ARTICLE **Communicated by Kevin Murphy**

Learning a Generative Model of Images by Factoring Appearance and Shape

Nicolas Le Roux
nicolas@le-roux.name
Microsoft Research Cambridge, Machine Learning and Perception, Cambridge CB3 0FB, U.K.

Nicolas Heess
n.m.o.heess@sms.ed.ac.uk
Neuroinformatics and Computational Neuroscience Doctoral Training Centre, Institute for Adaptive and Neural Computation, School of Informatics, University of Edinburgh, Edinburgh EH8 9AB, U.K.

Jamie Shotton
jamiesho@microsoft.com
John Winn
jwinn@microsoft.com
Microsoft Research Cambridge, Machine Learning and Perception, Cambridge CB3 0FB, U.K.

Computer vision has grown tremendously in the past two decades. Despite all efforts, existing attempts at matching parts of the human visual system's extraordinary ability to understand visual scenes lack either scope or power. By combining the advantages of general low-level generative models and powerful layer-based and hierarchical models, this work aims at being a first step toward richer, more flexible models of images. After comparing various types of restricted Boltzmann machines (RBMs) able to model continuous-valued data, we introduce our basic model, the masked RBM, which explicitly models occlusion boundaries in image patches by factoring the appearance of any patch region from its shape. We then propose a generative model of larger images using a field of such RBMs. Finally, we discuss how masked RBMs could be stacked to form a deep model able to generate more complicated structures and suitable for various tasks such as segmentation or object recognition.

1 Introduction

Despite much progress in the field of computer vision in recent years, interpreting and modeling the bewildering structure of natural images remains

Nicolas Le Roux and Nicolas Heess contributed equally to this article.

Color versions of all figures in this article are presented in the online version, available at http://www.mitpressjournals.org/doi/abs/10.1162/NECO_a_00086.

a challenging problem. The limitations of even the most advanced systems become strikingly obvious when contrasted with the ease, flexibility, and robustness with which the human visual system analyzes and interprets an image. Computer vision is a problem domain where the structure that needs to be represented is complex and strongly task dependent and the input data are often highly ambiguous. Against this background, we believe that rich, generative models are necessary to extract an accurate and meaningful representation of the world, detailed enough to make them suitable for a wide range of visual tasks. This work is a first step toward building such a general-purpose generative model able to perform varied high-level tasks on natural images. The model integrates concepts from computer vision that combine some very general knowledge about the structure of our visual world with ideas from deep unsupervised learning. In particular, it draws on ideas such as:

- The separation of shape and appearance and the explicit treatment of occlusions
- A generic, learned model of shapes and appearances
- The unsupervised training of a generative model on a large database, exploiting graphical models that foster efficient inference and learning
- The modeling of large images using a field of more local experts
- The potential for a hierarchical latent representation of objects

Some of these ideas have been explored independent of each other and in models that focused on particular aspects of images or that were applied to very limited (e.g., category specific) data sets. Here we demonstrate how these techniques, in combination, give rise to a promising model of generic natural images.

One premise of the work described in this article is that generative models hold important advantages in computer vision. Their most obvious advantage over discriminative methods is perhaps that they are more amenable to unsupervised learning, which seems of crucial importance in a domain where labeled training data are often expensive while unlabeled data are now easy to obtain. Equally important, however, is that in vision, we are rarely interested in solving a single task such as object classification. Instead we typically need to extract information about different aspects of an image and at different levels of abstraction—for example, recognizing whether an object is present, identifying its position and those of its parts, and separating pixels belonging to the object from the background or occluding objects (segmentation). Many lower-level tasks, such as segmentation, are not even well defined without reference to more abstract structure (e.g., the object or part to be segmented), and information in natural images, especially when it is low level and local, is often highly ambiguous. These considerations strongly suggest that we need a model that can represent and learn a rich prior of image structure at many different levels of abstraction and also allow efficiently combining bottom-up (from the data) with

top-down (from the prior) information during inference. Probabilistic, generative models naturally offer the appropriate framework for doing such inference. Furthermore, unlike in the discriminative case, they are trained not with respect to a particular task-specific label (which in most cases provides very little information about the complex structure present in an image) but rather to represent the data efficiently. This makes it much more likely that the required rich prior can ultimately be learned, especially if a suitable (e.g., a hierarchical) model structure is assumed. In this article we briefly review the most closely related works, even though such a review will necessarily be incomplete.

Some generative models can extract information about shape and appearance, illumination, occlusion and other factors of variation in an unsupervised manner (Frey & Jojic, 2003; Williams & Titsias, 2004; Kannan, Jojic, & Frey, 2005; Winn & Jojic, 2005; Kannan, Winn, & Rother, 2006). Though these models have successfully been applied to sets of relatively homogeneous images, such as images of particular object classes or movies of a small number of objects, they have limited scope and are typically not suitable for more heterogeneous data, let alone generic natural images.

Generic image structure is the domain of models such as the sparse coding approach by Olshausen and Field (1996; Lewicki & Olshausen, 1999; Hyvärinen, Hoyer, & Inki, 2001; Karklin and Lewicki, 2009) or the more recent work, broadly referred to as deep learning architectures (Osindero & Hinton, 2008; Lee, Ekanadham, & Ng, 2008). Unlike the models in the previous category, these models of generic image structure have very little built-in knowledge about the formation of natural images and are trained on large, unlabeled image databases. In particular, for the second group of models, the hope is that by learning increasingly deep (i.e., multilayered) representations of natural images, these models will capture structures of increasing complexity and at larger scales. Although this line of work has produced interesting results, so far the models are typically limited to small image patches (with some exceptions, see, e.g., Lee, Grosse, Ranganath, and Ng, 2009 and Raina, Madhavan, & Ng, 2009). Furthermore, most models so far, including hierarchical ones, appear to learn only very simple, low-level properties of natural images and are far from learning more abstract, higher-level concepts, suggesting that these models might still be too limited to capture the wealth of structure in natural images.

A large body of computer vision literature has focused on hierarchical image representations of various kinds, in particular on the recursive compositions of objects from parts, and many of these works employ generative (probabilistic) formulations of the hierarchy (see Bienenstock, Geman, & Potter, 1997; Jin & Geman, 2006; Fidler & Leonardis, 2007; Ommer & Buhmann, 2010; Zhu, Lin, Huang, Chen, & Yuille, 2008; Todorovic & Ahuja, 2008; Bouchard and Triggs, 2005; Zhu & Mumford, 2006, for some examples). The focus here is often less on modeling full images (in particular, not in such a manner that new images could be generated from these models) than

on developing a representation for recognition or segmentation. Learning such models, in particular the structure of the hierarchy, can be challenging although progress has recently been made (e.g., Fidler & Leonardis, 2007; Ommer & Buhmann, 2010; Zhu et al., 2008; Todorovic & Ahuja, 2008). One important insight that has arisen from these compositional models of images, but also from tree-structured belief network models of images (e.g., Bouman & Shapiro, 1994; Luettgen & Willsky, 1995), is the notion that such a hierarchy needs to be flexible and allowed to vary in structure so as to match the underlying dependencies present in any particular image. This issue has been addressed in the work on dynamic trees (Williams & Adams, 1999; Storkey & Williams, 2003), and also in the credibility network model (Hinton, Ghahramani, & Teh, 2000), among others. However, these methods still fall short of being able to capture the complexity of natural images: for example, dynamic trees do not impose a depth ordering or learn an explicit shape model as a prior over tree structures.

Most of the work described in the previous paragraphs focuses on certain aspects of natural images. The question as to what kinds of models are suitable for comprehensively modeling the very different types of structure that typically co-occur in images has featured prominently in the work of Zhu and his coworkers (Guo, Zhu, & Wu, 2003, 2007; Tu, Chen, Yuille, & Zhu, 2005; Zhu & Mumford, 2006). Recently they proposed a generative model that combines submodels of different types for capturing the different kinds of structure occurring in natural images at different levels of abstraction and scale, ranging from low-level structures such as image textures to high-level part-based representations of objects and, ultimately, full visual scenes. This model appears to be one of the most comprehensive available to date, but due to its complexity, it currently fails to leverage one of the potential advantages of generative models in that unsupervised learning seems extremely difficult. Thus, training relies quite heavily on hand-labeled data, which are expensive to get.

In light of all these works, we aim at providing a unified probabilistic framework able to deal with generic, large images in an efficient manner from both a representation and an inference point of view.

The base component of our model is the restricted Boltzmann machine (Smolensky, 1986; Freund & Haussler, 1994), which is a Boltzmann machine (Ackley, Hinton, & Sejnowski, 1985) restricted to have bipartite connectivity. Section 2 presents and compares various RBMs able to model continuous-valued data, which will prove useful when we model appearances of objects. Section 3 presents the masked RBM, which extends the already rich modeling capacity of an RBM with a depth-ordered segmentation model. The masked RBM represents the shape and appearance of image regions separately, and it explicitly reasons about occlusion. The shape of objects is modeled by another RBM, introduced in section 4. This opens up new application domains (such as image segmentation and inpainting), and, importantly, leads to a much more efficient representation of image structure

than standard RBMs, which can be learned in a fully unsupervised manner from training images. Despite its complexity and power, our model allows efficient approximate inference and learning. Section 5 is a thorough evaluation of this model's quality using both toy data and natural image patches, demonstrating how explicit incorporation of knowledge about natural images formation considerably increases the efficiency of the learned representation.

