Model Selection for Bayesian Autoencoders

Anonymous Author(s)
Affiliation
Address
email

Abstract

We develop a novel method for carrying out model selection for Bayesian autoencoders (BAEs) by means of prior hyper-parameter optimization. Inspired by the common practice of type-II maximum likelihood optimization and its equivalence to Kullback-Leibler divergence minimization, we propose to optimize the distributional sliced-Wasserstein distance (DSWD) between the output of the autoencoder and the empirical data distribution. The advantages of this formulation are that we can estimate the DSWD based on samples and handle high-dimensional problems. We carry out posterior estimation of the BAE parameters via stochastic gradient Hamiltonian Monte Carlo and turn our BAE into a generative model by fitting a flexible Dirichlet mixture model in the latent space. Consequently, we obtain a powerful alternative to variational autoencoders, which are the preferred choice in modern applications of autoencoders for representation learning with uncertainty. We evaluate our approach qualitatively and quantitatively using a vast experimental campaign on a number of unsupervised learning tasks and show that, in small-data regimes where priors matter, our approach provides state-of-the-art results, outperforming multiple competitive baselines.

1 Introduction

The problem of learning useful representations of data that facilitate the solution of downstream tasks such as clustering, generative modeling and classification, is at the crux of the success of many machine learning applications [see, e.g., 5, and references therein]. From a plethora of potential solutions to this problem, unsupervised approaches based on autoencoders [11] are particularly appealing as, by definition, they do not require label information and have proved effective in tasks such as dimensionality reduction and information retrieval [23].

Autoencoders are neural network models composed of two parts, usually referred to as the encoder and the decoder. The encoder maps each input $\mathbf{x}_i$ to a set of lower-dimensional latent variables $\mathbf{z}_i$. The decoder maps the latent variables $\mathbf{z}_i$ back to the observations $\mathbf{x}_i$. The bottleneck introduced by the low-dimensional latent space is what characterizes the compression and representation learning capabilities of autoencoders. It is not surprising that these models have connections with principal component analysis [3], factor analysis and density networks [35], and latent variable models [30].

In applications where quantification of uncertainty is a primary requirement or where data is scarce, it is important to carry out a Bayesian treatment of these models by specifying a prior distribution over their parameters, i.e., the weights of the encoder/decoder. However, estimating the posterior distribution over the parameters of these models, which we refer to as Bayesian autoencoders (BAEs), is generally intractable and requires approximations. Furthermore, the need to specify priors for a large number of parameters, coupled with the fact that autoencoders are not generative models, has motivated the development of Variational Autoencoders (VAEs) as an alternative that can overcome these limitations [27]. Indeed, VAEs have found tremendous success and have become one of the preferred methods in modern machine-learning applications [see, e.g., 28, and references therein].

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

To recap, three potential limitations of BAEs hinder their widespread applicability in order to achieve a similar or superior adoption to their variational counterpart: (i) lack of generative modeling capabilities; (ii) intractability of inference and (iii) difficulty of setting sensible priors over their parameters. In this work we revisit BAEs and deal with these limitations in a principled way. In particular, we address the first limitation in (i) by employing density estimation in the latent space. Furthermore, we deal with the second limitation in (ii) by exploiting recent advances in Markov chain Monte Carlo (MCMC) and, in particular, stochastic gradient Hamiltonian Monte Carlo (SGHMC) [9]. Finally, we believe that the third limitation (iii), which we refer to as the difficulty of carrying out *model selection*, requires a more detailed treatment because choosing sensible priors for Bayesian neural networks is an extremely difficult problem, and this is the main focus of this work.

Contributions. Specifically, in this paper we provide a novel, practical, and elegant way of performing model selection for BAEs, which allows us to revisit these models for applications where VAEs are currently the primary choice. We start by considering the common practice of estimating prior (hyper-)parameters via type-II maximum likelihood, which is equivalent to minimizing the Kullback-Leibler divergence (KL) between the distribution induced by the BAE and the data generating distribution. Because of the intractability of this objective and the difficulty to estimate it through samples, we resort to an alternative formulation where we replace the KL with the distributional sliced-Wasserstein distance (DSWD) between these two distributions. The advantages of this formulation are that we can estimate the DSWD based on samples and, thanks to the slicing, we can handle large dimensional problems. Once BAE hyper-parameters are optimized, we estimate the posterior distribution over the BAE parameters via SGHMC [9], which is a powerful sampler that operates on mini-batches and has proven effective for Bayesian deep/convolutional networks [49, 57, 25]. Furthermore, we turn our BAE into a generative model by fitting a flexible mixture model in the latent space, namely the Dirichlet Process Mixture Model (DPMM). We evaluate our approach qualitatively and quantitatively using a vast experimental campaign on a number of unsupervised learning tasks, with particular emphasis on the challenging task of generative modeling when the number of observations is small.

1.1 Related work

VAEs provide a theoretically-grounded and popular framework for representation learning and deep generative modeling. However, training VAEs poses considerable practical and theoretical challenges yet to be solved. In practice, the learned aggregated posterior distribution of the encoder rarely matches the latent prior, and this hurts the quality of generated samples. Several methods have been proposed to deal with this problem by using a more expressive form of priors on the latent space [41, 10, 48, 4]. Similar to our work, there is a line of research that employs a form of ex-post density estimation on the learned latent space [12, 8, 16]. Wasserstein Autoencoders (WAEs) [47] impose a new form of regularization on latent space by reformulating the objective function as an optimal transport (OT) problem. There have been previous attempts to apply the Bayesian approach to VAEs. For example, [14] treats the parameters of VAE's encoder and decoder in a Bayesian manner to deal with out-of-distribution samples. Most of these works focus on imposing prior or regularization on the latent or weight space of autoencoders. In this work, we take a different route, as we aim to impose prior knowledge directly on the output space. Indeed, our work is motivated by recent attempts to rethink prior specification for Bayesian neural networks (BNNs). It is extremely difficult to choose a sensible prior on the parameters of BNNs [see, e.g., 39, and references therein] because their effect on the distribution of the induced functions is difficult to characterize. Thus, recent attempts in the literature have turned their attention towards defining priors in the space of functions [40, 54, 19, 46]. Closest to our work is that of [49], which matches the functional prior induced by BNNs to Gaussian Process (GP) priors by means of the Kantorovich-Rubinstein dual form of the Wasserstein distance. Different from this line of works, we consider a general framework to impose a functional prior for BNNs in an unsupervised learning setting.

