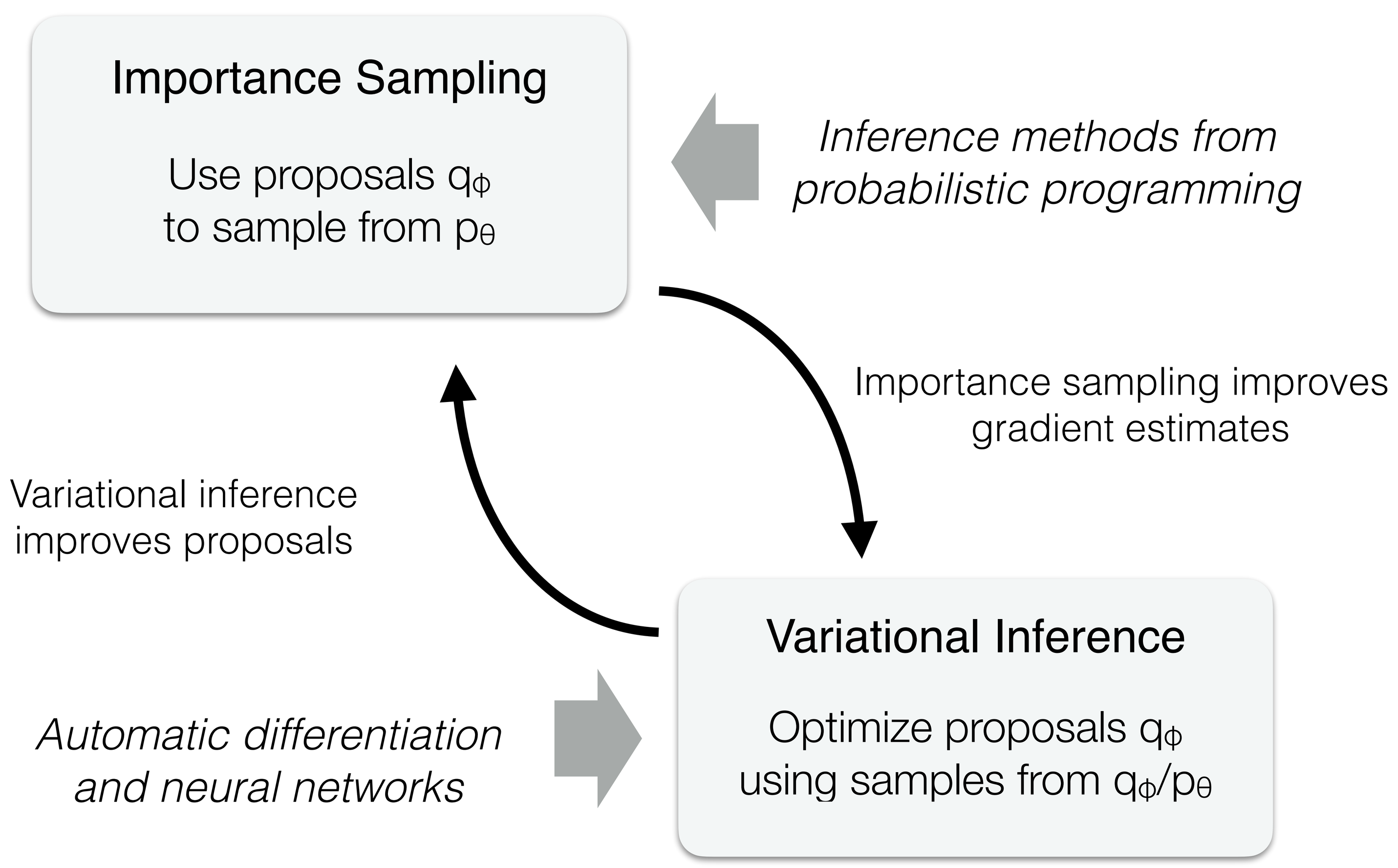
Composing Importance Samplers with Lenses

Eli Sennesh[1] Sam Stites[1] Jan-Willem van de Meent[1,2]

[1]Khoury College of Computer Sciences, Northeastern University, Massachusetts, USA [2]AMLab, University of Amsterdam, Amsterdam, The Netherlands

1 Importance Sampling and Variational Inference



2 Properly Weighted Program Evaluations

Denotational Semantics. *We assume the existence of denotational semantics, which for a program* f, *define a prior and unnormalized density*

$$[\![\mathtt{f}(c')]\!]_\gamma(\tau) = \gamma_f(\tau; c'), \qquad [\![\mathtt{f}(c')]\!]_p(\tau) = p_f(\tau; c').$$

