SELF-REFLECTIVE VARIATIONAL AUTOENCODER

Anonymous authors
Paper under double-blind review

ABSTRACT

The Variational Autoencoder (VAE) is a powerful framework for learning probabilistic latent variable generative models. However, typical assumptions on the approximate posterior distributions can substantially restrict its capacity for inference and generative modeling. Variational inference based on neural autoregressive models respects the conditional dependencies of the exact posterior, but this flexibility comes at a cost: the resulting models are expensive to train in high-dimensional regimes and can be slow to produce samples. In this work, we introduce an orthogonal solution, which we call *self-reflective inference*. By redesigning the hierarchical structure of existing VAE architectures, self-reflection ensures that the stochastic flow preserves the factorization of the exact posterior, sequentially updating the latent codes in a manner consistent with the generative model. We empirically demonstrate the advantages of matching the variational posterior to the exact posterior—on binarized MNIST self-reflective inference achieves state-of-the-art performance without resorting to complex, computationally expensive components such as autoregressive layers. Moreover, we design a variational normalizing flow that employs the proposed architecture, yielding predictive benefits compared to its purely generative counterpart. Our proposed modification is quite general and it complements the existing literature; self-reflective inference can naturally leverage advances in distribution estimation and generative modeling to improve the capacity of each layer in the hierarchy.

1 INTRODUCTION

The advent of deep learning has led to great strides in both supervised and unsupervised learning. One of the most popular recent frameworks for the latter is the Variational Autoencoder (VAE), in which a probabilistic encoder and generator are jointly trained via backpropagation to simultaneously perform sampling and variational inference. Since the introduction of the VAE (Kingma & Welling, 2014), or more generally, the development of techniques for low-variance stochastic backpropagation of Deep Latent Gaussian Models (DLGMs) (Rezende et al., 2014) (see Figure 1), research has rapidly progressed towards improving their generative modeling capacity and/or the quality of their variational approximation. However, as deeper and more complex architectures are introduced, care must be taken to ensure the correctness of various modeling assumptions, whether explicit or implicit. In particular, when working with hierarchical models it is easy to unintentionally introduce mismatches in the generative and inference models, to the detriment of both. In this work, we demonstrate the existence of such a modeling pitfall common to much of the recent literature on DLGMs. We discuss why this problem emerges, and we introduce a simple—yet crucial—modification to the existing architectures to address the issue.

Vanilla VAE architectures make strong assumptions about the posterior distribution—specifically, it is standard to assume that the posterior is approximately factorial. More recent research has investigated the effect of such assumptions which govern the variational posterior (Wenzel et al., 2020) or prior (Wilson & Izmailov, 2020) in the context of uncertainty estimation in Bayesian neural networks. In many scenarios, these restrictions have been found to be problematic. A large body of recent work attempts to improve performance by building a more complex encoder and/or decoder with convolutional layers and more modern architectures (such as ResNets (He et al., 2016)) (Salimans et al., 2015; Gulrajani et al., 2017) or by employing more complex posterior distributions constructed with autoregressive layers (Kingma et al., 2016; Chen et al., 2017). Other work (Tomczak & Welling, 2018; Klushyn et al., 2019a) focuses on refining the prior distribution of the latent

codes. Taking a different approach, hierarchical VAEs (Rezende et al., 2014; Gulrajani et al., 2017; Sønderby et al., 2016; Maaløe et al., 2019; Klushyn et al., 2019b) leverage increasingly deep and interdependent layers of latent variables, similar to how subsequent layers in a discriminative network are believed to learn more and more abstract representations. These architectures exhibit superior generative and reconstructive capabilities since they allow for modeling of much richer latent spaces. While the benefits of incorporating hierarchical latent variables is clear, all existing architectures suffer from a modeling mismatch which results in sub-optimal performance: *the variational posterior does not respect the factorization of the exact posterior distribution of the generative model*.

In earlier works on hierarchical VAEs (Rezende et al., 2014), inference proceeds bottom-up, counter to the top-down generative process. To better match the order of dependence of latent variables to that of the generative model, later works (Sønderby et al., 2016; Bachman, 2016) split inference into two stages: first a deterministic bottom-up pass which does necessary precomputation for evidence encoding, followed by a stochastic top-down pass which incorporates the hierarchical latents to form a closer variational approximation to the exact posterior. Crucially, while these newer architectures ensure that the order of the latent variables mirrors that of the generative model, the overall variational posterior does not match because of the strong restrictions on the variational distributions of each layer.

Contributions. In this work, we propose to restructure common hierarchical VAE architectures with a series of bijective layers which enable communication between the inference and generative networks, refining the latent representations. Concretely, our contributions are as follows:

- We motivate and introduce a straightforward *rearrangement of the stochastic flow of the model* which addresses the aforementioned modeling mismatch. This modification substantially compensates for the observed performance gap between models with only simple layers and those with complex autoregressive networks (Kingma et al., 2016; Chen et al., 2017).
- We formally prove that this refinement results in **a hierarchical VAE whose variational posterior respects the precise factorization of the exact posterior.** To the best of our knowledge, this is the first deep architecture to do so without resorting to computationally expensive autoregressive components or making strong assumptions (e.g., diagonal Gaussian) on the distributions of each layer (Sønderby et al., 2016)—assumptions that lead to degraded performance.
- We experimentally demonstrate the benefits of the improved representation capacity of this model, which stems from the corrected factorial form of the posterior. We achieve state-of-the-art perfomance on MNIST among models without autoregressive layers, and our model performs on par with recent, fully autoregressive models such as Kingma et al. (2016). Due to the simplicity of our architecture, we achieve these results for a fraction of the computational cost in both training and inference.
- We design a *hierarchical variational normalizing flow* that deploys the suggested architecture in order to recursively update the base distribution and the conditional bijective transformations. This architecture significantly improves upon the predictive performance and data complexity of a Masked Autoregressive Flow (MAF) (Papamakarios et al., 2017) on CIFAR-10.