We then move from image patches to full images by introducing the field of masked RBMs in section 6, leveraging the modeling power we obtained at the patch level, before concluding in section 7.

Finally, as future work, we propose in section 8 a hierarchical formulation of the basic model that gives rise to a flexible, reconfigurable tree-structured representation that would allow us to learn image structures at different scales and levels of abstraction.

2 Binary and Continuous-Valued RBMs

In this section, we introduce the standard RBM, defined over binary variables and than present several RBMs able to model continuous-valued data.

2.1 The Binary RBM. A binary RBM with n hidden units is a parametric model of the joint distribution between binary hidden variables h_j (explanatory factors, collected in vector $\mathbf{h}$) and binary observed variables v_i (the observed data, collected in vector $\mathbf{v}$), of the form

$$\log P(\mathbf{v}, \mathbf{h}) = \mathbf{v}^T W \mathbf{h} + \mathbf{b}^T \mathbf{v} + \mathbf{c}^T \mathbf{h} - \log Z^1, \tag{2.1}$$

with[1] parameters $\theta = (W, \mathbf{b}, \mathbf{c})$ and $v_i, h_j \in \{0, 1\}$ (Z is the normalizing constant).

One can show that conditional distributions $P(\mathbf{v}|\mathbf{h})$ and $P(\mathbf{h}|\mathbf{v})$ are factorial and thus easy to sample from Hinton (2002). Although the marginal distribution $P(\mathbf{v})$ is not tractable, it can be easily computed up to a normalizing constant. The bipartite structure of an RBM allows both inference and learning to be performed efficiently using Gibbs sampling (Hinton, Osindero, & Teh, 2006).

2.2 Modeling Continuous Values with an RBM. Since we are building a generative model of RGB images, we will need to use generative models of (potentially bounded) real-valued vectors of the red, green, and blue channel values. Surprisingly little work has been done on designing efficient RBMs for real-valued data.

[1] Throughout the article, we slightly abuse notation and use the variable Z for all partition functions, although they depend on the energy function.

The general foundations for using RBMs to model distributions in the exponential family were laid in Welling, Rosen-Zvi, and Hinton (2005), where one particular instantiation of this family was investigated for modeling discrete data using continuous latent variables. To date, using other members of this family to learn data variance has not been explored.

Some authors have used RBMs in the context of continuous values, using a truncated exponential (Larochelle, Erhan, Courville, Bergstra, & Bengio, 2007), gaussians with fixed variance (Freund & Haussler, 1994; Lee et al., 2008), or rectified linear units (Nair & Hinton, 2010). In none of these cases is the variance learned. In the case of the truncated exponential, even though the variance does depend on the parameters, it is a deterministic function of the mean and cannot be separately optimized. We will thus refer to this model as having fixed variance.

We now present several kinds of RBMs able to model continuous-valued data.

2.3 Truncated Exponential. The use of the truncated exponential with an RBM is a direct extension of the original formulation to continuous values. The energy functions remains identical,

$$E(\mathbf{v}, \mathbf{h}) = -\mathbf{v}^T W \mathbf{h} - \mathbf{b}^T \mathbf{v} - \mathbf{c}^T \mathbf{h}, \tag{2.2}$$

but $\mathbf{v}$ may now take any value in $[0, 1]$. The conditional $P(\mathbf{v}|\mathbf{h})$ is a truncated exponential.

2.4 Gaussian RBM with Fixed Variance. Gaussian RBMs have already been studied (Freund & Haussler, 1994) and used (Lee et al., 2008), but always in the context of a fixed variance. The energy function is of the form

$$E(\mathbf{v}, \mathbf{h}) = \frac{1}{2\sigma^2} \mathbf{e}^T \mathbf{v}^2 - \frac{1}{\sigma^2} \left(\mathbf{b}^T \mathbf{v} + \mathbf{c}^T \mathbf{h} + \mathbf{v}^T W \mathbf{h} \right), \tag{2.3}$$

where $\mathbf{v}^2$ is the vector whose ith element is v_i^2 and $\mathbf{e} = [1, 1, \ldots, 1]^T$. This model is restricted to be a mixture of isotropic gaussians.

Choosing a fixed variance to use with this model is problematic: large variances make training very noisy, while small variances cause training to get stuck in local maxima. The heuristic approach aims at avoiding the problems of a large, fixed variance by using the mean of $P(\mathbf{v}|\mathbf{h})$, rather than a sample from it, during training. We will show the results obtained with the fixed variance model trained normally (Gaussian Fixed) and trained using this heuristic (Gaussian Heuristic).

2.5 Gaussian RBM with Learned Variance. We now present an extension of the gaussian RBM model that allows the modeling of the variance. We consider two similar models: the first uses the same hidden units to

model both the mean and the precision (Gaussian Joint), and the second uses different sets of hidden units for each (Gaussian Separate).

2.5.1 Joint Modeling of Mean and Precision. The energy function for this model represents the mean and precision jointly using a common set of hidden units:

$$E(\mathbf{v}, \mathbf{h}) = -\mathbf{v}^T W^{\mathrm{m}} \mathbf{h} - \left(\mathbf{v}^2\right)^T W^{\mathrm{p}} \mathbf{h} - \mathbf{v}^T \mathbf{b}^{\mathrm{m}} - \left(\mathbf{v}^2\right)^T \mathbf{b}^{\mathrm{p}} - \mathbf{c}^T \mathbf{h}. \tag{2.4}$$

Denoting precision $\Lambda = -2(W^{\mathrm{p}}\mathbf{h} + \mathbf{b}^{\mathrm{p}})$, we have

$$P(v_i|\mathbf{h}) \sim \mathcal{N}\left(\frac{W^{\mathrm{m}}_{i,:}\mathbf{h} + b^{\mathrm{m}}_i}{\Lambda_i}, \frac{1}{\Lambda_i}\right). \tag{2.5}$$

In this model, the biases $\mathbf{b}^{\mathrm{p}}$ and weights W^{p} are forced to be negative.

2.5.2 Separate Modeling of Mean and Precision. Here, the energy function uses one set of hidden units $\mathbf{h}^{\mathrm{m}}$ to model the mean and a separate set of hidden units $\mathbf{h}^{\mathrm{p}}$ to model the precision:

$$\begin{aligned} E(\mathbf{v}, \mathbf{h}^{\mathrm{m}}, \mathbf{h}^{\mathrm{p}}) = &-\mathbf{v}^T W^{\mathrm{m}} \mathbf{h}^{\mathrm{m}} - \left(\mathbf{v}^2\right)^T W^{\mathrm{p}} \mathbf{h}^{\mathrm{p}} - \mathbf{v}^T \mathbf{b}^{\mathrm{m}} - \left(\mathbf{v}^2\right)^T \mathbf{b}^{\mathrm{p}} \\ &- \left(\mathbf{c}^{\mathrm{m}}\right)^T \mathbf{h}^{\mathrm{m}} - \left(\mathbf{c}^{\mathrm{p}}\right)^T \mathbf{h}^{\mathrm{p}}. \end{aligned} \tag{2.6}$$

Denoting $\Lambda = -2(W^{\mathrm{p}}\mathbf{h}^{\mathrm{p}} + \mathbf{b}^{\mathrm{p}})$, we now have

$$P(v_i|\mathbf{h}^{\mathrm{m}}, \mathbf{h}^{\mathrm{p}}) \sim \mathcal{N}\left(\frac{W^{\mathrm{m}}_{i,:}\mathbf{h}^{\mathrm{m}} + b^{\mathrm{m}}_i}{\Lambda_i}, \frac{1}{\Lambda_i}\right). \tag{2.7}$$

In this model, the biases $\mathbf{b}^{\mathrm{p}}$ and weights W^{p} are forced to be negative.

2.6 Beta RBM. In the beta RBM, the conditional distributions $P(\mathbf{v}|\mathbf{h})$ are beta distributions whose means and variances are learned during training.

Before going any further, we would like to recall the link between RBMs and products of experts (for a detailed explanation, see Freund & Haussler, 1994). When we sum out over all possible values of $\mathbf{h}$ in the energy function of an RBM, the unnormalized probability of a state $\mathbf{x}$ is the product of as many experts as there are hidden units, each expert being a mixture of two distributions—one when the hidden unit is turned on, one when it is turned off.

If we were to simply apply the formula given by Welling et al. (2005), the energy function of the beta RBM would be

$$\begin{aligned} E(\mathbf{v}, \mathbf{h}) = &-\log(\mathbf{v})^T W \mathbf{h} - \log(\mathbf{e} - \mathbf{v})^T U \mathbf{h} \\ &+ (\mathbf{e} - \mathbf{a})^T \log(\mathbf{v}) + (\mathbf{e} - \mathbf{b})^T \log(\mathbf{e} - \mathbf{v}) - \mathbf{c}^T \mathbf{h}. \end{aligned} \tag{2.8}$$

In this formulation, each expert is a mixture of a uniform and a beta distribution. Unfortunately, training such an RBM proved very difficult, as turning a hidden unit on could only increase the precision of the conditional distribution. Furthermore, there is no easy way of enforcing the positivity constraint on the parameters of the beta distributions (enforcing all the elements of $\mathbf{a}$, $\mathbf{b}$, W, and U to be positive resulted in too hard a constraint).