Inference. *Given the unnormalized density, we want to approximate the corresponding normalized density*

$$\pi_f(\tau; c') = \frac{\gamma_f(\tau; c')}{Z_f(c')}, \qquad Z_f(c') = \int d\tau \, \gamma_f(\tau; c').$$

Conditioned evaluation: We can evaluate a program **f** by drawing randomness from a trace τ to obtain a new trace τ'.

$$\{\tau \vdash \mathtt{f}(a)\} \rightsquigarrow \{\tau', v \vdash b\}$$

Definition: we require that each primitive program be *strictly properly weighted* with respect to its unnormalized density $[\![\mathtt{f}(a)]\!]_\gamma$

$$\begin{aligned}\mathbb{E}_{\mathtt{f}(a)}\left[v h(\tau)\right] &= \int d\tau \, \gamma_{\mathtt{f}}(\tau; a) \, h(\tau) \\ &= Z_{\mathtt{f}}(a) \mathbb{E}_{\pi_{\mathtt{f}}(\tau \mid \omega_{\mathtt{f}}; a)}\left[h(\tau)\right].\end{aligned}$$

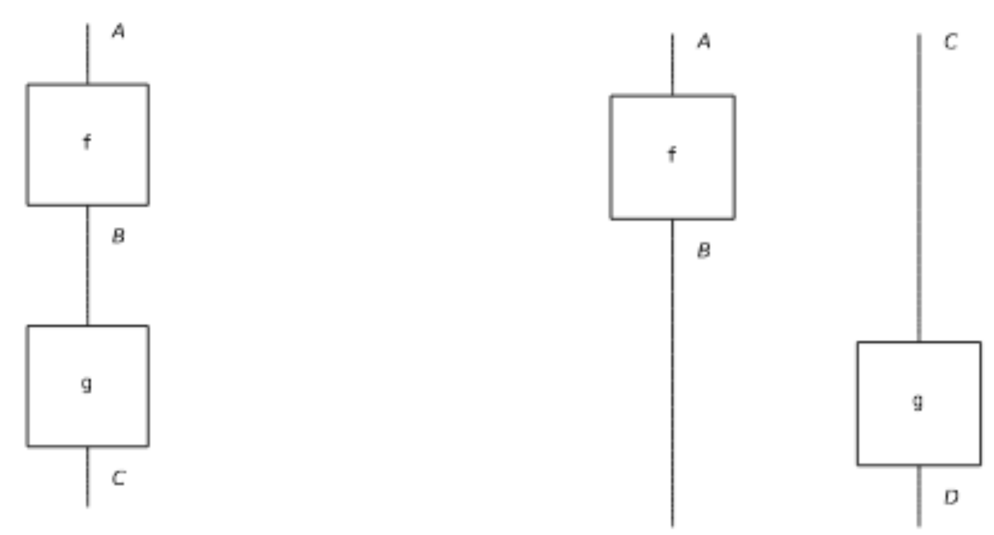
Sampler
(*can be a black box*)

Quantity of interest
(*return value of program*)

Constant of proportionality
(*marginal likelihood*)

Density of interest
(*program posterior*)

3 String diagrams as a first-order PPL



Operational Semantics.
We form a Cartesian monoidal ***category*** *of importance samplers; assign every string diagram an unnormalized density, and guarantee strict proper weighting.*

```
L = f ; primitive programs
L =/ id_A ; identity morphisms
L =/ L ≫ L ; sequential composition
L =/ L × L ; parallel composition
L =/ Δ_k A ; k-way duplication of outputs
```

Theorem 3 (The filtering procedure targets the filtering density) *For any string diagram f with inputs $a : A$ and any test statistic $h(\tau)$, the weighted expectation of samples targets the filtering distribution $[\![f]\!]_\gamma = \gamma_f(\tau; a)$. Precisely,*

$$\mathbb{E}_{f(a)}\left[w\, h(\tau)\right] = \int d\tau \, \gamma_f(\tau; a) \, h(\tau) = Z_f(a) \, \mathbb{E}_{\pi_f(\tau; a)}\left[h(\tau)\right].$$

Proof By induction. Trivial for the base case of the identity morphism. True by definition for the primitive lens morphism, thanks to the strict proper weighting of the primitive probabilistic program f. The inductive cases follow from the weighting rules given above, particularly the fact that we simply multiply weights across the diagram. ∎