Finally, it should be noted that our contribution is quite general and can naturally leverage recent advances in variational inference and deep autoencoders (Chen et al., 2017; Kingma et al., 2016; Tomczak & Welling, 2018; Burda et al., 2016; Dai & Wipf, 2019; van den Oord et al., 2016a; Rezende & Viola, 2018) as well as architectural improvements to density estimation (Gulrajani et al., 2017; Dinh et al., 2017; Kingma & Dhariwal, 2018; Durkan et al., 2019; van den Oord et al., 2016b; Gregor et al., 2015). We suspect that combining our model with other state-of-the-art methods could further improve the attained performance, which we leave to future work.

2 Variational Autonencoders

A Variational Autoencoder (VAE) (Kingma & Welling, 2014; 2019) is a generative model which is capable of generating samples $\boldsymbol{x} \in \mathbb{R}^D$ from a distribution of interest $p(\boldsymbol{x})$ by utilizing latent variables $\boldsymbol{z}$ coming from a prior distribution $p(\boldsymbol{z})$. To perform inference, the marginal likelihood should be computed which involves integrating out the latent variables:

$$p(\boldsymbol{x}) = \int p(\boldsymbol{x}, \boldsymbol{z})\, d\boldsymbol{z}. \tag{1}$$

In general, this integration will be intractable and a lower bound on the marginal likelihood is maximized instead. This is done by introducing an approximate posterior distribution $q(\boldsymbol{z} \mid \boldsymbol{x})$ and applying Jensen's inequality:

$$\log p(\boldsymbol{x}) = \log \int p(\boldsymbol{x}, \boldsymbol{z})\, d\boldsymbol{z} = \log \int \frac{q(\boldsymbol{z} \mid \boldsymbol{x})}{q(\boldsymbol{z} \mid \boldsymbol{x})} p(\boldsymbol{x}, \boldsymbol{z})\, d\boldsymbol{z} \geq \int q(\boldsymbol{z} \mid \boldsymbol{x}) \log \left[\frac{p(\boldsymbol{x} \mid \boldsymbol{z}) p(\boldsymbol{z})}{q(\boldsymbol{z} \mid \boldsymbol{x})} \right] d\boldsymbol{z}$$
$$\implies \log p(\boldsymbol{x}) \geq \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}[\log p(\boldsymbol{x} \mid \boldsymbol{z})] - D_{KL}(q(\boldsymbol{z} \mid \boldsymbol{x}) \parallel p(\boldsymbol{z})) \triangleq \mathcal{L}(\boldsymbol{x}; \boldsymbol{\theta}, \boldsymbol{\phi}), \tag{2}$$

where $\boldsymbol{\theta}$, $\boldsymbol{\phi}$ parameterize $p(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{\theta})$ and $q(\boldsymbol{z} \mid \boldsymbol{x}; \boldsymbol{\phi})$ respectively. For ease of notation, we may omit $\boldsymbol{\theta}, \boldsymbol{\phi}$ in the derivations. This objective is called the Evidence Lower BOund (ELBO) and can be optimized efficiently for continuous $\boldsymbol{z}$ via stochastic gradient descent (Kingma & Welling, 2014; Rezende et al., 2014).

3 Self-Reflective Variational Inference

With this background, we are now ready to introduce our main contribution: the first deep probabilistic model which ensures that the variational posterior matches the factorization of the exact posterior induced by its generative model. We refer to this architecture as the Self-Reflective Variational Autoencoder (SeRe-VAE). Figure 2 displays the overall proposed architecture. We expound upon its components in the following subsections.

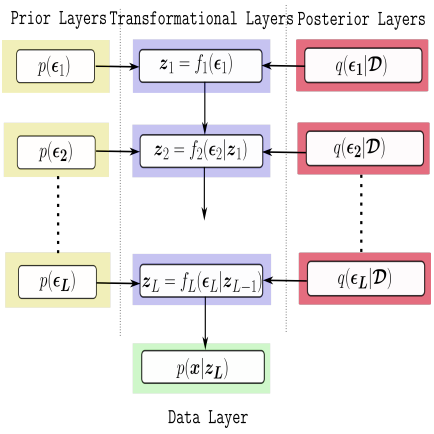

Figure 1: Independent VAE.

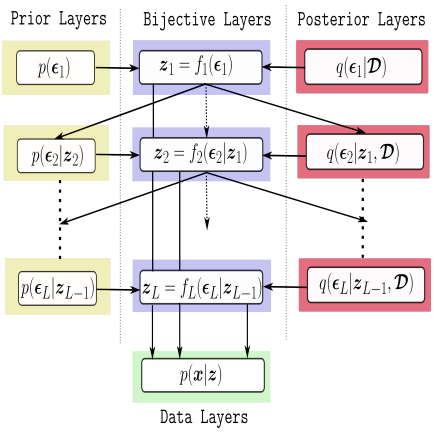

Figure 2: Self-Reflective VAE.

