Variational Inference using Normalizing Flows

Pierre Segonne[1]

***Abstract*— Variational Inference aims to approximate posterior probability distributions. Several approaches have been developed over the years; originally restrained to problems for which variational inference led to closed form approximations, the method was then generalized with the less accurate but simpler use of pre-specificed families of posterior approximations. In the latter case, the inference performance is limited by the complexity of the specified family. The use of normalizing flows, which constructs versatile and arbitrarily complex approximate posterior distributions by transforming a simple initial distribution through the application of successive invertible mappings, constitutes an efficient, scalable and flexible variational inference technique, as will be demonstrated in this study. All code used throughout this project is gathered in the following repository: https://github.com/pierresegonne/VINF.**

I. INTRODUCTION

Probability theory is key to quantify uncertainty in any phenomenon and is therefore essential to the development of thinking machines. Indeed, any machine, or model, which needs to generate representations from given observations in order to accomplish its mission requires the formation and the numerical evaluation of beliefs; probability theory provides well defined rules to induce and manipulate such beliefs.

Furthermore, probabilistic modelling embeds directly probability theory into machine learning models. Probabilistic models describe how the interactions between observed variables, latent (in the sense of unobserved) variables and weights can predict target variables, while considering all variables as random variables with their associated probabilities.

To demonstrate interesting properties of probabilistic models, let's consider a probabilistic linear regression.

In probabilistic graphical models, as exposed in Figure 1, edges provide information about the conditional independence of the random variables they link. Specifically, any two nodes are conditionally independent given the value of their parents in the graph. This yields, for the previous model that:

$$p(y, x, \omega) = p(y|x, \omega)p(\omega) \tag{1}$$

The probability $p(y|x, \omega)$ is called the **likelihood** of the target data and $p(\omega)$ the **prior** probability of the parameter ω. The application of Bayes' Theorem immediately yields an expression for $p(\omega|y, x)$, the **posterior** probability:

[1]MSc Student, DTU (Denmarks Tekniske Universitet), Email: s182172@student.dtu.dk

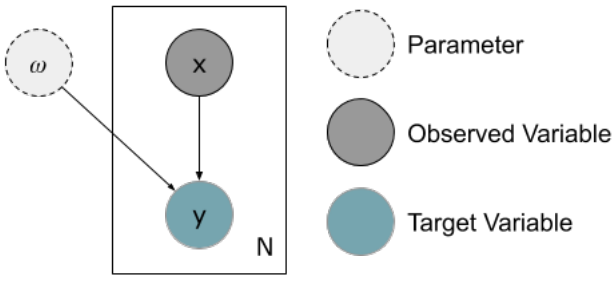

Fig. 1: Probabilistic Graphical Model for a Probabilistic Linear Regression

$$p(\omega|y, x) = \frac{p(y|\omega, x)p(\omega|x)}{p(y|x)} = \frac{p(y|\omega, x)p(\omega)}{p(y|x)} \tag{2}$$

And thus provides a direct way to train the model. It can first be noted that the denominator of Equation 2, the normalizing constant, is independent of ω, and therefore can be ignored to find the optimal value of ω, ω^*

$$\omega^* = \max_{\omega} p(\omega|y, x) = \max_{\omega} p(y|\omega, x)p(\omega) \tag{3}$$

It thus appears that in this model, the knowledge of the posterior distribution is crucial to determining optimal parameters for the model. In this simple example, it is possible to postulate reasonable probability distributions to express the likelihood and the prior, thus enabling the derivation of optimal weights ω^* (See Appendix A, VIII). The estimate ω^* would here be called the maximum a posteriori (MAP) estimate.

Beyond this example, which demonstrates that the knowledge of the posterior distribution is indeed paramount when working with Bayesian models, there are many cases for which having an expression of the entire posterior probability distribution is desired. For example, in the case of the linear regression, having knowledge of the full posterior distribution facilitates the estimation of the marginal likelihood of a new observation. For a trained model, a new observation $\hat{x}$ and a potential prediction $\hat{y}$, the marginal likelihood is

$$\begin{aligned} p(\hat{y}|\hat{x}, x, y) &= \int p(\hat{y}, \omega|\hat{x}, x, y)d\omega \\ &= \int p(\hat{y}|\hat{x}, \omega)p(\omega|x, y)d\omega \end{aligned} \tag{4}$$

And thus, having both an analytical expression for the likelihood $p(\hat{y}|\hat{x}, \omega)$ and the posterior $p(\omega|x, y)$ would allow to compute the marginal likelihood of new datapoints.

Additionally, the knowledge of the entire posterior distribution allows for more robust and varied applications. In this case, and notably because the **evidence** $p(y|x)$ is generally not known in closed form and takes exponential time to compute, it is required to result to an approximation of the posterior distribution. Using observed data to approximate the posterior distribution is a form of inference.

A first approach to approximating the posterior distribution relies on stochastic processes that, given an infinite amount of computing resources, would produce an exact result. The approximation then arises from the finitude of the resources available to execute such method. Markov chain Monte Carlo (MCMC) is one of the most common technique employed with that regard. It relies on the use of an ergodic Markov Chain whose stationary distribution is the posterior distribution, and from which samples are drawn to construct an empirical estimate of the end distribution. Even though advances such as the Metropolis-Hastings algorithm ([1], [2]) or the Gibbs sampler ([3]) greatly increased the practical usability of the method, it is still too slow for many real world applications. See [4] for more details.

The second approach, that will be used throughout this study, is called Variational Inference. It relies on the use of optimisation techniques to determine the most adequate distribution q^* from a family of probability distributions $\mathcal{Q}$, in the sense that it provides the best approximation of the true probability distribution p. The Kullback-Leibler (KL) divergence[1] provides an effective measure of the closeness of two probability densities [5] and can hence be used to determine q^*. Considering a set of observed variables $\boldsymbol{x} = x_{1:N}$ and the latent variables $\boldsymbol{z} = z_{1:M}$[2]

$$q^*(\boldsymbol{z}) = \arg\min_{q \in \mathcal{Q}} \mathrm{KL}(q(\boldsymbol{z})||p(\boldsymbol{z}|\boldsymbol{x})) \tag{5}$$

The previous definition reveals an important property of Variational Inference, the quality of the approximation of the true posterior is constrained by the family of distributions considered $\mathcal{Q}$.

For example, in Figure 2 is shown in red an arbitrary posterior distribution. If the family $\mathcal{Q}$ is restricted to Gaussian distributions, the obtained approximation will be similar to the q distribution displayed in blue. The family of Gaussian distributions being unimodal, restricting the family of possible distributions to it cannot result in a very close approximation.

[1]The KL divergence is defined, for two continuous densities p and q as $KL(p||q) = \int_{-\infty}^{\infty} \log(\frac{p(z)}{q(z)})p(z)dz$, for demonstration of its positiveness, see Appendix B (VIII)

[2]In the previous example, ω would constitute a latent variable, as it is not a deterministic parameter.

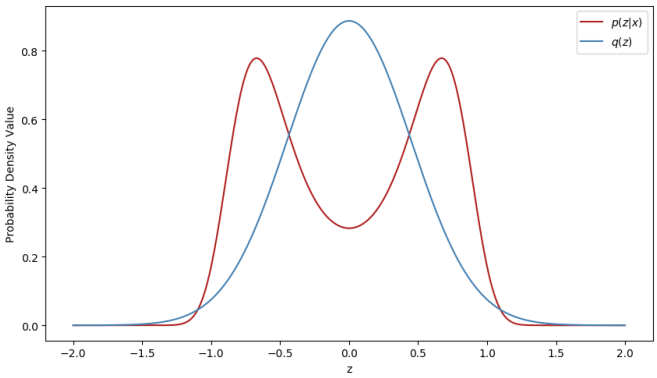

Fig. 2: Demonstration of the restriction imposed by the distribution family $\mathcal{Q}$ on the approximation of the posterior $p(\boldsymbol{z}|\boldsymbol{x})$

Furthermore, it is generally not possible to directly use the KL divergence to determine the optimal approximate distribution q^*, indeed

$$\begin{aligned} \mathrm{KL}(q(\boldsymbol{z})||p(\boldsymbol{z}|\boldsymbol{x})) &= \mathop{\mathbb{E}}_{q}[\log(q(\boldsymbol{z}))] - \mathop{\mathbb{E}}_{q}[\log(p(\boldsymbol{z}|\boldsymbol{x}))] \\ &= \mathop{\mathbb{E}}_{q}[\log(q(\boldsymbol{z}))] - \mathop{\mathbb{E}}_{q}[\log(p(\boldsymbol{z}, \boldsymbol{x}))] \\ &\quad + \log(p(\boldsymbol{x})) \end{aligned} \tag{6}$$

Which explicitly shows the dependence of the KL divergence on the evidence $p(\boldsymbol{x})$, that is generally unknown. The evidence is nevertheless a constant with respect to the latent variables $\boldsymbol{z}$. By denoting the evidence lower bound (ELBO) function as

$$\mathrm{ELBO}(q) = -\mathop{\mathbb{E}}_{q}[\log(q(\boldsymbol{z}))] + \mathop{\mathbb{E}}_{q}[\log(p(\boldsymbol{z}, \boldsymbol{x}))] \tag{7}$$

Then Equation 6 can be re-written as

$$\begin{aligned} \log(p(\boldsymbol{x})) &= \mathrm{KL}(q(\boldsymbol{z})||p(\boldsymbol{z}|\boldsymbol{x})) + \mathrm{ELBO}(q) \\ &\geq \mathrm{ELBO}(q) \end{aligned} \tag{8}$$

Where the bottom inequality is induced by the positiveness of the KL divergence (See Appendix B, VIII for a demonstration, it can also be shown using Jensen's inequality as in [6]), thus explaining the name of the ELBO function. Even though it has not been validated in theory, using the ELBO as an approximation of the evidence, or **marginal likelihood** works in practice for approximate inference [7], and as suggested by Equation 8, maximising the ELBO is equivalent in the end to minimising the KL divergence, which is the ultimate goal of inference.

It is worth noting that in the case where $\mathcal{Q}$ contains the true posterior distribution, $p(\boldsymbol{z}|\boldsymbol{x})$, then the KL divergence will be minimised by adopting it as the optimal approximate posterior distribution $q^*(\boldsymbol{z}) = p(\boldsymbol{z}|\boldsymbol{x})$. In that case, and if the model contains deterministic model parameters, it is possible to use the ELBO to gradually converge towards the true posterior distribution, as is accomplished with the EM

algorithm [8]. In this study, parameters will nevertheless be included in the latent space as random variables, and the ELBO used for optimisation.

Overall, Variational Inference is now often used as the preferred method to train probabilistic models for complex problems or large dataset, notably due to its flexibility and its computational efficiency[3]. Common applications are amongst others, semi-supervised classification [9], generative models for images [10] and speech synthesis [11].

This study aims at demonstrating key concepts of Variational Inference, and detailing practical considerations about the implementation of Normalizing Flows, with the hope that such technique can drastically improve the performance and usability of Variational Inference methods. To do so, it will first present how to practically use Variational Inference on a demonstration example, with a technique called Mean Field approximation. The limitations of this technique will then be exposed and the concept of Normalizing Flows will be detailed as an attempt to alleviate them. Finally results of experiments aiming at evaluating the practical usability of Normalizing Flows will be presented and will support an eventual discussion on Variational Inference using Normalizing Flows.

II. Mean Field Approximation

As seen in Introduction, the choice of the family of distributions considered $\mathcal{Q}$ to approximate the true posterior distribution is crucial to the execution of Variational Inference. For the mean field approximation, the family is chosen such that its elements respect

$$q(\boldsymbol{z}) = \prod_{i=1}^{m} q_i(\boldsymbol{z}_i) \tag{9}$$

Where the respective $\boldsymbol{z}_i$ represent a partition of the latent variables $\boldsymbol{z}$ into disjoint groups.

A. General Considerations

Noting $q_i(\boldsymbol{z}_i)$ as q_i to simplify notations, the ELBO function becomes, for continuous random variables

$$\text{ELBO}(q) = \int \prod_i q_i [\log(p(\boldsymbol{z}, \boldsymbol{x})) - \sum_i \log(q_i)] d\boldsymbol{z} \tag{10}$$

Focusing on a single factor q_j, the following holds

$$\begin{aligned}
\text{ELBO}(q) &= \int q_j [\int \log(p(\boldsymbol{x}, \boldsymbol{z})) \prod_{i \neq j} q_i d\boldsymbol{z}_i] d\boldsymbol{z}_j \\
&\quad - \int q_j \log(q_j) d\boldsymbol{z}_j + \text{const} \\
&= \int q_j \log(\hat{p}(\boldsymbol{x}, \boldsymbol{z}_j)) d\boldsymbol{z}_j - \int q_j \log(q_j) d\boldsymbol{z}_j \\
&\quad + \text{const} \\
&= -\text{KL}(q_j(\boldsymbol{z}_j) || \hat{p}(\boldsymbol{x}, \boldsymbol{z}_j)) + \text{const}
\end{aligned} \tag{11}$$

Where $\hat{p}(\boldsymbol{x}, \boldsymbol{z}_j)$ is defined by

$$\begin{aligned}
\log(\hat{p}(\boldsymbol{x}, \boldsymbol{z}_j)) &= \mathop{\mathbb{E}}_{i \neq j} [\log(p(\boldsymbol{x}, \boldsymbol{z}))] + \text{const} \\
&= \int \log(p(\boldsymbol{x}, \boldsymbol{z})) \prod_{i \neq j} q_i d\boldsymbol{z}_i + \text{const}
\end{aligned} \tag{12}$$

Equation 11 shows that it is possible to maximise the lower bound by minimising the KL divergence between $q_j(\boldsymbol{z}_j)$ and $\hat{p}(\boldsymbol{x}, \boldsymbol{z}_j)$, i.e by taking $q_j(\boldsymbol{z}_j) = \hat{p}(\boldsymbol{x}, \boldsymbol{z}_j)$, finally resulting in

$$\log(q_j^*(\boldsymbol{z}_j)) = \mathop{\mathbb{E}}_{i \neq j} [\log(p(\boldsymbol{x}, \boldsymbol{z}))] + \text{const} \tag{13}$$

The constant here appearing as the normalising constant of the distribution q_j. The expression found in Equation 13 is very interesting, even though it does not provide an explicit form for the approximate distribution q, as the expression for the factor q_j depends on the expression of the other factors $\{q_i\}_{i \neq j}$, it does provide an iterative method to converge towards a good approximation of the true best possible approximation. Cycling through the factors and updating consequently the estimates does indeed prove to converge, as the lower bound is convex in each of the factors $\{q_i\}$ [12].

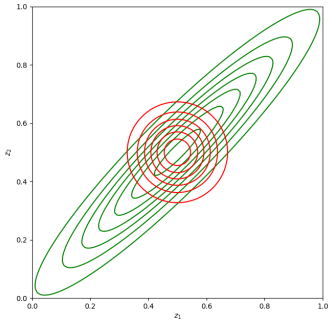

Fig. 3: Comparison between an arbitrary two dimensional Gaussian with full covariance as posterior distribution (red) and the approximated distribution obtained with the mean field method (red)

One of the drawbacks of the Mean Field approximation is its tendency to produce posterior distributions that are too compact, as displayed on Figure 3. In this case, that results from using $q(\boldsymbol{z}) = q(\boldsymbol{z}_1)q(\boldsymbol{z}_2)$ to approximate a 2D Gaussian posterior, the mean is correctly captured, but the covariance of both factors is controlled by the direction of smallest variance for the true posterior. This is due to the

[3] at least in comparison to MCMC

zero-avoidance behaviour of the KL divergence; it leads to distributions q that avoid regions where p is near zero.

B. Gaussian Mixture Model

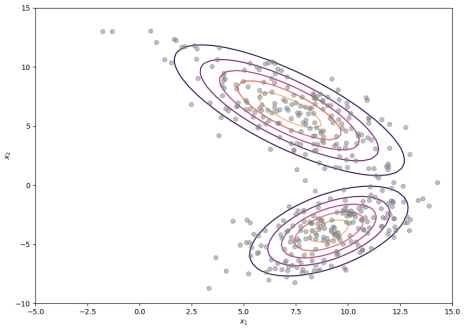

Fig. 4: Example of a 2D mixture of Gaussians

In the Gaussian mixture model, the observed dataset $\mathcal{D} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_N\}$ consists of random variables generated from a mixture of K gaussians of dimension d. Figure 4 provides a good example for a 2D Gaussian mixture with two components. The contour plots represent the probability densities of the respective components and the grey points are sampled from the resulting mixture. To each observed data $\boldsymbol{x}_i$ point is associated a latent variable $\boldsymbol{z}_i$ that is a 1-of-K binary vector indicating contributions from each Gaussian involved in the mixture, with the mixing coefficients $\boldsymbol{\pi}$.

The conditional distribution of $\boldsymbol{z}$ given the mixing coefficients is

$$p(\boldsymbol{z}|\boldsymbol{\pi}) = \prod_{n=1}^{N}\prod_{k=1}^{K}\pi_k^{z_{nk}} \tag{14}$$

Each of the components influence the likelihood of any data vector, and their conditional distribution, given the latent variables and underlying component parameters -μ_k for the mean of the kth Gaussian component and Λ_k for its precision matrix- can be expressed as

$$p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) = \prod_{n=1}^{N}\prod_{k=1}^{K}\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1})^{z_{nk}} \tag{15}$$

To simplify the following analysis, conjugate priors will be adopted for the parameters. A Dirichlet distribution will be assumed over the mixing coefficients

$$p(\boldsymbol{\pi}) = \text{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha}_0) = C(\boldsymbol{\alpha}_0)\prod_{k=1}^{K}\pi_k^{\alpha_0 - 1} \tag{16}$$

Where $C(\boldsymbol{\alpha}_0)$ is the normalising constant[4] for the distribution and the parameters $\boldsymbol{\alpha}_0 = \{\alpha_0, \ldots, \alpha_0\}$ are initially chosen all equal by symmetry.

