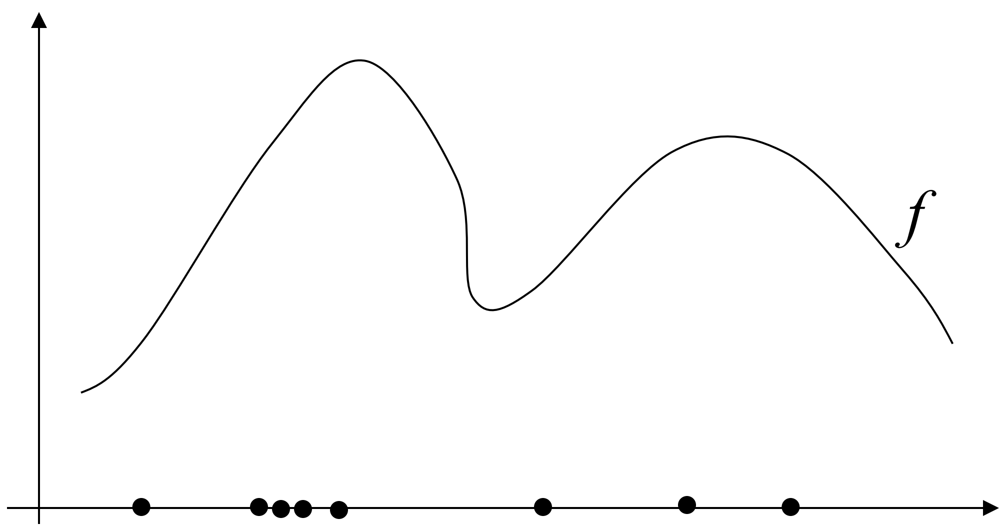
CMSC740
Advanced Computer Graphics

Fall 2025

Matthias Zwicker

Importance Sampling

- Goal: draw arbitrary number of samples from density that is proportional (as much as possible) to integrand f

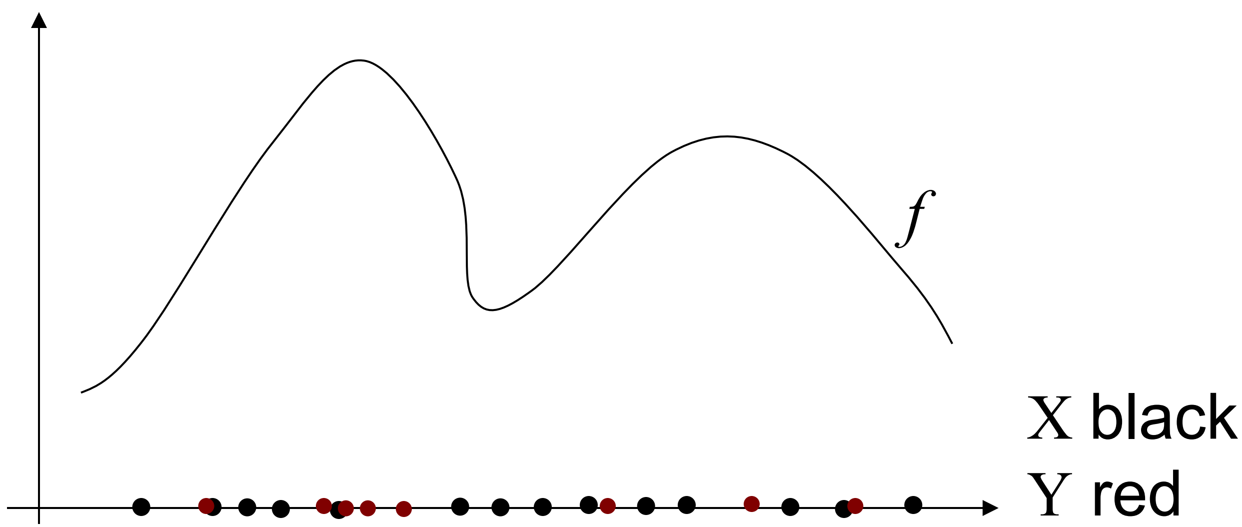


Samples proportional to f

Sampling-importance resampling

https://onlinelibrary.wiley.com/doi/full/10.1111/1467-9469.00360#b24%20#b25

- Goal: set of samples Y drawn from density proportional to integrand f
- Approach
  1. **Sampling:** Draw set of n samples $\mathrm{X}=\{\mathrm{x}_0,\ldots\mathrm{x}_{\mathrm{n}-1}\}$ from "simple" proposal distribution q (e.g. uniform)
  2. **Resampling:** Draw set of samples Y from X, with probability to include sample i in X given by $\Pr[\mathrm{i}] = w_i / \Sigma_j\, w_j$ and $w_i = f(\mathrm{x_i})/q(\mathrm{x_i})$
- As n goes to infinity, Y will have density proportional to f