3.1 Generative Model

Figure 3 displays the overall stochastic flow of the generative network. Our generative model consists of a hierarchy of L stochastic layers, as in Rezende et al. (2014) (see Figure 1). However, in this work, the data $\boldsymbol{x} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_L) \in \mathbb{R}^D$ is partitioned into L blocks, with each layer generating only $\boldsymbol{x}_l \in \mathbb{R}^{D_l}$, with $\sum_l D_l = D$. At each layer l, N_l-dimensional latent variables $\boldsymbol{\epsilon}_l \in \mathbb{R}^{N_l}$ are first sampled from a simple prior distribution (`Prior Layer`) and subsequently transformed to latent variables $\boldsymbol{z}_l \in \mathbb{R}^{N_l}$ by a bijective function $f_l : \mathbb{R}^{N_l} \to \mathbb{R}^{N_l}$ (`Bijective Layer`). To distinguish between the two sets of latent variables in our model, throughout this paper we refer to $\boldsymbol{\epsilon}_l$ as the *base latent variables* and $\boldsymbol{z}_l$ as the *latent codes*. The latent codes $\boldsymbol{z}_l$ are subsequently passed to the stochastic layer responsible for generating the observed data (`Data Layer`). Moreover, the layers in the hierarchy are connected in three ways: i) the prior layer l can access the latent codes $\boldsymbol{z}_{l-1}$ ii) $\boldsymbol{z}_{l-1}$ is fed to the next bijection f_l iii) the data layer l receives the data block $\boldsymbol{x}_{l-1}$ generated by the previous data layer. Intuitively, this choice is justified because the latent codes $\boldsymbol{z}_l$ of layer l, conditioned on $\boldsymbol{z}_{l-1}$, will be successively refined based on how well $\boldsymbol{z}_{l-1}$ reconstructed $\boldsymbol{x}_{l-1}$, yielding progressively more meaningful latent representations. In the following subsections, we describe these steps in detail.

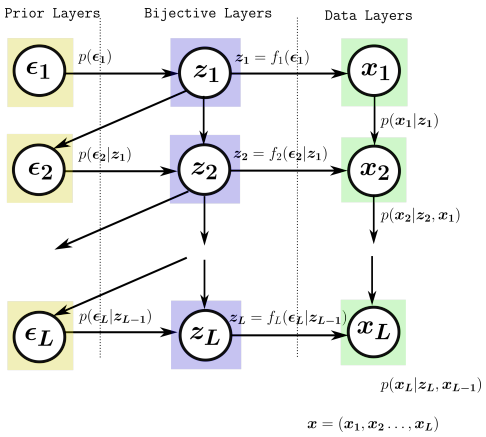

Figure 3: Self-Reflective Generative Model.

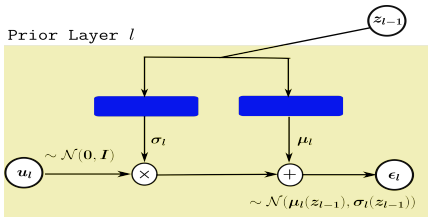

Figure 4: Amortized Gaussian Prior Layer.

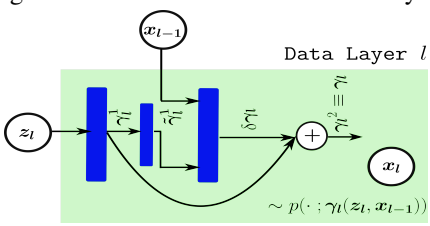

Figure 5: Residual Data Layer.

3.1.1 AMORTIZED PRIOR LAYERS

As already mentioned, all but the first prior distributions are conditioned on earlier latent factors yielding conditional prior distributions $p(\boldsymbol{\epsilon}_l|\boldsymbol{z}_{l-1}; \boldsymbol{\alpha}_l)$. We achieve this by modeling the parameters of the prior distribution $\boldsymbol{\alpha}_l$ as functions of previous latent codes: $\boldsymbol{\alpha}_l \triangleq \boldsymbol{\alpha}_l(\boldsymbol{z}_{l-1}; \boldsymbol{c}_l^\alpha)$, which are in turn parametrized by $\boldsymbol{c}_l^\alpha$. $\boldsymbol{c}_l^\alpha$ is constant and common for all the data points. Figure 4 shows the computational graph of a Gaussian prior layer according to the reparametrization trick (Kingma & Welling, 2014). In this case, the parameters $\boldsymbol{\alpha}_l = (\boldsymbol{\mu}_l, \boldsymbol{\sigma}_l)$ are the mean and standard deviation of the Gaussian distribution (note they are different for each observation). The parameters are computed by one of two modules, Residual Blocks (ResNet) or MultiLayer Perceptrons (MLP) (depicted as blue rectangles in Figure 4), that receive as input the latent codes $\boldsymbol{z}_{l-1}$ and use weights $\boldsymbol{c}_l^\alpha$. In the rest of the paper, we use the notations $p(\boldsymbol{\epsilon}_l|\boldsymbol{z}_{l-1}; \boldsymbol{\alpha}_l)$ and $p(\boldsymbol{\epsilon}_l; \boldsymbol{\alpha}_l(\boldsymbol{z}_{l-1}))$ interchangeably.