The mean and precision parameters for each components will follow a Gaussian-Wishart distribution, as it represents the conjugate prior when both mean and precision are unknown [13],

$$\begin{aligned} p(\boldsymbol{\mu}, \boldsymbol{\Lambda}) &= p(\boldsymbol{\mu}|\boldsymbol{\Lambda})p(\boldsymbol{\Lambda}) \\ &= \prod_{k=1}^{K}\mathcal{N}(\boldsymbol{\mu}_k|\boldsymbol{m}_0, (\beta_0\boldsymbol{\Lambda}_k)^{-1})\mathcal{W}(\boldsymbol{\Lambda}_k|\boldsymbol{W}_0, \nu_0) \end{aligned} \tag{17}$$

Here again, the parameters of the distribution $\boldsymbol{m}_0$, β_0, $\boldsymbol{W}_0$ and ν_0 are initially chosen equal by symmetry. The interactions between the observed data, latent variables and parameters is exhibited in Figure 5, which represents the graphical model for the mixture of gaussians. In the model, $\boldsymbol{\pi}$ denotes $\{\pi_k\}$, $\boldsymbol{\mu}$ denotes $\{\boldsymbol{\mu}_k\}$ and $\boldsymbol{\Lambda}$, $\{\boldsymbol{\Lambda}_k\}$

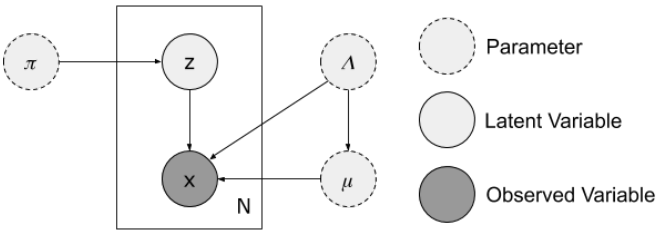

Fig. 5: Probabilistic Graphical Model for the Gaussian Mixture

The graphical model directly allows to split the joint probability into

$$p(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) = p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{\mu}, \boldsymbol{\Lambda})p(\boldsymbol{z}|\boldsymbol{\pi})p(\boldsymbol{\pi})p(\boldsymbol{\mu}|\boldsymbol{\Lambda})p(\boldsymbol{\Lambda}) \tag{18}$$

Let's now introduce the mean field approximation. By splitting up the latent variables from the parameters, the goal of the inference will become finding an optimal distribution q^* over $\boldsymbol{z}$, $\boldsymbol{\pi}$, $\boldsymbol{\mu}$, $\boldsymbol{\Lambda}$ satisfying

$$q(\boldsymbol{z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) = q(\boldsymbol{z})q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) \tag{19}$$

It was seen previously, through the general case study that the different subsets of factors can be iteratively updated. First, for the latent variables $\boldsymbol{z}$, applying Equation 13 results in

$$\log(q^*(\boldsymbol{z})) = \mathop{\mathbb{E}}_{\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Lambda}}[\log(p(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}))] + \text{const} \tag{20}$$

Which then, considering that in the decomposition of the joint probability distribution only $p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{\mu}, \boldsymbol{\Lambda})$ and $p(\boldsymbol{z}|\boldsymbol{\pi})$ depend on $\boldsymbol{z}$, becomes

$$\begin{aligned} \log(q^*(\boldsymbol{z})) = \mathop{\mathbb{E}}_{\boldsymbol{\pi}}[\log(p(\boldsymbol{z}|\boldsymbol{\pi}))] + \mathop{\mathbb{E}}_{\boldsymbol{\mu},\boldsymbol{\Lambda}}[\log(p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{\mu}, \boldsymbol{\Lambda}))] \\ + \text{const} \end{aligned} \tag{21}$$

And then because

[4] It is defined, for $\boldsymbol{\alpha} = \{\alpha_1, \ldots, \alpha_K\}$ by $C(\boldsymbol{\alpha}) = \frac{\Gamma(\sum_{k=1}^{K}\alpha_k)}{\prod_{k=1}^{K}\Gamma(\alpha_k)}$

$$\begin{aligned}\mathop{\mathbb{E}}_{\boldsymbol{\pi}}[\log(p(\boldsymbol{z}|\boldsymbol{\pi}))] &= \int \log(p(\boldsymbol{z}|\boldsymbol{\pi}))q(\boldsymbol{\pi})d\boldsymbol{\pi} \\ &= \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk}\int \log(\boldsymbol{\pi}_k))q(\boldsymbol{\pi})d\boldsymbol{\pi} \\ &= \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk}\mathbb{E}[\log(\pi_k)] \\ \mathop{\mathbb{E}}_{\boldsymbol{\mu},\boldsymbol{\Lambda}}[\log(p(\boldsymbol{x}|\boldsymbol{z},\boldsymbol{\mu},\boldsymbol{\Lambda}))] &= \frac{1}{2}\mathbb{E}[\log(|\boldsymbol{\Lambda}_k|)] - \frac{D}{2}\log(2\pi) \\ &\quad - \frac{1}{2}\mathop{\mathbb{E}}_{\boldsymbol{\mu}_k,\boldsymbol{\Lambda}_k}[(\boldsymbol{x}_n-\boldsymbol{\mu}_k)^T\boldsymbol{\Lambda}_k(\boldsymbol{x}_n-\boldsymbol{\mu}_k)]\end{aligned} \tag{22}$$

The log-probability of the approximate factor relative to the latent variables finally becomes

$$\log(q^*(\boldsymbol{z})) = \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk}\log(\rho_{nk}) + \text{const} \tag{23}$$

Where ρ_{nk} was introduced to simplify the notation. It is defined as

$$\begin{aligned}\log(\rho_{nk}) &= \mathbb{E}[\log(\pi_k)] + \frac{1}{2}\mathbb{E}[\log(|\boldsymbol{\Lambda}_k|)] - \frac{D}{2}\log(2\pi) \\ &\quad - \frac{1}{2}\mathop{\mathbb{E}}_{\boldsymbol{\mu}_k,\boldsymbol{\Lambda}_k}[(\boldsymbol{x}_n-\boldsymbol{\mu}_k)^T\boldsymbol{\Lambda}_k(\boldsymbol{x}_n-\boldsymbol{\mu}_k)]\end{aligned} \tag{24}$$

It is possible to derive an analytical expression for each component of this expression. The following considerations will apply for the general case, in the case where parameters of the prior distributions $\boldsymbol{\alpha}$, $\boldsymbol{m}_k$, β_k, $\boldsymbol{W}_k$ and ν_k might not be all equal anymore.

If $\boldsymbol{X} = \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_K\}$ follows a Dirichlet distribution, $p(\boldsymbol{X}) = Dir(\boldsymbol{X}|\boldsymbol{\alpha})$, then it holds that

$$\mathbb{E}[\log(\boldsymbol{X}_i)] = \psi(\alpha_i) - \psi(\sum_j \alpha_j) \tag{25}$$

Where ψ is the digamma function [14], therefore,

$$\mathbb{E}[\log(\pi_k)] = \psi(\alpha_k) - \psi(\sum_j \alpha_j) \tag{26}$$

Furthermore, if $\boldsymbol{M} \sim \mathcal{W}(\boldsymbol{V}, n)$, then it holds that

$$\mathbb{E}[\log(|\boldsymbol{M}|)] = \psi_p(\frac{n}{2}) + p\log(2) + \log(|\boldsymbol{V}|) \tag{27}$$

Where p is the dimension of the matrix $\boldsymbol{M}$, and where ψ_p is the multivariate digamma function [13], which is defined as [15]

$$\psi_p(a) = \frac{\partial \log(\Gamma_p(a))}{\partial a} = \sum_{i=1}^{p}\psi(a + \frac{1-i}{2}) \tag{28}$$

This eventually results in

$$\mathbb{E}[\log(|\boldsymbol{\Lambda}_k|)] = D\log(2) + \log(|\boldsymbol{W}_k|) + \sum_{i=1}^{D}\psi(\frac{\nu_k+1-i}{2}) \tag{29}$$

And finally, to determine an analytical expression for $\mathbb{E}_{\boldsymbol{\mu}_k,\boldsymbol{\Lambda}_k}[(\boldsymbol{x}_n-\boldsymbol{\mu}_k)^T\boldsymbol{\Lambda}_k(\boldsymbol{x}_n-\boldsymbol{\mu}_k)]$, it is first important to remind that in the general case, for $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{\Sigma} \in \mathbb{R}^{n\times n}$,

$$\boldsymbol{x}^T\boldsymbol{\Sigma}\boldsymbol{x} = \text{tr}(\boldsymbol{x}^T\boldsymbol{\Sigma}\boldsymbol{x}) = \text{tr}(\Sigma\boldsymbol{x}\boldsymbol{x}^T) \tag{30}$$

And thus

$$\begin{aligned}&\mathop{\mathbb{E}}_{\boldsymbol{\mu}_k,\boldsymbol{\Lambda}_k}[(\boldsymbol{x}_n-\boldsymbol{\mu}_k)^T\boldsymbol{\Lambda}_k(\boldsymbol{x}_n-\boldsymbol{\mu}_k)] = \\ &\int\int(\boldsymbol{x}_n-\boldsymbol{\mu}_k)^T\boldsymbol{\Lambda}_k(\boldsymbol{x}_n-\boldsymbol{\mu}_k)q(\boldsymbol{\mu}_k,\boldsymbol{\Lambda}_k)d\boldsymbol{\mu}_k d\boldsymbol{\Lambda}_k = \\ &\int\int \text{tr}(\boldsymbol{\Lambda}_k(\boldsymbol{x}_n-\boldsymbol{\mu}_k)(\boldsymbol{x}_n-\boldsymbol{\mu}_k)^T)q(\boldsymbol{\mu}_k|\boldsymbol{\Lambda}_k)q(\boldsymbol{\Lambda}_k)d\boldsymbol{\mu}_k d\boldsymbol{\Lambda}_k\end{aligned} \tag{31}$$

By definition, $\boldsymbol{\mu}_k|\boldsymbol{\Lambda}_k \sim \mathcal{N}(\boldsymbol{m}_k, (\beta_k\boldsymbol{\Lambda}_k)^{-1})$. Thus, $\mathbb{E}_{\boldsymbol{\mu}_k|\boldsymbol{\Lambda}_k}[\boldsymbol{\mu}_k] = \boldsymbol{m}_k$, and it also holds [16] (page 35) that $\mathbb{E}_{\boldsymbol{\mu}_k|\boldsymbol{\Lambda}_k}[\boldsymbol{\mu}_k\boldsymbol{\mu}_k^T] = \boldsymbol{m}_k\boldsymbol{m}_k^T + (\beta_k\boldsymbol{\Lambda}_k)^{-1}$, which can be combined to obtain

$$\begin{aligned}&\mathop{\mathbb{E}}_{\boldsymbol{\mu}_k|\boldsymbol{\Lambda}_k}[(\boldsymbol{x}_n-\boldsymbol{\mu}_k)(\boldsymbol{x}_n-\boldsymbol{\mu}_k)^T] = \\ &\mathop{\mathbb{E}}_{\boldsymbol{\mu}_k|\boldsymbol{\Lambda}_k}[\boldsymbol{x}_n\boldsymbol{x}_n^T - \boldsymbol{\mu}_k\boldsymbol{x}_n^T - \boldsymbol{x}_n\boldsymbol{\mu}_k^T + \boldsymbol{\mu}_k\boldsymbol{\mu}_k^T] = \\ &(\boldsymbol{x}_n-\boldsymbol{m}_k)(\boldsymbol{x}_n-\boldsymbol{m}_k)^T + (\beta_k\boldsymbol{\Lambda}_k)^{-1}\end{aligned} \tag{32}$$

Which can finally be re-arranged, thanks to elementary properties of the trace, and because $\mathbb{E}_{\boldsymbol{\Lambda}_k}[\boldsymbol{\Lambda}_k] = \nu_k\boldsymbol{W}_k$, into

$$\begin{aligned}&\mathop{\mathbb{E}}_{\boldsymbol{\mu}_k,\boldsymbol{\Lambda}_k}[(\boldsymbol{x}_n-\boldsymbol{\mu}_k)^T\boldsymbol{\Lambda}_k(\boldsymbol{x}_n-\boldsymbol{\mu}_k)] = \\ &\int \text{tr}(\boldsymbol{\Lambda}_k(\boldsymbol{x}_n-\boldsymbol{m}_k)(\boldsymbol{x}_n-\boldsymbol{m}_k)^T + (\beta_k\boldsymbol{\Lambda}_k)^{-1})q(\boldsymbol{\Lambda}_k)d\boldsymbol{\Lambda}_k = \\ &\frac{D}{\beta_k} + \int(\boldsymbol{x}_n-\boldsymbol{m}_k)^T\boldsymbol{\Lambda}_k(\boldsymbol{x}_n-\boldsymbol{m}_k)q(\boldsymbol{\Lambda}_k)d\boldsymbol{\Lambda}_k = \\ &\frac{D}{\beta_k} + \nu_k(\boldsymbol{x}_n-\boldsymbol{m}_k)^T\boldsymbol{W}_k(\boldsymbol{x}_n-\boldsymbol{m}_k)\end{aligned} \tag{33}$$

These considerations, despite their length and complexity confirm a very interesting property of the mean field approximation that, as was displayed in Equation 13; each factor, here the latent variables $\boldsymbol{z}$, can be updated iteratively based on the value of the other factors. Equation 23, put to the exponential becomes

$$\begin{aligned}q^*(\boldsymbol{z}) &\propto \prod_{n=1}^{N}\prod_{k=1}^{K}\rho_{nk}^{z_{nk}} \\ &= \prod_{n=1}^{N}\prod_{k=1}^{K} r_{nk}^{z_{nk}}\end{aligned} \tag{34}$$

Where the respective r_{nk} are the normalized counterparts to the ρ_{nk},

$$
\begin{aligned}
r_{nk} &= \frac{\rho_{nk}}{\sum_{j=1}^{K}\rho_{nj}} \\
\log(\rho_{nk}) &= \psi(\alpha_k) - \psi(\sum_{j=1}^{K}\alpha_j) - \frac{D}{2}\log(2\pi) \\
&\quad + \frac{1}{2}(D\log(2) + \log(|\boldsymbol{W}_k|) + \sum_{i=1}^{D}\psi(\frac{\nu_k+1-i}{2})) \\
&\quad - \frac{1}{2}(\frac{D}{\beta_k} + \nu_k(\boldsymbol{x}_n - \boldsymbol{m}_k)^T\boldsymbol{W}_k(\boldsymbol{x}_n - \boldsymbol{m}_k))
\end{aligned} \tag{35}
$$

The optimal solution $q^*(\boldsymbol{z})$ takes on the same form as the prior -which was chosen for that reason-, and it follows that

$$
\mathbb{E}[z_{nk}] = r_{nk} \tag{36}
$$

Suggesting consequently that the coefficients r_{nk} can be attributed the role of responsabilities, and therefore can be used, similarly as in the E step of the EM algorithm [17], to update the statistics of the observed dataset

$$
\begin{aligned}
N_k &= \sum_{n=1}^{N} r_{nk} \\
\overline{\boldsymbol{x}}_k &= \frac{1}{N_k}\sum_{n=1}^{N} r_{nk}\boldsymbol{x}_n \\
\boldsymbol{S}_k &= \frac{1}{N_k}\sum_{n=1}^{N} r_{nk}(\boldsymbol{x}_n - \overline{\boldsymbol{x}}_k)(\boldsymbol{x}_n - \overline{\boldsymbol{x}}_k)^T
\end{aligned} \tag{37}
$$

Similarly as what has just been accomplished, it is possible to derive an update rule for the parameters $\boldsymbol{\pi}$, $\boldsymbol{\mu}$ and $\boldsymbol{\Lambda}$. Applying Equation 13 again results in

$$
\log(q^*(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda})) = \mathop{\mathbb{E}}_{\boldsymbol{z}}[\log(p(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}))] + \text{const} \tag{38}
$$

Again, the decomposition of the joint probability distribution can be applied. The expectations of $p(\boldsymbol{\pi})$ and $p(\boldsymbol{\mu}, \boldsymbol{\Lambda})$ with regard to $\boldsymbol{z}$ are immediate to obtain, and because

$$
\begin{aligned}
\mathop{\mathbb{E}}_{\boldsymbol{z}}[\log(p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{\mu}, \boldsymbol{\Lambda}))] &= \sum_{n=1}^{N}\sum_{k=1}^{K}\log(\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k))\mathbb{E}[z_{nk}] \\
&= \sum_{n=1}^{N}\sum_{k=1}^{K}\log(\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k))r_{nk}
\end{aligned} \tag{39}
$$

The optimal factor can be re-expressed as

$$
\begin{aligned}
\log(q^*(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda})) &= \log(p(\boldsymbol{\pi})) + \sum_{k=1}^{K}\log(p(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k)) \\
&\quad + \mathbb{E}_{\boldsymbol{z}}[\log(p(\boldsymbol{z}|\boldsymbol{pi}))] \\
&\quad + \sum_{n=1}^{N}\sum_{k=1}^{K}\log(\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k))r_{nk} \\
&\quad + \text{const}
\end{aligned} \tag{40}
$$

The terms depending on $\boldsymbol{\pi}$ and $\{\boldsymbol{\mu}, \boldsymbol{\Lambda}\}$ are well split, suggesting that the paramter factor can further be decomposed into $q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) = q(\boldsymbol{\pi})q(\boldsymbol{\mu}, \boldsymbol{\Lambda}) = q(\boldsymbol{\pi})\prod_{k=1}^{K}q(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k)$

Isolating the terms related to $\boldsymbol{\pi}$ in Equation 40, results in

$$
\begin{aligned}
\log(q^*(\boldsymbol{\pi})) &= (\alpha_0 - 1)\sum_{k=1}^{K}\log(\pi_k) + \sum_{n=1}^{N}\sum_{k=1}^{K}r_{nk}\log(\pi_k) \\
&\quad + \text{const}
\end{aligned} \tag{41}
$$

Which directly indicates that $q^*(\boldsymbol{\pi})$ is a Dirichlet distribution over $\boldsymbol{\pi}$ with parameters

$$
\alpha_k = \alpha_0 + N_k \tag{42}
$$

This provides the first update rule for the parameter, which parallels the M step of the EM algorithm.

