Statistics and Computing (2024) 34:164
https://doi.org/10.1007/s11222-024-10481-x

ORIGINAL PAPER

# Learning variational autoencoders via MCMC speed measures

Marcel Hirt[1] · Vasileios Kreouzis[2] · Petros Dellaportas[2,3]

Received: 24 August 2023 / Accepted: 25 July 2024
© The Author(s) 2024

**Abstract**

Variational autoencoders (VAEs) are popular likelihood-based generative models which can be efficiently trained by maximising an evidence lower bound. There has been much progress in improving the expressiveness of the variational distribution to obtain tighter variational bounds and increased generative performance. Whilst previous work has leveraged Markov chain Monte Carlo methods for constructing variational densities, gradient-based methods for adapting the proposal distributions for deep latent variable models have received less attention. This work suggests an entropy-based adaptation for a short-run metropolis-adjusted Langevin or Hamiltonian Monte Carlo (HMC) chain while optimising a tighter variational bound to the log-evidence. Experiments show that this approach yields higher held-out log-likelihoods as well as improved generative metrics. Our implicit variational density can adapt to complicated posterior geometries of latent hierarchical representations arising in hierarchical VAEs.

**Keywords** Adaptive MCMC · Generative models · Hierarchical models · HMC · MALA

## 1 Introduction

VAEs (Kingma and Welling 2014; Rezende et al. 2014) are powerful latent variable models that routinely use neural networks to parameterise conditional distributions of observations given a latent representation. This renders such models' maximum likelihood estimation (MLE) intractable, so one commonly resorts to extensions of expectation-maximisation (EM) approaches that maximise a lower bound on the data log-likelihood. These objectives introduce a variational or encoding distribution of the latent variables that approximates the true posterior distribution of the latent variable given the observation. However, VAEs have shortcomings; for example, they can struggle to generate high-quality images. These shortcomings have been attributed to failures to match corresponding distributions in the latent space. First, the VAE prior can significantly differ from the aggregated approximate posterior (Hoffman and Johnson 2016; Rosca et al. 2018). To alleviate this prior hole phenomenon, previous work has considered more flexible priors, such as mixtures (Tomczak and Welling 2017), normalising flows (Kingma et al. 2016), hierarchical priors (Sønderby et al. 2016; Klushyn et al. 2019), energy-based models (Du and Mordatch 2019; Aneja et al. 2021) or diffusion models (Vahdat et al. 2021; Sinha et al. 2021). Second, the encoding distribution can be significantly different from the true posterior distribution. It has been an ongoing challenge to reduce this approximation error by constructing new flexible variational families such as parametric constructions (Barber and Bishop 1998; Tran et al. 2015; Han et al. 2016; Guo et al. 2016; Abadi et al. 2016; Louizos and Welling 2016; Locatello et al. 2018; Louizos and Welling 2017), with normalising flows (Rezende and Mohamed 2015; Kingma et al. 2016; Papamakarios et al. 2019) being a popular example. Other works resort to auxiliary variables (Ranganath et al. 2016) with implicit (Tran et al. 2017; Mescheder et al. 2017) or semi-implicit (Yin and Zhou 2018; Molchanov et al. 2019; Titsias and Ruiz 2019; Yu et al. 2023) models that require appropriate adjustments to the variational objectives.

This work utilises adaptive MCMC kernels to construct an implicit variational distribution, that, by the reversibility of the associated Markov kernel, decreases an upper bound of the Kullback–Leibler (KL) between an initial encod-

✉ Petros Dellaportas
p.dellaportas@ucl.ac.uk

[1] School of Social Sciences and School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore, Singapore

[2] Department of Statistical Science, University College London, London, UK

[3] Department of Statistics, Athens University of Economics and Business, Athina, Greece

Published online: 06 August 2024

ing distribution and the true posterior. In summary, this paper (i) develops gradient-based adaptive MCMC methods that give rise to flexible implicit variational densities for training VAEs; (ii) shows that non-diagonal preconditioning schemes are beneficial for learning hierarchical structures within VAEs; and (iii) illustrates the improved generative performance for different data sets and MCMC schemes. Our code is available at https://github.com/kreouzisv/smvaes.

## 2 Background

We are interested in learning deep generative latent variable models using VAEs. Let $\mathsf{X} \subset \mathbb{R}^{d_x}$, $\mathsf{Z} \subset \mathbb{R}^{d_z}$ and assume some *prior* density $p_\theta(z)$ for $z \in \mathsf{Z}$, with all densities assumed with respect to the Lebesgue measure. The prior density can be fixed or made dependent on some parameters $\theta \in \Theta$. Consider a conditional density $p_\theta(x|z)$, also called *decoder*, with $z \in \mathsf{Z}$, $x \in \mathsf{X}$ and parameters also denoted $\theta$. We can interpret this decoder as a generative network that tries to explain a data point $x$ using a latent variable $z$. This latent structure yields the following generative distribution of the data

$$p_\theta(x) = \int_{\mathsf{X}} p_\theta(x|z) p_\theta(z) \mathrm{d}z.$$

Assume a ground truth measure $p_d$ on $\mathsf{X}$, which can be seen as the empirical distribution of some observed data set. We want to maximise the log-likelihood with respect to $p_d$, *i.e.* $\max_{\theta \in \Theta} \int_{\mathcal{X}} \log p_\theta(x) p_d(\mathrm{d}x)$. Variational inference approaches for maximising this log-likelihood proceed by introducing so-called *encoder* distributions $q_\phi(z|x)$ with parameter $\phi \in \Phi$. These encoder distributions can be used to construct a tractable surrogate objective which minorises the log-likelihood and becomes tight if the encoder distribution coincides with the posterior distribution. In particular, letting $\{q_\phi(z|x) \colon \phi \in \Phi\}$ be a parameterised family of encoders, one can define the so-called *evidence lower bound* (ELBO),

$$\mathcal{L}(\theta, \phi, x) = \mathbb{E}_{q_\phi(z|x)} \left[ \log p_\theta(x|z) \right] - \mathrm{KL}(q_\phi(z|x) | p_\theta(z))$$

averaged over $x \sim \mu$. Here, $\mathrm{KL}(q(z)|p(z)) = \int_{\mathsf{Z}} q(z) (\log q(z) - \log p(z)) \,\mathrm{d}z \geq 0$ denotes the Kullback–Leibler divergence between two densities $q$ and $p$. Recalling the *posterior* density $p_\theta(z|x) \propto p_\theta(z) p_\theta(x|z)$, one can see directly that the ELBO constitutes a surrogate objective that minorises the log-likelihood,

$$\mathcal{L}(\theta, \phi, x) = \log p_\theta(x) - \mathrm{KL}(q_\phi(z|x) | p_\theta(z|x)).$$

## 3 Related work

Many approaches have been proposed for combining MCMC with variational inference. Salimans et al. (2015) and Wolf et al. (2016) construct a variational bound on an extended state space that includes multiple samples of the Markov chain. This was extended in Caterini et al. (2018) using tempering and illustrated connections with SMC samplers. Instead of considering variational objectives on augmented state spaces, our approach follows more closely the work of Hoffman (2017), Levy et al. (2018), Hoffman et al. (2019). In particular, we follow their approach to estimate the gradients of the decoder parameters and the initial variational distribution. However, our approach considers an unexplored gradient-based adaptation of the Markov chain that also allows us to learn, for instance, non-diagonal preconditioning matrices. Titsias (2017) suggested a model reparameterisation using a transport mapping, while Ruiz and Titsias (2019) suggested using a variational contrastive divergence instead of a KL divergence used herein. Thin et al. (2020) presented a variational objective on an extended space of the accept/reject variables that allows for entropy estimates of the distribution of the final state of the Markov chain. Nijkamp et al. (2020) have used short-run MCMC approximations based on unadjusted Langevin samplers to train multi-layer latent variable models without learning an auto-encoding model, with extensions to learn energy-based priors in Pang et al. (2020). We follow the approach in Pang et al. (2020) for learning the generative parameters, with the difference being that we utilise adaptive and Metropolis-adjusted algorithms, instead of unadjusted Langevin samplers. Ruiz et al. (2021) used couplings for Markov chains to construct unbiased estimates of the marginal log-likelihood for VAEs. The introduced Markov chains are samples from an extension of the iterated sampling importance resampling algorithm (Andrieu et al. 2010) and target an augmented posterior distribution for the IWAE bound (Burda et al. 2015). Non-adaptive MCMC transitions have also been utilised in Wu et al. (2020) to build stochastic normalising flows that approximate the posterior distribution in VAEs, but are trained by minimising a KL divergence between the forward and backward path probabilities, see also Hagemann et al. (2022). More recently, Taniguchi et al. (2022) considered an amortised energy function over the encoder parameters and used a MALA algorithm to sample from its invariant distribution. In particular, they considered a MALA algorithm that operates on the parameter space of the encoder parameters instead of utilising MCMC algorithms in the latent space as in our work. Peis et al. (2022) learn an initial encoding distribution based on a sliced Kernel Stein Discrepancy and then apply a non-adapted HMC algorithm.

## 4 Training VAEs with MCMC speed measures

In this work, we will construct an approximation to the posterior distribution $p_\theta(z|x)$ by first sampling from an initial tractable distribution $q^0_{\phi_0}(z|x)$ and then recursively update $z$ by applying a sequence of $K$ Markov kernels. More precisely, for $x \in \mathcal{X}, \theta \in \Theta, \phi \in \Phi$, let $M^k_{\theta,\phi_k}(\cdot|x)$ denote a parameterised Markov kernel which is reversible with respect to the posterior $p_\theta(z|x)$. We then define the following variational family

$$\mathcal{Q}_x = \{q^K_{\theta,\phi}(\cdot|x) = q^0_{\phi_0}(\cdot|x) M^1_{\theta,\phi_1}(\cdot|x) \cdots M^K_{\theta,\phi_K}(\cdot|x)\ ,\\ \phi_k \in \Phi_k, \phi = (\phi_0, \ldots, \phi_K), \theta \in \Theta\},$$

where $(qM)(z'|x) = \int_{\mathsf{Z}} q(z|x) M(z, z'|x) \mathrm{d}z$ for a conditional density $q(\cdot|x)$ and Markov kernel $M(\cdot|x)$ that depends on $x$. Although $q^K_{\theta,\phi}$ can be evaluated explicitly for the choice of Markov kernels considered here (Thin et al. 2020), we do not require this. Instead, we rely on the fact (Ambrosio et al. 2005), Lemma 9.5.4, that due to the reversibility of the Markov kernels with respect to $p_\theta(z|x)$, it holds that

$$\mathrm{KL}\left(q^K_{\theta,\phi}(z|x) | p_\theta(z|x)\right) \leq \mathrm{KL}\left(q^0_{\phi_0}(z|x) | p_\theta(z|x)\right). \tag{1}$$

The non-asymptotic convergence of the Markov chain depends on the posterior distribution as well as on the specific MCMC algorithm used, see for example Dwivedi et al. (2019), Mangoubi and Vishnoi (2019), Chewi et al. (2021), Wu et al. (2022), Altschuler and Chewi (2023), Chen and Gatmiry (2023) for the MALA case and (Chen et al. 2019b; Lee et al. 2020, 2021) for HMC often under convexity or smoothness with isoperimetry assumptions.

