Learning generative models with visual attention

Yichuan Tang\textsuperscript{1}, Nitish Srivastava\textsuperscript{1}, and Ruslan Salakhutdinov\textsuperscript{2}

\{tang.nitish,rsalakhu\}@cs.toronto.edu

\textsuperscript{1}Department of Computer Science, University of Toronto, Toronto, Ontario, Canada.
\textsuperscript{2}Department of Computer Science and Statistics, University of Toronto, Ontario, Canada.

Abstract

Attention has long been proposed by psychologists as important for effectively dealing with the enormous sensory stimulus available in the neocortex. Inspired by the visual attention models in computational neuroscience and the need of object-centric data for generative models, we describe for generative learning framework using attentional mechanisms. Attentional mechanisms can propagate signals from region of interest in a scene to an aligned canonical representation, where generative modeling takes place. By ignoring background clutter, generative models can concentrate their resources on the object of interest. Our model is a proper graphical model where the 2D Similarity transformation is a part of the top-down process. A ConvNet is employed to provide good initializations during posterior inference which is based on Hamiltonian Monte Carlo. Upon learning images of faces, our model can robustly attend to face regions of novel test subjects. More importantly, our model can learn generative models of new faces from a novel dataset of large images where the face locations are not known.

1 Introduction

Generative models of objects and images are interesting because learning them often require finding useful latent representations. These models often assign an unnormalized log probability density over the input stimulus, with the goal of learning high level features or codes useful for discrimination without explicit supervisory signals. In vision, generative models include the Boltzmann Machine family of models: Restricted Boltzmann Machines \cite{1}, Deep Belief Nets \cite{2}, Deep Boltzmann Machines \cite{3}. Mixture models such as Mixtures of Factor Analyzers \cite{4} and Mixtures of Gaussians have also been used in modeling natural image patches \cite{5}. More recently denoising auto-encoders have been proposed as being able to model the transition operator that has the same invariant distribution as the data generating distribution \cite{6}.

Generative models have advantages over discriminative models when part of the images are occluded or missing. Occlusions are very common in realistic settings and have been largely ignored in recent works in deep learning. In addition, prior knowledge can be easily incorporated in generative models in the forms of structured latent variables, such as lighting and deformable parts. However, as a picture is worth a thousand words, the enormous amount of content in full images makes generative learning difficult \cite{7,8}. Therefore, generative models have found most success in learning small patches of natural images and objects: Zoran and Weiss \cite{5} learned mixture of Gaussians over $8 \times 8$ image patches; Salakhutdinov and Hinton \cite{3} used $64 \times 64$ centered and uncluttered stereo images of toy objects on a clear background; Tang et al. \cite{9} used $24 \times 24$ images of centered and cropped faces. The fact that these models require curated training data limits their applicability on using the (virtually) unlimited unlabeled data.

In this paper, we propose a framework to infer in a big image the region of interest for generative modeling. This would allow us to learn a generative model of faces on a very large dataset of
distributions are defined as: the visible nodes from the joint distribution. Its energy function and corresponding conditional of hidden nodes, and the mixing proportions of the components are defined by marginalizing out with shared parameters, where the number of mixture components is exponential in the number recognition [7, 20, 21, 22]. The Gaussian RBM can be viewed as a mixture of diagonal Gaussians successfully applied to tasks including image classification, video action recognition, and speech To model the real-valued data, we can use the Gaussian RBM (GRBM) [19]. GRBMs have been compute.

Directed models, an RBM’s conditional distribution over hidden nodes is factorial and very easy to

where

\[ Z = \sum_{\{v, h\}} \exp(-E(v, h)) \]

is the normalization constant. Unlike directed models, an RBM’s conditional distribution over hidden nodes is factorial and very easy to compute.