Furthermore, using the marginalisation of $\boldsymbol{\Lambda}_k$, $q(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k) = q(\boldsymbol{\mu}_k|\boldsymbol{\Lambda}_k)q(\boldsymbol{\Lambda}_k)$ and considering only the terms involving $\boldsymbol{\mu}_k$ and $\boldsymbol{\Lambda}_k$ in Equation 40 results in

$$
q^*(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k) = \mathcal{N}(\boldsymbol{\mu}_k|\boldsymbol{m}_k, (\beta_k\boldsymbol{\Lambda}_k)^{-1})\mathcal{W}(\boldsymbol{\Lambda}_k|\boldsymbol{W}_k, \nu_k) \tag{43}
$$

Where the parameters are updated with the following expressions

$$
\begin{aligned}
\beta_k &= \beta_0 + N_k \\
\boldsymbol{m}_k &= \frac{1}{\beta_k}(\beta_0\boldsymbol{m}_0 + N_k\overline{\boldsymbol{x}}_k) \\
\boldsymbol{W}_k^{-1} &= \boldsymbol{W}_0^{-1} + N_k\boldsymbol{S}_k + \frac{\beta_0 N_k}{\beta_0 + N_k}(\overline{\boldsymbol{x}}_k - \boldsymbol{m}_0)(\overline{\boldsymbol{x}}_k - \boldsymbol{m}_0)^T \\
\nu_k &= \nu_0 + N_k
\end{aligned} \tag{44}
$$

Equations 37, 42 and 44 thus provide, as expected, an iterative method to converge towards the optimal distribution $q^*(\boldsymbol{z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) = q^*(\boldsymbol{z})q^*(\boldsymbol{\pi})q^*(\boldsymbol{\mu}, \boldsymbol{\Lambda}) \simeq \hat{p}(\boldsymbol{x}, \boldsymbol{z})\hat{p}(\boldsymbol{x}, \boldsymbol{\pi})\hat{p}(\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{\Lambda})$ which can be implemented as is, and provide a direct way to attribute each point to the Gaussian component with the strongest influence, i,e the greatest responsibility $c_n = \arg\max_{k\in[1,K]} r_{nk}$. Figure 6 displays an example of clustering realised using this technique, yellow points are attributed to the top Gaussian, and purple points are attributed to the bottom Gaussian.

Furthermore, knowing the probability density function of the trained Gaussian mixture can be very useful. It notably allows the evaluation of the likelihood of new points $p(\hat{\boldsymbol{x}}, \boldsymbol{x})$. Similarly to what was exposed in Equation 4, the marginalisation with regard to the new latent variables $\hat{\boldsymbol{z}}$ and parameters $\boldsymbol{\pi}$, $\boldsymbol{\mu}$ and $\boldsymbol{\Lambda}$ results in

$$
\begin{aligned}
p(\hat{\boldsymbol{x}}|\boldsymbol{x}) &= \sum_{\hat{\boldsymbol{z}}}\int\int\int p(\hat{\boldsymbol{x}}, \hat{\boldsymbol{z}}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}|\boldsymbol{x})d\boldsymbol{\pi}d\boldsymbol{\mu}d\boldsymbol{\Lambda} \\
&= \sum_{\hat{\boldsymbol{z}}}\int\int\int p(\hat{\boldsymbol{x}}|\hat{\boldsymbol{z}}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda})p(\hat{\boldsymbol{z}}|\boldsymbol{\pi}) \\
&\qquad p(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}|\boldsymbol{x})d\boldsymbol{\pi}d\boldsymbol{\mu}d\boldsymbol{\Lambda}
\end{aligned} \tag{45}
$$

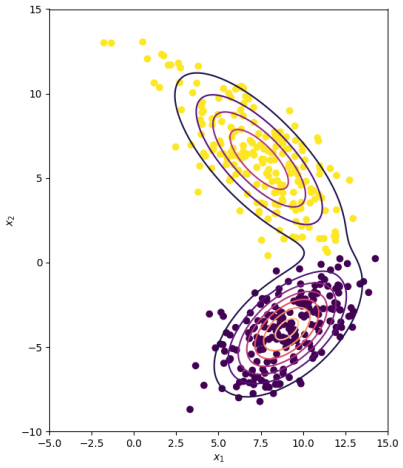

Fig. 6

The likelihood $p(\hat{\boldsymbol{x}}|\hat{\boldsymbol{z}}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda})$ is known, so is the conditional distribution for the latent variables $p(\hat{\boldsymbol{z}}|\boldsymbol{\pi})$, but the posterior distribution for the parameters is intractable, forcing the usage of the factorised approximation $q(\boldsymbol{\pi})q(\boldsymbol{\mu}, \boldsymbol{\Lambda})$

$$p(\hat{\boldsymbol{x}}|\boldsymbol{x}) = \sum_{k=1}^{K} \int\int\int \pi_k \mathcal{N}(\hat{\boldsymbol{x}}|\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1}) \\ q(\boldsymbol{\pi})q(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k) d\boldsymbol{\pi} d\boldsymbol{\mu}_k d\boldsymbol{\Lambda}_k \tag{46}$$

Which finally results in

$$p(\hat{\boldsymbol{x}}|\boldsymbol{x}) = \frac{\sum_{k=1}^{K} \alpha_k \mathrm{St}(\hat{\boldsymbol{x}}|\boldsymbol{m}_k, \boldsymbol{L}_k, \nu_k + 1 - D)}{\sum_{k=1}^{K} \alpha_k} \tag{47}$$

Where St represents the student distribution and

$$\boldsymbol{L}_k = \frac{(\nu_k + 1 - D)\beta_k}{(1 + \beta_k)} \boldsymbol{W}_k \tag{48}$$

In Figure 6 is also represented the contour plot associated with such probability density function. It corresponds well to what could be expected, as for a large number of data points N, the density function exposed in Equation 47 reduces to a mixture of Gaussians [13]; The two Gaussians components are clearly visible and are linked for continuity reasons.

The previous demonstration shows the ability to generate, from the simple assumption that the Mean Field technique is based on, expressive, reliable and accurate variational approximations of the true posterior distribution. It also shows very clearly that only relying on the factorisation of the approximate distribution, using acquired knowledge to determine the appropriate conjugate priors and relying on advanced mathematical skills to derive the correct update expressions and probability density function is cumbersome and not scalable to problems where the data does not seem to originate from a combination of known probability distributions. It is therefore not applicable in many real world cases, for which few properties about the distribution of data are known beforehand, such as in image or audio analysis for example.

III. Automatic Variational Inference

It is nevertheless possible to develop another technique, both conceptually simpler and more versatile, to obtain a variational approximation of the true posterior distribution. The evidence lower bound was defined in Equation 7 as

$$\mathrm{ELBO}(q) = -\mathop{\mathbb{E}}_{q}[\log(q(\boldsymbol{z}))] + \mathop{\mathbb{E}}_{q}[\log(p(\boldsymbol{z}, \boldsymbol{x}))] \tag{49}$$

Finding the approximate distribution q that approximates best the true posterior distribution by maximising the ELBO function cannot be realistically achieved in the general case; the expected values composing the ELBO being intractable for many choices of approximate distributions. Restricting the choice of the approximation q to distributions that can be parametrised by a set of parameters $\boldsymbol{\phi}$ is a first step towards simplifying this complex optimisation problem. Indeed, finding the most appropriate approximation becomes an optimisation problem for the parameters of the distribution q.

$$\begin{aligned} \boldsymbol{\phi}^* &= \arg\max_{\boldsymbol{\phi}} \mathrm{ELBO}(q_{\boldsymbol{\phi}}) \\ &= \arg\max_{\boldsymbol{\phi}} \mathop{\mathbb{E}}_{q_{\boldsymbol{\phi}}}[\log(p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x}))] - \mathop{\mathbb{E}}_{q_{\boldsymbol{\phi}}}[\log(q_{\boldsymbol{\phi}}(\boldsymbol{z}))] \end{aligned} \tag{50}$$

Where $q_{\boldsymbol{\phi}}$ designates the distribution q parametrised with $\boldsymbol{\phi}$ and similarly, $p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x})$, the joint probability parametrised by the model parameters $\boldsymbol{\theta}$. In the general case, finding an analytical expression for the optimal set of parameters $\boldsymbol{\phi}^*$ is impossible. Nevertheless, assuming the ELBO function is differentiable with respect to $\boldsymbol{\phi}$, then optimisation methods, such as gradient ascent [18] [19], can generate reasonable optima. If it can be expected that an analytical expression of $\mathbb{E}_{q_{\boldsymbol{\phi}}}[\log(q_{\boldsymbol{\phi}}(\boldsymbol{z}))]$ and therefore, of its gradient with regard to $\boldsymbol{\phi}$ can be determined, it is nevertheless not the case for the other term of the ELBO function, $\mathbb{E}_{q_{\boldsymbol{\phi}}}[\log(p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x}))]$. Fortunately, expectations are integrals, and **Monte Carlo integration** [4] can therefore be used to approximate them

$$\mathop{\mathbb{E}}_{q_{\boldsymbol{\phi}}}[\log(p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x}))] \simeq \frac{1}{S} \sum_{s=1}^{S} \log(p_{\boldsymbol{\theta}}(\boldsymbol{z}_s, \boldsymbol{x})) \\ \text{where } \boldsymbol{z}_s \sim q_{\boldsymbol{\phi}} \tag{51}$$

Such estimate is unbiased and its standard deviation is of order $\frac{1}{\sqrt{S}}$ where S is the number of samples used [20].

It thus becomes possible to obtain an approximate analytical expression for the gradient of the ELBO with regard to the approximation parameters $\nabla_{\boldsymbol{\phi}}\mathrm{ELBO}(q_{\boldsymbol{\phi}})$. Using modern automatic differentiation tools such as *tensorflow*[5],[6] [22] or *pytorch*[7] [23], it becomes

[5] https://www.tensorflow.org/

[6] *tensorflow* encloses a probabilistic module [21] that makes it naturally well suited to work with models using probability distributions. This is why *tensorflow* was used throughout this project.

[7] https://pytorch.org/

straightforward to perform the optimisation of the parameters of the approximated posterior distribution. In practice, using minibatches and stochastic gradient descent allows to scale this approach to very large datasets [24]. This form of approximation has been named *doubly-stochastic estimation* [25], because both the gradient estimation and the gradient descent use stochastic methods.

To simplify even further this method, one can choose to use a Gaussian parametrised, for which ϕ then becomes $\phi = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}$. If the covariance is chosen as diagonal, then the **reparametrisation** trick can be used. For each dimension involved, the following holds

$$z \sim \mathcal{N}(\mu, \sigma^2) \Leftrightarrow z = \mu + \sigma\epsilon \ , \ \epsilon \sim \mathcal{N}(0, 1) \tag{52}$$

And therefore, for a single dimension,

$$\mathop{\mathbb{E}}_{q_{\boldsymbol{\phi}}}[\log(p_{\boldsymbol{\theta}}(z, x))] \Leftrightarrow \mathop{\mathbb{E}}_{\mathcal{N}(\epsilon|0,1)}[\log(p_{\boldsymbol{\theta}}(\mu + \sigma\epsilon, x))] \tag{53}$$

Which can be extended to all dimensions, as a diagonal covariance matrix implies that all dimensions are independent and that the resulting distribution can be factorised accordingly. This is a special case of the Mean Field approximation. This provides an even easier way to generate an optimum for the parameters of the approximate posterior distribution and provides in practice a more stable convergence [26]. This method has been introduced at different occasions, under the names of Stochastic Backpropagation [27] and Stochastic Gradient Variational Bayes [10].

It is furthermore possible to use a neural network, called here an inference network, to represent the parameters of the approximation $q_{\boldsymbol{\phi}}$, thus generating $\boldsymbol{\mu}(\boldsymbol{x})$ and $\boldsymbol{\Sigma}(\boldsymbol{x})$ in the case where $q_{\boldsymbol{\phi}}$ is a Gaussian. This enhances the efficiency of the method, especially for problems with a great number of dimensions (images for example). Because the problems that will be considered here are mainly for demonstration purposes and have therefore a low number of dimensions, the addition of inference networks was ignored for the rest of the study. Adding an inference network to the previous method constitutes an approach coined as Amortised Variational Inference [26].

Figure 7 displays, in red, the contour plot of the learned approximated distribution $q_{\boldsymbol{\phi}}$ compared to the true posterior distribution. The family of parametrised distribution considered here consists of Gaussian distributions of dimension two, with a diagonal covariance matrix. The plot reveals that the approximation captures one of the component of the mixture but, as was already explained with Figure 3, is too compact compared to the true Gaussian component. It is normal that the approximation only captures one component, as no multi modal distribution is included in the considered parametric family of approximate distributions; it is also normal that the learned approximation does not

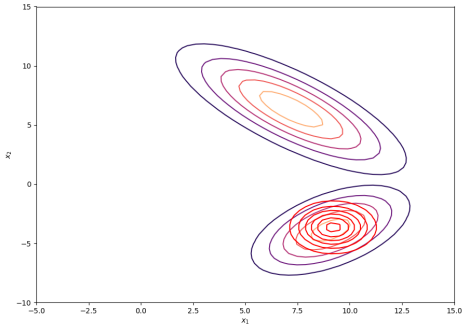

Fig. 7: Fitting of Automatic Variational Inference on the Gaussian mixture posterior

cover both components due to the zero-avoidance behaviour of the KL divergence that is implicitly minimised during the training process. Overall, this demonstrates that even though Automatic Variational Inference is a very useful tool due to its simplicity, the approximation that this approach produces is limited in its accuracy by the complexity of the parametric family chosen for the approximation, as was foreshadowed with Figure 2.

IV. Normalizing Flows

To rectify this limitation, it is nevertheless possible to transform initially simple distributions, such as the Gaussian, into arbitrarily complex approximate posterior distributions. With the theoretical assurance that Variational Inference does not induce overfitting [13], it can be expected that this transformation provides a serious increase in the performance of approximations learned through Automatic Variational Inference.

Normalizing Flows [28] [29] consists in a chain of invertible mappings that can be applied to transform an initial probability distribution into a vastly different probability distribution, through the repeated application of the change of variable. To demonstrate how such alteration is possible, let's consider an invertible smooth mapping $f : \mathbb{R}^d \to \mathbb{R}^d$, with its inverse being noted f^{-1}, applied onto a random variable $\boldsymbol{z}$ with distribution $q(\boldsymbol{z})$. The output $\boldsymbol{z'} = f(\boldsymbol{z})$ will also be a random variable. its distribution $q'(\boldsymbol{z'})$ can be expressed as a function of the original distribution

$$q'(\boldsymbol{z'}) = q(\boldsymbol{z})|\det\frac{\partial f^{-1}}{\partial \boldsymbol{z'}}| \tag{54}$$

To see why, let's study the simple 1D case. Under the change of variables, the probability contained in a differential area must be invariant. Thus

$$|q'(\boldsymbol{z'})d\boldsymbol{z'}| = |q(\boldsymbol{z})d\boldsymbol{z}| \tag{55}$$

Or

$$q'(\boldsymbol{z'}) = q(\boldsymbol{z})|\frac{d\boldsymbol{z}}{d\boldsymbol{z'}}| = q(\boldsymbol{z})|\frac{df^{-1}}{d\boldsymbol{z'}}(\boldsymbol{z'})| \tag{56}$$

Which can be extended to the multidimensional case already expressed in Equation 54 [30]. Furthermore, the inverse function theorem enounces that if the Jacobian of the inverse mapping f^{-1} is the inverse of the Jacobian of the mapping f, $\frac{\partial f^{-1}}{\partial \boldsymbol{z}'} = (\frac{\partial f}{\partial \boldsymbol{z}})^{-1}$, and therefore, because for any invertible matrix A, $\det(A^{-1}) = \det(A)^{-1}$, the resulting distribution can be re-expressed as

$$q'(\boldsymbol{z}') = q(\boldsymbol{z})|\det\frac{\partial f}{\partial \boldsymbol{z}}|^{-1} \tag{57}$$

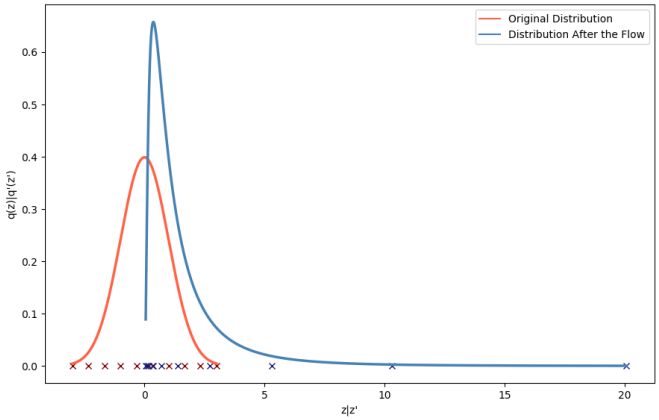

Fig. 8: Effect of an exponential flow on a standard Gaussian distribution

The exponential function is smooth and invertible. It can therefore be used as a legitimate normalizing flow. In this simple case, both the forward mapping $f = \exp$ and the inverse mapping $f^{-1} = \log$ are known and it is thus quite simple to display the effect of the flow on both the space of the random variables and on the associated probability distributions. The application of the exponential flow on a standard Gaussian $q(z) = \mathcal{N}(z|0,1)$ results in Figure 8. The red curve represents the original distribution, that is defined on the space spanned by the red crosses, which are samples for the random variable z, evenly spaced between -3 and 3. In blue is represented the resulting distribution after the flow

$$q'(z') = q(f^{-1}(z'))|\frac{df^{-1}}{dz'}(z')| = \mathcal{N}(\log(z')|0,1)|\frac{1}{z'}| \tag{58}$$

Which is logically a log-normal distribution. In blue are the respective images z' of the samples of the random variable z. This figure gives an intuitive understanding of the effect of a normalizing flow on a random variable and its associated probability distribution. By warping the space on which lie the considered random variables, the resulting probability distribution is affected. In the previous case, the warping would consist of taking the negative half-line of $\mathbb{R}$ and compacting it on the right side of zero, while dilating the positive half-line of $\mathbb{R}$. As a result, the density of the resulting random variables would greatly increase on the right side of zero, where space as been compacted, and gradually decrease when moving away from 0. This warping of space can then become arbitrarily more complex as the flow considered also gains in depth, or complexity. Figure 9 represents such warping in two dimensions, for more complex flows that will be detailed further in this study.

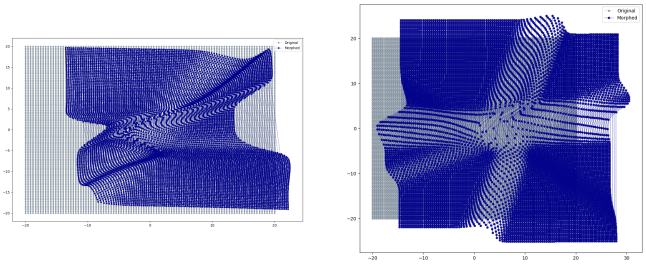

Fig. 9: Effect of a Normalizing Flow on a 2 dimensional meshgrid. The warping of space induced by the flow is very clear when looking at the image $\boldsymbol{z}_k$ (blue) of the original space points $\boldsymbol{z}_0$ (grey)

To generate more complex flows it is possible to chain several invertible smooth mappings $f_1, \dots, f_K$ such that

$$\begin{aligned} \boldsymbol{z}_K &= f_K \circ \dots \circ f_1(\boldsymbol{z}_0) \\ \log(q_K(\boldsymbol{z}_K)) &= \log(q_0(\boldsymbol{z}_0)) - \sum_{k=1}^{K} \log(|\det\frac{\partial f_k}{\partial \boldsymbol{z}_{k-1}}|) \end{aligned} \tag{59}$$

While defining smooth invertible mappings is pretty straightforward, computing the Jacobian determinant generally requires $\mathcal{O}(LD^3)$ operations [26] [31], where D is the dimension of the Jacobian and L the number of chained mappings, and is therefore not scalable.

