VAE with a VampPrior

Tomczak, J.M.; Welling, M.

Publication date
2018

Document Version
Final published version

Published in
Proceedings of Machine Learning Research

License
Other

Citation for published version (APA):

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
VAE with a VampPrior

Jakub M. Tomczak
University of Amsterdam

Max Welling
University of Amsterdam

Abstract

Many different methods to train deep generative models have been introduced in the past. In this paper, we propose to extend the variational auto-encoder (VAE) framework with a new type of prior which we call "Variational Mixture of Posteriors" prior, or VampPrior for short. The VampPrior consists of a mixture distribution (e.g., a mixture of Gaussians) with components given by variational posteriors conditioned on learnable pseudo-inputs. We further extend this prior to a two layer hierarchical model and show that this architecture with a coupled prior and posterior, learns significantly better models. The model also avoids the usual local optima issues related to useless latent dimensions that plague VAEs. We provide empirical studies on six datasets, namely, static and binary MNIST, OMNIGLOT, Caltech 101 Silhouettes, Frey Faces and Histopathology patches, and show that applying the hierarchical VampPrior delivers state-of-the-art results on all datasets in the unsupervised permutation invariant setting and the best results or comparable to SOTA methods for the approach with convolutional networks.

1 Introduction

Learning generative models that are capable of capturing rich distributions from vast amounts of data like image collections remains one of the major challenges of machine learning. In recent years, different approaches to achieving this goal were proposed by formulating alternative training objectives to the log-likelihood \[1, 3, 23\] or by utilizing variational inference \[3\]. The latter approach could be made especially efficient through the application of the reparameterization trick resulting in a highly scalable framework now known as the variational auto-encoders (VAE) \[19, 39\]. Various extensions to deep generative models have been proposed that aim to enrich the variational posterior \[10, 29, 32, 35, 39, 40\]. Recently, it has been noticed that in fact the prior plays a crucial role in mediating between the generative decoder and the variational encoder. Choosing a too simplistic prior like the standard normal distribution could lead to over-regularization and, as a consequence, very poor hidden representations \[10\].

In this paper, we take a closer look at the regularization term of the variational lower bound inspired by the analysis presented in \[26\]. Re-formulating the variational lower bound gives two regularization terms: the average entropy of the variational posterior, and the cross-entropy between the averaged (over the training data) variational posterior and the prior. The cross-entropy term can be minimized by setting the prior equal to the average of the variational posteriors over training points. However, this would be computationally very expensive. Instead, we propose a new prior that is a variational mixture of posteriors prior, or VampPrior for short. Moreover, we present a new two-layer VAE that combined with our new prior can learn a very powerful hidden representation.

The contribution of the paper is threefold:

- We follow the line of research of improving the VAE by making the prior more flexible. We propose a new prior that is a mixture of variational posteriors conditioned on learnable pseudo-data. This allows the variational posterior to learn more a potent latent representation.
- We propose a new two-layered generative VAE model with two layers of stochastic latent variables based on the VampPrior idea. This architecture effectively avoids the problems of unused latent dimensions.
- We show empirically that the VampPrior always outperforms the standard normal prior in different VAE architectures and that the hierarchical
VampPrior based VAE achieves state-of-the-art or comparable to SOTA results on six datasets.

2 Variational Auto-Encoder

Let $x$ be a vector of $D$ observable variables and $z \in \mathbb{R}^M$ a vector of stochastic latent variables. Further, let $p_\theta(x, z)$ be a parametric model of the joint distribution. Given data $X = \{x_1, \ldots, x_N\}$ we typically aim at maximizing the average marginal log-likelihood, \( \frac{1}{N} \ln p(X) = \frac{1}{N} \sum_{i=1}^{N} \ln p(x_i) \), with respect to parameters. However, when the model is parameterized by a neural network (NN), the optimization could be difficult due to the intractability of the marginal likelihood. One possible way of overcoming this issue is to apply variational inference and optimize the following lower bound:

\[
E_{x \sim q(x)}[\ln p(x)] \geq E_{x \sim q(x)} \left[ E_{q_\phi(z|x)}[\ln p_\theta(x|z) + \ln p_\lambda(z) - \ln q_\phi(z|x)] \right] + \hat{\Delta} \mathcal{L}(\phi, \theta, \lambda),
\]

where $q(x) = \frac{1}{N} \sum_{n=1}^{N} \delta(x - x_n)$ is the empirical distribution, $q_\phi(z|x)$ is the variational posterior (the encoder), $p_\theta(x|z)$ is the generative model (the decoder) and $p_\lambda(z)$ is the prior, and $\phi, \theta, \lambda$ are their parameters, respectively.