To model the real-valued data, we can use the Gaussian RBM (GRBM) [19]. GRBMs have been successfully applied to tasks including image classification, video action recognition, and speech recognition [7, 20, 21, 22]. The Gaussian RBM can be viewed as a mixture of diagonal Gaussians with shared parameters, where the number of mixture components is exponential in the number of hidden nodes, and the mixing proportions of the components are defined by marginalizing out the visible nodes from the joint distribution. Its energy function and corresponding conditional distributions are defined as:

\[ E_{GRBM}(v, h; \theta) = - \sum_i b_i v_i - \sum_j c_j h_j - \sum_{i,j} W_{ij} v_i h_j, \]

where \( \theta = \{W, b, c\} \) are the model parameters. The probability distribution of the configuration \( \{v, h\} \) is: \( p(v, h; \theta) = \frac{p^*(v, h)}{Z(\theta)} = \frac{\exp(-E(v, h))}{Z(\theta)} \), where we have used \( p^*(\cdot) \) to represent the unnormalized probability distribution and \( Z(\theta) = \sum_{v, h} \exp(-E(v, h)) \) is the normalization constant. Unlike

Attention as a form of routing was originally proposed by Anderson and Essen [13] and then extended by Olshausen et al. [14]. Dynamic routing has also been hypothesized as providing a way for achieving shift and size invariance in the visual cortex [15]. Tsotsos et al. [16] proposed a model combining search and attention called Selective Tuning model. Larochelle and Hinton [17] proposed a way of using third order Boltzmann Machines to combine information gathered from many foveal glimpses. Their model chooses where to look next and how to compile the information to try to find locations which are most predictive for the class of the object. Reichert et al. [18] proposed a hierarchical model to show that certain aspects of object-based attention can be modeled by Deep Boltzmann Machines.

Inspired by Olshausen et al. [14], we use 2D similarity transformations from computer vision to implement the scaling, rotation, and shift operation required for routing. Our main motivation is to enable the learning of generative models in big images where the location of the object of interest is not unknown.

2 Restricted Boltzmann Machines

Before we describe our model, we briefly review Restricted Boltzmann Machines. A Restricted Boltzmann Machine (RBM) is a type of Markov Random Field, or an undirected graphical model that has a bipartite structure with two sets of binary stochastic nodes: the visible \( v \in \{0, 1\}^{N_v} \) and hidden \( h \in \{0, 1\}^{N_h} \) nodes [11]. The RBM has visible to hidden connections but no intra-layer connections. For any configuration of the nodes, we can define an energy function as:

\[ E_{RBM}(v, h; \theta) = - \sum_i b_i v_i - \sum_j c_j h_j - \sum_{i,j} W_{ij} v_i h_j, \]

where \( \theta = \{W, b, c\} \) are the model parameters. The probability distribution of the configuration \( \{v, h\} \) is: \( p(v, h; \theta) = \frac{p^*(v, h)}{Z(\theta)} = \frac{\exp(-E(v, h))}{Z(\theta)} \), where we have used \( p^*(\cdot) \) to represent the unnormalized probability distribution and \( Z(\theta) = \sum_{v, h} \exp(-E(v, h)) \) is the normalization constant. Unlike

\[ p(h_j = 1|v) = \frac{1}{1 + \exp(-\sum_i W_{ij} v_i - c_j)}, \]

\[ p(v_i|h) = \mathcal{N}(v_i; \mu_i, \sigma_i^2), \quad \text{where} \quad \mu_i = b_i + \sigma_i^2 \sum_j W_{ij} h_j. \]

If a second level RBM is used to model the activities of the hidden units of the first layer GRBM, we can combine their energy functions to form a Deep Belief Net (DBN) [2].
Let $\mathbf{v} \in \mathbb{R}^D$ represent the low resolution canonical image to be modeled generatively. For faces, $\mathbf{v}$ could be a $24 \times 24$ aligned and cropped face image. While we use the DBN in this work, other generative models can also be used instead. Let $\mathcal{I}$ be the high resolution image of the entire scene, where we want to use attention to propagate objects of interest up to the canonical representation $\mathbf{v}$. For example, $\mathcal{I}$ could be a $256 \times 256$ image.