It is nevertheless entirely possible to find mappings such that the Jacobian determinant terms can be computed in linear time $\mathcal{O}(LD)$.

A. Planar Flows

The first type of mapping considered is

$$f(\boldsymbol{z}) = \boldsymbol{z} + \boldsymbol{u}h(\boldsymbol{w}^T\boldsymbol{z} + b) \tag{60}$$

For which $\boldsymbol{u} \in \mathbb{R}^D$, $\boldsymbol{w} \in \mathbb{R}^D$ and $b \in \mathbb{R}$ are parameters and h a smooth element wise non-linearity [26]. This family of transformation respect the linear-time requirement for the computation of the Jacobian determinant. Indeed, by linearity of the derivative and because $\frac{d\boldsymbol{a}^T\boldsymbol{x}}{d\boldsymbol{x}} = \boldsymbol{a}$ ([16] page 10), the following holds

$$\begin{aligned} |\det\frac{\partial f}{\partial \boldsymbol{z}}| &= |\det(\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{z}} + \frac{\partial h(\boldsymbol{w}^T\boldsymbol{z} + b)}{\partial \boldsymbol{z}}\boldsymbol{u}^T)| \\ &= |\det(\boldsymbol{I} + h'(\boldsymbol{w}^T\boldsymbol{z} + b)\frac{\partial(\boldsymbol{w}^T\boldsymbol{z} + b)}{\partial \boldsymbol{z}}\boldsymbol{u}^T)| \\ &= |\det(\boldsymbol{I} + h'(\boldsymbol{w}^T\boldsymbol{z} + b)\boldsymbol{w}\boldsymbol{u}^T)| \end{aligned} \tag{61}$$

Considering the matrix determinant lemma, that states that for $\boldsymbol{A}$ an invertible matrix and $\boldsymbol{x}$ and $\boldsymbol{y}$ vectors, $\det(\boldsymbol{A} + \boldsymbol{x}\boldsymbol{y}^T) = (1 + \boldsymbol{y}^T\boldsymbol{A}^{-1}\boldsymbol{x})\det(\boldsymbol{A})$, and with the notation $\psi(\boldsymbol{z}) = h'(\boldsymbol{w}^T\boldsymbol{z} + b)\boldsymbol{w}$, it results that

$$|\det\frac{\partial f}{\partial \boldsymbol{z}}| = |(1+\boldsymbol{u}^T\psi(\boldsymbol{z}))\det(\boldsymbol{I})| = |1+\boldsymbol{u}^T\psi(\boldsymbol{z})| \quad (62)$$

Thus inserting the found expression for the Jacobian determinant into Equation 59, the resulting distribution after a planar flow will verify

$$\log(q_K(\boldsymbol{z}_K)) = \log(q_0(\boldsymbol{z}_0)) - \sum_{k=1}^{K}\log(|1+\boldsymbol{u}_k^T\psi_k(\boldsymbol{z}_{k-1})|) \quad (63)$$

In practice, the function $h(x) = \tanh(x)$ will be used, and it is important to note that not all sets of parameters make f invertible. A sufficient condition for the invertibility of f is $\boldsymbol{w}^T\boldsymbol{u} \geq -1$ [26]. This is enforced by considering, instead of $\boldsymbol{u}$ for each mapping the modified parameter

$$\hat{\boldsymbol{u}} = \boldsymbol{u} + (m(\boldsymbol{w}^T\boldsymbol{u}) - \boldsymbol{w}^T\boldsymbol{u})\frac{\boldsymbol{w}}{||\boldsymbol{w}||^2} \quad (64)$$

Where m is the function $m(x) = -1 + \log(1+e^x)$. Unfortunately, even though invertibility of the mapping is assured, it is impossible to derive an expression in the general case.

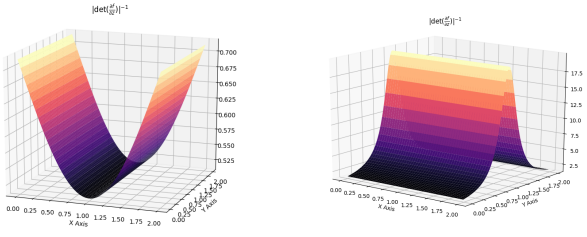

Fig. 10: Evaluation of $|\frac{\partial f}{\partial \boldsymbol{z}}|^{-1}$ for an expansion away from the line $x = 1$ (Left) and a contraction towards the line $y = 1$ (Right)

The name of Planar Flows is derived from its effect on probability distributions. Figure 10 displays the evolution of $|\frac{\partial f}{\partial \boldsymbol{z}}|^{-1}$ based on the value of $\boldsymbol{z}$ for two different sets of parameters that respectively produce the two possible effects of a planar flow on the probability space. Depending on the value of $\boldsymbol{w}$, $\boldsymbol{u}$ and b, the flow will either induce an expansion (left) or a contraction (right) of the initial density along the hyperplane $\boldsymbol{w}^T\boldsymbol{z}+b=0$. Indeed, the extrema of $|\frac{\partial f}{\partial \boldsymbol{z}}|^{-1}$ are defined by the extrema of $\psi(\boldsymbol{z})$, which in turn is defined by the extrema of $h'(\boldsymbol{w}^T\boldsymbol{z}+b)$, 0, for $h(x) = \tanh(x)$. Please note that for the generation of this Figure, the parameters were chosen such that the flow is invertible but the modified version of $\boldsymbol{u}$ described in Equation 64 was not employed.

These two effects can furthermore be observed more concretely with Figure 11, Figure 12 and Figure 13. They demonstrate how an original standard Gaussian distribution can first be contracted along the line $y = 1$ and then expanded away from the hyperplane $x = 1$

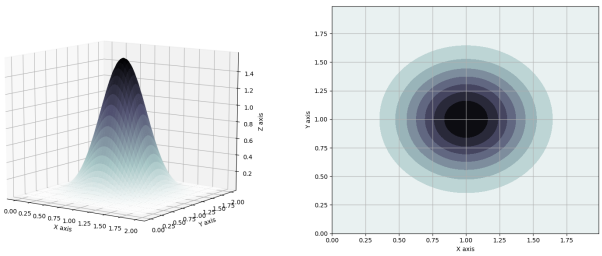

Fig. 11: Visualisation of a standard unit Gaussian distribution. On the left is displayed the two dimensional probability function and on the right the associated contour plot

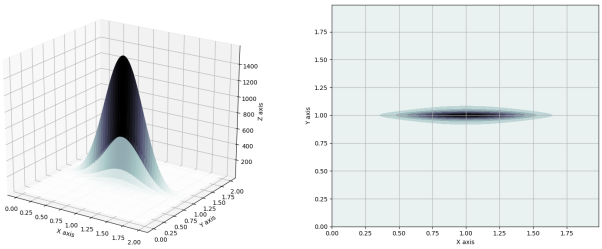

Fig. 12: Visualisation of the probability distribution resulting of the application of a planar flow inducing a contraction towards the line $y = 1$ on a standard unit Gaussian

B. Radial Flows

The second type of mapping considered in this study is expressed as

$$f(\boldsymbol{z}) = \boldsymbol{z} + \beta h(\alpha, r)(\boldsymbol{z} - \boldsymbol{z}_r) \quad (65)$$

Where $r = ||\boldsymbol{z} - \boldsymbol{z}_r||_2$, $h(\alpha, r) = \frac{1}{\alpha + r}$ and where $\boldsymbol{z}_r \in \mathbb{R}^D$, $\alpha \in \mathbb{R}^+$ and $\beta \in \mathbb{R}$ are the parameters of the mapping. The subscript in $\boldsymbol{z}_r$ indicates that it should be considered as a reference point. Again, this family of transformation respect the linear time requirement for the computation of the Jacobian determinant. Let's detail why. First

$$\begin{aligned}|\det\frac{\partial f}{\partial \boldsymbol{z}}| &= |\det(\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{z}} + \beta\frac{\partial h(\alpha,r)}{\partial \boldsymbol{z}}(\boldsymbol{z}-\boldsymbol{z}_r)^T + \beta h(\alpha,r)\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{z}})| \\ &= |\det((1+\beta h(\alpha,r))\boldsymbol{I} + \beta\frac{\partial h(\alpha,r)}{\partial \boldsymbol{z}}(\boldsymbol{z}-\boldsymbol{z}_r)^T)|\end{aligned} \quad (66)$$

Then, because according to [16]

$$\frac{\partial r}{\partial \boldsymbol{z}} = \frac{\partial||\boldsymbol{z}-\boldsymbol{z}_r||_2}{\partial \boldsymbol{z}} = \frac{\boldsymbol{z}-\boldsymbol{z}_r}{||\boldsymbol{z}-\boldsymbol{z}_r||_2} \quad (67)$$

The chain rule can be applied as follows

$$\frac{\partial h(\alpha,r)}{\partial \boldsymbol{z}} = h'(\alpha,r)\frac{\boldsymbol{z}-\boldsymbol{z}_r}{||\boldsymbol{z}-\boldsymbol{z}_r||_2} \quad (68)$$

The matrix $\frac{\boldsymbol{z}-\boldsymbol{z}_r}{||\boldsymbol{z}-\boldsymbol{z}_r||_2}(\boldsymbol{z}-\boldsymbol{z}_r)^T$ has rank 1, and can thus be diagonalised into

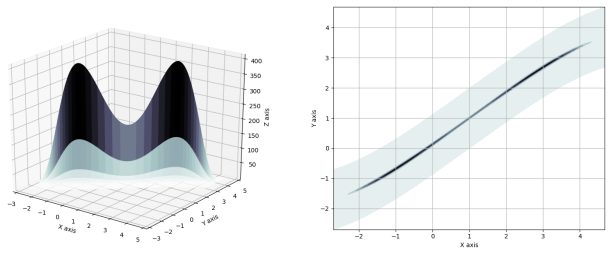

Fig. 13: Visualisation of the probability distribution resulting of the application of a planar flow inducing a contraction towards the line $y = 1$ and several expansion away from the line $x = 1$ on a standard unit Gaussian

$$\frac{\boldsymbol{z} - \boldsymbol{z}_r}{||\boldsymbol{z} - \boldsymbol{z}_r||_2}(\boldsymbol{z} - \boldsymbol{z}_r)^T = \boldsymbol{P}\begin{bmatrix} r & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix}\boldsymbol{P}^{-1} \tag{69}$$

Finally, because $\det(\boldsymbol{PAP}^{-1}) = \det(\boldsymbol{A})$, the determinant of the radial flow mapping can be expressed as

$$|\det\frac{\partial f}{\partial \boldsymbol{z}}| = (1 + \beta h(\alpha, r))^{D-1}(1 + \beta h(\alpha, r) + \beta h'(\alpha, r)r) \tag{70}$$

Which can be used to express the resulting probability distribution after the flows

$$\log(q_K(\boldsymbol{z}_k)) = \log(q_0(\boldsymbol{z}_0)) - \sum_{k=1}^{K}[(D-1)\log(1 + \beta h(\alpha, r)) + \log(1 + \beta h(\alpha, r) + \beta h'(\alpha, r)r)] \tag{71}$$

Here again, the choice of the parameters will determine whether the mapping is invertible or not. $\boldsymbol{z}$ can be decomposed as follows $\boldsymbol{z} = \boldsymbol{z}_r + r\hat{\boldsymbol{z}}$, and allows to re-arrange f as follows

$$f(\boldsymbol{z}) = \boldsymbol{z}_r + (1 + \frac{\beta}{\alpha + r})r\hat{\boldsymbol{z}} \tag{72}$$

Denoting $\boldsymbol{z}' = f(\boldsymbol{z})$ results in

$$\hat{\boldsymbol{z}} = \frac{\boldsymbol{z}' - \boldsymbol{z}_r}{r(1 + \frac{\beta}{\alpha + r})} \tag{73}$$

By definition, $||\hat{\boldsymbol{z}}||_2 = 1$, and thus

$$||\boldsymbol{z}' - \boldsymbol{z}_r||_2 = r(1 + \frac{\beta}{\alpha + r}) \tag{74}$$

For this to be invertible, it is sufficient for $r(1+\frac{\beta}{\alpha+r})$ to be a non-decreasing function [26]. Looking at its derivative with regard to r, it follows that $\beta \geq -\frac{(\alpha+r)^2}{\alpha}$, or more simply that $\beta \geq -\alpha$ must hold. This is enforced by using a corrected version of β,

$$\hat{\beta} = -\alpha + m(\beta) \tag{75}$$

Where $m(x) = \log(1+e^x)$. Furthermore, Equation 74 can be re-arranged the following second order equation in r

$$r^2 + r(\alpha + \beta - ||\boldsymbol{z}' - \boldsymbol{z}_r||_2) - \alpha||\boldsymbol{z}' - \boldsymbol{z}_r||_2 = 0 \tag{76}$$

Whose solution r^* provides an analytical expression for the inverse mapping, based on Equation 73

$$f^{-1}(\boldsymbol{z}') = \boldsymbol{z}_r + \frac{\boldsymbol{z}' - \boldsymbol{z}_r}{1 + \frac{\beta}{\alpha + r^*}} \tag{77}$$

The existence of such solution depends on the value of the parameters and of $\boldsymbol{z}'$. By noting $k = ||\boldsymbol{z}' - \boldsymbol{z}_r||_2$ and $\Delta = (\alpha + \beta - k)^2 + 4k\alpha$

$$\begin{cases} \Delta = 0 \Rightarrow r^* = \frac{-(\alpha+\beta-k)}{2} \\ \Delta > 0 \Rightarrow r^* = \frac{-(\alpha+\beta-k)+\sqrt{(\alpha+\beta-k)^2+4k\alpha}}{2} \\ \text{Else, No real solution} \end{cases} \tag{78}$$

The non-definition of the solution, and thus, the inverse function on the entire domain of $\boldsymbol{z}'$ restricts the practical usability of the inverse function to perform density estimation for example.

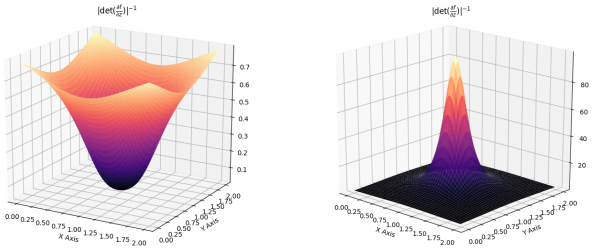

Fig. 14: Evaluation of $|\frac{\partial f}{\partial \boldsymbol{z}}|^{-1}$ for a radial expansion away from the point [1,1] (Left) and a radial contraction towards the point [1,1] (Right)

The name of Radial Flow is also derived from their effects on probability distributions. Figure 14 displays again the evolution of $|\frac{\partial f}{\partial \boldsymbol{z}}|^{-1}$ based on the given value of $\boldsymbol{z}$ for two sets of parameters that respectively produce the two possible effects of a radial flow on the probability space. Depending on the parameter values, the flow will either induce an expansion (left) or a contraction (right) of the initial density around the reference point $\boldsymbol{z}_r$. Again, please note that the correction on the parameter β presented in Equation 75 was not used to generate the plots of this section, while β and α were still chosen so that the invertibility condition would be respected.

The effects of radial flows can be observed further with Figure 15 and Figure 16. They demonstrate how an original standard Gaussian distribution can first be contracted towards the point [0.75, 0.75] and then expanded away from the point [0.85, 0.85].

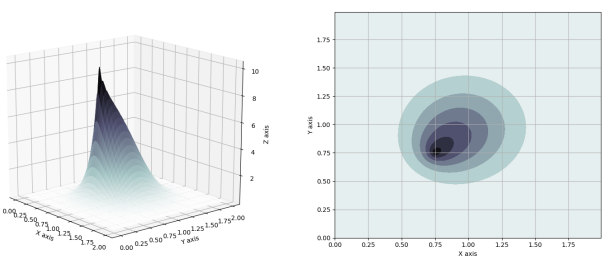

Fig. 15: Visualisation of the probability distribution resulting of the application of a radial flow inducing a contraction towards the point [0.75, 0.75] on a standard unit Gaussian

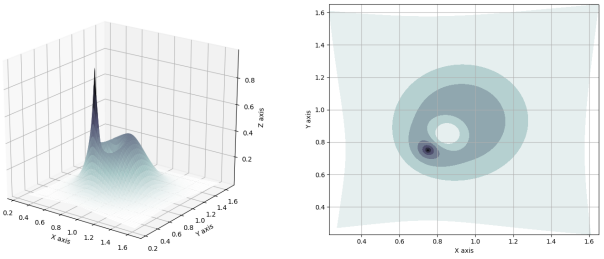

Fig. 16: Visualisation of the probability distribution resulting of the application of a radial flow inducing a contraction towards the point [0.75, 0.75] and an expansion away from the point [0.85, 0.85] on a standard unit Gaussian

V. Implementation

It is now be pretty straightforward to implement Automatic Variational Inference with Normalizing Flows. As the goal of the study is to understand the practical usability of the technique, the tests were conducted on purely theoretical posterior distributions that present interesting properties. As such, the observed variables $\boldsymbol{x}$ are either non existing or so few that the use of mini-batches is not required. The parameters of the model, $\boldsymbol{\phi}$ consist of the parameters of the base distribution, $q_0 = \mathcal{N}(\bullet|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ where $\boldsymbol{\Sigma}$ is chosen diagonal, plus the parameters of the K layers in the flow applied on q_0. The knowledge of the theoretical expressions for the joint probability distribution $p(\boldsymbol{z}, \boldsymbol{x})$ makes it easy to compute the associated Monte Carlo integral with respect to the approximated distribution after the flows q_k, and thus the ELBO function. It was seen that the Monte Carlo integral generates an unbiased estimate of the true expected value of $\log(p(\boldsymbol{z}, \boldsymbol{x}))$ with regard to the variational distribution which has a standard deviation of order $\frac{1}{\sqrt{S}}$ where S is the number of samples used for the approximation. Using a greater number of samples should therefore lead to a better evaluation of the true value of the ELBO function, and therefore to a more stable training, without a great increase in computational cost[8]. The heuristic value of $S = 500$ was used throughout the study, as it leads to a satisfactory training stability, as shown on Figure 17

[8]The difference in training time between $S = 10$, $S = 100$ and $S = 500$ was not found to be significant. Obviously, S cannot be chosen arbitrarily large as it would cause memory issues

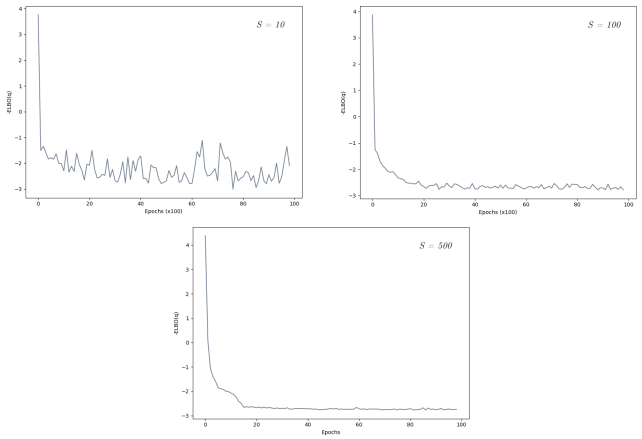

Fig. 17: Influence of the number of samples S used in each training iteration on the stability of the training. The larger the number of samples S is, the more stable training is. From top left to bottom, $S = 10$, $S = 100$, $S = 500$

Using *tensorflow*, all computations form a single computational graph which allows for easy computation of the gradient of the ELBO with respect to $\boldsymbol{\phi}$. During training the running loss is recorded, which allows to save the best performing models along the way and to stop before the end of supposed training time if the performance of the model stops improving. The Adam [32] optimiser with an initial learning rate of 1×10^{-2} was employed during training and the parameters of the model were initialised with a centered Gaussian initialiser with standard deviation 1×10^{-2}.