### 4.1 Learning the warm start distribution

We consider first a standard ELBO

$$\mathcal{L}_0(\theta, \phi_0, x) \tag{2}$$
$$= \mathbb{E}_{q^0_{\phi_0}(z|x)}\left[\log p_\theta(x|z)\right] - \mathrm{KL}(q^0_{\phi_0}(z|x)|p(z)).$$

Relation (1) motivates to learn $\phi_0$ by maximising $\mathcal{L}_0$ $(\theta, \phi_0, x)$. Indeed, due to

$$\mathrm{KL}\left(q^K_{\theta,\phi}(z|x) | p_\theta(z|x)\right) \leq \mathrm{KL}\left(q^0_{\phi_0,}(z|x) | p_\theta(z|x)\right)\\ = \log p_\theta(x) - \mathcal{L}_0(\theta, \phi_0, x),$$

maximising $\mathcal{L}_0(\theta, \phi_0, x)$ decreases an upper bound of the KL divergence between the variational density $q^K_{\theta,\phi}(\cdot|x)$ and the posterior density for fixed $\theta$ and $\phi_1, \ldots \phi_K$. While decreasing an upper bound of the KL divergence may not necessarily decrease the actual KL divergence, we found this choice to work well in practice. It also allows to utilised pre-trained encoding distributions with parameters $\phi_0$ from a standard VAE as a parameter initialisation. On a high level, upper and lower bounds on the mixing time of MALA or HMC for log-concave targets hinge on a well-chosen *warm* initial distribution, as well as a small condition number of the target distribution, adjusted for the preconditioning matrix of the sampler, see for example Wu et al. (2022); Altschuler and Chewi (2023). For $m$-strongly convex and $L$-smooth targets $p_\theta(z|x)$, one can obtain, see Dwivedi et al. (2019), a $\beta$-warm distribution $q^0_{\phi_0}(\cdot|x) = \mathcal{N}(\mu^\star, L^{-1}\,\mathrm{I})$, i.e. it holds that

$$\sup_A \frac{\int_A q^0_{\phi_0}(z|x)\mathrm{d}z}{\int p_\theta(z|x)\mathrm{d}z} \leq \beta$$

over all measurable sets $A$ for $\beta = \kappa^{d_z/2}$, with condition number $\kappa = L/m$, where $\mu^\star$ is the mode of $p_\theta(z|x)$. By optimising the bound $\mathcal{L}_0(\theta, \phi_0, x)$, we can expect to find parameters $\phi_0$ so that the mean of the variational distribution is close to a mode of the true posterior. Because acceptance probabilities in regions that are unlikely under the target $p_\theta(z|x)$ lead to small acceptance probabilities, warm start distributions lead to faster convergence as they avoid such bottlenecks in the state space. One can obtain warm starts by controlling the forward chi-squared divergence

$$\chi^2(q^0_{\theta,\phi^0}(z|x)|p_\theta(z|x)) = \int_{\mathsf{Z}} \left(\frac{p_\theta(z|x)}{q^0_{\phi_0}(z|x)}\right)^2 p_\theta(z|x))\mathrm{d}z,$$

or more generally a Reny divergence of order strictly larger than one, see Altschuler and Chewi (2023). Such objectives are more challenging to optimise, with variational approaches typically requiring multiple Monte Carlo samples (Hernandez-Lobato et al. 2016; Finke and Thiery 2019; Geffner and Domke 2021; Li et al. 2023). It may be of interest to explore in future work such different variational objectives for learning $\phi_0$.

### 4.2 Markov kernels

We also need to specify the Markov kernels. We use *reparameterisable* Metropolis-Hastings kernels with the potential function $U_\theta(z|x) = -\log p_\theta(x|z) - \log p_\theta(z)$ corresponding to the target $\pi_\theta(z) = p_\theta(z|x) \propto \exp(-U_\theta(z|x))$. More precisely, for $A \in \mathcal{B}(\mathsf{Z})$,

$$M^k_{\theta,\phi_1}(z, A|x) = \int_{\mathsf{Z}} \nu(\mathrm{d}v)\Big[\left(1 - \alpha(z, z')\right)\delta_z(A)\\ + \alpha(z, z'|x)\delta_{z'}(A)\Big]_{z' = \mathcal{T}_{\theta,\phi_1}(v|z,x)}\\ = \int_{\mathsf{Z}} \Big[\left(1 - \alpha(z, z')\right)\delta_z(A)$$

$$+ \alpha(z, z'|x)\delta_{z'}(A)\Big] r_{\theta,\phi_1}(z, z'|x)\mathrm{d}z'$$

where $\alpha(z, z'|x)$ is an *acceptance rate* for moving from state $z$ to $z'$, $\mathcal{T}_{\theta,\phi_1}(\cdot|z, x)$ is a *proposal mapping* and $\nu$ is a parameter-free density over Z. Expressing the proposal density $r_{\theta,\phi_1}(z, z'|x)$ through a proposal mapping $\mathcal{T}_{\theta,\phi_1}(\cdot|z, x)$ having as input a parameter-free variable $v$ with density $\nu$, allows us to apply the reparameterisation trick (Kingma and Welling 2014; Rezende et al. 2014; Titsias and Lázaro-Gredilla 2014). Although the different Markov kernels could have different parameters $\phi_k$ for $k \in \{1, \ldots, K\}$, we assume for simplicity that they all share the parameters $\phi_1$, thereby helping the method to scale more easily to large values of $K$.

### 4.3 Speed measure adaptation

For a random walk Markov chain with isotropic proposal density $r(z, \cdot|x) = \mathcal{N}(z, \sigma^2 \mathrm{I})$ at position $z$, the speed measure (Roberts et al. 1997) is defined as $\sigma^2 \times \alpha(z|x)$, where $a(z|x) = \int \alpha(z, z'|x) r(z, z'|x)\mathrm{d}z'$ is the average acceptance rate. To encourage fast mixing for the Markov chain across all dimensions jointly, Titsias and Dellaportas (2019) suggested a generalisation of this speed measure that amounts to choosing the parameters $h$ and $C$ from the proposal so that the proposal has both high acceptance rates, but also a high entropy

$$\mathcal{H}_{\theta,\phi_1} = -\int_{\mathrm{Z}} r_{\theta,\phi_1}(z, z'|x) \log r_{\theta,\phi_1}(z, z'|x)\mathrm{d}z'.$$

More precisely, we consider the generalised speed measure

$$s_{\theta,\phi_1}(z|x) = e^{\beta \mathcal{H}_{\theta,\phi_1}} \times a(z|x)$$

for some hyper-parameter $\beta > 0$. While maximising $s_{\theta,\phi_1}(z|x)$, or equivalently,

$$\log s_{\theta,\phi_1}(z|x) = \log \left[ \int_{\mathrm{Z}} \alpha(z, \mathcal{T}_{\theta,\phi_1}(v|z, x))\nu(\mathrm{d}v) \right] + \beta \mathcal{H}_{\theta,\phi_1},$$

is intractable, we follow Titsias and Dellaportas (2019) and maximise a lower bound thereof due to Jensen's inequality,

$$\log s_{\theta,\phi_1}(z|x) \geq \mathcal{F}(\phi_1, z, x) = \left[ \int_{\mathrm{Z}} \log \alpha(z, \mathcal{T}_{\theta,\phi_1}(v|z, x))\nu(\mathrm{d}v) + \beta \mathcal{H}_{\theta,\phi_1} \right],$$

averaged over $(x, z) \sim \mu(x) q^0_{\phi_0}(z|x)$ where $\beta > 0$ is some hyper-parameter that can be updated online to achieve a desirable average acceptance rate $\alpha^\star$.

### 4.4 MALA

Consider first a Metropolis Adjusted Langevin Algorithm (MALA). We assume that $\phi_1$ parameterises a non-singular matrix $C$, possibly dependent on $x$, which can be, for instance, a diagonal matrix or a Cholesky factor. In this case, we can write the proposed state $z'$ as

$$z' = \mathcal{T}_{\theta,\phi_1}(v|z, x) = z - \frac{h^2}{2} C C^\top \nabla U_\theta(z|x) + hCv \tag{3}$$

for some step size $h > 0$ that is part of the parameter $\phi_1$ and where $v \sim \nu = \mathcal{N}(0, \mathrm{I})$. The log-acceptance rate is $\log a(z, z') = \min\{0, -\Delta(v, z, z')\}$ based on the energy error

$$\Delta(v, z, z') = U_\theta(z'|x) - U_\theta(z|x) - \frac{1}{2}\|v\|^2 + \frac{1}{2}\left\| v - \frac{h}{2} C \left\{ \nabla U_\theta(z|x) + \nabla U_\theta(z'|x) \right\} \right\|^2,$$

evaluated at $z' = \mathcal{T}_{\theta,\phi_1}(v|z, x)$. The proposal density of the Markov kernel

$$r_{\theta,\phi_1}(z, z'|x) = \mathcal{N}\left( z - \frac{h^2}{2} C C^\top \nabla U_\theta(z|x), h^2 C C^\top \right)$$

can be viewed as the pushforward density of $\mathcal{N}(0, \mathrm{I})$ with respect to the transformation $\mathcal{T}_{\theta,\phi_1}(v|z, x)$. Its entropy is

$$\mathcal{H}_{\theta,\phi_1} = -\int_{\mathrm{Z}} r_{\theta,\phi_1}(z, z'|x) \log r_{\theta,\phi_1}(z, z'|x)\mathrm{d}z' = \mathrm{const} + \log|\det(hC)|,$$

which is constant for $z \in \mathrm{Z}$ in the standard MALA case, although it can depend on $x$ for MALA with state-dependent proposals.

