Bayesian Generative Adversarial Networks

March 30, 2017

Abstract

Generative Adversarial Networks (GANs) belong to the class of generative models and have received a lot of attention recently. GANs are interesting because they learn to implicitly represent the likelihood function of training data. This is known to work exceptionally well for image data. A trained GAN is able to generate samples that look almost realistic. However, the restriction to an implicit representation means that a GA can only produce samples from the likelihood function. A GAN cannot estimate the likelihood directly. It remains intractable.

...

...

In this Meetup we will take a look at the situation in which a prior distribution is placed over the weights and biases of the GAN. This is called a Bayesian GAN in the literature. Bayesian GANs allow for the modelling of parameter uncertainty. Inferring th posterior distribution for Bayesian GANs poses new challenges due to the absence of a tractable likelihood. We will discuss a recent preprint article by Tran et al. (arXiv:1702.08896), in which an inference method is proposed that is based on variationa inference. The method finds a posterior by fitting the latent variables of a family of distributions. Since the variational family is implicit, the inferred posterior will be implicit as well.

Bayesian Generative Adversarial Networks

Outline

1. What is a model?
2. What is a GAN?
3. What is a Bayesian GAN?
4. What is a deep implicit model?
5. Inference for deep implicit models
6. A toy model in Edward

What Is A Model?

Abstract mathematical description of an aspect of the world.

Only valid in a certain regime, and typically under simplifying assumptions.

> *Essentially, all models are wrong, but some are useful.*

George E.P. Box

What Is A GAN?

Basic setup of a Generative Adversarial Network (GAN) is a game between two players:

- The **generator** creates samples from the same distribution as the training data.
- The **discriminator** classifies samples and determines whether they are real or fake.

Training:

- The **discriminator** learns using supervised learning techniques.
- The **generator** learns the distribution of the training data and is trained to fool the discriminator.

What Is A GAN?

The players are represented by differentiable functions:

- The **discriminator** is a function $d(\mathbf{x};\theta)$ of samples $\mathbf{x}$.
- The **generator** is a function $g(\varepsilon;\beta)$ taking random $\varepsilon \sim s(\cdot)$ and turning them into fake samples $\mathbf{x}$.

There are two cost functions:

- The **discriminator** must minimize $\mathcal{D}(\theta,\beta)$ while only varying θ.
- The **generator** must minimize $\mathcal{G}(\theta,\beta)$ while only varying β.

The solution is a Nash equilibrium (θ^*,β^*), a tuple where

- $\mathcal{D}$ has a local minimum with respect to θ.
- $\mathcal{G}$ has a local minimum with respect to β.

What Is A Bayesian GAN?

In a **GAN**, the generator function g with

$$\mathbf{x}_n = g(\varepsilon_n|\beta), \quad \varepsilon_n \sim s(\cdot), \quad n = 1, \ldots, N,$$

is an **implicit** representation of a **likelihood** $p(\mathbf{x}_n|\beta)$ for the observation $\mathbf{x}_n$:

$$\mathbf{x}_n \sim p(\cdot|\beta).$$

In a **Bayesian GAN**, we put a **prior** $p(\beta)$ on the parameters β:

$$\beta \sim p(\cdot).$$

This allows for explicit modelling of uncertainties in β.

What Is A Deep Implicit Model?

Bayesian GAN:

$$\mathbf{x}_n = g(\varepsilon_n|\beta), \quad \varepsilon_n \sim s(\cdot), \quad n = 1, \ldots, N.$$

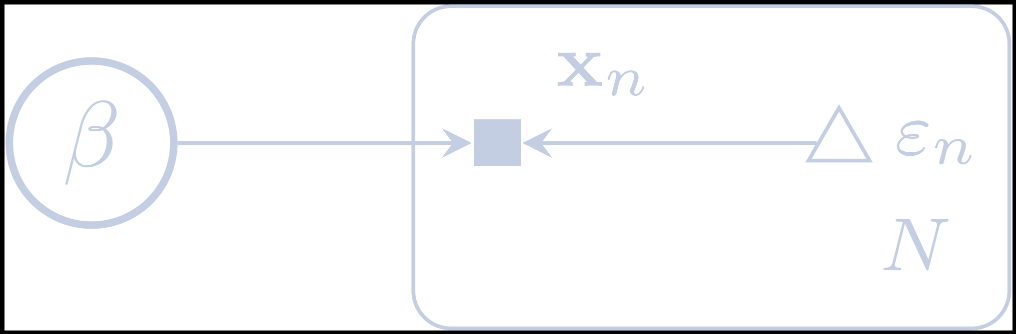
Deep **implicit** model with $L+1$ layers:

$$\begin{aligned}
\mathbf{x}_n &= g_0(\varepsilon_{n,0}|\mathbf{z}_{n,1}, \beta), \quad \varepsilon_{n,0} \sim s(\cdot), \\
\mathbf{z}_{n,1} &= g_1(\varepsilon_{n,1}|\mathbf{z}_{n,2}, \beta), \quad \varepsilon_{n,1} \sim s(\cdot), \\
&\vdots \\
\mathbf{z}_{n,L-1} &= g_{L-1}(\varepsilon_{n,L-1}|\mathbf{z}_{n,L}, \beta), \quad \varepsilon_{n,L-1} \sim s(\cdot), \\
\mathbf{z}_{n,L} &= g_L(\varepsilon_{n,L}|\beta), \quad \varepsilon_{n,L} \sim s(\cdot).
\end{aligned}$$

This defines the likelihoods $p(\mathbf{x}_n|\mathbf{z}_n, \beta)$ and $p(\mathbf{z}_n|\beta)$ **implicitly**.

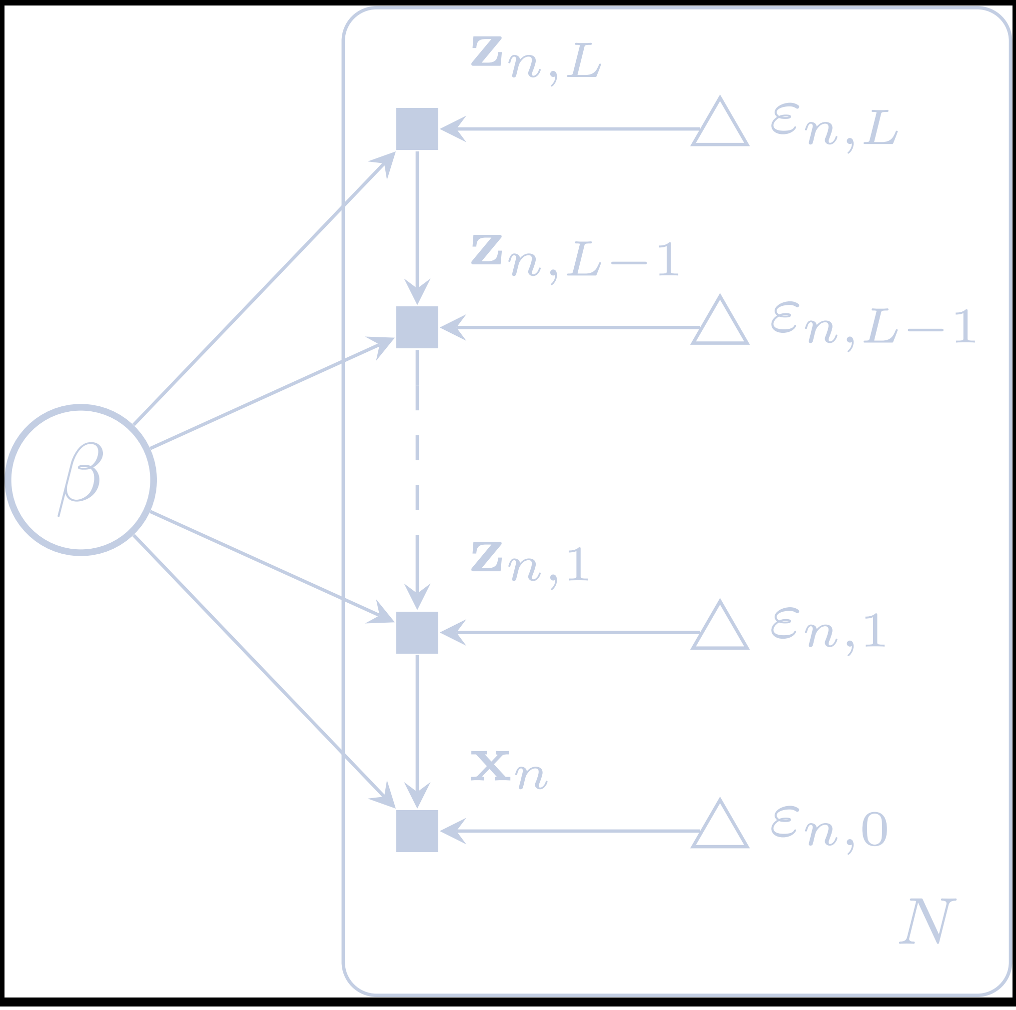
What Is A Deep Implicit Model?