3.1.2 AMORTIZED BIJECTIVE LAYERS

In order to increase the representation capacity of the base latent variables $\boldsymbol{\epsilon}_l$, they are transformed by a deterministic bijective function f_l parametrized by $\boldsymbol{\beta}_l$. Once again, we model these parameters as a function—which is in turn parametrized by $\boldsymbol{c}_l^\beta$—of the previous layer's latent codes: $\boldsymbol{\beta}_l \triangleq \boldsymbol{\beta}_l(\boldsymbol{z}_{l-1}; \boldsymbol{c}_l^\beta)$. $\boldsymbol{c}_l^\beta$ is constant and common for all data. For an affine transformation $f_l : \mathbb{R}^{N_l} \to \mathbb{R}^{N_l}$, we model $\boldsymbol{z}_l = \boldsymbol{c}_l + (diag(\boldsymbol{d}_l) + \boldsymbol{u}_l \boldsymbol{u}_l^T) \times \boldsymbol{\epsilon}_l$. Here, $\boldsymbol{c}_l, \boldsymbol{u}_l, \boldsymbol{d}_l \in \mathbb{R}^{N_l}$ are functions of $\boldsymbol{z}_{l-1}$ implemented by ResNet or MLP computational blocks (with weights $\boldsymbol{c}_l^\beta$), $\boldsymbol{\beta}_l = (\boldsymbol{c}_l, \boldsymbol{d}_l, \boldsymbol{u}_l)$, and $\boldsymbol{d}_l \geq 0$ to ensure bijectivity.

3.1.3 RESIDUAL DATA LAYERS

Each data block $\boldsymbol{x}_l \in \mathbb{R}^{D_l}$ is generated by a random variable drawn from a suitable distribution parametrized by $\boldsymbol{\gamma}_l$ and conditioned on i) the latent codes $\boldsymbol{z}_l$ and ii) the data block $\boldsymbol{x}_{l-1}$ generated by the previous layer: $\boldsymbol{x}_l \sim p(\cdot|\boldsymbol{z}_l, \boldsymbol{x}_{l-1}; \boldsymbol{\gamma}_l)$. The conditional distribution is built, as before, by rendering the parameters $\boldsymbol{\gamma}_l$ as functions of $\boldsymbol{z}_l, \boldsymbol{x}_{l-1}$ which are in turn parametrized by a constant, common to all the observations, $\boldsymbol{c}_l^\gamma$ such that $\boldsymbol{\gamma}_l \triangleq \boldsymbol{\gamma}_l(\boldsymbol{z}_l, \boldsymbol{x}_{l-1}; \boldsymbol{c}_l^\gamma)$. In order to account for these two conditioning factors, the parameters $\boldsymbol{\gamma}_l$ are estimated in a two-step fashion with a *residual distributional layer* shown in Figure 5. The first estimation of $\boldsymbol{\gamma}_l$, $\boldsymbol{\gamma}_l^1$, is obtained solely from the latent codes $\boldsymbol{z}_l$. At the second step of the estimation, the evidence $\boldsymbol{x}_{l-1}$ is taken into account, by computing a residual function $\delta\boldsymbol{\gamma}_l$, that rectifies the first estimation in an additive manner: $\boldsymbol{\gamma}_l = \boldsymbol{\gamma}_l^1(\boldsymbol{z}_l) + \delta\boldsymbol{\gamma}_l(\boldsymbol{x}_{l-1}, \boldsymbol{z}_l)$. Finally, the weights of the three computational blocks (blue rectangles) in Figure 5 define $\boldsymbol{c}_l^\gamma$.

The joint distribution of the latent codes $\boldsymbol{z} = (\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_L)$ and the observed data $\boldsymbol{x}$ of the generative model is:

$$p(\boldsymbol{x}, \boldsymbol{z}) = p(\boldsymbol{z}_1) \times p(\boldsymbol{x}_1|\boldsymbol{z}_1) \times \prod_{l=2}^{L} p(\boldsymbol{z}_l \mid \boldsymbol{z}_{l-1}) \times p(\boldsymbol{x}_l \mid \boldsymbol{z}_l, \boldsymbol{x}_{l-1}). \tag{3}$$

We denote the parameters of the generative model as $\boldsymbol{\theta} \triangleq \{\boldsymbol{c}^{\alpha}_{1:L}, \boldsymbol{c}^{\beta}_{1:L}, \boldsymbol{c}^{\gamma}_{1:L}\}$.

3.2 Inference Model

The inference network is identical to the generative network shown in Figure 3, except that the prior layers are replaced by the posterior layers for the generation of the base latent variables $\boldsymbol{\epsilon}_l$. Specifically, the variational encoder of the SeRe-VAE is defined as follows:

$$q(\boldsymbol{\epsilon}_1, \boldsymbol{\epsilon}_2, \ldots, \boldsymbol{\epsilon}_L \mid \boldsymbol{\mathcal{D}}) = q(\boldsymbol{\epsilon}_1 \mid \boldsymbol{\mathcal{D}}) \times \prod_{l=2}^{L} q(\boldsymbol{\epsilon}_l \mid \boldsymbol{z}_{l-1}, \boldsymbol{\mathcal{D}}). \tag{4}$$

Compared to other hierarchical architectures, in the proposed model the inference layers are conditioned on the output of the preceding bijective layer — these components are shared between the generative and the inference network (see Figure 2). In this work, we assume Gaussian diagonal base distributions $q(\boldsymbol{\epsilon}_l \mid \boldsymbol{z}_{l-1}, \boldsymbol{\mathcal{D}})$ and we deploy a residual parametrization, similar to the one presented in Figure 5 for the data layer. Note that the posterior distribution is also receiving two conditioning factors: the evidence $\boldsymbol{\mathcal{D}}$ and the latent codes $\boldsymbol{z}_{l-1}$. The first estimation of the mean and standard deviation is obtained solely from $\boldsymbol{z}_{l-1}$; in light of the evidence $\boldsymbol{\mathcal{D}}$, this estimation is rectified by a residual term. The special case of a 2-level residual Gaussian layer along with the network for the data preprocessing is provided in the appendix.