There are various ways of optimizing this lower bound but for continuous $z$ this could be done efficiently through the re-parameterization of $q_\phi(z|x)$ \[19, 33\], which yields a variational auto-encoder architecture (VAE). Therefore, during learning we consider a Monte Carlo estimate of the second expectation in (1) using $L$ sample points:

\[
\hat{\mathcal{L}}(\phi, \theta, \lambda) = E_{x \sim q(x)} \left[ \frac{1}{L} \sum_{l=1}^{L} \left( \ln p_\theta(x|z_\phi^{(l)}) + \ln p_\lambda(z_\phi^{(l)}) - \ln q_\phi(z_\phi^{(l)}|x) \right) \right].
\]

where $z_\phi^{(l)}$ are sampled from $q_\phi(z|x)$ through the re-parameterization trick.

The first component of the objective function can be seen as the expectation of the negative reconstruction error that forces the hidden representation for each data case to be peaked at its specific MAP value. On the contrary, the second and third components constitute a kind of regularization that drives the encoder to match the prior.

We can get more insight into the role of the prior by inspecting the gradient of $\hat{\mathcal{L}}(\phi, \theta, \lambda)$ in (2) and (3) with respect to a single weight $\phi_i$ for a single data point $x$, see Eq. (17) and (18) in Supplementary Material for details. We notice that the prior plays a role of an "anchor" that keeps the posterior close to it, i.e., the term in round brackets in Eq. (18) is 0 if the posterior matches the prior.

Typically, the encoder is assumed to have a diagonal covariance matrix, i.e., $q_\phi(z|x) = \mathcal{N}(z|\mu_\phi(x), \text{diag}(\sigma_\phi^2(x)))$, where $\mu_\phi(x)$ and $\sigma_\phi^2(x)$ are parameterized by a NN with weights $\phi$, and the prior is expressed using the standard normal distribution, $p_\lambda(z) = \mathcal{N}(z|0, I)$. The decoder utilizes a suitable distribution for the data under consideration, e.g., the Bernoulli distribution for binary data or the normal distribution for continuous data, and it is parameterized by a NN with weights $\theta$.

3 The Variational Mixture of Posteriors Prior

Idea The variational lower-bound consists of two parts, namely, the reconstruction error and the regularization term between the encoder and the prior. However, we can re-write the training objective in (1) to obtain two regularization terms instead of one \[20\]:

\[
\mathcal{L}(\phi, \theta, \lambda) = E_{x \sim q(x)} \left[ E_{q_\phi(z|x)}[\ln p_\theta(x|z)] \right] + E_{x \sim q(x)} \left[ \mathbb{H}(q_\phi(z|x)) \right] + \left( E_{x \sim q(x)}[-\ln p_\lambda(z)] - \frac{1}{N} \sum_{n=1}^{N} q_\phi(z|x_n) \right)
\]

where the first component is the negative reconstruction error, the second component is the expectation of the entropy $\mathbb{H}[\cdot]$ of the variational posterior and the last component is the cross-entropy between the aggregated posterior \[10, 26\], $q(z) = \frac{1}{N} \sum_{n=1}^{N} q_\phi(z|x_n)$, and the prior. The second term of the objective encourages the encoder to have large entropy (e.g., high variance) for every data case. The last term aims at matching the aggregated posterior and the prior.

