One-Shot Generalization in Deep Generative Models

Danilo J. Rezende* DANILOR@GOOGLE.COM
Shakir Mohamed* SHAKIR@GOOGLE.COM
Ivo Danihelka DANIHELKA@GOOGLE.COM
Karol Gregor KAROLG@GOOGLE.COM
Daan Wierstra WIERSTRA@GOOGLE.COM
Google DeepMind, London

Abstract

Humans have an impressive ability to reason about new concepts and experiences from just a single example. In particular, humans have an ability for one-shot generalization: an ability to encounter a new concept, understand its structure, and then be able to generate compelling alternative variations of the concept. We develop machine learning systems with this important capacity by developing new deep generative models, models that combine the representational power of deep learning with the inferential power of Bayesian reasoning. We develop a class of sequential generative models that are built on the principles of feedback and attention. These two characteristics lead to generative models that are among the state-of-the art in density estimation and image generation. We demonstrate the one-shot generalization ability of our models using three tasks: unconditional sampling, generating new exemplars of a given concept, and generating new exemplars of a family of concepts. In all cases our models are able to generate compelling and diverse samples—having seen new examples just once—providing an important class of general-purpose models for one-shot machine learning.

1. Introduction

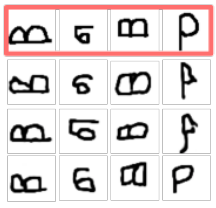


Figure 1. Given the first row, our model generates new exemplars.

Consider the images in the red box in figure 1. We see each of these new concepts just once, understand their structure, and are then able to imagine and generate compelling alternative variations of each concept, similar to those drawn in the rows beneath the red box. This is an ability that humans have for *one-shot generalization*: an ability to generalize to new concepts given just one or a few examples. In this paper, we develop new models that possess this capacity for one-shot generalization—models that allow for one-shot reasoning from the data streams we are likely to encounter in practice, that use only limited forms of domain-specific knowledge, and that can be applied to diverse sets of problems.

There are two notable approaches that incorporate one-shot generalization. Salakhutdinov et al. (2013) developed a probabilistic model that combines a deep Boltzmann machine with a hierarchical Dirichlet process to learn hierarchies of concept categories as well as provide a powerful generative model. Recently, Lake et al. (2015) presented a compelling demonstration of the ability of probabilistic models to perform one-shot generalization, using Bayesian program learning, which is able to learn a hierarchical, non-parametric generative model of handwritten characters. Their approach incorporates specific knowledge of how strokes are formed and the ways in which they are combined to produce characters of different types, exploiting similar strategies used by humans. Lake et al. (2015) see the capacity for one-shot generalization demonstrated by Bayesian programming learning 'as a challenge for neural models'. By combining the representational power of deep neural networks embedded within hierarchical latent variable models, with the inferential power of approximate Bayesian reasoning, we show that this is a challenge that can be overcome. The resulting deep generative models are general-purpose image models that are accurate and scalable, among the state-of-the-art, and possess the important capacity for one-shot generalization.

Deep generative models are a rich class of models for density estimation that specify a generative process for observed data using a hierarchy of latent variables. Models that are directed graphical models have risen in popularity and include discrete latent variable models such as sigmoid belief networks and deep auto-regressive networks (Saul et al., 1996; Gregor et al., 2014), or continuous latent variable models such as non-linear Gaussian belief networks and deep latent Gaussian models (Rezende et al.,

*Equal contributions. *Proceedings of the 33^{rd} International Conference on Machine Learning*, New York, NY, USA, 2016. JMLR: W&CP volume 48.

2014; Kingma & Welling, 2014). These models use deep networks in the specification of their conditional probability distributions to allow rich non-linear structure to be learned. Such models have been shown to have a number of desirable properties: inference of the latent variables allows us to provide a causal explanation for the data that can be used to explore its underlying factors of variation and for exploratory analysis; analogical reasoning between two related concepts, e.g., styles and identities of images, is naturally possible; any missing data can be imputed by treating them as additional latent variables, capturing the the full range of correlation between missing entries under any missingness pattern; these models embody minimum description length principles and can be used for compression; these models can be used to learn environment-simulators enabling a wide range of approaches for simulation-based planning.

Two principles are central to our approach: feedback and attention. These principles allow the models we develop to reflect the principles of *analysis-by-synthesis*, in which the analysis of observed information is continually integrated with constructed interpretations of it (Yuille & Kersten, 2006; Erdogan et al., 2015; Nair et al., 2008). Analysis is realized by attentional mechanisms that allow us to selectively process and route information from the observed data into the model. Interpretations of the data are then obtained by sets of latent variables that are inferred sequentially to evaluate the probability of the data. The aim of such a construction is to introduce internal feedback into the model that allows for a 'thinking time' during which information can be extracted from each data point more effectively, leading to improved inference, generation and generalization. We shall refer to such models as *sequential generative models*. Models such as DRAW (Gregor et al., 2015), composited variational auto-encoders (Huang & Murphy, 2015) and AIR (Eslami et al., 2016) are existing models in this class, and we will develop a general class of sequential generative models that incorporates these and other latent variable models and variational auto-encoders.