Algorithm 1 Variational Inference with Normalizing Flows

Parameters: $\boldsymbol{\phi}$ for the variational approximation
while Training is not finished **do**
  $\boldsymbol{z}_0 \sim q_0(\bullet)$
  $\boldsymbol{z}_K \leftarrow f_K \circ \ldots \circ f_1(\boldsymbol{z}_0)$
  $\Delta\boldsymbol{\phi} \propto -\nabla_{\boldsymbol{\phi}}\text{ELBO}(q_{\boldsymbol{\phi}})$
end while

Because the inference network was stripped away, the complexity of sampling $\boldsymbol{z}_0$ at each training step and computing the log Jacobian determinant scales with $\mathcal{O}(KD)$ where K is the number of layers in the flow and D the dimension of the latent variables.

VI. Results

As hinted by the previous section, the implementation used throughout the study (See Algorithm 1) is geared towards sampling from the variational approximation. As such, density estimation is not possible in the general case. Indeed, the expression of the distribution shaped by a normalizing flow is expressed as

$$\begin{aligned} q_K(\boldsymbol{z}_K) &= q_0((f_K \circ \ldots \circ f_1)^{-1}(\boldsymbol{z}_K)) \prod_{k=1}^{K} |\det \frac{\partial f_k}{\partial \boldsymbol{z}_{k-1}}|^{-1} \\ \boldsymbol{z}_{k-1} &= (f_K \circ \ldots \circ f_k)^{-1}(\boldsymbol{z}_K) \end{aligned} \tag{79}$$

And thus, if no expression for the respective inverse flow is known (as for the Planar Flows), it is not possible for any given variable $\boldsymbol{z}_K$ to derive the associated probability. The implementation nevertheless allows an easy access to $\boldsymbol{z}_0$, $\boldsymbol{z}_K$ and the accumulated log Jacobian determinant. It means that it is straightforward to investigate the quality of the variational approximation by displaying the generated samples and their associated probability. To improve the clarity of the representations of samples, they are aggregated into either histograms (for 1D) or hexbins[9] (for higher dimensions).

Furthermore, to increase the comparability of the resulting approximation with regard to the true posterior, an approximated probability density was computed based on the samples. This approximation relies on Kernel Density Estimation [33], using a Gaussian kernel and Scott's rule for bandwidth selection [34].

A. Demonstrative Distributions

Multimodal 1D Gaussian: Let's consider a model with a single parameter, that will be considered as a latent variable

$$x = z^2 + \epsilon \tag{80}$$

Where $\epsilon \sim \mathcal{N}(0, \sigma^2)$. Naturally, $x|z \sim \mathcal{N}(z^2, \sigma^2)$. Considering a univariate standard Gaussian prior for the latent variable $z \sim \mathcal{N}(0, 1)$, by Bayes theorem, the true posterior can be expressed as

$$p(z|x) \propto p(x|z)p(z) = \mathcal{N}(x|z^2, \sigma^2)\mathcal{N}(z|0, 1) \tag{81}$$

This simple true posterior distribution has an interesting property; it is multimodal. For a given x and σ, the posterior probability density function admits a maximum reached in two different points,

$$\begin{cases} z = \sqrt{x} \\ z = -\sqrt{x} \end{cases} \tag{82}$$

And it thus cannot be approximated accurately by a Gaussian distribution. Nevertheless, Figure 18 reveals that the planar flow allows the modification of the original Gaussian distribution (left) into a distribution that fits very closely the joint probability $p(z, x)$ (right). In light blue is displayed the samples z_0 and z_K obtained from the distribution. The dark blue curve represents the actual probability density $q_0(z_0)$ and $q_K(z_K)$ while the light blue curve displays the density function generated through Kernel Density Estimation. The influence of the flow on the original Gaussian distribution is pretty clear, by expanding the probability space away from hyperplane $x = 0$, it is able to transform the original mode of the distribution into two new modes, symmetric with regard to $x = 0$. The difference in scale between the true posterior and the generated approximation is the consequence of the neglected evidence.

[9]Hexbins, in 2D, split the plane in hexagons. In each hexagon is counted the number of samples that it covers and the color of each hexagon depends on the number of samples covered.

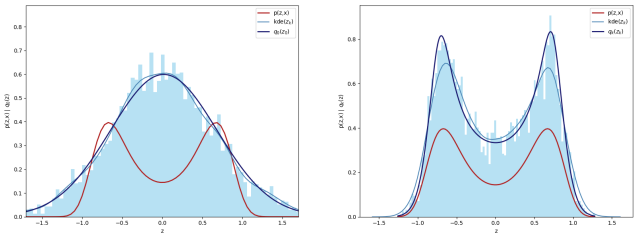

Fig. 18: Variational approximation against the true joint probability for the observed variable $x = 0.5$ and parameter $\sigma = 0.1$. On the left is displayed the initial Gaussian distribution q_0 before the planar flow and on the right q_K after the flow ($K = 3$)

Gaussian Mixtures: Going back to the Gaussian mixture presented in Section II, it can now be expected that normalizing flows can modify sufficiently the probability space on which is defined the original Gaussian distribution so that the resulting probability distribution matches closely the mixture represented by its contour in Figure 19.

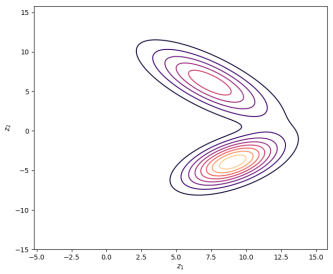

Fig. 19: True posterior distribution for the Gaussian mixture model presented in Section II

.

Figure 20 reveals that indeed the trained approximated distribution produces samples that correspond very closely to samples expected from the Gaussian mixture. Both gaussians appear very clearly, have matching shapes and the bottom one also features a higher mode that the top one. The approximated probability density function generated from the samples with Kernel Density Estimation also matches fairly closely the exact true posterior distribution.

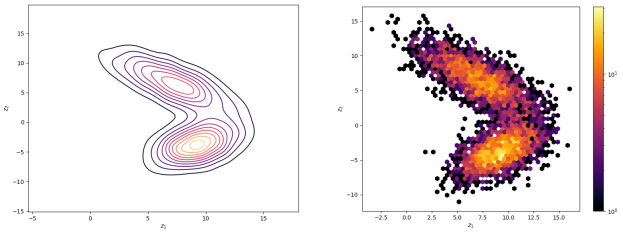

Fig. 20: Approximated probability density function estimated with Kernel Density Estimation (Left) from the samples (Right) generated from the trained variational distribution on the Gaussian mixture. Parameters, $K = 16$, epochs $= 5000$

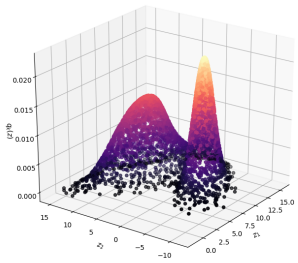

Fig. 21: Samples $\boldsymbol{z}_K$ (x and y axis) and their respective probabilities $q_K(\boldsymbol{z}_K)$ (z axis), generated from the trained variational distribution on the Gaussian mixture. Parameters, $K = 16$, epochs $= 5000$

Let's also consider another Gaussian mixture model, considerably simpler, to test whether the flexible family of variational approximations resulting from the application of normalizing flows include symmetric distributions. Let's define the likelihood probability as purely determined by the latent variables and the mixture parameters

$$p(\boldsymbol{x}|\boldsymbol{z}) = \pi_1 \mathcal{N}(\boldsymbol{z}|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}) + \pi_2 \mathcal{N}(\boldsymbol{z}|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}) \tag{83}$$

If, for simplicity's sake, the latent variables are assumed to follow an improper uniform prior $\boldsymbol{z} \sim \mathcal{U}$, the posterior distribution becomes immediately

$$p(\boldsymbol{z}|\boldsymbol{x}) \propto p(\boldsymbol{x}|\boldsymbol{z}) = \pi_1 \mathcal{N}(\boldsymbol{z}|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}) + \pi_2 \mathcal{N}(\boldsymbol{z}|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}) \tag{84}$$

Which, for $\pi_1 = \pi_2 = 0.5$ and $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ being symmetric with regard to the origin of the plane, generates the desired symmetric properties exhibited in Figure 22

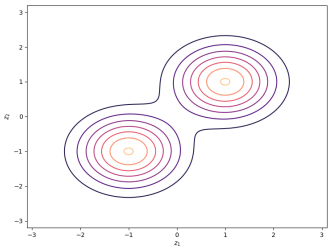

Fig. 22: True posterior distribution for the Gaussian mixture with improper prior. Parameters $\boldsymbol{\mu}_1 = [1, 1]$, $\boldsymbol{\mu}_1 = [-1, -1]$, $\boldsymbol{\Sigma} = \begin{bmatrix} 0.45 & 0 \\ 0 & 0.45 \end{bmatrix}$

Figure 23 and Figure 24 reveal that the planar flows generate a variational family broad and flexible enough to account for the symmetric and multimodal properties inherent to this Gaussian mixture model. The samples, and their respective probabilities, reveal that the learned distribution q_K indeed matches very closely the original Gaussian mixture.

Figure 25 furthermore provides an interesting visualisation of how the flow is able to transform samples originating from a Gaussian distribution into samples that seem to be generated from the true posterior distribution. The samples

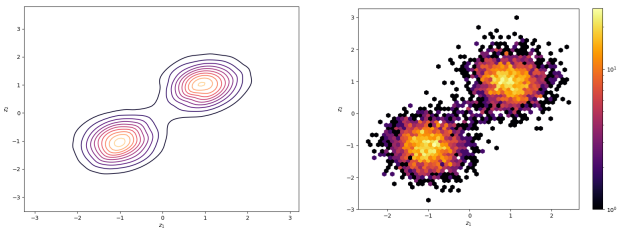

Fig. 23: Approximated probability density function estimated with Kernel Density Estimation (Left) from the samples (Right) generated from the trained variational distribution on the symmetric Gaussian mixture with improper prior. Parameters, $K = 16$, epochs $= 5000$

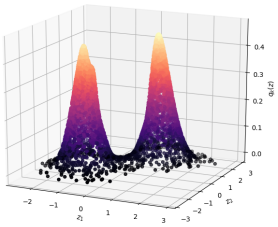

Fig. 24: Samples $\boldsymbol{z}_K$ (x and y axis) and their respective probabilities $q_K(\boldsymbol{z}_K)$ (z axis), generated from the trained variational distribution on the symmetric Gaussian mixture with improper prior. Parameters, $K = 16$, epochs $= 5000$

seem to be originally split horizontally, along the red/blue and green/yellow demarcation, and then rotated diagonally and finally reshaped so that each subgroup is contained in a circular shape.

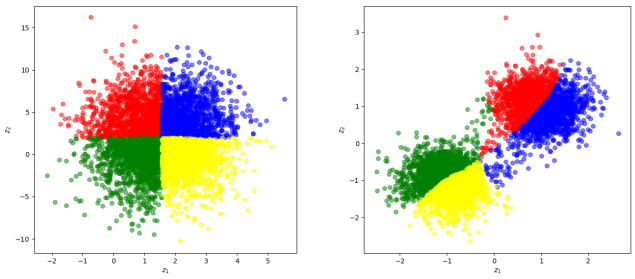

Fig. 25: Visualisation of the effect of the planar flow on the samples generated by the trained variational approximation. On the left are displayed samples from the original Gaussian $\boldsymbol{z}_0$ and on the right their image through the successive mappings that constitute the flow $\boldsymbol{z}_K$

The Banana Distribution: Next, let's introduce a posterior distribution that displays simple hierarchical properties. It is commonly referred to as the *banana distribution* and its posterior, which actually does not necessitate any observed variable can be expressed as

$$p(\boldsymbol{z}|\boldsymbol{x}) = p(\boldsymbol{z}) = \mathcal{N}(z_2|0, 4)\mathcal{N}(z_1|\frac{z_2^2}{4}, 1) \tag{85}$$

The hierarchical property arises from the dependence of the first dimension of any latent variable, on the second

dimension $\mathcal{N}(z_1|\frac{z_2^2}{4}, 1)$. Indeed if each dimension of the latent variable is instead considered as a distinct variable, then z_1 would depend on z_2. The shape of the true posterior is displayed in Figure 26.

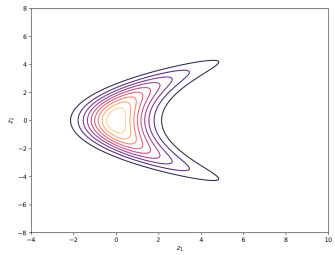

Fig. 26: Contour of the true posterior distribution for the banana model

Planar flows don't take into considerations dependencies between latent variables, unlike other types of normalising flows that will be detailed in the concluding discussion, and the shape of the true posterior distribution is not very complex. Their application should therefore result in a close match between the variation approximation and the true distribution. Figure 27 and Figure 28 shows that indeed, the trained distribution generates samples that match the general shape of the true posterior distribution. The arms of the distribution nevertheless clearly extend beyond what is expected based on the true posterior distribution. Nevertheless, even using MCMC sampling, with the very accurate No-U-Turn Sampler (NUTS), and a high number of tuning samples (3000) [35] produces a similar distribution of samples from the banana distribution, as shown in Figure 29. This indicates that the learned approximation through variational inference with normalizing flows does not underperform.

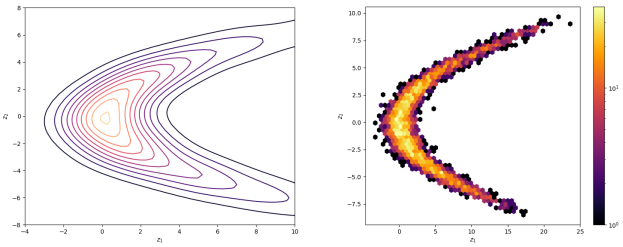

Fig. 27: Approximated probability density function estimated with Kernel Density Estimation (Left) from the samples (Right) generated from the trained variational distribution on the banana model. Parameters, $K = 16$, epochs $= 5000$

Arbitrary Distributions: Finally, it is relevant to also include arbitrary distributions[10], that do not emanate from a probabilistic model, but that instead display interesting properties to test the shaping ability of the normalizing flows.

First, let's define the circle distribution as follows

[10]They may not even be probability distributions, but they serve for demonstration purposes

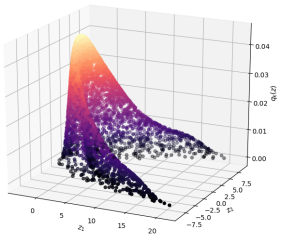

Fig. 28: Samples $\boldsymbol{z}_K$ (x and y axis) and their respective probabilities $q_K(\boldsymbol{z}_K)$ (z axis), generated from the trained variational distribution on the banana model. Parameters, $K = 16$, epochs $= 5000$

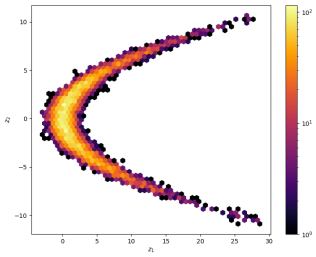

Fig. 29: Samples drawn using MCMC sampling for the banana model

$$p(\boldsymbol{z}) = \frac{e^{\frac{1}{2}(\frac{||\boldsymbol{z}||_2 - 4}{0.4})^2}}{e^{-0.2(\frac{z_1-2}{0.8})^2} + e^{-0.2(\frac{z_1+2}{0.8})^2}} \tag{86}$$

A contour of this distribution is displayed in Figure 30

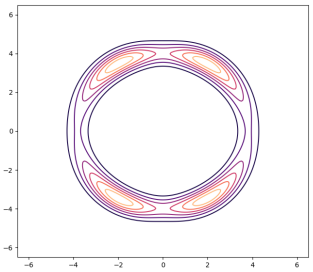

Fig. 30: Contour of the true posterior distribution for the circle case

Figure 31 and Figure 32 show that again the use of planar flows result in a satisfactory approximation. The general shape of the circle is clearly respected, the modes of the inferred distribution match closely the target modes and most importantly, the absence of samples in the middle of the circle is remarkable.

Figure 33 gives a hint to understand how the flow affected the samples and enabled such fitting. The fact that it is hard to understand exactly how the samples were transformed from the left of the figure to the right underlines the transformative power of such technique.

