sion) the activation of hidden units z_j is computed as $z_j = \epsilon_j \cdot f(\mathbf{pa}_j)$ with $\epsilon_j \sim p(\epsilon_j) = \text{Bernoulli}(0.5)$, and where the parameters are learned by following the gradient of the log-likelihood lower bound: $\nabla_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{\epsilon}} \left[\log p_{\boldsymbol{\theta}}(\mathbf{t}^{(i)} | \mathbf{x}^{(i)}, \boldsymbol{\epsilon}) \right]$; this gradient can sometimes be computed exactly (Maaten et al., 2013) and can otherwise be approximated with a Monte Carlo estimate (Hinton et al., 2012). The two parameterizations explained in section 3.1 offer us a useful new perspective on 'dropout'. A 'dropout' hidden unit (together with its injected noise $\boldsymbol{\epsilon}$) can be seen as the DNCP of latent random variables, whose CP is $z_j | \mathbf{pa}_j \sim p_{\boldsymbol{\theta}}(\mathbf{z}_j = \epsilon_j \cdot f(\mathbf{pa}_j) | \mathbf{pa}_j))$. A practical implication is that 'dropout'-type neural networks can therefore be interpreted and treated as hierarchical Bayes nets, which opens the door to alternative approaches to learning the parameters, such as Monte Carlo EM or variational methods.

While 'dropout' is designed as a regularization method, other work on stochastic neural networks exploit the power of stochastic hidden units for generative modeling, e.g. (Frey & Hinton, 1999; Rezende et al., 2014; Tang & Salakhutdinov, 2013) applying (partially) MCMC or (partically) factorized variational approaches to modelling the posterior. As we will see in sections 4 and 6, the choice of parameterization has a large impact on the posterior dependencies and the efficiency of posterior inference. However, current publications lack a good justification for their choice of parameterization. The analysis in section 4 offers some important insight in where the centered or non-centered parameterizations of such networks are more appropriate.

3.4. A differentiable MC likelihood estimator

We showed that many hierarchical continuous latent-variable models can be transformed into a DNCP $p_{\boldsymbol{\theta}}(\mathbf{x}, \boldsymbol{\epsilon})$, where all latent variables (the introduced auxiliary variables $\boldsymbol{\epsilon}$) are root nodes (see eq. (8)). This has an important implication for learning since (contrary to a CP) the DNCP can be used to form a differentiable Monte Carlo estimator of the marginal likelihood:

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}) \simeq \log \frac{1}{L} \sum_{l=1}^{L} \prod_{j} p_{\boldsymbol{\theta}}(\mathbf{x}_j | \mathbf{pa}_j^{(l)})$$

where the parents $\mathbf{pa}_j^{(l)}$ of the observed variables are either root nodes or functions of root nodes whose values are sampled from their marginal: $\boldsymbol{\epsilon}^{(l)} \sim p(\boldsymbol{\epsilon})$. This MC estimator can be differentiated w.r.t. $\boldsymbol{\theta}$ to obtain an MC estimate of the log-likelihood gradient $\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\mathbf{x})$, which can be plugged into stochastic optimization methods such as Adagrad for approximate ML or MAP. When performed one datapoint at a time, we arrive at our on-line Maximum Monte Carlo Likelihood (MMCL) algorithm.

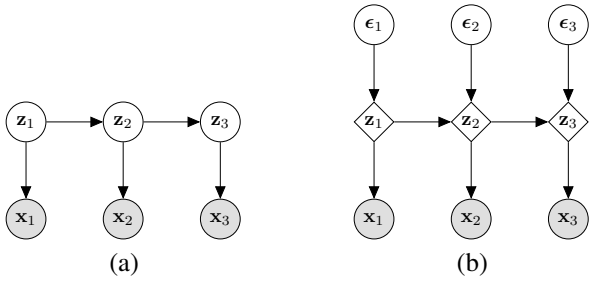

Figure 2. **(a)** An illustrative hierarchical model in its centered parameterization (CP). **(b)** The differentiable non-centered parameterization (DNCP), where $\mathbf{z}_1 = g_1(\boldsymbol{\epsilon}_1, \boldsymbol{\theta})$, $\mathbf{z}_2 = g_2(\mathbf{z}_1, \boldsymbol{\epsilon}_2, \boldsymbol{\theta})$ and $\mathbf{z}_3 = g_3(\mathbf{z}_2, \boldsymbol{\epsilon}_3, \boldsymbol{\theta})$, with auxiliary latent variables $\boldsymbol{\epsilon}_k \sim p_{\boldsymbol{\theta}}(\boldsymbol{\epsilon}_k)$. The DNCP exposes a neural network within the hierarchical model, which we can differentiate efficiently using back-propagation.

Table 1. Limiting behaviour of squared correlations between z and its parent y_i when z is in the centered (CP) and non-centered (DNCP) parameterizaton.

	$\rho^2_{y_i,z}$ (CP)	$\rho^2_{y_i,e}$ (DNCP)
$\lim_{\sigma \to 0}$	1	0
$\lim_{\sigma \to +\infty}$	0	$\frac{\beta w_i^2}{\beta w_i^2 + \alpha}$
$\lim_{\beta \to 0}$	$\frac{w_i^2}{w_i^2 - \alpha\sigma^2}$	0
$\lim_{\beta \to -\infty}$	0	1
$\lim_{\alpha \to 0}$	$\frac{1}{1 - \beta\sigma^2}$	$\frac{\beta\sigma^2}{\beta\sigma^2 - 1}$
$\lim_{\alpha \to -\infty}$	0	0

4. Effects of parameterizations on posterior dependencies

What is the effect of the proposed reparameterization on the efficiency of inference? If the latent variables have linear-Gaussian conditional distributions, we can use the metric of squared correlation between the latent variable and any of its children in their posterior distribution. If after reparameterization the squared correlation is decreased, then in general this will also result in more efficient inference.

For non-linear Gaussian conditional distributions, the log-PDF can be locally approximated as a linear-Gaussian using a second-order Taylor expansion. Results derived for the linear case can therefore also be applied to the non-linear case; the correlation computed using this approximation is a local dependency between the two variables.

Denote by z a scalar latent variable we are going to reparameterize, and by $\mathbf{y}$ its parents, where y_i is one of the parents. The log-PDF of the corresponding conditional dis-