The Variational Homoencoder: Learning to learn high capacity generative models from few examples

Luke B. Hewitt[12] **Maxwell I. Nye**[1] **Andreea Gane**[1] **Tommi Jaakkola**[1] **Joshua B. Tenenbaum**[1]

[1]Massachusetts Institute of Technology
[2]MIT-IBM Watson AI Lab

Abstract

Hierarchical Bayesian methods can unify many related tasks (e.g. k-shot classification, conditional and unconditional generation) as inference within a single generative model. However, when this generative model is expressed as a powerful neural network such as a PixelCNN, we show that existing learning techniques typically fail to effectively use latent variables. To address this, we develop a modification of the Variational Autoencoder in which encoded observations are decoded to new elements from the same class. This technique, which we call a *Variational Homoencoder* (VHE), produces a hierarchical latent variable model which better utilises latent variables. We use the VHE framework to learn a hierarchical PixelCNN on the Omniglot dataset, which outperforms all existing models on test set likelihood and achieves strong performance on one-shot generation and classification tasks. We additionally validate the VHE on natural images from the YouTube Faces database. Finally, we develop extensions of the model that apply to richer dataset structures such as factorial and hierarchical categories.

1 INTRODUCTION

Learning from few examples is possible only with strong inductive biases. In machine learning these biases can be hand designed, such as a model's parametrisation, or can be the result of a meta-learning algorithm. Furthermore they may be task-specific, as in discriminative modelling, or may describe the world causally so as to be naturally reused across many tasks. Recent work has approached one- and few-shot learning from all of these perspectives.

Much research has focused on developing neural architectures for few-shot classification (Koch, 2015; Vinyals et al., 2016; Snell et al., 2017; Santoro et al., 2016). These discriminatively-trained networks take as input a test example and a 'support set' of examples from several novel classes, and determine the most likely classification of the test example within the novel classes. A second approach, as explored in Ravi & Larochelle (2016); Finn et al. (2017), is to use only a standard classification network but adapt its parameters to the support examples with a learned initialisation and update rule. In either case, such discriminative models can achieve state-of-the-art few-shot classification performance, although they provide no principled means for transferring knowledge to other tasks.

An alternative approach centers on few-shot learning of *generative* models, from which good classification ought to come for free. Much recent work on meta-learning aims to take one or a few observations from a set D as input, and produce a distribution over new elements $p(x|D)$ by some learning procedure, expressed either as a neural network (Rezende et al., 2016; Bartunov & Vetrov, 2016; Reed et al., 2017) or by adapting the parameters of an unconditional model (Reed et al., 2017).

A promising route to learning generative models is hierarchical Bayesian inference, which aims to capture shared structure between instances through *shared* latent variables. A recent example is developed in Lake et al. (2015): a compositional, causal generative model of handwritten characters which achieves state-of-the-art results at few-shot character classification, alphabet classification, and both conditional and unconditional generation. However, this model was hand engineered for the Omniglot domain, and so leaves open the challenge of how to learn such hierarchical Bayesian models using only a generic architecture. The recently proposed *Neu-*

A PyTorch implementation of the Variational Homoencoder can be found at github.com/insperatum/vhe.

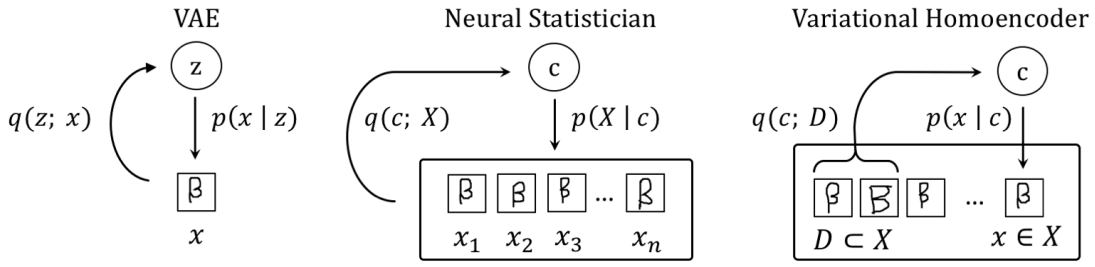

Figure 1: Single step of gradient training in various models. A VAE treats all datapoints as independent, so only a single random element need be encoded (with $q(z;x)$) and decoded (with $p(x|z)$) each step. A Neural Statistician instead feeds a full set of elements X through both encoder ($q(c;X)$) and decoder ($p(X|c)$) networks, in order to share a latent variable c. In a VHE, we bound the full likelihood $p(X)$ using only random subsamples D and x for encoding/decoding. Optionally, the decoder $p(x|c)$ may be defined through a local latent variable z.

ral Statistician (Edwards & Storkey, 2016) offers one means towards this, using amortised variational inference to support learning in a deep hierarchical generative model.

In this work we aim to learn generative models, expressed using high capacity neural network architectures, from just a few examples of a concept. To this end we propose the *Variational Homoencoder* (VHE), combining several advantages of the models described above:

1. Like conditional generative approaches (*e.g.* Rezende et al. (2016)), we train on a few-shot generation objective which matches how our model may be used at test time. However, by introducing an encoding cost, we simultaneously optimise a likelihood lower bound for a hierarchical generative model, in which structure shared across elements is made explicit by shared latent variables.