This is all illustrated in the diagrams of Fig. 1. The left panel shows the model from [14]. The right panel is a graphical diagram featuring our generative model with attentional mechanisms. $h_1$ is the latent hidden nodes of the GRBM (In this work, we use a DBN, resulting in an additional $h_2$ layer). $x, y, r, s$ (position, rotation, and scale) are the parameters of the 2D Similarity transformation. The black box "shifter circuit" hides the generative process of 2D Similarity transformation as well as the approximate inference process facilitated by a predictive convolutional neural network (ConvNet), discussed in Sec. 4.1.

The 2D Similarity transformation is used to rotate, scale, and translate the canonical $\mathbf{v}$ onto the canvas that is $\mathcal{I}$. Let $\mathbf{p} = [x \ y]^T$ be a pixel coordinate (e.g. $[0,0], [0,1],...$) and $\{ \mathbf{p} \}$ be the set of all coordinates of an image. Let the "gaze" variables $\mathbf{u} \in \mathbb{R}^4 \equiv [\triangle x, \triangle y, \theta, s]$ be the parameter of the similarity transformation. While intuitive, in order to simplify derivations and to make transformations linear w.r.t. the transformation parameters, we can equivalently define $\mathbf{u} = [a, b, t_x, t_y]$, where $a = s \sin(\theta) - 1$ and $b = s \cos(\theta)$, see [23] for details. We define function $w := w(\mathbf{p}, \mathbf{u}) \rightarrow \mathbf{p}'$ as the transformation function to warp points $\mathbf{p}$ to $\mathbf{p}'$:

$$
\mathbf{p}' \triangleq \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 + a & -b \\ b & 1 + a \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}.
$$

(4)

We use the notation $\mathcal{I}(\{ \mathbf{p} \})$ to denote a new image sampled from $\mathcal{I}$ using the set of coordinates of $\{ \mathbf{p} \}$. During sampling, simple bilinear interpolation is used for anti-aliasing. Let image $\mathbf{x}(\{ \mathbf{p} \}, \mathbf{u})$ be sampled from a region $\mathcal{I}$ according to $\{ \mathbf{p} \}$ and $\mathbf{u}$:

$$
\mathbf{x} \triangleq \mathcal{I}(w(\{ \mathbf{p} \}, \mathbf{u})).
$$

(5)

For clarity we will use $\mathbf{x}$, but it is dependent on both $\{ \mathbf{p} \}$ and $\mathbf{u}$. The dimensionality of $\mathbf{x}$ is equal to the cardinality of $\{ \mathbf{p} \}$. Let $\{ \mathbf{p} \}$ be the set of pixel coordinates of $\mathbf{v}$, then given $\mathbf{v}$ and $\mathbf{u}$, we have a Gaussian distribution over $\mathbf{x}$ with a diagonal covariance matrix variance $\sigma^2$:

$$
p(\mathbf{x}|\mathbf{v}, \mathbf{u}) = \mathcal{N}(\mathbf{v}; \sigma^2)
$$

(6)

The fact that we do not seek to model the rest of the regions/pixels of $\mathcal{I}$ is by design. By using 2D Similarity transformation to mimic attention, we can discard the complex background of the scene and let the generative model focus on the object of interest. The proposed framework is governed by an overall energy function:
\[
E(x, v, u | I) = \frac{1}{2} \sum_i (x_i - v_i)^2 + f(u) + \frac{1}{2} \sum_i \frac{(v_i - b_i)^2}{\sigma_i^2} - \sum_j c_j h_j - \sum_{ij} W_{ij} v_i h_j
\]

where \( f(u) \) is the log prior over the gaze variables. In this paper, we simply use an improper prior: \( f(\cdot) \) to be a constant for all \( u \). We also share the same \( \sigma_i \) for both the GRBM as well as for modeling \( p(x|v, u) \). This simplifies the inference for Eq. \ref{eq:7}

4 Inference

While the generative equations in the last section are straightforward and intuitive, inference in these models is typically intractable due to the complicated energy landscape of the posterior. During inference, we wish to compute the distribution over the gaze \( u \) using inference, we wish to compute the distribution over the gaze \( u \). Dur-

While the generative equations in the last section are straightforward and intuitive, inference in these kinds of higher-ordered models.

