Overdispersed Variational Autoencoders

Harshil Shah*, David Barber* † and Aleksandar Botev*
*Department of Computer Science, University College London
†Alan Turing Institute
harshil.shah.15@ucl.ac.uk, david.barber@ucl.ac.uk, aleksandar.botev.14@ucl.ac.uk

***Abstract*—The ability to fit complex generative probabilistic models to data is a key challenge in AI. Currently, variational methods are popular, but remain difficult to train due to high variance of the sampling methods employed. We introduce the overdispersed variational autoencoder and overdispersed importance weighted autoencoder, which combine overdispersed black box variational inference with the variational autoencoder and importance weighted autoencoder respectively. We use the log likelihood lower bounds and reparametrisation trick from the variational and importance weighted autoencoders, but rather than drawing samples from the variational distribution itself, we use importance sampling to draw samples from an overdispersed (i.e. heavier-tailed) proposal in the same family as the variational distribution. We run experiments on two different datasets, and show that this technique produces a lower variance estimate of the gradients, and reaches a higher bound on the log likelihood of the observed data.**

I. Introduction

A generative model specifies a conditional distribution $p(\mathbf{x}|\mathbf{h})$ of observed data $\mathbf{x}$ given hidden (latent) variables $\mathbf{h}$. Combined with a distribution $p(\mathbf{h})$ over the hidden variables, the model can be used to generate observations by sampling from $p(\mathbf{h})$ and then $p(\mathbf{x}|\mathbf{h})$. Learning a generative model can be achieved by maximising the likelihood of the observed data with respect to the model parameters. The expectation maximisation (EM) algorithm [1] is guaranteed to find a local maximum of the likelihood, by maximising the log likelihood. However, EM only works in the simplest of models, specifically those where the true posterior distribution of the latent variables is tractable. The variational EM [2] approach addresses models where this is not the case, by introducing an approximation to the true posterior (known as the variational distribution). The resulting algorithm maximises an evidence[1] lower bound (ELBO). This usually involves assuming a parametrised form for the variational distribution, and taking gradient steps with respect to these parameters in order to maximise the ELBO.

The recent variational autoencoder (VAE) [3], [4] has made variational inference possible on a large scale, by providing a stochastic objective function which is used to jointly optimise the generative and variational parameters. It does so by drawing samples from the variational distribution to form a Monte Carlo estimate of the ELBO. The derivatives of this lower bound are then computed by reparametrising the variational distribution as a differentiable function of a 'base' distribution, and gradient steps are taken in both the generative and variational parameters. A recent variant of the VAE is the importance weighted autoencoder (IWAE) [5], which uses importance sampling to maximise a tighter lower bound on the log likelihood than does the VAE.

Where typical variational inference algorithms propose drawing samples from the variational distribution, overdispersed black-box variational inference (O-BBVI) [6] proposes drawing samples from a distribution with the same functional form as the variational distribution, but with heavier tails. The aim of this is to cover those regions where the true posterior has high density, but the variational distribution may not. This results in a lower variance estimate of the gradients, particularly when the variational distribution is a poor fit to the true posterior. However, unlike the VAE and IWAE, O-BBVI uses a REINFORCE style update [7], which is likely to have higher variance than using the reparametrisation trick.

In this paper, we introduce the overdispersed variational autoencoder (OVAE) and overdispersed importance weighted autoencoder (OIWAE), which combine the idea of an overdispersed proposal distribution with the reparametrisation technique from the VAE and IWAE respectively. We prove, empirically, that compared to the VAE and IWAE, this technique produces a lower variance estimate of the gradients with respect to the generative and variational parameters. The resulting trained models achieve higher bounds on the log likelihood of the observed data.

II. Background

In this section, we review expectation maximisation, the variational autoencoder (VAE) and importance weighted autoencoder (IWAE).

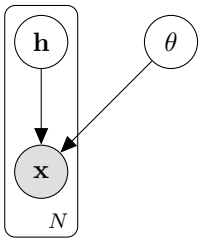


Fig. 1. The generative model under consideration

The generative model under consideration is that of figure 1, where $\mathbf{x}$ is the vector of observations, $\mathbf{h}$ is a latent vector, and θ are the generative model parameters. The data generating process is as follows:

This work was supported by The Alan Turing Institute under the EPSRC grant EP/N510129/1

[1]The evidence is another term for the log likelihood of the observed data under the model.

1) A value for the latent vector $\mathbf{h}$ is generated from the prior $p(\mathbf{h})$.
2) A value for the observation vector $\mathbf{x}$ is generated from the conditional distribution $p_\theta(\mathbf{x}|\mathbf{h})$.

This means that the generative model is:

$$p_\theta(\mathbf{h}, \mathbf{x}) = p(\mathbf{h})p_\theta(\mathbf{x}|\mathbf{h}) \tag{1}$$

The values of the generative parameters θ, and the latent vector $\mathbf{h}$ are unknown. The task is to infer the values of the generative parameters θ that maximise the likelihood $p_\theta(\mathbf{x})$.

A. Expectation Maximisation

For any valid distribution $q(\mathbf{h})$, the evidence lower bound (ELBO) can be formed using Jensen's inequality, as follows:

$$\begin{aligned}\log p_\theta(\mathbf{x}) &= \log\left(\mathbb{E}_{q(\mathbf{h})}\left[\frac{p_\theta(\mathbf{h},\mathbf{x})}{q(\mathbf{h})}\right]\right) &(2)\\ &\geq \mathbb{E}_{q(\mathbf{h})}\left[\log\left(\frac{p_\theta(\mathbf{h},\mathbf{x})}{q(\mathbf{h})}\right)\right] &(3)\\ &= \mathcal{L}(\mathbf{x})\end{aligned}$$

The Variational Expectation Maximisation (vEM) [2] finds the values of the generative parameters θ that maximise the log likelihood by alternating between:

- **E step**: optimise $\mathcal{L}(\mathbf{x})$ with respect to the distribution over the latent variables $q(\mathbf{h})$ while holding the generative parameters θ fixed.
- **M step**: optimise $\mathcal{L}(\mathbf{x})$ with respect to the generative parameters θ while holding the distribution over the latent variables $q(\mathbf{h})$ fixed.