Usually, the prior is chosen in advance, e.g., a standard normal prior. However, one could find a prior that optimizes the ELBO by maximizing the following Lagrange function with the Lagrange multiplier $\beta$:

\[
\max_{p_\lambda(z)} -E_{x \sim q(x)}[-\ln p_\lambda(z)] + \beta \left( \int p_\lambda(z)dz - 1 \right).
\]

The solution of this problem is simply the aggregated posterior:

\[
p_\lambda^*(z) = \frac{1}{N} \sum_{n=1}^{N} q_\phi(z|x_n).
\]

However, this choice may potentially lead to overfitting \[10, 26\] and definitely optimizing the recognition model would become very expensive due to the sum over all training points. On the other hand, having a simple
prior like the standard normal distribution is known to result in over-regularized models with only few active latent dimensions [9].

In order to overcome issues like overfitting, over-regularization and high computational complexity, the optimal solution, i.e., the aggregated posterior, can be further approximated by a mixture of variational posteriors with pseudo-inputs:

\[ p_{\lambda}(z) = \frac{1}{K} \sum_{k=1}^{K} q_{\phi}(z|u_k), \tag{9} \]

where \( K \) is the number of pseudo-inputs, and \( u_k \) is a \( D \)-dimensional vector we refer to as a pseudo-input. The pseudo-inputs are learned through backpropagation and can be thought of as hyperparameters of the prior, alongside parameters of the posterior \( \phi \), \( \lambda = \{u_1, \ldots, u_K, \phi\} \). Importantly, the resulting prior is multimodal, thus, it prevents the variational posterior to be over-regularized. On the other hand, incorporating pseudo-inputs prevents from potential overfitting once we pick \( K \ll N \), which also makes the model less expensive to train. We refer to this prior as the variational mixture of posteriors prior (VampPrior).

A comparison to a mixture of Gaussians prior

A simpler alternative to the VampPrior that still approximates the optimal solution of the problem in (7) is a mixture of Gaussians (MoG), \( p_{\lambda}(z) = \frac{1}{K} \sum_{k=1}^{K} \mathcal{N}(\mu_k, \text{diag}(\sigma_k^2)) \). The hyperparameters of the prior \( \lambda = \{\mu_k, \text{diag}(\sigma_k^2)\}_{k=1}^{K} \) are trained by backpropagation similarly to the pseudo-inputs. The MoG prior influences the variational posterior in the same manner to the standard prior and the gradient of the ELBO with respect to the encoder’s parameters takes an analogous form to (17) and (18), see Supplementary Material for details.

In the case of the VampPrior, on the other hand, we obtain two advantages over the MoG prior:

- First, by coupling the prior with the posterior we entertain fewer parameters and the prior and variational posteriors will at all times “cooperate” during training.
- Second, this coupling highly influences the gradient wrt a single weight of the encoder, \( \phi_l \), for a given \( x \), see Eq. (20) and (21) in Supplementary Material for details. The differences in (20) and (21) are close to 0 as long as \( q_{\phi}(z_l|\phi|x) \approx q_{\phi}(z_l|u_k) \). Thus, the gradient is influenced by pseudo-inputs that are dissimilar to \( x \), i.e., if the posterior produces different hidden representations for \( u_k \) and \( x \). In other words, since this has to hold for every training case, the gradient points towards a solution where the variational posterior has high variance. On the contrary, the first part of the objective in (19) causes the posteriors to have low variance and map to different latent explanations for each data case. These effects distinguish the VampPrior from the MoG prior utilized in the VAE so far [9, 29].

A connection to the Empirical Bayes

The idea of the Empirical Bayes (EB), also known as the type-II maximum likelihood, is to find hyperparameters \( \lambda \) of the prior over latent variables \( z \), \( p(z|\lambda) \), by maximizing the marginal likelihood \( p(x|\lambda) \). In the case of the VAE and the VampPrior the pseudo-inputs, alongside the parameters of the posterior, are the hyperparameters of the prior and we aim at maximizing the ELBO with respect to them. Thus, our approach is closely related to the EB and in fact it formulates a new kind of Bayesian inference that combines the variational inference with the EB approach.