3.3 Exact Bayes Propagation

In this section, we provide the formal justification for the choice of equation 4; we prove that back-propagation of our model preserves the factorization of the true posterior, without resorting to complex graph inversion as in Webb et al. (2018). We use the following straightforward lemma:

Lemma 1 *Let $f : \mathbb{R}^N \to \mathbb{R}^N$ be an invertible transformation such that both f and f^{-1} are differentiable everywhere. Then for any $\boldsymbol{z} \in \mathbb{R}^N$, $p(\boldsymbol{\epsilon}|\boldsymbol{z}) = p(\boldsymbol{\epsilon}|f(\boldsymbol{z}))$.*

Proof: By Bayes's Theorem and the change of variables formula (Rudin, 2006; Bogachev, 2007),

$$p(\boldsymbol{\epsilon}|f(\boldsymbol{z})) = \frac{p(f(\boldsymbol{z})|\boldsymbol{\epsilon}) \ \times \ p(\boldsymbol{\epsilon})}{p(f(\boldsymbol{z}))} = \frac{p(\boldsymbol{z}|\boldsymbol{\epsilon}) \times \mid \det J_f(\boldsymbol{z}) \mid^{-1} \ \times p(\boldsymbol{\epsilon})}{p(\boldsymbol{z}) \times \mid \det J_f(\boldsymbol{z}) \mid^{-1}} = p(\boldsymbol{\epsilon}|\boldsymbol{z}),$$

where $J_f(\boldsymbol{z})$ is the Jacobian matrix of f evaluated at $\boldsymbol{z}$, which has non-zero determinant by assumption. □

We now present our main theoretical result, which says that the factorization of our model's variational posterior exactly matches that of the generative distribution.

Proposition 1 *The factorization of the variational posterior defined in equation 4 respects the factorization of the exact posterior distribution induced by the generative model in equation 3.*

Proof: Let $p(\boldsymbol{\epsilon}_1, \boldsymbol{\epsilon}_2, \ldots, \boldsymbol{\epsilon}_L \mid \boldsymbol{\mathcal{D}})$ be the posterior distribution induced by the generative model defined in equation 3, as illustrated in Figure 3. Then, according to the probability product rule the posterior distribution can be expressed as:

$$p(\boldsymbol{\epsilon}_1, \boldsymbol{\epsilon}_2, \ldots, \boldsymbol{\epsilon}_L \mid \boldsymbol{\mathcal{D}}) = p(\boldsymbol{\epsilon}_1 \mid \boldsymbol{\mathcal{D}}) \times \prod_{l=2}^{L} p(\boldsymbol{\epsilon}_l \mid \boldsymbol{\epsilon}_{<l}, \boldsymbol{\mathcal{D}}), \tag{5}$$

where $\boldsymbol{\epsilon}_{<l} \triangleq \{\boldsymbol{\epsilon}_1, \boldsymbol{\epsilon}_2, \ldots, \boldsymbol{\epsilon}_{l-1}\}$. We will apply the *Bayes ball* rule (Jordan, 2003) to simplify equation 5. Consider an arbitrary layer l of the hierarchy. Because f_{l-1} is a bijector, by Lemma 1 we have

$$p(\boldsymbol{\epsilon}_l \mid \boldsymbol{\epsilon}_{<l}, \boldsymbol{\mathcal{D}}) = p(\boldsymbol{\epsilon}_l \mid \boldsymbol{\epsilon}_{l-1}, \boldsymbol{\epsilon}_{<l-1}, \boldsymbol{\mathcal{D}}) = p(\boldsymbol{\epsilon}_l \mid \boldsymbol{z}_{l-1}, \boldsymbol{\epsilon}_{<l-1}, \boldsymbol{\mathcal{D}}).$$

Now, note that $\boldsymbol{\epsilon}_l$ is *D-separated* from $\boldsymbol{\epsilon}_{l-1}, \ldots, \boldsymbol{\epsilon}_1$ since all paths from $\boldsymbol{\epsilon}_l$ to $\boldsymbol{\epsilon}_{<l}$ pass through the observed nodes $\boldsymbol{z}_{l-1}$ or $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{l-1}$ (see Figure 6 for an example). Therefore, we have

$$p(\boldsymbol{\epsilon}_l \mid \boldsymbol{z}_{l-1}, \boldsymbol{\epsilon}_{<l-1}, \boldsymbol{\mathcal{D}}) = p(\boldsymbol{\epsilon}_l \mid \boldsymbol{z}_{l-1}, \boldsymbol{\mathcal{D}}). \tag{6}$$