One way to perform inference in our model is to resort to Gibbs sampling by computing the alternating set of conditional posteriors as follows: The conditional distribution over the canonical image \( v \) takes the following form:

\[
p(v|u, h; I) = \mathcal{N}\left(\frac{\mu + x(\{p\}, u)}{2}; \sigma^2\right)
\]  \hspace{1cm} \text{(7)}

\( \mu_i = b_i + \sigma_i^2 \sum_j W_{ij} h_j \) is the top-down influence of the GRBM/DBN. If we know the gaze variable and the top layer hidden variables, then \( v \) is simply a Gaussian where the mean is average of the top-down influence and bottom-up information from \( x \). The conditional distribution over \( h \) given \( v \) is given by Eq. \ref{eq:2}.

The conditional posterior over the gaze variables \( u \) is given by:

\[
\log p(u|x, v) = \log p(x|u, v) + \log p(u) = \frac{1}{2} \sum_i \frac{(x_i - v_i)^2}{\sigma_i^2} + \text{const.}
\]  \hspace{1cm} \text{(8)}

By using Bayes theorem, the unnormalized log probability of \( p(u|x, v) \) is defined in Eq. \ref{eq:8} We stress this equation is atypical in that the random variable of interest \( u \) actually affects the conditioning variable \( x \). We can explore the gaze variables by using gradient information using Hamiltonian Monte Carlo (HMC) algorithm \cite{24, 25}. Intuitively, conditioned on the canonical object our model has in “mind” \( v \), HMC search the big image \( I \) to find a region \( x \) with a good match to \( v \).

If we are interested in only finding the MAP of \( p(u|x, v) \), we might want to use second-order optimization algorithms to find the optimal \( u \). This would be equivalent to the well known Lucas-Kanade framework in computer vision, developed for image alignments \cite{26}. However, HMC has the advantage of being a proper MCMC sampler with refreshed momentum to help with jumping out of local maximum.

The HMC algorithm first specifies the Hamiltonian over the position variables \( u \) and auxiliary momentum variables \( p \) : \( H(u, p) = U(u) + K(p) \). Where the potential function \( U(u) = \frac{1}{2} \sum_i \frac{(x_i(u) - v_i)^2}{\sigma_i^2} \) and the kinetic energy function \( K(p) = \frac{1}{2} \sum_i p_i^2 \). The dynamics of the system is defined by these equations:

\[
\frac{\partial u}{\partial t} = p, \quad \frac{\partial p}{\partial t} = -\frac{\partial H}{\partial u}
\]  \hspace{1cm} \text{(9)}

\[
\frac{\partial H}{\partial u} = (x(u) - v) \frac{\partial x(u)}{\partial u}
\]  \hspace{1cm} \text{(10)}

\[
\frac{\partial x}{\partial u} = \frac{\partial x}{\partial \omega(\{p\}, u)} = \frac{\partial x_i}{\partial \omega(\{p\}, u)}
\]  \hspace{1cm} \text{(11)}
Eq. 11 decomposes into sums over single coordinate positions \( \vec{p}_i = [x, y]^T \). Let us denote \( \vec{p}'_i = w(\vec{p}_i, \vec{u}) \) to be the coordinate \( \vec{p}_i \) warped by \( \vec{u} \). For the first term on the RHS of Eq. 11

\[
\frac{\partial x_i}{\partial w(\vec{p}_i, \vec{u})} = \nabla I(\vec{p}'_i), \quad \text{(dimension 1 by 2 )}
\] (12)