A connection to the Information Bottleneck

We have shown that the aggregated posterior is the optimal prior within the VAE formulation. This result is closely related to the Information Bottleneck (IB) approach [1, 38] where the aggregated posterior naturally plays the role of the prior. Interestingly, the VampPrior brings the VAE and the IB formulations together and highlights their close relation. A similar conclusion and a thorough analysis of the close relation between the VAE and the IB through the VampPrior is presented in [2].

4 Hierarchical VampPrior Variational Auto-Encoder

Hierarchical VAE and the inactive stochastic latent variable problem

A typical problem encountered during training a VAE is the inactive stochastic units problem [1, 25]. Our VampPrior VAE seems to be an effective remedy against this issue, simply because the prior is designed to be rich and multimodal, preventing the KL term from pulling individual posteriors towards a simple (e.g., standard normal) prior. The inactive stochastic units problem is even worse for learning deeper VAEs (i.e., with multiple layers of stochastic units). The reason might be that stochastic dependencies within a deep generative model are top-down in the generative process and bottom-up in the variational process. As a result, there is less information obtained from real data at the deeper stochastic layers, making them more prone to become regularized towards the prior.

In order to prevent a deep generative model to suffer from the inactive stochastic units problem we propose...
A new two-layered VAE as follows:

\[ q_\phi(z_1|x, z_2) \quad q_\psi(z_2|x), \]

while the generative part is the following:

\[ p_\theta(x|z_1, z_2) \quad p_\lambda(z_1|z_2) \quad p(z_2), \]

with \( p(z_2) \) given by a VampPrior. The model is depicted in Figure 1(b).

In this model we use normal distributions with diagonal covariance matrices to model \( z_1 \in \mathbb{R}^{M_1} \) and \( z_2 \in \mathbb{R}^{M_2} \), parameterized by NNs. The full model is given by:

\[ p(z_2) = \frac{1}{K} \sum_{k=1}^{K} q_\psi(z_2|u_k), \]

where \( \mu_k \in \mathbb{R}^{M_2}, \sigma_k^2 \in \mathbb{R}^{M_2} \) are trainable parameters;

- the VampPrior with a random subset of real training data as non-trainable pseudo-inputs (VampPrior data).

Including the standard prior can provide us with an answer to the general question if there is even a need for complex priors. Utilizing the mixture of Gaussians verifies whether it is beneficial to couple the prior with the variational posterior or not. Finally, using a subset of real training images determines to what extent it is useful to introduce trainable pseudo-inputs.

5 Experiments

5.1 Setup

In the experiments we aim at: (i) verifying empirically whether the VampPrior helps the VAE to train a representation that better reflects variations in data, and (ii) inspecting if our proposition of a two-level generative model performs better than the one-layered model. In order to answer these questions we further verify three alternative priors:

- the standard Gaussian prior (SG):

\[ p(z_2) = \mathcal{N}(0, I); \]

- the mixture of Gaussians prior (MoG):

\[ p(z_2) = \frac{1}{K} \sum_{k=1}^{K} \mathcal{N}(\mu_k, \text{diag}(\sigma_k^2)), \]

where \( \mu_k \in \mathbb{R}^{M_2}, \sigma_k^2 \in \mathbb{R}^{M_2} \) are trainable parameters.

We carry out experiments using six image datasets: static MNIST [21], dynamic MNIST [34], OMNIGLOT...
Caltech 101 Silhouette [27], Frey Faces [1] and Histopathology patches [39]. More details about the datasets can be found in the Supplementary Material.

In the experiments we modeled all distributions using MLPs with two hidden layers of 300 hidden units in the unsupervised permutation invariant setting. We utilized the gating mechanism as an element-wise non-linearity [8]. We utilized 40 stochastic hidden units for both $z_1$ and $z_2$. Next we replaced MLPs with convolutional layers with gating mechanism. Eventually, we verified also a PixelCNN [42] as the decoder. For Frey Faces and Histopathology we used the discretized logistic distribution of images as in [18], and for other datasets we applied the Bernoulli distribution.