2 Preliminaries on Bayesian Autoencoders

An autoencoder (AE) is a neural network parameterized by a set of parameters $\mathbf{w}$, which transforms an unlabelled dataset, $\mathbf{x} \stackrel{\text{def}}{=} \{\mathbf{x}_n\}_{n=1}^N$, into a set of reconstructions $\hat{\mathbf{x}} \stackrel{\text{def}}{=} \{\hat{\mathbf{x}}_n\}_{n=1}^N$, with $\mathbf{x}_n, \hat{\mathbf{x}}_n \in \mathbb{R}^D$. An AE is composed of two components: (1) an encoder f_{enc} which maps an input sample $\mathbf{x}_n$ to a

latent code $\mathbf{z}_n \in \mathbb{R}^K$, $K \ll D$; and (2) a decoder f_{dec} which maps the latent code to a reconstructed datapoint $\hat{\mathbf{x}}_n$. In short, $\hat{\mathbf{x}} = f(\mathbf{x}; \mathbf{w}) = (f_{\text{dec}} \circ f_{\text{enc}})(\mathbf{x})$, where we denote $\mathbf{w} := \{\mathbf{w}_{\text{enc}}, \mathbf{w}_{\text{dec}}\}$ the union of parameters of the encoder and decoder. The Bayesian treatment of AEs dictates that a prior distribution $p(\mathbf{w})$ is placed over all parameters of f_{enc} and f_{dec}, and that this prior knowledge is transformed into a posterior distribution by means of Bayes' theorem,

$$p(\mathbf{w} \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid \mathbf{w}) p(\mathbf{w})}{p(\mathbf{x})}, \tag{1}$$

where $p(\mathbf{x} \mid \mathbf{w})$ is the conditional likelihood that factorizes as $p(\mathbf{x} \mid \mathbf{w}) = \prod_{n=1}^{N} p(\mathbf{x}_n \mid \mathbf{w})$. Note that each conditional likelihood term is determined by the model architecture, the choice of $\mathbf{w}$, and the input $\mathbf{x}_n$, but in order to keep the notation uncluttered, we write them simply as $p(\mathbf{x}_n \mid \mathbf{w})$.

Likelihood model. In the Bayesian scheme, the prior and likelihood are both modeling choices. Before giving an in-depth treatment on priors for BAEs in the next section, we briefly discuss the likelihood, which can be chosen according to the type of data. In our experiments, we mainly investigate image datasets, where pixel values are normalized in the $[0, 1]$ range. Therefore, we rely on the *continuous Bernoulli* distribution [33]:

$$p(\mathbf{x}_n \mid \mathbf{w}) = \prod_{i=1}^{D} K(\lambda_i) \lambda_i^{\mathbf{x}_{n,i}} (1 - \lambda_i)^{1 - \mathbf{x}_{n,i}} := p(\mathbf{x}_n \mid \hat{\mathbf{x}}_n), \tag{2}$$

where $K(\lambda_i)$ is a properly defined normalization constant [33] and $\lambda_i = f_i(\mathbf{x}_n; \mathbf{w}) = \hat{\mathbf{x}}_{n,i} \in [0, 1]$ is the i-th output from the BAE given the input $\mathbf{x}_n$. We note that, as $\hat{\mathbf{x}}_n$ depends deterministically on $\mathbf{w}$, we will use the above expression to refer to both $p(\mathbf{x}_n \mid \mathbf{w})$ and $p(\mathbf{x}_n \mid \hat{\mathbf{x}}_n)$, where the latter term will be of crucial importance when we define the functional prior induced over the reconstruction $\hat{\mathbf{x}}$.

Inference. Although the posterior of BAEs is analytically intractable, it can be approximated by variational methods or using MCMC sampling. Within the large family of approximate Bayesian inference schemes, SGHMC [9] allows us to sample from the true posterior by efficiently simulating a Hamiltonian system [42]. Differently from more traditional methods, SGHMC can scale up to large datasets by relying on noisy but unbiased estimates of the potential energy function $U(\mathbf{w}) = -\sum_{n=1}^{N} \log p(\mathbf{x}_n \mid \mathbf{w}) - \log p(\mathbf{w})$. These can be computed by considering a mini-batch of size M of the data and approximating $\sum_{n=1}^{N} \log p(\mathbf{x}_n \mid \mathbf{w}) \approx \frac{N}{M} \sum_{j \in \mathcal{I}_M} \log p(\mathbf{x}_j \mid \mathbf{w})$, where $\mathcal{I}_M$ is a set of M random indices. More details on SGHMC can be found in the Appendix.

Pathologies of standard priors. The choice of the prior is important for the Bayesian treatment of any model as it characterizes the hypothesis space [36, 38]. Specifically for BAEs, one should note that placing a prior on the parameters of the encoder and decoder has an implicit effect on the prior over the network output (i.e. the reconstruction). In addition, the highly nonlinear nature of these models implies that interpreting the effect of the architecture is theoretically intractable and practically challenging. Several works argue that a vague prior such as $\mathcal{N}(0, 1)$ is good enough for some tasks and models, like classification with convolutional neural networks (CNNs) [53].

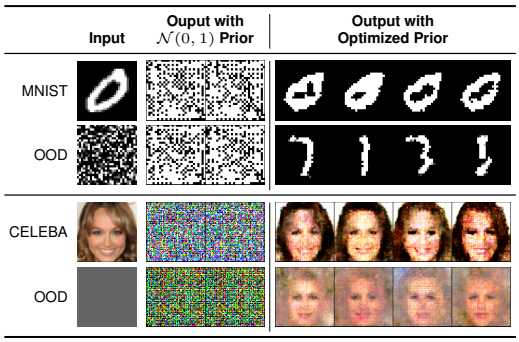

Figure 1: Realizations sampled from different priors given an input image. OOD stands for out-of-distribution.

However, for BAEs this is not enough, as illustrated in Fig. 1. The realizations obtained by sampling weights/biases from a $\mathcal{N}(0, 1)$ prior indicate that this choice provides poor inductive bias. Meanwhile, by encoding better beliefs via an optimized prior, which is the focus of the next section, the samples can capture main characteristics intrinsic to the data, even when the model is fed with out-of-distribution inputs.

3 Model Selection for Bayesian Autoencoders via Prior Optimization

One of the main advantages of the Bayesian paradigm is that we can incorporate prior knowledge into the model in a principled way. Let us assume a prior distribution $p_{\boldsymbol{\psi}}(\mathbf{w})$ on the parameters of the AE network, where now we are explicit on the set of (hyper-)parameters that determine the prior, i.e., $\boldsymbol{\psi}$. Specifying this prior for the BAE is not straightforward due to the complex nonlinear forms of f_{enc} and f_{dec}, which induce a non-trivial effect on the output (functional) prior:

$$p_{\boldsymbol{\psi}}(\hat{\mathbf{x}}) = \int f(\mathbf{x}; \mathbf{w}) p_{\boldsymbol{\psi}}(\mathbf{w}) \mathrm{d}\mathbf{w}, \tag{3}$$

where $\hat{\mathbf{x}} = f(\mathbf{x}; \mathbf{w})$ is the functional output of the BAE. Although $p_{\boldsymbol{\psi}}(\hat{\mathbf{x}})$ cannot be evaluated analytically, it is possible to draw samples from it.