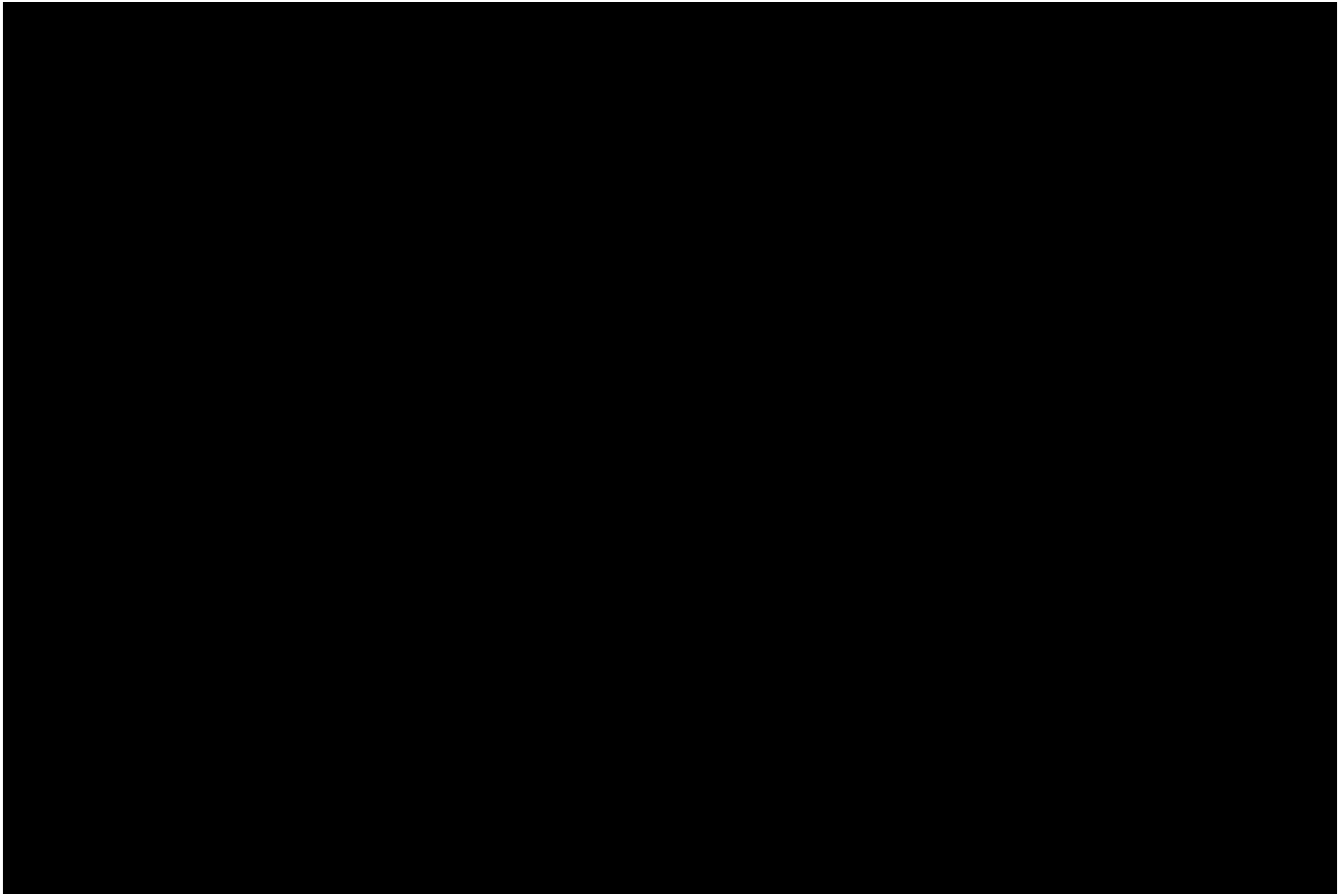
Bayesian GAN:



What Is A Deep Implicit Model?

Deep **implicit** model with $L+1$ layers:





What Is A Deep Implicit Model?

Together with the prior for β, a deep **implicit** model defines a hierarchical Bayesian model,

$$p(\mathbf{X}, \mathbf{Z}, \beta) = p(\beta) \prod_{n=1}^{N} p(\mathbf{x}_n | \mathbf{z}_n, \beta)\, p(\mathbf{z}_n | \beta).$$

Remainder of the talk: Inference or how do we get the **intractable** posterior $p(\mathbf{Z}, \beta | \mathbf{X})$?