### 4.5 HMC

Consider next a Hamiltonian Monte Carlo Algorithm (HMC) based on a leapfrog or velocity Verlet integrator with $L$ steps (Hairer et al. 2003; Bou-Rabee and Sanz-Serna 2018). We assume that $\phi_1$ parameterises a Cholesky factor matrix $C$ of the inverse *mass matrix* $M^{-1} = CC^\top$. The proposed state $z' = q_L$ is commonly computed recursively for $\ell \in \{1, \ldots, L\}$ via

$$p_{\ell+\frac{1}{2}} = p_\ell - \frac{1}{2}\nabla U_\theta(q_\ell|x)$$
$$q_{\ell+1} = q_\ell + hM^{-1} p_{\ell+\frac{1}{2}}$$
$$p_{\ell+1} = p_{\ell+\frac{1}{2}} - \frac{1}{2}\nabla U_\theta(q_{\ell+1}|x),$$

where $p_\ell$ is a sequence of *momentum* variables initialised at $p_0 = C^{-\top} v$ for $v \sim \mathcal{N}(0, \mathrm{I})$. It is possible (Livingstone et al. 2019; Durmus et al. 2017) to write the proposed state $z' = \mathcal{T}_{\theta,\phi_1}(v|z,x)$ in the representation

$$z' = z - \frac{Lh^2}{2} CC^\top \nabla U_\theta(z|x) + LhCv - h^2 CC^\top \Xi_L(v)$$

where

$$\Xi_L(v) = \sum_{\ell=1}^{L-1} (L-\ell) \nabla U_\theta(q_\ell) \tag{4}$$

is a weighted average of the potential energy gradients along the leapfrog trajectory. Consequently, the proposal density can be written as

$$\begin{aligned}\log r_{\theta,\phi_1}(z, \mathcal{T}_L(v)) = &\log \nu(v) - d \log L - \log |\det C| \\ &- \log |\det \mathrm{D}\Xi_L(v)| \,,\end{aligned}$$

where $\mathrm{D}\Xi_L(v)$ is the Jacobian of the non-linear function $\Xi_L$ in (4). However, the computational complexity of evaluating the log-determinant of the Jacobian of $\Xi_L$ scales poorly for high dimensional latent variables. We, therefore, consider the approximation suggested in Hirt et al. (2021) based on a local Gaussian assumption that the Hessian of the potential function $U_\theta$ along the leapfrog trajectory can be approximated by its value at the mid-point $q_{\lfloor L/2 \rfloor}$ of the trajectory. Under this assumption, the log-determinant of the Jacobian can be written as

$$\begin{aligned}&\log |\det \mathrm{D}\Xi_L(v)| \\ &\approx \log \left| \det \left( \mathrm{I} - \frac{L^2-1}{6} C^\top \nabla^2 U_\theta(q_{\lfloor L/2 \rfloor}|x) C \right) \right| ,\end{aligned}$$

which can be estimated by resorting to Russian roulette estimators (Behrmann et al. 2019; Chen et al. 2019a). The above approximation becomes exact for Gaussian targets with covariance matrix $\Sigma$, since $\nabla^2 U_\theta(q) = \Sigma^{-1}$ for any point $q$ in the state space.

### 4.6 Learning the generative model

Maximizing the log-likelihood function directly using

$$\nabla_\theta \log p_\theta(x) = \int_{\mathsf{Z}} p_\theta(z|x) \nabla_\theta \log p_\theta(x,z) \mathrm{d}z$$

is usually intractable as it requires samples from $p_\theta(z|x)$. On the other hand, optimizing the generative parameters by optimizing the classic variational bound $\mathcal{L}_0(\theta, \phi_0, x)$ based on the initial variational distribution does not allow us to leverage samples from the MCMC chain. Conversely, using a variational bound based on the implicit variational distribution $q^K_{\theta,\phi}(z|x)$ requires more refined approaches to compute its entropy (Thin et al. 2020). Instead, we use samples from an MCMC chain in conjunction with a perturbation of the MLE, as used previously, see, for instance, Han et al. (2017), Hoffman (2017), Nijkamp et al. (2020). More precisely, at iteration $t$, let $\theta^{(t)}$ and $\phi^{(t)}$ be the current estimate of the generative and variational parameters. Since maximising the log-likelihood is equivalent to minimising the KL divergence loss $D(\theta) = \mathrm{KL}(p_d(x)|p_\theta(x))$ over the generative parameters $\theta$, we consider the following perturbed loss function

$$\begin{aligned}S(\theta) &= D(\theta) + \mathrm{KL}(q^K_{\theta^{(t)},\phi^{(t)}}(z|x)|p_\theta(z|x))) \\ &= \mathrm{KL}(p_d(x) q^K_{\theta^{(t)},\phi^{(t)}}(z|x)|p_\theta(z,x))),\end{aligned}$$

see also Pang et al. (2020), Han et al. (2020). Note first that $S(\theta)$ becomes a tractable objective as it involves joint distributions over the latent variables and the data, in contrast to the log-likelihood objective involving marginal distributions. Second, $S(\theta)$ majorises $D(\theta)$, that is $S(\theta) \geq D(\theta)$. An EM-type algorithm would update $\theta^{(t)}$ to $\theta^{(t+1)}$ by minimising $S(\theta)$ for fixed variational parameters $\phi^{(t)}$ so that $S(\theta^{(t+1)}) \leq S(\theta^{(t)})$. We consider instead an alternating approach that follows the gradient of $\nabla S(\theta^{(t)})$ given by the average of

$$\int_{\mathsf{Z}} q^K_{\theta^{(t)},\phi^{(t)}}(z|x) \left[ \nabla_\theta \log p_\theta(z) + \nabla_\theta \log p_\theta(x|z) \right] \mathrm{d}z$$

over $x \sim p_d$, while also updating the variational and MCMC parameters $\phi^{(t)}$ in a single iteration.

### 4.7 Algorithm

Pseudo-code for the suggested algorithm is given in Algorithm 1 at a given iteration $t$, for illustration based on a mini-batch of size one. We have found that pre-training the decoder and encoder parameters $\theta$, respectively $\phi_0$, by optimizing the standard ELBO (2) before applying Algorithm 1, can decrease the overall training time. While we only consider MALA or HMC proposals in our experiments, other proposals with a tractable entropy, for instance those suggested in Li et al. (2020), can be used analogously.

## 5 Extension to hierarchical VAEs

We consider top-down hierarchical VAE (hVAE) architectures. Such models can leverage multiple layers $L$ of latent variables $(z^1, \ldots, z^L)$, $z^\ell \in \mathbb{R}^{n_\ell}$, where $z^L$ is the latent variable at the top and $z^1$ the latent variable at the bottom. Often $n_{\ell+1} \leq n_\ell$ to account for multiple resolutions. The generation of the latent variables follows the same order in both the

**Algorithm 1** Single training step for updating the generative model, initial encoding distribution and MCMC kernel.

```
Input: Number of Metropolis-Hastings steps $K$, learning rates $\rho^1, \rho^2, \rho^3, \rho^4$, current parameters $\theta^{(t)}, \phi_0^{(t)}, \phi_1^{(t)}, \beta^{(t)}$ and target acceptance rate $\alpha^\star$.
Sample $x \sim \mu$.
Sample $z_0 \sim q^0_{\phi_0^{(t)}}(\cdot|x)$ via reparameterisation.
Set $\widehat{\nabla}_{\phi_0^{(t)}} \mathcal{L}_0 = \nabla_{\phi_0^{(t)}} \log p_{\theta^{(t)}}(x|z_0) - \nabla_{\phi_0^{(t)}} \log q_{\phi_0^{(t)}}(z_0|x)$
Set $\widehat{\nabla}_{\phi^{(t)}} \mathcal{F} = 0$ and $\widehat{\alpha} = 0$.
Set $z \mapsto U_{\theta^{(t)}}(z|x) = -\log p_{\theta^{(t)}}(x|z) - \log p_{\theta^{(t)}}(z)$ for any $z \in \mathsf{Z}$ and let $M_{\theta^{(t)}, \phi_1^{(t)}}$ be an invariant Markov kernel of $e^{-U_{\theta^{(t)}}(\cdot|x)}$.
for $k$=1 to $K$ do
    Sample $z_k \sim M_{\theta^{(t)}, \phi_1^{(t)}}(z_{k-1}, \cdot|x)$ via $v_k \sim \mathcal{N}(0, \mathrm{I})$ and $z_k = \mathcal{T}_{\theta^{(t)}, \phi_1^{(t)}}(v_k|z_{k-1}, x)$.
    Set $\widehat{\alpha}$ += 1 if $z_k$ is accepted.
    Set $\widehat{\nabla}_{\phi_1^{(t)}} \mathcal{F}$ += $\nabla_{\phi_1^{(t)}} \big[ \log \alpha_{\theta^{(t)}, \phi_1^{(t)}}(z_{k-1}, z_k) - \beta^{(t)} \log r_{\phi_1^{(t)}}(z_{k-1}, z_k|x) \big]$.
Set $\widehat{\nabla}_{\theta^{(t)}} \mathcal{G} = \nabla_\theta \left[ \log p_\theta(x|z_K) + \log p_\theta(z_K) \right] |_{\theta=\theta^{(t)}}$.
Perform parameter updates:
$\phi_0^{(t+1)} = \phi_0^{(t)} + \rho^1 \widehat{\nabla}_{\phi_0^{(t)}} \mathcal{L}_0$
$\phi_1^{(t+1)} = \phi_1^{(t)} + \rho^2 \widehat{\nabla}_{\phi_1^{(t)}} \mathcal{F}_0$
$\theta^{(t+1)} = \theta^{(t)} + \rho^3 \widehat{\nabla}_{\theta^{(t)}} \mathcal{G}$
$\beta^{(t+1)} = \beta^{(t)}(1 + \rho^4(\frac{\widehat{\alpha}}{K} - \alpha^\star))$.
```

prior

$$(z^1, \dots, z^L) \sim p_\theta(z^1) p_\theta(z^2|z^1) \cdots p_\theta(z^L|z^{\leq L-1}), \tag{5}$$

for $z^{\leq \ell} = (z^1, \dots z^\ell)$, and in the approximate posterior,

$$(z^1, \dots, z^L)|x \sim q^0_{\phi_0,\theta}(z^1|x) \cdots q^0_{\phi_0,\theta}(z^L|x, z^{\leq L-1}) \tag{6}$$

cf. Sønderby et al. (2016), Kingma et al. (2016), Nijkamp et al. (2020), Maaløe et al. (2019), Vahdat and Kautz (2020), Child (2021). More concretely, to build the auto-regressive densities, we consider a sequence of variables $d^\ell \in \mathbb{R}^{n'_\ell}$ that are deterministic given $z^\ell$ and defined recursively as

$$d^\ell = h_{\ell,\theta}(z^{\ell-1}, d^{\ell-1}) \tag{7}$$

for some neural network function $h_{\ell,\theta}$, where the $d^{\ell-1}$-argument is a possible skip connection in a residual architecture for $\ell > 1$ and some constant $d^1$. This implies that the dependence on all previous latent variables $z^{\leq \ell}$ is implemented via the first-order Markov model of the residual discrete states $d^1, \dots, d^\ell$. Suppose further that we instantiate (5) in the form