We therefore modified the energy so that each expert is a mixture of two beta distributions. By doing so, we symmetrize the hidden units and can have weaker constraints on the parameters while still retaining valid distributions. The new, modified energy function is then

$$\begin{aligned} E(\mathbf{v},\mathbf{h}) = &-\log(\mathbf{v})^T W_1 \mathbf{h} - \log(\mathbf{v})^T W_2(\mathbf{e}-\mathbf{h}) \\ &-\log(\mathbf{e}-\mathbf{v})^T U_1 \mathbf{h} - \log(1-\mathbf{v})^T U_2(\mathbf{e}-\mathbf{h}) \\ &+\mathbf{e}^T \log(\mathbf{v}) + \mathbf{e}^T \log(\mathbf{e}-\mathbf{v}) - \mathbf{c}^T \mathbf{h}, \end{aligned} \tag{2.9}$$

with the elements of W_1, W_2, U_1, and U_2 restricted to be positive (note that we no longer have the visible biases $\mathbf{a}$ and $\mathbf{b}$ as these may be included in the weight matrices). As beta distributions treat the boundary values (0 and 1) differently from the others, we extended their range to $[-\lambda, 1+\lambda]$ with $\lambda = (\frac{\sqrt{5}-1}{2})$.[2]

2.7 Assessment of the Quality of Each RBM. To choose the most appropriate RBM for the real-valued red, blue, and green channels, we compared all these models on natural image patches (of size 16×16), using three quantitative metrics: the reconstruction RMSE, the reconstruction log likelihood, and the imputation accuracy. The experiments were led on patches that were not seen during training.

2.7.1 Experimental Setup. All models were trained on a training set of $383,300$ color image patches of size 16×16. Patches were extracted on a regular 16×16 grid from images from three different object recognition data sets: Pascal VOC, MSR Cambridge, and the INRIA horse data set.[3] Red, green, and blue color channels are concatenated so that each model has 768 visible units. Where necessary, we used an appropriately sized validation set.

[2] $\lambda = \frac{\sqrt{5}-1}{2}$ has the properties that $\log(\lambda) = -\log(1+\lambda)$ and $\log(1+\lambda) - \log(\lambda) \approx 1$. The first property ensures that the range of inputs to the hidden units are symmetric around 0, and the second property ensures that $\log(\mathbf{v}+\lambda)$, $\log(1+\lambda-\mathbf{v})$ and $\mathbf{h}$ are approximately of the same amplitude when $\mathbf{v}$ lies in the interval $[0,1]$.

[3] Available online at http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2008/, http://research.microsoft.com/vision/cambridge/recognition/, and http://lear.inrialpes.fr/data, respectively.

We trained the model using gradient descent with persistent contrastive divergence (Tieleman, 2008) and batches of size 20. We used a small weight decay and decreased the learning rate every epoch (one run through all training patches), dividing each epoch into batches.

The hyperparameters were not treated equally:

- The weight decay and decrease constant were manually fixed to .0002 and .001, respectively.
- The learning rate was optimized using the validation set, taking the learning rate that gives the best log likelihood of the data given the inferred latent variables after one epoch.
- In the case of the beta RBM, to get an idea of the effect of parameter λ, we tried three different values of λ for the case of 256 hidden units. We decided beforehand to report for 512 and 1024 hidden units only the results for $\lambda = \frac{\sqrt{5}-1}{2}$.

Once the optimal learning rate was found, we trained each model for 20 epochs in batches of size 20 patches. Models were trained for three different sizes of the hidden layer: 256, 512, and 1024 hidden units.

2.7.2 Reconstruction RMSE. This experiment is used to determine the ability of each RBM to correctly model the mean of the data. Reconstruction is performed as follows. Given a test patch $\mathbf{v}_{\text{test}}$, we sample a configuration of the hidden states $\mathbf{h}^\star$ from the conditional distribution $P(\mathbf{h}|\mathbf{v}_{\text{test}})$. Given this configuration $\mathbf{h}^\star$, we compute the average value of the visible states $E[P(\mathbf{v}|\mathbf{h}^\star)]$. This is called a mean reconstruction of the test patch. Note that this is not the true average reconstruction since we consider only one configuration of the hidden states, not the full conditional distribution. Finally, we compute the pixel-wise squared error between the reconstruction and the original patch .

RMSE reconstruction accuracies for the different models (with 1024 hidden units) are shown in Figure 1a where the accuracies have been averaged across all test patches. Note that because the RMSE measure uses only the mean of $P(\mathbf{v}|\mathbf{h}^\star)$, the accuracy of the variance of $P(\mathbf{v}|\mathbf{h}^\star)$ is not assessed in these plots. A selection of test patches and their mean reconstructions is shown in Figures 2a and 2b.

The truncated exponential does a reasonable job of reconstructing the patches, but it is exceeded in performance by all three of the learned-variance models. This leads to the counterintuitive result that models designed to capture data variance prove to be significantly better at representing the mean. An explanation is that these models learn where they are able to represent the data accurately (e.g., in untextured regions) and where they cannot (e.g., near edges) and hence are able to focus their modeling power on the former rather than the latter, leading to an overall improvement in RMSE. The overall best performer is the beta RBM, which not only has the

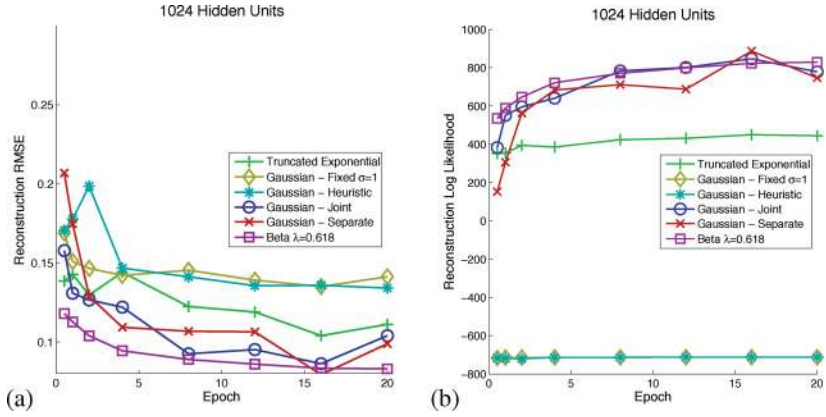

Figure 1: Reconstruction accuracy for different models. (a) RMSE of reconstructed test patches for different stages of training. (b) Log likelihood of reconstructed patches.

best average RMSE but also shows much greater stability during training in comparison to the gaussian models (as may be seen in Figure 1a).

2.7.3 Reconstruction Log Likelihood. This experiment is a proxy to the true log probability of the data. To obtain the true probability of a test patch, one could start a Markov chain from this same patch, run for an infinite amount of time, and compute the log probability of that patch under the final distribution (the choice of starting point would actually have no influence). Since this would be too expensive, we consider only an unbiased sample of the distribution obtained after one Markov step. We therefore perform the following experiment:

1. Given a test patch $\mathbf{v}_{\text{test}}$, we sample a configuration of the hidden states $\mathbf{h}^\star$ from the conditional distribution $P(\mathbf{h}|\mathbf{v}_{\text{test}})$.
2. Given this configuration of the hidden states, we compute the conditional probability of the test patch $P(\mathbf{v}_{\text{test}}|\mathbf{h}^\star)$, which is easily done given the factoriality of this distribution.

Results for all models are given in Figure 1b, again with 1024 hidden units. Unlike the RMSE reconstruction, the log likelihood jointly assesses the accuracy of the mean and variance of the model. Hence, differences from the RMSE reconstruction results indicate models where the variance is modeled more or less accurately. Unsurprisingly, the fixed variance models do very poorly on this metric since they have fixed, large variances. More interestingly, the joint gaussian model now achieves very similar performance to the beta, indicating that it is modeling the variance better than the beta (considering that it modeled the mean slightly worse). This may be due to

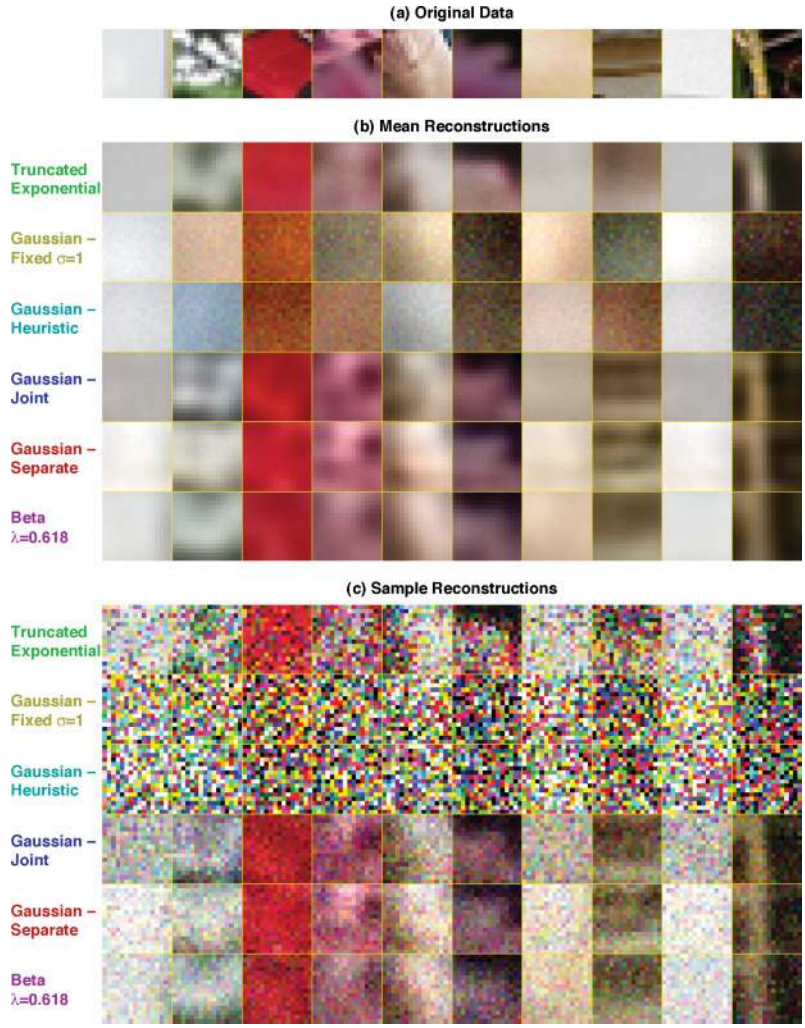

Figure 2: Reconstructions of patches from the test set.

the gaussian being light-tailed in comparison to the beta and hence able to put greater probability mass near the mean.