Inference For Deep Implicit Models

Instead of **intractable** model posterior $p(\mathbf{Z}, \beta|\mathbf{X})$ use **variational approximation** $q(\beta, \mathbf{Z}|\mathbf{X}; \lambda, \phi)$ with

$$q(\beta, \mathbf{Z}|\mathbf{X}; \lambda, \phi) = q(\beta|\mathbf{X}; \lambda) \prod_{n=1}^{N} q(\mathbf{z}_n|\mathbf{x}_n, \beta; \phi).$$

Minimize the **Kullback-Leibler divergence** from q to p,

$$\mathrm{KL}\left(q(\beta, \mathbf{Z}|\mathbf{X}; \lambda, \phi) \| p(\mathbf{Z}, \beta|\mathbf{X})\right) \equiv \mathbb{E}_{q(\beta, \mathbf{Z}|\mathbf{X}; \lambda, \phi)}\left[\log \frac{q(\beta, \mathbf{Z}|\mathbf{X}; \lambda, \phi)}{p(\mathbf{Z}, \beta|\mathbf{X})}\right]$$

with respect to λ, ϕ since it measures closeness between p and q.

The Evidence Lower Bound

Minimization of $\mathrm{KL}\left(q(\beta, \mathbf{Z}|\mathbf{X}; \lambda, \phi) \| p(\mathbf{Z}, \beta|\mathbf{X})\right)$ with respect to λ and ϕ is not possible, though, because both densities are **intractable**.

Maximize the Evidence Lower BOund (ELBO) instead,

$$\begin{aligned}\mathcal{L}(\lambda, \phi) &\triangleq \log p(\mathbf{X}) - \mathrm{KL}\left(q(\beta, \mathbf{Z}|\mathbf{X}; \lambda, \phi) \| p(\mathbf{Z}, \beta|\mathbf{X})\right) \\ &= \mathbb{E}_{q(\beta, \mathbf{Z}|\mathbf{X}; \lambda, \phi)}\left[\log p(\mathbf{X}, \mathbf{Z}, \beta) - \log q(\beta, \mathbf{Z}|\mathbf{X}; \lambda, \phi)\right].\end{aligned}$$

The Evidence Lower Bound

$$\mathcal{L}(\lambda, \phi) = \mathbb{E}_{q(\beta, \mathbf{Z}|\mathbf{X}; \lambda, \phi)}[\log p(\mathbf{X}, \mathbf{Z}, \beta) - \log q(\beta, \mathbf{Z}|\mathbf{X}; \lambda, \phi)]$$

Substitute:

$$p(\mathbf{X}, \mathbf{Z}, \beta) = p(\beta) \prod_{n=1}^{N} p(\mathbf{x}_n|\mathbf{z}_n, \beta)\, p(\mathbf{z}_n|\beta),$$
$$q(\beta, \mathbf{Z}|\mathbf{X}; \lambda, \phi) = q(\beta|\mathbf{X}; \lambda) \prod_{n=1}^{N} q(\mathbf{z}_n|\mathbf{x}_n, \beta; \phi).$$

Get (dependence on λ, ϕ omitted):

$$\mathcal{L} = \mathbb{E}_{q(\beta, \mathbf{Z}|\mathbf{X})}[\log p(\beta) - \log q(\beta|\mathbf{X})] + \sum_{n=1}^{N} \mathbb{E}_{q(\beta, \mathbf{z}_n|\mathbf{X})}[\log p(\mathbf{x}_n, \mathbf{z}_n|\beta) - \log q(\mathbf{z}_n|\mathbf{x}_n, \beta)]$$

Ratio Estimation

Subtract the constant **empirical distribution** on the observations $\mathbf{X}$, $\log q(\mathbf{X}) = \sum_{n=1}^{N} \log q(\mathbf{x}_n)$, from the latter term:

$$\sum_{n=1}^{N} \left\{ \mathbb{E}_{q(\beta, \mathbf{z}_n | \mathbf{X})} [\log p(\mathbf{x}_n, \mathbf{z}_n | \beta) - \log q(\mathbf{z}_n | \mathbf{x}_n, \beta)] - \log q(\mathbf{x}_n) \right\} .$$

Substitute $q(\mathbf{x}_n, \mathbf{z}_n | \beta) \equiv q(\mathbf{x}_n)\, q(\mathbf{z}_n | \mathbf{x}_n, \beta)$ and $q(\beta, \mathbf{z}_n | \mathbf{X}) = q(\beta | \mathbf{X})\, q(\mathbf{z}_n | \mathbf{x}_n, \beta)$:

$$\sum_{n=1}^{N} \mathbb{E}_{q(\beta | \mathbf{X})\, q(\mathbf{z}_n | \mathbf{x}_n, \beta)} \left[\log \frac{p(\mathbf{x}_n, \mathbf{z}_n | \beta)}{q(\mathbf{x}_n, \mathbf{z}_n | \beta)} \right] .$$

We are going to estimate the log-ratio $\log(p(\cdot | \beta) / q(\cdot | \beta))$.