\( \nabla I(\vec{p}'_i) \) denotes the sampling of the gradient images of \( I \) at the warped location \( \vec{p}_i \). For the second term on the RHS of Eq. 11 we note that we can re-write Eq. 4 as:

\[
\begin{bmatrix}
  x' \\
  y'
\end{bmatrix} = \begin{bmatrix}
  x & -y & 1 & 0 \\
  y & x & 0 & 1
\end{bmatrix} \begin{bmatrix}
  a \\
  b \\
  t_x \\
  t_y
\end{bmatrix} + \begin{bmatrix}
  x \\
  y
\end{bmatrix},
\] (13)

giving us

\[
\frac{\partial w(\vec{p}_i, \vec{u})}{\partial \vec{u}} = \begin{bmatrix}
  x & -y & 1 & 0 \\
  y & x & 0 & 1
\end{bmatrix}
\] (14)

HMC simulates the discretized system by performing leap-frog updates of \( \vec{u} \) and \( \vec{p} \) using Eq. 9.

Additional hyperparameters that need to be specified include the step size \( \epsilon \), number of leap-frog steps, and the mass of the variables (see [25] for details).

### 4.1 Approximate Inference

Since HMC is essentially performing gradient descent with extra momentum, it is prone to getting stuck at local optima. This is especially a problem for our task of finding the best transformation parameters. While the posterior over \( \vec{u} \) should be unimodal, the spatial frequency of the natural image signals can form many local minima. For example, in the left panel of Fig. 2(a), the big image \( I \) is enclose by the blue box, the canonical image \( v \) is enclosed by the green box. The current setting of \( \vec{u} \) aligns together the wrong eyes. However, it is hard to move the green box to the left due to the increase in the energy. Resampling the momentum variable every iteration can help give extra momentum to jump out of local minima. However, since we are modeling real-valued images using Gaussians as the residual, we have quadratic cost in the difference between \( x(\vec{u}) \) and \( v \), c.f. Eq. 8, making barriers between modes extremely high.

To alleviate this problem we need to have good initializations of \( \vec{u} \). We use a ConvNet [27] to perform efficient approximate inference, resulting in good initial guesses. Specifically, Given \( v, \vec{u} \) and \( I \), we predict the change in \( \vec{u} \) that will lead to the maximum \( \log p(\vec{u} | x, v) \). In other words, instead of using the gradient field for updating \( \vec{u} \), we learn a ConvNet to output a better vector field in the space of \( \vec{u} \).

We note that standard feedforward face detectors seek to model \( p(\vec{u} | I) \), while completely ignoring \( v \). In contrast, here we take \( v \) into account as well. The ConvNet is simply a way to initialize \( \vec{u} \) for the HMC algorithm. This is important in a proper graphical model as conditioning on \( v \) is appealing when multiple faces are present in the scene. Fig. 2(b) diagrams a hypothesized euclidean space of \( v \), where the black line is the manifold of canonical faces. The blue line is the manifold cropped faces \( x(\vec{u}) \). The blue manifold has a low intrinsic dimensionality of 4, spanned by \( \vec{u} \). At A and B, the blue comes close to black manifold. This means that there are at least two modes in the posterior over \( \vec{u} \). By conditioning on \( v \), we can narrow the posterior to a single mode, depending on whom we want to focus attention on. We demonstrate this exact capability in Sec. 6.3.

Fig. 2 demonstrates the iterative process of how approximate inference works in our model. Specifically, based on \( \vec{u} \), the ConvNet takes a window patch from \( I \) (e.g. 72 x 72) and \( v \) (e.g. 24 x 24) as
Step 1 Step 2 Step 3 Step 4

Figure 3: Inference process: \( \mathbf{u} \) in step 1 is randomly initialized. The average \( \mathbf{v} \) and the extracted \( \mathbf{x}(\mathbf{u}) \) form the input to a ConvNet for approximate inference, giving a new \( \mathbf{u} \). The new \( \mathbf{u} \) is used to sample \( p(\mathbf{v} | I, \mathbf{u}, \mathbf{h}) \). In step 3, alternating Gibbs sampling of the Gaussian RBM is performed. Step 4 repeats the approximate inference using the updated \( \mathbf{v} \) and \( \mathbf{x}(\mathbf{u}) \).

input, and predicts the output \([\Delta x, \Delta y, \Delta \theta, \text{scale}]\). In step 2, \( \mathbf{u} \) is updated accordingly, followed by step 3 of alternating Gibbs updates of \( \mathbf{v} \) and \( \mathbf{h} \) (Eqs. 7 and 2). The process is repeated.