2.7.4 Imputation Accuracy. As a further investigation of the models' abilities to represent the distribution over image patches, we assessed their performance at filling in missing pixels in test patches, a process known as imputation. We used the experimental process:

1. Given a test patch, randomly select a region of 1×1, 2×2 or 4×4 pixels, and consider these pixels to be missing.

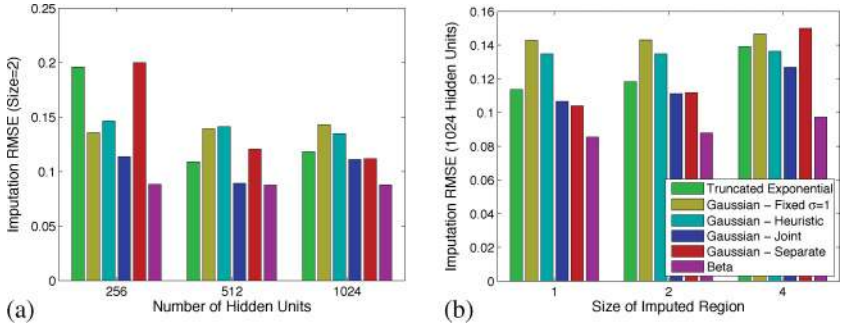

Figure 3: Imputation RMSE for different models, as a function of (a) the number of hidden units and (b) the size of the imputed region.

2. Initialize the missing pixels to the mean of the observed pixels.
3. Perform 16 bottom-up and top-down passes to impute the values of the missing pixels. In each top-down pass, the values of the observed pixels are fixed, while the values of the missing pixels are sampled from $P(\mathbf{v}|\mathbf{h})$. Enough passes are chosen to allow mixing to occur (bear in mind that we are sampling from the conditional distribution of the unobserved pixels given the observed pixels, which is highly concentrated).

The RMSEs between the imputed and true pixel values for the different models are shown in Figure 3a for models with differing numbers of hidden units and in Figure 3b for different-sized imputation regions. Again, the beta RBM leads to the best performance in all cases, with the less stable joint gaussian RBM typically coming in second.

2.8 Conclusion. Across experiments, the beta RBM proved more robust and slightly more accurate than all the other types of RBM. We therefore decided to use it to model appearances. Nevertheless, one should bear in mind that there is room for improvement and other, higher-quality continuous-valued RBMs may exist.

3 The Masked Restricted Boltzmann Machine

An RBM will capture high-order interactions between visible units, to the limit of its representational power determined by the number of hidden units. If there are not enough hidden units to perfectly model the training distribution, one can observe a blurring effect: when two input variables are almost always similar to each other and sometimes radically different, the RBM will not capture this rare difference and will assign a mean value to both variables. When the appearance of image patches is being modeled,

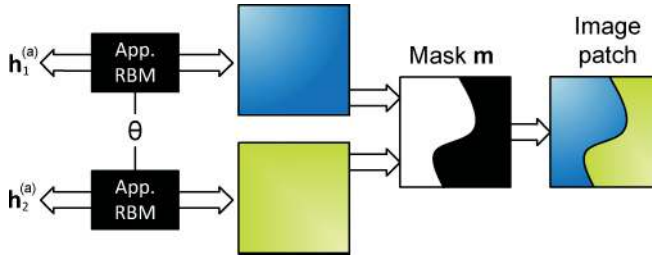

Figure 4: The Masked RBM. A masked RBM models an image patch as the composition of two or more latent patches, each generated from a separate appearance RBM with shared parameters θ. The composition is controlled by a mask $\mathbf{m}$, indicating which of the latent image patches is to be used to model each visible image pixel.

any two nearby pixels will exhibit this property (being different only when an edge is present between these two pixels), thus resulting in a poor generative model of image patches (as shown in the $K = 1$ case of Figure 6). To avoid this effect, a standard RBM would require a number of hidden units equal to the product of the number of possible locations for an edge and the number of possible appearances. Not only would that number be prohibitive, it would also be highly inefficient since the vast majority of hidden units would remain unused most of the time. A more efficient way to bypass this constraint of consistency within the data set is to have K appearance RBMs, each generating a latent image patch $\widehat{\mathbf{v}}_k$, competing to explain each pixel in the patch. Whenever an edge is present, one RBM can explain the pixels on one side of the edge, while another RBM will explain pixels on the other side. We say that such a model has K **layers**. To determine which appearance RBM explains each pixel, we introduce a **mask** with one mask variable per pixel (m_i), which can take as many values as there are competing RBMs. The overall masked RBM is shown in Figure 4 and its associated factor graph is shown in Figure 5.

In the remaining, we use the following notation:

- Since most of the equations will involve all the layers, we will define a short-cut notation: for any variable t defined for each layer k, the set of variables $\{t_1, \ldots, t_K\}$ shall be replaced by $t_{1..K}$.
- $\mathbf{v}$ is the image patch.
- $\widehat{\mathbf{v}}_k$ is the kth latent patch.
- $\mathbf{h}_k^{(a)}$ the hidden state of the kth layer. The (a) superscript stands for "appearance," as we will introduce shape layers later on.

Using these notations and given a mask $\mathbf{m}$, the probability of a joint state $\mathbf{s} = \{\mathbf{v}, \widehat{\mathbf{v}}_{1..K}, \mathbf{h}_{1..K}^{(a)}\}$ is equal to

$$P(\mathbf{v}, \widehat{\mathbf{v}}_{1..K}, \mathbf{h}_{1..K}^{(a)} | \mathbf{m}) = \left(\prod_i \delta \left[\widehat{v}_{m_i,i} = v_i \right] \right) \left(\prod_k \mathrm{APP}(\widehat{\mathbf{v}}_k, \mathbf{h}_k^{(a)}) \right), \qquad (3.1)$$

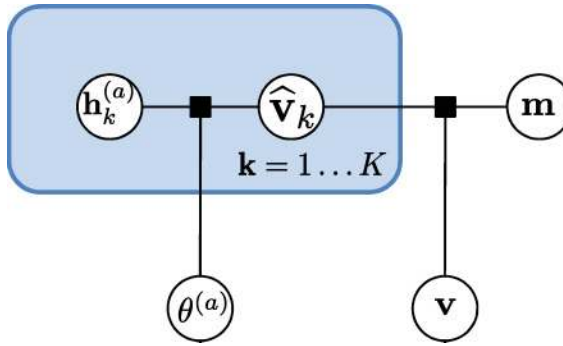

Figure 5: Factor graph of the masked RBM with a uniform mask prior. The joint distribution between the latent images $\widehat{\mathbf{v}}_k$ and corresponding hidden units $\mathbf{h}_k^{(a)}$ is modeled by an RBM with parameters $\theta^{(a)}$. $\theta^{(a)}$ is outside the plate and thus the same for all RBMs. The latent images are composed with a mask $\mathbf{m}$ to form the image patch $\mathbf{v}$.

where APP$(\widehat{\mathbf{v}}_k, \mathbf{h}_k^{(a)})$ is the joint probability of $(\widehat{\mathbf{v}}_k, \mathbf{h}_k^{(a)})$ under the chosen appearance RBM. The first term allows our model to assign infinite energy (and therefore zero probability) to configurations violating the constraint that if layer k is selected to explain pixel i (i.e. $m_i = k$), then we must have $\widehat{v}_{k,i} = v_i$.

To demonstrate the efficiency of using several masks, we infer the mask and hidden states of models with various K given an image and then reconstruct the image using the mask and these hidden states. The inference procedure is described in section A.1 of the appendix. For a fair comparison, we used the same total number of hidden variables for each value of K (accounting for the bits required to store the mask and the hidden units for each appearance model). The reconstruction with $K = 4$ thus used RBMs with many fewer hidden units ($n = 128$) than the one with $K = 1$ ($n = 1024$). From the results shown in Figure 6, we see that it is advantageous to assign a large number of bits to the mask rather than to the appearance. A more thorough evaluation of the masked RBM is presented in section 5.