2. Edwards & Storkey (2016) has learned hierarchical Bayesian models by applying Variational Autoencoders to sets, such as classes of images. However, their approach requires feeding a full set through the model per gradient step (Figure 1), rendering it intractable to train on very large sets. In practice, they avoid computational limits by training on smaller, random subsets. In a VHE, we instead optimise a likelihood bound for the complete dataset, while *constructing this bound* by subsampling. This approach can not only improve generalisation, but also departs from previous work by extending to models with richer latent structure, for which the joint likelihood cannot be factorised.

3. As with a VAE, the VHE objective includes both an *encoding-* and *reconstruction-* cost. However, by sharing latent variables across a large set of elements, the *encoding cost* per element is reduced significantly. This facilitates use of powerful autoregressive decoders, which otherwise often suffer from ignoring latent variables (Chen et al., 2016). We demonstrate the significance of this by applying a VHE to the Omniglot dataset. Using a PixelCNN decoder (Oord et al., 2016), our generative model is arguably the first with a general purpose architecture to both attain near state-of-the-art one-shot classification performance and produce high quality samples in one-shot generation.

2 BACKGROUND

2.1 VARIATIONAL AUTOENCODERS

When dealing with latent variable models of the form $p(x) = \int_z p(z)p(x|z)\mathrm{d}z$, the integration is necessary for both learning and inference but is often intractable to compute in closed form. *Variational Autoencoders* (VAEs, Kingma & Welling (2013)) provide a method for learning such models by utilising neural-network based approximate posterior inference. Specifically, a VAE comprises a generative network $p_\theta(z)p_\theta(x|z)$ alongside a separate inference network $q_\phi(z;x)$. These are trained jointly to maximise a single objective:

$$\mathcal{L}_X(\theta,\phi) = \sum_{x\in X}\Big[\log p_\theta(x) - \mathrm{D}_{KL}\Big(q_\phi(z;x) \parallel p_\theta(z|x)\Big)\Big] \tag{1}$$

$$= \sum_{x\in X}\left[\mathop{\mathbb{E}}_{q_\phi(z;x)} \log p_\theta(x|z) - \mathrm{D}_{KL}\Big(q_\phi(z;x) \parallel p_\theta(z)\Big)\right] \tag{2}$$

As can be seen from Equation 1, this objective $\mathcal{L}_X$ is a lower bound on the total log likelihood of the dataset

Algorithm 1: Minibatch training for the *Variational Homoencoder*. Minibatches are of size M. Stochastic inference network q uses subsets of size N.

initialize (θ, ϕ) — *Parameters for decoder p and encoder q*
repeat
  sample (x_k, i_k) for $k = 1, \ldots, M$ — *Minibatch of elements with corresponding class labels*
  sample $D_k \subseteq X_{i_k}$ for $k = 1, \ldots, M$ — *where $|D_k| = N$*
  sample $c_k \sim q_\phi(c; D_k)$ for $k = 1, \ldots, M$
  (optional) sample $z_k \sim q_\phi(z; c_k, x_k)$ for $k = 1, \ldots, M$
  $\mathbf{g} \approx \frac{1}{M} \sum_k \nabla \mathcal{L}_{\theta,\phi}(x_k; D_k, |X_{i_k}|)$ — *Reparametrization gradient estimate using* $\mathbf{c}, \mathbf{z}$
  $(\theta, \phi) \leftarrow (\theta, \phi) + \lambda \mathbf{g}$ — *Gradient step, e.g SGD*
until convergence of (θ, ϕ)

$\sum_{x \in X} \log p_\theta(x)$, while $q_\phi(z; x)$ is trained to approximate the true posterior $p_\theta(z|x)$ as accurately as possible. If it could match this distribution exactly then the bound would be tight so that the VAE objective equals the true log likelihood of the data. In practice, the resulting model is typically a compromise between two goals: pulling p_θ towards a distribution that assigns high likelihood to the data, but also towards one which allows accurate inference by q_ϕ. Equation 2 provides a formulation for the same objective which can be optimised stochastically, using Monte-Carlo integration to approximate the expectation. For brevity, we will omit subscripts θ, ϕ for the remainder of this paper.

2.2 VARIATIONAL AUTOENCODERS OVER SETS

The *Neural Statistician* (Edwards & Storkey, 2016) is a Variational Autoencoder in which each item to be encoded is itself a set, such as the set $X^{(i)}$ of all images with a particular class label i:

$$X^{(i)} = \{x_1^{(i)}, x_2^{(i)}, \cdots, x_n^{(i)}\} \tag{3}$$

The generative model for sets, $p(X)$, is described by introduction of a corresponding latent variable c. Given c, individual $x \in X$ are conditionally independent:

$$p(X) = \int_c p(c) \prod_{x \in X} p(x|c) \mathrm{d}c \tag{4}$$

The likelihood is again intractable to compute, but it can be bounded below via:

$$\log p(X) \geq \mathcal{L}_X = \mathop{\mathbb{E}}_{q(c;X)} \left[\sum_{x \in X} \log p(x|c) \right] - \mathrm{D}_{KL}\big(q(c; X) \parallel p(c)\big) \tag{5}$$

