Copula-like Variational Inference

Marcel Hirt
Department of Statistical Science
University College of London, UK
marcel.hirt.16@ucl.ac.uk

Petros Dellaportas
Department of Statistical Science
University College of London, UK
Department of Statistics
Athens University of Economics and Business, Greece
and The Alan Turing Institute, UK

Alain Durmus
CMLA
École normale supérieure Paris-Saclay,
CNRS, Université Paris-Saclay, 94235 Cachan, France.
alain.durmus@cmla.ens-cachan.fr

Abstract

This paper considers a new family of variational distributions motivated by Sklar's theorem. This family is based on new copula-like densities on the hypercube with non-uniform marginals which can be sampled efficiently, *i.e.* with a complexity linear in the dimension of state space. Then, the proposed variational densities that we suggest can be seen as arising from these copula-like densities used as base distributions on the hypercube with Gaussian quantile functions and sparse rotation matrices as normalizing flows. The latter correspond to a rotation of the marginals with complexity $\mathcal{O}(d \log d)$. We provide some empirical evidence that such a variational family can also approximate non-Gaussian posteriors and can be beneficial compared to Gaussian approximations. Our method performs largely comparably to state-of-the-art variational approximations on standard regression and classification benchmarks for Bayesian Neural Networks.

1 Introduction

Variational inference [28, 63, 4] aims at performing Bayesian inference by approximating an intractable posterior density π with respect to the Lebesgue measure on $\mathbb{R}^d$, based on a family of distributions which can be easily sampled from. More precisely, this kind of inference posits some variational family Q of densities $(q_\xi)_{\xi\in\Xi}$ with respect to the Lebesgue measure and intends to find a good approximation $q_{\xi^\star}$ belonging to Q by minimizing the Kullback-Leibler (KL) with respect to π over Q, *i.e.* $\xi^\star \approx \arg\min_{\xi\in\Xi} \mathrm{KL}(q_\xi|\pi)$. Further, suppose that $\pi(x) = \mathrm{e}^{-U(x)}/\mathrm{Z}$ with $U \colon \mathbb{R}^d \to \mathbb{R}$ measurable and $\mathrm{Z} = \int_{\mathbb{R}^d} \mathrm{e}^{-U(x)}\mathrm{d}x < \infty$ is an unknown normalising constant. Then, for any $\xi \in \Xi$,

$$\mathrm{KL}(q_\xi|\pi) = -\int_{\mathbb{R}^d} q_\xi(x) \log \frac{\pi(x)}{q_\xi(x)} \mathrm{d}x = -\mathbb{E}_{q_\xi(x)}\left[-U(x) - \log q_\xi(x)\right] + \log \mathrm{Z} \,. \tag{1}$$

Since Z does not depend on q_ξ, minimizing $\xi \mapsto \mathrm{KL}(q_\xi|\pi)$ is equivalent to maximizing $\xi \mapsto \log \mathrm{Z} - \mathrm{KL}(q_\xi|\pi)$. A standard example is Bayesian inference over latent variables x having a prior density π_0 for a given likelihood function $L(y^{1:n}|x)$ and n observations $y^{1:n} = (y^1, \ldots, y^n)$. The target density is the posterior $p(x|y^{1:n})$ with $U(x) = -\log \pi_0(x) - \log L(y^{1:n}|x)$ and the objective that is commonly maximized,

$$\mathcal{L}(\xi) = \mathbb{E}_{q_\xi(x)}\left[\log \pi_0(x) + \log L(y^{1:n}|x) - \log q_\xi(x)\right] \tag{2}$$

Preprint. Under review.

is called a variational lower bound or ELBO. One of the main features of variational inference methods is their ability to be scaled to large datasets using stochastic approximation methods [23] and applied to non-conjugate models by using Monte Carlo estimators of the gradient [54, 33, 56, 59, 36]. However, the approximation quality hinges on the expressiveness of the distributions in $\mathbb{Q}$ and restrictive assumptions on the variational family that allow for efficient computations such as mean-field families, tend to be too restrictive to recover the target distribution. Constructing an approximation family $\mathbb{Q}$ that is both flexible to closely approximate the density of interest and at the same time computationally efficient has been an ongoing challenge. Much effort has been dedicated to find flexible and rich enough variational approximations, for instance by assuming a Gaussian approximation with different types of covariance matrices. For example, full-rank covariance matrices have been considered in [1, 27, 59] and low-rank perturbations of diagonal matrices in [1, 44, 51, 45]. Finally, covariance matrices with a Kronecker structure have been proposed in [40, 65]. Besides, more complex variational families have been suggested: such as mixture models [17, 21, 44, 38, 37], implicit models [43, 25, 62, 64], where the density of the variational distribution is intractable. Finally, variational inference based on normalizing flows has been developed in [55, 32, 60, 41, 3]. As a special case and motivated by Sklar's theorem [58], variational inference based on families of copula densities and one-dimensional marginal distributions have been considered by [61] where it is assumed that the copula is a vine copula [2] and by [22] where the copula is assumed to be a Gaussian copula together with non-parametric marginals using Bernstein polynomials. Recall that $c : [0,1]^d \to \mathbb{R}_+$ is a copula if and only if its marginals are uniform on $[0,1]$, *i.e.* $\int_{[0,1]^{d-1}} c(u_1, \cdots, u_d) \mathrm{d}x_1 \ldots \mathrm{d}u_{i-1} \mathrm{d}u_{i+1} \cdots \mathrm{d}u_d = \mathbb{1}_{[0,1]}(u_i)$ for any $i \in \{1, \ldots, d\}$ and $u_i \in \mathbb{R}$. In the present work, we pursue these ideas but propose instead of using a family copula densities, simply a family of densities $\{c_\theta : [0,1]^d \to \mathbb{R}_+\}_{\theta \in \Theta}$ on the hypercube $[0,1]^d$. This idea is motivated from the fact that we are able to provide such a family which is both flexible and allow efficient computations.

The paper is organised as follow. In Section 2, we recall how one can sample more expressive distributions and compute their densities using a sequence of bijective and continuously differentiable transformations. In particular, we illustrate how to apply this idea in order to sample from a target density by first sampling a random variable U from its copula density c and then applying the marginal quantile function to each component of U. A new family of copula-like densities on the hypercube is constructed in Section 3 that allow for some flexibility in their dependence structure, while enjoying linear complexity in the dimension of the state space for generating samples and evaluating log-densities. A flexible variational distribution on $\mathbb{R}^d$ is introduced in Section 4 by sampling from such a copula-like density and then applying a sequence of transformations that include $\frac{1}{2} d \log d$ rotations over pairs of coordinates. We illustrate in Section 6 that for some target densities arising for instance as the posterior in a logistic regression model, the proposed density allows for a better approximation as measured by the KL-divergence compared to a Gaussian density. We conclude with applying the proposed methodology on Bayesian Neural Network models.

2 Variational Inference and Copulas

In order to obtain expressive variational distributions, the variational densities can be transformed through a sequence of invertible mappings, termed normalizing flows [56].

To be more specific, assume a series $\{\mathscr{T}_t : \mathbb{R}^d \to \mathbb{R}^d\}_{t=1}^T$ of C^1-diffeomorphisms and a sample $X_0 \sim q_0$, where q_0 is a density function on $\mathbb{R}^d$. Then the random variable $X_T = \mathscr{T}_T \circ \mathscr{T}_{T-1} \circ \cdots \circ \mathscr{T}_1(X_0)$ has a density q_T that satisfies

$$\log q_T(x_T) = \log q_0(x) - \sum_{t=1}^{T} \log \det \left| \frac{\partial \mathscr{T}_t(x_t)}{\partial x_t} \right|, \tag{3}$$

with $x_t = \mathscr{T}_t \circ \mathscr{T}_{t-1} \circ \cdots \circ \mathscr{T}_1(x)$. To allow for scalable inferences with such densities, the transformations $\mathscr{T}_t$ must be chosen so that the determinant of their Jacobians can be computed efficiently. One possibility that satisfies this requirement is to choose volume-preserving flows that have a Jacobian-determinant of one. This can be achieved by considering transformations $\mathscr{T}_t : x \mapsto H_t x$ where H_t is an orthogonal matrix as proposed in [60] using a Householder-projection matrix H_t.