$$z^\ell = \mu_{\ell,\theta}(d^\ell) + \sigma_{\ell,\theta}(d^\ell) \odot \epsilon^\ell \tag{8}$$

for some functions $\mu_{\ell,\theta}$ and $\sigma_{\ell,\theta}$, with $\epsilon^\ell$ denoting iid Gaussian random variables. This construction leads to the auto-regressive structure in the prior (5). To describe the variational approximation in (6), we consider a bottom-up network that defines deterministic variables $d'^\ell \in \mathbb{R}^{n'_\ell}$ recursively by setting $d'^{L+1} = x$ and $d'^\ell = h'_{\ell,\phi_0}(d'^{\ell+1})$ for $1 \leq \ell \leq L$ for functions $h'_{\ell,\phi_0}$. We assume a residual parameterisation (Vahdat and Kautz 2020; Vahdat et al. 2021) for $q^0_{\phi_0}(z^\ell|x, z^{\leq \ell-1})$ in the form

$$\begin{aligned} z^\ell &= \mu_{\ell,\theta}(d^\ell) + \sigma_{\ell,\theta}(d^\ell) \mu'_{\ell,\phi}(d^\ell, d'^\ell) \\ &\quad + (\sigma_{\ell,\theta}(d^\ell) \sigma'_{\ell,\phi_0}(d^\ell, d'^\ell)) \odot \epsilon^\ell \end{aligned} \tag{9}$$

for some functions $\mu'_{\ell,\phi_0}$ and $\sigma'_{\ell,\phi_0}$. This implies that

$$\begin{aligned} &\mathrm{KL}(q^0_{\phi_0}(z^\ell|x, z^{\leq \ell-1}) | p_\theta(z^\ell|z^{\leq \ell-1})) \\ &= \frac{1}{2} \Bigg[ \sum_{i=1}^{n^\ell} \sigma'_{\ell,\phi_0}(d^\ell, d'^\ell)_i^2 - n^\ell + \mu'_{\ell,\phi}(d^\ell, d'^\ell)_i^2 \\ &\quad + \log \sigma'_{\ell,\phi_0}(d^\ell, d'^\ell)_i^2 \Bigg]. \end{aligned} \tag{10}$$

The observations $x$ are assumed to depend explicitly only on $z^L$ and $d^L$ through some function $g_\theta$ in the sense that $x|z^1, \dots, z^L \sim p_\theta(x|g_\theta(z^L))$. The generative model of the latent variables $z^1, \dots z^L$ in (5) is written in a centred parameterisation (Papaspiliopoulos et al. 2007) that makes them dependent a priori. Our experiments will illustrate that these dependencies can make it challenging to sample from the posterior distribution for MCMC schemes that are not adaptive.

We want to clarify that we interpret a hVAE as a special case of the VAE with a hierarchical structure of the latent variables $z = (z^1, \dots, z^L) \in \mathbb{R}^n$, $z^\ell \in \mathbb{R}^{n_\ell}$, $n = \sum_{i=1}^L n_\ell$. An alternative viewpoint would be to consider the VAE in Sect. 4 that utilises MCMC steps as a hierarchical VAE wherein each step of the Markov chain corresponds to a new layer of an hVAE, with all latent variables $z_0, \dots, z_K \in \mathbb{R}^n$ living in the same latent space. More precisely, from such an alternative perspective, the latent variable $z_0$ sampled from the prior $p_\theta$ or initial encoder $q^0_{\phi_0}$ can be seen as the first latent variable of an hVAE at the bottom layer, while the transition densities in the generative auto-regressive distributions in (5) are modelled as Metropolis-Hastings kernels. In our viewpoint, we minimise the KL of the joint latent variables $(z^1, \dots, z^L)$ as in (10) for learning the initial variational parameters $\phi_0$ that parameterises the encodings of $(z^1, \dots, z^L)$ jointly. However, performing variational inference in the alternative viewpoint would require different approaches. Our approach also differs from those in score-based diffusion models that utilise score functions to transition between hierarchical latent variables, see Appendix A for details.

# 6 Numerical experiments

## 6.1 Evaluating model performance with marginal log likelihood

We start by considering different VAE models and inference strategies on four standard image data sets (MNIST, Fashion-MNIST, Omniglot and SVHN) and evaluate their performance in terms of their test log-likelihood estimates.

## 6.2 Marginal log-likelihood estimation

We start to evaluate the performance of different variations of VAEs using the marginal log-likelihood of the model on a held-out test set for a variety of benchmark datasets. In doing so, we resort to importance sampling to estimate the marginal log-likelihood using $S$ importance samples via

$$\log \hat{p}_{\mathrm{IS}}(x) = \log \frac{1}{S} \sum_{s=1}^{S} \frac{p_\theta(x|z_s) p_\theta(z_s)}{r(z_s|x)}, \, z_s \sim r(\cdot|x),$$

where $r$ is an importance sampling density. Following Ruiz and Titsias (2019), in the case of a standard VAE, we choose $r(z|x) = \mathcal{N}(\mu^z_{\phi_0}(x), \tau \Sigma^z_{\phi_0}(x))$ for some scaling constant $\tau \geq 1$, assuming that $q^0_{\phi_0}(z_0|x) = \mathcal{N}(\mu^z_{\phi_0}(x), \Sigma^z_{\phi_0}(x))$ with diagonal covariance matrix $\Sigma^z_{\phi_0}(x)$. For the case with MCMC sampling using $K$ steps, we choose $r(z_s|x) = \mathcal{N}(z_K(x), \tau \Sigma^z_{\phi_0}(x))$, where $z_K(x)$ is an estimate of the posterior mean from the MCMC chain.

## 6.3 VAE models

Using the metric described above, we evaluate our model and compare it against other popular adjustments of VAEs for various data sets. In terms of comparing models, we focused on comparing our model, denoted VAE-gradMALA and VAE-gradHMC, against i) a Vanilla VAE, ii) VAEs utilising MCMC samplers that are adapted using a dual-averaging scheme (Hoffman and Gelman 2014; Nesterov 2009) that we refer to as VAE-gradMALA and VAE-gradHMC. We also compare against iii) VAEs using more expressive priors such as a Mixture of Gaussians (MoG), denoted VAE-MoG cf. Jiang et al. (2017), Dilokthanakul et al. (2016), or a Variational Mixture of Posteriors Prior (VAMP), see Tomczak and Welling (2017), denoted VAE-VAMP. For the MNIST example, we consider a Bernoulli likelihood with a latent space of dimension 10. We pre-trained the model for 90 epochs with a standard VAE, and subsequently trained the model for 10 epochs with MCMC. We used a learning rate of 0.001 for both algorithms. For the remaining datasets, we pre-trained for 290 epochs with a standard VAE, followed by training for 10 epochs with MCMC. We used a learning rate of 0.005, while the dimension of the latent space is 10, 20, and 64 for Fashion-MNIST, Omniglot and SVHN, respectively. For the SVHN dataset, we considered a 256-logistic likelihood with a variance fixed at $\sigma^2 = 0.1$, see Salimans et al. (2017) for details. In terms of the neural network architecture used for the encoder and the decoder, more information can be found in the codebase. All models use the same decoders and (initial) encoding distributions. The inference times of the models trained either with dual-averaging or with the gradient-based generalised speed measure objective are comparable. We use $K = 10$ MCMC steps.

## 6.4 Experimental results

Table 1 summarises the estimated log-likelihoods for the different data sets. The results therein show the means of three independent runs, with their standard deviations in brackets. For the case of SVHN, the estimate is transformed to be represented in bits per dimension. We observe that among the considered methods that utilize MCMC samplers within VAEs, our approach performs better across the datasets we explored. We note that for the considered decoder and encoder architectures, the use of more flexible generative models by using more flexible priors such as a VAMP prior, can yield higher log-likelihoods. However, the choice of more flexible priors is completely complementary to the inference approach suggested in this work. Indeed, we illustrate in Sects. 6.7 and 6.8 that our MCMC adaptation strategy performs well for more flexible hierarchical priors.

## 6.5 Evaluating generative performance with kernel inception distance (KID)

### 6.5.1 Generative metrics

The generative performance of our proposed model is additionally quantitatively assessed by computing the Kernel Inception Distance (KID) relative to a subset of the ground truth data. We chose the KID score instead of the more traditional Fréchet inception distance (FID), due to the inherent bias of the FID estimator (Bińkowski et al. 2018). To compute the KID score, for each image from a held-out test set, we sample a latent variable from the prior density and then pass it through the trained decoder of the corresponding model to generate a synthetic image. Images are resized to (150,150,3) using the bi-cubic method, followed by a forward pass through an inception-V3 model using the Imagenet weights. This yields a set of Inception features for the synthetic and held-out test set. The computation of the KID score for these features is based on a polynomial kernel, similarly to Bińkowski et al. (2018). For all datasets, we utilised a learning rate of 0.001 for both the VAE and MCMC algorithms. We trained the VAE for 100 epochs and performed sampling with

**Table 1** Importance sampling estimate of the log-likelihood (with highest values in bold for each setup of either a standard Gaussian or a learnable prior) on the test set based on $S = 10000$ and $\tau = 1.5$

| Model | MNIST | Fashion- MNIST | Omniglot | SVHN |
|---|---|---|---|---|
| VAE | −81.16 (0.2) | −116.65 (0.1) | −117.46 (0.2) | 7.203 (0.005) |
| VAE- gradMALA | −79.94 (0.1) | −115.84 (0.1) | −116.93 (0.3) | 7.209 (0.005) |
| VAE- dsMALA | −80.36 (0.1) | −116.32 (0.2) | −117.48 (0.2) | 7.203 (0.001) |
| VAE- gradHMC | **−79.52** (0.2) | **−115.77**(0.1) | **−116.69**(0.3) | **7.179** (0.001) |
| VAE- dsHMC | −79.89 (0.1) | −116.02 (0.1) | −116.88 (0.1) | 7.187 (0.003) |
| VAE- MoG | −80.52 (0.1) | −116.40 (0.3) | −119.14 (0.1) | 7.205 (0.001) |
| VAE-VAMP | **−78.48** (0.1) | **−114.30** (0.1) | **−117.23**(0.1) | **7.197** (0.001) |

The values denote the mean of three independent runs, while the standard deviation is given within brackets. The MoG and VAMP VAEs use different priors

**Table 2** Estimates of KID for each model considered across different datasets

| Model | MNIST | Fashion- MNIST | SVHN | CIFAR-10 |
|---|---|---|---|---|
| VAE | 1.084 (0.05) | 0.925 (0.03) | 0.197 (0.01) | 1.348 (0.01) |
| VAE- gradHMC | **0.431** (0.02) | **0.852** (0.03) | **0.126** (0.01) | **1.153** (0.01) |
| VAE- dsHMC | 0.653 (0.01) | 0.908 (0.06) | 0.183 (0.02) | 1.587 (0.10) |
| VAE- MoG | 0.542 (0.01) | 0.990 (0.01) | **0.190** (0.01) | **1.444** (0.01) |
| VAE-VAMP | **0.434** (0.05) | **0.610** (0.03) | 0.210 (0.01) | 1.657 (0.03) |

Lowest values in bold for each setup of either a standard Gaussian or a learnable prior

The values denote the mean of three seeds, while the standard deviation is shown within brackets. The MoG and VAMP VAEs use different priors

the MCMC algorithms for 50 epochs if applicable, yielding a total training of 150 epochs across all cases. The likelihood functions used were Bernoulli for the MNIST and Fashion-MNIST datasets, while the logistic-256 likelihood (Salimans et al. 2017) was used for the SVHN and Cifar-10 datasets, with a fixed variance of $\sigma^2 = 0.1$ and $\sigma^2 = 0.05$, respectively. The dimension of the latent variable was fixed to 10 for the MNIST datasets, while it was set to 64 and 256 for the SVHN and CIFAR-10 datasets. More details regarding the neural network architecture used for training the VAE can be found in the codebase.