Unfortunately, calculating the variational lower bound for each set X requires evaluating both $q(c; X)$ and $p(X|c)$, meaning that the entire set must be passed through both networks for each gradient update. This can become computationally challenging for classes with hundreds of examples. Instead, previous work (Edwards & Storkey, 2016) ensures that sets used for training are always of small size by maximising a log-likelihood bound for randomly sampled subsets $D \subset X$:

$$\mathop{\mathbb{E}}_{D \subset X} \left[\mathop{\mathbb{E}}_{q(c;D)} \left[\sum_{x \in D} \log p(x|c) \right] - \mathrm{D}_{KL}\big(q(c; D) \parallel p(c)\big) \right] \tag{6}$$

As we demonstrate in section 4, this subsampling decreases the model's incentive to capture correlations within a class, reducing utilisation of the latent variables. This poses a significant challenge when scaling up to more powerful generative networks, which require a greater incentive to avoid simply memorising the global distribution. Our work addresses this by replacing the variational lower-bound in Equation 6 with a new objective, which better incentivises the use of latent variables, leading to improved generalisation.

3 VARIATIONAL HOMOENCODERS

Rather than bound the likelihood of subsamples D from a set, as in Edwards & Storkey (2016), we instead use subsampling to construct a lower bound on the complete set X. We use a constrained variational distribution $q(c; D), D \subseteq X$ for posterior inference and an unbiased stochastic approximation $\log p(x|c), x \in X$ for the likelihood. This bound will typically be loose due to stochasticity in sampling D, and we view this as a regularization strategy: we aim to learn latent representations that are quickly inferable from a small number of instances, and the VHE objective is tailored for this purpose.

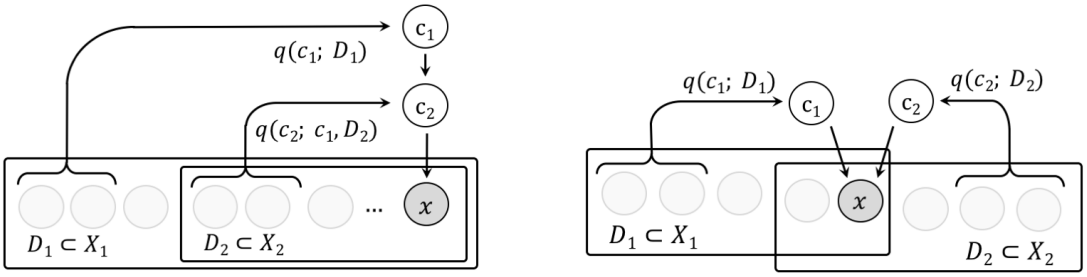

Figure 2: Application of VHE framework to hierarchical (*left*) and factorial (*right*) models. Given an element x such that $x \in X_1$ and $x \in X_2$, an approximate posterior is constructed for the corresponding shared latent variables c_1, c_2 using subsampled sets $D_1 \subset X_1, D_2 \subset X_2$.

3.1 STOCHASTIC LOWER BOUND

We would like to learn a generative model for sets X of the form

$$p(X) = \int p(c) \prod_{x \in X} p(x|c) \mathrm{d}c \tag{7}$$

We will refer our full dataset as a union of disjoint sets $\mathcal{X} = X_1 \sqcup X_2 \sqcup \ldots \sqcup X_n$, and use $X_{(x)}$ to refer to the set $X_i \ni x$. Using the standard consequent of Jensen's inequality, we can lower bound the log-likelihood of each set X using an arbitrary distribution q. In particular, we give q as a fixed function of arbitrary data.

$$\log p(X) \geq \mathop{\mathbb{E}}_{q(c;D)} \log p(X|c) - \mathrm{D}_{KL}\big[q(c;D) \parallel p(c)\big], \qquad \forall D \subset X \tag{8}$$

Splitting up individual likelihoods, we may rewrite

$$\log p(X) \geq \mathop{\mathbb{E}}_{q(c;D)} \Big[\sum_{x \in X} \log p(x|c) \Big] - \mathrm{D}_{KL}\big[q(c;D) \parallel p(c)\big], \qquad \forall D \subset X \tag{9}$$

$$= \sum_{x \in X} \Big[\mathop{\mathbb{E}}_{q(c;D)} \log p(x|c) - \frac{1}{|X|} \mathrm{D}_{KL}\big[q(c;D) \parallel p(c)\big] \Big], \qquad \forall D \subset X \tag{10}$$

$$\stackrel{\text{def}}{=} \sum_{x \in X} \mathcal{L}(x; D, |X|), \qquad \forall D \subset X \tag{11}$$

Finally, we can replace the universal quantification with an expectation under any distribution of D (e.g. uniform sampling from X without replacement):

$$\log p(X) \geq \mathop{\mathbb{E}}_{D \subset X} \sum_{x \in X} \mathcal{L}(x; D, |X|) \tag{12}$$

$$= \sum_{x \in X} \mathop{\mathbb{E}}_{D \subset X} \mathcal{L}(x; D, |X|) \tag{13}$$

$$\log p(\mathcal{X}) \geq \sum_{x \in \mathcal{X}} \mathop{\mathbb{E}}_{D \subset X_{(x)}} \mathcal{L}(x; D, |X_{(x)}|) \tag{14}$$

This formulation suggests a simple modification to the VAE training procedure, as shown in Algorithm 1. At each iteration we select an element x, use resampled elements $D \subset X_{(x)}$ to construct the approximate posterior $q(c; D)$, and rescale the encoding cost appropriately.