Disadvantage: need to evaluate f for all samples in X (wasted work)

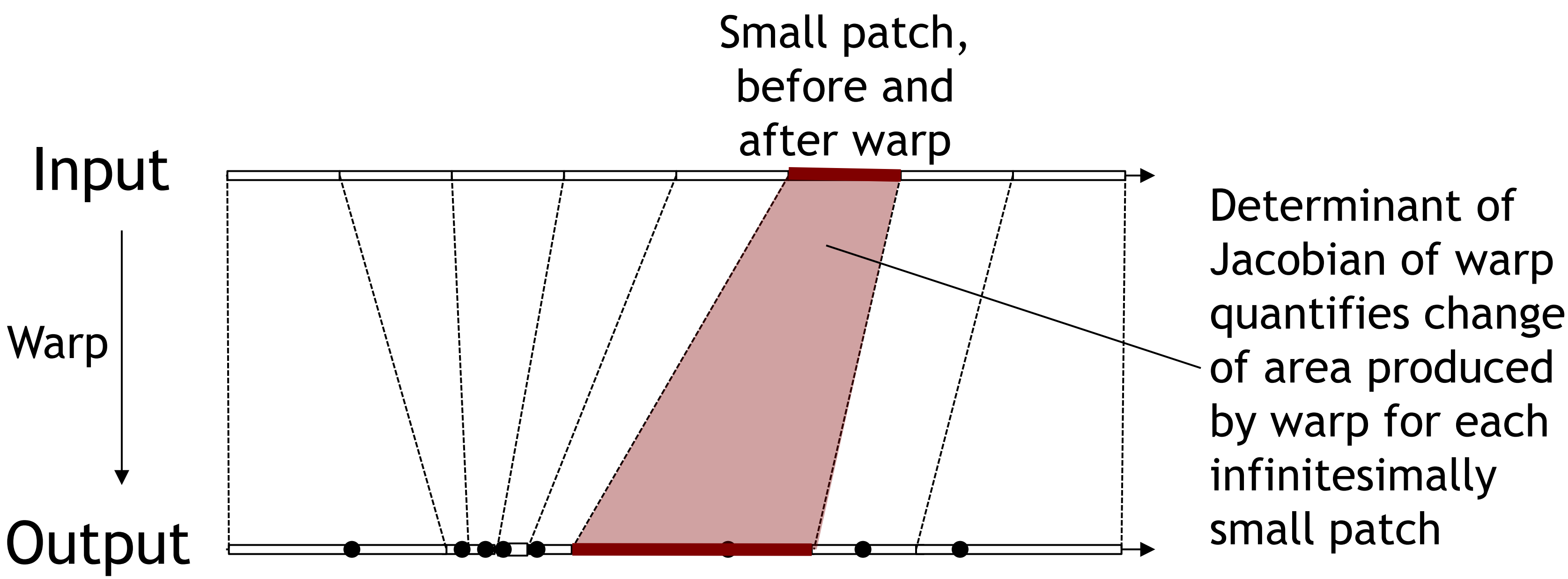
Related work in graphics: Generalized Resampled Importance Sampling: Foundations of ReSTIR

Neural Importance Sampling

- Goal: given small set of samples Y drawn from density proportional to integrand f
  1. Train a neural network that enables generating additional samples from the same distribution
  2. Ensure that the probability density for each sample is available (required for Monte Carlo integration)
  3. Generate as many samples and their densities as desired
- Note: 1. is essentially same problem statement as in "generative AI" (generate new samples of probability density given by a small set of samples); 2. is an additional requirement

Neural Importance Sampling

1. Draw small set of samples Y approximately proportional to f using sampling-importance resampling
2. Train warping function to approximate distribution of Y as a 1-to-1 warp of the domain of the integrand; determinant of Jacobian of warp inversely proportional to generated density



Warping function
(simple example: piecewise linear;
in practice: normalizing flow network)

Neural Importance Sampling

1. Draw small set of samples Y approximately proportional to f using sampling-importance resampling
2. Train warping function to approximate distribution of Y as a 1-to-1 warp of the domain of the integrand; determinant of Jacobian of warp inversely proportional to generated density
3. Generate new samples by feeding uniform random samples into warping function, which maps samples to desired distribution

- Warping function requirements
  - One-to-one mapping
  - Easily invertible
  - Easy to calculate determinant of Jacobian
  - Represent complicated warps
- Solution: normalizing flow networks
  https://en.wikipedia.org/wiki/Flow-based_generative_model
- Train neural network parameters using maximum likelihood estimation based on samples Y
  https://en.wikipedia.org/wiki/Maximum_likelihood_estimation