An alternative construction of the same form can be used to construct a density using Sklar's theorem [58, 46]. It establishes that given a target density π on $(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$, there exists a continuous function $C\colon\ [0,1]^d \to [0,1]$ and a probability space supporting a random variable $U = (U_1, \ldots, U_d)$ valued in $[0,1]^d$, such that for any $x \in \mathbb{R}^d$, and $u \in [0,1]^d$,

$$\mathbb{P}\left(U_1 \leqslant u_1, \cdots, U_d \leqslant u_d\right) = C(u_1, \cdots, u_d)\ , \qquad \int_{-\infty}^{x_1} \cdots \int_{-\infty}^{x_d} \pi(t)\mathrm{d}t = C(F_1(x_1), \ldots, F_d(x_d)) \tag{4}$$

where for any $i \in \{1, \ldots, d\}$, F_i is the cumulative distribution function associated with π_i, so for any $x_i \in \mathbb{R}$, $F_i(x_i) = \int_{-\infty}^{x_i} \pi_i(t_i)\mathrm{d}t_i$ and π_i is the i^{th} marginal of π, so for any $x_i \in \mathbb{R}$, $\pi_i(x_i) = \int_{\mathbb{R}^{d-1}} \pi(x)\mathrm{d}x_1 \cdot \mathrm{d}x_{i-1}\mathrm{d}x_{i+1} \ldots \mathrm{d}x_d$. To illustrate how one can obtain such a continuous function C and random variable U, recall that π_i is assumed to be absolutely continuous with respect to the Lebesgue measure. Then for $(X_1, \ldots, X_d) \sim \pi$, the random variable $U = \mathscr{G}^{-1}(X) = (F_1(X_1), \ldots, F_d(X_d))$, where $\mathscr{G}\colon\ [0,1]^d \to \mathbb{R}^d$, with

$$\mathscr{G}\colon u \mapsto (F_1^{-1}(u_1), \ldots, F_d^{-1}(u_d)), \tag{5}$$

follows a law on the hypercube with uniform marginals. It can be readily shown that the cumulative distribution function C of U is continuous and satisfies (4). Note that taking the derivative of (4) yields

$$\pi(x) = c(F_1(x_1), \ldots, F_d(x_d)) \prod_{i=1}^{d} \pi_i(x_i)\ ,$$

where $c(u_1, \ldots, u_d) = \frac{\partial}{\partial u_1} \cdots \frac{\partial}{\partial u_d} C(u_1, \ldots, u_d)$ is a copula density function by definition of C. One possibility to approximate a target density π is then to consider a parametric family of copula density functions $(c_\theta)_{\theta \in \Theta}$ for $\Theta \in \mathbb{R}^{p_c}$ and one parametric family of a d-dimensional vector of density functions $(f_1, \ldots, f_d)_{\phi \in \Phi}\colon \mathbb{R}^d \to \mathbb{R}^d$ for $\Phi \subset \mathbb{R}^{p_f}$, and try to estimate $\theta \in \Theta$ and $\phi \in \Phi$ to get a good approximation of π via variational Bayesian methods. This idea was proposed by [22] and [61], where Gaussian and vine copulas were used, respectively. The main hurdle for using such family is their computational cost which can be prohibitive since the dimension of Θ is of order d^2.

3 Copula-like Density

In this paper, we consider another approach which relies on a copula-like density function on $[0,1]^d$. Indeed, instead of an exact copula density function on $[0,1]^d$ with uniform marginals, we consider simply a density function on $[0,1]^d$ which allows to have a certain degree of freedom in the number of parameters we want to use. The family of copula-like density that we consider are given by

$$c_\theta(v_1, \ldots, v_d) = \frac{\boldsymbol{\Gamma}(\alpha^*)}{\mathrm{B}(a,b)} \left[\prod_{\ell=1}^{d} \left\{\frac{v_\ell^{\alpha_\ell - 1}}{\boldsymbol{\Gamma}(\alpha_\ell)}\right\}\right] (v^*)^{-\alpha^*} \quad \cdot \left(\max_{i \in \{1,\ldots,d\}} v_i\right)^a \left[\left(1 - \max_{i \in \{1,\ldots,d\}} v_i\right)^{b-1}\right]\ , \tag{6}$$

with the notation $v^* = \sum_{i=1}^{d} v_i$ and $\alpha^* = \sum_{i=1}^{d} \alpha_i$. Therefore $\theta = (a, b, (\alpha_i)_{i \in \{1,\ldots,d\}}) \in (\mathbb{R}_+^* \times \mathbb{R}_+^* \times (\mathbb{R}_+^*)^d) = \Theta$.

To sample from the proposed copula-like density, we use the following probabilistic construction, which is proven in Appendix A.

Proposition 1. *Let $\theta \in \Theta$ and suppose that*

1. $(U_1, \ldots, U_d) \sim$ *Dirichlet*$(\alpha_1, \ldots, \alpha_d)$*;*
2. $G \sim$ *Beta*(a, b)*;*
3. $(V_1, \ldots, V_d) = (GU_1/U^*, \ldots, GU_d/U^*)$*, where* $U^* = \max_{i \in \{1,\ldots,d\}} U_i$.

Then the distribution of $(V_1, \ldots, V_d)$ has density with respect to the Lebesgue measure given by (6).

The basic idea is to first sample a random variable $\tilde{U} = (GU_1, \ldots, GU_d)$ from a Beta-Liouville distribution, see [13], having support within the simplex. To obtain a sample on the hypercube, all components of $\tilde{U}$ are subsequently scaled by the inverse of the maximum of the sample $\tilde{U}$.

Now note that also $V^- = (1 - V_1, \ldots 1 - V_d)$ is a sample on the hypercube if $V \sim c_\theta$, as is the convex combination $W = (W_1, \ldots, W_d)$ where $W_i = \delta_i V_i + (1 - \delta_i)(1 - V_i)$ for any $\delta \in [0, 1]^d$. Then, we can write $W = \mathscr{H}(V)$, where

$$\mathscr{H} \colon v \mapsto (1 - \delta) \operatorname{Id} + \{\operatorname{diag}(2\delta) - \operatorname{Id}\} v \ , \tag{7}$$

and Id is the identity operator. It is straightforward to see that $\mathscr{H}$ is a C^1-diffeomorphism for $\delta \in ([0, 1] \backslash \{0.5\})^d$ from the hypercube into $I_1 \times \cdots \times I_d$, where $I_i = [\delta_i, 1 - \delta_i]$ if $\delta_i \in [0, 0.5)$ and $I_i = [1 - \delta_i, \delta_i]$ if $\delta_i \in (0.5, 1]$. Note that the Jacobian-determinant of $\mathscr{H}$ is efficiently computable and is simply equal to $|\prod_{i=1}^{d}(2\delta_i - 1)|$ for $\delta \in [0, 1]^d$.

4 Rotated Variational Density

While one might be tempted to optimize over δ, we found such an approach to be challenging in practice, as for $\delta_i \to 0.5$, the distribution of U approaches a point mass density and the transformation $\mathscr{H}$ becomes non-invertible. An alternative would be to choose a random $\delta \in \{\epsilon, 1 - \epsilon\}^d$ such as $\epsilon = 0.01$ and keep it fixed. Recall that a Dirichlet random variable can be defined as independent Gamma random variables divided by their sum, and so it exhibits negative correlations. While these correlations can be strong in low dimensions, we found them to be too weak in high dimensional settings and samples from c_θ tend to show predominately zero or positive correlations in high dimensions. Samples $U = \mathscr{H}(V)$ for $V \sim c_\theta$ and a random δ allow for either positive or negative correlations. However, the sampled value of δ might not be optimal and so to correct for an improved orientation, we consider additional rotations of the marginals. Rotated copulas have been used before in low dimensions, see for instance [34], however, the set of orthogonal matrices has $d(d - 1)/2$ free parameters.
We consider rotation matrices $\mathcal{R}_d$ that are given as a product of $d/2 \log d$ Givens rotations, following the FFT-style butterfly-architecture proposed in [16], see also [42] and [47] where such an architecture was used for approximating Hessians and kernel functions, respectively. Recall that a Givens rotation matrix [20] is a sparse matrix with one angle as its parameter that rotates two dimensions by this angle. If we assume for the moment that $d = 2^k$, $k \in \mathbb{N}^*$, then we consider k rotation matrices denoted $\mathcal{O}_1, \ldots \mathcal{O}_k$ where for any $i \in \{1, \ldots, k\}$, $\mathcal{O}_i$ contains $d/2$ independent rotations, *i.e.* is the product of $d/2$ independent Givens rotations. Givens rotations are arranged in a butterfly architecture that provides for a minimal number of rotations so that all coordinates can interact with one another in the rotation defined by $\mathcal{R}_d$. For illustration, consider the case $d = 4$, where the rotation matrix is fully described using $4 - 1$ parameters $\nu_1, \nu_2, \nu_3 \in \mathbb{R}$ by $\mathcal{R}_4 = \mathcal{O}_1 \mathcal{O}_2$ with