VHE objective:

$$\mathop{\mathbb{E}}_{\substack{x \in \mathcal{X} \\ D \subset X_{(x)}}} \left[\mathop{\mathbb{E}}_{q(c;D)} \log p(x|c) - \frac{1}{|X_{(x)}|} \mathrm{D}_{KL}\big[q(c;D) \parallel p(c)\big] \right] \tag{15}$$

If the generative model $p(x|c)$ also describes a separate latent variable z for each element, we may simply introduce a second inference network $q(z; c, x)$ in order to further bound the reconstruction error of Equation 15:

VHE objective with per-element latent variables:

$$\mathop{\mathbb{E}}_{\substack{x \in \mathcal{X} \\ D \subset X_{(x)}}} \left[\mathop{\mathbb{E}}_{\substack{q(c;D) \\ q(z;c,x)}} \log p(x|c,z) - \mathrm{D}_{KL}\big[q(z;c,x) \parallel p(z|c)\big] - \frac{1}{|X_{(x)}|} \mathrm{D}_{KL}\big[q(c;D) \parallel p(c)\big] \right] \tag{16}$$

3.2 APPLICATION TO STRUCTURED DATASETS

The above derivation applies to a dataset partitioned into *disjoint* subsets $\mathcal{X} = X_1 \sqcup X_2 \sqcup \ldots \sqcup X_n$, each with a corresponding latent variable c_i. However, many datasets

offer a richer organisational structure, such as the hierarchical grouping of characters into alphabets (Lake et al., 2015) or the factorial categorisation of rendered faces by identity, pose and lighting (Kulkarni et al., 2015).

Provided that such organisational structure is known in advance, we may generalise the training objective in Equation 14 to include a separate latent variable c_i for each group X_i within the dataset, even when these groups overlap. To do this we first rewrite this bound in its most general form, where $\mathbf{c}$ collects all latent variables:

$$\log p(\mathcal{X}) \geq \mathop{\mathbb{E}}_{Q(\mathbf{c};\mathbf{D})} \Big[\sum_{x \in X} \log p(x|\mathbf{c}) \Big] - \mathrm{D}_{KL}\big[Q(\mathbf{c};\mathbf{D}) \parallel P(\mathbf{c})\big] \tag{17}$$

As shown in Figure 2, a separate $D_i \subset X_i$ may be subsampled for inference of each latent variable c_i, so that $Q(\mathbf{c}) = \prod_i q_i(c_i; D_i)$. This leads to an analogous training objective (Equation 18), which may be applied to data with factorial or hierarchical category structure. For the hierarchical case, this objective may be further modified to infer layers sequentially, derived in Supplementary Material.

$$\log p(\mathcal{X}) \geq \sum_{x \in \mathcal{X}} \mathop{\mathbb{E}}_{\substack{D_i \subset X_i \\ \text{for each} \\ i : x \in X_i}} \Bigg[\mathop{\mathbb{E}}_{\substack{q_i(c_i;D_i) \\ \text{for each} \\ i : x \in X_i}} \log p(x|\mathbf{c}) - \sum_{i : x \in X_i} \frac{1}{|X_i|} \mathrm{D}_{KL}\big(q_i(c_i; D_i) \parallel p(c_i)\big) \Bigg] \tag{18}$$

3.3 POWERFUL DECODER MODELS

As evident in Equation 10, the VHE objective provides a formal motivation for KL rescaling in the variational objective (a common technique to increase use of latent variables in VAEs) by sharing these variables across many elements. This is of particular importance when using autoregressive decoder models, for which a common failure mode is to learn a decoder $p(x|z)$ with no dependence on the latent space, thus avoiding the encoding cost. In the context of VAEs, this particular issue has been discussed by Chen et al. (2016) who suggest crippling the decoder as a potential remedy.

The same failure mode can occur when training a VAE for sets, particularly when the sets D are of small size and thus have low total correlation. *Variational Homoencoders* suggest a potential remedy to this, encouraging use of the latent space by reusing the same latent variables across a large set X. This allows a VHE to learn useful representations even with $|D| = 1$, while at the same time utilising a powerful decoder to achieve highly accurate density estimation. In our experiments, we exploit the VHE's ability to use powerful decoders: specifically, we learn a generative model with a PixelCNN decoder, which is not possible with previous frameworks.

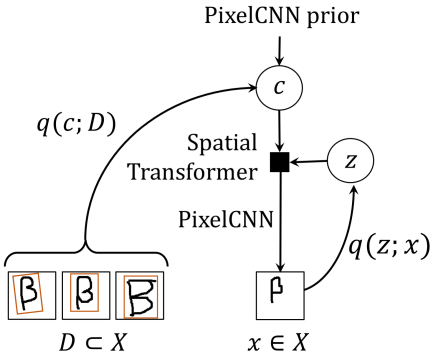

Figure 3: PixelCNN *VHE* architecture used for Omniglot and Youtube Faces. A spatial transformer network $q(c; D)$ encodes a subset D of a character class into a class latent variable c with the same width and height as the input image. A separate encoder $q(z; x)$, parametrized by a convolutional network, encodes position information in the target image into a latent variable z. A PixelCNN prior is used for c, and a Gaussian prior for z. During decoding, c and z are combined by a spatial transformer and used to condition a PixelCNN decoder network $p(x|\mathrm{STN}(c, z))$.