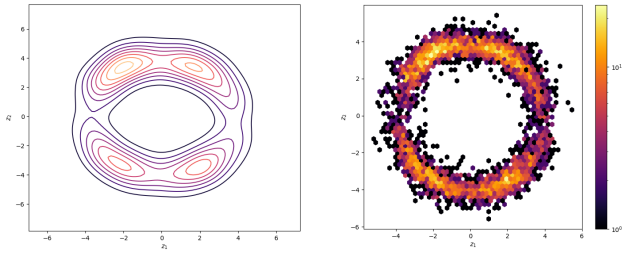

Fig. 31: Approximated probability density function estimated with Kernel Density Estimation (Left) from the samples (Right) generated from the trained variational distribution for the circle case. Parameters, $K = 32$, epochs $= 10000$

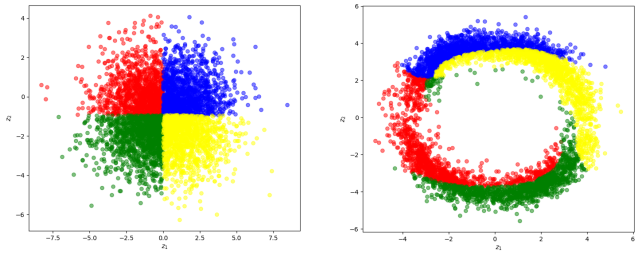

Fig. 33: Visualisation of the effect of the planar flow on the samples generated by the trained variational approximation. On the left are displayed samples from the original Gaussian $\boldsymbol{z}_0$ and on the right their image through the successive mappings that constitute the flow $\boldsymbol{z}_K$

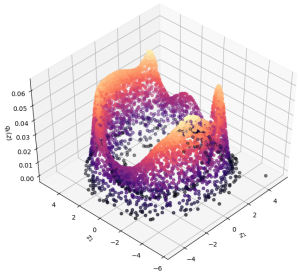

Fig. 32: Samples $\boldsymbol{z}_K$ (x and y axis) and their respective probabilities $q_K(\boldsymbol{z}_K)$ (z axis), generated from the trained variational distribution for the circle case. Parameters, $K = 32$, epochs $= 10000$

Finally, to show that the implementation used throughout the study matches the performance displayed in the paper originally introducing the technique, [26], the same energy functions that were used for demonstration will also be experimented with. Figure 34 present the density functions of the four different energies considered. They all present interesting properties. The first is completely symmetric, with a large gap between the two regions of high density, which can potentially induce the approximation to get stuck on a local minimum. The second present some clear periodic variations. The third also present some periodic variations, with a divergence, and finally, the fourth one also present periodic variations with an even greater divergence.

The associated density functions are

$$\begin{aligned}
\log(p_1(\boldsymbol{z})) &= -\frac{1}{2}\left(\frac{||\boldsymbol{z}||_2 - 2}{0.4}\right)^2 \\
&\quad + \log\left(e^{-\frac{1}{2}(\frac{z_1-2}{0.6})^2} + e^{-\frac{1}{2}(\frac{z_1+2}{0.6})^2}\right) \\
\log(p_2(\boldsymbol{z})) &= -\left(\frac{z_2 - w_1(\boldsymbol{z})}{0.4}\right)^2 \\
\log(p_3(\boldsymbol{z})) &= \log\left(e^{-\frac{1}{2}(\frac{z_2-w_1(\boldsymbol{z})}{0.35})^2} + e^{-\frac{1}{2}(\frac{z_2-w_1(\boldsymbol{z})+w_2(\boldsymbol{z})}{0.35})^2}\right) \\
\log(p_4(\boldsymbol{z})) &= \log\left(e^{-\frac{1}{2}(\frac{z_2-w_1(\boldsymbol{z})}{0.4})^2} + e^{-\frac{1}{2}(\frac{z_2-w_1(\boldsymbol{z})+w_3(\boldsymbol{z})}{0.35})^2}\right)
\end{aligned} \tag{87}$$

Where the helping function w_1, w_2 and w_3 are defined as

$$\begin{aligned}
w_1(\boldsymbol{z}) &= \sin(\frac{2\pi z_1}{4}) \\
w_2(\boldsymbol{z}) &= 3e^{-\frac{1}{2}(\frac{z_1-1}{0.6})^2} \\
w_3(\boldsymbol{z}) &= 3\sigma(\frac{z1-1}{0.3}) \\
\sigma(x) &= \frac{1}{1+e^{-x}}
\end{aligned} \tag{88}$$

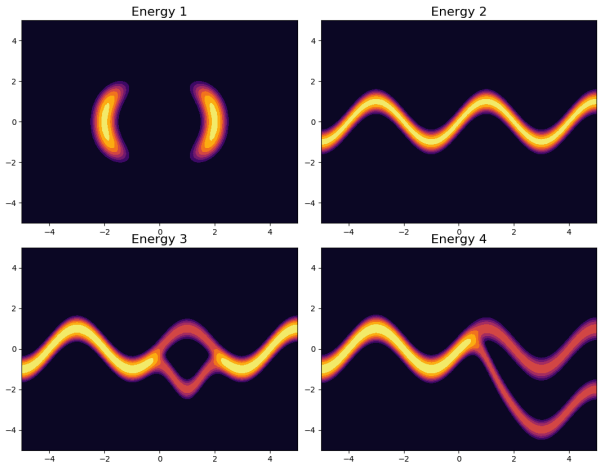

Fig. 34: True density for the demonstrative energy functions. The brighter, the higher the value for any density function

Figure 35 displays the samples generated by the trained variational approximation for the four different energy functions. The samples generally match closely the expected densities and lead to the conclusion that the implementation of the planar flows used throughout the study is indeed performing as expected. It is important nevertheless to point out that obtaining a satisfying approximation was here not as straightforward as for the examples seen previously. For the first energy function, and as expected the approximation does sometimes perform poorly as the training process leads to a local minimum, which results in the fitting of only one of the regions of high density. No mechanism was found that would guarantee the convergence towards the complete true posterior. For the fourth energy function, the exponentials involved would sometimes overflow. To increase the numerical stability of

the method, the *log-sum-exp* trick was used. For any given a, the following holds

$$\log\left(\sum_{i=1}^{n} e^{x_i}\right) = a + \log\left(\sum_{i=1}^{n} e^{x_i - a}\right) \tag{89}$$

A practical value for a is to take $a = \max_i x_i$, as it forces the greatest value in the argument of the exponentials to be 0. In the case of the fourth energy function, this dramatically improved the stability of the training and allowed the obtention of a satisfactory approximation of the density function.

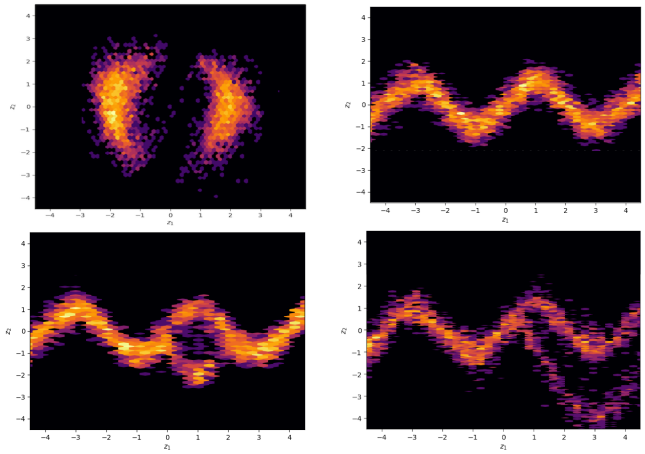

Fig. 35: Samples generated from the trained variational approximations with planar flows for the four demonstrative energy functions. Parameters $K = 32$, epochs $= 15000$

B. The Eight School Model

Finally, let's evaluate more precisely the simple normalizing flows introduced by Danilo Rezende and Shakir Mohamed in [26] on a slightly more complex problem, whose structure is notoriously hard for variational inference methods to capture. The *Eight-School Model* was introduced by A. Gelman et al in [36] Section 5.5; in eight different schools was tested a coaching program designed to prepare students for a SAT test[11], the outcome being the performance of the students on the actual test. For each school was recorded the average effect of program on the test results and its standard error, through an analysis of covariance adjustment. The results are presented in Table I.

This model is known to be the simplest hierarchical normal model [37]. The observed results of the coaching programs can understandably be assumed to be independent. The model postulates that the mean observed effect in the school i, y_i is sampled form a normal distribution centered around the true latent mean effect θ_i, with the observed standard deviation.

$$y_i|\theta_i \sim \mathcal{N}(\theta_i, \sigma_i^2) \tag{90}$$

[11]The SAT (Scholastic Aptitude Test) is a test used for college admissions in the United States of America.

School	Estimated program effect y_i	Standard deviation of program effect σ_i
1	28	15
2	8	10
3	-3	16
4	7	11
5	-1	9
6	1	11
7	18	10
8	12	18

TABLE I: Observed program effect for the preparation to the SAT test in the eight different schools

It is further considered, because the same program was tested in all schools, that the latent mean effects of the program are sampled from a shared normal distribution with parameters μ and τ

$$\theta_i|\mu, \tau \sim \mathcal{N}(\mu, \tau^2) \tag{91}$$

With the parameters being given the priors

$$\begin{aligned} \mu &\sim \mathcal{N}(0, 5) \\ \tau &\sim \text{Half-Cauchy}(0, 5) \end{aligned} \tag{92}$$

Figure 36 represents the overall *Eight-School Model* as a probabilistic graphical model for which the hierarchical relationship between the observed variables and the latent variables is very clear.

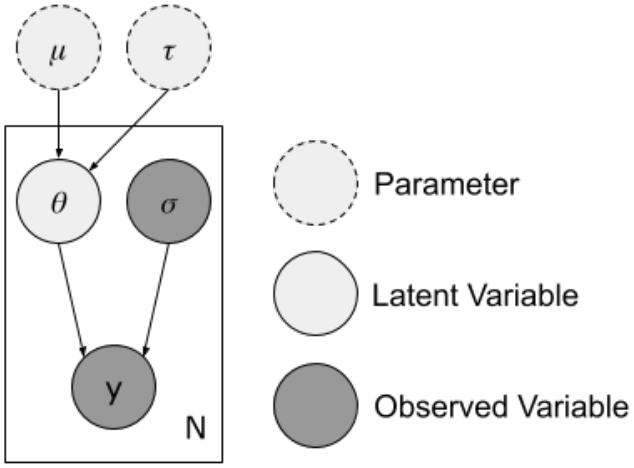

Fig. 36: Probabilistic Graphical Model for the *Eight-School Model*

The graphical model is useful to determine the joint probability distribution. Noting $\boldsymbol{y} = \{y_1, \ldots, y_8\}$, $\boldsymbol{\sigma} = \{\sigma_1, \ldots, \sigma_8\}$ and $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_8\}$, it holds that

$$\begin{aligned} p(\boldsymbol{y}, \boldsymbol{\theta}, \boldsymbol{\sigma}, \mu, \tau) &= p(\boldsymbol{y}|\boldsymbol{\theta}, \boldsymbol{\sigma}) p(\boldsymbol{\theta}|\mu, \tau) p(\mu) p(\tau) \\ &= \mathcal{N}(\boldsymbol{y}|\boldsymbol{\theta}, \boldsymbol{\sigma}) \mathcal{N}(\boldsymbol{\theta}|\mu, \tau) \\ &\quad \mathcal{N}(\mu|0, 5) \text{Half-Cauchy}(\tau|0, 5) \end{aligned} \tag{93}$$

Considering that the random variables associated to a given school are independent from those associated to any

other school, it follows that for this model the true posterior distribution is

$$\begin{aligned} p(\boldsymbol{\theta}, \mu, \tau | \boldsymbol{y}, \boldsymbol{\sigma}) &\propto p(\boldsymbol{y}, \boldsymbol{\theta}, \boldsymbol{\sigma}, \mu, \tau) \\ &= \prod_{i=1}^{8} \mathcal{N}(y_i | \theta_i, \sigma_i) \prod_{i=1}^{8} \mathcal{N}(\theta_i | \mu, \tau) \\ &\quad \mathcal{N}(\mu | 0, 5) \text{Half-Cauchy}(\tau | 0, 5) \end{aligned} \tag{94}$$

The attentive reader would have noticed that using the automatic variational inference technique directly can potentially result in negative values for τ, the standard deviation of the distribution over the true latent mean effect! Indeed, using the approach presented earlier and used without more precautions until here will approximate the distribution over the random variable τ as a Gaussian distribution, which can generate negative values. As presented in [20], it is nevertheless possible to force an invertible transformation on the latent variable τ so that its support become the real space. For τ, the original support of the random variable is $\mathbb{R}^{*+}$ and thus, the logarithm function is the natural choice here, as $\text{supp}(\log(\tau)) = \mathbb{R}$. This implies that the resulting probability density function must account for this change. By the chain rule, for any other random variable x it then holds that

$$\begin{aligned} p(x, \log(\tau)) &= p(x, \tau) |\det(J_{\log^{-1}}(\log(\tau))| \\ &= p(x, \tau) |\det(J_{\exp}(\log(\tau)))| \\ &= p(x, \tau) \tau \end{aligned} \tag{95}$$

and thus the probability that will be used in the ELBO for training the approximation should be re-expressed as

$$\begin{aligned} p(\boldsymbol{y}, \boldsymbol{\theta}, \boldsymbol{\sigma}, \mu, \log(\tau)) = \prod_{i=1}^{8} \mathcal{N}(y_i | \theta_i, \sigma_i) \prod_{i=1}^{8} \mathcal{N}(\theta_i | \mu, \tau) \\ \mathcal{N}(\mu | 0, 5) \text{Half-Cauchy}(\tau | 0, 5) \tau \end{aligned} \tag{96}$$

Consequently, $\log(\tau)$ can safely be approximated by a Gaussian without the possibility of resulting in negative standard deviation. The use of such transformation to handle random variables with constrained domains, accordingly to the title of [20], will be referred to as *Automatic Differentiation Variation Inference* or in short, ADVI.

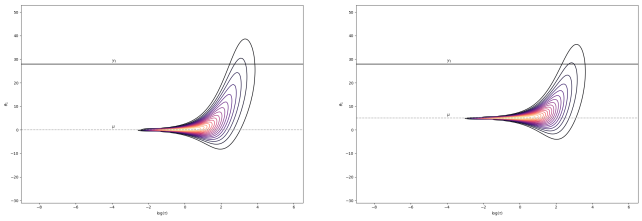

Fig. 37: Contour plot for the true joint probability $p(\log(\tau), \theta_1 | \mu) = \mathcal{N}(y_1 | \theta_1, \sigma_1) \mathcal{N}(\theta_1 | \mu, \tau) \mathcal{N}(\mu | 0, 5) \text{H-C}(\tau | 0, 5) \tau$ for $\mu = 0$ on the left and $\mu = 5$ on the right

The $\hat{k}$ Diagnostic: To assess more qualitatively how well the trained variational posterior approximates the true posterior distribution, the $\hat{k}$ diagnostic will be used for the *Eight-School model*. First introduced by A. Vehtari et al in [38], the $\hat{k}$ diagnostic evaluates the convergence of importance sampling ratios, which indicates a good approximation.

Importance sampling is closely related to the Monte Carlo integration method. In the general case, if it was possible to draw samples from the true posterior $p(\boldsymbol{z}|\boldsymbol{x})$ then it would be possible through the Monte Carlo integration to the expectation of any integrable function h of the latent variable with regard to the posterior

$$\mathop{\mathbb{E}}_{p}[h(\boldsymbol{z})] \simeq \frac{1}{S} \sum_{s=1}^{S} h(\boldsymbol{z}_s) \ , \ \boldsymbol{z}_s \sim p(\bullet | \boldsymbol{x}) \tag{97}$$

If sampling from p is not possible, then it is possible to use a proposal distribution q (such as the variational approximation for example) to compute the importance sampling estimate instead

$$\begin{aligned} \mathop{\mathbb{E}}_{p}[h(\boldsymbol{z})] &= \int h(\boldsymbol{z}) p(\boldsymbol{z}|\boldsymbol{x}) d\boldsymbol{z} \\ &= \int h(\boldsymbol{z}) \frac{p(\boldsymbol{z}|\boldsymbol{x})}{q(\boldsymbol{z})} q(\boldsymbol{z}) d\boldsymbol{z} \\ &\simeq \frac{1}{\sum_{s=1}^{S} r_s} \sum_{s=1}^{S} h(\boldsymbol{z}_s) r_s \ , \ \boldsymbol{z}_s \sim q(\bullet) \end{aligned} \tag{98}$$

Where the importance sampling ratios r_s are defined as $r_s = \frac{p(\boldsymbol{z}_s|\boldsymbol{x})}{q(\boldsymbol{z}_s)}$. The problem with importance sampling using such ratios is that the estimate may have infinite variance, if for example the ratios have a heavy right tail, which indicates a poor approximation of the true posterior by the variational approximation [38]. This problem can actually be quite common for the variational approximation as it was seen that the minimisation of the KL divergence induces a zero-avoiding behaviour in the approximation, meaning that in many cases the approximation will have a lighter tail than the true posterior, and therefore that the resulting importance sampling ratios will have a heavy tail.

A. Vehtari et al in [38] detail a remedy to this problem, the Pareto smoothed importance sampling (PSIS). A generalised Pareto distribution has three parameters, its shape k and its location and scale (μ, τ). With PSIS, a generalized Pareto distribution is fitted on the $M = \min(\frac{S}{5}, 3\sqrt{3})$ largest importance ratios. The estimated shape parameter is determined as $\hat{k}$ and the M largest ratios are replaced with their expected value under the Pareto distribution. The resulting estimate is much more stable than standard importance sampling, and furthermore, the estimated shape parameter $\hat{k}$ provides a diagnostic to verify that the determined variational approximation $q(\boldsymbol{z})$ is close to the true posterior distribution $p(\boldsymbol{z}|\boldsymbol{x})$.

Indeed, as detailed in [37], any positive value $\hat{k}$ can be viewed as an estimate to

$$k = \inf\left\{k' > 0 : D_{\frac{1}{k'}}(p||q) < \infty\right\}$$
$$\text{where } D_\alpha(p||q) = \frac{1}{\alpha - 1}\log\int p(\boldsymbol{z})^\alpha q(\boldsymbol{z})^{1-\alpha} d\boldsymbol{z} \tag{99}$$

Where D_α is a Rényi divergence of order α between $p(\boldsymbol{z}|\boldsymbol{x})$ and $q(\boldsymbol{z})$. Notably, when $k > 1$, $D_1 = KL$ becomes infinite, indicating a very poor variational approximation. A. Vehtari et al [38] conducted an empirical study to determine thresholds for the value of $\hat{k}$, which indicates the following

- if $\hat{k} < 0.5$, PSIS has a fast convergence rate, suggesting that the variational approximation q is close to the true posterior probability density.
- If $0.5 < \hat{k} < 0.7$, there is still convergence and acceptable Monte Carlo error for PSIS. It indicates that even if the fitting of the approximation is not perfect, it can still has practical use.
- If $\hat{k} > 0.7$, then the PSIS convergence is too slow. This means that the variational approximation should be re-considered as a correct approximation of the true posterior.