Our contributions are:

- We develop sequential generative models that provide a generalization of existing approaches, allowing for sequential generation and inference, multi-modal posterior approximations, and a rich new class of deep generative models.
- We demonstrate the clear improvement that the combination of attentional mechanisms in more powerful models and inference has in advancing the state-of-the-art in deep generative models.
- Importantly, we show that our generative models have the ability to perform one-shot generalization. We explore three generalization tasks and show that our models can imagine and generate compelling alternative variations of images after having seen them just once.

2. Varieties of Attention

Attending to parts of a scene, ignoring others, analyzing the parts that we focus on, and sequentially building up an interpretation and understanding of a scene: these are natural parts of human cognition. This is so successful a strategy for reasoning that it is now also an important part of many machine learning systems. This repeated process of attention and interpretation, analysis and synthesis, is an important component of the generative models we develop.

In its most general form, any mechanism that allows us to selectively route information from one part of our model to another can be regarded as an attentional mechanism. Attention allows for a wide range of invariances to be incorporated, with few additional parameters and low computational cost. Attention has been most widely used for classification tasks, having been shown to improve both scalability and generalization (Larochelle & Hinton, 2010; Chikkerur et al., 2010; Xu et al., 2015; Jaderberg et al., 2015; Mnih et al., 2014; Ba et al., 2015). The attention used in discriminative tasks is a 'reading' attention that transforms an image into a representation in a canonical coordinate space (that is typically lower dimensional), with the parameters controlling the attention learned by gradient descent. Attention in unsupervised learning is much more recent (Tang et al., 2014; Gregor et al., 2015). In latent variable models, we have two processes—inference and generation—that can both use attention, though in slightly different ways. The generative process makes use of a *writing or generative attention*, which implements a selective updating of the output variables, e.g., updating only a small part of the generated image. The inference process makes use of *reading attention*, like that used in classification. Although conceptually different, both these forms of attention can be implemented with the same computational tools. We focus on image modelling and make use of spatial attention. Two other types of attention, randomized and error-based, are discussed in appendix B.

Spatially-transformed attention. Rather than selecting a patch of an image (taking glimpses) as other methods do, a more powerful approach is to use a mechanism that provides invariance to shape and size of objects in the images (general affine transformations). Tang et al. (2014) take such an approach and use 2D similarity transforms to provide basic affine invariance. Spatial transformers (Jaderberg et al., 2015) are a more general method for providing such invariance, and is our preferred attentional mechanism. Spatial transformers (ST) process an input image $\mathbf{x}$ using parameters $\boldsymbol{\lambda}$ to generate an output:

$$\mathrm{ST}(\mathbf{x}, \boldsymbol{\lambda}) = [\kappa_h(\boldsymbol{\lambda}) \otimes \kappa_w(\boldsymbol{\lambda})] * \mathbf{x},$$

where κ_h and κ_w are 1-dimensional kernels, $\otimes$ indicates the tensor outer-product of the two kernels and $*$ indicates a convolution. Huang & Murphy (2015) develop occlusion-

aware generative models that make use of spatial transformers in this way. When used for reading attention, spatial transformers allow the model to observe the input image in a canonical form, providing the desired invariance. When used for writing attention, it allows the generative model to independently handle position, scale and rotation of parts of the generated image, as well as their content. An direct extension is to use multiple attention windows simultaneously (see appendix).

3. Iterative and Attentive Generative Models

3.1. Latent Variable Models and Variational Inference

Generative models with latent variables describe the probabilistic process by which an observed data point can be generated. The simplest formulations such as PCA and factor analysis use Gaussian latent variables $\mathbf{z}$ that are combined linearly to generate Gaussian distributed data points $\mathbf{x}$. In more complex models, the probabilistic description consists of a hierarchy of L layers of latent variables, where each layer depends on the layer above in a non-linear way (Rezende et al., 2014). For deep generative models, we specify this non-linear dependency using deep neural networks. To compute the marginal probability of the data, we must integrate over any unobserved variables:

$$p(\mathbf{x}) = \int p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z} \tag{1}$$

In deep latent Gaussian models, the prior distribution $p(\mathbf{z})$ is a Gaussian distribution and the likelihood function $p_\theta(\mathbf{x}|\mathbf{z})$ is any distribution that is appropriate for the observed data, such as a Gaussian, Bernoulli, categorical or other distribution, and that is dependent in a non-linear way on the latent variables. For most models, the marginal likelihood (1) is intractable and we must instead approximate it. One popular approximation technique is based on variational inference (Jordan et al., 1999), which transforms the difficult integration into an optimization problem that is typically more scalable and easier to solve. Using variational inference we can approximate the marginal likelihood by a lower bound, which is the objective function we use for optimization:

$$\mathcal{F} = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \mathrm{KL}[q_\phi(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})] \tag{2}$$

The objective function (2) is the negative free energy, which allows us to trade-off the reconstruction ability of the model (first term) against the complexity of the posterior distribution (second term). Variational inference approximates the true posterior distribution by a known family of approximating posteriors $q_\phi(\mathbf{z}|\mathbf{x})$ with variational parameters ϕ. Learning now involves optimization of the variational parameters ϕ and model parameters θ.