4 Lenses compute forward and backward

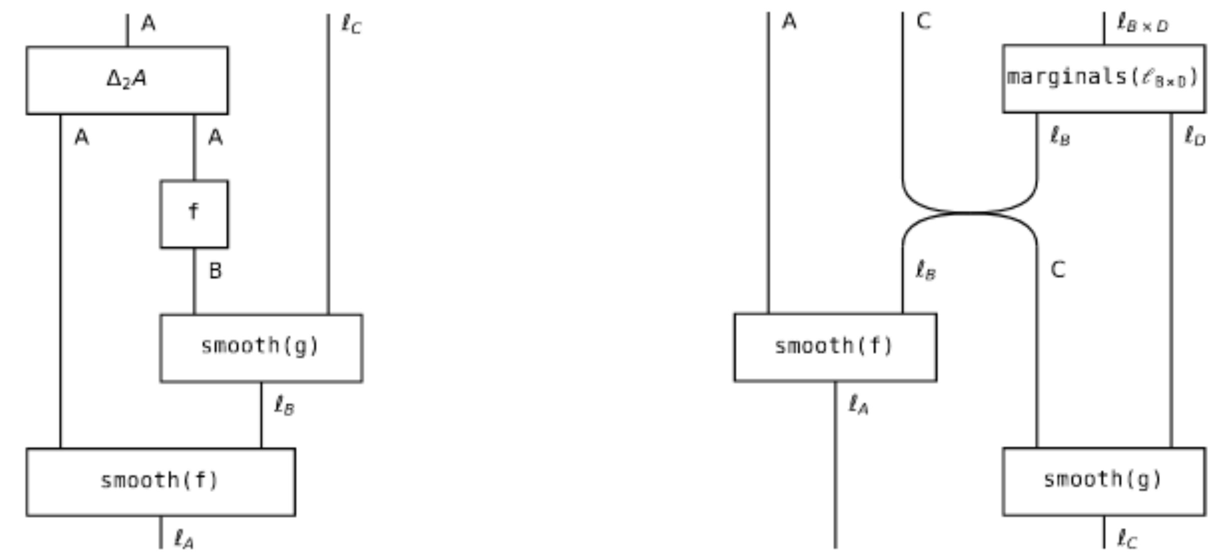
From any Cartesian monoidal category where we can define a primitive "backwards pass", we get a ***lens category*** *with backwards passes for all diagrams. Example: automatic differentiation*

$$\begin{pmatrix} f \\ g \end{pmatrix} : \begin{pmatrix} A \\ \ell_A \end{pmatrix} \rightsquigarrow \begin{pmatrix} B \\ \ell_B \end{pmatrix}$$

$$\mathtt{filter} : A \rightsquigarrow B$$

$$\mathtt{smooth} : A \times \ell_B \rightsquigarrow \ell_A$$

5 Lenses compose forward and backward



(a) Data flow in `smooth` computation on the sequential composition of two programs

(b) Data flow in `smooth` computation on the parallel composition of two programs

Theorem 6 (Bayes' rule for primitive (nested) importance samplers) *Given a link function ℓ_B that represents some $[\![\ell_B(b)]\!]_p = p_{\boldsymbol{g}}(\tau_{\boldsymbol{g}}; b)$, the* `smooth` *pass enforces strict proper weighting with respect to the posterior conditional density $[\![f]\!]_\gamma = \hat{\gamma}_{\boldsymbol{f}}(\tau_{\boldsymbol{f}} \mid \tau_{\boldsymbol{g}}; a)$. We write*

$$\mathbb{E}_{\tau_q \sim \boldsymbol{q}(a, \ell_B)}\left[\mathbb{E}_{\tau_p \sim \{\tau_1 \vdash \boldsymbol{f}(a)\}}\left[v h(\tau_p)\right]\right] = \int d\tau_p \, \hat{\gamma}_{\boldsymbol{f}}(\tau_p; a) \, h(\tau_p).$$

6 Example: deep generative mixture model

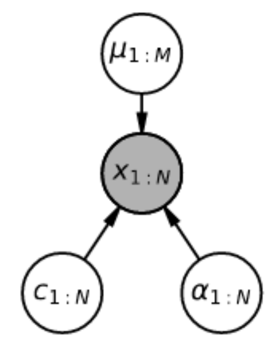


Figure 1: A probabilistic graphical model for a deep generative mixture model. A graphical model separates each random variable from each other.

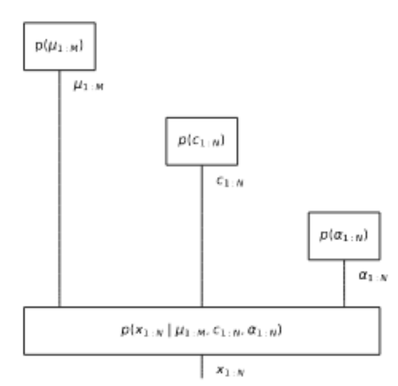


Figure 3: A deep generative mixture model, displayed as a string diagram. The string diagram shows each conditional distribution as a box, and the random variables connecting them as "strings" or "wires" connecting the boxes.

https://github.com/probtorch/combinators