$$\mathcal{O}_1 \mathcal{O}_2 = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & c_3 & -s_3 \\ 0 & 0 & s_3 & c_3 \end{bmatrix} \begin{bmatrix} c_2 & 0 & -s_2 & 0 \\ 0 & c_2 & 0 & -s_2 \\ s_2 & 0 & c_2 & 0 \\ 0 & s_2 & 0 & c_2 \end{bmatrix} = \begin{bmatrix} c_1 c_2 & -s_1 c_2 & -c_1 s_2 & s_1 s_2 \\ s_1 c_2 & c_1 c_2 & -s_1 s_2 & -c_1 s_s \\ c_3 s_2 & -s_3 s_2 & c_3 c_2 & -s_3 c_s \\ s_3 s_2 & c_3 s_2 & s_3 c_2 & c_3 c_2 \end{bmatrix},$$

where $c_i = \cos(\nu_i)$ and $s_i = \sin(\nu_i)$. We provide a precise recursive definition of $\mathcal{R}_d$ in Appendix B where we also describe the case where d is not a power of two. In general, we have a computational complexity of $O(d \log d)$, due to the fact that $\mathcal{R}_d$ is a product of $\mathcal{O}(\log d)$ matrices each requiring $\mathcal{O}(d)$ operations. Moreover, note that $\mathcal{R}_d$ is parametrized by $d - 1$ parameters $(\nu_i)_{i \in \{1 \ldots d-1\}}$ and each $\mathcal{O}_i$ can be implemented as a sparse matrix, which implies a memory complexity of $\mathcal{O}(d)$. Furthermore, since $\mathcal{O}_i$ is orthonormal, we have $\mathcal{O}_i^{-1} = \mathcal{O}_i^T$ and $|\det \mathcal{O}_i| = 1$.

To construct an expressive variational distribution, we consider as a base distribution q_0 the proposed copula-like density c_θ. We then apply the transformations $\mathscr{T}_1 = \mathscr{H}$ and $\mathscr{T}_2 = \mathscr{G}$. The operator $\mathscr{G}$ in (5) is defined via quantile functions of densities $f_1, \ldots, f_d$, for which we choose Gaussian densities with parameter $\phi_f = (\mu_1, \ldots, \mu_d, \sigma_1^2, \ldots, \sigma_d^2) \in \mathbb{R}^d \times \mathbb{R}_+^d$. As a final transformation, we apply the volume-preserving operator $\mathscr{T}_3 \colon x \mapsto \mathcal{O}_1 \cdots \mathcal{O}_{\log d} x$ that has parameter $\phi_{\mathcal{R}} = (\nu_1, \ldots, \nu_{d-1}) \in \mathbb{R}^{d-1}$. Altogether, the parameter for the marginal-like densities that we optimize over is $\phi = (\phi_f, \phi_{\mathcal{R}})$.

Simulation from the variational density boils down to

1. sampling $(V_1, \ldots, V_d) \sim c_\theta$ using Proposition 1;
2. setting $U = \mathscr{H}(V)$ where $\mathscr{H}$ is defined in (7);
3. setting $X' = \mathscr{G}(U)$, where $\mathscr{G}$ is defined in (5);
4. setting $X = \mathcal{O}_1 \cdots \mathcal{O}_{\log d} X'$.

Note that we apply the rotations after we have transformed samples from the hypercube into $\mathbb{R}^d$, as the hypercube is not closed under Givens rotations. The variational density can then be evaluated using the normalizing flow formula (3). We optimize the variational lower bound $\mathcal{L}$ in (2) using reparametrization gradients, proposed by [33, 56, 59], but with an implicit reparametrization, cf. [14], for Dirichlet and Beta distributions. Such reparametrized gradients for Dirichlet and Beta distributions are readily available for instance in tensorflow probability [9]. Using Monte Carlo samples of unbiased gradient estimates, one can optimize the variational bound using some version of stochastic gradient descent. A more formal description is given in Appendix C.

We would like to remark that such sparse rotations can be similarly applied to proper copulas. While there is no additional flexibility by rotating a full-rank Gaussian copula, applying such rotations to a Gaussian copula with a low-rank correlation yields a Gaussian distribution with a more flexible covariance structure if combined with Gaussian marginals. We leave it for future work to examine if such Gaussian distributions are beneficial for variational inference compared to previously considered Gaussian families.

5 Related Work

Conceptually, our work is closely related to [61, 22]. It differs from [61] in that it can be applied in high dimensions without having to search first for the most correlated variables using for instance a sequential tree selection algorithm [11]. The approach in [22] considered a Gaussian dependence structure, but has only been considered in low-dimensional settings. On a more computational side, our approach is related to variational inference with normalizing flows [55, 32, 60, 41, 3]. In contrast to these works that introduce a parameter-free base distribution commonly in $\mathbb{R}^d$ as the latent state space, we also optimize over the parameters of the base distribution which is supported on the hypercube instead, although distributions supported for instance on the hypersphere as a state space have been considered in [7]. Moreover, such approaches have been often used in the context of generative models using Variational Auto-Encoders (VAEs), yet it is in principle possible to apply the proposed variational copula-like inference in an amortized fashion for VAEs.
Furthermore, a somewhat similar copula-like construction in the context of importance sampling has been proposed in [8]. However, sampling from this density requires a rejection step to ensure support on the hypercube, which would make optimization of the variational bound less straightforward. Lastly, [29] proposed a method to approximate copulas using mixture distributions, but these approximations have not been analysed neither in high dimensions nor in the context of variational inference.

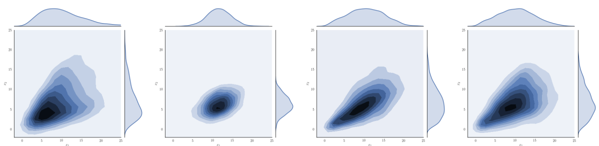
6 Experiments

6.1 Bayesian Logistic Regression

Consider the target distribution π on $(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$ arising as the posterior of a d-dimensional logistic regression, assuming a Normal prior $\pi_0 = \mathcal{N}(0, \tau^{-1} I)$, $\tau = 0.01$, and likelihood function $L(y^i|x) = f(y^i x^\top \mathrm{a}^i)$, $f(z) = \frac{1}{1+\mathrm{e}^{-z}}$ with n observations $y^i \in \{-1, 1\}$ and fixed covariates $\mathrm{a}^i \in \mathbb{R}^d$ for $i \in \{1, \ldots n\}$. We analyse a previously considered synthetic dataset where the posterior distribution is non-Gaussian, yet it can be well approximated with our copula-like construction. Concretely, we consider the synthetic dataset with $d = 2$ as in [48], Section 8.4 and [30] by generating 30 covariates $\mathrm{a} \in \mathbb{R}^2$ from a Gaussian $\mathcal{N}((1, 5)^\top, I)$ for instances in the first class, while we generate 30 covariates from $\mathcal{N}((-5, 1)^\top, 1.1^2 I)$ for instances in the second class. Samples from the target distribution using a Hamiltonian Monte Carlo (HMC) sampler [12, 49] are shown in Figure 1a and one observes non-Gaussian marginals that are positively correlated with heavy right tails. Using a Gaussian variational approximation with either independent marginals or a full covariance matrix as shown in Figure 1b does not adequately approximate the target distribution. Our copula-like construction is able to approximate the target more closely, both without any rotations (Figure 1c) and with a rotation of the marginals (Figure 1d) This is also supported by the ELBO obtained for the different variational families given in Table 1.

Table 1: Comparison of the ELBO between different variational families for the logistic regression experiment.