Instead of optimization by the variational EM algorithm, we take an amortized inference approach and represent the distribution $q(\mathbf{z}|\mathbf{x})$ as a recognition or inference model, which we also parameterize using a deep neural network. Inference models amortize the cost of posterior inference and makes it more efficient by allowing for generalization across the inference computations using a set of global variational parameters ϕ. In this framework, we can think of the generative model as a decoder of the latent variables, and the inference model as its inverse, an encoder of the observed data into the latent description. As a result, this specific combination of deep latent variable model (typically latent Gaussian) with variational inference that is implemented using an inference model is referred to as a variational auto-encoder (VAE). VAEs allow for a single computational graph to be constructed and straightforward gradient computations: when the latent variables are continuous, gradient estimators based on pathwise derivative estimators are used (Rezende et al., 2014; Kingma & Welling, 2014; Burda et al., 20) and when they are discrete, score function estimators are used (Mnih & Gregor, 2014; Ranganath et al., 2014; Mansimov et al., 2016).

3.2. Sequential Generative Models

The generative models as we have described them thus far can be characterized as single-step models, since they are models of i.i.d data that evaluate their likelihood functions by transforming the latent variables using a non-linear, feed-forward transformation. A *sequential generative model* is a natural extension of the latent variable models used in VAEs. Instead of generating the K latent variables of the model in one step, these models sequentially generate T groups of k latent variables ($K = kT$), i.e. using T computational steps to allow later groups of latent variables to depend on previously generated latent variables in a non-linear way.

3.2.1. Generative Model

In their most general form, sequential generative models describe the observed data over T time steps using a set of latent variables $\mathbf{z}_t$ at each step. The generative model is shown in the stochastic computational graph of figure 2(a), and described by:

$$\text{Latent variables} \quad \mathbf{z}_t \sim \mathcal{N}(\mathbf{z}_t|\mathbf{0},\mathbf{I}) \;\; t = 1,\ldots,T \tag{3}$$

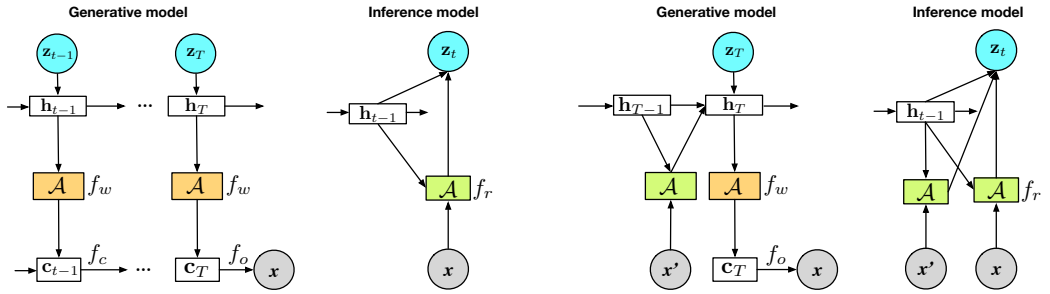
$$\text{Context} \quad \mathbf{v}_t = f_v(\mathbf{h}_{t-1}, \mathbf{x}'; \theta_v) \tag{4}$$

$$\text{Hidden state} \quad \mathbf{h}_t = f_h(\mathbf{h}_{t-1}, \mathbf{z}_t, \mathbf{v}_t; \theta_h) \tag{5}$$

$$\text{Hidden Canvas} \quad \mathbf{c}_t = f_c(\mathbf{c}_{t-1}, \mathbf{h}_t; \theta_c) \tag{6}$$

$$\text{Observation} \quad \mathbf{x} \sim p(\mathbf{x}|f_o(\mathbf{c}_T; \theta_o)) \tag{7}$$

Each step generates an independent set of K-dimensional latent variables $\mathbf{z}_t$ (equation (3)). If we wish to condition the model on an external context or piece of side-information $\mathbf{x}'$, then a deterministic function f_v (equation (4)) is used to read the context-images using an attentional mechanism. A deterministic transition function f_h introduces the sequential dependency between each of the latent variables, incorporating the context if it exists (equation (5)). This allows any transition mechanism to be used



(a) Unconditional generative model. (b) One-step of the conditional generative model.

Figure 2. Stochastic computational graph showing conditional probabilities and computational steps for sequential generative models. $\mathcal{A}$ represents an attentional mechanism that uses function f_w for writings and function f_r for reading.

and our transition is specified as a long short-term memory network (LSTM, Hochreiter & Schmidhuber (1997). We explicitly represent the creation of a set of hidden variables $\mathbf{c}_t$ that is a *hidden canvas* of the model (equation (6)). The canvas function f_c allows for many different transformations, and it is here where generative (writing) attention is used; we describe a number of choices for this function in section 3.2.3. The generated image (7) is sampled using an observation function $f_o(\mathbf{c}; \theta_o)$ that maps the last hidden canvas $\mathbf{c}_T$ to the parameters of the observation model. The set of all parameters of the generative model is $\theta = \{\theta_h, \theta_c, \theta_o\}$.