While the M step often has a closed form solution (or if not, can simply be solved using stochastic gradient ascent), the E step can be broken down further. The ELBO can be rewritten as follows:

$$\begin{aligned}\mathcal{L}(\mathbf{x}) &= \mathbb{E}_{q(\mathbf{h})}\left[\log p_\theta(\mathbf{x})\right] &(4)\\ &\quad \mathbb{E}_{q(\mathbf{h})}\left[\log p_\theta(\mathbf{h}|\mathbf{x}) - \log q(\mathbf{h})\right]\\ &= \log p_\theta(\mathbf{x}) - KL\left[q(\mathbf{h}) \,||\, p_\theta(\mathbf{h}|\mathbf{x})\right] &(5)\end{aligned}$$

The first term in equation (5) is the log likelihood, while the second is the Kullback-Leibler (KL) divergence [8] between $q(\mathbf{h})$ and the true posterior $p_\theta(\mathbf{h}|\mathbf{x})$. It is provable [9] that $KL[q||p] \geq 0$ with equality if and only if $q = p$. It is easy to see, then, that in the E step, the ELBO $\mathcal{L}(\mathbf{x})$ is maximised by setting $q(\mathbf{h}) = p_\theta(\mathbf{h}|\mathbf{x})$.

B. VAE

In most models of interest, the true posterior $p_\theta(\mathbf{h}|\mathbf{x})$ is intractable, and therefore the E step must be modified. Instead of having an E step where the distribution $q(\mathbf{h})$ is set to the true posterior $p_\theta(\mathbf{h}|\mathbf{x})$, the variational autoencoder (VAE) [3] introduces the variational parameters ϕ which parametrise the distribution $q_\phi(\mathbf{h}|\mathbf{x})$. Denoting $w_{\theta,\phi}(\mathbf{h}) = \frac{p_\theta(\mathbf{h},\mathbf{x})}{q_\phi(\mathbf{h}|\mathbf{x})}$, the ELBO is defined to be:

$$\mathcal{L}^{\text{VAE}}(\mathbf{x}) \equiv \mathbb{E}_{q_\phi(\mathbf{h}|\mathbf{x})}\left[\log w_{\theta,\phi}(\mathbf{h})\right] \tag{6}$$

Gradient steps are taken with respect to both the generative parameters θ and the variational parameters ϕ in order to optimise the bound.

The derivatives with respect to the parameters could be computed using a REINFORCE style estimator [7]. However, this is believed to be a high variance estimator of the derivatives. Instead, under certain mild conditions [3], the latent vector $\mathbf{h} \sim q_\phi(\mathbf{h}|\mathbf{x})$ can be reparametrised using a differentiable transformation $g_\phi(\boldsymbol{\epsilon}, \mathbf{x})$, for some variable $\boldsymbol{\epsilon}$ such that $\mathbf{h} = g_\phi(\boldsymbol{\epsilon}, \mathbf{x})$ where $\boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon})$. Then, denoting $w_{\theta,\phi}(\boldsymbol{\epsilon}) = \frac{p_\theta(g_\phi(\boldsymbol{\epsilon},\mathbf{x}),\mathbf{x})}{q_\phi(g_\phi(\boldsymbol{\epsilon},\mathbf{x})|\mathbf{x})}$ the derivatives with respect to the parameters are computed as follows:

$$\begin{aligned}\nabla_{\theta,\phi}\mathcal{L}^{\text{VAE}}(\mathbf{x}) &= \nabla_{\theta,\phi}\mathbb{E}_{q_\phi(\mathbf{h}|\mathbf{x})}\left[\log w_{\theta,\phi}(\mathbf{h})\right] &(7)\\ &= \nabla_{\theta,\phi}\mathbb{E}_{p(\boldsymbol{\epsilon})}\left[\log w_{\theta,\phi}(\boldsymbol{\epsilon})\right] &(8)\\ &= \mathbb{E}_{p(\boldsymbol{\epsilon})}\left[\nabla_{\theta,\phi}\log w_{\theta,\phi}(\boldsymbol{\epsilon})\right] &(9)\\ &\simeq \frac{1}{S}\sum_{s=1}^{S}\nabla_{\theta,\phi}\log w_{\theta,\phi}(\boldsymbol{\epsilon}^{(s)}) &(10)\\ &\quad \text{where } \boldsymbol{\epsilon}^{(s)} \sim p(\boldsymbol{\epsilon})\end{aligned}$$

C. IWAE

The importance weighted autoencoder (IWAE) [5] maximises a tighter lower bound on the log likelihood than does the VAE. Denoting $w_{\theta,\phi}(\mathbf{h}^{(s)}) = \frac{p_\theta(\mathbf{h}^{(s)},\mathbf{x})}{q_\phi(\mathbf{h}^{(s)}|\mathbf{x})}$, and therefore $\bar{w}^{\mathbf{h}}_{\theta,\phi} = \frac{1}{S}\sum_{s=1}^{S} w_{\theta,\phi}(\mathbf{h}^{(s)})$, the IWAE bound is:

$$\mathcal{L}^{\text{IWAE}}_S(\mathbf{x}) \equiv \mathbb{E}_{\mathbf{h}^{(1:S)}\sim q_\phi(\mathbf{h}|\mathbf{x})}\left[\log\left(\bar{w}^{\mathbf{h}}_{\theta,\phi}\right)\right] \tag{11}$$

This bound corresponds to the S-sample importance weighted estimate of the log-likelihood, and is tighter than the VAE bound of equation (6) when $S > 1$. When $S = 1$, the two bounds are identical. It is also shown in [5], that when $T \geq S, \mathcal{L}^{\text{IWAE}}_T(\mathbf{x}) \geq \mathcal{L}^{\text{IWAE}}_S(\mathbf{x})$.

To compute the derivatives with respect to the generative and variational parameters, the reparametrisation trick [5], is again employed. The latent vector $\mathbf{h} \sim q_\phi(\mathbf{h}|\mathbf{x})$ is reparametrised using a differentiable transformation $g_\phi(\boldsymbol{\epsilon}, \mathbf{x})$, for some variable $\boldsymbol{\epsilon}$ such that $\mathbf{h} = g_\phi(\boldsymbol{\epsilon}, \mathbf{x})$ where $\boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon})$. Then, denoting $w_{\theta,\phi}(\boldsymbol{\epsilon}^{(s)}) = \frac{p_\theta(g_\phi(\boldsymbol{\epsilon}^{(s)},\mathbf{x}),\mathbf{x})}{q_\phi(g_\phi(\boldsymbol{\epsilon}^{(s)},\mathbf{x})|\mathbf{x})}$, and therefore $\bar{w}^{\boldsymbol{\epsilon}}_{\theta,\phi} = \frac{1}{S}\sum_{s=1}^{S} w_{\theta,\phi}(\boldsymbol{\epsilon}^{(s)})$, the derivatives become:

$$\begin{aligned}\nabla_{\theta,\phi}\mathcal{L}^{\text{IWAE}}(\mathbf{x}) &= \nabla_{\theta,\phi}\mathbb{E}_{\mathbf{h}^{(1:S)}\sim q_\phi(\mathbf{h}|\mathbf{x})}\left[\log\left(\bar{w}^{\mathbf{h}}_{\theta,\phi}\right)\right] &(12)\\ &= \nabla_{\theta,\phi}\mathbb{E}_{\boldsymbol{\epsilon}^{(1:S)}\sim p(\boldsymbol{\epsilon})}\left[\log\left(\bar{w}^{\boldsymbol{\epsilon}}_{\theta,\phi}\right)\right] &(13)\\ &= \mathbb{E}_{\boldsymbol{\epsilon}^{(1:S)}\sim p(\boldsymbol{\epsilon})}\left[\nabla_{\theta,\phi}\log\left(\bar{w}^{\boldsymbol{\epsilon}}_{\theta,\phi}\right)\right] &(14)\\ &\simeq \nabla_{\theta,\phi}\log\left(\bar{w}^{\boldsymbol{\epsilon}}_{\theta,\phi}\right) &(15)\end{aligned}$$

where $\boldsymbol{\epsilon}^{(s)} \sim p(\boldsymbol{\epsilon})$. Note, again, that a single sample is used for each $\boldsymbol{\epsilon}^{(s)}$ in the expectation.

III. Overdispersion

In this section, we review the idea of overdispersion, from overdispersed black-box variational inference (O-BBVI) [6]. Below, a slight abuse of notation is used, for the purpose

of providing clarity and intuition behind the overdispersion technique. Denote:

$$f(\mathbf{h}) = \nabla_\phi \left(\log p_\theta(\mathbf{h}_\phi, \mathbf{x}) - \log q_\phi(\mathbf{h}_\phi|\mathbf{x}) \right) \tag{16}$$

Then, the VAE gradient of equation (9) is rewritten, without the reparametrisation trick, as:

$$\mathbb{E}_{q_\phi(\mathbf{h}|\mathbf{x})} \left[f(\mathbf{h}) \right] \tag{17}$$

Note that this is not mathematically valid because the derivative with respect to the variational parameters ϕ cannot be taken inside the expectation, unless the reparametrisation trick is used. However, this is done solely to provide intuition behind the overdispersion technique.

The standard Monte Carlo approach to estimate this expression would be to draw samples from the variational distribution $q_\phi(\mathbf{h}|\mathbf{x})$, and evaluate the quantity:

$$\frac{1}{S} \sum_{s=1}^{S} f(\mathbf{h}^{(s)}) \quad \text{where } \mathbf{h}^{(s)} \sim q_\phi(\mathbf{h}|\mathbf{x}) \tag{18}$$

This is an unbiased estimate of the true gradient, but this estimate can have high variance [10]. To find a lower variance estimate, we turn to importance sampling. Samples are drawn from a proposal distribution, $r(\mathbf{h}|\mathbf{x})$, and weighted by $\frac{q_\phi(\mathbf{h}|\mathbf{x})}{r(\mathbf{h}|\mathbf{x})}$. Notice that:

$$\mathbb{E}_{q_\phi(\mathbf{h}|\mathbf{x})} \left[f(\mathbf{h}) \right] = \mathbb{E}_{r(\mathbf{h}|\mathbf{x})} \left[\frac{q_\phi(\mathbf{h}|\mathbf{x})}{r(\mathbf{h}|\mathbf{x})} f(\mathbf{h}) \right] \tag{19}$$

Therefore, the importance sampling estimate of the gradient would be:

$$\frac{1}{S} \sum_{s=1}^{S} \frac{q_\phi(\mathbf{h}^{(s)}|\mathbf{x})}{r(\mathbf{h}^{(s)}|\mathbf{x})} f(\mathbf{h}^{(s)}) \quad \text{where } \mathbf{h}^{(s)} \sim r(\mathbf{h}|\mathbf{x}) \tag{20}$$

A. The optimal proposal

The importance sampling literature [11] states that the optimal proposal distribution (that which would minimise the variance of the gradient estimate of equation (20)) is not $q_\phi(\mathbf{h}|\mathbf{x})$, but rather:

$$r_i^*(\mathbf{h}|\mathbf{x}) = \frac{q_\phi(\mathbf{h}|\mathbf{x})|f_i(\mathbf{h})|}{\int q_\phi(\mathbf{z}|\mathbf{x})|f_i(\mathbf{z})| \, d\mathbf{z}} \propto q_\phi(\mathbf{h}|\mathbf{x})|f_i(\mathbf{h})| \tag{21}$$

where the subscript i denotes the i^{th} component of the gradient.

It is also provable that, under the optimal proposal, the importance sampling estimate has lower variance than does the simple Monte Carlo estimate. However, the optimal distribution $r_i^*(\mathbf{h}|\mathbf{x})$ is, in general, intractable. In O-BBVI, an alternative proposal $r_{\phi,\tau}(\mathbf{h}|\mathbf{x})$ is used from the same family as the variational distribution $q_\phi(\mathbf{h}|\mathbf{x})$, but with an additional vector (of the same dimensionality as $\mathbf{h}$) of dispersion parameters, with elements $\tau_i \geq 1$ to control the dispersion of the distribution. When $\tau_i > 1$, the proposal distribution assigns higher mass to the tails of $q_\phi(h_i|\mathbf{x})$, and when $\tau_i = 1$, $r_{\phi,\tau_i}(h_i|\mathbf{x}) = q_\phi(h_i|\mathbf{x})$.

To see why the use of an overdispersed proposal distribution is effective [6], consider the expression $f(\mathbf{h})$ from equation (16). As is known from the vEM algorithm, the optimal variational distribution is the true posterior $p_\theta(\mathbf{h}|\mathbf{x})$, in which case $f(\mathbf{h}) = 0$. However, when $q_\phi(\mathbf{h}|\mathbf{x})$ is a bad fit to the true posterior, there are values of $\mathbf{h}$ for which the posterior is high, but the variational distribution is small. In these cases, $f(\mathbf{h})$ will have a large absolute value, and these realisations lie in the tails of the variational distribution. Therefore, from the relation in equation (21), we can see that the optimal proposal would push more mass towards the tails of $q_\phi(\mathbf{h}|\mathbf{x})$. Therefore an overdispersed proposal should result in lower variance when estimating the gradient of the ELBO.

IV. Overdispersed variational autoencoders

In this section, we describe the overdispersed variational autoencoder (OVAE) and the overdispersed importance weighted autoencoder (OIWAE). Throughout, it is assumed that the variational distribution $q_\phi(\mathbf{h}|\mathbf{x})$ takes a fully factorised (mean field) form, such as a Gaussian distribution with a diagonal covariance matrix. This means that, in the variational distribution, the components of $\mathbf{h}$ are independent of each other.

