} := \boldsymbol{c}_j(\boldsymbol{y}^{(j)})$

  - each *coupling layer* $\boldsymbol{c}_j$ is a 1-to-1 map
    easy to invert and with an *easy to compute Jacobian*

i-flow: details

- *Single* ***Coupling Layer*** $\boldsymbol{c}$:

1) input: $\boldsymbol{y}$ with dimension $d := \dim(\boldsymbol{X})$

2) partition the set of dimensions $\{1, \ldots, d\}$
into two disjoint *non-empty* sets A and B

3) partition $\boldsymbol{y}$ accordingly:
(***masking***)

$$\boldsymbol{y}_A := [y_i]_{i \in A}$$

$$\boldsymbol{y}_B := [y_i]_{i \in B}$$

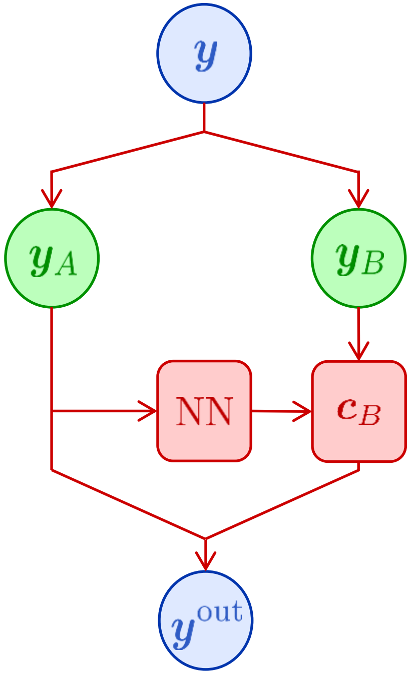
4) fix $\boldsymbol{y}_A$ and transform $\boldsymbol{y}_B$:

$$\boldsymbol{y}_A \overset{\boldsymbol{c}}{\mapsto} \boldsymbol{y}_A$$

$$\boldsymbol{y}_B \overset{\boldsymbol{c}}{\mapsto} \boldsymbol{c}_B(\boldsymbol{y}_B; \mathrm{NN}(\boldsymbol{y}_A))$$

1-to-1 map

Neural Network



5) output: $\boldsymbol{y}^{\mathrm{out}}$ with dimension d
obtained by rearranging components of $[\boldsymbol{c}(\boldsymbol{y}_A), \boldsymbol{c}(\boldsymbol{y}_B)]$ in the same order as $\boldsymbol{y}$

i-flow: details

- ***Chain** of J coupling layers:*
  - input: $\boldsymbol{y}$ with dimension $d := \dim(\boldsymbol{X})$
  - apply $\boldsymbol{c}_1$ with partition $[A_1, B_1]$ to $\boldsymbol{y}^{(1)} := \boldsymbol{y}$

  …

  - apply $\boldsymbol{c}_j$ with partition $[A_j, B_j]$ to $\boldsymbol{y}^{(j)} := \boldsymbol{c}_{j-1}(\boldsymbol{y}^{(j-1)})$

  …

  - return $\boldsymbol{x} := \boldsymbol{c}_J(\boldsymbol{y}^{(J)})$

i-flow: details

- ***Chain*** *of* J *coupling layers:*
  - input: $\boldsymbol{y}$ with dimension $d := \dim(\boldsymbol{X})$
  - apply $\boldsymbol{c}_1$ with partition $[A_1, B_1]$ to $\boldsymbol{y}^{(1)} := \boldsymbol{y}$

  …

  - apply $\boldsymbol{c}_j$ with partition $[A_j, B_j]$ to $\boldsymbol{y}^{(j)} := \boldsymbol{c}_{j-1}(\boldsymbol{y}^{(j-1)})$

  …

  - return $\boldsymbol{x} := \boldsymbol{c}_J(\boldsymbol{y}^{(J)})$

Make sure to *capture all possible correlations* between every dimension of $\boldsymbol{y}$:

- *each dimension* should by *transformed* (being in a B-set) *at least once*
- *all dimensions* should be *transformed equal number of times*
- the ***total number of coupling layers*** in the chain should be *at least*

$$J_{\min} := \begin{cases} d & \text{for } d \le 5 \\ 2\lceil \log_2 d \rceil & \text{for } d > 5 \end{cases}$$

i-flow: details

- *Coupling layer:*

$$\boldsymbol{y}_A \overset{\boldsymbol{c}}{\mapsto} \boldsymbol{y}_A$$

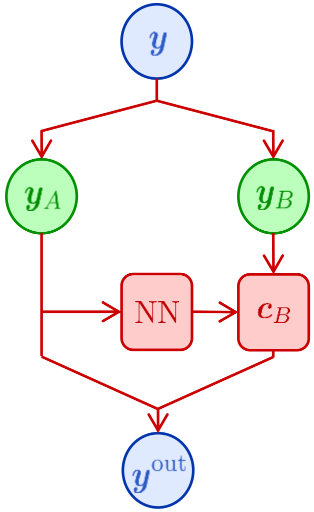
$$\boldsymbol{y}_B \overset{\boldsymbol{c}}{\mapsto} \boldsymbol{c}_B(\boldsymbol{y}_B; \underbrace{\mathrm{NN}(\boldsymbol{y}_A))}$$