4 EXPERIMENTAL RESULTS

4.1 HANDWRITTEN CHARACTER CLASSES

To demonstrate that the VHE objective can facilitate learning with more expressive generative networks, we trained a variety of models on the Omniglot dataset exploring the interaction between model architecture and training objective. We consider two model architectures: a standard deconvolutional network based on Edwards & Storkey (2016), and a hierarchical PixelCNN architecture inspired by the PixelVAE (Gulrajani et al., 2016). For each, we compare models trained with the VHE objective against three alternative objectives.

For our hierarchical PixelCNN architecture (Figure 3) each character class is associated with a spatial latent variable c (a character 'template') with a PixelCNN prior, and each image x is associated with its own latent variable z (its 'position') with a Gaussian prior. To generate x, a Spatial Transformer Network (STN) (Jaderberg

Table 1: Comparison of VHE, Neural Statistician, and intermediate objectives with both deconvolutional and PixelCNN architectures. Rescaling encourages use of the latent space, while resampling encourages generalisation from the support set. The VHE is able to utilise the PixelCNN to achieve the highest classification accuracy.

	KL / nats*	Accuracy (5-shot)
Deconvolutional Architecture		
Neural Statistician [3]	31.34	95.6%
Resample (Eq 19)	25.74	94.0%
Rescale (Eq 20)	477.65	95.3%
VHE (resample + rescale, Eq 16)	452.47	95.6%
PixelCNN Architecture		
Neural Statistician	14.90	66.0%
Resample	0.22	4.9%
Rescale	506.48	62.8%
VHE (resample + rescale)	268.37	**98.8%**

*$\mathrm{D}_{KL}\big(q(c;D) \parallel p(c)\big)$, train set

et al., 2015) applies z to the class template c, and the result is input to a Gated PixelCNN $p(x|\mathrm{STN}(c,z))$ (Oord et al., 2016). The position encoder $q(z;x)$ is given by a CNN, and the class encoder $q(c;D)$ by an STN averaged over D. Both produce diagonal Gaussian distributions.

Using both PixelCNN and deconvolutional architectures, we trained models by several objectives. We compare a VHE model against a Neural Statistician baseline, with each trained on sampled subsets $D \subset X$ with $|D| = 5$ (as in Edwards & Storkey (2016)). Secondly, since the VHE introduces both data-resampling and KL-rescaling as modifications to this baseline, we separate the contributions of each using two intermediate objectives:

Resample only:

$$\underbrace{\mathop{\mathbb{E}}_{\substack{D\subset X\\ x\in X}}}_{\substack{\text{resample}\\ \text{decoded}\\ \text{element}}} \left[\mathop{\mathbb{E}}_{q(c;D)} \log p(x|c) - \frac{1}{|D|}\mathrm{D}_{KL}\big[q(c;D) \parallel p(c)\big] \right] \tag{19}$$

Rescale only:

$$\mathop{\mathbb{E}}_{\substack{D\subset X\\ x\in D}} \left[\mathop{\mathbb{E}}_{q(c;D)} \log p(x|c) - \underbrace{\frac{1}{|X|}}_{\text{rescale KL}} \mathrm{D}_{KL}\big[q(c;D) \parallel p(c)\big] \right] \tag{20}$$

Table 2: Comparison of classification accuracy with previous work. The VHE objective allows us to use a powerful decoder network, yielding state-of-the-art few-shot classification amongst deep generative models.

	Classification Accuracy (20-way) 1-shot	5-shot
Generative models, $\log p(X)$		
Generative Matching Networks [1]	77.0%	91.0%
Neural Statistician [3]	93.2%	98.1%
VHE	**95.2%**	**98.8%**
Discriminative models, $\log q(y;x,X,Y)$		
Matching Networks [21]	93.8%	98.7%
Convnet with memory module [9]	95.0%	98.6%
mAP-DLM [20]	95.4%	98.6%
Model-Agnostic Meta-learning [4]	95.8%	**98.9%**
Prototypical Networks [19]	**96.0%**	**98.9%**
(VHE, within-alphabet[1])	81.3%	90.3%

All models were trained on a random sample of 1200 Omniglot classes using images scaled to 28x28 pixels, dynamically binarised, and augmented by 8 rotations/reflections to produce new classes. We additionally used 20 small random affine transformations to create new instances within each class. Models were optimised using Adam (Kingma & Welling, 2013), and we used training error to select the best parameters from 5 independent training runs. We also implemented the 'sample dropout' trick of Edwards & Storkey (2016), but found that this had no effect on performance. At test time we classify an example x by Monte Carlo estimation of the expected conditional likelihood under the variational posterior $E_{q(c;D)}p(x|c)$, with 20 samples from $q(c;D)$. x is then classified to class with support set D that maximises this expected conditional likelihood.