In practice, the implementation provided on *GitHub* of PSIS by A. Vehtari[12] himself was used to compute the estimated $\hat{k}$ for any trained approximation.

MCMC Sampling: For the *Eight-School Model*, it is not possible anymore to compare samples, and their associated KDE estimates, generated from the approximated distribution to the true posterior distribution. As a fallback, MCMC samples, generated with the NUTS sampler [35] and *PyMC3*[13] can provide a reasonable basis for comparison. The generation of the MCMC samples reveals one of the first specificity of the model, the strong dependence between the standard deviation τ and the latent mean effect in school i θ_i, the latter behind conditioned by the former. This results in a joint distribution that presents a very distinguishable neck, as can be seen in Figure 38. For low standard deviations, the variance of the associated samples for θ_i should decrease accordingly. The presence of this neck makes it difficult to generate samples that truly represent the joint distribution. For this reason, relying on the standard expression of the model results in a poor exploration of the distribution[14]. Instead, a **non-centered** variation of the model was employed for MCMC sampling. For the latent mean effect of the coaching program, the following reparametrisation was employed

$$\theta_i = \mu + \tau\widetilde{\theta}_i$$

$$\begin{aligned}\widetilde{\theta}_i &\sim \mathcal{N}(0,1)\\ \mu &\sim \mathcal{N}(0,5)\\ \tau &\sim \text{Half-Cauchy}(0,5)\end{aligned} \tag{100}$$

Which encourages the exploration of the distribution as the latent mean effects θ_i need not be fitted directly anymore, but instead a standard Gaussian is fitted and the θ_i can be recovered with a scaling and a translation. The samples obtained are displayed in Figure 38, Figure 39 and 40.

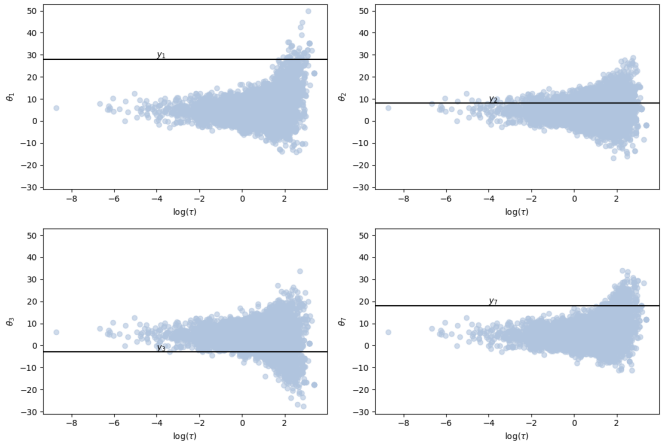

Fig. 38: MCMC samples for the *Eight-School* model showing $\log(\tau)$ against θ_i for various schools. Parameters $N = 5000$, 2 chains and 1000 tuning iterations

On Figure 38, Figure 39 is also displayed for each school the observed mean effect with a black horizontal line. The influence of the observed y_i over the sampled latent mean effect θ_i is noticeable; if the observed effect is far away from 0, the resulting samples will be skewed towards the observed effect, as can be seen for θ_1 (positively skewed) or for θ_3 (negatively skewed).

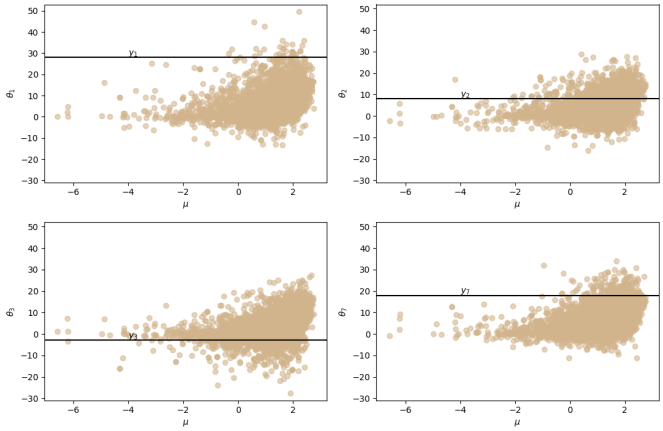

Fig. 39: MCMC samples for the *Eight-School* model showing μ against θ_i for various schools. Parameters $N = 5000$, 2 chains and 1000 tuning iterations

[12] https://github.com/avehtari/PSIS

[13] https://docs.pymc.io/

[14] For more details see https://docs.pymc.io/notebooks/Diagnosing_biased_Inference_with_Divergences.html?highlight=eight_school

Finally, Figure 40, which compares the samples -and the associated approximate distribution estimated with KDE- (blue) and the true prior distributions for the parameters μ and τ, reveals that even the MCMC samples cannot fit the expected distributions perfectly. The distribution of the sampled μ, even though it follows a Gaussian shape, clearly displays a shift towards the positive values, as its mode is located around 5 and not 0 as expected. A possible explanation for this is that the average observed value is $\bar{y} = 8.75$, and that it thus forces the approximation towards positive values for the latent means and thus for the true average of the latent means.

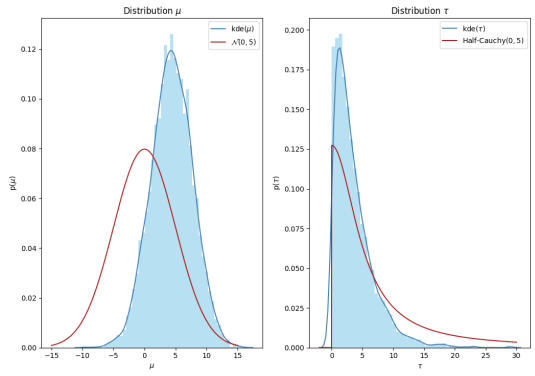

Fig. 40: MCMC samples for the *Eight-School* model for μ and τ against the true respective prior distributions $\mathcal{N}(0, 5)$ and Half-Cauchy(0,5). Parameters $N = 5000$, 2 chains and 1000 tuning iterations

To evaluate the approximation generated using Variational Inference methods, the following procedure was followed. For a given model and parameters, training was repeated 5 times. After each respective training, a diagnostic was established, that sampled 50 different sets of 5000 samples to evaluate the $\hat{k}$ diagnostic and the ELBO function. The average $\hat{k}$ and ELBO was then computed amongst the set of samples and reported for that training run. The average $\hat{k}$ and ELBO, and their respective standard deviation, was then determined for the 5 training runs, and will serve for comparison. Finally, a visualisation of the samples obtained from the trained approximate distribution that lead to the best diagnostic, focusing on θ_1, was also generated.

A baseline for comparison was first established by using ADVI without any normalizing flow. The resulting diagnostic, or fitting metrics, $\hat{k}$ and ELBO(q_k), are exposed in Table II, and the associated samples displayed in Figure 41 and Figure 42.

Centered ?	Mean $\hat{k}$	Std $\hat{k}$	Mean ELBO	Std ELBO
Yes	0.78	0.049	-60.71	0.027
No	0.11	0.036	-60.20	0.022

TABLE II: Fitting Metrics for the ADVI approximation on the *Eight-School Model*. Parameters epochs $= 15000$

The non-centered model clearly outperforms, as expected from the MCMC sampling attempts, its centered counterpart. This is understandable as in the non-centered case, the variational approximation needs only to fit a standard Gaussian that is then translated and scaled rather than a Gaussian whose parameters also need to be inferred. The top part of Figure 41 shows that for the centered model the learned approximation restricted to $(\log(\tau), \theta_1)$ is indeed a Gaussian which can therefore not learn the dependence relationship linking the two variables. In the non-centered case, it can be seen that the scaling operation allows to capture a part of the dependency between τ and θ_1, as for larger τ, the scaling clearly has a greater effect on the samples. Nevertheless, in both cases, the skewness is not captured at all, as revealed by the comparison between the distributions of samples μ generated from the ADVI (purple) and MCMC approximation (blue). Overall, the non-centered ADVI approximation provides a good baseline, with a very acceptable $\hat{k}$ diagnostic, which suggest a good convergence of importance sampling ratios, while the centered ADVI approximation is not satisfactory by any means.

The addition of normalizing flows (with planar flows) leads to an closer fit for the variational approximation. The resulting fitting metrics is exposed in Table III, and the samples are displayed in Figure 43 and Figure 44.

Centered ?	Mean $\hat{k}$	Std $\hat{k}$	Mean ELBO	Std ELBO
Yes	0.65	0.050	-60.05	0.012
No	0.23	0.039	-59.50	0.006

TABLE III: Fitting Metrics for the ADVI+NF approximation on the *Eight-School Model*. Parameters $K = 64$, epochs $= 15000$

The effect of the flows is not as spectacular as what was observed for the demonstrative examples. It seems that the hierarchical properties and the increase in the number of dimensions make it more difficult to warp the latent space so that the resulting distribution matches closely the true posterior distribution. This can be understood pretty intuitively, the higher the dimension of the latent space (10 dimensions here, $\theta_1, \ldots, \theta_8$, μ and τ), the more complex becomes the problem of finding hyperplanes that modify the latent space in an adequate way. The increase in performance compared to standard ADVI is nevertheless significant (the new ELBO values being outside of the 95% confidence interval around the ELBO values for both the centered and non-centered ADVI models), and for the conclusion of the $\hat{k}$ diagnostic even changes -as $\hat{k} < 0.7$- for the centered model. The distribution of samples generated from the variational approximation also seem to be a closer fit of the MCMC samples. In the centered case, a small improvement is noticeable in capturing the dependency between τ and θ_1, and for the non-centered case, the skewness is more accurately captured, with a distribution of θ_1 that is not centered anymore.

To investigate whether the fitting ability of the flows is impacted by the dimensionality of the problem, a restricted *Eight-School Model* was considered, for which only the first school is taken into account. The latent variables are thus reduced to only θ_1, τ and μ. The resulting diagnostic are exposed in Table IV and the generated samples displayed in Figure 45 for the centered model and in Figure 46 for the non-centered model.

Centered ?	Mean $\hat{k}$	Std $\hat{k}$	Mean ELBO	Std ELBO
Yes	0.46	0.048	-5.24	0.005
No	0.28	0.042	-5.13	0.003

TABLE IV: Fitting Metrics for the ADVI+NF approximation on the *Eight-School Model*, reduced to the first school only. Parameters $K = 64$, epochs $= 15000$

The improvement in the fitting is striking. For $\hat{k}$ diagnostics drop below 0.5, which suggests that the variational approximation is close to the true posterior probability density, and the samples displayed match very closely the MCMC samples, so much actually, that the MCMC samples are almost not visible anymore in the case of the non-centered model. The skewness is captured, and contrary to the MCMC samples, the approximated distribution of μ stays centered (please note that the comparison is not entirely fair as the MCMC procedure was trying to approximate the true posterior for all 8 schools at once). The color map that indicates the probability $q_k(\boldsymbol{z}_k)$ of each sample drawn from the variational approximation can be compared with Figure 37. The centered model, even though its fitting metrics are worse, and its associated samples are not as similar as the MCMC samples, seem to approximate more accurately the density function of the true posterior.

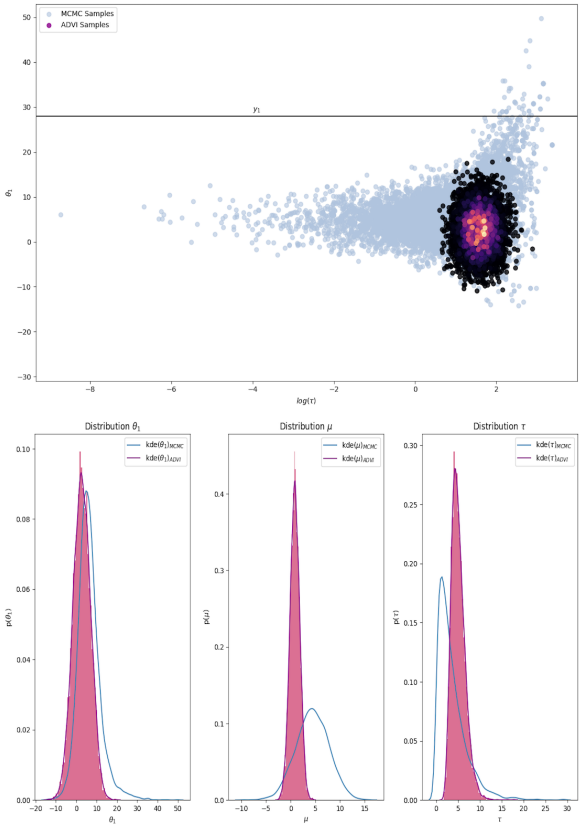

Fig. 41: Comparison of the trained variational approximation using the **ADVI framework** against the MCMC approximation. On top are displayed samples of $(\log(\tau), \theta_1)$ from both approximations, and on bottom are displayed the distributions of θ_1, μ and τ against the estimated true densities computed with KDE on the MCMC samples. Parameters, epochs=15000

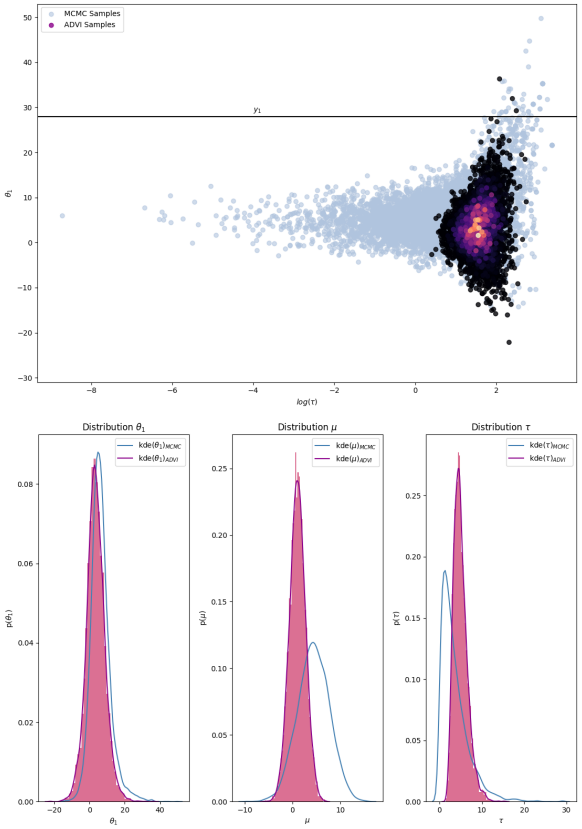

Fig. 42: Comparison of the trained variational approximation using the **ADVI framework for the non-centered model** against the MCMC approximation. On top are displayed samples of $(\log(\tau), \theta_1)$ from both approximations, and on bottom are displayed the distributions of θ_1, μ and τ against the estimated true densities computed with KDE on the MCMC samples. Parameters, epochs=15000

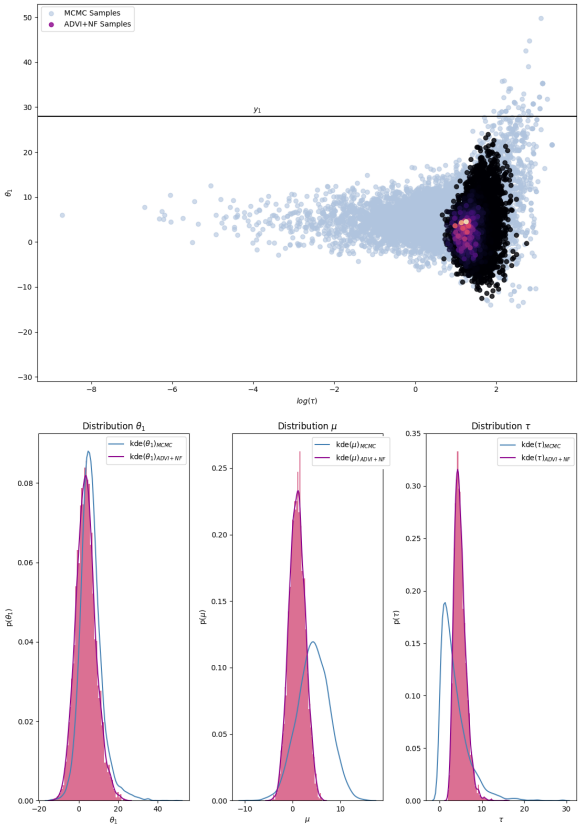

Fig. 43: Comparison of the trained variational approximation using the **ADVI framework with Normalizing Flows** against the MCMC approximation. On top are displayed samples of $(\log(\tau), \theta_1)$ from both approximations, and on bottom are displayed the distributions of θ_1, μ and τ against the estimated true densities computed with KDE on the MCMC samples. Parameters, $K = 64$, epochs=15000

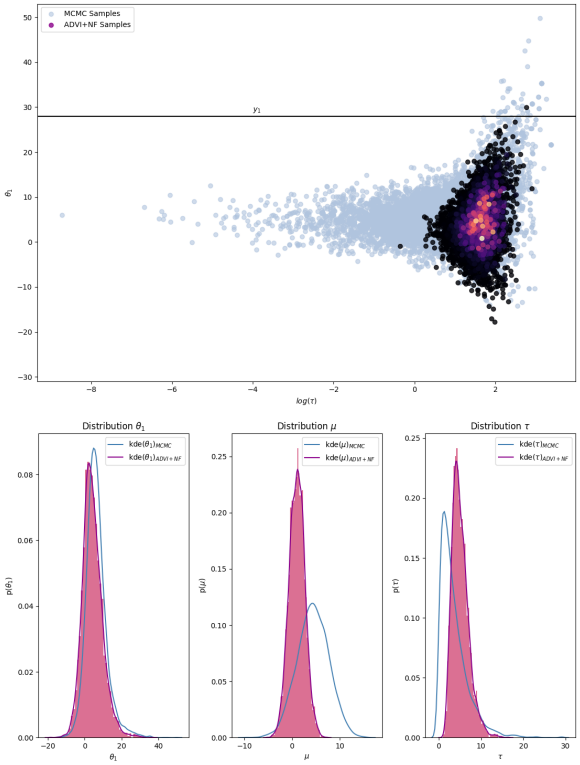

Fig. 44: Comparison of the trained variational approximation using the **ADVI framework with Normalizing Flows for the non-centered model** against the MCMC approximation. On top are displayed samples of $(\log(\tau), \theta_1)$ from both approximations, and on bottom are displayed the distributions of θ_1, μ and τ against the estimated true densities computed with KDE on the MCMC samples. Parameters, $K = 64$, epochs=15000