4 Modeling Shape and Occlusion

Equation 3.1 defines a conditional distribution of the image, the latent patches, and the hidden states given the mask. To get a full probability distribution over the joint variables, we must also define a distribution over the mask. In this article, we consider three mask models: a uniform distribution over all possible masks, a multinomial RBM that we denote the softmax model, and a model that has been designed to handle occlusions, which we call the occlusion-based model. The latter two models will allow us to learn a model of the shapes present in natural images.

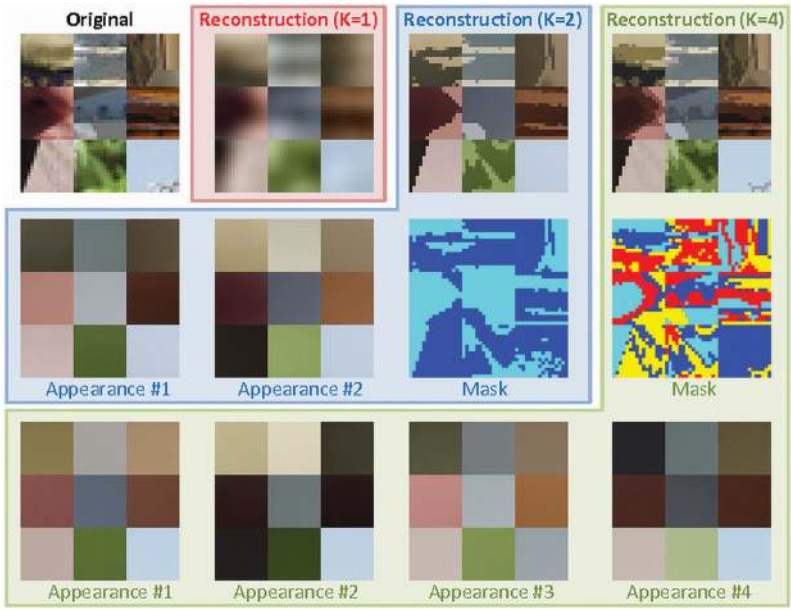

Figure 6: Reconstructions of nine patches using a masked RBM with $K = 1, 2$ or 4 appearance models. *Appearance #k* are the sets of nine latent images $\hat{\mathbf{v}}_k$, for all k's. When $K = 1$, the model is an ordinary beta RBM and is unable to capture sharp edges in the image. When $K = 2$ or $K = 4$ beta RBMs are used in a masked RBM, the reconstruction accuracy is much greater, and the masks capture the shape of the object in the image. The inferred masks and mean patch from each of the beta RBMs are shown. The experiment is detailed at the end of section 3. All models have the same total number of hidden variables.

The learning and inference procedures in these models may be found in appendix A.

4.1 The Uniform Model. The simplest mask model is the uniform distribution over $\mathbf{m}$. In this model, no mask is preferred a priori, and the inferred masks are solely determined by the image. We use this model as a baseline.

4.2 The Softmax Model. The softmax model consists of K binary RBMs with shared parameters competing to explain each mask pixel. Each RBM defines a joint distribution over its visible state $\mathbf{s}_k$, which is a binary shape, and its binary hidden state $\mathbf{h}_k^{(s)}$ (the (s) superscript stands for "shape"). The K binary shapes $\mathbf{s}_k$ are then combined to form the mask $\mathbf{m}$, which is a K-valued vector of the same size as the $\mathbf{s}_k$'s. To determine the value of m_i

given the K sets of hidden states $\mathbf{h}_k^{(s)}$, one needs to compute a softmax over the K different inputs. The joint probability distribution of this model is

$$P(\mathbf{m}, \mathbf{s}_{1..K}, \mathbf{h}_{1..K}^{(s)}) \propto \left(\prod_i \delta(s_{m_i,i} = 1) \prod_{k \neq m_i} \delta(s_{k,i} = 0) \right) \times \left(\prod_k \mathrm{SHAPE}(\mathbf{s}_k, \mathbf{h}_k^{(s)}) \right), \tag{4.1}$$

where $\mathrm{SHAPE}(\mathbf{s}_k, \mathbf{h}_k^{(s)})$ is the joint probability of $(\mathbf{s}_k, \mathbf{h}_k^{(s)})$ under the chosen shape RBM (a binary RBM in our case). The right-hand side of the equation is unnormalized due to configurations violating the constraints (e.g., $s_{k,i} = 0$ for all k).

The first and second terms state that only one shape may be "on" at any given pixel and that the index of the selected shape is the value of the mask at that pixel. Inference is relatively straightforward in this model, but at the cost of poor handling of occlusion. Indeed, this model makes the implicit assumption that all the objects are at the same depth. This gives rise to two problems:

1. When object A is occluding object B, the shape of object B is considered absent in the occluded region rather than unobserved. As a consequence, the model is forced to learn the shape of the visible regions of occluded layers. For example, with a digit against a background, the model is required to learn the shape of the visible region of the background—in other words, the inverted digit shape.
2. There is no direct correspondence between the hidden states of any single layer and the corresponding object shape, since the observed shape will jointly depend on the K inputs. In an object recognition system, this would reduce the ability to recognize an object by its shape, where the object is partially occluded.

4.3 The Occlusion Model. An occlusion occurs when an object is at least partially hidden by another one. In the occlusion model, we explicitly represent this hiding by introducing an ordering π of the layers ($\pi(k)$ being the position in the relative depth ordering of layer k, that is, $\pi(k) = 1$ indicates that k is the front-most layer and $\pi(k) = K$ indicates that k is the rear-most layer), where each layer contains a shape. For this shape to be visible, there must not be any other shape at the same location in the layers

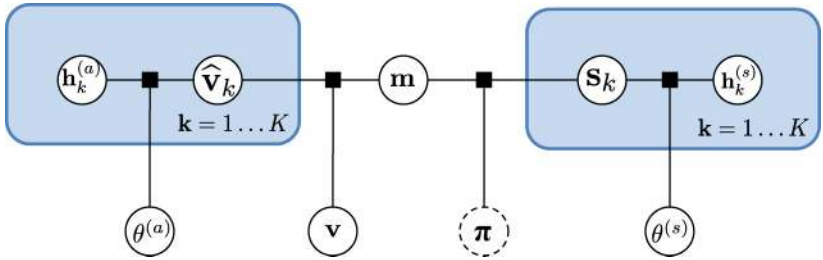

Figure 7: Factor graph of the masked RBM with a nonuniform mask prior. The joint distribution between the shapes $\mathbf{s}_k$ and the hidden shape states $\mathbf{h}_k^{(s)}$ is modeled by an RBM with parameters $\theta^{(s)}$. $\theta^{(s)}$ is outside the plate and thus the same for all RBMs. The ordering π is used only in the occlusion model.

above. The joint probability distribution for this model is:

$$P(\mathbf{m}, \mathbf{s}_{1..K}, \mathbf{h}_{1..K}^{(s)}, \pi) \propto P(\pi) \left(\prod_i \delta(s_{m_i,i} = 1) \prod_{k:\pi(k)<\pi(m_i)} \delta(s_{k,i} = 0) \right) \times \left(\prod_k \text{SHAPE}(\mathbf{s}_k, \mathbf{h}_k^{(s)}) \right). \quad (4.2)$$

The general factor graph corresponding to the masked RBM with nonuniform mask prior is shown in Figure 7. There are two main differences between the occlusion model and the softmax model:

1. We now have a prior $P(\pi)$ over the depth ordering (which is chosen to be uniform).
2. If $m_i = k$, then we must have $\mathbf{s}_{k,i} = 1$ (as in the softmax model), but we require only that $\mathbf{s}_{k',i} = 0$ for the layers k' in front of the layer k (rather than for all the layers, as is the case in the softmax model). $\mathbf{s}_{k'',i}$ for k'' behind layer k are unobserved (occluded). This idea is illustrated in Figure 8.

In the case of the occlusion model, there is a direct correspondence between the hidden states and the shape of the object (see Figure 10). Figure 9 specializes the general factor graph for the masked RBM with nonuniform mask prior from Figure 7 for the masked RBM with occlusion mask model and shows a schematic of the full model as a chain graph. The inference procedure for the depth ordering π is described in section B.1 in appendix B.

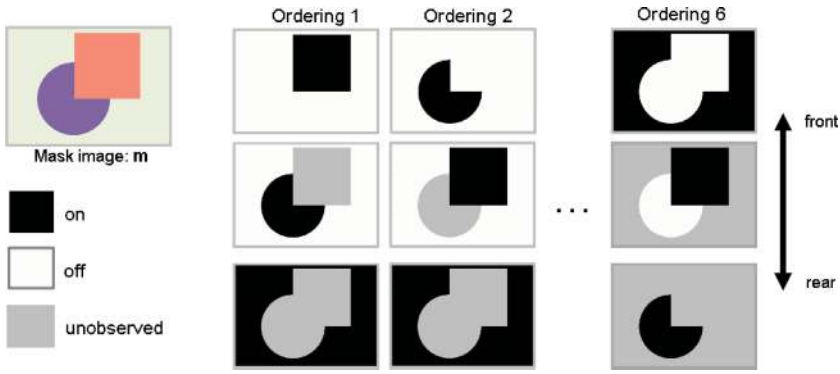

Figure 8: Depth inference in the occlusion model. The mask image (top left) comprises three regions, so there are $3! = 6$ possible depth orderings. Together with the mask, the ordering defines which shape pixels $\mathbf{s}_{k,i}$ are observed and which are unobserved. This is illustrated for three of the six possible orderings (white regions: shape off; black regions: shape on; gray regions: shape unobserved). Unobserved pixels (corresponding to $U_{\pi,k}(\mathbf{m})$ in equation B.1) can be filled in by the shape model. Thus, for a shape model that favors circles, squares, and homogeneous backgrounds, ordering 1 is preferable to all other orderings (including 2 and 6).