Since this applies to every layer, it follows that the exact posterior equation 5 can also be expressed as

$$p(\boldsymbol{\epsilon}_1, \boldsymbol{\epsilon}_2, \ldots, \boldsymbol{\epsilon}_L \mid \boldsymbol{\mathcal{D}}) = p(\boldsymbol{\epsilon}_1 \mid \boldsymbol{\mathcal{D}}) \times \prod_{l=2}^{L} p(\boldsymbol{\epsilon}_l \mid \boldsymbol{z}_{l-1}, \boldsymbol{\mathcal{D}}), \tag{7}$$

exactly matching the factorization of the approximate posterior in equation 4. □

3.4 General Remarks

In contrast to Rezende et al. (2014), in our model i) the prior layers are not independent, but rather are conditioned on the previous layers in the hierarchy; and ii) the transformational layers are restricted to be bijective. The proposed model also differs from other hierarchical architectures (Gulrajani et al., 2017; Sønderby et al., 2016; Maaløe et al., 2019); in these models the layers of the prior are conditioned upon the previous prior layers and not upon bijective layers that are shared between the generative and inference model. One additional key difference between our model and all previous work is the coupling between data layers. In Section 4, we provide empirical results demonstrating the benefits of these modeling choices. Following the above analysis, we make some observations about the hierarchy of shared bijective layers in the model:

- It allows for complex transformations of the latent variables via the bijective functions f_l.
- It resembles the precision-weighted combination of the generative and inference parts suggested in Sønderby et al. (2016), where in our case the "averaging" is applied through the shared bijective layers which can express distributions more complex than diagonal Gaussian.
- By reducing the set of conditioning variables from $\boldsymbol{\epsilon}_{<l}$ to $\boldsymbol{z}_{l-1}$, the hierarchical bijective layers offer a convenient way to precisely and efficiently factorize the variational distribution, alleviating the bottleneck present in high-dimensional autoregressive approaches.
- The model, albeit hierarchical, is less prone to posterior collapse, since each layer is responsible for the generation of a different portion of the data.
- Finally, the use of these layers can be viewed as a hierarchical application of the *reparameterization trick* (Kingma & Welling, 2014) which is now conducive to a closed-form computation of the KL-divergence since $D_{KL}(q(\boldsymbol{z} \mid \boldsymbol{x}) \parallel p(\boldsymbol{z})) = D_{KL}(q(\boldsymbol{\epsilon} \mid \boldsymbol{x}) \parallel p(\boldsymbol{\epsilon}))$ (due to bijectivity of f_l). The prior and posterior layers can be Gaussian distributions, obviating the need for expensive Monte Carlo approximations of the KL divergence and increasing the stability of the training process, while the final latent representations of the encoder can be arbitrarily complex.

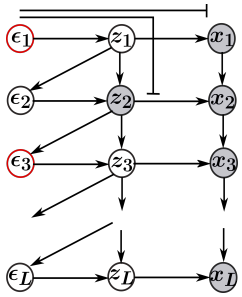

Figure 6: *D-separation between stochastic layers.* By the Bayes ball rule, all paths from $\boldsymbol{\epsilon}_1$ to $\boldsymbol{\epsilon}_3$ pass either through $\boldsymbol{x}_1$ or $\boldsymbol{z}_2$, which D-separate them. Therefore, $\boldsymbol{\epsilon}_1 \perp\!\!\!\perp \boldsymbol{\epsilon}_3 | \boldsymbol{z}_2, \boldsymbol{\mathcal{D}}$.

4 Experimnetal Studies

4.1 Dynamically binarized MNIST

We empirically evaluate the SeRe-VAE on dynamically binarized MNIST. As in Burda et al. (2016); Sønderby et al. (2016); Kingma et al. (2016), the binary-valued observations are sampled after each epoch with the Bernoulli expectations being set equal to the real, normalized pixel values in the dataset which prevents overfitting.

4.1.1 PERFORMANCE OF THE MLP SERE-VAE

To demonstrate that our model's improved performance is due to the restructuring of the stochastic flow and not sophisticated layers, we use simple multilayer-perceptron (MLP) components; we similarly forgo importance weighting (Burda et al., 2016). We adopt a 10-layer architecture, with $N_l = 10$ latent variables per layer, for a total of 100 latent features being passed to the decoder after being transformed by an affine bijector as described in Section 3.1.2. We partition the image into $L = 10$ equally sized blocks (except for the last one) from left to right in a raster fashion. Finally, we use independent deterministic encoders for the data preprocessing. The full details of our implementation are delegated to the supplementary material. We again emphasize the overall simplicity of our architecture, choosing instead to focus on the benefits of the corrected posterior factorization. As shown in Table 1, our model (SeRe-VAE) outperforms existing models of the same complexity

Table 1: Dynamically binarized MNIST Performance for VAEs without ResNet layers. 1000 importance samples were used for the estimation of the marginal likelihood. For the Ladder VAE performance, we refer to Table1 in Sønderby et al. (2016). The models were trained with a single importance sample unless otherwise noted (IW=1).

Model	Details	$log\ p(x) \geq$
Self-Reflective	10 layers / 10 variables each, diagonal Gaussian prior	$\mathbf{-81.17}$
Importance Weighted Ladder	5 layers / 128 variables total, #IW samples=10	-81.74
Ladder	5 layers / 128 variables total	-81.84
Self-Reflective IAF	10 layers / 10 variables each, Standard Normal Prior	-81.96
Inverse Autoregressive Flow	1 layer / 100 variables, Standard Normal Prior	-83.04
Deep Latent Gaussian Model	10 layers / 10 variables each, diagonal Gaussian prior	-84.53
Relaxed Bernoulli VAEs	30 latent variables, exact factorization	-90