The New Evidence Lower Bound

Once we have an estimate of the log-ratio, $r(\mathbf{x}_n, \mathbf{z}_n, \beta) \simeq \log(p(\cdot|\beta)/q(\cdot|\beta))$, we can maximize the new ELBO:

$$\begin{aligned}\mathcal{L} \propto\ & \mathbb{E}_{q(\beta,\mathbf{Z}|\mathbf{X})}[\log p(\beta) - \log q(\beta|\mathbf{X})] \\ & + \sum_{n=1}^{N} \mathbb{E}_{q(\beta|\mathbf{X})\, q(\mathbf{z}_n|\mathbf{x}_n,\beta)}[r(\mathbf{x}_n, \mathbf{z}_n, \beta)]\,.\end{aligned}$$

But we are not there yet.

Back to Ratio Estimation

Use a GAN-style algorithm:

- Given a sample from $p(\cdot)$ ("fake") or $q(\cdot)$ ("real"), we seek to **estimate the probability** that it was drawn from $p(\cdot)$ ("fake"):
  - High probability: "fake" (from p).
  - Low probability: "real" (from q).
- Model this using $\sigma(r(\cdot;\theta))$, where:
  - r is a **GAN-like discrepancy function.**
  - θ are the parameters of r.
  - $\sigma(r) = (1 + \mathrm{e}^{-r})^{-1}$ is the logistic function that transforms the output of r such that it is in $(0, 1)$.

Negative Loss Function

Train estimator $r(\cdot;\theta)$ by **maximizing** a negative loss function,

$$\begin{aligned}\mathcal{D} = \mathbb{E}_{p(\mathbf{x}_n,\mathbf{z}_n|\beta)}[l_{\mathrm{fake}}(r(\mathbf{x}_n,\mathbf{z}_n,\beta;\theta))] \\ + \mathbb{E}_{q(\mathbf{x}_n,\mathbf{z}_n|\beta)}[l_{\mathrm{real}}(r(\mathbf{x}_n,\mathbf{z}_n,\beta;\theta))]\end{aligned}$$