Variational family	ELBO
Mean-field Gaussian	-3.42
Full-covariance Gaussian	-2.97
Copula-like without rotations	-2.30
Copula-like with rotations	-2.19



(a) (b) (c) (d)

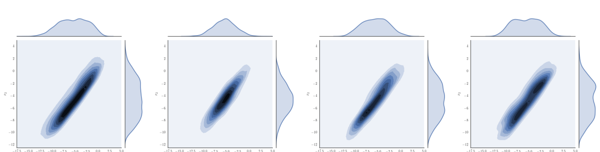
Figure 1: Target density for logistic regression using a HMC sampler in 1a with different variational approximations: Gaussian variational approximation with a full covariance matrix in 1b, copula-like variational approximation without any rotation in 1c and copula-like variational approximation with a rotation in 1d.

6.2 Centred Horseshoe Priors

We illustrate our approach in a hierarchical Bayesian model that posits a priori a strong coupling of the latent parameters. As an example, we consider a Horseshoe prior [6] that has been considered in the variational Gaussian copula framework in [22]. To be more specific, consider the generative model $y|\lambda \sim \mathcal{N}(0, \lambda)$, with $\lambda \sim \mathcal{C}^+(0, 1)$, where $\mathcal{C}^+$ is a half-Cauchy distribution, *i.e.* $X \sim \mathcal{C}^+(0, b)$ has the density $p(x) \propto 1_{\mathbb{R}_+}(x)/(x^2 + b^2)$. Note that we can represent a half-Cauchy distribution with Inverse Gamma and Gamma distributions using $X \sim \mathcal{C}^+(0, b) \iff X^2|Y \sim \mathcal{IG}(\frac{1}{2}, \frac{1}{Y}); Y \sim \mathcal{IG}(\frac{1}{2}, \frac{1}{b^2})$, cf. [50], with a rate parametrisation of the inverse gamma density $p(x) \propto 1_{\mathbb{R}_+}(x) x^{a-1} e^{-b/x}$ for $X \sim \mathcal{IG}(a, b)$. We revisit the toy model in [22] fixing $y = 0.01$. The model thus writes in a centred form as $\eta \sim \mathcal{G}(\frac{1}{2}, 1)$ and $\lambda|\eta \sim \mathcal{IG}(\frac{1}{2}, \eta)$. Following [22], we consider the posterior density on $\mathbb{R}^2$ of the log-transformed variables $(x_1, x_2) = (\log \eta_1, \log \lambda_1)$. In Figure 2, we show the approximate posterior distribution using a Gaussian family (2b) and a copula-like family (2c), together with samples from a HMC sampler (2a). A copula-like density yields a higher ELBO, see Table 2. The experiments in [22] have shown that a Gaussian copula with a non-parametric mixture model fits the marginals more closely. To illustrate that it is possible to arrive at a more flexible variational family by using a mixture of copula-like densities, we have used a mixture of 3 copula-like densities in Figure 2d. Note that it is possible to accommodate multi-modal marginals using a Gaussian quantile transformation with a copula-like density.

Table 2: Comparison of the ELBO between different variational families for the centred horseshoe model.

Variational family	ELBO
Mean-field Gaussian	-1.24
Full-covariance Gaussian	-0.04
Copula-like	0.04
3-mixture copula-like	0.08



(a) (b) (c) (d)

Figure 2: Target density for the horseshoe model using a HMC sampler in 2a with different variational approximations: Gaussian variational approximation with a full covariance matrix in 2b, copula-like variational approximation including a rotation in 2c and a mixture of three copula-like densities with a one rotation and marginal-like density in 2d.

Table 3: Test root mean-squared error for UCI regression datasets. Standard errors in parenthesis.

	Copula-like	Bayes-by-Backprop	SLANG	Dropout
Boston	3.43 (0.22)	3.43 (0.20)	3.21 (0.19)	**2.97 (0.19)**
Concrete	5.76 (0.14)	6.16 (0.13)	5.58 (0.12)	**5.23 (0.12)**
Energy	**0.55 (0.01)**	0.97 (0.09)	0.64 (0.04)	1.66 (0.04)
Kin8nm	**0.08 (0.00)**	**0.08 (0.00)**	**0.08 (0.00)**	0.10 (0.01)
Naval	**0.00 (0.00)**	**0.00 (0.00)**	**0.00 (0.00)**	0.01 (0.01)
Power	**4.02 (0.04)**	4.21 (0.03)	4.16 (0.04)	**4.02 (0.04)**
Wine	0.64 (0.01)	0.64 (0.01)	0.65 (0.01)	**0.62 (0.01)**
Yacht	1.35 (0.08)	1.13 (0.06)	**1.08 (0.09)**	1.11 (0.09)
Protein	**4.20 (0.01)**	NA	NA	4.27 (0.01)

6.3 Bayesian Neural Networks with Normal Priors

We consider an L-hidden layer fully-connected neural network where each layer l, $1 \leqslant l \leqslant L+1$ has width d_l and is parametrised by a weight matrix $W^l \in \mathbb{R}^{d_{l-1} \times d_l}$ and bias vector $b^l \in \mathbb{R}^{d_l}$. Let $\xi \in \mathbb{R}^{d_0}$ denote the input to the network and f be a point-wise non-linearity such as the ReLU function $f(a) = \max\{0, a\}$ and define the activations $a^l \in \mathbb{R}^{d_l}$ by $a^{l+1} = \sum_i h_i^l W_{i\cdot}^l + b^l$ for $l \geqslant 1$, and the post-activations as $h^l = f(a^l)$ for $l \geqslant 2$, and h^1 being the input vector. We consider a regression likelihood function $L(\cdot|a^{L+2}, \sigma) = \mathcal{N}(a^{L+2}, \exp(0.5\sigma))$, and denote the concatenation of all parameters W, b and σ as x. We assume independent Normal priors for the entries of the weight matrix and bias vector with mean 0 and variance σ_0^2. Furthermore, we assume that $\log\sigma \sim \mathcal{N}(0, 16)$. Inference with the proposed variational family is applied on commonly considered UCI regression datasets, repeating the experimental set-up used in [15]. In particular, we use neural networks with ReLU activation functions and one hidden layer of size 50 for all datasets with the exception of the protein dataset that utilizes a hidden layer of size 100. We choose the hyper-parameter $\sigma_0^2 \in \{0.01, 0.1, 1., 10., 100.\}$ that performed best on a validation dataset in terms of its predictive log-likelihood on a validation. Optimization was performed using Adam [31] with a learning rate of 0.002. We compare the predictive performance against Bayes-by-Backprop [5] using a mean-field model, with the results as reported in [45] for a mean-field Bayes-by-Backprop and low-rank Gaussian approximation proposed therein called SLANG. The test root mean-squared errors are given in Table 5 and we provide the test log-likelihood in Appendix D , Table 5. Furthermore, we also report the results for Dropout inference [15]. The proposed approach yields better predictive performance compared to a mean-field approximation and performs similarly to a Gaussian approximation. Dropout tends to perform slightly better. However, we remark that Dropout with a Normal prior and a variational mixture distribution that includes a Dirac delta function as one component gives rise to a different objective, since the prior is not absolutely continuous with respect to the approximate posterior, see [24].