Prior parameterization. The only two requirements needed to design a parameterization for the prior are: to be able to (1) draw samples from it and (2) to compute its log-density at any point. The latter is required by many inference algorithms such as SGHMC. We consider a fully-factorized Gaussian prior over weights and biases at layer l:

$$p(w_l) = \mathcal{N}(w_l; \mu_{l_w}, \sigma^2_{l_w}), \quad p(b_l) = \mathcal{N}(b_l; \mu_{l_b}, \sigma^2_{l_b}), \tag{4}$$

Notice that, as we shall see in § 3.2 and § 3.3, in order to estimate our prior hyper-parameters, we will require gradient back-propagation through the stochastic variables w_l and b_l. Thus, we treat these parameters in a deterministic manner by means of the reparameterization trick [45, 27].

3.1 Another route for Bayesian Occam's razor

A common way to estimate hyper-parameters (i.e., prior parameters $\boldsymbol{\psi}$) is to rely on the Bayesian Occam's razor (a.k.a. *empirical Bayes*), which dictates that the marginal likelihood $p_{\boldsymbol{\psi}}(\mathbf{x})$ should be optimized with respect to $\boldsymbol{\psi}$. There are countless examples where such simple procedure succeeds in practice [see, e.g., 44]. The marginal likelihood is obtained by marginalizing out the outputs $\hat{\mathbf{x}}$ and the model parameters $\mathbf{w}$,

$$p_{\boldsymbol{\psi}}(\mathbf{x}) = \int p(\mathbf{x} \,|\, \hat{\mathbf{x}}) p_{\boldsymbol{\psi}}(\hat{\mathbf{x}}) \mathrm{d}\hat{\mathbf{x}}\,, \tag{5}$$

where $p(\mathbf{x} \,|\, \hat{\mathbf{x}})$ and $p_{\boldsymbol{\psi}}(\hat{\mathbf{x}})$ are given by Eq. 2 and Eq. 3, respectively. Unfortunately, in our context it is impossible to carry out this optimization due to the intractability of Eq. 5.

Classic results in the statistics literature draw parallels between maximum likelihood estimation (MLE) and KL minimization [2],

$$\arg\max_{\boldsymbol{\psi}} \int \pi(\mathbf{x}) \log p_{\boldsymbol{\psi}}(\mathbf{x}) \mathrm{d}\mathbf{x} = \arg\min_{\boldsymbol{\psi}} \underbrace{\int \pi(\mathbf{x}) \log \frac{\pi(\mathbf{x})}{p_{\boldsymbol{\psi}}(\mathbf{x})} \mathrm{d}\mathbf{x}}_{\mathrm{KL}[\pi(\mathbf{x}) \,\|\, p_{\boldsymbol{\psi}}(\mathbf{x})]}\,, \tag{6}$$

where $\pi(\mathbf{x})$ is the true data distribution. This equivalence provides us with an interesting insight on an alternative view of marginal likelihood optimization as minimization of the divergence between the true data distribution and the marginal $p_{\boldsymbol{\psi}}(\mathbf{x})$.

This alternative view still does not help us in obtaining a viable optimization strategy, even if we use $\mathbf{x}$ to estimate an empirical $\tilde{\pi}(\mathbf{x})$; the empirical evaluation and optimization of KL divergences is indeed a well-known challenging problem [15], although this is possible (for the KL or any other f-divergence), for example, by leveraging results from convex analysis such as in the convex minimization framework of [43]. However, we can now attempt to replace the intractable KL divergence with another divergence to recover tractability. Inspired by recent works on deriving sensible priors for Bayesian neural networks [49], we employ the Wasserstein distance, which, as we will see later, can be estimated efficiently using samples only, even for high-dimensional distributions.

To summarize: (1) we would like to do prior selection by carrying out type-II MLE; (2) the MLE objective is analytically intractable but the connection with KL minimization allows us to (3) swap the divergence with the Wasserstein distance, yielding a practical framework for choosing priors.

3.2 Matching the marginal distribution to the data distribution via Wasserstein distance minimization

Given the two probability measures π and $p_{\boldsymbol{\psi}}$, both defined on $\mathbb{R}^D$ for simplicity, the p-Wasserstein distance between π and $p_{\boldsymbol{\psi}}$ is given by

$$W_p^p(\pi, p_{\boldsymbol{\psi}}) = \inf_{\gamma \in \Gamma(\pi, p_{\boldsymbol{\psi}})} \int \|\mathbf{x} - \mathbf{x}'\|^p \gamma(\mathbf{x}, \mathbf{x}') \mathrm{d}\mathbf{x}\mathrm{d}\mathbf{x}' \,, \tag{7}$$

where $\Gamma(\pi, p_{\boldsymbol{\psi}})$ is the set of all possible distributions $\gamma(\mathbf{x}, \mathbf{x}')$ such that the marginals are $\pi(\mathbf{x})$ and $p_{\boldsymbol{\psi}}(\mathbf{x}')$ [51]. While usually analytically unavailable or computationally intractable, for $D = 1$ the distance has a simple closed form solution, that can be easily estimated using samples only [29].

The distributional sliced-Wasserstein distance (DSWD) takes advantage of this result by projecting the estimation of distances for high-dimensional distributions into simpler estimation of multiple distances in one dimension. The projection is done using the Radon transform $\mathcal{R}$, an operator that maps a generic density function φ defined in $\mathbb{R}^D$ to the set of its integrals over hyperplanes in $\mathbb{R}^D$,

$$\mathcal{R}\varphi(t, \boldsymbol{\theta}) := \int \varphi(\mathbf{r}) \delta(t - \mathbf{r}^\top \boldsymbol{\theta}) \mathrm{d}\mathbf{r} \,, \quad \forall t \in \mathbb{R} \,, \;\; \forall \boldsymbol{\theta} \in \mathbb{S}^{D-1} \,, \tag{8}$$

where $\mathbb{S}^{D-1}$ is the unit sphere in $\mathbb{R}^D$ and $\delta(\cdot)$ is the Dirac delta [20]. Using the Radon transform, for a given direction (or *slice*) $\boldsymbol{\theta}$ we can project the two densities π and $p_{\boldsymbol{\psi}}$ into one dimension and we can solve the optimal transport problem in this projected space. Furthermore, to avoid unnecessary computations, instead of considering all possible directions in $\mathbb{S}^{D-1}$, DSWD proposes to find the optimal probability measure of slices $\sigma(\boldsymbol{\theta})$ on the unit sphere $\mathbb{S}^{D-1}$,