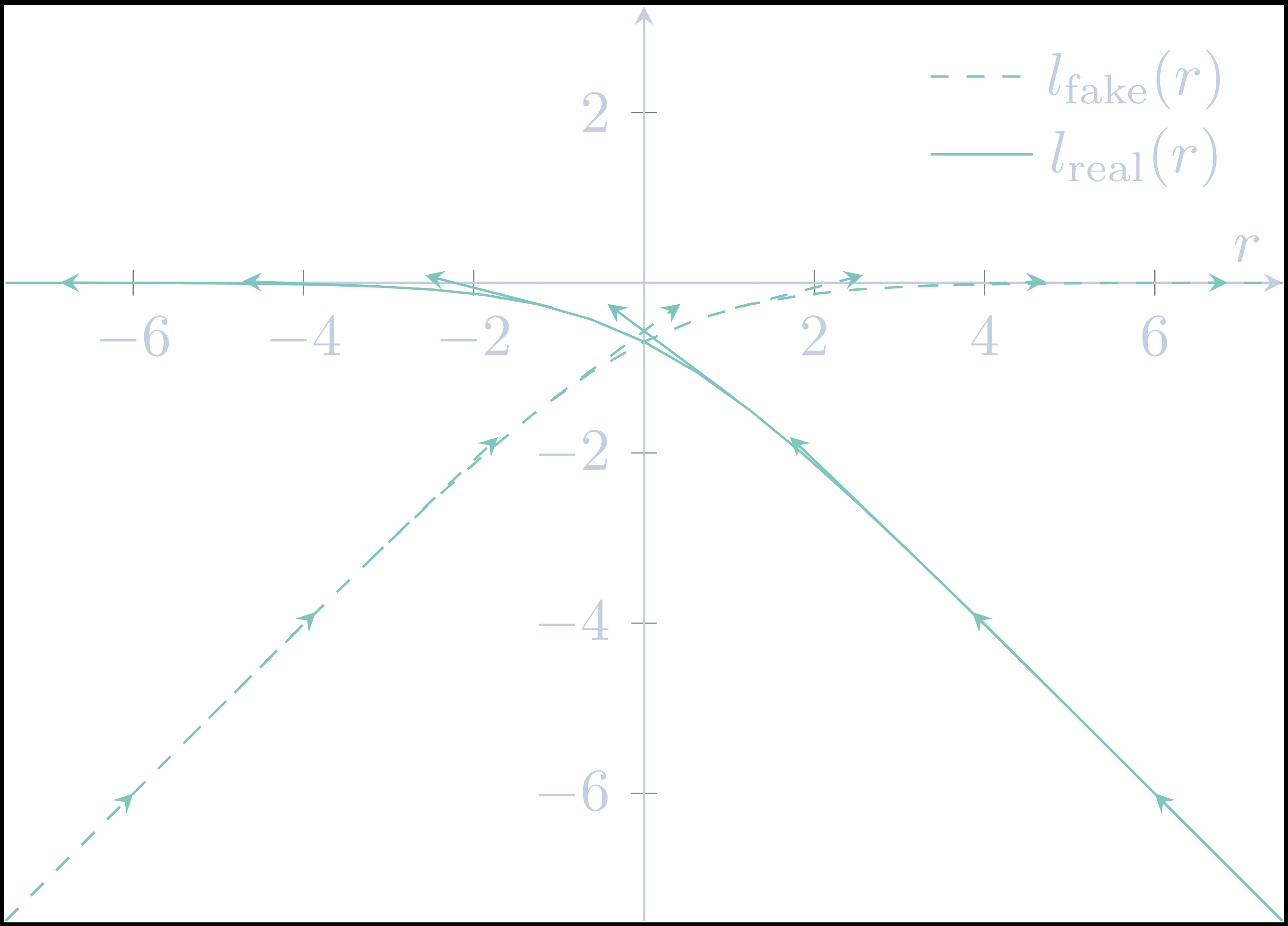
with

$$\begin{aligned}l_{\mathrm{fake}}(r) &= \log\sigma(r) = -\log(1+\mathrm{e}^{-r}) \equiv \mathrm{softminus}(r), \\ l_{\mathrm{real}}(r) &= \log(1-\sigma(r)) = -\log(1+\mathrm{e}^{r}) \equiv -\mathrm{softplus}(r).\end{aligned}$$

Thus:

- r is encouraged to be **large** when sample is "fake" (from p).
- r is encouraged to be **small** when sample is "real" (from q).

Negative Loss Function



r is encouraged to be **large** when sample is "fake" (from p).

r is encouraged to be **small** when sample is "real" (from q).

Why Does This Work?

Why does maximizing $\mathcal{D}$ with respect to r give us an estimator for the log-ratio $\log(p(\cdot|\beta)/q(\cdot|\beta))$? Consider:

$$\mathcal{D}[r] = \mathbb{E}_{x\sim p}[\mathrm{softminus}(r(x))] - \mathbb{E}_{x\sim q}[\mathrm{softplus}(r(x))]\,.$$

Pretend we can solve the maximization problem in function space:

$$\begin{aligned}0 = \frac{\delta}{\delta r(x)}\mathcal{D} &= p(x)\left[\frac{\mathrm{d}}{\mathrm{d}s}\mathrm{softminus}(s)\right]_{s=r(x)} \\ &\quad - q(x)\left[\frac{\mathrm{d}}{\mathrm{d}s}\mathrm{softplus}(s)\right]_{s=r(x)} \\ &= \frac{1}{1+\mathrm{e}^{r(x)}}\left\{p(x) - q(x)\,\mathrm{e}^{r(x)}\right\}\end{aligned}$$

Why Does This Work?

What's the **optimal function**? It's the solution of

$$0 = p(x) - q(x)\,\mathrm{e}^{r(x)},$$

that is: $r^*(x) = \log(p(x)/q(x))$ for $q(x) > 0$ or

$$r^*(\mathbf{x}_n, \mathbf{z}_n, \beta; \lambda, \phi)) = \log \frac{p(\mathbf{x}_n, \mathbf{z}_n | \beta)}{q(\mathbf{x}_n, \mathbf{z}_n | \beta; \lambda, \phi)}.$$

Thus we can **use our estimate $r(\cdot;\theta)$ as a proxy to the log-ratio**!

This is possible as long as the family $r(\cdot;\theta)$ is expressive enough to come close to the optimal function $r^*(\mathbf{x}_n, \mathbf{z}_n, \beta; \lambda, \phi))$.