5 Inferring Appearance and Shape of Objects in Images

Our goal is to learn a good generative model of images by extracting a factorial latent representation (appearance and shape) of objects in natural images. To assess how well this goal is achieved, we seek to answer a set of questions:

- How visually similar are the samples from our model to samples coming from the same distribution as the training set? Although poor samples characterize a bad generative model, the converse is not true, as samples too close to the training data show a lack of generalization of our model, which is not desirable. Despite the flaws of this measure, we think it can provide meaningful insight on what has actually been learned.
- Do samples from our model exhibit the same statistics as those computed on test patches?
- Are test patches likely under our model?
- Did we really factor appearance and shape? Are the latent representations we extract meaningful? Are they independent of the depth ordering of the objects in the image? Are the depth orderings correct?

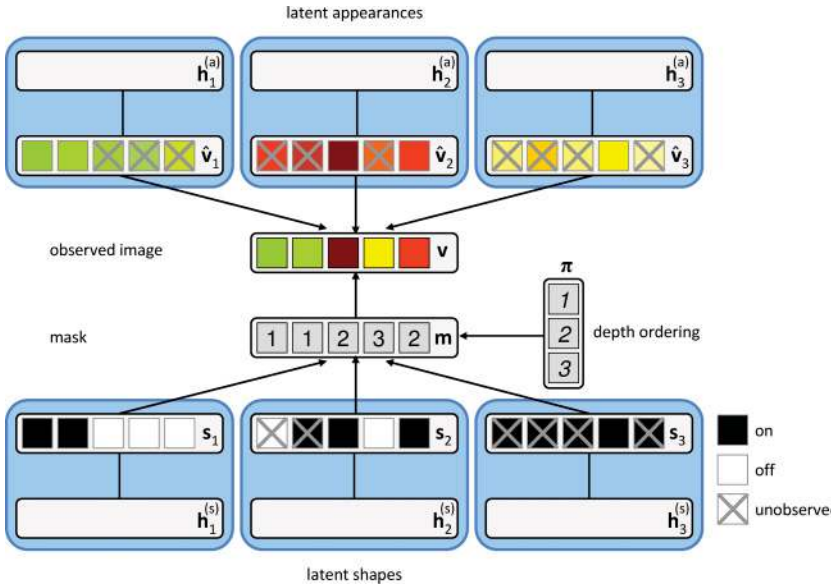

Figure 9: Schematic of the masked RBM with occlusion mask model. Specialization of the general factor graph from Figure 7 as a chain graph-like schematic for the masked RBM with occlusion mask prior. $K = 3$ instantiations of the RBMs for shapes and appearances are shown separately (these multiple instantiations are collapsed into plates in Figure 7; note that as shown in Figure 7, the parameters are shared between the multiple instantiations of the appearance and shape RBMs, respectively). Unlike the general factor graph, this figure distinguishes between undirected and effectively directed interactions between variables. In addition to the model structure, the figure shows a particular instantiation of the observed image ($\mathbf{v}$), the mask ($\mathbf{m}$), and the depth ordering (π). Given this particular instantiation of the mask, some of the visible units of the appearance RBMs ($\hat{\mathbf{v}}_k$) are tied to the pixels' values of the observed image; others are unobserved as indicated. Similarly, the mask, together with the depth ordering, determine which of the visible units of the shape RBMs ($\mathbf{s}_k$) are observed. Unobserved shape and appearance variables are filled in during inference, as explained in Figure 8. Note that the hidden units of the shape and appearance RBMs are not shown individually.

The first three questions relating samples from our model and test data can be answered on both a toy data set and a real data set of natural images. However, a toy data set offers the additional advantage of providing the ground truth objects from which the patches have been created, which makes it easier to assess the quality of the generative model.

The last questions are trickier to answer in the context of natural images since we have no control over the ordering of the objects. However, there are

some natural patches for which there is little ambiguity over that ordering. If the model is able to infer a plausible answer in these cases, this should be a good indicator of the quality of its inference of the depth ordering of the objects (and thus a measure of the invariance of the inferred latent shapes to this ordering).

5.1 Training. This section describes the training procedure for the masked RBM, as this model proved much more complicated to train than a standard RBM. Details on the data sets used are provided in the next sections. Additional details about the training procedure are contained in appendix D. The training was done in several stages of increasing complexity for efficiency reasons:

1. We first trained a single unmasked RBM until low-frequency filters appeared. This allowed us to quickly obtain a good initialization for the filters typically obtained in the masked RBMs (since the edges are captured by the masks, none of them are high frequency) by avoiding having to infer the mask at each iteration;
2. Initializing with the filters from the previous step, we then trained a masked RBM with a uniform mask model (which means we trained only the appearance RBM) and $K = 2$. Using a lower K allows us to speed up inference while still providing good initial filters for the final stage. K was then switched to 4 until parameters converged. The reason that we trained the appearance model in the context of a masked RBM is to avoid wasting capacity in modeling complicated shapes that will be handled by the mask.
3. We froze the parameters of the appearance RBM and inferred an initial segmentation (mask) of our training data. We used the binary region shapes extracted from the masks to pretrain the binary RBM of the shape model.
4. We trained the shape model in the context of the full masked RBM, performing joint inference of the shape, appearance, mask, and depth, with an occlusion shape model using the binary RBM trained in the previous step as initialization.
5. We fine-tuned both the appearance RBM and the shape RBM by performing the joint inference of the parameters of both models (the masks being inferred at each iteration using the current state of the RBMs), using the correct shape model.

Bootstrapping allowed faster learning of this complex model. Also, experiments seemed to indicate that it helps to find a better global solution and avoid undesirable local minima.

5.2 Toy Masks Data Set. The toy masks data set is composed of 4000 14×14 mask patches generated from the superposition of an MNIST digit (from class 3) and a shape (a circle, a square, or a triangle). In this data set,

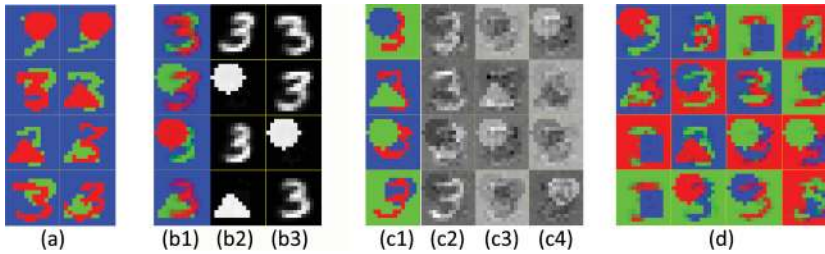

Figure 10: Learning shapes under occlusion. (a) Training data. (b1) Samples from the occlusion model (20 hidden units per layer), obtained by composing latent shapes (b2 and b3). (c1) Samples from the softmax model with 70 hidden units (left-most column) using contributions of the three layers (c2, c3, and c4). (d) Samples from the softmax model with 20 hidden units per layer. The softmax model can compensate for its limitations by using more hidden units, but its performance quickly deteriorates when it has limited capacity, yielding invalid samples (in the bottom right sample, the "3" goes through the square).

neither digits nor shapes are shown in isolation, and each digit example appears in exactly one image. Since the digit is in the background on half of the patches, half of the digit examples are only partially visible. Samples of this data set (which are masks) are shown in Figure 10a: each pixel can take three values (represented by the colors red, green, and blue)—one for each object in the patch (the background being the third object). Which color is assigned to each object is irrelevant (the actual values are not used to infer the depth ordering); it matters only that they are assigned different colors.

5.2.1 Quality of the Generative Shape Model. We trained our mask model using three layers ($K = 3$). Figure 10 shows samples from the occlusion model with 20 hidden units (b), the softmax model with 70 hidden units (c), and the softmax model with 20 hidden units (d). Samples from the occlusion model are drawn by sampling from the two RBMs governing the top-most and second-most layer independently and then composing these samples, as prescribed by equation 4.2. One can see that when 20 hidden units are being used, the samples drawn from the occlusion-based mask model are much more convincing than those drawn from the softmax model. Indeed, the latter generated samples with improper occlusions or deformed digits. It is also interesting to note that the occlusion model generalized to samples not seen in the training set, like the two MNIST digits that occlude each other. Furthermore, columns b2 and b3 show samples of the latent shapes, proving that the occlusion model learned a model of the individual shapes—despite the fact that it has never seen them in isolation.