6.4 Bayesian Neural Networks with Structured Priors

We illustrate our approach on a larger Bayesian neural network. To induce sparsity for the weights in the network, we consider a (regularised) Horseshoe prior [53] that has also been used increasingly as an alternative prior in Bayesian neural network to allow for sparse variational approximations, see [39, 18] for mean-field models and [19] for a structured Gaussian approximation. We consider again an L-hidden layer fully-connected neural network where we assume that the weight matrix $W^l \in \mathbb{R}^{d_{l-1} \times d_l}$ for any $l \in \{1, \ldots, L+1\}$ and any $i \in \{1, \ldots, d_{l-1}\}$ satisfies a priori

$$W_{i\cdot}^l | \lambda_i^l, \tau^l, c \sim \mathcal{N}(0, (\tau^l \tilde{\lambda}_i^l)^2 I) \propto \mathcal{N}(0, (\tau^l \lambda_i^l))^2 I)\mathcal{N}(0, c^2), \tag{8}$$

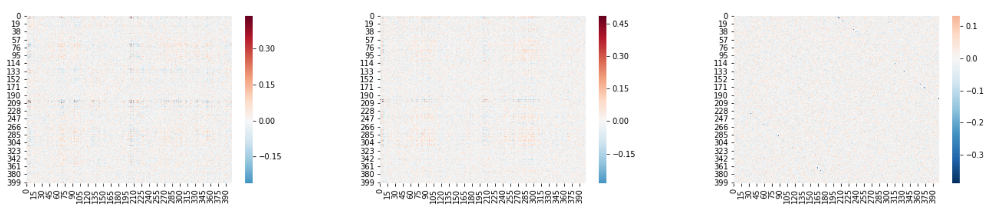
where$(\tilde{\lambda_i}^l)^2 = \frac{c^2(\lambda_i^l)^2}{c^2+\tau^2(\lambda_i^l)^2}$, $\lambda_i^l \sim \mathcal{C}^+(0,1)$, $\tau_i^l \sim \mathcal{C}^+(0, b_\tau)$ and $c^2 \sim \mathcal{IG}(\frac{\nu}{2}, \nu\frac{s^2}{2})$ for some hyper-parameters $b_\tau, \nu, s^2 > 0$. The vector $W_{i\cdot}^{(l)}$ represents all weights that interact with the i-th input neuron. The first Normal factor in (8) is a standard Horseshoe prior with a per layer global parameter τ^l that adapts to the overall sparsity in layer l and shrinks all weights in this layer to zero, due to the fact that $\mathcal{C}^+(0, b_\tau)$ allows for substantial mass near zero. The local shrinkage parameter λ_i^l allow for signals in the i-th input neuron because $\mathcal{C}^+(0,1)$ is heavy-tailed. However, this can leave

Table 4: MNIST prediction errors.

Variational approximation and prior assumptions	Error Rate
Copula-like with horseshoe prior and size 200×200	1.70 %
Mean-field with horseshoe prior and size 200×200	3.82 %
Gaussian of rank $L = 8$ with Gaussian prior and size 400×400, cf. [45]	1.92%
Gaussian of rank $L = 32$ with Gaussian prior and size 400×400, cf. [45]	1.73%
Mean-field (Bayes-by-Backprob) with Gaussian prior and size 400×400, cf. [45]	1.82%
Bayesian Hypernet with weight normalization and size 800×1200, cf. [35]	1.37%

large weights un-shrunk, and the second Normal factor in (8) induces a Student-$t_\nu(0, s^2)$ regularisation for weights far from zero, see [53] for details. We can rewrite the model in a non-centred form [52], where the latent parameters are a priori independent, see also [39, 26, 18, 19] for similar variational approximations. We write the model as $\eta_i^l \sim \mathcal{G}(\frac{1}{2}, 1)$, $\hat{\lambda}_i^l \sim \mathcal{IG}(\frac{1}{2}, 1)$, $\kappa^l \sim \mathcal{G}(\frac{1}{2}, 1/b_\tau^2)$, $\hat{\tau}^l \sim \mathcal{IG}(\frac{1}{2}, 1)$, $\beta_i^l \sim \mathcal{N}(0, I)$, $W_{i\cdot}^l = \tau^l \tilde{\lambda}_i^l \beta_i^l$, $\tau^l = \sqrt{\hat{\tau}^l \kappa^l}$, $\lambda_i^l = \sqrt{\hat{\lambda}_i^l \eta_i^l}$ and $(\tilde{\lambda}_i^l)^2 = \frac{c^2 (\lambda_i^l)^2}{c^2 + (\tau^l)^2 (\lambda_i^l)^2}$. The target density is the posterior of these variables, after applying a log-transformation if their prior is an (inverse) Gamma law.

We performed classification on MNIST using a 2-hidden layer fully-connected network where the hidden layers are of size 200 each. Further details about the algorithmic details are given in Appendix D. We report the prediction errors using a mean-field family and a copula-like family in Table 4, where we also report the prediction errors from [45] for a 400×400 fully-connected network with a Gaussian prior on the weights using a Gaussian variational approximation that either factorises completely or assumes an L-rank plus diagonal structure. Additionally, we include the results of [35] that use RealNVP [10] as normalising flows in a large network that is reparametrised using a weight normalization [57] and becomes scalable by opting to consider only variational inference over the Euclidean norm of $W_{i\cdot}^l$ and performing point estimation for the direction of the weight vector $W_{i\cdot}^l / ||W_{i\cdot}^l||_2$. Like a mean-field model, such a parametrisation does not allow for a flexible dependence structure of the weights within one layer. While we do not claim that the covariance of a copula-like approximation reflects the covariance of the posterior more closely, we have observed experimentally that it allows for some flexibility in modelling the dependence of the weight matrices. We show in Figure 3 the correlation of $W_{\cdot j}^l$, *i.e.* all neurons incident into unit j of layer l for some j and l, where it can be observed that the proposed variational approximation exhibits non-trivial dependence particularly in the last layer.



(a) $\text{Cor}(\text{vec}(W_{\cdot 0}^3, W_{\cdot 1}^3)) - I$ for copula-like density

(b) $\text{Cor}(\text{vec}(W_{\cdot 2}^3, W_{\cdot 3}^3)) - I$ for copula-like density

(c) $\text{Cor}(\text{vec}(W_{\cdot 0}^3, W_{\cdot 1}^3)) - I$ for mean-field density.

Figure 3: Empirical correlation matrix of some columns of the last weight matrix, where the shown weights are incident into the logits of digits 0 and 1 for a copula-like density in 3a, respectively for a mean-field density in 3c and into the logits of digits 2 and 3 in 3b for a copula-like density.

7 Conclusion

We have addressed the challenging problem of constructing a family of distributions that allows for some flexibility in its dependence structure, whilst also having a reasonable computational complexity. It has been shown experimentally that it can constitute a useful replacement of a Gaussian approximation without requiring many algorithmic changes.

Acknowledgements

Alain Durmus acknowledges support from Chaire BayeScale "P. Laffitte". This research has been partly financed by the Alan Turing Institute under the EPSRC grant EP/N510129/1. The authors acknowledge the use of the UCL Legion High Performance Computing Facility (Legion@UCL), and associated support services, in the completion of this work.

References

[1] David Barber and Christopher M Bishop. Ensemble learning for multi-layer networks. In *Advances in neural information processing systems*, pages 395–401, 1998.

[2] Tim Bedford and Roger M Cooke. Probability density decomposition for conditionally dependent random variables modeled by vines. *Annals of Mathematics and Artificial intelligence*, 32(1-4):245–268, 2001.

[3] Rianne van den Berg, Leonard Hasenclever, Jakub M Tomczak, and Max Welling. Sylvester normalizing flows for variational inference. *arXiv preprint arXiv:1803.05649*, 2018.

[4] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

[5] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *Proceedings of The 32nd International Conference on Machine Learning*, pages 1613–1622, 2015.

[6] Carlos M Carvalho, Nicholas G Polson, and James G Scott. The horseshoe estimator for sparse signals. *Biometrika*, 97(2):465–480, 2010.

[7] Tim R Davidson, Luca Falorsi, Nicola De Cao, Thomas Kipf, and Jakub M Tomczak. Hyperspherical variational auto-encoders. *arXiv preprint arXiv:1804.00891*, 2018.

[8] Petros Dellaportas and Mike G Tsionas. Importance sampling from posterior distributions using copula-like approximations. *Journal of Econometrics*, 2018.

[9] Joshua V Dillon, Ian Langmore, Dustin Tran, Eugene Brevdo, Srinivas Vasudevan, Dave Moore, Brian Patton, Alex Alemi, Matt Hoffman, and Rif A Saurous. Tensorflow distributions. *arXiv preprint arXiv:1711.10604*, 2017.

[10] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

[11] Jeffrey Dissmann, Eike C Brechmann, Claudia Czado, and Dorota Kurowicka. Selecting and estimating regular vine copulae and application to financial returns. *Computational Statistics & Data Analysis*, 59:52–69, 2013.

[12] Simon Duane, Anthony D Kennedy, Brian J Pendleton, and Duncan Roweth. Hybrid monte carlo. *Physics letters B*, 195(2):216–222, 1987.

[13] Kai Wang Fang. *Symmetric Multivariate and Related Distributions*. Chapman and Hall/CRC, 2017.

[14] Michael Figurnov, Shakir Mohamed, and Andriy Mnih. Implicit reparameterization gradients. *arXiv preprint arXiv:1805.08498*, 2018.