Table 1 collects classification results of models trained using each of the four alternative training objectives, for both architectures. For a deconvolutional architecture, we find little difference in classification performance between all four training objectives, with the Neural Statistician and VHE models achieving equally high accuracy.

[1]The few-shot classification task defined by Lake et al. (2015) is to identify an image to one of 20 character classes, where *all 20 classes belong to the same (unseen) alphabet*. However, most work since has evaluated on an easier one-shot classification task, in which the 20 support characters are drawn from the entire test set (so are typically more dissimilar). We find that our model performs significantly worse on the within-alphabet variant, and so include results to facilitate future comparison on this more challenging task. Attaining near-human classification accuracy on this variant remains an open challenge for neural network models.

Figure 4: 5-shot samples generated by each model (more in Supplement). With a PixelCNN architecture, both Neural Statistician and Resample objectives lead to underutilisation of the latent space, producing unfaithful samples.

For the hierarchical PixelCNN architecture, however, significant differences arise between training objectives. In this case, a Neural Statistician learns a strong global distribution over images but makes only minimal use of latent variables c. This means that, despite the use of a higher capacity model, classification accuracy is much poorer (66%) than that achieved using a deconvolutional architecture. For the same reason, conditional samples display an improved sharpness but are no longer identifiable to the cue images on which they were conditioned (Figure 4). Our careful training suggests that this is not an optimisation difficulty but is core to the objective, as discussed in Chen et al. (2016).

By contrast, a VHE is able to gain a large benefit from the hierarchical PixelCNN architecture, with a 3-fold reduction in classification error (5-shot accuracy 98.8%) and conditional samples which are simultaneously sharp and identifiable (Figure 4). This improvement is in part achieved by increased utilisation of the latent space, due to rescaling of the KL divergence term in the objective. However, our results show that this common technique is insufficient when used alone, leading to overfitting to cue images with an equally severe impairment of classification performance (accuracy 62.8%). Rather, we find that KL-rescaling and data resampling must be used together in order for the benefit of the powerful PixelCNN architecture to be realised.

Table 2 lists the classification accuracy achieved by VHEs with both $|D| = 1$ and $|D| = 5$, as compared to existing deep learning approaches. We find that both networks are not only state-of-the-art amongst deep generative models, but also competitive against the best discriminative models trained directly for few-shot classification. Unlike these discriminative models, a VHE is also able to generate new images of a character in one shot, producing samples which are both realistic and faithful to the class of the cue image (Figure 5).

As our goal is to model shared structure across images, we evaluate generative performance using joint log likelihood of the entire Omniglot test set (rather than separately across images). From this perspective, a single element VAE will perform poorly as it treats all datapoints as independent, optimising a sum over log likelihoods for each element. By sharing latents across elements of the same class, a VHE can improve upon this considerably.

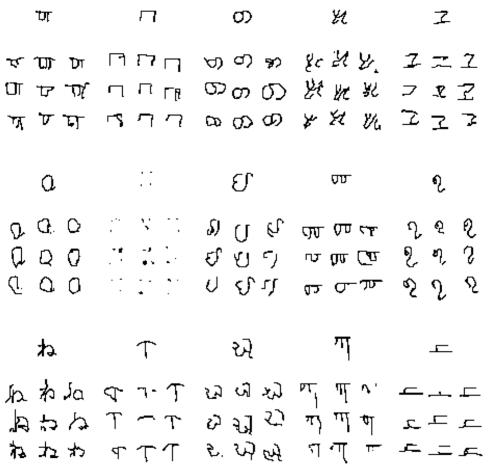

Figure 5: One-shot same-class samples generated by our model. Cue images were sampled from previously unseen classes.

For likelihood evaluation, our most appropriate comparison is with Generative Matching Networks (Bartunov & Vetrov, 2016) as they also model dependencies within a class. Thus, we trained models under the same train/test split as them, with no data augmentation. We evaluate the joint log likelihood of full character classes from the test set, normalised by the number of elements, using importance weighting with k=500 samples from $q(c; X)$. As can be seen in Tables 3 and 4, our hierarchical PixelCNN architecture is able to achieve state-of-the-art log likelihood results only when trained using the VHE objective.

Table 3: Joint NLL of Omniglot test set, compared across architectures and objectives.

	Test NLL per image
Deconvolutional Architecture	
NS [3]	102.84 nats
Resample	110.30 nats
Rescale	109.01 nats
VHE (resample + rescale)	104.67 nats
PixelCNN Architecture	
NS	73.50 nats
Resample	66.42 nats
Rescale	71.37 nats
VHE (resample + rescale)	**61.22 nats**

Table 4: Comparison of deep generative models by joint NLL of Omniglot test set.