5 Learning

Inference in our model localizes objects of interest and is akin to object detection. More importantly, we want to be able to learn generative models big images of the scene where objects are in unknown locations (gaze labels are missing). While learning with gaze labels \( \mathbf{u} \) is easy (since we can first obtain a center-aligned image of the object), the gaze labels are expensive to obtain and are often not available for images in an unconstrained environment.

To learn without gaze labels we propose a simple Monte Carlo Expectation-Maximization algorithm. During the E-step, we use the inference algorithm developed in Sec. 4 to sample from the posterior over the latent gaze variables \( \mathbf{u} \) and the canonical variables \( \mathbf{v} \). During the M-step, we can update the weights of the top layer Gaussian RBM to model these inferred data. In addition, we can update the parameters of the ConvNet which performs approximate inference. Due to the fact that the first E-step requires a good inference algorithm, we need to pre-train the ConvNet using labeled gaze data as part of a bootstrap process. Obtaining training data for this initial phase is not a problem as we can jitter/rotate/scale to create data. In Sec. 6.2, we demonstrate the ability to learn a good generative model of face images from the CMU Multi-PIE dataset (not using gaze labels) after first learning with gaze labels on the Caltech faces dataset.

6 Experiments

We used two face dataset in our experiments. The first dataset is a frontal face dataset, called the Caltech Faces from 1999, collected by Markus Weber. In this dataset, there are 450 faces of 27 unique individuals under different lighting conditions, expressions, and backgrounds. The original images had a resolution of 896 by 692. We downsample it by a factor of 2 in our experiments. The dataset also contains the manually labeled eyes and mouth coordinates, which will serve as the gaze labels. We also used the CMU Multi-PIE dataset [28] which contains 337 subjects, captured under 15 view points and 19 illumination conditions in four recording sessions for a total of more than 750,000 images. We demonstrate our model’s ability to perform approximate inference, to learn without labels, and to perform identity-based focus given an image with two people.

6.1 Approximate inference

We first investigate the critical inference algorithm of \( p(\mathbf{u}|\mathbf{v}, I) \) on the Caltech Faces dataset. We first run 4 steps of approximate inference detailed in Sec. 4.1 and diagrammed in Fig. 3 followed by three iterations of 20 leap-frog steps of HMC. Since we do not initially know the correct \( \mathbf{v} \), we initialize \( \mathbf{v} \) to be the average face across all subjects.

Qualitative results of two test subjects are in Fig. 4. The initial gaze box is colored yellow on the left. Subsequent gaze updates progress from yellow to blue. After ConvNet based approximate inference gives a good initialization, starting from step 5, 5 iterations of 20 leap-frog steps of HMC are used to sample the posterior and slightly improves the results. Fig. 5 shows the quantitative results of
Figure 4: Example of approximate inference steps. \( v \) is 24 \( \times \) 24, \( x \) is 72 \( \times \) 72. Approximate inference quickly finds a good initialization for \( u \), while HMC makes small adjustments.

Figure 5: (a) Accuracy as a function of gaze initialization (pixel offset). Blue plot is the percent success of at least 50% IOU. Red plot is the average IOU. (b) Accuracy as a function of the number of approximate inference steps when initializing 50 pixels away. (c) Accuracy improvements of HMC as a function of gaze initializations.

Intersection of Union (IOU) of the ground truth face box and the inferred face box. The results show that inference is very robust to initialization and requires only a few steps of approximate inference to converge. HMC clearly improves model performance, resulting in an IOU increase of about 5% for localization. This is impressive given that none of the test subjects were part of the training and the background is different from backgrounds in the training set.