Gradient of $\mathcal{D}$ with Respect to θ

$$\begin{aligned}\boldsymbol{\nabla}_\theta \mathcal{D}(\theta) = {} & \mathbb{E}_{p(\mathbf{x}_n, \mathbf{z}_n \vert \beta)} [\boldsymbol{\nabla}_\theta \, l_{\text{fake}}(r(\mathbf{x}_n, \mathbf{z}_n, \beta; \theta))] \\ & + \mathbb{E}_{q(\mathbf{x}_n, \mathbf{z}_n \vert \beta)} [\boldsymbol{\nabla}_\theta \, l_{\text{real}}(r(\mathbf{x}_n, \mathbf{z}_n, \beta; \theta))]\end{aligned}$$

Maximizing The ELBO

$$\mathcal{L}(\phi, \lambda, \theta) \propto \mathbb{E}_{q(\beta, \mathbf{Z}|\mathbf{X};\lambda)}[\log p(\beta) - \log q(\beta|\mathbf{X};\lambda)] \\ + \sum_{n=1}^{N} \mathbb{E}_{q(\beta|\mathbf{X};\lambda)\, q(\mathbf{z}_n|\mathbf{x}_n,\beta;\phi)}[r(\mathbf{x}_n, \mathbf{z}_n, \beta; \theta)] \,.$$

Needed gradients for SGA: $\nabla_\phi \mathcal{L}$, $\nabla_\lambda \mathcal{L}$.

Problem: ϕ and λ are appearing in the **probability measures** $q(\cdot)$ of the expectation value $\mathbb{E}_{q(\cdot)}$.

Solution: Use a differentiable transformation T to move ϕ and λ out of these measures.

Maximizing The ELBO

$$\mathcal{L}(\phi,\lambda,\theta) \propto \mathbb{E}_{q(\beta,\mathbf{Z}|\mathbf{X};\lambda)}[\log p(\beta) - \log q(\beta|\mathbf{X};\lambda)] + \sum_{n=1}^{N} \mathbb{E}_{q(\beta|\mathbf{X};\lambda)\, q(\mathbf{z}_n|\mathbf{x}_n,\beta;\phi)}[r(\mathbf{x}_n,\mathbf{z}_n,\beta;\theta)] \,.$$

Global and local transformations:

1. $\mathbb{E}_{q(\beta|\mathbf{X};\lambda)}[f(\beta)] \rightarrow \mathbb{E}_{\delta_{\text{global}}\sim s(\cdot)}[f\left(\mathbf{T}_{\text{global}}(\delta_{\text{global}};\lambda)\right)]$,
2. $\mathbb{E}_{q(\mathbf{z}_n|\mathbf{x}_n,\beta;\phi)}[h(\mathbf{z}_n)] \rightarrow \mathbb{E}_{\delta_{\text{local}}\sim s(\cdot)}[h\left(\mathbf{T}_{\text{local}}(\delta_{\text{local}},\mathbf{x}_n;\phi)\right)]$.

Example: $s(\cdot)$ is, say, a standard multivariate normal distribution. $\mathbf{T}$ can then be used t map this to a multivariate normal distribution with location μ and covariance $\mathbf{\Sigma}$.

Gradient of $\mathcal{L}$ with Respect to ϕ And λ

$$\nabla_\phi \mathcal{L} = \sum_{n=1}^{N} \mathbb{E}_{q(\beta|\mathbf{X};\lambda)}\left[\mathbb{E}_{\delta_{\text{local}}\sim s(\cdot)}[\nabla_\phi\, r(\mathbf{x}_n, \mathbf{z}_n, \beta; \theta)]\right],$$

$$\begin{aligned}\nabla_\lambda \mathcal{L} = {} & \mathbb{E}_{\delta_{\text{global}}\sim s(\cdot)}[\nabla_\lambda\left(\log p(\beta) - \log q(\beta|\mathbf{X};\lambda)\right)] \\ & + \sum_{n=1}^{N} \mathbb{E}_{\delta_{\text{global}}\sim s(\cdot)}\left[\mathbb{E}_{q(\mathbf{z}_n|\mathbf{x}_n,\beta;\phi)}[\nabla_\lambda\, r(\mathbf{x}_n, \mathbf{z}_n, \beta; \theta)]\right].\end{aligned}$$

The Finished Algorithm

Input:

Model: **implicit** likelihood $p(\mathbf{X}, \mathbf{Z}|\beta)$, tractable prior $p(\beta)$.