$$DSW_p(\pi, p_{\boldsymbol{\psi}}) := \sup_{\sigma \in \mathbb{M}_C} \Big(\mathbb{E}_{\sigma(\boldsymbol{\theta})} W_p^p \big(\mathcal{R}\pi(t, \boldsymbol{\theta}), \mathcal{R}p_{\boldsymbol{\psi}}(t, \boldsymbol{\theta}) \big) \Big)^{1/p}, \tag{9}$$

where, for $C > 0$, $\mathbb{M}_C$ is the set of probability measures σ such that $\mathbb{E}_{\boldsymbol{\theta}, \boldsymbol{\theta}' \sim \sigma} \big[\boldsymbol{\theta}^\top \boldsymbol{\theta}' \big] \leq C$ (a constraint that aims to avoid directions to lie in only one small area). The direct computation of DSW_p in Eq. 9 is still challenging but admits an equivalent dual form,

$$\sup_{h \in \mathcal{H}} \left\{ \Big(\mathbb{E}_{\bar{\sigma}(\boldsymbol{\theta})} \big[W_p^p \big(\mathcal{R}\pi(t, h(\boldsymbol{\theta})), \mathcal{R}p_{\boldsymbol{\psi}}(t, h(\boldsymbol{\theta})) \big) \big] \Big)^{1/p} - \lambda_C \mathbb{E}_{\boldsymbol{\theta}, \boldsymbol{\theta}' \sim \bar{\sigma}} \Big[\big| h(\boldsymbol{\theta})^\top h(\boldsymbol{\theta}') \big| \Big] \right\} + \lambda_C C \,, \tag{10}$$

where $\bar{\sigma}$ is a uniform distribution in $\mathbb{S}^{D-1}$, $\mathcal{H}$ is the set of functions $h : \mathbb{S}^{D-1} \to \mathbb{S}^{D-1}$ and λ_C is a regularization hyper-parameter. The formulation in Eq. 10 is obtained by employing the Lagrangian duality theorem and by reparameterizing $\sigma(\boldsymbol{\theta})$ as push-forward transformation of a uniform measure in $\mathbb{S}^{D-1}$ via h. Now, by parameterizing h using a deep neural network with parameters $\boldsymbol{\phi}$, defined as $h_{\boldsymbol{\phi}}$, Eq. 10 becomes an optimization problem with respect to the network parameters. The final step is to approximate the analytically intractable expectations with Monte Carlo integration,

$$\max_{\boldsymbol{\phi}} \left\{ \left[\frac{1}{K} \sum_{i=1}^{K} \big[W_p^p \big(\mathcal{R}\pi(t, h_{\boldsymbol{\phi}}(\boldsymbol{\theta}_i)), \mathcal{R}p_{\boldsymbol{\psi}}(t, h_{\boldsymbol{\phi}}(\boldsymbol{\theta}_i)) \big) \big] \right]^{1/p} - \frac{\lambda_C}{K^2} \sum_{i,j=1}^{K} |h_{\boldsymbol{\phi}}(\boldsymbol{\theta}_i)^\top h_{\boldsymbol{\phi}}(\boldsymbol{\theta}_j)| \right\} + \lambda_C C \,,$$

with $\boldsymbol{\theta}_i \sim \bar{\sigma}(\boldsymbol{\theta})$. Finally, we can use stochastic gradient methods to update $\boldsymbol{\phi}$ and then use the resulting optima for the estimation of the original distance. We encourage the reader to check the detailed explanation of this formulation, including its derivation and some practical considerations for implementation, available in the Appendix.

3.3 Summary

We aim at learning the prior on the BAE parameters by optimizing the marginal $p_{\boldsymbol{\psi}}(\mathbf{x})$ obtained after integrating out the weights from the joint $p_{\boldsymbol{\psi}}(\mathbf{x}, \mathbf{w})$. The connection with *empirical Bayes* and KL minimization suggests that we can find the optimal $\boldsymbol{\psi}^\star$ by minimizing the KL between the true data distribution $\pi(\mathbf{x})$ and the marginal $p_{\boldsymbol{\psi}}(\mathbf{x})$. However, matching these two distributions is non-trivial due to their high dimensionality and the unavailability of their densities. To overcome this problem, we propose a sample-based approach using the distributional sliced 2-Wasserstein distance (Eq. 10) as objective:

$$\boldsymbol{\psi}^\star = \arg\min_{\boldsymbol{\psi}} \Big[DSW_2 \big(p_{\boldsymbol{\psi}}(\mathbf{x}), \pi(\mathbf{x}) \big) \Big]. \tag{11}$$

This objective function is flexible and does not require the closed-form of either $p_{\boldsymbol{\psi}}(\mathbf{x})$ or $\pi(\mathbf{x})$. The only requirement is that we can draw samples from these two distributions. Note that we can sample from $p_{\boldsymbol{\psi}}(\mathbf{x})$, by first computing $\hat{\mathbf{x}}$ after sampling from $p_{\boldsymbol{\psi}}(\mathbf{w})$ and then perturbing the generated $\hat{\mathbf{x}}$ by sampling from the likelihood $p(\mathbf{x} \mid \hat{\mathbf{x}})$. For the continuous Bernoulli likelihood this operation can be implemented by using the reparameterization form that allows to backpropagate gradients [33].

4 Experiments

Competing approaches. We compare our proposal with a wide selection of methods from the literature. For autoencoding methods, we choose the vanilla **VAE** [27], the **$\boldsymbol{\beta}$-VAE** [22] and **WAE** (Wasserstein AE) [47]. In addition, we consider models with more complex encoders (**VAE + Sylvester flows** [50]), generators (**2-stage VAE** [12]), and priors (**VAE + VampPrior** [48]). For CELEBA we also include a comparison with Generative Adversarial Networks (GANs), with the vanilla setup of **NS-GAN** [18, 34] and the more recent **DiffAugment-GAN** [58, 26]. Finally, we also compare against BAE with the standard $\mathcal{N}(0, 1)$ prior. Unless otherwise stated, all models—including ours—share the same latent dimensionality ($K = 50$). We defer a more detailed description of these models and architectures to the Appendix.