We also compared our inference algorithm to template matching in the task of face detection. We took the first 5 subjects as test subjects and the rest as training. We can localize with 97% accuracy (IOU > 0.5) using our inference algorithm. In comparison, euclidean distance template matching with known scale obtained 78% and Normalized Cross Correlation [29] obtained 93% accuracy.

6.2 Learning without gaze labels

The novelty of our model is that it can learn on new large images of faces without label information. To demonstrate, we first trained a Deep Belief Net with 1024 \( h_1 \) layer 1 and 200 \( h_2 \) layer 2 hidden units, as well as a ConvNet for approximate inference on the Caltech faces with gaze labels. Fig. 6(a) shows samples drawn from the learned DBN, demonstrating that it has learned a reasonable generative model of faces. The model was then given 1418 Multi-PIE images with the goal of estimating the gaze variables \( u \) in these images. A threshold on \( \log p(x(u)|u, v) \) was used to determine success or failure. We were able to successfully localize 951 out of 1418 images. For the successfully localized images, we used Eq. 7 to infer the latent canonical \( v \). Using the inferred \( v \) as data, the M-step was performed by using Fast Persistent Contrastive Divergence [30] to learn the DBN. Fig. 6(b) clearly shows that the DBN has learned a reasonable generative model of individuals.

\footnote{\( u \) is randomly initialized around the central part of test images, \( \pm 30 \) pixels, scale from 0.5 to 1.5.}
Figure 6: Left: samples of a 2-layer DBN learned on Caltech. Right: samples from updated DBN after training on CMU Multi-PIE without gaze labels.

Figure 7: E-step for learning on CMU Multi-PIE. (a),(b),(c) are successful. (d) is a failure case.

Figure 8: Left: Conditioned on different $v$ will result in a different $\Delta u$. Note that the initial $u$ is exactly the same for two trials. Right: additional examples. The only difference between the top and bottom is the conditioned $v$.

from the CMU face database. Fig. 7 further shows some inference examples on the CMU Multi-PIE images.

6.3 Inference with ambiguity

The posterior $p(u|x, v)$ is conditioned on $v$, which means that where to attend must be a function of the canonical object $v$ the model has in “mind”. This is clearly useful when multiple objects/faces are present in the scene c.f. Fig. 2(b). To explore this, we first synthetically generate a dataset by concatenating together two faces from the Caltech dataset. We then train approximate inference ConvNet as in Sec. 4.1 and test on the held-out subjects. Indeed, as predicted, Fig. 8 shows that depending on which canonical image is conditioned, the same exact gaze initialization leads to two very different gaze shifts. Note that this phenomenon is observed across different scales and location of the initial initialization. For example, in right-bottom of Fig. 8 the initialized yellow box is mostly on the female to the left, but because conditioned $v$ is that of the right male, attention is shifted to the right.

7 Conclusion

In this paper we have proposed a probabilistic graphical model framework for learning generative models using attention. Experiments on face modeling have shown that ConvNet based approximate inference combined with HMC sampling is sufficient to explore the complicated posterior. More importantly, we can generatively learn objects of interest from novel datasets without location labels. Future work will include experimenting with faces as well as other objects in a large scene. While currently the ConvNet approximate inference is trained in a supervised manner, reinforcement learning could be used instead.
References

[1] P. Smolensky. Information processing in dynamical systems: Foundations of harmony theory. In D. E. Rumelhart and J. L. McClelland, editors, Parallel Distributed Processing, volume 1, chapter 6, pages 194–281. MIT Press, Cambridge, 1986.

[2] G. E. Hinton, S. Osindero, and Y. W. Teh. A fast learning algorithm for deep belief nets. Neural Computation, 18(7):1527–1554, 2006.

[3] R. Salakhutdinov and G. Hinton. Deep Boltzmann machines. In AISTATS, 2009.

[4] Geoffrey E. Hinton, Peter Dayan, and Michael Revow. Modeling the manifolds of images of handwritten digits. IEEE Transactions on Neural Networks, 8(1):65–74, 1997.