For learning the ADAM algorithm [17] with normalized gradients [13] was utilized with the learning rate in \(10^{-4}, 5 \cdot 10^{-4}\) and mini-batches of size 100. Additionally, to boost the generative capabilities of the decoder, we used the warm-up for 100 epochs [5]. The weights of the neural networks were initialized according to [12]. The early-stopping with a look ahead of 50 iterations was applied. For the VampPrior we used 500 pseudo-inputs for all datasets except OMNIGLOT for which we utilized 1000 pseudo-inputs. For the VampPrior data we randomly picked training images instead of the learnable pseudo-inputs.

We denote the hierarchical VAE proposed in this paper with MLPs by HVAE, while the hierarchical VAE with convolutional layers and additionally with a PixelCNN decoder are denoted by convHVAE and PixelHVAE, respectively.

5.2 Results

Quantitative results We quantitatively evaluate our method using the test marginal log-likelihood (LL) estimated using the Importance Sampling with 5,000 sample points [6, 33]. In Table 1 we present a comparison between models with the standard prior and the VampPrior. The results of our approach in comparison to the state-of-the-art methods are gathered in Table 2, 3, 4 and 5 for static and dynamic MNIST, OMNIGLOT and Caltech 101 Silhouettes, respectively.

First we notice that in all cases except one the application of the VampPrior results in a substantial improvement of the generative performance in terms of the test LL comparing to the standard normal prior (see Table 1). This confirms our supposition that a combination of multimodality and coupling the prior with the posterior is superior to the standard normal prior. Further, we want to stress out that the VampPrior outperforms other priors like a single Gaussian or a mixture of Gaussians (see Table 2). These results provide an additional evidence that the VampPrior leads to a more powerful latent representation and it differs from the MoG prior. We also examined whether the presented two-layered model performs better than the widely used hierarchical architecture of the VAE. Indeed, the newly proposed approach is more powerful even with the SG prior (HVAE \((L = 2) + \text{SG}\)) than the standard two-layered VAE (see Table 2). Applying the MoG prior results in an additional boost of performance. This provides evidence for the usefulness of a multimodal prior. The VampPrior data gives only slight improvement comparing to the SG prior and because of the application of the fixed training data as the pseudo-inputs it is less flexible than the MoG. Eventually, coupling the variational posterior with the prior and introducing learnable pseudo-inputs gives the best performance.

Additionally, we compared the VampPrior with the MoG prior and the SoG prior in Figure 2 for varying number of pseudo-inputs/components. Surprisingly, taking more pseudo-inputs does not help to improve the performance and, similarly, considering more mixture components also resulted in drop of the performance. However, again we can notice that the VampPrior is more flexible prior that outperforms the MoG.

An inspection of histograms of the log-likelihoods (see Supplementary Material) shows that the distributions of LL values are heavy-tailed and/or bimodal. A possible explanation for such characteristics of the histograms is the existence of many examples that are relatively simple to represent (first mode) and some really hard examples (heavy-tail). Comparing our ap-

![Figure 2: A comparison between the HVAE \((L = 2)\) with SG prior, MoG prior and VampPrior in terms of ELBO and varying number of pseudo-inputs/components on static MNIST.](http://www.cs.nyu.edu/~roweis/data/frey_rawface.mat)
VAMPPrior. For last two datasets an average of bits per data dimension is given. In the case of Frey Faces, for all models the standard deviation was no larger than 0.03 and that this why it is omitted in the table.

<table>
<thead>
<tr>
<th>DATASET</th>
<th>VAE (L = 1)</th>
<th>HVAE (L = 2)</th>
<th>convHVAE (L = 2)</th>
<th>PixelHVAE (L = 2)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>standard</td>
<td>VampPrior</td>
<td>standard</td>
<td>VampPrior</td>
</tr>
<tr>
<td>staticMNIST</td>
<td>−88.56</td>
<td>−85.57</td>
<td>−86.05</td>
<td>−83.19</td>
</tr>
<tr>
<td>dynamicMNIST</td>
<td>−84.50</td>
<td>−82.38</td>
<td>−82.42</td>
<td>−81.24</td>
</tr>
<tr>
<td>Omniglot</td>
<td>−108.50</td>
<td>−104.75</td>
<td>−103.52</td>
<td>−101.18</td>
</tr>
<tr>
<td>Caltech 101</td>
<td>−123.43</td>
<td>−114.55</td>
<td>−112.08</td>
<td>−108.28</td>
</tr>
<tr>
<td>Frey Faces</td>
<td>4.63</td>
<td>4.57</td>
<td>4.31</td>
<td>4.51</td>
</tr>
<tr>
<td>Histopathology</td>
<td>6.07</td>
<td>6.04</td>
<td>5.82</td>
<td>5.75</td>
</tr>
</tbody>
</table>