Generative process. Differently from VAEs and other methods, deterministic and Bayesian AEs are not generative models. To generate new samples with BAEs we employ ex-post density estimation over the learned latent space, by fitting a density estimator $p_{\vartheta}(\mathbf{z})$ to $\{\mathbf{z}_i = \mathbb{E}_{p(\mathbf{w}_{\text{enc}} \mid \mathbf{x})}[f_{\text{enc}}(\mathbf{x}_i; \mathbf{w}_{\text{enc}})]\}$. In this work, we employ a nonparametric model for density estimation based on Dirichlet Process Mixture Model (DPMM) [7], so that its complexity is automatically adapted to the data; see also [6] for alternative ways to turn AEs into generative models. After estimating $p_{\vartheta}(\mathbf{z})$, a new sample can be generated by drawing $\mathbf{z}_{\text{new}}$ from $p_{\vartheta}(\mathbf{z})$ and $\hat{\mathbf{x}}_{\text{new}} = \mathbb{E}_{p(\mathbf{w}_{\text{dec}} \mid \mathbf{x})}[f_{\text{dec}}(\mathbf{z}_{\text{new}}; \mathbf{w}_{\text{dec}})]$.

Evaluation metrics. To evaluate the reconstruction quality, we use the test log-likelihood (LL), which tells us how likely the test targets are generated by the corresponding model. The predictive log-likelihood is a proper scoring rule that depends on both the accuracy of predictions and their uncertainty [17]. To assess the quality of the generated images, instead, we employ the widely used Fréchet Inception Distance (FID) [21]. We note that, as GANs are not inherently equipped with an explicit likelihood model, we only report their FID scores. Finally, all our experiments and evaluations are repeated four times, with different random training splits.

4.1 Analysis of the effect of the prior

To demonstrate the effect of our model selection strategy, we consider scenarios in the small-data regime where the prior might not be necessarily tuned on the training set. In this way we are able to impose inductive bias beyond what is available in the training data. We investigate two cases:

- MNIST [31]: We use 100 examples of the 0 digits to tune the prior. The training set consists of examples of 1-9 digits, whereas the test set contains 10 000 instances of all digits. We aim to demonstrate the ability of our approach to incorporate prior knowledge about completely unseen data with different characteristics into the model.
- FREY-YALE [13]: We use 1 956 examples of FREY faces to optimize the prior. The training set and test set are comprised of YALE faces. We demonstrate the benefit of using a different dataset but from the same domain (e.g. face images) to specify the prior distribution.

Visual inspection. Fig. 2 shows some qualitative results (additional images are available in the Appendix), while Fig. 3 shows the convergence of the Wasserstein distance during prior optimization in our proposal. From a visual inspection we see that, on MNIST, by encoding knowledge about the "0" digit into the prior, the BAE can reconstruct this digit fairly well although we only use "1" to "9" digits for inference (differently from the BAE with standard prior). Similarly, on FREY-YALE, we see that by encoding knowledge from another dataset in the same domain, the optimized prior can impose a softer constraint compared to using directly this dataset for inference. In addition, if we use directly the union of FREY and YALE faces for training (methods denoted with a ★), VAE yields images that are similar to FREY instead of YALE faces, while generated images from BAE with $\mathcal{N}(0, 1)$ prior are of lower quality. This again highlights the advantage of our approach to specifying an informative prior compared to using that data for training. Another important benefit of our Bayesian treatment of AEs is that we can quantify the *uncertainty* for both reconstructed and generated images. The last row of Fig. 2 illustrates the uncertainty estimate corresponding to the BAE with optimized prior on

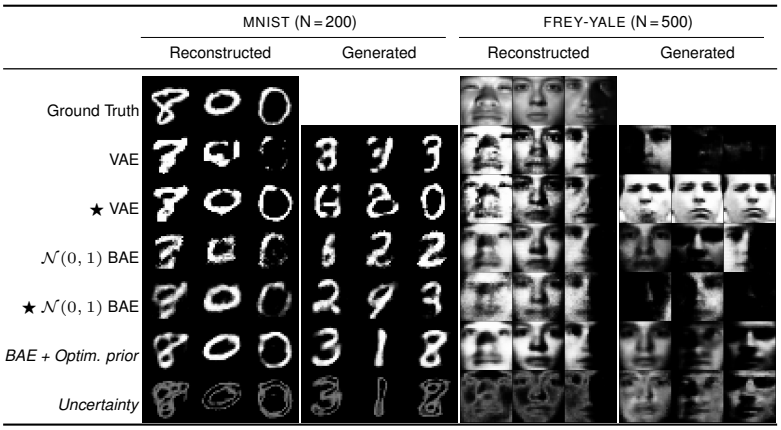

Figure 2: Qualitative evaluation for MNIST and YALE. Here, ★ indicates using the union of the training data and the data used to optimize prior to train the model. The last row depicts standard deviation of reconstructed/generated images estimated by BAE using the optimized prior.

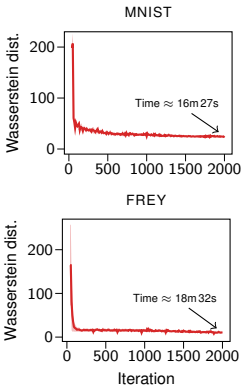

Figure 3: Convergence of the proposed Wasserstein minimization scheme.

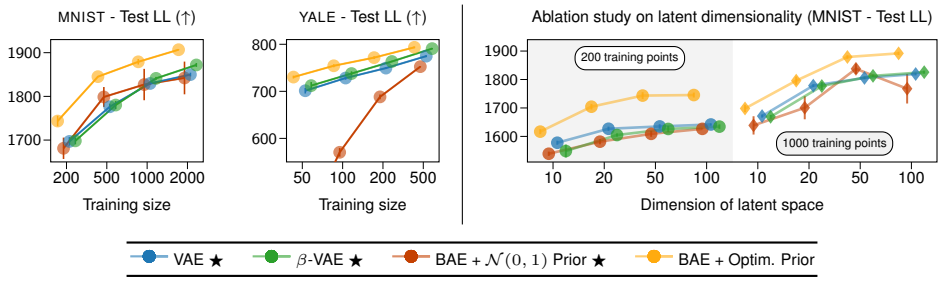

Figure 4: Test log-likelihood (LL) of MNIST and YALE. *Left:* test LL as a function of training size; *Right:* test LL as a function of latent dimensionality.

MNIST and YALE datasets. Our model exhibits increased uncertainty for semantically and visually challenging pixels such as the left part of the second "0" digit image in the MNIST example. We also observe that the uncertainty is greater for generated images compared to reconstructed images as illustrated in the YALE example. This is reasonable because the reconstruction process is guided by the input data rather than synthesizing new data according to a random latent code.

Visualization of inductive bias on MNIST. To have an intuition of the inductive bias induced by the optimized prior, we visualize a low-dimensional projection of parameters sampled from the prior and the posterior [24]. As we see in Fig. 5, the hypothesis space induced by the $\mathcal{N}(0,1)$ prior is huge, compared to where the true solution should lie. Effectively this is another visualization of the famous Bayesian Occam's razor plot by David MacKay [37], where the model has very high complexity and poor inductive biases. On the other hand, by considering our proposal to do *model selection*, the hypothesis space of the optimized prior is reduced to regions close to the full posterior. Additional visualizations are available in the Appendix.