[15] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rdInternational Conference on Machine Learning*, 2016.

[16] Alan Genz. Methods for generating random orthogonal matrices. *Monte Carlo and Quasi-Monte Carlo Methods*, pages 199–213, 1998.

[17] Samuel J Gershman, Matthew D Hoffman, and David M Blei. Nonparametric variational inference. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 235–242. Omnipress, 2012.

[18] Soumya Ghosh and Finale Doshi-Velez. Model selection in bayesian neural networks via horseshoe priors. *arXiv preprint arXiv:1705.10388*, 2017.

[19] Soumya Ghosh, Jiayu Yao, and Finale Doshi-Velez. Structured variational learning of bayesian neural networks with horseshoe priors. *arXiv preprint arXiv:1806.05975*, 2018.

[20] Gene H Golub and Charles F Van Loan. *Matrix computations*, volume 3. JHU Press, 2012.

[21] Fangjian Guo, Xiangyu Wang, Kai Fan, Tamara Broderick, and David B Dunson. Boosting variational inference. *arXiv preprint arXiv:1611.05559*, 2016.

[22] Shaobo Han, Xuejun Liao, David Dunson, and Lawrence Carin. Variational gaussian copula inference. In *Artificial Intelligence and Statistics*, pages 829–838, 2016.

[23] Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

[24] J Hron, De G Matthews, Z Ghahramani, et al. Variational bayesian dropout: Pitfalls and fixes. In *35th International Conference on Machine Learning, ICML 2018*, volume 5, pages 3199–3219, 2018.

[25] Ferenc Huszár. Variational inference using implicit distributions. *arXiv preprint arXiv:1702.08235*, 2017.

[26] John Ingraham and Debora Marks. Variational inference for sparse and undirected models. In *International Conference on Machine Learning*, pages 1607–1616, 2017.

[27] Tommi Jaakkola and Michael Jordan. A variational approach to bayesian logistic regression models and their extensions. In *Sixth International Workshop on Artificial Intelligence and Statistics*, volume 82, page 4, 1997.

[28] Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

[29] Mohamad A Khaled and Robert Kohn. On approximating copulas by finite mixtures. *arXiv preprint arXiv:1705.10440*, 2017.

[30] Mohammad Emtiyaz Khan, Didrik Nielsen, Voot Tangkaratt, Wu Lin, Yarin Gal, and Akash Srivastava. Fast and scalable bayesian deep learning by weight-perturbation in adam. *arXiv preprint arXiv:1806.04854*, 2018.

[31] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[32] Diederik P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems*, pages 4743–4751, 2016.

[33] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *Proceedings of the 2nd International Conference on Learning Representations (ICLR)*, 2014.

[34] Ioannis Kosmidis and Dimitris Karlis. Model-based clustering using copulas with applications. *Statistics and computing*, 26(5):1079–1099, 2016.

[35] David Krueger, Chin-Wei Huang, Riashat Islam, Ryan Turner, Alexandre Lacoste, and Aaron Courville. Bayesian hypernetworks. *arXiv preprint arXiv:1710.04759*, 2017.

[36] Alp Kucukelbir, Dustin Tran, Rajesh Ranganath, Andrew Gelman, and David M Blei. Automatic differentiation variational inference. *The Journal of Machine Learning Research*, 18(1):430–474, 2017.

[37] Francesco Locatello, Gideon Dresdner, Rajiv Khanna, Isabel Valera, and Gunnar Rätsch. Boosting black box variational inference. *arXiv preprint arXiv:1806.02185*, 2018.

[38] Francesco Locatello, Rajiv Khanna, Joydeep Ghosh, and Gunnar Ratsch. Boosting variational inference: an optimization perspective. In *International Conference on Artificial Intelligence and Statistics*, pages 464–472, 2018.

[39] Christos Louizos, Karen Ullrich, and Max Welling. Bayesian compression for deep learning. In *Advances in Neural Information Processing Systems*, pages 3290–3300, 2017.

[40] Christos Louizos and Max Welling. Structured and efficient variational deep learning with matrix gaussian posteriors. In *Proceedings of the 33rdInternational Conference on Machine Learning*, 2016.

[41] Christos Louizos and Max Welling. Multiplicative normalizing flows for variational bayesian neural networks. In *International Conference on Machine Learning*, pages 2218–2227, 2017.

[42] Michael Mathieu and Yann LeCun. Fast approximation of rotations and hessians matrices. *arXiv preprint arXiv:1404.7195*, 2014.

[43] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. In *International Conference on Machine learning (ICML)*, 2017.

[44] Andrew C Miller, Nicholas J Foti, and Ryan P Adams. Variational boosting: Iteratively refining posterior approximations. In *International Conference on Machine Learning*, pages 2420–2429, 2017.

[45] Aaron Mishkin, Frederik Kunstner, Didrik Nielsen, Mark Schmidt, and Mohammad Emtiyaz Khan. Slang: Fast structured covariance approximations for bayesian deep learning with natural gradient. In *Advances in Neural Information Processing Systems*, pages 6246–6256, 2018.

[46] David S Moore and Marcus C Spruill. Unified large-sample theory of general chi-squared statistics for tests of fit. *The Annals of Statistics*, pages 599–616, 1975.

[47] Marina Munkhoeva, Yermek Kapushev, Evgeny Burnaev, and Ivan Oseledets. Quadrature-based features for kernel approximation. In *Advances in Neural Information Processing Systems*, pages 9165–9174, 2018.

[48] Kevin P Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[49] Radford M Neal et al. Mcmc using hamiltonian dynamics. *Handbook of markov chain monte carlo*, 2(11):2, 2011.

[50] Sarah E Neville, John T Ormerod, MP Wand, et al. Mean field variational bayes for continuous sparse signal shrinkage: pitfalls and remedies. *Electronic Journal of Statistics*, 8(1):1113–1151, 2014.

[51] Victor M-H Ong, David J Nott, and Michael S Smith. Gaussian variational approximation with a factor covariance structure. *Journal of Computational and Graphical Statistics*, 27(3):465–478, 2018.

[52] Omiros Papaspiliopoulos, Gareth O Roberts, and Martin Skold. Non-centred parameterisations for hierarchical models and data augmentation. *Bayesian Statistics*, 7, 2003.

[53] Juho Piironen, Aki Vehtari, et al. Sparsity information and regularization in the horseshoe and other shrinkage priors. *Electronic Journal of Statistics*, 11(2):5018–5051, 2017.

[54] Rajesh Ranganath, Sean Gerrish, and David M Blei. Black box variational inference. In *AISTATS*, pages 814–822, 2014.

[55] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *Proceedings of The 32nd International Conference on Machine Learning*, pages 1530–1538, 2015.

[56] Danilo J Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1278–1286, 2014.

[57] Tim Salimans and Durk P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *Advances in Neural Information Processing Systems*, pages 901–909, 2016.

[58] M Sklar. Fonctions de repartition an dimensions et leurs marges. *Publ. inst. statist. univ. Paris*, 8:229–231, 1959.

[59] Michalis Titsias and Miguel Lázaro-Gredilla. Doubly stochastic variational bayes for non-conjugate inference. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1971–1979, 2014.

[60] Jakub M Tomczak and Max Welling. Improving variational auto-encoders using householder flow. *arXiv preprint arXiv:1611.09630*, 2016.

[61] Dustin Tran, David Blei, and Edo M Airoldi. Copula variational inference. In *Advances in Neural Information Processing Systems*, pages 3564–3572, 2015.

[62] Dustin Tran, Rajesh Ranganath, and David M Blei. Deep and hierarchical implicit models. *arXiv preprint arXiv:1702.08896*, 2017.

[63] Martin J Wainwright and Michael I Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2):1–305, 2008.

[64] Mingzhang Yin and Mingyuan Zhou. Semi-implicit variational inference. *arXiv preprint arXiv:1805.11183*, 2018.

[65] Guodong Zhang, Shengyang Sun, David Duvenaud, and Roger Grosse. Noisy natural gradient as variational inference. *arXiv preprint arXiv:1712.02390*, 2017.