such as the DLGM and Ladder VAE (LVAE), those of higher complexity such as Inverse Autoregressive Flow (IAF), and models trained with importance weighted samples (IW-LVAE). Note that the architecture of the DLGM is identical to that of SeRe-VAE; to ensure a fair comparison, the DLGM was given larger feature maps in the encoders to compensate for the additional bijective layer inputs in the SeRe-VAE. Therefore, *the performance benefits are solely attributed to the inclusion of the latent codes in subsequent stochastic layers in the hierarchy* (compare Figures 1 and 2). Our model outperforms the LVAE models, despite using a smaller latent dimensionality (128 vs. 100) and being trained with a single importance sample. Moreover, our model exhibits superior performance compared to the autoregressive IAF; this discrepancy could stem from the 1-layer architecture or the fact that a standard normal prior was used. This result indicates that a prior of equivalent expressive capacity communicating with the bijective layer could yield additional improvement. Finally, in our experiments the 10-layer IAF took nearly *twice as long* to train compared to the SeRe-VAE. Finally, the Relaxed Bernoulli VAE (Webb et al., 2018) respects the factorization of the true posterior but scales up to 30 latent variables while not supporting recurrent refinement across layers. The learning curves, the architectural details and the training hyperparameters are provided in the appendix.

4.1.2 PERFORMANCE OF THE RESNET SERE-VAE

To demonstrate the capacity of our model when combined with complex layers, we replaced the MLPs with ResNets as in Salimans et al. (2015) while preserving the same number of latent variables. As shown in Table 2, our model performs better than all recent models that do not use expensive coupling or autoregressive layers and on par with models of higher complexity. Especially for BIVA, it should be mentioned that more, 168 vs 100 of our model, latent variables are used. The full architectural details are provided in the appendix.

4.2 PERFORMANCE OF A SELF-REFLECTIVE, VARIATIONAL MASKED AUTOREGRESSIVE FLOW ON CIFAR-10

In this section, we introduce a *hierarchical latent variable normalizing flow*: the first VAE with a decoder (`Data Layer`) consisting of normalizing flow transformations—realizing improvements

Table 2: Dynamically binarized MNIST performance for VAEs with sophisticated layers. 1000 importance samples were used for the estimation of the marginal likelihood. All performances listed here are taken from Maaløe et al. (2019) and Durkan et al. (2019). All models were trained with a single importance sample.

Model	$log\ p(x) \geq$
Models with autoregressive (AR) or coupling (C) components	
VLAE (Chen et al., 2017)	-79.03
Pixel RNN (van den Oord et al., 2016b)	-79.20
RQ-NSF (C) (Durkan et al., 2019)	-79.63
Pixel VAE (Gulrajani et al., 2017)	-79.66
RQ-NSF (AR) (Durkan et al., 2019)	-79.71
IAF VAE (Kingma et al., 2016)	-79.88
DRAW (Gregor et al., 2015)	-80.97
Pixel CNN (van den Oord et al., 2016a)	-81.30
Models without autoregressive or coupling components	
SeRe-VAE	$\mathbf{-79.50}$
BIVA (Maaløe et al., 2019)	-80.47
Discrete VAE (Rolfe, 2017)	-81.01

over its purely generative counterpart. Due to space constraints we refer the reader to the appendix for a review of normalizing flows, as well as the full technical details of our architecture. A high-level description is provided here. The latent variables are generated by the proposed network shown in Figure 3. Subsequently, the latent variables $\boldsymbol{z}$ are incorporated in the flow in two ways: i) conditioning the base distribution and ii) conditioning the bijective transformations. In the case of a Masked Autoregressive Flow (MAF) (Papamakarios et al., 2017) or an Inverse Autoregressive Flow (Kingma et al., 2016), the latter amounts to designing *conditional* MADE layers (Germain et al., 2015) that account for a mask offset so that the additional inputs $\boldsymbol{z}$ are not masked out. The first amounts to building an amortized Gaussian layer (see Figure 4). We used a 5 layer hierarchy of 40 latent variables each. We adopted a unit rank Gaussian base distribution in the decoder—parameterized as in Equation (9) in Rezende et al. (2014)—and diagonal Gaussian prior and posterior layers. We used neural spline bijective layers with coupling transformations (Durkan et al., 2019), which boosted the performance compared to affine transformations. We refer to our source code and the supplementary material for the implementation details. In Table 3, we compare against generative MAF models with the same or larger width, with or without training dataset augmentation with horizontal image flips and different number of MADEs. Our variational model exhibits significant improvement over the baselines.

Table 3: Performance of different MAFs on CIFAR-10.

Model	Variational	#MADE layers	Width	Flipped Images	Test Loglikelihood
SeRe-MAF	**Yes**	**10 (2 flows, 5 layers)**	**1024**	**No**	$\geq$ **3190 (ELBO)**
MAF	No	10	1024	No	2670
MAF (5) (Papamakarios et al., 2017)	No	5	2048	Yes	2936
MAF (10) (Papamakarios et al., 2017)	No	10	2048	Yes	3049

5 Conclusion and Discussion

In this paper, we presented self-reflective variational inference (SeRe-VAE), a structural modification for hierarchical VAEs that combines top-down inference and iterative feedback between the generative and inference network through shared bijective layers. This modification increases the representation capacity of existing VAEs, leading to smaller latent spaces and vast computational benefits without compromising the generative capacity of the model. We further introduced hierarchical latent variable normalizing flows which utilize the proposed architecture to recurrently refine the base distribution and the bijectors from the latent codes of the previous layer. For our experiments, we used uncoupled deterministic encoders; it would be interesting to explore any predictive benefits of a bottom-up deterministic pass of the inference network, especially for modeling natural images. The architecture could be further refined by adopting hierarchical stochastic layers. Finally, integration of pixel-regressive decoders and importance-weighted variations of the proposed scheme constitute directions for future research.