A. OVAE

Taking equation (6), but now drawing samples from the overdispersed proposal distribution $r_{\phi,\tau}(\mathbf{h}|\mathbf{x})$, we obtain the lower bound for the OVAE:

$$\mathcal{L}^{\text{OVAE}}(\mathbf{x}) \equiv \mathbb{E}_{r_{\phi,\tau}(\mathbf{h}|\mathbf{x})} \left[\frac{q_\phi(\mathbf{h}|\mathbf{x})}{r_{\phi,\tau}(\mathbf{h}|\mathbf{x})} \log \left(w_{\theta,\phi}(\mathbf{h}) \right) \right] \tag{22}$$

Computing the importance weights $\frac{q_\phi(\mathbf{h}|\mathbf{x})}{r_{\phi,\tau}(\mathbf{h}|\mathbf{x})}$ is numerically unstable in high dimensions since both the numerator and denominator tend to 0. However, because the variational distribution takes a fully factorised form, the components of $\mathbf{h}$ are independent of each other. This means that a restriction can be made, such that only the i^{th} dimension is drawn from the overdispersed proposal, while the other dimensions are drawn from the variational distribution:

$$\mathcal{L}^{\text{OVAE}}(\mathbf{x}) = \mathbb{E}_{r_{\phi,\tau_i}(h_i|\mathbf{x})} \left[\frac{q_\phi(h_i|\mathbf{x})}{r_{\phi,\tau_i}(h_i|\mathbf{x})} \times \mathbb{E}_{q_\phi(\mathbf{h}_{\neg i}|\mathbf{x})} \left[\log \left(w_{\theta,\phi}(\mathbf{h}) \right) \right] \right] \tag{23}$$

This means that the importance weight is computed only for a single dimension, which is now numerically stable. Note that the dimension i is selected randomly at each training iteration.

To compute the derivatives with respect to the generative and variational parameters, the reparametrisation trick, as per the VAE [3], is employed. The vector $\mathbf{h}$ is reparametrised using the differentiable transformation $g_{\phi,\tau_i}(\boldsymbol{\epsilon}, \mathbf{x})$ for some variable $\boldsymbol{\epsilon}$ such that $\mathbf{h} = g_{\phi,\tau_i}(\boldsymbol{\epsilon}, \mathbf{x})$ where $\boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon})$. Note that:

$$\mathbf{h}_{\neg i} = g_\phi(\boldsymbol{\epsilon}_{\neg i}, \mathbf{x}) \sim q_\phi(\mathbf{h}_{\neg i}|\mathbf{x}) \tag{24}$$

$$h_i = g_{\phi,\tau_i}(\epsilon_i, \mathbf{x}) \sim r_{\phi,\tau_i}(h_i|\mathbf{x}) \tag{25}$$

We denote:

$$v_{\phi,\tau_i}(\epsilon_i) = \frac{q_\phi(g_{\phi,\tau_i}(\epsilon_i, \mathbf{x})|\mathbf{x})}{r_{\phi,\tau_i}(g_{\phi,\tau_i}(\epsilon_i, \mathbf{x})|\mathbf{x})}$$
$$w_{\theta,\phi,\tau_i}(\boldsymbol{\epsilon}) = \frac{p_\theta(g_{\phi,\tau_i}(\boldsymbol{\epsilon}, \mathbf{x}), \mathbf{x})}{q_\phi(g_{\phi,\tau_i}(\boldsymbol{\epsilon}, \mathbf{x})|\mathbf{x})}$$

Then, the derivatives are:

$$\nabla_{\theta,\phi}\mathcal{L}^{\text{OVAE}}(\mathbf{x}) = \mathbb{E}_{p(\epsilon_i)}\left[\nabla_{\theta,\phi} v_{\phi,\tau_i}(\epsilon_i) \times \mathbb{E}_{p(\boldsymbol{\epsilon}_{\neg i})}\left[\log w_{\theta,\phi,\tau_i}(\boldsymbol{\epsilon})\right]\right] \tag{26}$$

In practice, only a single sample is used to estimate the inner expectation, while S samples are used for the outer expectation. Therefore the quantity computed is:

$$\hat{\nabla}_{\theta,\phi}\mathcal{L}^{\text{OVAE}}(\mathbf{x}) = \frac{1}{S}\sum_{s=1}^{S} \nabla_{\theta,\phi} v_{\phi,\tau_i}(\epsilon_i^{(s)}) \times \log w_{\theta,\phi,\tau_i}(\boldsymbol{\epsilon}^{(s)}) \tag{27}$$

where $\boldsymbol{\epsilon}^{(s)} \sim p(\boldsymbol{\epsilon})$.

1) Dispersion parameters: As per O-BBVI [6], we optimise the dispersion parameters τ to minimise the variance of the estimated gradient in equation (27). After some algbera, we have:

$$\mathbb{V}\left[\hat{\nabla}_{\theta,\phi}\mathcal{L}^{\text{OVAE}}(\mathbf{x})\right] = \frac{1}{S}\mathbb{E}_{p(\boldsymbol{\epsilon})}\left[\left(\nabla_{\theta,\phi} v_{\phi,\tau_i}^{(s)} \log w_{\theta,\phi,\tau_i}^{(s)}\right)^2\right] - \frac{1}{S}\left(\hat{\nabla}_{\theta,\phi}\mathcal{L}^{\text{OVAE}}(\mathbf{x})\right)^2 \tag{28}$$

Note that we have used the shorthand notation $v_{\phi,\tau_i}^{(s)} = v_{\phi,\tau_i}(\epsilon_i^{(s)})$ and $w_{\theta,\phi,\tau_i}^{(s)} = w_{\theta,\phi,\tau_i}(\boldsymbol{\epsilon}^{(s)})$. Notice that the second term is independent of τ_i, so the gradient of the variance can be estimated as:

$$\hat{\nabla}_{\tau_i}\mathbb{V}\left[\hat{\nabla}_{\theta,\phi}\mathcal{L}^{\text{OVAE}}(\mathbf{x})\right] = \frac{1}{S^2}\sum_{s=1}^{S}\nabla_{\tau_i}\left(\nabla_{\theta,\phi}\, v_{\phi,\tau_i}^{(s)} \times \log\left(w_{\theta,\phi,\tau_i}\right)\right)^2 \tag{29}$$

Note that we use the same set of S samples that were used to compute $\hat{\nabla}_{\theta,\phi}\mathcal{L}^{\text{OVAE}}(\mathbf{x})$ to compute $\hat{\nabla}_{\tau_i}\mathbb{V}\left[\hat{\nabla}_{\theta,\phi}\mathcal{L}^{\text{OVAE}}(\mathbf{x})\right]$.