Table 2: Test LL for static MNIST.

<table>
<thead>
<tr>
<th>MODEL</th>
<th>LL</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAE (L = 1) + NF [32]</td>
<td>−85.10</td>
</tr>
<tr>
<td>VAE (L = 2)</td>
<td>−87.86</td>
</tr>
<tr>
<td>IWAE (L = 2) [6]</td>
<td>−85.32</td>
</tr>
<tr>
<td>HVAE (L = 2) + SG</td>
<td>−85.89</td>
</tr>
<tr>
<td>HVAE (L = 2) + MoG</td>
<td>−85.07</td>
</tr>
<tr>
<td>HVAE (L = 2) + VampPrior data</td>
<td>−85.71</td>
</tr>
<tr>
<td>AVB + AC (L = 1) [28]</td>
<td>−80.20</td>
</tr>
<tr>
<td>VLAE [7]</td>
<td>−79.03</td>
</tr>
<tr>
<td>VAE + IAF [18]</td>
<td>−79.88</td>
</tr>
<tr>
<td>convHVAE (L = 2) + VampPrior</td>
<td>−81.09</td>
</tr>
<tr>
<td>PixelHVAE (L = 2) + VampPrior</td>
<td>−79.78</td>
</tr>
</tbody>
</table>

Table 3: Test LL for dynamic MNIST.

<table>
<thead>
<tr>
<th>MODEL</th>
<th>LL</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAE (L = 2) + VGP [40]</td>
<td>−81.32</td>
</tr>
<tr>
<td>CAGEM-0 (L = 2) [25]</td>
<td>−81.60</td>
</tr>
<tr>
<td>IVAE (L = 5) [30]</td>
<td>−81.74</td>
</tr>
<tr>
<td>HVAE (L = 2) + VampPrior data</td>
<td>−81.71</td>
</tr>
<tr>
<td>HVAE (L = 2) + VampPrior</td>
<td>−81.24</td>
</tr>
<tr>
<td>VLAE [7]</td>
<td>−78.53</td>
</tr>
<tr>
<td>VAE + IAF [18]</td>
<td>−79.10</td>
</tr>
<tr>
<td>PixelVAE [15]</td>
<td>−78.96</td>
</tr>
<tr>
<td>convHVAE (L = 2) + VampPrior</td>
<td>−79.78</td>
</tr>
<tr>
<td>PixelHVAE (L = 2) + VampPrior</td>
<td>−78.45</td>
</tr>
</tbody>
</table>

Table 4: Test LL for OMNIGLOT.

<table>
<thead>
<tr>
<th>MODEL</th>
<th>LL</th>
</tr>
</thead>
<tbody>
<tr>
<td>VR-max (L = 2) [24]</td>
<td>−103.72</td>
</tr>
<tr>
<td>IWAE (L = 2) [7]</td>
<td>−103.38</td>
</tr>
<tr>
<td>IVAE (L = 5) [30]</td>
<td>−102.11</td>
</tr>
<tr>
<td>HVAE (L = 2) + VampPrior</td>
<td>−101.18</td>
</tr>
<tr>
<td>VLAE [7]</td>
<td>−89.83</td>
</tr>
<tr>
<td>convHVAE (L = 2) + VampPrior</td>
<td>−97.56</td>
</tr>
<tr>
<td>PixelHVAE (L = 2) + VampPrior</td>
<td>−89.76</td>
</tr>
</tbody>
</table>

Table 5: Test LL for Caltech 101 Silhouettes.