[5] Daniel Zoran and Yair Weiss. From learning models of natural image patches to whole image restoration. In ICCV. IEEE, 2011.

[6] Yoshua Bengio, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. In Advances in Neural Information Processing Systems 26, 2013.

[7] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In Intl. Conf. on Machine Learning, pages 609–616, 2009.

[8] Marc’Aurelio Ranzato, Joshua Susskind, Volodymyr Mnih, and Geoffrey Hinton. On Deep Generative Models with Applications to Recognition. In CVPR, 2011.

[9] Yichuan Tang, Ruslan Salakhutdinov, and Geoffrey E. Hinton. Deep Boltzmann machines. In AISTATS, 2009.

[10] Geoffrey E. Hinton, Peter Dayan, and Michael Revow. Modeling the manifolds of images of handwritten digits. IEEE Transactions on Neural Networks, 8(1):65–74, 1997.

[11] Geoffrey E. Hinton, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. In Advances in Neural Information Processing Systems 26, 2013.

[12] Yoshua Bengio, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. In Advances in Neural Information Processing Systems 26, 2013.

[13] Geoffrey E. Hinton, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. In Advances in Neural Information Processing Systems 26, 2013.

[14] Geoffrey E. Hinton, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. In Advances in Neural Information Processing Systems 26, 2013.

[15] Geoffrey E. Hinton, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. In Advances in Neural Information Processing Systems 26, 2013.

[16] Geoffrey E. Hinton, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. In Advances in Neural Information Processing Systems 26, 2013.

[17] Geoffrey E. Hinton, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. In Advances in Neural Information Processing Systems 26, 2013.

[18] Geoffrey E. Hinton, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. In Advances in Neural Information Processing Systems 26, 2013.

[19] Geoffrey E. Hinton, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. In Advances in Neural Information Processing Systems 26, 2013.

[20] Geoffrey E. Hinton, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. In Advances in Neural Information Processing Systems 26, 2013.

[21] Geoffrey E. Hinton, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. In Advances in Neural Information Processing Systems 26, 2013.

[22] Geoffrey E. Hinton, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. In Advances in Neural Information Processing Systems 26, 2013.

[23] Geoffrey E. Hinton, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. In Advances in Neural Information Processing Systems 26, 2013.

[24] Geoffrey E. Hinton, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. In Advances in Neural Information Processing Systems 26, 2013.

[25] Geoffrey E. Hinton, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. In Advances in Neural Information Processing Systems 26, 2013.

[26] Geoffrey E. Hinton, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. In Advances in Neural Information Processing Systems 26, 2013.

[27] Geoffrey E. Hinton, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. In Advances in Neural Information Processing Systems 26, 2013.

[28] Geoffrey E. Hinton, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. In Advances in Neural Information Processing Systems 26, 2013.

[29] Geoffrey E. Hinton, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. In Advances in Neural Information Processing Systems 26, 2013.

[30] Geoffrey E. Hinton, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. In Advances in Neural Information Processing Systems 26, 2013.

[31] Geoffrey E. Hinton, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. In Advances in Neural Information Processing Systems 26, 2013.

[32] Geoffrey E. Hinton, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. In Advances in Neural Information Processing Systems 26, 2013.

[33] Geoffrey E. Hinton, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. In Advances in Neural Information Processing Systems 26, 2013.

[34] Geoffrey E. Hinton, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. In Advances in Neural Information Processing Systems 26, 2013.

[35] Geoffrey E. Hinton, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. In Advances in Neural Information Processing Systems 26, 2013.

[36] Geoffrey E. Hinton, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. In Advances in Neural Information Processing Systems 26, 2013.

[37] Geoffrey E. Hinton, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. In Advances in Neural Information Processing Systems 26, 2013.

[38] Geoffrey E. Hinton, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. In Advances in Neural Information Processing Systems 26, 2013.

[39] Geoffrey E. Hinton, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. In Advances in Neural Information Processing Systems 26, 2013.

[40] Geoffrey E. Hinton, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. In Advances in Neural Information Processing Systems 26, 2013.