Appendices

A Proof of Proposition 1

Proof. Let $f : \mathbb{R}^d \to \mathbb{R}_+$ be a positive and bounded function. We have by definition, using the expression of the density of the Dirichlet and Beta distributions, see [13], and setting $u_d = 1 - \sum_{i=1}^{d-1} u_i$,

$$\begin{aligned}\mathbb{E}\left[f(V_1,\ldots,V_n)\right] &= \frac{\boldsymbol{\Gamma}(\alpha^\star)}{\mathrm{B}(a,b)} \int_{[0,1]^d} f\left\{gu_1/\max_{j\in\{1,\ldots,d\}} u_j, \ldots, gu_d/\max_{j\in\{1,\ldots,d\}} u_j\right\} \\ &\qquad \times g^{a-1}(1-g)^{b-1}\left\{\prod_{\ell=1}^d \frac{u_\ell^{\alpha_\ell-1}}{\boldsymbol{\Gamma}(\alpha_\ell)}\right\}\mathrm{Leb}(g,u_1,\ldots,u_{d-1}) \\ &= \sum_{k=1}^d \frac{\boldsymbol{\Gamma}(\alpha^\star)}{\mathrm{B}(a,b)} A_k \;, \end{aligned} \tag{9}$$

where

$$\begin{aligned} A_k = \int_{[0,1]^d} \mathbb{1}\left\{u_k = \max_{j\in\{1,\ldots,d\}} u_j\right\} f\left\{gu_1/u_k,\ldots,gu_d/u_k\right\} \\ \times g^{a-1}(1-g)^{b-1}\left\{\prod_{\ell=1}^d \frac{u_\ell^{\alpha_\ell-1}}{\boldsymbol{\Gamma}(\alpha_\ell)}\right\}\mathrm{Leb}(g,u_1,\ldots,u_{d-1}) \;. \end{aligned} \tag{10}$$

Then by symmetry, without loss of generality, we only need to consider A_1. Using the change of variable, $(g,u_1,u_2,\ldots,u_{d-1}) \mapsto (g,u_1,gu_2/u_1,\ldots,gu_{d-1}/u_1)$, which is a C^1-diffeomorphism from $\Delta_1 = \{(g,u_1,\ldots,u_{d-1}) \in [0,1]^d : u_1 = \max_{j\in\{1,\ldots,d\}} u_j\}$ to $\tilde{\Delta}_1 = \{(g,u_1,w_2,\ldots,w_{d-1}) \in [0,1]^d : \max_{j\in\{2,\ldots,d-1\}} w_j \leqslant g, g/u_1 - g - \sum_{j=2}^{d-1} w_j \leqslant g\}$, we get that

$$\begin{aligned} A_1 &= \int_{\Delta_1} f\left\{g,\ldots,gu_d/u_1\right\} g^{a-1}(1-g)^{b-1}\left\{\prod_{\ell=1}^d \frac{u_\ell^{\alpha_\ell-1}}{\boldsymbol{\Gamma}(\alpha_\ell)}\right\}\mathrm{Leb}(g,u_1,w_2,\ldots,w_{d-1}) \\ &= \int_{\tilde{\Delta}_1} f\left\{g,w_2,\ldots,w_{d-1},g/u_1 - g - \sum_{i=2}^{d-1} w_i\right\} g^{a-1}(1-g)^{b-1} \\ &\quad \times \left\{\prod_{\ell=2}^{d-2} \frac{(u_1 w_\ell/g)^{\alpha_\ell-1}}{\boldsymbol{\Gamma}(\alpha_\ell)}\right\} \frac{u_1^{\alpha_1-1}}{\boldsymbol{\Gamma}(\alpha_1)} \frac{(1-u_1-\sum_{i=2}^{d-1} u_1 w_i/g)^{\alpha_d-1}}{\boldsymbol{\Gamma}(\alpha_d)} \frac{g^{d-2}}{u_1^{d-2}}\mathrm{Leb}(g,u_1,w_2,\ldots,w_{d-1}) \\ &= \int_{\tilde{\Delta}_1} f\left\{g,w_2,\ldots,w_{d-1},g/u_1 - g - \sum_{i=2}^{d-1} w_i\right\} g^{a-1}(1-g)^{b-1} \\ &\quad \times \left\{\prod_{\ell=2}^{d-2} \frac{w_\ell^{\alpha_\ell-1}}{\boldsymbol{\Gamma}(\alpha_\ell)}\right\} \frac{u_1^{\alpha^\star-2}}{\boldsymbol{\Gamma}(\alpha_1)} \frac{(g/u_1 - g - \sum_{i=2}^{d-1} w_i)^{\alpha_d}}{\boldsymbol{\Gamma}(\alpha_d)} g^{-\alpha^\star+\alpha_1+1}\mathrm{Leb}(g,u_1,w_2,\ldots,w_{d-1}) \\ &= \int_{\tilde{\Delta}_1} f\left\{g,w_2,\ldots,w_{d-1},g/u_1 - g - \sum_{i=2}^{d-1} w_i\right\} g^{a-1}(1-g)^{b-1} \\ &\quad \times \left\{\prod_{\ell=2}^{d-2} \frac{w_\ell^{\alpha_\ell-1}}{\boldsymbol{\Gamma}(\alpha_\ell)}\right\} \frac{g^{\alpha_1-1}}{\boldsymbol{\Gamma}(\alpha_1)} \frac{(g/u_1 - g - \sum_{i=2}^{d-1} w_i)^{\alpha_d-1}}{\boldsymbol{\Gamma}(\alpha_d)} (u_1/g)^{\alpha^\star-2}\mathrm{Leb}(g,u_1,w_2,\ldots,w_{d-1}) \;. \end{aligned}$$

Now using the change of variable $(g,u_1,w_2,\ldots,w_{d-1}) \mapsto (g, g/u_1 - \sum_{i=2}^{d-1} w_i, w_2,\ldots,w_{d-1}) = (g,w_d,\ldots,w_{d-1})$, which is a C^1-diffeomorphism from $\tilde{\Delta}_1$ to

$$\bar{\Delta}_1 = \left\{(g,w_d,w_2,\ldots,w_{d-1}) : \max_{j\in\{1,\ldots,d\}} w_j \leqslant g\right\} \;,$$

we obtain since $g/u_1 = g + \sum_{j=2}^d w_j$ that

$$A_1 = \int_{\tilde{\Delta}_1} f(g, w_2, \dots, w_{d-1}, w_d)) g^{a-1}(1-g)^{b-1}$$
$$\times \left\{ \prod_{\ell=2}^{d} \frac{w_\ell^{\alpha_\ell - 1}}{\boldsymbol{\Gamma}(\alpha_\ell)} \right\} \frac{g^{\alpha_1}}{\boldsymbol{\Gamma}(\alpha_1)} \left\{ g + \sum_{j=1}^{d-1} w_j \right\}^{-\alpha^\star} \quad \mathrm{Leb}(g, w_1, w_2, \dots, w_{d-1}) \; .$$

Combining this result, (9) and (10) completes the proof. □

B Butterfly rotation matrices

Suppose $d = 2^k$ for some $k \in \mathbb{N}$ and let $c_i = \cos \nu_i$ and $s_i = \sin \nu_i$. For $d = 1$, define $\mathcal{R}_1 = [1]$. Assume $\mathcal{R}_d$ has been defined. Then define

$$\mathcal{R}_{2d} = \begin{bmatrix} \mathcal{R}_d c_d & -\mathcal{R}_d s_d \\ \tilde{\mathcal{R}}_d s_d & \tilde{\mathcal{R}}_d c_d \end{bmatrix},$$

where $\tilde{\mathcal{R}}_d$ has the same form as $\mathcal{R}_d$ except that the c_i and s_i indices are all increased by d. So for instance

$$\mathcal{R}_2 = \begin{bmatrix} c_1 & -s_1 \\ s_1 & c_1 \end{bmatrix} \quad , \quad \tilde{\mathcal{R}}_2 = \begin{bmatrix} c_3 & -s_3 \\ s_3 & c_3 \end{bmatrix}.$$