REFERENCES

Philip Bachman. An architecture for Deep, Hierarchical Generative Models. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 2016.

Vladimir I Bogachev. *Measure theory*, volume 1. Springer Science & Business Media, 2007.

Yuri Burda, Roger B. Grosse, and Ruslan Salakhutdinov. Importance Weighted Autoencoders. In *4th International Conference on Learning Representations, ICLR*, 2016.

Xi Chen, Diederik P. Kingma, Tim Salimans, Yan Duan, Prafulla Dhariwal, John Schulman, Ilya Sutskever, and Pieter Abbeel. Variational Lossy Autoencoder. In *5th International Conference on Learning Representations, ICLR*, 2017.

Bin Dai and David P. Wipf. Diagnosing and Enhancing VAE models. In *7th International Conference on Learning Representations, ICLR*, 2019.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using Real NVP. In *5th International Conference on Learning Representations, ICLR*, 2017.

Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Neural spline flows. In *Advances in Neural Information Processing Systems 32*, 2019.

Mathieu Germain, Karol Gregor, Iain Murray, and Hugo Larochelle. MADE: Masked Autoencoder for Distribution Estimation. In *Proceedings of the 32nd International Conference on Machine Learning, ICML*, 2015.

Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra. DRAW: A Recurrent Neural Network for Image Generation. In *Proceedings of the 32nd International Conference on Machine Learning, ICML*, 2015.

Ishaan Gulrajani, Kundan Kumar, Faruk Ahmed, Adrien Ali Taïga, Francesco Visin, David Vázquez, and Aaron C. Courville. PixelVAE: A Latent Variable Model for Natural Images. In *5th International Conference on Learning Representations, ICLR*, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, June 2016.

Michael I Jordan. An introduction to probabilistic graphical models, 2003.

Diederik P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems 31*, 2018.

Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR*, 2014.

Diederik P. Kingma and Max Welling. An Introduction to Variational Autoencoders. *Foundations and Trends in Machine Learning*, 12(4):307–392, 2019. doi: 10.1561/2200000056. URL `https://doi.org/10.1561/2200000056`.

Diederik P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved Variational Inference with Inverse Autoregressive Flow. In *Advances in Neural Information Processing Systems 29*, 2016.

Alexej Klushyn, Nutan Chen, Richard Kurle, Botond Cseke, and Patrick van der Smagt. Learning Hierarchical Priors in VAEs. In *Advances in Neural Information Processing Systems 32*, 2019a.

Alexej Klushyn, Nutan Chen, Richard Kurle, Botond Cseke, and Patrick van der Smagt. Learning Hierarchical Priors in VAEs. In *Advances in Neural Information Processing Systems 32*, 2019b.

Lars Maaløe, Marco Fraccaro, Valentin Liévin, and Ole Winther. BIVA: A Very Deep Hierarchy of Latent Variables for Generative Modeling. In *Advances in Neural Information Processing Systems 32*, 2019.

George Papamakarios, Theo Pavlakou, and Iain Murray. Masked Autoregressive Flow for Density Estimation. In *Advances in Neural Information Processing Systems 30*, 2017.

Danilo Jimenez Rezende and Fabio Viola. Taming VAEs. In *arXiv preprint arXiv:1810.00597*, 2018.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In *Proceedings of the 31st International Conference on Machine Learning , ICML*, 2014.

Jason Tyler Rolfe. Discrete Variational Autoencoders. In *5th International Conference on Learning Representations, ICLR*, 2017.

Walter Rudin. *Real and complex analysis*. Tata McGraw-hill education, 2006.

Tim Salimans, Diederik Kingma, and Max Welling. Markov Chain Monte Carlo and Variational Inference: Bridging the Gap. In *Proceedings of the 32nd International Conference on Machine Learning, ICML*, 2015.

Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. Ladder Variational Autoencoders. In *Advances in Neural Information Processing Systems 29*, 2016.

Jakub Tomczak and Max Welling. VAE with a VampPrior. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*. PMLR, 2018.

Aäron van den Oord, Nal Kalchbrenner, Lasse Espeholt, Koray Kavukcuoglu, Oriol Vinyals, and Alex Graves. Conditional Image Generation with PixelCNN Decoders. In *Advances in Neural Information Processing Systems 29*, 2016a.

Aäron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel Recurrent Neural Networks. In *Proceedings of the 33nd International Conference on Machine Learning, ICML*, 2016b.

Stefan Webb, Adam Golinski, Rob Zinkov, N Siddharth, Tom Rainforth, Yee Whye Teh, and Frank Wood. Faithful inversion of generative models for effective amortized inference. In *Advances in Neural Information Processing Systems*, pp. 3070–3080, 2018.

Florian Wenzel, Kevin Roth, Bastiaan S Veeling, Jakub Światkowski, Linh Tran, Stephan Mandt, Jasper Snoek, Tim Salimans, Rodolphe Jenatton, and Sebastian Nowozin. How Good is the Bayes Posterior in Deep Neural Networks Really? *arXiv preprint arXiv:2002.02405*, 2020.

Andrew Gordon Wilson and Pavel Izmailov. Bayesian Deep Learning and a Probabilistic Perspective of Generalization. *arXiv preprint arXiv:2002.08791*, 2020.