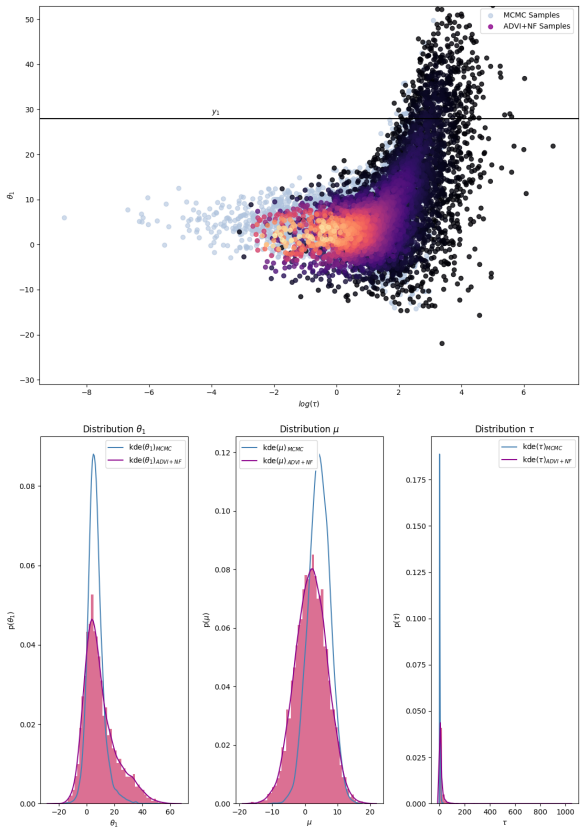

Fig. 45: Comparison of the trained variational approximation using the **ADVI framework with Normalizing Flows, reduced to the first school only** against the MCMC approximation. On top are displayed samples of $(\log(\tau), \theta_1)$ from both approximations, and on bottom are displayed the distributions of θ_1, μ and τ against the estimated true densities computed with KDE on the MCMC samples. Parameters, $K = 64$, epochs=15000

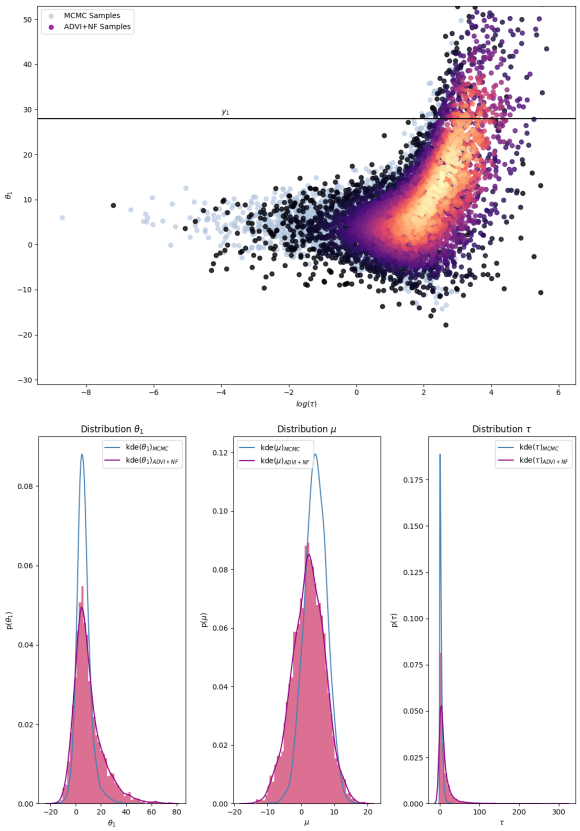

Fig. 46: Comparison of the trained variational approximation using the **ADVI framework with Normalizing Flows, reduced to the first school only, for the non centered model** against the MCMC approximation. On top are displayed samples of $(\log(\tau), \theta_1)$ from both approximations, and on bottom are displayed the distributions of θ_1, μ and τ against the estimated true densities computed with KDE on the MCMC samples. Parameters, $K = 64$, epochs=15000

The previous findings suggest there is a relationship between the complexity of the model and the practical performance of the normalizing flows, for a given flow depth. In theory, increasing the depth of the flow expands the possible transformations that can be applied on the original distribution. As a result, the resulting variational family Q broadens to include potentially better performing approximate distributions. To study the influence of the depth of the flow, i.e the number of layers of successive invertible mappings that constitute the flow -called here *number of flows*-, various variational approximations learned with ADVI+NF with a varying number of flows are compared for the *Eight-School Model* reduced to only the first two schools, in its centered version[15]. Table V exhibits the measured evolution of the fitting metrics based on the number of flows, also displayed in Figure 47. Figure 48 displays the evolution of samples generated from variational approximations obtained using a varying number of flows.

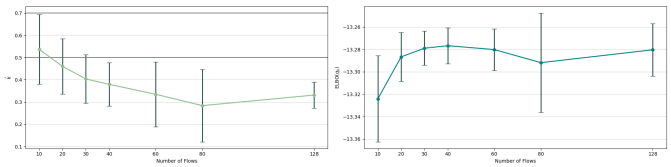

Fig. 47: Evolution of the two fitting metrics for the ADVI+NF approximation on the *Eight-School Model* reduced to only the first two schools. On the left is shown the progression of $\hat{k}$ as a function of the number of layers in the normalizing flow and on the right, ELBO(q_k) as a function of the number of layers. The error bars corresponds to the average $\pm$ 1.96 standard deviations.

Number of Flows	Mean $\hat{k}$	Std $\hat{k}$	Mean ELBO	Std ELBO
10	0.54	0.080	-13.32	0.020
20	0.46	0.064	-13.29	0.011
30	0.40	0.056	-13.28	0.008
40	0.38	0.050	-13.28	0.008
60	0.33	0.074	-13.28	0.009
80	0.28	0.083	-13.29	0.023
128	0.33	0.030	-13.28	0.012

TABLE V: Fitting Metrics for the ADVI+NF approximation on the *Eight-School Model*, reduced to the first two schools only

The results suggest that indeed, up to $K = 80$, the deepening of the flow indeed results in an improvement of the variational approximation. The $\hat{k}$ decreases very clearly, and drops quite certainly below the threshold $\hat{k} = 0.5$ for $K = 40$, the ELBO increases as well, although not as clearly. The great incertitude for the ELBO measured for $K = 80$ unfortunately prevents drawing any clear conclusions, but the generated samples for $K = 10$, $K = 20$, $K = 40$ and $K = 80$ quite clearly that the deeper the flow is, the closer the samples from the VI approximation match their MCMC counterpart, notably because they explore deeper the neck of the distribution.

[15]The centered version was chosen as it is the worst performing and can thus provide a lower bound for the potential beneficiary effect of deepening the flow.

It is finally interesting to note that for $K = 128$, the improvement of the approximation stops to be so clear. The $\hat{k}$ is not significantly lower than for shallower flows, and the improvement in the ELBO is negligible. This suggests a practical limitation to the beneficiary effect of deepening normalizing flows. No mention of this effect is present in Danilo Rezende and Shakir Mohamed's paper [26]. It is likely that this effect is related to the vanishing gradient problem [39], resulting from the use of the non-linear hyperbolic tangent function in each flow. It would therefore be overly optimistic to hope to increase the overall performance of the variational approximation generated with normalizing flows on the complete eight-school model by simply increasing the depth of the flow.

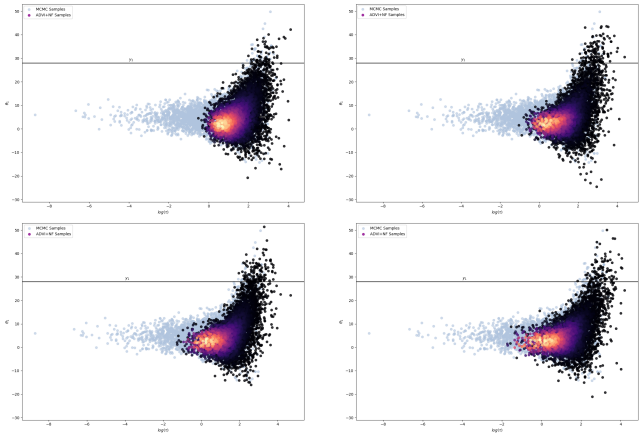

Fig. 48: Evolution of the samples generated by the variational approximation based on the number of flows. From top left to bottom right, $K = 10$, $K = 20$, $K = 40$, $K = 80$

VII. Discussion

This study detailed how normalizing flows, which are constituted by a succession of invertible mappings, can transform probability distributions, Gaussian typically, into new distributions, with highly complex features. Normalizing flows, employed together with an automatic and differentiated approach to variational inference, relying on Monte Carlo gradient estimation for the optimisation of the ELBO function, provides an efficient, scalable and versatile way to approximate arbitrarily complex posterior distribution of probabilistic models. Demonstrative posterior distributions, featuring varied non standard features were used to reveal the remarkable flexibility of variational approximations obtained with normalizing flows. Furthermore, the eight-school model was used to first show that in the case of a significantly more complex model, with a higher dimensional latent space and hierarchical dependencies, the normalizing flows indeed provide a significant improvement in the variational approximation. It also revealed that the performance of the flows is limited by the number of layers in the flow. Theoretically, the deeper the flow is, the richer posterior approximation it should generate, but it was seen that in practice, deepening the flow only has

a beneficial impact up to a certain depth. It was postulated that the flows were subject to the vanishing gradient problem, and a natural extension of this study would be to experiment with non-linearities that are less sensitive to this issue.

Finally, while this study proves that the use of normalizing flows indeed improves the performance of variational inference techniques, it mostly demonstrates that their implementation is not able to overcome all the objections to using variational inference as a default approach for statistical inference. Contrarly to what was postulated by Danilo Rezende and Shakir Mohamed in [26]

> "*Normalizing flows allow us to control the complexity of the posterior at run-time by simply increasing the flow length of the sequence*"

There are clear limitations to the practical use of normalising flows. It is nevertheless a very promising method, which if used with better suited mappings, such as in *Inverse Autoregressive Flows* [40] or in *Masked Autoregressive Flows* [41], can result in practical techniques with state of the art performance, as was achieved with the *Parallel WaveNet* [11] model which is currently used by Google Assistant to generate realistic speech.

VIII. Source Code

All code used throughout this study is accessible at https://github.com/pierresegonne/VINF.

Acknowledgements

This special course was supervised by Michael Riis Andersen.

REFERENCES

[1] W. K. Hastings, "Monte carlo sampling methods using markov chains and their applications," 1970.

[2] A. E. Gelfand and A. F. Smith, "Sampling-based approaches to calculating marginal densities," *Journal of the American statistical association*, vol. 85, no. 410, pp. 398–409, 1990.

[3] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 721–741, 1984.

[4] C. Robert and G. Casella, *Monte Carlo statistical methods*. Springer Science & Business Media, 2013.

[5] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[6] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "An introduction to variational methods for graphical models," *Machine learning*, vol. 37, no. 2, pp. 183–233, 1999.

[7] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *Journal of the American statistical Association*, vol. 112, no. 518, pp. 859–877, 2017.

[8] 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 (Methodological)*, vol. 39, no. 1, pp. 1–22, 1977.

[9] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling, "Semi-supervised learning with deep generative models," in *Advances in neural information processing systems*, pp. 3581–3589, 2014.

[10] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[11] A. v. d. Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. v. d. Driessche, E. Lockhart, L. C. Cobo, F. Stimberg, *et al.*, "Parallel wavenet: Fast high-fidelity speech synthesis," *arXiv preprint arXiv:1711.10433*, 2017.

[12] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[13] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[14] S. Kotz, N. Balakrishnan, and N. L. Johnson, *Continuous multivariate distributions, Volume 1: Models and applications*. John Wiley & Sons, 2004.

[15] A. T. James *et al.*, "Distributions of matrix variates and latent roots derived from normal samples," *The Annals of Mathematical Statistics*, vol. 35, no. 2, pp. 475–501, 1964.

[16] K. B. Petersen and M. S. Pedersen, *The Matrix Cookbook*. Technical University of Denmark, 2012.

[17] G. J. McLachlan and T. Krishnan, *The EM algorithm and extensions*, vol. 382. John Wiley & Sons, 2007.

[18] A.-L. Cauchy, "Méthodologie générale pour la résolution des systèmes d'équations simultanées," *Comptes rendus hebdomadaires des séances de l'Académie des sciences*, pp. 536–538, 1847.

[19] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

[20] A. Kucukelbir, D. Tran, R. Ranganath, A. Gelman, and D. M. Blei, "Automatic differentiation variational inference," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 430–474, 2017.

[21] J. V. Dillon, I. Langmore, D. Tran, E. Brevdo, S. Vasudevan, D. Moore, B. Patton, A. Alemi, M. Hoffman, and R. A. Saurous, "Tensorflow distributions," *arXiv preprint arXiv:1711.10604*, 2017.

[22] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pp. 265–283, 2016.

[23] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[24] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*, pp. 177–186, Springer, 2010.

[25] M. Titsias and M. Lázaro-Gredilla, "Doubly stochastic variational bayes for non-conjugate inference," in *International conference on machine learning*, pp. 1971–1979, 2014.

[26] D. J. Rezende and S. Mohamed, "Variational inference with normalizing flows," *arXiv preprint arXiv:1505.05770*, 2015.

[27] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," *arXiv preprint arXiv:1401.4082*, 2014.

[28] E. G. Tabak, E. Vanden-Eijnden, *et al.*, "Density estimation by dual ascent of the log-likelihood," *Communications in Mathematical Sciences*, vol. 8, no. 1, pp. 217–233, 2010.

[29] E. G. Tabak and C. V. Turner, "A family of nonparametric density estimation algorithms," *Communications on Pure and Applied Mathematics*, vol. 66, no. 2, pp. 145–164, 2013.

[30] J. L. Devore and K. N. Berk, *Modern mathematical statistics with applications*. Springer, 2012.

[31] O. Rippel and R. P. Adams, "High-dimensional probability estimation with deep density models," *arXiv preprint arXiv:1302.5125*, 2013.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[33] G. R. Terrell and D. W. Scott, "Variable kernel density estimation," *The Annals of Statistics*, pp. 1236–1265, 1992.

[34] D. W. Scott, *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015.

[35] M. D. Hoffman and A. Gelman, "The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo.," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1593–1623, 2014.

[36] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin, *Bayesian data analysis*. CRC press, 2013.

[37] Y. Yao, A. Vehtari, D. Simpson, and A. Gelman, "Yes, but did it work?: Evaluating variational inference," *arXiv preprint arXiv:1802.02538*, 2018.

[38] A. Vehtari, D. Simpson, A. Gelman, Y. Yao, and J. Gabry, "Pareto smoothed importance sampling," *arXiv preprint arXiv:1507.02646*, 2015.

[39] S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber, *et al.*, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies," 2001.

[40] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling, "Improved variational inference with inverse autoregressive flow," in *Advances in neural information processing systems*, pp. 4743–4751, 2016.

[41] G. Papamakarios, T. Pavlakou, and I. Murray, "Masked autoregressive flow for density estimation," in *Advances in Neural Information Processing Systems*, pp. 2338–2347, 2017.

Appendix A: Probabilistic Linear Regression

Provided that for the parameter ω, the prediction value for a data point x is $\omega^T x$, it can be postulated that the target variable follows a normal distribution centered around the predicted value, for a given standard deviation σ.

$$y|\omega, x \sim \mathcal{N}(\omega^T x, \sigma^2 I) \tag{101}$$

If furthermore, it is expected that the weights of the model also follow a Gaussian distribution, centered, and with a given standard deviation γ

$$\omega \sim \mathcal{N}(0, \gamma^2 I) \tag{102}$$

Then the MAP estimate can be obtained by maximising the log-probability of the posterior distribution

$$\begin{aligned}\omega^* &= \max_{\omega} \log(p(y|\omega, x)p(\omega)) \\ &= \max_{\omega} -\frac{1}{2\sigma^2}(y - \omega^T x)^T(y - \omega^T x) - \frac{1}{2\gamma}\omega^T\omega \\ &= \max_{\omega} ||y - \omega^T x||_2^2 + (\frac{\sigma}{\gamma})^2||\omega||_2^2\end{aligned} \tag{103}$$

Appendix B: Demonstration of the Positiveness of the Kullback-Leibler Divergence

For the following demonstration, two probability distributions p and q over continuous random variables z will be considered.

First, $\forall x > 0$, $\log(x) \leq x - 1$ so it holds that

$$\begin{aligned}-\int_{-\infty}^{\infty} p(z)\log(\frac{q(z)}{p(z)})dz &\geq -\int_{-\infty}^{\infty} p(z)(\frac{q(z)}{p(z)} - 1)dz \\ &\geq -\int_{-\infty}^{\infty} q(z)dz + \int_{-\infty}^{\infty} p(z)dz \\ &\geq 0\end{aligned} \tag{104}$$

Which thus yields directly that

$$KL(p||q) \geq 0 \tag{105}$$

And also results in Gibbs' inequality

$$-\int_{-\infty}^{\infty} p(z)\log(q(z))dz \geq -\int_{-\infty}^{\infty} p(z)\log(p(z))dz \tag{106}$$

Appendix C: Demonstration of the log-sum-exp

Given $y = \log(\sum_{i=1}^{n} e^{x_i})$, the following holds

$$\begin{aligned}y &= \log\left(\sum_{i=1}^{n} e^{x_i}\right) \\ \Leftrightarrow e^y &= \sum_{i=1}^{n} e^{x_i} \\ \Leftrightarrow e^{y-a} &= \sum_{i=1}^{n} e^{x_i - a} \\ \Leftrightarrow y - a &= \log\left(\sum_{i=1}^{n} e^{x_i - a}\right)\end{aligned} \tag{107}$$

finally resulting in the log-sum-exp trick

$$\log\left(\sum_{i=1}^{n} e^{x_i}\right) = a + \log\left(\sum_{i=1}^{n} e^{x_i - a}\right) \tag{108}$$