In the softmax model, the layers cooperate to generate a particular image of occluding shapes. It is not possible to sample from the individual layers separately, but one can still inspect the inputs to the three layers of visible

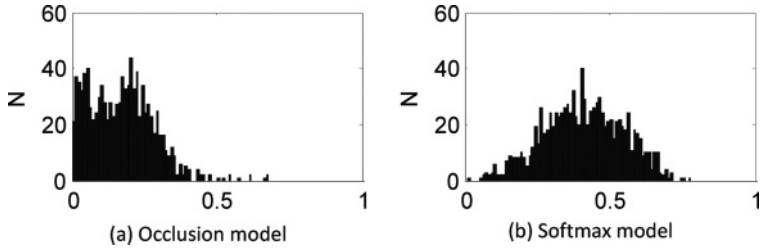

Figure 11: Histograms of the root mean squared differences between the activation of the hidden units inferred depending on the relative positions of the MNIST digit and the shape in the test image for the softmax model and the occlusion model.

units that are tied together by the softmax. These inputs are shown in Figure 10 (c2, c3, and c4). It is clear that no shape is generated by a single layer but that all three layers have to interact. In the first row, for instance, all three inputs contain a 3 (with either positive or negative weights) although it is absent from the resulting sample. Though harmful (because they require additional modeling power), these cancellations are inevitable in the softmax model. While the occlusion model learns about the individual image elements, the softmax model has to represent all their possible arrangements explicitly, which is less efficient and thus requires a larger number of hidden units. This also leads to a set of hidden units, which is far less indicative of the shape in the image than in the occlusion model.

5.2.2 Sensitivity to Occlusion. To assess the importance of the difference in representation between the softmax and the occlusion mask models, we created pairs of images containing one digit and one shape (the same digit and the same shape were used in both images of a pair). In the first image, the digit was in front of the shape, and in the second image, the shape was in front of the digit. We compared the inferred shape latent variables for the two cases and computed their root mean squared difference. Because our main motivation is to recognize objects whether or not they are occluded, we would like the shape latent variables to be as similar as possible in the two cases. Unsurprisingly, the occlusion-based mask model clearly outperforms the softmax model, as may be seen in Figure 11. Furthermore, in our experiments, the occlusion model inferred the correct ordering more than 95% of the time (chance being 17%, as there are three layers and six possible orderings).

This toy data set emphasizes the need for modeling occlusion when extracting a meaningful representation of the shapes present in images.

5.3 Natural Image Patches. The experiments on toy data demonstrated that the occlusion model is able to learn and recognize shapes under

occlusion and is able to perform depth inference given a mask image with occluding shapes. The second set of experiments on natural images assesses the joint model consisting of the shape and the appearance model. For this purpose, we trained the full model with $K = 3$ on 21,000 16×16 patches extracted from natural color images. The mask model used in all these experiments is the occlusion model. The appearance RBM had 128 hidden units, and the shape RBM had 384 hidden units. (Details on the training procedure can be found in section D.2 in appendix D.)

As outlined above, our criteria for assessing the model on this data set were:

- Whether samples from the model looked qualitatively similar to the natural image patches that we had trained the model on (see section 5.3.2)
- Whether samples from the model exhibited the same statistics as natural image patches (see section 5.3.3)
- Whether inference on natural image patches would give plausible results (see section 5.3.4)

5.3.1 Sampling from a Confident Continuous-Valued RBM. When learning the appearances of the objects with the beta RBM, each expert becomes extremely confident. This is even more striking in the masked context, where the noise model does not need to explain the sharp variations of appearance at the boundaries of objects. While this is a good thing from a generative point of view, it leads to a very poor mixing of the Gibbs chain. Indeed, as the conditional distributions $P(\mathbf{v}|\mathbf{h})$ become very peaked, so do the distributions $P(\mathbf{h}|\mathbf{v})$, and the relationship between $\mathbf{v}$ and $\mathbf{h}$ becomes quasi-deterministic. This makes it hard to:

- Learn the parameters in the final stage, as the samples from the negative chain are highly correlated between consecutive time steps
- Draw samples to assess the quality of the generative model
- Compute an accurate approximation to the partition function to estimate the log probability of test patches

The first issue was dealt with by using tempered transitions (Salakhutdinov, 2009) twice per sweep through the training set. To improve sampling, we trained a binary RBM on top of our beta RBM. Because such RBMs mix much more easily, we could draw samples by running a Gibbs chain in this top binary RBM before performing a top-down pass in the bottom beta RBM. Unfortunately, even then, annealed importance sampling (AIS) (Salakhutdinov & Murray, 2008) proved unreliable. We therefore decided not to include log-probability results whose validity we could not properly assess.

It is worth emphasizing that the inference of the hidden variables given the visible ones does not suffer from these issues (it is still fast and exact),

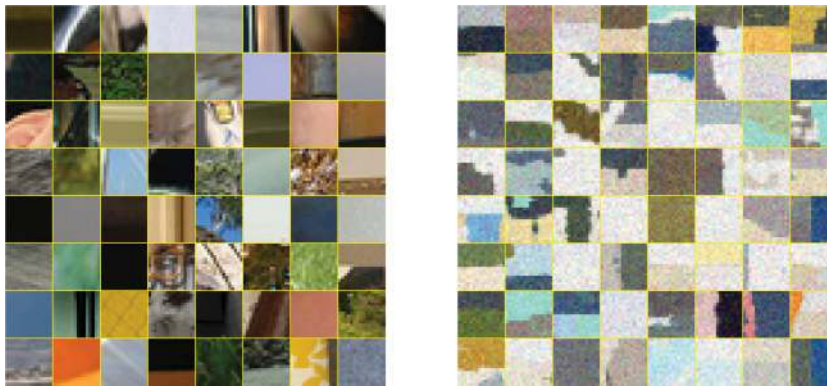

Figure 12: True natural patches (left) and samples from the masked RBM (right).

nor does the optional learning of a layer above (since it will then deal with binary data).

5.3.2 Visual Assessment of the Samples. Sampling from the mask model was performed by sampling the binary RBMs in the shape layers (15,000 steps of Gibbs sampling) and composing them according to a randomly chosen depth ordering. Masks were then combined with samples from the appearance model (5000 steps of Gibbs samplings). The full samples from the masked RBM are shown in Figure 12 (right). Although they do not exhibit as much structure as true natural image patches (see Figure 12, left), the presence of multiple sharp edges makes them look much more convincing than the typical blurred samples one may obtain from a single RBM. Moreover, the samples clearly capture important characteristics of the training patches (such as the dominance of homogeneous regions and the shape of the boundaries of these regions), despite the relative simplicity of the model and the fact that K was chosen to be small.

5.3.3 Image Statistics. We assess the quality of the samples from the masked RBM by comparing the statistics of responses of different types of filters (even and odd Gabor filters and random zero-mean filters) with the statistics of real image patches. Before computing the filter responses, we converted all the patches to grayscale. We compared four kinds of patches:

- Natural patches.
- Patches sampled from the masked RBM. The appearances and the shapes are true samples from the model. This model used $K = 3$ layers.
- Patches sampled from a single, unmasked RBM.

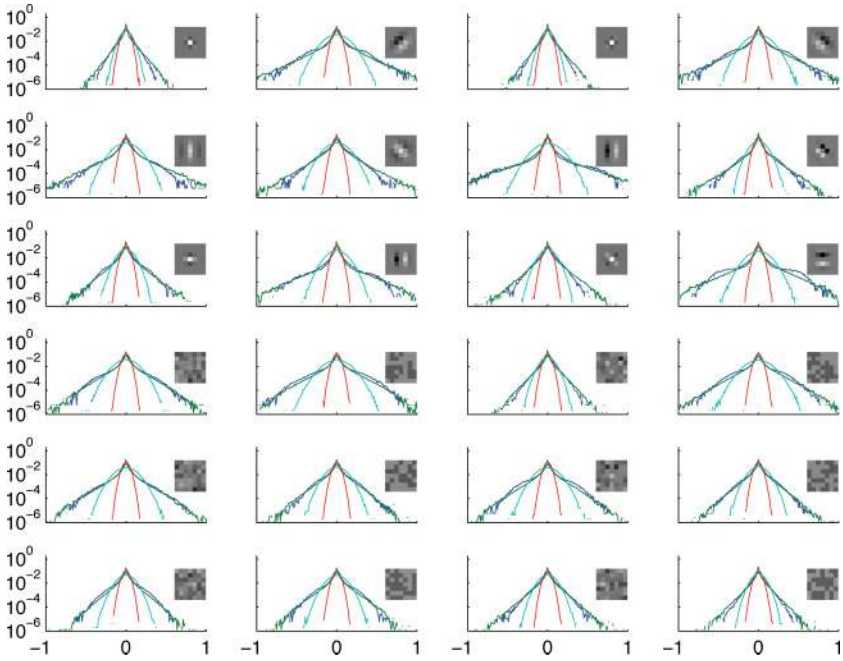

Figure 13: Filter responses for various kinds of patches. Green: real image patches. Blue: samples from the masked model with $K = 3$. Red: samples from the appearance model only (no shape). Cyan: Gaussian noise with the same covariance as the real patches. For each histogram, the corresponding filter is shown as an inset. Whereas the samples generated from a single RBM exhibit a gaussian-like response, the response obtained from samples from the masked RBM closely matches those obtained from real image patches.

- Patches generated from gaussian noise with the same covariance as natural patches.