	Test NLL per image
Independent models	$\frac{1}{n}\log \prod_i p(x_i)$
DRAW [5]	< 96.5 nats
Conv DRAW [6]	< 91.0 nats
VLAE [2]	89.83 nats
Conditional models	$\frac{1}{n}\log \prod_i p(x_i \mid x_{1:i-1})$
Generative Matching Networks [1]	62.42 nats[2]
Shared-latent models	$\frac{1}{n}\log \mathbb{E}_{p(c)} \prod_i p(x_i \mid c)$
Variational Homoencoder	**61.22 nats**

4.2 YOUTUBE FACES

To confirm that our approach can be used to produce naturalistic images, we compare VHE and Neural Statistician models trained on images from the YouTube Faces Database(Wolf et al., 2011), comprising 3,425 videos of 1,595 celebrities downloaded from YouTube. For our experiments, we use the aligned and cropped to face version, additionally cropping each image by 50% in both height and width, and rescale to 40x40 pixels. Our training, validation, and test sets consist of one video per person and 48 images per video. We use 954 videos for the training set and 641 videos for the test set.

We consider two architectures: the hierarchical PixelCNN network used for Omniglot experiments, and the deconvolution network used to model faces in Edwards & Storkey (2016). As above, we train each model using both VHE and NS objectives with $|D| = 5$.

Table 5: Classification results for YouTube faces dataset. The VHE PixelCNN utilises the latent space most effectively, and therefore achieves the highest few-shot classification accuracy and test image NLL.

	Test NLL per image	Accuracy (200-way) 1-shot	5-shot
Deconvolutional			
Neural Statistician	12512.4	39.2%	49.0%
VHE	12717.6	37.2%	44.8%
PixelCNN			
Neural Statistician	4229.8	92.1%	98.5%
VHE	**4091.3**	**92.5%**	**98.9%**

[2]We thank the authors of Bartunov & Vetrov (2016) for providing us with this comparison.

Classification results for trained models are shown in Table 5, and conditionally generated samples in Figure 6. As with Omniglot experiments, we find that the VHE objective improves use of the hidden layer c, leading to more accurate classification and conditional generation than the Neural Statistician. While the deconvolutional architecture is capable of producing realistic images (see Edwards & Storkey (2016)), our results show that it is not powerful enough to perform accurate few-shot classification. On the other hand, the PixelCNN architecture trained using the Neural Statistician objective achieves accurate few-shot classification, but generates poor images. The only network able to produce realistic images *and* perform accurate classification is the PixelCNN trained using our VHE objective.

4.3 MODELLING RICH CATEGORY STRUCTURE

To demonstrate how the VHE framework may apply to models with richer category structure, we built both a hierarchical and a factorial VHE (Figure 2) using simple modifications to the above architectures. For the hierarchical VHE, we extended the deconvolutional model with an extra latent layer a using the same encoder and decoder architecture as c. This was used to encode alphabet level structure for the Omniglot dataset, learning a generative model for alphabets of the form

$$p(\mathcal{A}) = \int p(a) \prod_{X_i \in \mathcal{A}} \int p(c_i|a) \prod_{x_{ij} \in X_i} p(x_{ij}|c_i, a) \mathrm{d}c_i \mathrm{d}a \tag{21}$$

Again, we trained this model using a single objective, using separately resampled subsets D^a and D^c to infer each latent variable (see Supplement). We then tested

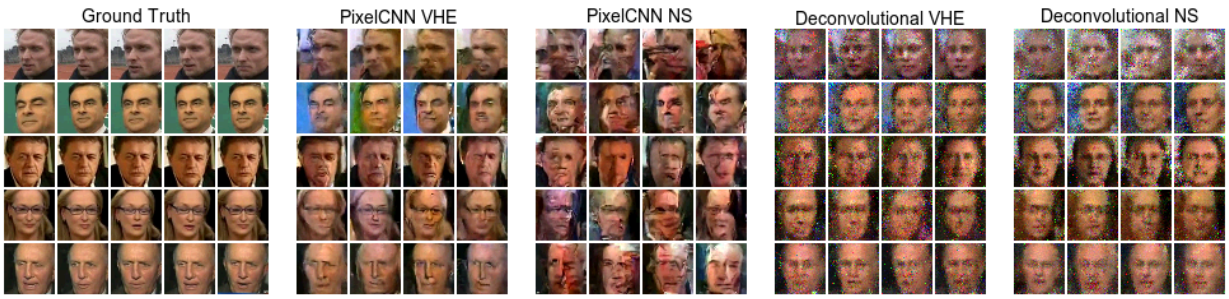

Figure 6: 5-shot samples of YouTube faces generated using both PixelCNN and deconvolutional architectures. Note that, for accurate comparison, we *sample* images from the decoder rather than taking the conditional mode as is common. For the deconvolutional models, this leads to images which appear more noisy than shown in previous work.

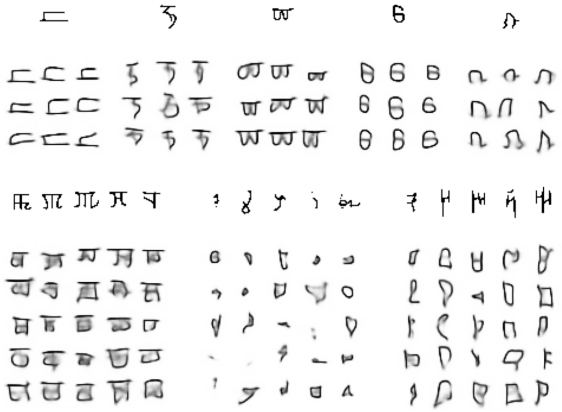

Figure 7: Conditional samples from character (top) and alphabet (bottom) levels of the same hierarchical model.