B. OIWAE

Taking the IWAE lower bound from equation (11), but instead, drawing samples from the overdispersed proposal distribution $r_{\phi,\tau}(\mathbf{h}|\mathbf{x})$, we obtain the OIWAE lower bound. As with the OVAE, computing the importance weights $\frac{q_\phi(\mathbf{h}|\mathbf{x})}{r_{\phi,\tau}(\mathbf{h}|\mathbf{x})}$ is numerically unstable in high dimensions, therefore only the i^{th} dimension is drawn from the overdispersed proposal:

$$\mathcal{L}^{\text{OIWAE}}(\mathbf{x}) = \mathbb{E}_{h_i^{(1:S)}\sim r_{\phi,\tau_i}(h_i|\mathbf{x})}\left[\left(\prod_{s=1}^{S}\frac{q_\phi(h_i^{(s)}|\mathbf{x})}{r_{\phi,\tau_i}(h_i^{(s)}|\mathbf{x})}\right) \times \mathbb{E}_{\mathbf{h}_{\neg i}^{(1:S)}\sim q_\phi(\mathbf{h}_{\neg i}|\mathbf{x})}\left[\log\left(\bar{w}_{\theta,\phi}^{\mathbf{h}}\right)\right]\right] \tag{30}$$

There is another issue here, which is that taking the product of importance weights across samples, $\prod_{s=1}^{S}\frac{q_\phi(h_i^{(s)}|\mathbf{x})}{r_{\phi,\tau_i}(h_i^{(s)}|\mathbf{x})}$, is also numerically unstable. Therefore, in the OIWAE, only one dimension of one sample is drawn from the overdispersed proposal:

$$\mathcal{L}^{\text{OIWAE}}(\mathbf{x}) = \mathbb{E}_{h_i^{(t)}\sim r_{\phi,\tau_i}(h_i|\mathbf{x})}\left[\frac{q_\phi(h_i^{(t)}|\mathbf{x})}{r_{\phi,\tau_i}(h_i^{(t)}|\mathbf{x})} \times \mathbb{E}_{\mathbf{h}^{(1:S)\neg(t)}\sim q_\phi(\mathbf{h}|\mathbf{x}),\ \mathbf{h}_{\neg i}^{(t)}\sim q_\phi(\mathbf{h}_{\neg i}|\mathbf{x})}\left[\log\left(\bar{w}_{\theta,\phi}^{\mathbf{h}}\right)\right]\right] \tag{31}$$

Note that both the overdispersion sampling index t and the dimension i are selected at random at each training iteration.

To compute the derivatives, the reparametrisation trick is again employed. The vector $\mathbf{h}$ is reparametrised using the differentiable transformation $g_{\phi,\tau_i}(\boldsymbol{\epsilon}, \mathbf{x})$ for some variable $\boldsymbol{\epsilon}$ such that $\mathbf{h} = g_{\phi,\tau_i}(\boldsymbol{\epsilon}, \mathbf{x})$ where $\boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon})$. Note that:

$$\mathbf{h}^{(1:S)\neg(t)} = g_\phi(\boldsymbol{\epsilon}^{(1:S)\neg(t)}, \mathbf{x}) \sim q_\phi(\mathbf{h}|\mathbf{x}) \tag{32}$$
$$\mathbf{h}_{\neg i}^{(t)} = g_\phi(\boldsymbol{\epsilon}_{\neg i}^{(t)}, \mathbf{x}) \sim q_\phi(\mathbf{h}_{\neg i}|\mathbf{x}) \tag{33}$$
$$h_i^{(t)} = g_{\phi,\tau_i}(\epsilon_i^{(t)}, \mathbf{x}) \sim r_{\phi,\tau_i}(h_i|\mathbf{x}) \tag{34}$$

Therefore, using the same notation as defined for the OVAE:

$$\nabla_{\theta,\phi}\mathcal{L}^{\text{OIWAE}}(\mathbf{x}) = \mathbb{E}_{\epsilon_i^{(t)}\sim p(\epsilon_i)}\left[\nabla_{\theta,\phi} v_{\phi,\tau_i}^{(t)} \times \mathbb{E}_{\boldsymbol{\epsilon}^{(1:S)\neg(t)}\sim p(\boldsymbol{\epsilon}),\ \boldsymbol{\epsilon}_{\neg i}^{(t)}\sim p(\boldsymbol{\epsilon}_{\neg i})}\left[\log\left(\bar{w}_{\theta,\phi,\tau_i}^{\boldsymbol{\epsilon}}\right)\right]\right] \tag{35}$$

Using a single sample to approximate both expectations, the quantity computed is:

$$\hat{\nabla}_{\theta,\phi}\mathcal{L}^{\text{OIWAE}}(\mathbf{x}) = \nabla_{\theta,\phi} v_{\phi,\tau_i}^{(t)} \log\left(\bar{w}_{\theta,\phi,\tau_i}^{\boldsymbol{\epsilon}}\right) \tag{36}$$

As with the IWAE itself, the term:

$$\log\left(\bar{w}_{\theta,\phi,\tau_i}^{\boldsymbol{\epsilon}}\right) = \log\left(\frac{1}{S}\sum_{s=1}^{S}\frac{p_\theta(g_{\phi,\tau_i}(\boldsymbol{\epsilon}^{(s)}, \mathbf{x}), \mathbf{x})}{q_\phi(g_{\phi,\tau_i}(\boldsymbol{\epsilon}^{(s)}, \mathbf{x})|\mathbf{x})}\right)$$

can be difficult to compute, because both the numerator and denominator of the weights $\frac{p_\theta(g_{\phi,\tau_i}(\boldsymbol{\epsilon}^{(s)}, \mathbf{x}), \mathbf{x})}{q_\phi(g_{\phi,\tau_i}(\boldsymbol{\epsilon}^{(s)}, \mathbf{x})|\mathbf{x})}$ become very close to 0. Instead, it is much easier, numerically, to work with the log weights, as follows. Denoting $u_{\theta,\phi,\tau_i}^{(s)} = e^{v_{\phi,\tau_i}^{(t)}}\, w_{\theta,\phi,\tau_i}^{(s)}$, we have, as per [5], that:

$$\hat{\nabla}_{\theta,\phi}\mathcal{L}^{\text{OIWAE}}(\mathbf{x}) = \nabla_{\theta,\phi}\log\left(\frac{1}{S}\sum_{s=1}^{S} u_{\theta,\phi,\tau_i}^{(s)}\right) \tag{37}$$
$$= \sum_{s=1}^{S}\tilde{u}_{\theta,\phi,\tau_i}^{(s)}\nabla_{\theta,\phi}\log\left(u_{\theta,\phi,\tau_i}^{(s)}\right) \tag{38}$$