3.2.2. Free Energy Objective

Given the probabilistic model (3)-(7) we can obtain an objective function for inference and parameter learning using variational inference. By applying the variational principle, we obtain the free energy objective:

$$\begin{aligned} \log p(\mathbf{x}) &= \log \int p(\mathbf{x}|\mathbf{z}_{1:T}) p(\mathbf{z}_{1:T}) d\mathbf{z}_{1:T} \geq \mathcal{F} \\ \mathcal{F} &= \mathbb{E}_{q(\mathbf{z}_{1:T})}[\log p_\theta(\mathbf{x}|\mathbf{z}_{1:T})] \\ &- \textstyle\sum_{t=1}^{T} \mathrm{KL}[q_\phi(\mathbf{z}_t|\mathbf{z}_{<t}\mathbf{x}) \| p(\mathbf{z}_t)], \end{aligned} \tag{8}$$

where $\mathbf{z}_{<t}$ indicates the collection of all latent variables from step 1 to $t-1$. We can now optimize this objective function for the variational parameters ϕ and the model parameters θ, by stochastic gradient descent using a mini-batch of data. As with other VAEs, we use a single sample of the latent variables generated from $q_\phi(\mathbf{z}|\mathbf{x})$ when computing the Monte Carlo gradient. To complete our specification, we now specify the hidden-canvas functions f_c and the approximate posterior distribution $q_\phi(\mathbf{z}_t)$.

3.2.3. Hidden Canvas functions

The canvas transition function $f_c(\mathbf{c}_{t-1}, \mathbf{h}_t; \theta_c)$ (6) updates the hidden canvas by first non-linearly transforming the current hidden state of the LSTM $\mathbf{h}_t$ (using a function f_w) and fuses the result with the existing canvas $\mathbf{c}_{t-1}$. In this work we use hidden canvases that have the same size as the original images, though they could be either *larger or smaller* in size and can have any number of channels (four in this paper). We consider two ways with which to update the hidden canvas:

Additive Canvas. As the name implies, an additive canvas updates the canvas by simply adding a transformation of the hidden state $f_w(\mathbf{h}_t; \theta_c)$ to the previous canvas state $\mathbf{c}_{t-1}$. This is a simple, yet effective (see results) update rule:

$$f_c(\mathbf{c}_{t-1}, \mathbf{h}_t; \theta_c) = \mathbf{c}_{t-1} + f_w(\mathbf{h}_t; \theta_c), \tag{9}$$

Gated Recurrent Canvas. The canvas function can be updated using a convolutional gated recurrent unit (CGRU) architecture (Kaiser & Sutskever, 2015), which provides a non-linear and recursive updating mechanism for the canvas and are simplified versions of convolutional LSTMs (further details of the CGRU are given in appendix B). The canvas update is:

$$f_c(\mathbf{c}_{t-1}, \mathbf{h}_t; \theta_c) = \mathrm{CGRU}(\mathbf{c}_{t-1} + f_w(\mathbf{h}_t; \theta_c)) \tag{10}$$

In both cases, the function $f_w(\mathbf{h}_t; \theta_w)$ is a *writing* or *generative attention* function, that we implement as a spatial transformer; inputs to the spatial transformer are its affine parameters and a 10×10 image to be transformed, both of which are provided by the LSTM output.

The final phase of the generative process transforms the hidden canvas at the last time step $\mathbf{c}_T$ into the parameters of the likelihood function using the output function $f_o(\mathbf{c}; \theta_o)$. Since we use a hidden canvas that is the same size as the original images but that have a different number of filters, we implement the output function as a 1×1 convolution. When the hidden canvas has a different size, a convolutional network is used instead.

3.2.4. Dependent Posterior Inference

We use a structured posterior approximation that has an auto-regressive form, i.e. $q(\mathbf{z}_t|\mathbf{z}_{<t}, \mathbf{x})$. We implement this distribution as an inference network parameterized by a deep network. The specific form we use is:

$$\text{Sprite} \qquad \mathbf{r}_t = f_r(\mathbf{x}, \mathbf{h}_{t-1}; \phi_r) \tag{11}$$

$$\text{Sample} \quad \mathbf{z}_t \sim \mathcal{N}(\mathbf{z}_t|\boldsymbol{\mu}(\mathbf{s}_t, \mathbf{h}_{t-1}; \phi_\mu), \sigma(\mathbf{r}_t, \mathbf{h}_{t-1}; \phi_\sigma)) \tag{12}$$

At every step of computation, we form a low-dimensional representation $\mathbf{r}_t$ of the input image using a non-linear transformation f_r of the input image and the hidden state of the model.This function is *reading* or *recognition attention* using a spatial transformer, whose affine parameters are given by the LSTM output. The result of reading is a sprite $\mathbf{r}_t$ that is then combined with the previous state $\mathbf{h}_{t-1}$ through a further non-linear function to produce the mean $\boldsymbol{\mu}_t$ and variance $\boldsymbol{\sigma}_t$ of a K-dimensional diagonal Gaussian distribution. We denote all the parameters of the inference model by $\phi = \{\phi_r, \phi_\mu, \phi_\sigma\}$. Although the conditional distributions $q(\mathbf{z}_t|\mathbf{z}_{<t})$ are Gaussian, the joint posterior posterior $p(\mathbf{z}_{1:T}) = \prod_t p(\mathbf{z}_t|\mathbf{z}_{<t})$ is non-Gaussian and multimodal due to the non-linearities used, enabling more accurate inference.