### 6.5.2 VAE models and quantitative evaluation

Similarly to Sect. 6.1, we perform a series of experiments that compare our adaptation scheme to other popular VAE modifications across different data sets. In Table 2, we summarise the results of our experiments reporting mean KID scores from three different seeds with the standard deviation in brackets. We notice a similar pattern to that from Sect. 6.1, where our proposed method outperforms other MCMC-related methods. At the same time, we observe that models with more expressive priors such as the VAMP prior, can perform equally or slightly better, particularly in the case of a low-dimensional latent state space, such as for MNIST and Fashion-MNIST. However, in the case of higher dimensional latent space, such as used for CIFAR-10 with $d_z = 256$, we observe that our method shows considerable improvement compared to the other methods.

### 6.5.3 Qualitative results

In addition to computing the KID score, we qualitatively inspect the reconstructed images and the images sampled from the model. In Fig. 1, we can see reconstruction for the best three performing models. Figure 2 contains unconditionally generated samples for the same models. We observe that, indeed, KID scores qualitatively correlate with more expressive generations and reconstructions. In particular, we observe a slight decrease in blurriness and an increase in the resolution of smaller details such as the car-light of the red car in Fig. 1. Moreover, the unconditionally generated images in Fig. 2 exhibit more expressive colour patterns.

## 6.6 Evaluating model performance in small sample size data

### 6.6.1 Data augmentation task

In addition to testing our proposed approach against the above benchmark datasets, we also test our approach in a real-world dataset comprised of complex images that, however, are characterised by a relatively small sample size. We chose the Alzheimer's Disease Neuroimaging Initiative (ADNI) [1] brain MRI dataset, which is comprised of 4000 Brain MRI Scans of individuals suffering from Dementia, and individuals from

[1] https://www.kaggle.com/datasets/sachinkumar413/alzheimer-mri-dataset.

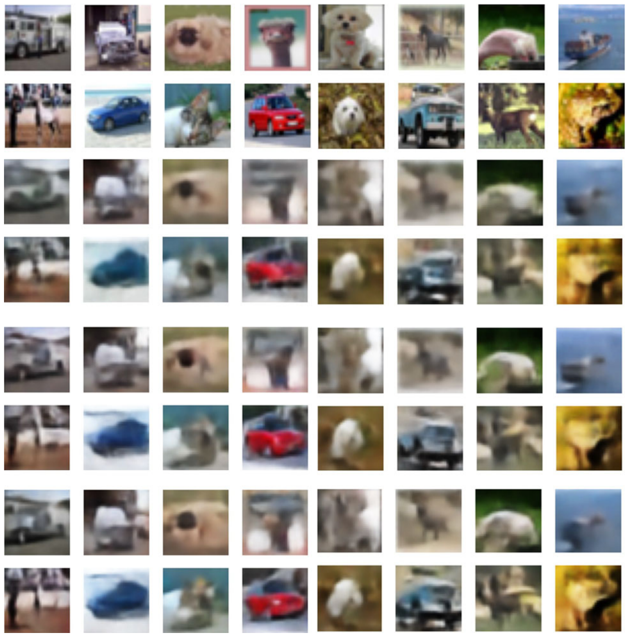

**Fig. 1** Model reconstruction images for the top three performing models tested on CIFAR-10 in terms of the KID-score evaluated on model samples. The first two rows illustrate the ground truth, the next two show reconstructions from the Vanilla VAE model, the next two illustrate reconstructions from the dsHMC model, and the last two rows illustrate reconstructions from the gradHMC coupled VAE

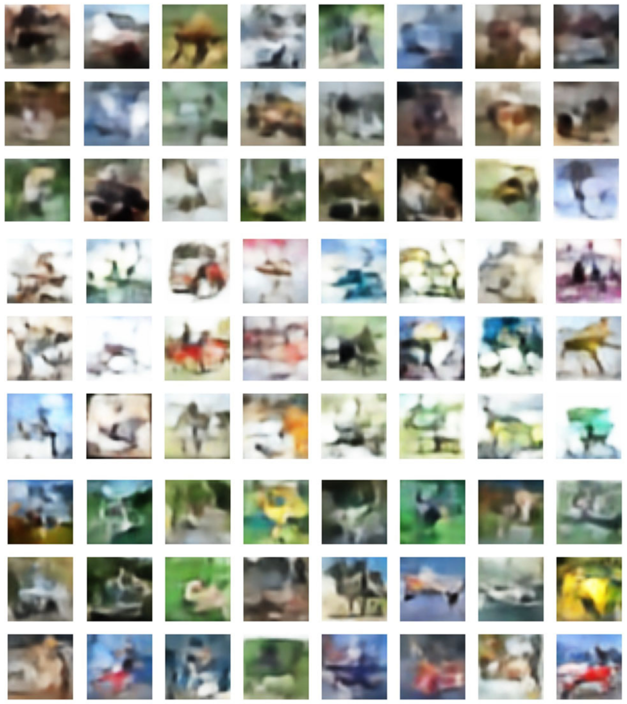

**Fig. 2** Model sampled images for the top three performing models tested on CIFAR-10 in terms of the KID-score evaluated on model samples. The first three rows illustrate samples from the Vanilla VAE model, the next three rows illustrate samples from the dsHMC model, and the last three rows illustrate samples from the gradHMC coupled VAE

a healthy control group, in a ratio of 1:3, respectively. The small sample size as well as the imbalance in the dataset pose a problem for classification tasks that are often addressed by different data augmentations. We illustrate here that the proposed generative model can be used to generate additional samples that are representative of each sub-population in the dataset, namely healthy controls and diseased individuals. We first trained VAEs for each separate class on the dataset,[2] using a VAE learning rate of 0.001 and an MCMC learning rate of 0.01, whenever applicable.

### 6.6.2 Generative performance

The VAEs were trained for 2000 epochs with 100 epochs of MCMC coupled training, whenever it was applicable. The KID score presented in Table 3 was based on the whole dataset (that is, including both training and test sets), because the KID score can under-perform for the small size of the test set in the minority class. The neural networks utilised in the encoder and the decoder were similar to those of Sect. 6.5, consisting of two dense layers of 200 units each for the decoder and the encoder. Moreover, the latent dimension for all experiments was fixed at 20, while the likelihood utilised was a logistic-256 (Salimans et al. 2017) with a fixed variance of $\sigma^2 = 0.05$. After training, a series of 200, 500, 1000, and 2000 images were generated for the minority class, which were then augmented with the generated images. Classification performance for classifier models trained on this augmented dataset was then compared against classifier models trained on the non-augmented dataset. More details regarding the architectures used for the VAE and classifier models can be found in the codebase. We observed that one obtains the best performance in terms of the classification metrics for the dataset augmented with 200 images and we thus report these values in Table 3. We find that a VAE with a gradient-based adaptation of the HMC sampler has better generative performance, particularly for the dementia group. The minority class, *i.e.* the dementia group, was augmented by the addition of synthetic data from the generative models. Qualitative results showing the generated samples are given in Fig. 3 for the standard VAE model, in addition to those VAE models that are combined with MCMC. We notice that our proposed method captures more brain characteristics for both the demented and normal patients, due to the presence of various brain structures throughout the generated samples, while also capturing class-specific characteristics, such as a greater degree of brain matter loss in the dementia class.

[2] We merged the two classes of the dementia cases to a single dementia case.

**Table 3** Estimates of the KID score for each respective class in the ADNI brain MRI dataset and classification metrics from the data augmentation task across different models

| Model | KID/Dementia | KID/Controls | Bacc | TPR | TNR |
|---|---|---|---|---|---|
| VAE | 12.44 (0.8) | 12.64 (2.35) | 0.968 (0.01) | 0.986 (0.003) | 0.950 (0.002) |
| VAE- gradHMC | **10.25** (0.85) | **9.76** (1.11) | **0.971** (0.01) | **0.989** (0.005) | **0.954** (0.002) |
| VAE- dsHMC | 12.02 (1.67) | 10.81 (1.28) | **0.964** (0.01) | **0.989** (0.005) | 0.940 (0.001) |
| No- Augmentation | – | – | 0.878 (0.02) | 0.824 (0.024) | 0.932 (0.025) |

Lowest KID values and highest classification metrics in bold
Standard deviations in brackets

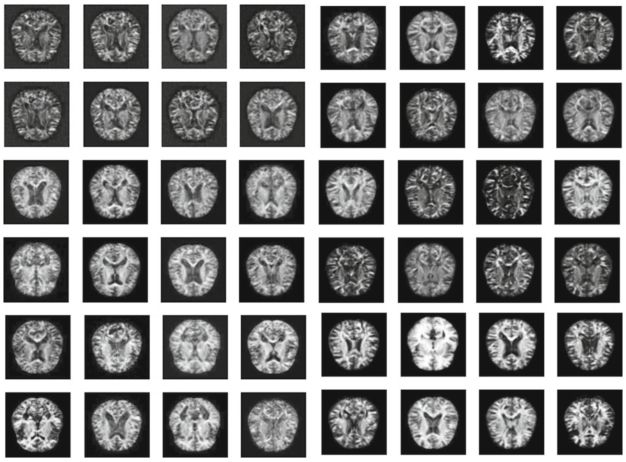

**Fig. 3** Model samples from VAE variations trained on either demented (first four columns) or normal patients (last four columns). The first two rows are samples from the Vanilla VAE model, the next two rows from the VAE using dual-average adaptation and the last two rows from our proposed method using a VAE with entropy-based adaptation

### 6.6.3 Classification results

We performed a classification between the two groups with results summarised in Table 3. It illustrates first that augmenting data with a trained VAE improves the classification in general, and second, that augmentations with our proposed method lead to a small, yet significant increase in Balanced Accuracy, True Positive Rate (TPR) and True Negative Rate (TNR). These results are consistent with the improved quality of the generated samples using our approach and we thus believe that our method can be leveraged for effective data augmentations.