1) Dispersion parameters: As per O-BBVI [6], and as we did for the OVAE, we again optimise the dispersion parameters τ to minimise the variance, across samples, of the estimated gradient in equation (38). After some algebra, we have:

$$\mathbb{V}\left[\hat{\nabla}_{\theta,\phi}\mathcal{L}^{\text{OIWAE}}(\mathbf{x})\right] = \mathbb{E}_{\boldsymbol{\epsilon}^{(1:S)}\sim p(\boldsymbol{\epsilon})}\left[\sum_{s=1}^{S}\left(\tilde{u}_{\theta,\phi,\tau_i}^{(s)}\nabla_{\theta,\phi}\log\left(u_{\theta,\phi,\tau_i}^{(s)}\right)\right)^2\right] - \frac{1}{S}\left(\nabla_{\theta,\phi}\mathcal{L}^{\text{OIWAE}}(\mathbf{x})\right)^2 \tag{39}$$

The second term is independent of τ_i, so the gradient of the variance can be estimated as:

$$\hat{\nabla}_{\tau_i} \mathbb{V}\left[\hat{\nabla}_{\theta,\phi}\mathcal{L}^{\mathrm{OIWAE}}(\mathbf{x})\right] = \sum_{s=1}^{S} \nabla_{\tau_i}\left(\tilde{u}^{(s)}_{\theta,\phi,\tau_i} \nabla_{\theta,\phi} \times \log\left(u^{(s)}_{\theta,\phi,\tau_i}\right)\right)^2 \quad (40)$$

A single sample is used for the expectation. Note that we use the same set of S samples that were used to compute $\hat{\nabla}_{\theta,\phi}\mathcal{L}^{\mathrm{OIWAE}}(\mathbf{x})$ to compute $\hat{\nabla}_{\tau_i}\mathbb{V}\left[\hat{\nabla}_{\theta,\phi}\mathcal{L}^{\mathrm{OIWAE}}(\mathbf{x})\right]$.

V. Experiments

We compare the performance of the OVAE against the VAE, and of the OIWAE against the IWAE, in terms of their log likelihood lower bounds on held out test sets of the MNIST [12] and Frey Faces[2] (FF) datasets.

- **MNIST**: A dataset of images of handwritten digits, where the observations are binarised 28×28 images. We used the standard training set of 60,000 images, and evaluated the ELBO on the test set of 10,000 images. Note that we use the same binarisation as was used in [5].
- **Frey Faces**: A dataset of images of Brendan Frey, where the observations are grayscale 28×20 images. This is a considerably smaller dataset, of 1,965 total images, which we randomly split into a training set of 1,572 images and a test set of 393 images.

A. Design

The latent dimensionalities used are:

$$\mathbf{h}^{MNIST} \in \mathbb{R}^{50} \text{ and } \mathbf{h}^{FF} \in \mathbb{R}^{25}$$

This is because the Frey Faces dataset is significantly smaller, and therefore presents a higher risk of overfitting.

1) Generative model: The prior used, for both datasets, is:

$$p(\mathbf{h}) = \mathcal{N}(\mathbf{0}, \mathbf{I}) \quad (41)$$

The observation probabilities, given the value of the latent variables, are:

$$p_\theta^{MNIST}(\mathbf{x}|\mathbf{h}) = \mathrm{Bern}(\boldsymbol{\pi}_\theta(\mathbf{h})) \quad (42)$$

$$p_\theta^{FF}(\mathbf{x}|\mathbf{h}) = \mathcal{N}(\boldsymbol{\mu}_\theta(\mathbf{h}), \boldsymbol{\Sigma}_\theta(\mathbf{h})) \quad (43)$$

The parameters of the Bernoulli and Gaussian distributions are the output layers of neural networks, which take the latent vector $\mathbf{h}$ as their input. Note that $\boldsymbol{\Sigma}_\theta(\mathbf{h})$ is a diagonal matrix. The generative parameters, θ, are therefore the weights and biases of these networks.

[2]http://www.cs.nyu.edu/∼roweis/data.html

		Hidden layers	Output nonlinearity
MNIST	$\boldsymbol{\pi}_\theta(\mathbf{h})$	2×200 units	sigmoid
	$\boldsymbol{\eta}_\phi^{MNIST}(\mathbf{x})$	2×200 units	linear
	$\boldsymbol{\Omega}_\phi^{MNIST}(\mathbf{x})$	2×200 units	exp
FF	$\boldsymbol{\mu}_\theta(\mathbf{h})$	2×100 units	sigmoid
	$\boldsymbol{\Sigma}_\theta(\mathbf{h})$	2×100 units	exp
	$\boldsymbol{\eta}_\phi^{FF}(\mathbf{x})$	2×100 units	linear
	$\boldsymbol{\Omega}_\phi^{FF}(\mathbf{x})$	2×100 units	exp

TABLE I
THE NEURAL NETWORK ARCHITECTURES USED IN THE EXPERIMENTS.

2) Variational distribution: The variational distribution, for both datasets, is:

$$q_\phi(\mathbf{h}|\mathbf{x}) = \mathcal{N}(\boldsymbol{\eta}_\phi(\mathbf{x}), \boldsymbol{\Omega}_\phi(\mathbf{x})) \quad (44)$$

The parameters of this Gaussian are again the output layers of neural networks, this time taking the observation vector $\mathbf{x}$ as their input. Note that $\boldsymbol{\Omega}_\phi(\mathbf{x})$ is a diagonal matrix. The variational parameters ϕ are the weights and biases of these networks.

We used the same network architectures as [5]; they are described below:

3) Overdispersed proposal distribution: The overdispersed proposal distribution, for both datasets, is:

$$r_{\phi,\tau_i}(h_i|\mathbf{x}) = \mathcal{N}(\boldsymbol{\eta}_\phi(\mathbf{x})_i, \tau_i \times \boldsymbol{\Omega}_\phi(\mathbf{x})_{ii}) \quad (45)$$

where $\boldsymbol{\eta}_\phi(\mathbf{x})_i$ denotes the i^{th} element of the mean vector, and $\boldsymbol{\Omega}_\phi(\mathbf{x})_{ii}$ denotes the ii^{th} element of the (diagonal) covariance matrix, of the variational distribution. $\tau_i \geq 1$ is the i^{th} dispersion parameter.

4) Training: For a fair comparison, we use the same number of samples (5) for each of the four methods (VAE, OVAE, IWAE, OIWAE), where a single sample is used for the inner expectation of the OVAE, as per equation (27).