3.2.5. Model Properties and Complexity

The above sequential generative model and inference is a generalization of existing models such as DRAW (Gregor et al., 2015) , composited VAEs (Huang & Murphy, 2015) and AIR (Eslami et al., 2016). This generalization has a number of differences and important properties. One of the largest deviations is the introduction of the hidden canvas into the generative model that provides an important richness to the model, since it allows a pre-image to be constructed in a hidden space before a final corrective transformation, using the function f_o, is used. The generative process has an important property that allows the model be sampled without feeding-back the results of the canvas $\mathbf{c}_t$ to the hidden state $\mathbf{h}_t$—such a connection is not needed and provides more efficiency by reducing the number of model parameters. The inference network in our framework is also similarly simplified. We do not use a separate recurrent function within the inference network (like DRAW), but instead share parameters of the LSTM from the prior—the removal of this additional recursive function has no effect on performance.

Another important difference between our framework and existing frameworks is the type of attention that is used. Gregor et al. (2015) use a generative attention based on Gaussian convolutions parameterized by a location and scale, and Tang et al. (2014) use 2D similarity transformations. We use a much more powerful and general attention mechanism based on spatial transformers (Jaderberg et al., 2015; Huang & Murphy, 2015).

The overall complexity of the algorithm described matches the typical complexity of widely-used methods in deep learning. For images of size $I \times I$, the spatial transformer has a complexity that is linear in the number of pixels of the attention window. For a $J \times J$ attention window, with $J \leq I$, the spatial transformer has a complexity of $O(NTJ^2)$, for T sequential steps and N data points. All other components have the standard quadratic complexity in the layer size, hence for L layers with average size D, this gives a complexity of $O(NLD^2)$.

4. Image Generation and Analysis

We first show that our models are state-of-the-art, obtaining highly competitive likelihoods, and are able to generate high-quality samples across a wide range of data sets with different characteristics.

For all our experiments, our data consists of binary images and we use use a Bernoulli likelihood to model the probability of the pixels. In all models we use 400 LSTM hidden units. We use 12×12 kernels for the spatial transformer, whether used for recognition or generative attention. The latent variable $\mathbf{z}_t$ are 4-dimensional Gaussian distributions and we use a number of steps that vary from 20-80. The hidden canvas has dimensions that are the size of the images with four channels. We present the main results here and any additional results in Appendix A. All the models were trained for approximatively 800K iterations with mini-batches of size 24. The reported likelihood bounds for the training set are computed by averaging the last 1K iterations during training. The reported likelihood bounds for the test set were computed by averaging the bound for $24,000$ random samples (sampled with replacement) and the error bars are the standard-deviations of the mean.

4.1. MNIST and Multi-MNIST

We highlight the behaviour of the models using two data sets based on the MNIST benchmark. The first experiment uses the binarized MNIST data set of Salakhutdinov & Murray (2008), that consists of 28×28 binary images with 50,000 training and 10,000 test images. Table 1 compares the log-likelihoods on this binarized MNIST data set using existing models, as well as the models developed in this paper (with variances of our estimates in parentheses). The sequential generative model that uses the spatially-transformed attention with the CGRU hidden canvas provides the best performance among existing work on this data set. We show samples from the model in figure 3.

We form a multi-MNIST data set of 64×64 images that consists of two MNIST digits placed at random locations in the image (having adapted the cluttered MNIST generator from Mnih et al. (2014) to procedurally generate the data). We compare the performance in table 2 and show samples from this model in figure 3. This data set is much harder than MNIST to learn, with much slower convergence. The additive canvas with spatially-transformed attention provides a reliable way to learn from this data.

Importance of each step

These results also indicate that longer sequences can lead to better performance. Every step taken by the model adds a term to the objective function (2) corresponding to the KL-divergence between the prior distribution and the con-

Table 1. Test set negative log-likelihood on MNIST.

Model	Test NLL
From Gregor et al. (2015) and Burda et al. (20)	
DBM 2hl	≈84.62
DBN 2hl	≈84.55
NADE	88.33
DLGM-VAE	≈ 86.60
VAE + HVI/Norm Flow	≈ 85.10
DARN	≈ 84.13
DRAW (64 steps, no attention)	≤ 87.40
DRAW (64 steps, Gaussian attention)	≤ 80.97
IWAE (2 layers; 50 particles)	≈ 82.90

Sequential generative models

Attention	Canvas	Steps	Train	Test NLL
Spatial tr.	CGRU	80	78.5	≤**80.5(0.3)**
Spatial tr.	Additive	80	80.1	≤81.6(0.4)
Spatial tr.	CGRU	30	80.1	≤81.5(0.4)
Spatial tr.	Additive	30	79.1	≤82.6(0.5)
Fully conn.	CGRU	80	80.0	≤98.7(0.8)

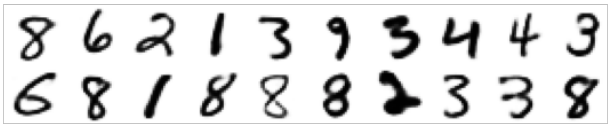


Figure 3. Generated samples for MNIST. For a video of the generation process, see https://youtu.be/ptLdYd8FXRA

tribution to the approximate posterior distribution at that step. Figure 4 shows the KL-divergence for each iteration for two models on MNIST up to 20 steps. The KL-divergence decays towards the end of the sequence, indicating that the latent variables $\mathbf{z}_t$ have diminishing contribution to the model as the number of steps grow. Unlike VAEs where we often find that there are many dimensions which contribute little to the likelihood bound, the sequential property allows us to more efficiently allocate and decide on the number of latent variables to use and means of deciding when to terminate the sequential computation.