Figure 8: Previously unseen characters redrawn with both the colour and stroke width of a second character. For each group, the top two images denote the content (left) and style (right).

our model at both one-shot character generation and 5-shot alphabet generation, using samples from previously unseen alphabets. Our single trained model is able to learn structure at both layers of abstraction (Figure 7)

For the factorial VHE, we extended the Omniglot dataset by assigning each image to one of 30 randomly generated styles (independent of its character class), modifying both the colour and pen stroke characteristics of each image. We then extended the PixelCNN model to include a 6-dimensional latent variable s to represent the *style* of an image, alongside the existing c to represent the *character*. We used a CNN for style encoder $q(s; D^s)$, and for each image location we condition the PixelCNN decoder using the outer product $s \otimes c_{ij}$.

We then test this model on a *style transfer* task by feeding separate images into the character encoder $q(c; D^c)$ and style encoder $q(s; D^s)$, then rendering a new image from the inferred (c, s) pair. We find that synthesised samples are faithful to the respective character and style of both support images (Figure 8), demonstrating the ability of a factorial VHE to successfully disentangle these two image factors using separate latent variables.

5 CONCLUSION

We introduce the *Variational Homoencoder*: a hierarchical Bayesian approach to learning expressive generative models from few examples. We test the VHE by training a hierarchical PixelCNN on the Omniglot dataset, and achieve state-of-the-art results: our model is arguably the first which uses a general purpose architecture to both produce high quality samples and attain near state-of-the-art one-shot classification performance. We further validate our approach on a dataset of face images, and find that the VHE significantly improves the visual quality and classification accuracy achievable with a PixelCNN decoder. Finally, we show that the VHE framework extends naturally to models with richer latent structure, which we see as a promising direction for future work.

References

[1] Sergey Bartunov and Dmitry P Vetrov. Fast adaptation in generative models with generative matching networks. *arXiv preprint arXiv:1612.02192*, 2016.

[2] Xi Chen, Diederik P Kingma, Tim Salimans, Yan Duan, Prafulla Dhariwal, John Schulman, Ilya Sutskever, and Pieter Abbeel. Variational lossy autoencoder. *arXiv preprint arXiv:1611.02731*, 2016.

[3] Harrison Edwards and Amos Storkey. Towards a neural statistician. *arXiv preprint arXiv:1606.02185*, 2016.

[4] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400*, 2017.

[5] Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra. Draw: A recurrent neural network for image generation. *arXiv preprint arXiv:1502.04623*, 2015.

[6] Karol Gregor, Frederic Besse, Danilo Jimenez Rezende, Ivo Danihelka, and Daan Wierstra. Towards conceptual compression. In *Advances In Neural Information Processing Systems*, pp. 3549–3557, 2016.

[7] Ishaan Gulrajani, Kundan Kumar, Faruk Ahmed, Adrien Ali Taiga, Francesco Visin, David Vazquez, and Aaron Courville. Pixelvae: A latent variable model for natural images. *arXiv preprint arXiv:1611.05013*, 2016.

[8] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, pp. 2017–2025, 2015.

[9] Łukasz Kaiser, Ofir Nachum, Aurko Roy, and Samy Bengio. Learning to remember rare events. *arXiv preprint arXiv:1703.03129*, 2017.

[10] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[11] Gregory Koch. *Siamese neural networks for one-shot image recognition*. PhD thesis, University of Toronto, 2015.

[12] Tejas D Kulkarni, William F Whitney, Pushmeet Kohli, and Josh Tenenbaum. Deep convolutional inverse graphics network. In *Advances in Neural Information Processing Systems*, pp. 2539–2547, 2015.

[13] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350 (6266):1332–1338, 2015.

[14] Aaron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. Conditional image generation with pixelcnn decoders. *arXiv preprint arXiv:1606.05328*, 2016.

[15] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. 2016.

[16] Scott Reed, Yutian Chen, Thomas Paine, Aäron van den Oord, SM Eslami, Danilo Rezende, Oriol Vinyals, and Nando de Freitas. Few-shot autoregressive density estimation: Towards learning to learn distributions. *arXiv preprint arXiv:1710.10304*, 2017.

[17] Danilo Rezende, Ivo Danihelka, Karol Gregor, Daan Wierstra, et al. One-shot generalization in deep generative models. In *Proceedings of The 33rd International Conference on Machine Learning*, pp. 1521–1529, 2016.

[18] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. One-shot learning with memory-augmented neural networks. *arXiv preprint arXiv:1605.06065*, 2016.

[19] Jake Snell, Kevin Swersky, and Richard S Zemel. Prototypical networks for few-shot learning. *arXiv preprint arXiv:1703.05175*, 2017.

[20] Eleni Triantafillou, Richard Zemel, and Raquel Urtasun. Few-shot learning through an information retrieval lens. In *Advances in Neural Information Processing Systems*, pp. 2252–2262, 2017.

[21] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, pp. 3630–3638, 2016.

[22] Lior Wolf, Tal Hassner, and Itay Maoz. Face recognition in unconstrained videos with matched background similarity. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 529–534. IEEE, 2011.