All of the weights of the networks were initialised using the heuristic of [13] and the biases were initialised as 0. For the OVAE and OIWAE, the dispersion parameters were all initialised as 2. The parameters were optimised using Adam [14], with parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, and a learning rate of 10^{-4}. We used minibatches of size 20, and drew 5 samples for each data point.

B. Results

In the table below, we report the log likelihood lower bounds achieved by each of the algorithms, on both datasets, after training is completed. Note that these lower bounds are measured using 5,000 samples for each data point in the test set, as per [5]. The values in the table below are computed using the regular VAE and IWAE bounds, i.e. samples are not drawn from the overdispersed proposal; overdispersion is used only when computing the gradients during training.

	VAE	OVAE	IWAE	OIWAE
MNIST	-86.47[3]	-85.37	-85.54[3]	-84.70
Frey Faces	1147.01	1177.11	1213.85	1271.29

TABLE II
THE TEST SET ELBOS ACHIEVED, AFTER TRAINING IS COMPLETED.

For both datasets, the OVAE outperforms the VAE, and the OIWAE outperforms the IWAE, in terms of achieving a higher log likelihood lower bound. Interestingly, none of the four algorithms appeared to be prone to overfitting, even on the smaller Frey Faces dataset, evidenced by the training ELBOs not being significantly higher than the test set ELBOs.

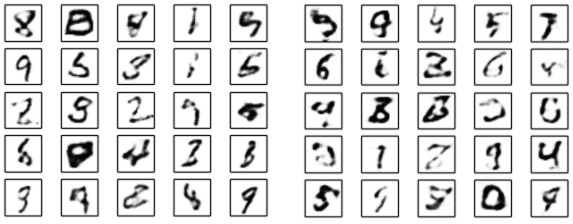
1) Generated output: We generate sample outputs from both the prior and posterior for both datasets. To generate output from the prior, we draw a sample of the latent $\mathbf{h}^{(s)} \sim p(\mathbf{h})$, and then draw an observation sample $\mathbf{x}^{(s)} \sim p_\theta(\mathbf{x}|\mathbf{h}^{(s)})$. To generate output from the posterior of a given image $\tilde{\mathbf{x}}$, we draw a sample of the latent $\mathbf{h}^{(s)} \sim q_\phi(\mathbf{h}|\tilde{\mathbf{x}})$, and then draw an observation sample $\mathbf{x}^{(s)} \sim p_\theta(\mathbf{x}|\mathbf{h}^{(s)})$.

Below are shown samples from both the priors and posteriors learned using the OVAE and OIWAE. To generate output from the prior, we draw a sample of the latent $\mathbf{h}^{(s)} \sim p(\mathbf{h})$, and then draw an observation sample $\mathbf{x}^{(s)} \sim p_\theta(\mathbf{x}|\mathbf{h}^{(s)})$. To generate output from the posterior of a given image $\tilde{\mathbf{x}}$, we draw a sample of the latent $\mathbf{h}^{(s)} \sim q_\phi(\mathbf{h}|\tilde{\mathbf{x}})$, and then draw an observation sample $\mathbf{x}^{(s)} \sim p_\theta(\mathbf{x}|\mathbf{h}^{(s)})$.

MNIST:

Prior

OVAE OIWAE



Posterior

OVAE OIWAE

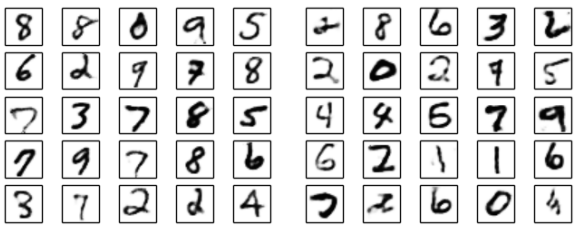


TABLE III
PRIOR AND POSTERIOR SAMPLES OF MNIST DIGITS

[3]The test set ELBOs for MNIST, trained using the VAE and IWAE, are taken from [5]

Frey Faces:

Prior

OVAE OIWAE

Posterior

OVAE OIWAE

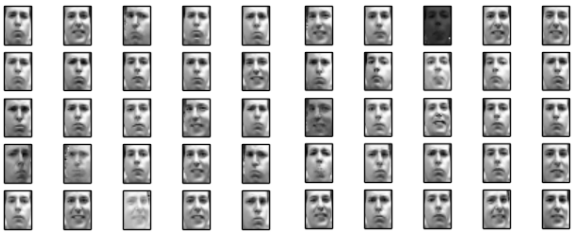


TABLE IV
PRIOR AND POSTERIOR SAMPLES OF FREY FACES

2) Variance of gradient estimates: Below, we show the sample variances of the gradient estimates during the first 1,000 iterations of training. It is clear to see that, for both datasets, the OVAE updates, for the first 1000 training iterations, are indeed of lower variance than are those for the VAE. This is less so the case when comparing the OIWAE to the IWAE, although it still holds true for the first 300 iterations on MNIST. This is consistent with the hypothesis that when the variational distribution is a poor fit to the true posterior (which is especially likely to be the case in the early training iterations), the overdispersed proposal should produce lower variance updates than using the variational distribution.

MNIST:

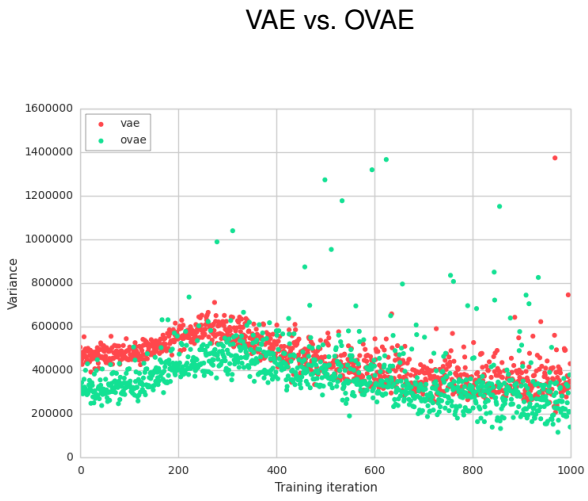


Fig. 2. Variances of gradient estimates, training the VAE and OVAE on MNIST

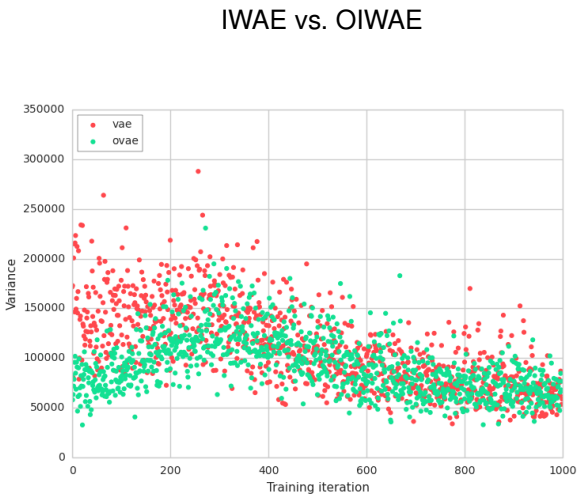


Fig. 3. Variances of gradient estimates, training the IWAE and OIWAE on MNIST

Frey Faces:

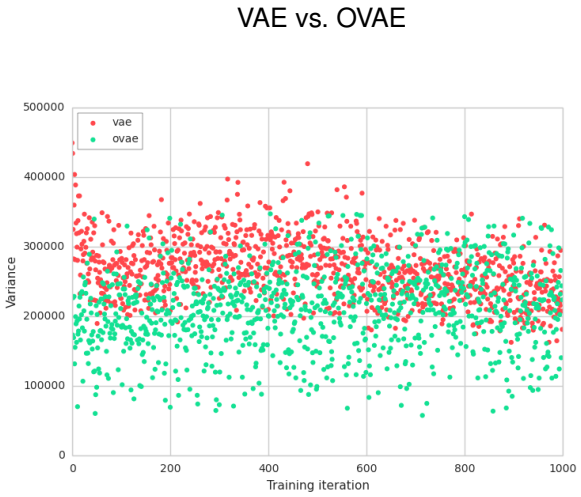


Fig. 4. Variances of gradient estimates, training the VAE and OVAE on Frey Faces

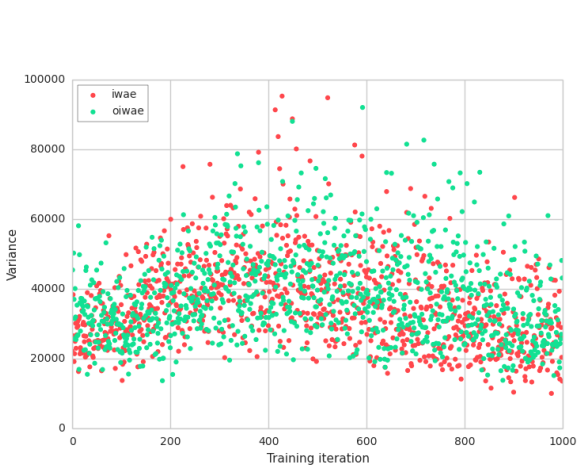


Fig. 5. Variances of gradient estimates, training the IWAE and OIWAE on Frey Faces

VI. Discussion

We have presented the OVAE and OIWAE, which are variants on the VAE and IWAE respectively. They use importance sampling to reduce the variance of the estimated gradients. While the variance-minimising proposal distribution is intractable, we explain why drawing from a proposal distribution which is heavier-tailed than the variational distribution is effective. The generative and variational parameters are optimised to maximise the lower bound on the log likelihood, and the dispersion parameters are optimised to minimise the variance of the estimated gradients. We have evaluated the performance of these methods on two datasets; in both cases they do indeed provide a lower variance gradient estimate than their counterparts, and reach a higher log likelihood lower bound.

There have recently been several developments to allow for the use of more expressive variational distributions than just those to which the reparametrisation trick can be easily applied. One such example would be the introduction of an auxiliary latent variable, which does not change the generative model (ADGM) [15]. One avenue for future work, therefore, could be to apply the overdispersion technique to the ADGM, in order to provide lower variance estimates of the gradients. There has also been research into applying stochastic backpropagation directly to mixture distributions [16], which would allow for the use of a mixture as the variational distribution. This could be combined with the OVAE and OIWAE to produce an autoencoder that captures complex posteriors, with low variance gradient estimates. A third avenue for future research would be to explore techniques such as control variates [17] and Rao-Blackwellization [18] to assess if the variance of the OVAE and OIWAE updates can be further reduced.

References

[1] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1–38, 1977.

[2] R. Neal and G. E. Hinton, "A view of the em algorithm that justifies incremental, sparse, and other variants," in *Learning in Graphical Models*. Kluwer Academic Publishers, 1998, pp. 355–368.

[3] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *CoRR*, vol. abs/1312.6114, 2013. [Online]. Available: http://arxiv.org/abs/1312.6114

[4] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, ser. JMLR Workshop and Conference Proceedings, vol. 32. JMLR.org, 2014, pp. 1278–1286. [Online]. Available: http://jmlr.org/proceedings/papers/v32/rezende14.html

[5] Y. Burda, R. B. Grosse, and R. Salakhutdinov, "Importance weighted autoencoders," *CoRR*, vol. abs/1509.00519, 2015. [Online]. Available: http://arxiv.org/abs/1509.00519

[6] F. J. R. Ruiz, M. K. Titsias, and D. M. Blei, "Overdispersed black-box variational inference," in *Uncertainty in Artificial Intelligence*, 2016.

[7] P. Dayan, G. E. Hinton, R. N. Neal, and R. S. Zemel, "The Helmholtz machine," *Neural Computation*, vol. 7, pp. 889–904, 1995.

[8] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951. [Online]. Available: http://dx.doi.org/10.1214/aoms/1177729694

[9] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[10] A. B. Owen, *Monte Carlo Theory, Methods and Examples*. To be published.

[11] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods (Springer Texts in Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005.

[12] Y. LeCun and C. Cortes, "MNIST handwritten digit database," 2010. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[13] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, May 13-15, 2010*, ser. JMLR Proceedings, Y. W. Teh and D. M. Titterington, Eds., vol. 9. JMLR.org, 2010, pp. 249–256. [Online]. Available: http://www.jmlr.org/proceedings/papers/v9/glorot10a.html

[14] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980

[15] L. Maaløe, C. K. Sønderby, S. K. Sønderby, and O. Winther, "Auxiliary deep generative models," in *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, ser. JMLR Workshop and Conference Proceedings, M. Balcan and K. Q. Weinberger, Eds., vol. 48. JMLR.org, 2016, pp. 1445–1453. [Online]. Available: http://jmlr.org/proceedings/papers/v48/maaloe16.html

[16] A. Graves, "Stochastic backpropagation through mixture density distributions," *CoRR*, vol. abs/1607.05690, 2016. [Online]. Available: http://arxiv.org/abs/1607.05690

[17] S. Ross, *Simulation*. Academic Press, 2002. [Online]. Available: https://books.google.co.uk/books?id=DApvQgAACAAJ

[18] G. Casella and C. Robert, *Rao-Blackwellization of Sampling Schemes*, ser. Documents de travail du CREST. INSEE, 1994. [Online]. Available: https://books.google.co.uk/books?id=AqWKPAAACAAJ