## 6.7 Linear hierarchical VAEs

We consider linear Gaussian models with a Gaussian prior $p_\theta(z) = \mathcal{N}(\mu_z, \Sigma_z)$ and a linear decoder mapping so that $p_\theta(x|z) = \mathcal{N}(Wz + b, \Sigma_{x|z})$ for $\mu_z, b \in \mathbb{R}^{d_z}$, $W \in \mathbb{R}^{d_x \times d_z}$ and covariance matrices $\Sigma_z$ and $\Sigma_{x|z}$ of appropriate dimension. The resulting generative model corresponds to a probabilistic PCA model (Tipping and Bishop 1999), see also Dai et al. (2018), Lucas et al. (2019) for further connections to VAEs. This section aims to illustrate that adaptation with a non-diagonal pre-conditioning matrix becomes beneficial to account for the dependence structure of the latent variables prevalent in such hierarchical models.

### 6.7.1 Hierarchical generative model

We can sample from the Gaussian prior $z \sim \mathcal{N}(\mu_z, \Sigma_z)$ in a hierarchical representation using two layers:

$$z^1 \sim \mathcal{N}(0, \mathrm{I}), \quad z^2|z^1 \sim \mathcal{N}(A_2 z^1 + c_2^\mu, \Lambda_{z^2|z^1})), \tag{11}$$

where $z = (z^1, z^2)$ and $\Lambda_{z^2|z^1} = \mathrm{diag}(\sigma^2_{z^2|z^1})$. To recover (11) from the general auto-regressive prior factorisation (5), assume that $d^1 = 0 \in \mathbb{R}^{n_1'}$, $n_1' = n_1'$. For $d = (d^\mu, d^\sigma)$, suppose that $\mu_{1,\theta}(d) = d^\mu$ is the projection on the first $n_1$ components, while $\sigma_{1,\theta}(d) = \exp(0.5 d^\sigma)$ describes the standard deviation based on the last $n_1$ components. Further, consider the linear top-down mapping

$$h_{2,\theta} \colon (z^1, d^1) \mapsto d^2 = \begin{bmatrix} A_2 & B_2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} z^1 \\ d^1 \end{bmatrix} + \begin{bmatrix} c_2^\mu \\ c_2^\sigma \end{bmatrix},$$

for the deterministic variables, where $c_2^\sigma = 2 \log \sigma_{z^2|z^1}$. We assume the same parameterisation for the prior densities of $z^2$ given $d^2$ as in the first layer: $\mu_{2,\theta}(d) = \mu_{1,\theta}(d) = d^\mu$, and $\sigma_{2,\theta}(d) = \sigma_{1,\theta}(d) = \exp(0.5 d^\sigma)$. We assume further that the decoder function depends explicitly only on the latent variables $z^2$ and $d^2$ at the bottom in the form of

$$\begin{aligned} p_\theta(x|z) &= \mathcal{N}(W_2^z z^2 + W_2^d d^2 + b, \Sigma_{x|z}) \\ &= \mathcal{N}(Wz + b + W_2^d c_2^\mu, \Sigma_{x|z}), \end{aligned}$$

for $W = \begin{bmatrix} W_2^d A_2 & W_2^z \end{bmatrix}$. Observe that the covariance matrix of the prior density is

$$\Sigma_z = \begin{bmatrix} \mathrm{I} & (A_2)^\top \\ A_2 & A_2 A_2^\top + \mathrm{I} \end{bmatrix}.$$

**Table 4** Condition Number $\kappa(\Sigma_{z|x}^{-1})$ of the posterior distribution for a smaller and larger linear hierarchical VAE model, each consisting of latent variables in two layers, having the dimensions (10,20) and (50,100), respectively

| Model | $\kappa(\Sigma_{z\|x}^{-1})_{(10,20)}$ | $\kappa(C^\top \Sigma_{z\|x}^{-1} C)_{(10,20)}$ | $\kappa(\Sigma_{z\|x}^{-1})_{(50,100)}$ | $\kappa(C^\top \Sigma_{z\|x}^{-1} C)_{(50,100)}$ |
|---|---|---|---|---|
| hVAE | 18.07 (0.45) | – | 508.55 (12.1) | – |
| gradMALA- D | 20.43 (0.99) | 19.02 (1.24) | 578.05 (84.5) | 434.24 (13.0) |
| dsMALA- D | 18.62 (1.34) | 18.62 (1.34) | 617.36 (48.4) | 617.36 (48.4) |
| gradHMC- D | 21.57 (1.48) | 22.67 (1.23) | 502.41 (36.0) | 431.14 (23.4) |
| dsHMC- D | 18.38 (1.27) | 18.38 (1.27) | 621.1 (54.9) | 621.1 (54.9) |
| gradMALA- LT | 23.55 (5.31) | **1.67** (0.06) | 475.93 (23.7) | **2.0** (0.02) |
| gradHMC- LT | 25.57 (2.63) | 1.68 (0.15) | 483.05 (38.0) | 2.24 (0.04) |

Condition number of the transformed posterior $\kappa(C^\top \Sigma_{z|x}^{-1} C)$

The marginal distribution of the data is $x \sim \mathcal{N}(\mu_x, \Sigma_x)$, where $\mu_x = W_2^z c_2^\mu + b$ and

$$\Sigma_x = W \Sigma_z W^\top + \Sigma_{x|z}.$$

The covariance matrix of the posterior density becomes

$$\Sigma_{z|x} = \Sigma_z - (W\Sigma_z)^\top \Sigma_x^{-1} W \Sigma_z. \tag{12}$$

Depending on the conditioning number of $\Sigma_{z|x}$, this matrix can be poorly-conditioned, which can hinder the performance of non-adaptive MCMC methods. Particularly for models that infer a high dependence between $z^1$ and $s^2$, the prior covariance $\Sigma_z$ can be ill-conditioned, which can lead to ill-conditioned posteriors. By contrast, with suitable preconditioning, we can expect MALA, HMC, and other MCMC methods to become more performant at sampling from the posterior distribution.

### 6.7.2 Encoding model

Assume a linear encoder model based on a linear bottom-up model so that $d'_3 = x$, and for $1 \leq \ell \leq 2$, suppose that $d'^{\ell} = W'_\ell d'^{\ell+1} + b'_\ell$ are bottom-up deterministic variables. We construct an encoding distribution by setting

$$\mu'_{\ell,\theta} \colon (d^\ell, d'^\ell) \mapsto B'_\ell \begin{bmatrix} d^\ell \\ d'_\ell \end{bmatrix} + c'_\ell$$

and $\sigma'_{\ell,\theta} \colon (d^\ell, d'^\ell) \mapsto \exp(b'_\ell)$ in the residual pasteurisation (9).

### 6.7.3 Experimental results

We first test if the adaptation scheme can adapt to the posterior covariance $\Sigma_{x|z}$ given in (12) of a linear hVAE model, *i.e.* if the condition number of $C \Sigma_{x|z} C^\top$ becomes small. As choices of $C$, we consider (i) a diagonal preconditioning matrix (denoted D) and (ii) a lower-triangular preconditioning matrix (denoted LT). Note that the dual-averaging adaptation scheme used here and in Hoffman and Gelman (2014) adapts only a single step-size parameter, thereby leaving the condition number unchanged. We tested two simulated data sets with corresponding latent dimensions $(n_1, n_2)$ of (10,20) and (50,100). More specifically, we simulated datasets with 1000 samples for each configuration, using the linear observation model with a standard deviation of 0.5. We used a hierarchical VAE with two layers and a learning rate of 0.001. For the dataset from the model with a latent dimension of (10,20), we pre-trained the VAE for 1000 epochs without MCMC, followed by training for 1000 epochs with MCMC. The number of MCMC steps was fixed at $K = 2$. For the dataset from the model generated from a higher dimensional latent space of dimension (50,100), we increased the number of training epochs from 1000 to 5000, while also increasing the number of MCMC steps from $K = 2$ to $K = 10$. For different choices for the size of the latent variables, Table 4 shows that both gradient-based adaptation schemes lead to a very small transformed condition number $\kappa(C^\top \Sigma_{z|x}^{-1} C)$ when a full preconditioning matrix is learnt, with smallest values in bold for each configuration of the latent dimensions. Notice also that for all models, the posterior becomes increasingly ill-conditioned for higher dimensional latent variables, as confirmed by the large values of $\kappa(\Sigma_{z|x}^{-1})$ in Table 4.

In addition to the condition number, we also investigate how the adaptation scheme affects the learned model in terms of the marginal log-likelihood, which is analytically tractable. The results summarised in Table 5 show that the gradient-based adaptation schemes indeed achieve a higher log-likelihood.

## 6.8 Non-linear hierarchical VAEs

Finally, we investigate the effect of combining MCMC with hVAE in the general non-linear case for hierarchical models. More precisely, we follow the general model setup in Sect. 5, which differs from the linear examples above by the inclusion of a ReLU activation in the considered neural networks. We consider a hVAE with two layers of size 5 and 10. The learning rate of the hVAE and MCMC algorithms

**Table 5** Difference between true and estimated data log-likelihood $\log p_\theta(x)$ for hierarchical VAEs with two layers and where the dimension of the latent variables $(z^1, z^2)$ are set to (10,20) and (50,100), respectively

| Model | $\Delta \log p(x)_{(10,20)}$ | $\Delta \log p(x)_{(50,100)}$ |
|---|---|---|
| hVAE | 24.91 (7.67) | 13.08 (0.26) |
| gradMALA- D | 1.54 (1.49) | 7.14 (0.11) |
| dsMALA- D | 2.68 (2.04) | 8.77 (0.11) |
| gradHMC- D | 1.16 (1.72) | 2.19 (0.05) |
| dsHMC- D | 2.59 (1.78) | 8.06 (0.48) |
| gradMALA- LT | 1.56 (1.61) | 1.96 (0.12) |
| gradHMC- LT | **1.14** (1.53) | **1.66** (0.15) |

The highest loglikelihood values for each configuration of the latent dimensions are in bold

**Table 6** Estimates of KID for each model considered across different datasets with lowest KID scores for each dataset in bold

| Model | MNIST | Fashion- MNIST |
|---|---|---|
| hVAE | 0.496 (0.062) | 1.269 (0.037) |
| gradMALA- D | 0.432 (0.057) | 1.038 (0.068) |
| dsMALA- D | 0.600 (0.040) | 1.312 (0.034) |
| gradHMC- D | 0.447 (0.073 ) | 1.079 (0.174) |
| dsHMC- D | 0.490 (0.098) | 1.151 (0.223) |
| gradMALA- LT | 0.475 (0.101) | 0.939 (0.143) |
| gradHMC- LT | **0.407** (0.030) | **0.916** (0.047) |

The values denote the mean of three seeds, while the standard deviation is shown within brackets

was set to 0.001. We use 200 epochs for training overall. For models that included MCMC sampling, we used the first 190 epochs for pre-training without MCMC. Additionally, the prior of the model was trained only during the hVAE portion of the algorithm. The resulting KID scores for MNIST and Fashion-MNIST can be found in Table 6. In this scenario, our proposed method outperforms other sampling schemes when combined with a hVAE model.