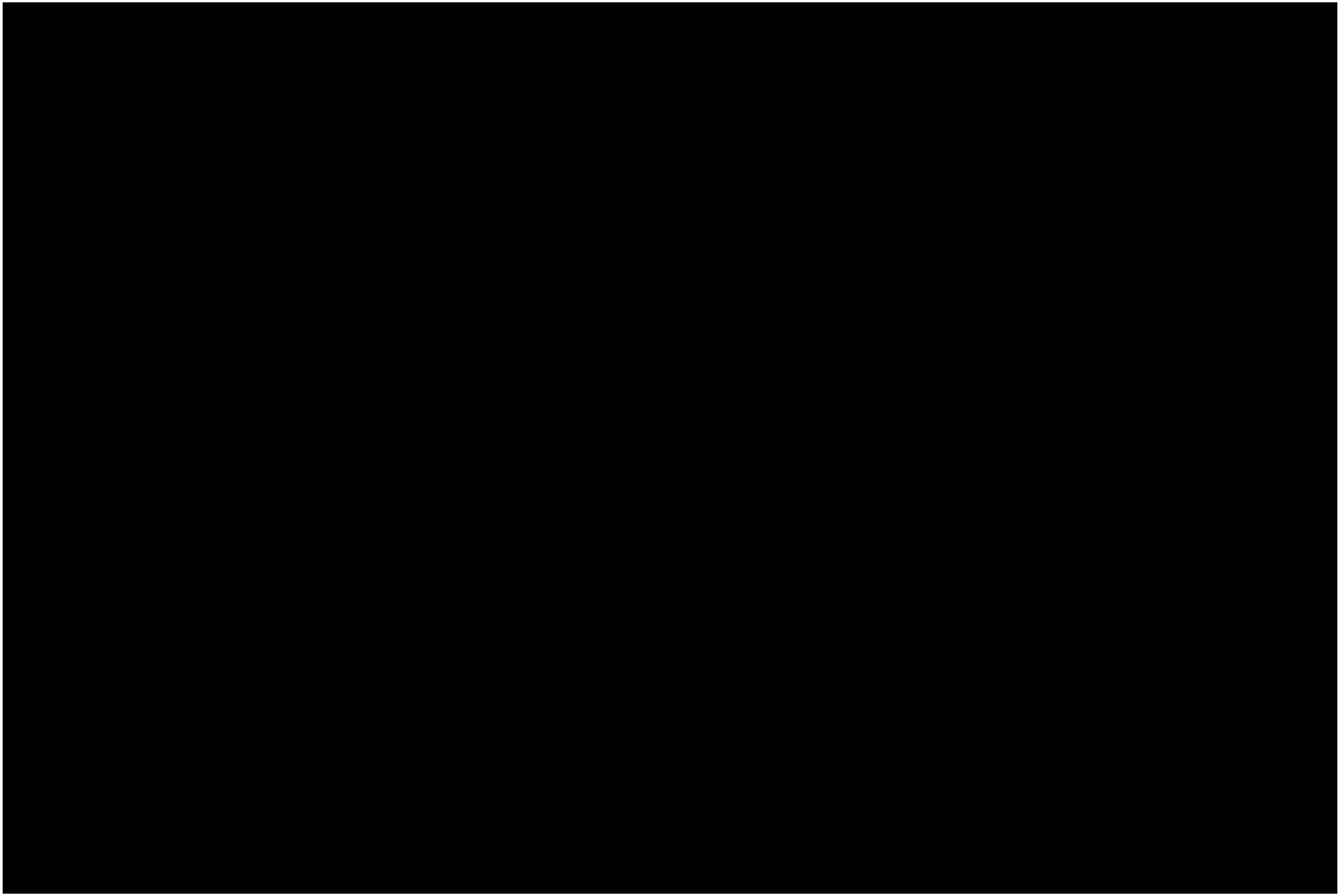
Variational approximation family: **implicit** likelihood $q(\mathbf{z}_n|\mathbf{x}_n, \beta; \phi)$, tractable prior $q(\beta|\mathbf{X}; \lambda)$.

Ratio estimate: $r(\cdot; \theta)$.

Output: Variational parameters ϕ and λ.

Algorithm:

Initialize ϕ, λ, and θ randomly.
While not converged do:
    Compute unbiased estimates of $\nabla_\theta \mathcal{D}(\theta)$, $\nabla_\phi \mathcal{L}$, and $\nabla_\lambda \mathcal{L}$.
    Update ϕ, λ, and θ using SGA.
End.



Can I Use This?

Bayesian GANs and GAN-style inference are available in **Edward**.

Where Can I Read More About This?

Tran et al., **Deep and Hierarchical Implicit Models**, arXiv:1702.08896

Tran et al., **Deep Probabilistic Programming**, arXiv:1701.03757

Mescheder et al., **Adversarial Variational Bayes: Unifying Variational Autoencoders and Generative Adversarial Networks**, arXiv:1701.04722

F. Huszár, **Variational Inference using Implicit Distributions**, arXiv:1702.08235

Goodfellow et al., **Generative Adversarial Nets**, arXiv:1406.2661

I. Goodfellow, **Generative Adversarial Networks**, arXiv:1701.00160, NIPS 2016 Tutorial

Blei et al., **Variational Inference: Foundations and Modern Methods**, NIPS 2016 Tutorial

1