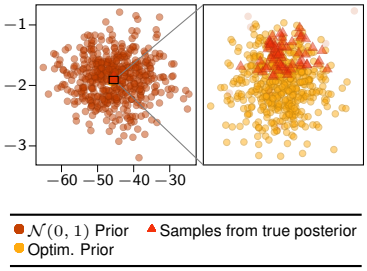

Figure 5: Visualization in 2D of samples from priors and posteriors of BAE's parameters. The setup is the same as before with MNIST.

Quantitative evaluation. For a quantitative analysis we rely on Fig. 4, where we study the effect on the reconstruction quality of different training sizes (on the *left*) and different latent dimensions (on the *right*). Since we observed that the results of VAE variants are not significantly different, we only show the results for β-VAE and we leave the extended results to the Appendix. From this experiment we can draw important conclusions. The BAE with optimized priors clearly outperforms the competing methods (and the BAE with standard prior) in the inference task for all training sizes,

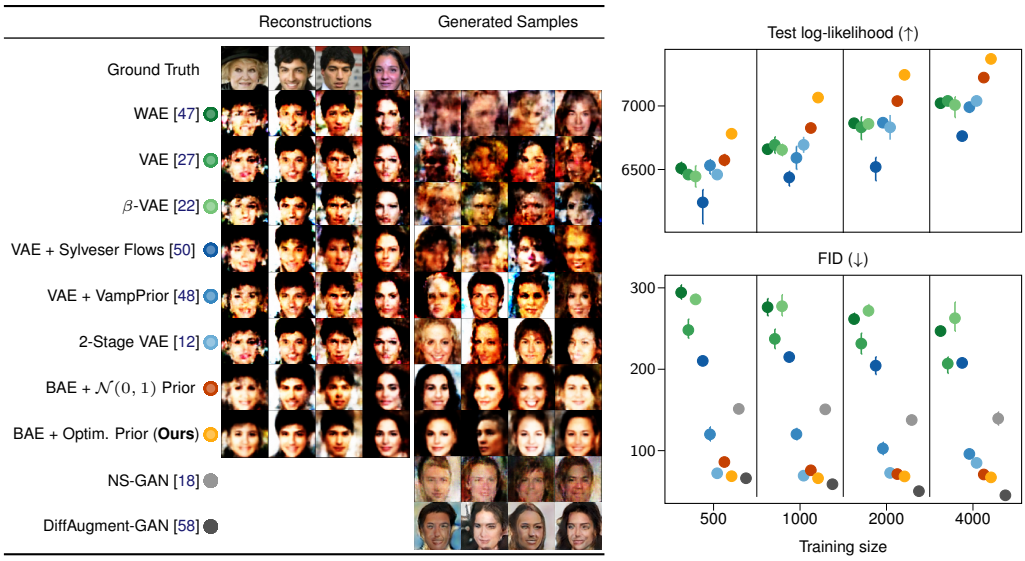

Figure 6: Qualitative (*left*) and quantitative evaluation (*right*) on CELEBA. The markers and bars represent the means and one standard deviations, respectively. In the (*left*) figure, the sizes of training data and the data for optimizing prior are 500 and 1000, respectively. The higher the log-likelihood (LL) and the lower FID the better.

with slightly diminishing effect for larger sets, as expected. Also, this pattern is true when looking at different latent dimensions (Fig. 4, *left*), where regardless of the dimensionality of the latent space, BAEs with optimized priors deliver higher performances.

4.2 Reconstruction and generation of CELEBA

We now look at a more challenging benchmark, the CELEBA dataset [32]. For our proposal, we use 1 000 examples that are randomly chosen from the original training set to learn the prior distribution. The test set consists of about 20 000 images. The goal of this experiment is to evaluate whether sacrificing part of the training data to specify a good prior is beneficial when compared to using that data for training the model. Fig. 6 shows qualitative results for the competing methods, their corresponding test LLs and FIDs for different training dataset sizes. In terms of test log-likelihoods (LLs) (Fig. 6, *top right*), we observe two clear patterns: (i) that BAE approaches perform considerably better than other methods and (ii) the VAE with Sylvester flows performs consistently poor across dataset sizes. This latter observation indicates that having a more expressive posterior for the encoder is not helpful when considering the small training sizes used in our experiments. More importantly, we see that the BAE using the optimized prior significantly outperforms other methods despite using less data for inference. These results largely agree with the quality of the reconstructions (first column of images in Fig. 6, *left*) in that BAE methods provide more visually appealing reconstructions when compared to other approaches.

We now evaluate the quality of the generated images (second column of images in Fig. 6, *left*) along with their FID scores [21]. Visually, it is clear that images generated from VAEs (standard, β, Sylvester and WAE) are very poor. This failure may originate from the fact that the aggregated posterior distribution of the encoder is not aligned with the prior on the latent space. This problem is more prominent in the case of small training data, where the encoder is not well-trained. The VampPrior tackles this problem by explicitly modeling the aggregated posterior, while 2-stage VAE uses another VAE to estimate the density of the learned latent space. By reducing the effect of the aggregated posterior mismatch, these strategies improve the quality of the generated images remarkably. These results are consistent with their corresponding FID scores (Fig. 6, *bottom right*) where we also see that BAE using the optimized prior consistently outperforms all variants of VAEs and NS-GAN. Finally, we see that DiffAugment-GAN, with the exception of using a training size of 500, yields better FID scores. However, this is not surprising as this model uses much more complex network architectures [26], combined with a powerful differentiable augmentation scheme. More importantly, it is clear that with few training samples our method generates more semantically meaningful images then all other approaches, including DiffAugment-GAN.

4.3 Prior adjustment versus posterior tempering

We have shown that the proposed framework for adjusting the prior is compatible with standard Bayesian practices, as it emulates type-II maximum likelihood. In other words, the distribution fitting that we induce by means of Wasserstein distance minimization relates to the marginal output of BAEs, very much in the same spirit of marginal likelihood maximization. The distribution is fit considering *all* possible functions, when marginalized through the likelihood, creating an implicit regularization effect. Our scheme does not give more weight to particular training instances, but it simply restricts the hypothesis space. This is unlike *posterior tempering* [56, 24, 52, 1, 55], which is commonly defined as $p_\tau(\mathbf{w} \,|\, \mathbf{x}) \propto p(\mathbf{x} \,|\, \mathbf{w})^{1/\tau} p(\mathbf{w})$, where $\tau > 0$ is a *temperature* value. With $\tau < 1$, tempering is known to improve performance in the case of small training data and using miss-specified priors, but it corresponds to artificially sharpening the posterior by over-counting the data τ times.