## 7 Conclusion

We have investigated the performance effect of training VAEs and hierarchical VAEs with MCMC speed measures and subsequently compared our proposed method with other widely used adaptive MCMC adaptations and VAE model variations. Adopting recent advances in the adaptive MCMC literature that are based on the notion of a generalised speed measure seem to provide, in the problems and datasets we tested, a more efficient learning algorithm for VAEs. Future research directions may focus on using our proposed method in models with deeper architectures in the encoder and the decoder, using our method in challenging inpainting problems and exploring its power at alleviating adversarial attacks as seen in Kuzina et al. (2022).

## A Appendix: Relation to score-based diffusion models

HVAEs can be interpreted as diffusion discretisations (Falck et al. 2022). Besides, score-based diffusion models (Sohl-Dickstein et al. 2015; Song and Ermon 2019; Ho et al. 2020; Song et al. 2020) can be interpreted as a hVAE by introducing a sequence of latent variables $(z^1, \ldots, z^{L-1})$ in the same space as the data $x = z^L$, where the generative distribution[3] $p_\theta(z^1, \ldots, z^{L-1})$ factories as in (5), while the inference distribution $q_\phi(z^1, \ldots z^{L-1}|x)$ is jointly Gaussian, for any given $x$, with known fixed parameters. The discrete-time diffusion model can be learned by maximising a variational lower bound over $\theta$,

$$\begin{aligned} &\log p_\theta(z_L) \\ \geq &\mathbb{E}_{q_\phi(z_{\leq L-1}|z_L)}\left[\log p_\theta(z_{\leq L}) - \log q_\phi(z_{L-1}|z_L)\right], \end{aligned}$$

which can be transformed into a denoising score-matching objective (Vincent 2011),

$$\frac{1}{2}\sum_{\ell=0}^{L-1}\gamma_\ell \|\nabla_{z_\ell} \log q_\phi(z^\ell|x) - s_\theta(z^\ell, \ell)\|^2$$

for suitable weights $\gamma_\ell > 0$. Here, $s_\theta$ is a learned score model that determines the mean of the generative distribution $p_\theta(z^{\ell+1}|z^\ell)$. More precisely, for suitable choices of the forward and backward dynamics (Song and Ermon 2019), one can view the generative path as an unadjusted Langevin algorithm based on the learned score function $s_\theta$. More general learning and sampling schemes can be used for such models, such as incorporating Hamiltonian dynamics with a fixed mass matrix in damped Langevin diffusions (Dockhorn et al. 2021; Pandey and Mandt 2023; Singhal et al. 2023). Diffusion models can also be used in a latent space (Vahdat et al. 2021; Rombach et al. 2022). We emphasise that our work instead considers Metropolis-adjusted Langevin or Hamiltonian dynamics based on the score function of the posterior $\nabla_z \log p_\theta(z|x) = \nabla_z \left[\log p_\theta(z) + \log p_\theta(x|z)\right]$, which is constant across the different MCMC steps. In the case of a hVAE, the score function is based on the joint posterior $\nabla_{(z^1,\ldots z^L)}\left[\log p_\theta(z^1, \ldots, z^L) + \log p_\theta(x|z^L)\right]$.

[3] To be consistent with the hVAE model above, the index ordering of the latent variables is reversed to the notation employed in many works on diffusion models.

**Author Contributions** M.H. and V.K. wrote the code and the manuscript. P.D. supervised the project and reviewed the manuscript.

## Declaration

**Conflict of interest** The authors declare no Conflict of interest.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

## References

Abadi, M., Barham, P., Chen, J., 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)

Altschuler, J.M., Chewi, S.: Faster high-accuracy log-concave sampling via algorithmic warm starts. In: 2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS). IEEE, pp. 2169–2176 (2023)

Ambrosio, L., Gigli, N., Savaré, G.: Gradient flows: in Metric Spaces and in the Space of Probability Measures. Springer, Berlin (2005)

Andrieu, C., Doucet, A., Holenstein, R.: Particle Markov Chain Monte Carlo methods. J. R. Stat. Soc. Ser. B (Stat. Methodol.) **72**(3), 269–342 (2010)

Aneja, J., Schwing, A., Kautz, J., et al.: A contrastive learning approach for training Variational Autoencoder priors. Adv. Neural. Inf. Process. Syst. **34**, 480–493 (2021)

Barber, D., Bishop, C.M. Ensemble learning for multi-layer networks. In: Advances in Neural Information Processing Systems, pp. 395–401 (1998)

Behrmann, J., Grathwohl, W., Chen, R.T., et al: Invertible residual networks. In: International Conference on Machine Learning, pp. 573–582 (2019)

Bińkowski, M., Sutherland, D.J., Arbel, M., et al: Demystifying MMD GANs. (2018) arXiv:1801.01401

Bou-Rabee, N., Sanz-Serna, J.M.: Geometric integrators and the Hamiltonian Monte Carlo method. Acta Numer **27**, 113–206 (2018)

Burda, Y., Grosse, R., Salakhutdinov, R.: Importance weighted autoencoders. (2015) arXiv:1509.00519

Caterini, A.L., Doucet, A., Sejdinovic, D.: Hamiltonian variational autoencoder. In: Advances in Neural Information Processing Systems, pp. 8167–8177 (2018)

Chen, T.Q., Behrmann, J., Duvenaud, D.K., et al: Residual flows for invertible generative modeling. In: Advances in Neural Information Processing Systems, pp. 9913–9923 (2019a)

Chen, Y., Gatmiry, K.: A simple proof of the mixing of metropolis-adjusted langevin algorithm under smoothness and isoperimetry (2023). arXiv:2304.04095

Chen, Y., Dwivedi, R., Wainwright, M.J., et al: Fast mixing of metropolized Hamiltonian Monte Carlo: benefits of multi-step gradients (2019b). arXiv:1905.12247

Chewi, S., Lu, C., Ahn, K., et al: Optimal dimension dependence of the metropolis-adjusted langevin algorithm. In: Conference on Learning Theory, PMLR, pp. 1260–1300 (2021)

Child, R.: Very deep VAEs generalize autoregressive models and can outperform them on images. In: International Conference on Learning Representations (2021)

Dai, B., Wang, Y., Aston, J., et al.: Connections with robust PCA and the role of emergent sparsity in variational autoencoder models. J. Mach. Learn. Res. **19**(1), 1573–1614 (2018)

Dilokthanakul, N., Mediano, P.A., Garnelo, M., et al: Deep unsupervised clustering with Gaussian mixture Variational Autoencoders (2016). arXiv:1611.02648

Dockhorn, T., Vahdat, A., Kreis, K.: Score-based generative modeling with critically-damped langevin diffusion. In: International Conference on Learning Representations (2021)

Du, Y., Mordatch, I.: Implicit generation and modeling with energy based models. Advances in Neural Information Processing Systems **32**, pp. 3608–3618 (2019)

Durmus, A., Moulines, E., Saksman, E.: On the convergence of Hamiltonian Monte Carlo (2017). arXiv:1705.00166

Dwivedi, R., Chen, Y., Wainwright, M.J., et al.: Log-concave sampling: metropolis-hastings algorithms are fast. J. Mach. Learn. Res. **20**(183), 1–42 (2019)

Falck, F., Williams, C., Danks, D., et al: A multi-resolution framework for U-Nets with applications to hierarchical VAEs. In: Advances in Neural Information Processing Systems (2022)

Finke, A., Thiery, A.H.: On importance-weighted autoencoders (2019). arXiv:1907.10477

Geffner, T., Domke, J.: On the difficulty of unbiased alpha divergence minimization. In: International Conference on Machine Learning, PMLR, pp. 3650–3659 (2021)

Guo, F., Wang, X., Fan, K., et al: Boosting variational inference (2016). arXiv:1611.05559

Hagemann, P., Hertrich, J., Steidl, G.: Stochastic normalizing flows for inverse problems: a Markov Chains viewpoint. SIAM/ASA J. Uncertain. Quantif. **10**(3), 1162–1190 (2022)

Hairer, E., Lubich, C., Wanner, G.: Geometric numerical integration illustrated by the Störmer–Verlet method. Acta Numer **12**, 399–450 (2003)

Han, S., Liao, X., Dunson, D., et al: Variational Gaussian copula inference. In: Artificial Intelligence and Statistics, pp. 829–838 (2016)

Han, T., Lu, Y., Zhu, S.C., et al: Alternating back-propagation for generator network. In: Proceedings of the AAAI Conference on Artificial Intelligence (2017)

Han, T., Zhang, J., Wu, Y.N.: From EM-projections to variational autoencoder. In: NeurIPS 2020 Workshop: Deep Learning through Information Geometry (2020)

Hernandez-Lobato, J., Li, Y., Rowland, M., et al: Black-box alpha divergence minimization. In: International Conference on Machine Learning, PMLR, pp. 1511–1520 (2016)

Hirt, M., Titsias, M., Dellaportas, P.: Entropy-based adaptive Hamiltonian Monte Carlo. Adv. Neural. Inf. Process. Syst. **34**, 28482–28495 (2021)

Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Adv. Neural. Inf. Process. Syst. **33**, 6840–6851 (2020)

Hoffman, M, Sountsov, P., Dillon, J.V., et al: Neutra-lizing bad geometry in Hamiltonian Monte Carlo using neural transport (2019). arXiv:1903.03704

Hoffman, M.D.: Learning deep latent Gaussian models with Markov chain Monte Carlo. In: International Conference on Machine Learning, pp. 1510–1519 (2017)