4.2. Omniglot

Unlike MNIST, which has a small number of classes with many images of each class and a large amount of data, the omniglot data set (Lake et al., 2015) consists of 105×105 binary images across 1628 classes with just 20 images per class. This data set allows us to demonstrate that attentional mechanisms and better generative models allow us to perform well even in regimes with larger images and limited amounts of data.

There are two versions of the omniglot data that have been previously used for the evaluation of generative models. One data set used by Burda et al. (20) consists of 28×28 images, but is different to that of Lake et al. (2015). We compare the available methods on the dataset from Burda et al. (20) in table 3 and find that the sequential models perform better than all competing approaches, further establishing the effectiveness of these models. Our second evaluation uses the dataset of Lake et al. (2015), which we downsampled to 52×52 using a 2×2 max-pooling. We compare different sequential models in table 4 and again find that spatially-transformed attention is a powerful general purpose attention and that the additive hidden canvas performs best.

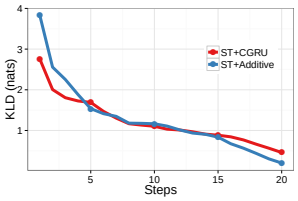


Figure 4. Per-step KL contribution on MNIST.

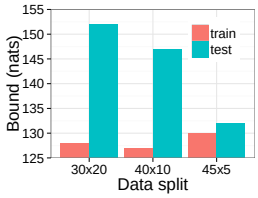


Figure 5. Gap between train and test bound on omniglot.

Table 2. Train and test NLL bounds on 64×64 Multi-MNIST.

Att	CT	Steps	Train	Test
Multi-ST	Additive	80	**177.2**	**176.9(0.5)**
Spatial tr.	Additive	80	183.0	182.0(0.6)
Spatial tr.	CGRU	80	196.0	194.9(0.5)
Fully conn.	CGRU	80	272.0	270.3(0.8)

4.3. Multi-PIE

The Multi-PIE dataset (Gross et al., 2010) consists of 48×48 RGB face images from various viewpoints. We have converted the images to grayscale and trained our model on a subset comprising of all 15-viewpoints but only 3 out of the 19 illumination conditions. Our simplification results in $93,130$ training samples and $10,000$ test samples. Samples from this model are shown in figure 7 and are highly compelling, showing faces in different orientations, different genders and are representative of the data. The model was trained using the logit-normal likelihood as in Rezende & Mohamed (2015).

5. One-Shot Generalization

Lake et al. (2015) introduce three tasks to evaluate one-shot generalization, testing weaker to stronger forms of generalization. The three tasks are: (1) unconditional (free) generation, (2) generation of novel variations of a given exemplar, and (3) generation of representative samples from a novel alphabet. Lake et al. (2015) conduct human evaluations as part of their assessment, which is important in contrasting the performance of models against the cognitive ability of humans; we do not conduct human benchmarks in this paper (human evaluation will form part of our follow-up work). Our focus is on the machine learning of one-shot generalization and the computational challenges associated with this task.

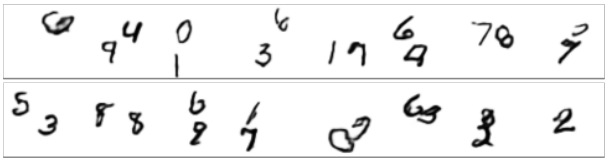


Figure 6. Generated samples for multi-MNIST. For a video of the generation process, see https://www.youtube.com/watch?v=HkDxmnIfWIM

Table 3. NLL on the 28×28 omniglot data.

Model	Test NLL
From Burda et al. (20)	
VAE (2 layer, 5 samples)	106.31
IWAE (2 layer, 50 samples)	103.38
RBM (500 hidden)	100.46
Seq Gen Model (20 steps, ST, additive)	$\leq$**96.5**
Seq Gen Model (80 steps, ST, additive)	$\leq$**95.5**

1. Unconditional Generation.
This is the same generation task reported for the data sets in the previous section. Figure 8 shows samples that reflect the characteristics of the omniglot data, showing a variety of styles including rounded patterns, line segments, thick and thin strokes that are representative of the data set. The likelihoods reported in tables 3 and 4 quantitatively establish this model as among the state-of-the-art.

2. Novel variations of a given exemplar.
This task corresponds to figure 5 in Lake et al. (2015)). At test time, the model is presented with a character of a type it has never seen before (was not part of its training set), and asked to *generate novel variations of this character*. To do this, we use a conditional generative model (figure 2(b), equation (4)). The context $\mathbf{x}'$ is the image that we wish the model to generate new exemplars of. To expose the boundaries of our approach, we test this under weak and strong one-shot generalization tests:

a) We use a data set whose training data consists of all available alphabets, but for which three character types from each alphabet have been removed to form the test set (3000 characters). This is a weak one-shot generalization test where, although the model has never seen the test set characters, it has seen related characters from the same alphabet and is expected to transfer that knowledge to this generation task.

b) We use exactly the data split used by Lake et al. (2015), which consists of 30 alphabets as the training set and the remaining 20 alphabets as the test set. This is a strong one-shot generalization test, since the model has seen neither the test character nor any alphabets from its family. This is a hard test for our model, since this split provides limited training data, making overfitting easier, and generalization harder.

c) We use two alternative training-test split of the data, a 40-10 and 45-5 split. We can examine the spectrum of difficulty of the previous one-shot generalization task by considering these alternative splits.