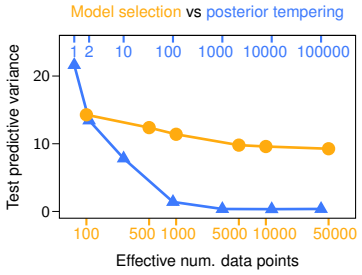

Figure 7: Average test predictive variance as a function of the number of data points used to optimize the prior, and the temperature (i.e. how many times the data points are over-counted).

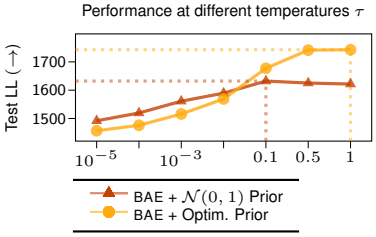

Figure 8: Test performance for temperature scaling with different priors. The dotted lines indicate the best performance.

To demonstrate the differences with our proposal, we setup a comparison on MNIST. In the empirical comparison of Fig. 7, we consider different temperatures and different sets of data points used to optimize the prior. As expected, the tempered posterior quickly collapses on the mode, while the posterior after our treatment retains a sufficiently constant variance, regardless of the number of data points used. It is also interesting to notice that with the $\mathcal{N}(0, 1)$ prior, the best temperature is $\tau = 0.1$, while for our approach that optimizes the prior is $\tau = 1$, further confirming that the model now is well specified (Fig. 8).

5 Conclusions

In this work, we have reconsidered the Bayesian treatment of autoencoders (AE) in light of recent advances in Bayesian neural networks. We have addressed the main challenge of BAEs, so that they can be rendered as viable alternative to generative models such as VAEs. More specifically, we have found that the main limitation of BAEs lies in the difficulty of specifying meaningful priors in the context of highly-structured data, which is ubiquitous in modern machine learning applications. Consequently, we have proposed to specify priors over the autoencoder weights by means of a novel optimization of prior hyper-parameters. Inspired by connections with marginal likelihood optimization, we derived a practical and efficient optimization framework, based on the minimization of the distributional sliced-Wasserstein distance between the distribution induced by the BAE and the data generating distribution. The resulting hyper-parameter optimization strategy leads to a novel way to perform model selection for BAEs, and we showed its advantages in an extensive experimental campaign.

Limitations and ethical concerns. Even if theoretically justified and empirically verified with extensive experimentation, our proposal for model selection still remains a *proxy* to the true marginal likelihood maximization. The DSWD formulation has nice properties of asymptotic convergence and computational tractability, but it may represent only one of the possible solutions. At the same time, we stress that the current literature does not cover this problem of BAEs at all, and we believe our approach is a considerable step towards the development of practical Bayesian methods for representation learning in modern applications characterized by large-scale structured data (including tabular and graph data, which are currently not covered). At the same time, the accessibility to these models to a wider audience and different kind of data might help to widespread harmful applications, which is a concern shared among all generative modeling approaches. An ethical analysis of the consequences of Bayesian priors in unsupervised learning scenarios is also worth an in-depth investigation, which goes beyond the scope of this work.

References

[1] L. Aitchison. A Statistical Theory of Cold Posteriors in Deep Neural Networks. In *International Conference on Learning Representations*, 2021.

[2] H. Akaike. Information Theory and an Extension of the Maximum Likelihood Principle. In *2nd International Symposium on Information Theory, 1973*, pages 268–281. Publishing House of the Hungarian Academy of Sciences, 1973.

[3] P. Baldi and K. Hornik. Neural Networks and Principal Component Analysis: Learning from Examples Without Local Minima. *Neural networks*, 2(1):53–58, 1989.

[4] M. Bauer and A. Mnih. Resampled Priors for Variational Autoencoders. In *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*, volume 89 of *Proceedings of Machine Learning Research*, pages 66–75. PMLR, 2019.

[5] Y. Bengio, A. Courville, and P. Vincent. Representation Learning: A Review and New Perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

[6] Y. Bengio, L. Yao, G. Alain, and P. Vincent. Generalized Denoising Auto-Encoders as Generative Models. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 899–907, 2013.

[7] D. M. Blei and M. I. Jordan. Variational Inference for Dirichlet Process Mixtures. *Bayesian Analysis*, 1(1):121 – 143, 2006.

[8] V. Böhm and U. Seljak. Probabilistic Auto-Encoder. *arXiv preprint arXiv:2006.05479*, 2020.

[9] T. Chen, E. Fox, and C. Guestrin. Stochastic Gradient Hamiltonian Monte Carlo. In *Proceedings of the 31st International Conference on Machine Learning, ICML 2014*, Proceedings of Machine Learning Research, pages 1683–1691, Bejing, China, 22–24 Jun 2014. PMLR.

[10] X. Chen, D. P. Kingma, T. Salimans, Y. Duan, P. Dhariwal, J. Schulman, I. Sutskever, and P. Abbeel. Variational Lossy Autoencoder. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[11] G. W. Cottrell, P. Munro, and D. Zipser. Image Compression by Back Propagation: A Demonstration of Extensional Programming. *Models of Cognition*, pages 208–240, 1989.

[12] B. Dai and D. Wipf. Diagnosing and Enhancing VAE Models. In *International Conference on Learning Representations*, 2019.

[13] Z. Dai, A. C. Damianou, J. González, and N. D. Lawrence. Variational Auto-encoded Deep Gaussian Processes. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[14] E. A. Daxberger and J. M. Hernández-Lobato. Bayesian Variational Autoencoders for Unsupervised Out-of-Distribution Detection. *arXiv preprint arXiv:1912.05651*, 2019.

[15] D. Flam-Shepherd, J. Requeima, and D. Duvenaud. Mapping Gaussian Process Priors to Bayesian Neural Networks. In *NeurIPS workshop on Bayesian Deep Learning*, 2017.

[16] P. Ghosh, M. S. M. Sajjadi, A. Vergari, M. Black, and B. Scholkopf. From Variational to Deterministic Autoencoders. In *International Conference on Learning Representations*, 2020.

[17] T. Gneiting and A. E. Raftery. Strictly Proper Scoring Rules, Prediction, and Estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[19] D. Hafner, D. Tran, T. P. Lillicrap, A. Irpan, and J. Davidson. Noise Contrastive Priors for Functional Uncertainty. In *Proceedings of the 35h Conference on Uncertainty in Artificial Intelligence, UAI 2019*, page 332, Tel Aviv, Israel, 22-25 July 2019 2019. AUAI Press.

[20] S. Helgason. *Integral Geometry and Radon Transforms*. Springer Science & Business Media, 2010.

[21] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6626–6637, 2017.

[22] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[23] G. E. Hinton and R. R. Salakhutdinov. Reducing the Dimensionality of Data with Neural Networks. *Science*, 313(5786):504–507, 2006.