<table>
<thead>
<tr>
<th>MODEL</th>
<th>LL</th>
</tr>
</thead>
<tbody>
<tr>
<td>IWAE (L = 1) [24]</td>
<td>−117.21</td>
</tr>
<tr>
<td>VR-max (L = 1) [24]</td>
<td>−117.10</td>
</tr>
<tr>
<td>HVAE (L = 2) + VampPrior</td>
<td>−108.28</td>
</tr>
<tr>
<td>VLAE [7]</td>
<td>−78.53</td>
</tr>
<tr>
<td>convHVAE (L = 2) + VampPrior</td>
<td>−104.22</td>
</tr>
<tr>
<td>PixelHVAE (L = 2) + VampPrior</td>
<td>−86.22</td>
</tr>
</tbody>
</table>

approach to the VAE reveals that the VAE with the VampPrior is not only better on average but it produces less examples with high values of LL and more examples with lower LL.

We hypothesized that the VampPrior provides a remedy for the inactive units issue. In order to verify this claim we utilized the statistics introduced in [6]. The results for the HVAE with the VampPrior in comparison to the two-level VAE and IWAE presented in [6] are given in Figure 3. The application of the VampPrior increases the number of active stochastic units four times for the second level and around 1.5 times more for the first level comparing to the VAE and the IWAE. Interestingly, the number of mixture components has a great impact on the number of active stochastic units in the second level. Nevertheless, even one mixture component allows to achieve almost three times more active stochastic units comparing to the vanilla VAE and the IWAE.

In general, an application of the VampPrior improves the performance of the VAE and in the case of two layers of stochastic units it yields the state-of-the-art results on all datasets for models that use MLPs. Moreover, our approach gets closer to the performance of models that utilize convolutional neural networks, such as, the one-layered VAE with the inverse autoregressive flow (IAF) [18] that achieves −79.88 on static MNIST and −79.10 on dynamic MNIST, the one-
layered Variational Lossy Autoencoder (VLAE) [7] that obtains $-79.03$ on static MNIST and $-78.53$ on dynamic MNIST, and the hierarchical PixelVAE [15] that gets $-78.96$ on dynamic MNIST. On the other two datasets the VLAE performs way better than our approach with MLPs and achieves $-89.83$ on OMNIGLOT and $-77.36$ on Caltech 101 Silhouettes.

In order to compare our approach to the state-of-the-art convolutional-based VAEs we performed additional experiments using the HVAE ($L = 2$) + VampPrior with convolutional layers in both the encoder and decoder. Next, we replaced the convolutional decoder with the PixelCNN [42] decoder (CONVHVAE and PIXELHVAE in Tables 2, 3). For the PIXELHVAE we were able to improve the performance to obtain $-79.78$ on static MNIST, $-78.45$ on dynamic MNIST, $-89.76$ on Omniglot, and $-86.22$ on Caltech 101 Silhouettes. The results confirmed that the VampPrior combined with a powerful encoder and a flexible decoder performs much better than the MLP-based approach and allows to achieve state-of-the-art performance on dynamic MNIST and OMNIGLOT.

**Qualitative results** The biggest disadvantage of the VAE is that it tends to produce blurry images [22]. We noticed this effect in images generated and reconstructed by VAE (see Supplementary Material). Moreover, the standard VAE produced some digits that are hard to interpret, blurry characters and very noisy silhouettes. The supremacy of HVAE + VampPrior is visible not only in LL values but in image generations and reconstructions as well because these are sharper.

We also examine what the pseudo-inputs represent at the end of the training process (see Figure 4). Interestingly, trained pseudo-inputs are prototypical objects (digits, characters, silhouettes). Moreover, images generated for a chosen pseudo-input show that the model encodes a high variety of different features such as shapes, thickness and curvature for a single pseudo-input. This means that the model is not just memorizing the data-cases. It is worth noticing that for small-sample size and/or too flexible decoder the pseudo-inputs can be hard to train and they can represent noisy prototypes (e.g., see pseudo-inputs for Frey Faces in Figure 4).