- ***Inverse:***

these are *equal*

$$\boldsymbol{y}_A^{\mathrm{out}} \xrightarrow{\boldsymbol{c}^{-1}} \boldsymbol{y}_A^{\mathrm{out}}$$

$$\boldsymbol{y}_B^{\mathrm{out}} \xrightarrow{\boldsymbol{c}^{-1}} \boldsymbol{c}_B^{-1}(\boldsymbol{y}_B^{\mathrm{out}}; \overbrace{\mathrm{NN}(\boldsymbol{y}_A^{\mathrm{out}}))}$$

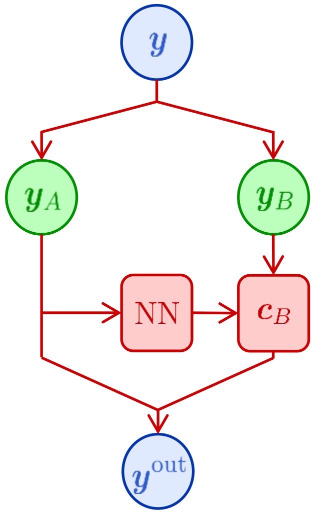


i-flow: details

- *Coupling layer:*

$$\boldsymbol{y}_A \overset{\boldsymbol{c}}{\mapsto} \boldsymbol{y}_A$$

$$\boldsymbol{y}_B \overset{\boldsymbol{c}}{\mapsto} \boldsymbol{c}_B(\boldsymbol{y}_B;\, \mathrm{NN}(\boldsymbol{y}_A))$$



- ***Inverse:***

$$\boldsymbol{y}_A^{\mathrm{out}} \xmapsto{\boldsymbol{c}^{-1}} \boldsymbol{y}_A^{\mathrm{out}}$$

$$\boldsymbol{y}_B^{\mathrm{out}} \xmapsto{\boldsymbol{c}^{-1}} \boldsymbol{c}_B^{-1}(\boldsymbol{y}_B^{\mathrm{out}};\, \mathrm{NN}(\boldsymbol{y}_A^{\mathrm{out}}))$$

- ***Jacobian:***

$$\left|\det\left(\frac{\partial \boldsymbol{c}(\boldsymbol{y})}{\partial \boldsymbol{y}}\right)\right|^{-1} \underset{\text{by reordering components (for simplicity)}}{=} \left|\det\begin{pmatrix} \overset{\text{identity}}{\boldsymbol{I}} & \boldsymbol{0} \\ \frac{\partial \boldsymbol{c}_B}{\partial \mathrm{NN}}\frac{\partial \mathrm{NN}}{\partial \boldsymbol{y}_A} & \frac{\partial \boldsymbol{c}_B}{\partial \boldsymbol{y}_B}\end{pmatrix}\right|^{-1} = \left|\det\left(\frac{\partial \boldsymbol{c}_B(\boldsymbol{y}_B;\, \mathrm{NN}(\boldsymbol{y}_A))}{\partial \boldsymbol{y}_B}\right)\right|^{-1}$$

i-flow: details

- *Coupling layer:*

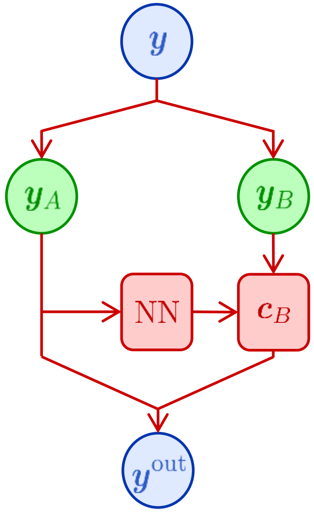
$$\boldsymbol{y}_A \overset{\boldsymbol{c}}{\mapsto} \boldsymbol{y}_A$$

$$\boldsymbol{y}_B \overset{\boldsymbol{c}}{\mapsto} \boldsymbol{c}_B(\boldsymbol{y}_B;\, \mathrm{NN}(\boldsymbol{y}_A))$$

- ***Inverse:***

$$\boldsymbol{y}_A^{\mathrm{out}} \xmapsto{\boldsymbol{c}^{-1}} \boldsymbol{y}_A^{\mathrm{out}}$$

$$\boldsymbol{y}_B^{\mathrm{out}} \xmapsto{\boldsymbol{c}^{-1}} \boldsymbol{c}_B^{-1}(\boldsymbol{y}_B^{\mathrm{out}};\, \mathrm{NN}(\boldsymbol{y}_A^{\mathrm{out}}))$$



no need to *invert*
nor compute the *Jacobian*
of the NN

- ***Jacobian:***

$$\left|\det\left(\frac{\partial \boldsymbol{c}(\boldsymbol{y})}{\partial \boldsymbol{y}}\right)\right|^{-1} = \left|\det\begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \frac{\partial \boldsymbol{c}_B}{\partial \mathrm{NN}}\frac{\partial \mathrm{NN}}{\partial \boldsymbol{y}_A} & \frac{\partial \boldsymbol{c}_B}{\partial \boldsymbol{y}_B} \end{pmatrix}\right|^{-1} = \left|\det\left(\frac{\partial \boldsymbol{c}_B(\boldsymbol{y}_B;\, \mathrm{NN}(\boldsymbol{y}_A))}{\partial \boldsymbol{y}_B}\right)\right|^{-1}$$