[24] P. Izmailov, W. Maddox, P. Kirichenko, T. Garipov, D. P. Vetrov, and A. G. Wilson. Subspace Inference for Bayesian Deep Learning. In *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI 2019, Tel Aviv, Israel, July 22-25, 2019*, volume 115 of *Proceedings of Machine Learning Research*, pages 1169–1179. AUAI Press, 2019.

[25] P. Izmailov, S. Vikram, M. D. Hoffman, and A. G. Wilson. What Are Bayesian Neural Network Posteriors Really Like? In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 June 2021, Virtual Event*, 2021.

[26] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila. Analyzing and Improving the Image Quality of StyleGAN. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 8107–8116. IEEE, 2020.

[27] D. P. Kingma and M. Welling. Auto-Encoding Variational Bayes. In *International Conference on Learning Representations*, 2014.

[28] D. P. Kingma and M. Welling. An Introduction to Variational Autoencoders. *Foundations and Trends in Machine Learning*, 12(4):307–392, 2019.

[29] S. Kolouri, K. Nadjahi, U. Simsekli, R. Badeau, and G. K. Rohde. Generalized Sliced Wasserstein Distances. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 261–272, 2019.

[30] N. D. Lawrence. Probabilistic Non-linear Principal Component Analysis with Gaussian Process Latent Variable Models. *Journal of Machine Learning Research*, 6:1783–1816, 2005.

[31] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[32] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep Learning Face Attributes in the Wild. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 3730–3738. IEEE Computer Society, 2015.

[33] G. Loaiza-Ganem and J. P. Cunningham. The Continuous Bernoulli: Fixing a Pervasive Error in Variational Autoencoders. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13266–13276, 2019.

[34] M. Lucic, K. Kurach, M. Michalski, S. Gelly, and O. Bousquet. Are GANs Created Equal? A Large-Scale Study. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 698–707, 2018.

[35] D. J. MacKay and M. N. Gibbs. Density Networks. *Statistics and Neural Networks: Advances at the Interface*, pages 129–144, 1999.

[36] D. J. C. Mackay. *Bayesian Methods for Adaptive Models*. PhD thesis, California Institute of Technology, USA, 1992. UMI Order No. GAX92-32200.

[37] D. J. C. Mackay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, first edition edition, June 2003.

[38] I. Murray and Z. Ghahramani. A Note on the Evidence and Bayesian Occam's Razor. Technical Report GCNU-TR 2005-003, Gatsby Computational Neuroscience Unit, University College London, 2005.

[39] E. T. Nalisnick. *On Priors for Bayesian Neural Networks*. PhD thesis, University of California, Irvine, USA, 2018.

[40] E. T. Nalisnick, J. Gordon, and J. M. Hernández-Lobato. Predictive Complexity Priors. In *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event*, volume 130 of *Proceedings of Machine Learning Research*, pages 694–702. PMLR, 2021.

[41] E. T. Nalisnick and P. Smyth. Stick-Breaking Variational Autoencoders. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[42] R. M. Neal. *MCMC Using Hamiltonian Dynamics*, chapter 5. CRC Press, 2011.

[43] X. Nguyen, M. J. Wainwright, and M. I. Jordan. Estimating Divergence Functionals and the Likelihood Ratio by Convex Risk Minimization. *IEEE Transactions on Information Theory*, 56(11):5847–5861, 2010.

[44] C. E. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[45] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014*, volume 32 of *Proceeding of Machine Learning Research*, pages 1278–1286, Beijing, China, 21-26 June 2014. PMLR.

[46] S. Sun, G. Zhang, J. Shi, and R. Grosse. Functional Variational Bayesian Neural Networks. In *International Conference on Learning Representations*, 2019.

[47] I. Tolstikhin, O. Bousquet, S. Gelly, and B. Schoelkopf. Wasserstein Auto-Encoders. In *International Conference on Learning Representations*, 2018.

[48] J. M. Tomczak and M. Welling. VAE with a VampPrior. In *International Conference on Artificial Intelligence and Statistics, AISTATS 2018, 9-11 April 2018, Playa Blanca, Lanzarote, Canary Islands, Spain*, volume 84 of *Proceedings of Machine Learning Research*, pages 1214–1223. PMLR, 2018.

[49] B.-H. Tran, S. Rossi, D. Milios, and M. Filippone. All You Need is a Good Functional Prior for Bayesian Deep Learning. *arXiv preprint arXiv:2011.12829*, 2020.

[50] R. van den Berg, L. Hasenclever, J. M. Tomczak, and M. Welling. Sylvester Normalizing Flows for Variational Inference. In *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence, UAI 2018, Monterey, California, USA, August 6-10, 2018*, pages 393–402. AUAI Press, 2018.

[51] C. Villani. *Optimal Transport: Old and New*, volume 338. Springer Science & Business Media, 2008.

[52] F. Wenzel, K. Roth, B. S. Veeling, J. Świątkowski, L. Tran, S. Mandt, J. Snoek, T. Salimans, R. Jenatton, and S. Nowozin. How Good is the Bayes Posterior in Deep Neural Networks Really? In *Proceeding of the 37th International Conference on Machine Learning, ICML 2020*, Virtual, 25-30 June 2020.

[53] A. G. Wilson and P. Izmailov. Bayesian Deep Learning and a Probabilistic Perspective of Generalization. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[54] W. Yang, L. Lorch, M. A. Graule, H. Lakkaraju, and F. Doshi-Velez. Incorporating Interpretable Output Constraints in Bayesian Neural Networks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[55] C. Zeno, I. Golan, A. Pakman, and D. Soudry. Why Cold Posteriors? On the Suboptimal Generalization of Optimal Bayes Estimates. In *Third Symposium on Advances in Approximate Bayesian Inference*, 2021.

[56] G. Zhang, S. Sun, D. Duvenaud, and R. B. Grosse. Noisy Natural Gradient as Variational Inference. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 5847–5856. PMLR, 2018.

[57] R. Zhang, C. Li, J. Zhang, C. Chen, and A. G. Wilson. Cyclical Stochastic Gradient MCMC for Bayesian Deep Learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[58] S. Zhao, Z. Liu, J. Lin, J. Zhu, and S. Han. Differentiable Augmentation for Data-Efficient GAN Training. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] Discussed in the Conclusions.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] Discussed in the Conclusions.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Submitted as supplementary material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Few comments in the Section 4 of the main paper and full descriptions in Appendix F.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] We report error bars over 4 seeds; the analysis has been too computationally expensive to use more seeds.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Few comments in the Section 4 of the main paper and full descriptions in Appendix F.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Appendix F.
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] We used publicly available datasets.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] We used publicly available datasets.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]