Table 4. Train and test NLL bounds on 52×52 omniglot

Att	CT	Steps	Train	Test
Multi-ST	CGRU	80	**120.6**	**134.1(0.5)**
Spatial tr.	Additive	40	128.7	136.1(0.4)
Spatial tr.	Additive	80	134.6	141.5(0.5)
Spatial tr.	CGRU	80	141.6	144.5(0.4)
Fully conn.	CGRU	80	170.0	351.5(1.2)

Figure 7. Generated samples for Multi-PIE using the model with Spatial Transformer + additive canvas (32 steps). For a video of the generation process including the boundaries of the writing attention grid, see https://www.youtube.com/watch?v=6S6Tx_OtvnA

We show the model's performance on the weak generalization test in figure 9, where the first row shows the exemplar image, and the subsequent rows show the variations of that image generated by the model. We show generations for the strong generalization test in figure 10. Our model also generates visually similar and reasonable variations of the image in this case. Unlike the model of Lake et al. (2015), which uses human stroke information and a model structured around the way in which humans draw images, our model is applicable to any image data, with the only domain specific information that is used being that the data is spatially arranged (which is exploited by the convolution and attention). This test also exposes the difficulty that the model has in coping with small amounts of data. We compare the difference between train and test log-likelihoods for the various data splits in figure 5. We see that there is a small gap between the training and test likelihoods in the regime where we have more data (45-5 split) indicating no overfitting. There is a large gap for the other splits, hence a greater tendency for overfitting in the low data regime. An interesting observation is that even for the cases where there is a large gap between train and test likelihood bounds (figure 5), the examples generated by the model (figure 10, left and middle) still generalize to unseen character classes. Data-efficiency is an important challenge for the large parametric models that we use and one we hope to address in future.

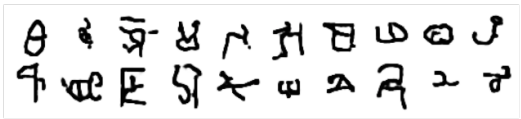


Figure 8. Unconditional samples for 52×52 omniglot (task 1). For a video of the generation process, see https://www.youtube.com/watch?v=HQEI2xfTgm4

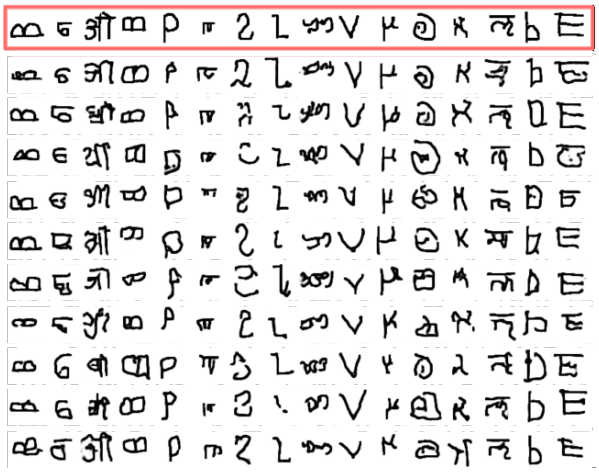


Figure 9. Generating new examplars of a given character for the weak generalization test (task 2*a*). The first row shows the test images and the next 10 are one-shot samples from the model.

3. Representative samples from a novel alphabet. This task corresponds to figure 7 in Lake et al. (2015), and conditions the model on anywhere between 1 to 10 samples of a novel alphabet and asks the model to *generate new characters consistent with this novel alphabet*. We show here the hardest form of this test, using only 1 context image. This test is highly subjective, but the model generations in figure 11 show that it is able to pick up common features and use them in the generations.

We have emphasized the usefulness of deep generative models as scalable, general-purpose tools for probabilistic reasoning that have the important property of one-shot generalization. But, these models do have limitations. We have already pointed to the need for reasonable amounts of data. Another important consideration is that, while our models can perform one-shot generalization, they do not perform one-shot *learning*. One-shot learning requires that a model is updated after the presentation of each new input, e.g., like the non-parametric models used by Lake et al. (2015) or Salakhutdinov et al. (2013). Parametric models such as ours require a gradient update of the parameters, which we do not do. Instead, our model performs a type of one-shot inference that during test time can perform inferential tasks on new data points, such as missing data completion, new exemplar generation, or analogical sampling, but does not learn from these points. This distinction between one-shot learning and inference is important; We aim to extend our approach to the online and one-shot learning setting in future.

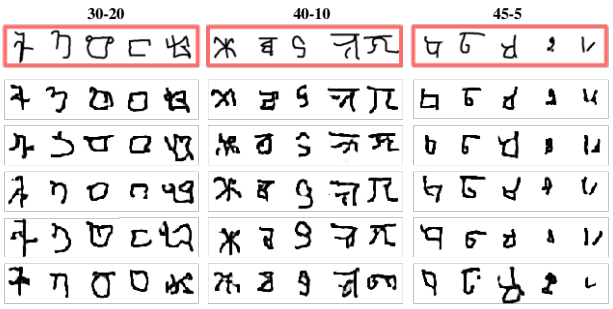


Figure 10. Generating new examplars of a given character for the strong generalization test (task 2*b,c*), with models trained with different amounts of data. Left: Samples from model trained on 30-20 train-test split; Middle: 40-10 split; Right: 45-5 split (right)

Figure 11. Generating new exemplars from a novel alphabet (task 3). The first row shows the test images, and the next 10 rows are one-shot samples generated by the model.