### 6 Related work

The VAE is a latent variable model that is usually trained with a very simple prior, i.e., the standard normal prior. In [30] a Dirichlet process prior using a stick-breaking process was proposed, while [14] proposed a nested Chinese Restaurant Process. These priors enrich the generative capabilities of the VAE, however, they require sophisticated learning methods and tricks to be trained successfully. A different approach is to use an autoregressive prior [7] that applies the IAF to random noise. This approach gives very promising results and allows to build rich representations. Nevertheless, the authors of [7] combine their prior with a convolutional encoder and an autoregressive decoder that makes it harder to assess the real contribution of the autoregressive prior to the generative model.

Clearly, the quality of generated images are dependent on the decoder architecture. One way of improving generative capabilities of the decoder is to use an infinite mixture of probabilistic component analyzers [37], which is equivalent to a rank-one covariance matrix. A more appealing approach would be to use deep autoregressive density estimators that utilize recurrent neural networks [12] or gated convolutional networks [11]. However, there is a threat that a too flexible decoder could discard hidden representations completely, turning the encoder to be useless [7].

Concurrently to our work, in [4] a variational VAE with a memory was proposed. This approach shares similarities with the VampPrior in terms of a learnable memory (pseudo-inputs in our case) and a multimodal prior. Nevertheless, there are two main differences. First, our prior is an explicit mixture while they sample components. Second, we showed that the optimal prior requires to be coupled with the variational posterior. In the experiments we showed that the VampPrior improves the generative capabilities of the VAE but in [4] it was noticed that the generative performance is comparable to the standard normal prior. We claim
that the success of the VampPrior follows from the utilization of the variational posterior in the prior, a part that is missing in [4].

Very recently, the VampPrior was shown to be a perfect match for the information-theoretic approach to learn latent representation [2]. Additionally, the authors of [2] proposed to use a weighted version of the VampPrior:

\[
p_\lambda(z) = \sum_{k=1}^{K} w_k q_\phi(z | u_k), \tag{16}
\]

where \(w_1, \ldots, w_K\) are trainable parameters such that \(\forall_k w_k \geq 0\) and \(\sum_k w_k = 1\). This allows the VampPrior to learn which components (i.e., pseudo-inputs) are meaningful and may prevent from potential overfitting.

7 Conclusion

In this paper, we followed the line of thinking that the prior is a critical element to improve deep generative models, and in particular VAEs. We proposed a new prior that is expressed as a mixture of variational posteriors. In order to limit the capacity of the prior we introduced learnable pseudo-inputs as hyper-parameters of the prior, the number of which can be chosen freely. Further, we formulated a new two-level generative model based on this VampPrior. We showed empirically that applying our prior can indeed increase the performance of the proposed generative model and successfully overcome the problem of inactive stochastic latent variables, which is particularly challenging for generative models with multiple layers of stochastic latent variables. As a result, we achieved state-of-the-art or comparable results to SOTA models on six datasets. Additionally, generations and reconstructions obtained from the hierarchical VampPrior VAE are of better quality than the ones achieved by the standard VAE.

We believe that it is worthwhile to further pursue the line of the research presented in this paper. Here we applied our prior to image data but it would be interesting to see how it behaves on text or sound, where the sequential aspect plays a crucial role. We have already showed that combining the VampPrior VAE with convolutional nets and a powerful autoregressive density estimator is beneficial but more thorough study is needed. Last but not least, it would be interesting to utilize a normalizing flow [32], the hierarchical variational inference [31], ladder networks [36] or adversarial training [28] within the VampPrior VAE. However, we leave investigating these issues for future work.

Acknowledgements

The authors are grateful to Rianne van den Berg, Yingzhen Li, Tim Genewein, Tameem Adel, Ben Poole and Alex Alemi for insightful comments.

The research conducted by Jakub M. Tomczak was funded by the European Commission within the Marie Skłodowska-Curie Individual Fellowship (Grant No. 702666, "Deep learning and Bayesian inference for medical imaging").
References