Suppose now that d is not a power of 2 and let $k = \lceil \log d \rceil$. We construct $\mathcal{R}_d$ as a product of k factors $\mathcal{O}_1 \cdots \mathcal{O}_k$ as used in the construction of $\mathcal{R}_{2^k}$. For any $i \in \{1, \dots k\}$, we then delete from $\mathcal{O}_i$ the last $2^k - d$ rows and columns. Then for every c_i in the remaining $d \times d$ matrix that is in the same column as a deleted s_i is replaced by 1. As an example, for $d = 5$, we have

$$\mathcal{R}_5 = \begin{bmatrix} c_1 & -s_1 & 0 & 0 & 0 \\ s_1 & c_1 & 0 & 0 & 0 \\ 0 & 0 & c_3 & -s_3 & 0 \\ 0 & 0 & s_3 & c_3 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_2 & 0 & -s_2 & 0 & 0 \\ 0 & c_2 & 0 & -s_2 & 0 \\ s_2 & 0 & c_2 & 0 & 0 \\ 0 & s_2 & 0 & c_2 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_4 & 0 & 0 & 0 & -s_4 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ s_4 & 0 & 0 & 0 & c_4 \end{bmatrix}.$$

C Optimization of the variational bound

Recall that for independent random variables $Z_i \sim \mathcal{G}(\alpha_i, 1)$, for $i \in \{1, \dots d\}$, we have $\left(\frac{Z_1}{\sum_{j=1}^d Z_j}, \cdots \frac{Z_d}{\sum_{j=1}^d Z_j} \right) \sim \mathrm{Dirichlet}(\alpha_1, \dots, \alpha_d)$, cf. [13]. Similarly, for independent random variables $Z_{d+1} \sim \mathcal{G}(a, 1)$ and $Z_{d+2} \sim \mathcal{G}(b, 1)$, it holds that $\frac{Z_{d+1}}{Z_{d+1} + Z_{d+2}} \sim \mathrm{Beta}(a, b)$.

Using Proposition 1, we can construct a function $(z, \xi) \mapsto f_\xi(z)$, $z = (z_1, \dots z_{d+2})$, that is almost everywhere continuously differentiable such that $f_\xi(Z_1, \dots Z_{d+2}) \sim q_\xi$, where q_ξ is the density of the proposed variational family with parameter $\xi = (\theta, \phi, \delta)$. Differentiability with respect to ϕ_f can be achieved by a continuous numerical approximation for the quantile function of a standard Gaussian and applying appropriate (re)normalisation. Furthermore, there exists an invertible standardization function $\mathcal{S}_\theta$ with $(z, \theta) \mapsto \mathcal{S}_\theta(z) = (\mathbb{P}(Z_1 \leqslant z_1), \dots, \mathbb{P}(Z_{d+2} \leqslant z_{d+2}))$ continuously differentiable such that $\mathcal{S}_\theta^{-1}(H)$ is equal to $(Z_1, \dots Z_{d+2})$ in distribution, where H is a $(d+2)$-dimensional vector of iid random variables with uniform marginals on $[0, 1]$. In particular, the distribution of H does not depend on ξ. The cumulative distribution function of Z_1 say at the point z_1 is the regularised incomplete Gamma function $\gamma(z_1, \alpha_1)$ that lacks an analytical expression though. However, one can apply automatic differentiation to a numerical method that approximates $\gamma(z_1, \alpha_1)$ yielding an approximation of $\frac{\partial \gamma(z_1, \alpha_1)}{\partial \alpha_1}$. Let us define

$$l(z, \xi) = \frac{\log L(y^{1:n} | f_\xi(z)) + \log \pi_0(f_\xi(z))}{\log q_\xi(f_\xi(z))}.$$

Then $\mathcal{L}(\xi) = \mathbb{E}\left[l(Z, \xi)\right] = \mathbb{E}\left[l(\mathcal{S}_\theta^{-1}(H), \xi)\right]$. Following the arguments in [14], we obtain for the gradient of the variational bound

$$\begin{aligned}\nabla_{\theta,\phi}\mathcal{L}(\xi) &= \mathbb{E}\left[\nabla_{\theta,\phi} l(\mathcal{S}_\theta^{-1}(H), \xi)\right] \\ &= \mathbb{E}\left[\nabla_z l(\mathcal{S}_\theta^{-1}(H), \xi)\nabla_{\theta,\phi}\mathcal{S}_\theta^{-1}(H) + \nabla_{\theta,\phi} l(\mathcal{S}_\theta^{-1}(H), \xi)\right] \\ &= \mathbb{E}\left[\nabla_z l(Z, \xi)\nabla_{\theta,\phi} Z + \nabla_{\theta,\phi} l(Z, \xi)\right], \end{aligned} \tag{11}$$

where $\nabla_\phi Z = 0$ and $\nabla_\theta Z = \nabla_\theta \mathcal{S}_\theta^{-1}(H)|_{H=\mathcal{S}_\theta(Z)}$ can be obtained by implicit differentiation of $S_\theta(Z) = H$ as $\nabla_\theta Z = -(\nabla_\theta \mathcal{S}_\theta(Z))^{-1}\nabla_\theta \mathcal{S}_\theta(Z)$. So for instance $\frac{\partial Z_1}{\partial_{\alpha_1}} = \frac{\partial\gamma(Z_1,\alpha_1)}{\partial_{\alpha_1}}\frac{1}{p_{\alpha_1}(Z_1)}$, with p_{α_1} being the density function of Z_1. We can thus optimize the variational bound using stochastic gradient descent with unbiased samples from (11). We remark that for instance in tensorflow probability [9], such implicit gradients are used by default as long as one simulates from the copula-like density using Proposition 1, implements the density function c_θ from (6) and applies the bijective transformations. In this case, optimization using the proposed density proceeds analogously as if one would use any reparametrisable variational family such as Gaussian distributions.

D Additional details for Bayesian Neural Networks

Table 5: Test log-likelihood for UCI regression datasets. Standard errors in parenthesis.

	Copula-like	Bayes-by-Backprop	SLANG	Dropout
Boston	-2.85 (0.07)	-2.66 (0.06)	-2.58 (0.05)	**-2.46 (0.06)**
Concrete	-3.29 (0.03)	-3.25 (0.02)	-3.13 (0.03)	**-3.04 (0.02)**
Energy	**-1.04 (0.02)**	-1.45 (0.02)	-1.12 (0.01)	-1.99 (0.02)
Kin8nm	**1.08 (0.01)**	**1.07 (0.00)**	1.06 (0.00)	0.95 (0.01)
Naval	**5.74 (0.05)**	4.61 (0.01)	4.76 (0.00)	3.80 (0.01)
Power	-2.82 (0.01)	-2.86 (0.01)	-2.84 (0.01)	**-2.80 (0.01)**
Wine	-1.01 (0.01)	-0.97 (0.01)	-0.97 (0.01)	**-0.93 (0.01)**
Yacht	-2.01 (0.04)	**-1.56 (0.03)**	-1.88 (0.01)	**-1.55 (0.03)**
Protein	-2.87 (0.00)	NA	NA	-2.87 (0.01)

In the MNIST experiments, we train the network on 50000 training points out of 60000 and report the prediction error rates for the test set of 10000 images. We used a batch-size of 200 and used 4 Monte Carlo samples to compute the gradients during training and 100 Monte Carlo samples for the prediction on the test set. We used Adam with a learning rate in $\{0.001, 0.0005, 0.0002\}$ for 20000 iterations. The hyper-parameter for the Horseshoe prior were $\nu = 4$, $s = 1$, so $c \sim \mathcal{IG}(2, 8)$, corresponding to a $t_4(0, 2^2)$ slab. Furthermore, for the global shrinkage factor, we have used $b_\tau \in \{0.1, 1\}$. The variational parameters of the copula-like density are restricted to be positive and we have defined them as the softmax: $x \mapsto \log(\exp(x) + 1)$ of unconstrained parameters, initialised so that $\text{softmax}^{-1}(\alpha_i) \sim \mathcal{N}(1, .01)$, $\text{softmax}^{-1}(a) = 15$ and $\text{softmax}^{-1}(b) = 2$. We have sampled λ_i independently for any $i \in \{1, \ldots, d\}$ such that $\mathbb{P}\left(\lambda_i = \epsilon\right) = \mathbb{P}\left(\lambda_i = 1 - \epsilon\right) = 0.5$ with $\epsilon = 0.01$. We initialised $\nu_i \sim \mathcal{U}(-0.2, 0.2)$ and the log-standard deviations of the marginal-like distribution as $\log \sigma_i = -3$. We aimed for an initial mean of 0 for β_i^l and of -3 for the log of the remaining variables. We therefore choose μ_i so that the quantile of an initial Monte Carlo estimate for the mean of V_i has the desired initial mean.