Hoffman, M.D., Gelman, A.: The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. J. Mach. Learn. Res. **15**(1), 1593–1623 (2014)

Hoffman, M.D., Johnson, M.J.: Elbo surgery: yet another way to carve up the variational evidence lower bound. In: Workshop in Advances in Approximate Bayesian Inference, NIPS (2016)

Jiang, Z., Zheng, Y., Tan, H., et al: Variational deep embedding: an unsupervised and generative approach to clustering. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence, pp. 1965–1972 (2017)

Kingma, D.P., Welling, M.: Auto-encoding Variational Bayes. In: Proceedings of the 2nd International Conference on Learning Representations (ICLR) (2014)

Kingma, D.P., Salimans, T., Jozefowicz, R., et al: Improved variational inference with inverse autoregressive flow. In: Advances in Neural Information Processing Systems, pp. 4743–4751 (2016)

Klushyn, A., Chen, N., Kurle, R., et al: Learning hierarchical priors in VAEs. Advances in Neural Information Processing Systems 32, pp. 2870–2879 (2019)

Kuzina, A., Welling, M., Tomczak, J.M.: Alleviating adversarial attacks on variational autoencoders with MCMC. In: Advances in Neural Information Processing Systems (2022)

Lee, Y.T., Shen, R., Tian, K.: Logsmooth gradient concentration and tighter runtimes for metropolized Hamiltonian Monte Carlo. In: Conference on Learning Theory, PMLR, pp. 2565–2597 (2020)

Lee, Y.T., Shen, R., Tian, K.: Lower bounds on metropolized sampling methods for well-conditioned distributions. Adv. Neural. Inf. Process. Syst. **34**, 18812–18824 (2021)

Levy, D., Hoffman, M.D., Sohl-Dickstein, J.: Generalizing Hamiltonian Monte Carlo with neural networks. In: International Conference on Learning Representations (2018)

Li, C., Wang, Y., Li, W., et al Forward chi-squared divergence based variational importance sampling (2023). arXiv:2311.02516

Li, Z., Chen, Y., Sommer, F.T.P: A neural network MCMC sampler that maximizes proposal entropy (2020). arXiv:2010.03587

Livingstone, S., Betancourt, M., Byrne, S., et al.: On the geometric ergodicity of Hamiltonian Monte Carlo. Bernoulli **25**(4A), 3109–3138 (2019)

Locatello, F., Dresdner, G., Khanna, R., et al Boosting black box variational inference. In: Advances in Neural Information Processing Systems, pp. 3401–3411 (2018)

Louizos, C., Welling, M.: Structured and efficient variational deep learning with matrix Gaussian posteriors. In: Proceedings of the 33rd International Conference on Machine Learning (2016)

Louizos, C., Welling, M.: Multiplicative normalizing flows for variational bayesian neural networks. In: International Conference on Machine Learning, pp. 2218–2227 (2017)

Lucas, J., Tucker, G., Grosse, R.B., et al: Don't blame the ELBO! a linear VAE perspective on posterior collapse. In: Advances in Neural Information Processing Systems, pp. 9408–9418 (2019)

Maaløe, L., Fraccaro, M., Liévin, V., et al.: Biva: a very deep hierarchy of latent variables for generative modeling. Adv. Neural. Inf. Process. Syst. **32**, 6551–6562 (2019)

Mangoubi, O., Vishnoi, N.K.: Nonconvex sampling with the metropolis-adjusted langevin algorithm. In: Conference on Learning Theory, PMLR, pp. 2259–2293 (2019)

Mescheder, L., Nowozin, S., Geiger, A.: Adversarial variational Bayes: unifying variational autoencoders and generative adversarial networks. In: International Conference on Machine learning (ICML) (2017)

Molchanov, D., Kharitonov, V., Sobolev, A., et al: Doubly semi-implicit variational inference. In: The 22nd International Conference on Artificial Intelligence and Statistics, PMLR, pp. 2593–2602 (2019)

Nesterov, Y.: Primal-dual subgradient methods for convex problems. Math. Program. **120**(1), 221–259 (2009)

Nijkamp, E., Pang, B., Han, T., et al: Learning multi-layer latent variable model via variational optimization of short run MCMC for approximate inference. In: European Conference on Computer Vision. Springer, pp. 361–378 (2020)

Pandey, K., Mandt, S.: A complete recipe for diffusion generative models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 4261–4272 (2023)

Pang, B., Han, T., Nijkamp, E., et al,: Learning latent space energy-based prior model. Advances in Neural Information Processing Systems **33**, pp.21994–22008 (2020)

Papamakarios, G., Nalisnick, E., Rezende, D.J., et al: Normalizing flows for probabilistic modeling and inference (2019). arXiv:1912.02762

Papaspiliopoulos, O., Roberts, G.O., Sköld, M.: A general framework for the parametrization of hierarchical models. Statistical Science, pp. 59–73 (2007)

Peis, I., Ma, C., Hernández-Lobato, J.M.: Missing data imputation and acquisition with deep hierarchical models and Hamiltonian Monte Carlo (2022). arXiv:2202.04599

Ranganath, R., Tran, D., Blei, D.M.: Hierarchical variational models. In: International Conference on Machine Learning (2016)

Rezende, D., Mohamed, S.: Variational inference with normalizing flows. In: Proceedings of The 32nd International Conference on Machine Learning, pp. 1530–1538 (2015)

Rezende, D.J., Mohamed, S., Wierstra, D.: Stochastic backpropagation and approximate inference in deep generative models. In: Proceedings of the 31st International Conference on Machine Learning (ICML-14), pp. 1278–1286 (2014)

Roberts, G., Gelman, A., Gilks, W.: Weak convergence and optimal scaling of random walk metropolis algorithms. Ann. Appl. Probab. **7**(1), 110–120 (1997)

Rombach, R., Blattmann, A., Lorenz, D., et al: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10684–10695 (2022)

Rosca, M., Lakshminarayanan, B., Mohamed, S.: Distribution matching in variational inference (2018). arXiv:1802.06847

Ruiz, F., Titsias, M.: A Contrastive divergence for combining variational inference and MCMC. In: International Conference on Machine Learning, pp. 5537–5545 (2019)

Ruiz, F.J., Titsias, M.K., Cemgil, T., et al: Unbiased gradient estimation for variational auto-encoders using coupled Markov chains. In: Uncertainty in Artificial Intelligence, PMLR, pp. 707–717 (2021)

Salimans, T., Kingma, D.P., Welling, M., et al: Markov Chain Monte Carlo and variational inference: bridging the gap. In: ICML, pp. 1218–1226 (2015)

Salimans, T., Karpathy, A., Chen, X., et al: Pixelcnn++: improving the pixelcnn with discretized logistic mixture likelihood and other modifications. In: International Conference on Learning Representations (2017)

Singhal, R., Goldstein, M., Ranganath, R.: Where to diffuse, how to diffuse and how to get back: automated learning in multivariate diffusions. In: International Conference on Learning Representations (2023)

Sinha, A., Song, J., Meng, C., et al.: D2c: diffusion-decoding models for few-shot conditional generation. Adv. Neural. Inf. Process. Syst. **34**, 12533–12548 (2021)

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., et al: Deep unsupervised learning using nonequilibrium thermodynamics. In: International Conference on Machine Learning, PMLR, pp. 2256–2265 (2015)

Sønderby, C.K., Raiko, T., Maaløe, L., et al.: Ladder variational autoencoders. Adv. Neural. Inf. Process. Syst. **29**, 3738–3746 (2016)

Song, Y., Ermon, S.: Generative modeling by estimating gradients of the data distribution. Advances in Neural Information Processing Systems 32, pp. 11918–11930 (2019)

Song, Y., Sohl-Dickstein, J., Kingma, D.P., et al: Score-based generative modeling through stochastic differential equations. In: International Conference on Learning Representations (2020)

Taniguchi, S., Iwasawa, Y., Kumagai, W., et al: Langevin autoencoders for learning deep latent variable models (2022). arXiv:2209.07036

Thin, A., Kotelevskii, N., Denain, J.S., et al: Metflow: a new efficient method for bridging the gap between Markov Chain Monte Carlo and variational inference (2020). arXiv:2002.12253

Tipping, M.E., Bishop, C.M.: Probabilistic principal component analysis. J. R. Stat. Soc. Ser. B (Stat. Methodol.) **61**(3), 611–622 (1999)

Titsias, M., Dellaportas, P.: Gradient-based adaptive Markov chain Monte Carlo. In: Advances in Neural Information Processing Systems, pp. 15704–15713 (2019)

Titsias, M., Lázaro-Gredilla, M.: Doubly stochastic variational bayes for non-conjugate inference. In: Proceedings of the 31st International Conference on Machine Learning (ICML-14), pp. 1971–1979 (2014)

Titsias MK (2017) Learning model reparametrizations: implicit variational inference by fitting MCMC distributions. arXiv:1708.01529

Titsias, M.K., Ruiz, F.: Unbiased implicit variational inference. In: The 22nd international conference on artificial intelligence and statistics, pp. 167–176 (2019)

Tomczak, J.M., Welling, M.: VAE with a VampPrior (2017). arXiv:1705.07120

Tran, D., Blei, D., Airoldi, E.M.: Copula variational inference. In: Advances in Neural Information Processing Systems, pp. 3564–3572 (2015)

Tran, D., Ranganath, R., Blei, D.M.: Deep and hierarchical implicit models. arXiv:1702.08896 (2017)

Vahdat, A., Kautz, J.: NVAE: a deep hierarchical variational autoencoder (2020). arXiv:2007.03898

Vahdat, A., Kreis, K., Kautz, J.: Score-based generative modeling in latent space. Advances in Neural Information Processing Systems **34** (2021)

Vincent, P.: A connection between score matching and denoising autoencoders. Neural Comput. **23**(7), 1661–1674 (2011)

Wolf, C., Karl, M., van der Smagt, P.: Variational inference with Hamiltonian Monte Carlo (2016). arXiv:1609.08203

Wu, H., Köhler, J., Noé, F.: Stochastic normalizing flows. Adv. Neural. Inf. Process. Syst. **33**, 5933–5944 (2020)

Wu, K., Schmidler, S., Chen, Y.: Minimax mixing time of the metropolis-adjusted Langevin algorithm for log-concave sampling. J. Mach. Learn. Res. **23**(270), 1–63 (2022)

Yin, M., Zhou, M.: Semi-implicit variational inference. In: International Conference on Machine Learning, pp. 5646–5655 (2018)

Yu, L., Xie, T., Zhu, Y., et al: Hierarchical semi-implicit variational iference with application to diffusion model acceleration. In: Thirty-Seventh Conference on Neural Information Processing Systems (2023)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.