6. Conclusion

We have developed a new class of general-purpose models that have the ability to perform one-shot generalization, emulating an important characteristic of human cognition. Sequential generative models are natural extensions of variational auto-encoders and provide state-of-the-art models for deep density estimation and image generation. The models specify a sequential process over groups of latent variables that allows it to compute the probability of data points over a number of steps, using the principles of feedback and attention. The use of spatial attention mechanisms substantially improves the ability of the model to generalize. The spatial transformer is a highly flexible attention mechanism for both reading and writing, and is now our default mechanism for attention in generative models. We highlighted the one-shot generalization ability of the model over a range of tasks that showed that the model is able to generate compelling and diverse samples, having seen new examples just once. However there are limitations of this approach, e.g., still needing a reasonable amount of data to avoid overfitting, which we hope to address in future work.

Acknowledgements

We thank Brenden Lake and Josh Tenenbaum for insightful discussions. We are grateful to Theophane Weber, Ali Eslami, Peter Battaglia and David Barrett for their valuable feedback.

References

Ba, J., Salakhutdinov, R., Grosse, R. B., and Frey, B. J. Learning wake-sleep recurrent attention models. In *NIPS*, pp. 2575–2583, 2015.

Burda, Y., Grosse, R., and Salakhutdinov, R. Importance weighted autoencoders. *ICLR*, 20.

Chikkerur, S., Serre, T., Tan, C., and Poggio, T. What and where: A Bayesian inference theory of attention. *Vision research*, 50(22):2233–2247, 2010.

Erdogan, G., Yildirim, I., and Jacobs, R. A. An analysis-by-synthesis approach to multisensory object shape perception. In *NIPS*, 2015.

Eslami, S. M., Heess, N., Weber, T., Tassa, Y., Kavukcuoglu, K., and Hinton, G. E. Attend, Infer, Repeat: Fast scene understanding with generative models. *arXiv preprint arXiv:1603.08575*, 2016.

Gregor, K., Danihelka, I., Mnih, A., Blundell, C., and Wierstra, D. Deep autoregressive networks. In *ICML*, 2014.

Gregor, K., Danihelka, I., Graves, A., Rezende, D. J., and Wierstra, D. DRAW: A recurrent neural network for image generation. In *ICML*, 2015.

Gross, R, Matthews, I, Cohn, J, Kanade, T, and Baker, S. Multi-pie. *Image and Vision Computing*, 28(5):807–813, 2010.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Huang, J. and Murphy, K. Efficient inference in occlusion-aware generative models of images. *arXiv preprint arXiv:1511.06362*, 2015.

Jaderberg, M., Simonyan, K., Zisserman, A., and Kavukcuoglu, K. Spatial transformer networks. In *NIPS*, 2015.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

Kaiser, L. and Sutskever, I. Neural GPUs learn algorithms. *arXiv preprint arXiv:1511.08228*, 2015.

Kingma, D. P. and Welling, M. Auto-encoding variational Bayes. In *ICLR*, 2014.

Lake, B. M., Salakhutdinov, R., and Tenenbaum, J. B. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

Larochelle, H. and Hinton, G. E. Learning to combine foveal glimpses with a third-order boltzmann machine. In *NIPS*, pp. 1243–1251, 2010.

Mansimov, E., Parisotto, E., Ba, J. L., and Salakhutdinov, R. Generating images from captions with attention. *ICLR*, 2016.

Mnih, A. and Gregor, K. Neural variational inference and learning in belief networks. In *ICML*, 2014.

Mnih, V., Heess, N., Graves, A., and Kavukcuoglu, K. Recurrent models of visual attention. In *NIPS*, pp. 2204–2212, 2014.

Nair, V., Susskind, J., and Hinton, G. E. Analysis-by-synthesis by learning to invert generative black boxes. In *ICANN*. 2008.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, 2011.

Ranganath, R., Gerrish, S., and Blei, D. M. Black box variational inference. In *AISTATS*, 2014.

Rezende, D. J. and Mohamed, S. Variational inference with normalizing flows. *ICML*, 2015.

Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. In *ICML*, 2014.

Salakhutdinov, R. and Murray, I. On the quantitative analysis of deep belief networks. In *ICML*, pp. 872–879, 2008.

Salakhutdinov, R., Tenenbaum, J. B., and Torralba, A. Learning with hierarchical-deep models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35 (8):1958–1971, 2013.

Saul, L. K., Jaakkola, T., and Jordan, M. I. Mean field theory for sigmoid belief networks. *Journal of artificial intelligence research*, 4(1):61–76, 1996.

Tang, Y., Srivastava, N., and Salakhutdinov, R. Learning generative models with visual attention. In *NIPS*, pp. 1808–1816, 2014.

Xu, K., Ba, J., Kiros, R., Courville, A., Salakhutdinov, R., Zemel, R., and Bengio, Y. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015.

Yuille, A. and Kersten, D. Vision as Bayesian inference: analysis by synthesis? *Trends in cognitive sciences*, 10 (7):301–308, 2006.