The results (displayed as log probability of each response value) are shown in Figure 13. For all filters, the response histograms of samples from the masked RBM (in blue) have much heavier tails than those for patches sampled from the unmasked RBM (in red) or the gaussian model (cyan), but they are similar to the responses obtained from real image patches (green). There is one systematic mismatch between natural image patches and the samples obtained from the masked RBM. Due to the pixel-independent noise model, the peak of the histograms at 0 is underestimated for the samples from the masked RBM (this is because nearby pixels have an extremely low probability of having the same value, unlike true image patches). However, if we replace samples from the appearance model with the mean activations of the visibles given the binary hiddens in the last step of the Gibbs

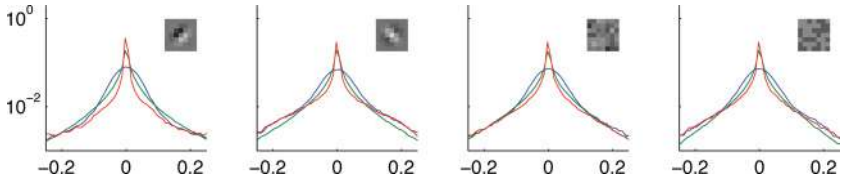

Figure 14: Difference between sampled and mean activations in a zoomed-in region close to the peak for the first four Gabor filters. Green: real image patches. Blue: samples from the masked model with $K = 3$ where the activations of the visible units have been sampled given the binary hidden states. Red: samples from the masked model with $K = 3$ where the activations of the visible units are the average of the activations given the binary hidden states. Due to the smoothness induced by the averaging, the peak at 0 is much more pronounced and is much closer to the one obtained with real image patches. Similar results were obtained for the other Gabor filters.

chains[4] and use those when composing the full, layered samples from the masked RBM, we get the filter responses shown in Figure 14 (only the region near the origin is shown). The tails remain the same, but the peak at 0 is more pronounced, closely matching the ones obtained with true image patches. We emphasize that the model has never been trained directly to match the statistics of natural images. Nevertheless, it reproduces some of their distinguishing features quite reliably. The improved matching, in particular the heavy tails, arose naturally with the use of a mask.

5.3.4 Inference of Relative Depths Based on Shape. The goal of this experiment is to investigate whether learning an efficient representation of the data leads to the model being able to reason about image regions and relative depths. For this purpose, we chose a simple scenario shown in Figure 15: patches that contained simple shape-based depth cues were extracted from an image (a). For each patch, the model inferred a segmentation mask with up to $K = 3$ regions (b.1), a relative depth ordering (front to back: red—green—blue), the potentially partially unobserved shapes of the two rear-most layers (b.2), and the appearances of the three layers. The inferred latent shapes allow removing the foreground shape and imputing the missing parts of the second layer shape (c.1 and c.2: segmentation mask with two layers and imputed image, respectively). For the examples shown, the model inferred segmentations, depth orderings, and latent shapes largely consistent with the full image.

[4]That is, we run a Gibbs chain in the appearance RBM for the same amount of time (5000 steps) sampling visibles and hidden units in each step. Only during the last step do we take the mean activation of the visible units given the binary hidden states (rather than a sample).

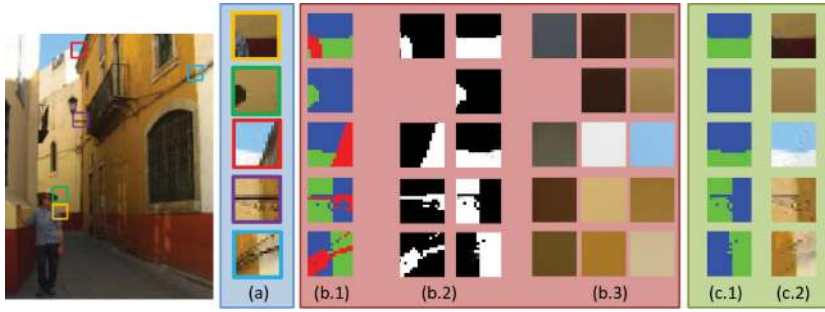

Figure 15: Starting from natural patches (a), we inferred the ordering of the layers (b.1), the latent shapes of the two front-most objects (b.2), and the three latent images (b.3). This allows us to recreate a mask image without the foreground (c.1) and the associated patch (c.2).

Inferring relative depth using very local shape information only (such as provided by our 16 × 16 patches) is a highly ambiguous problem in the general case—not just for a computational model but also for human observers. The fact that the model is able to perform such a task at all might be surprising considering that it has been trained on only individual image patches without any built-in prior (e.g., about smooth boundary shapes) or additional information, such as the context (the larger shapes that the fragments in the patch are part of), stereo data, or temporal information. Nevertheless, there are at least two plausible cues acquired by the model during training that are driving the results in Figure 15. One relatively naive cue the model uses is that it prefers to place smaller regions in the foreground. More important, however, it also prefers to explain image patches in terms of extended, roughly horizontal or vertical shapes. This behavior is rather robust and observed for all five examples in Figure 15, particularly for patch 3. It allows the model to complete the occluded shapes in a plausible manner and thus drives depth inference. We provide an evaluation of this phenomenon on a larger data set in 8 appendix E. Here, results cannot be easily explained in terms of region size, and we find the model to be in qualitative agreement with human observers (although we would not like to claim that the model matches human performance in general). This behavior seems reasonable given that such roughly horizontal and vertical shapes are particularly frequent in our training data so that representing, say, patch 3 in terms of such shapes is a likely explanation in light of these training data. Thus, learning an efficient representation of the data also has made the model pick up certain simple depth cues despite never having received any kind of depth information with the training data.

There are currently two main limitations to the model. First, the model has difficulties in correctly segmenting image patches that exhibit matting

or shading since this is not accounted for by the model. Also, the model currently does not have a suitable prior over the number of regions, so it has a tendency to oversegment patches that have fewer than K coherent regions (such as the second patch in Figure 15). Incorporating such a prior effectively corresponds to model selection and is nontrivial since we cannot compute the normalization constant of either the appearance or shape RBM, but we are currently working on suitable approximations.

5.4 On the Use of an RBM for Modeling Appearance. Looking at the very smooth latent patches of Figure 6, one may wonder if RBMs are the right model to use for appearances since they do not seem to be able to model complex textures.

First, we recall that some of the advantages of RBMs are the ease with which they can be trained, the speed of inference, the convenience of the distributed representation of the data, and their ability to be easily stacked into deeper structures, which will be important for the future hierarchical formulation of the model outlined in section 8. Also, provided that the number of hidden units is large enough, they can model more complicated structures, as shown in Figure 6 when $K = 1$. Thus, the choice for simpler RBMs stemmed from the observation that it is much more efficient (in terms of the quality of the reconstruction) to assign bits to the mask rather than to the appearance. Finally, many natural image patches in our data set were simple enough so that one did not need to use four latent appearances, but since there is not yet any procedure to select K automatically, this resulted in an oversegmentation of these patches during training, yielding overly smooth patches.

Thus, although the RBMs we used did not model complex textures (which were then accounted for by the mask), this would not necessarily be the case in other models or with larger RBMs (in terms of the size of the hidden layer), resulting in the mask's capturing changes only in such textures.

6 Field of Masked RBMs

When modeling image patches of size 16×16, we made the assumption that they were composed of K patch models of size 16×16 fully aligned with each other and with the image patch. As a consequence, each pixel in a patch can be explained by any one of the K different patch models. In order to move from image patches to entire images, we could use a larger number of bigger patch models (which would be the size of the image rather than 16×16). However, that would be very expensive (especially the depth inference in the occlusion model) and inefficient since this would not model translation invariance. Instead, we will again use patch models of size 16×16, which will be laid out across the image, partially overlapping each other. Of course, in the case of large images, the total number of such

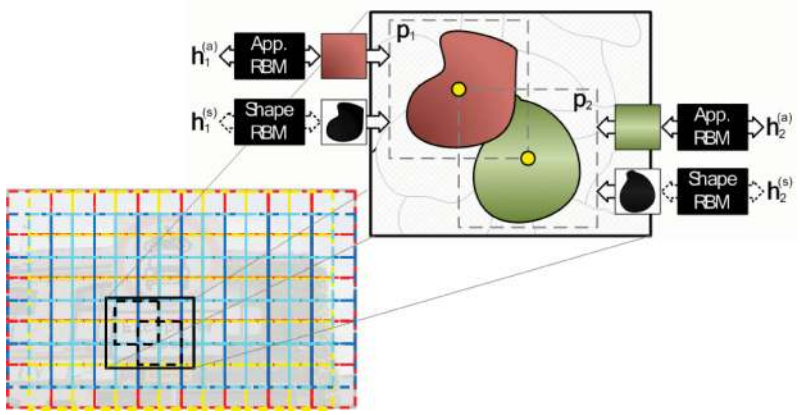

Figure 16: A field of masked RBMs, where an image is represented using a set of overlapping patch models. (Left) The image is covered by K (here $K = 4$) grids of non-overlapping, abutting patches (each grid is shown in a different color: red, yellow, cyan, blue). The different grids are spatially offset so that the patch boundaries in different grids do not align, and each pixel is covered by K partially overlapping patches that compete to explain the pixel. (Right) Blow-up of the interaction between two overlapping patch models. Competition between patch models leads to a segmentation of the image into superpixels, with one superpixel per patch. The appearance and the shape of each superpixel are modeled by separate RBMs.

patch models is much greater than the number of patch models any one pixel can be explained by.

A simple way of covering an entire image with these patch models is to tile it into a set of nonoverlapping image patches and model each such patch with a masked RBM, as in section 4. However, this approach leads to artifacts at the patch boundaries, since correlations between pixels on either side of these boundaries are ignored. These artifacts appear because the K patch appearance models that each pixel chooses between are aligned, so that their patch boundaries are in the same place. Moreover, and perhaps more important, the only translation invariance we get is very coarse (our model would be invariant to translations of 16 pixels or multiples thereof). A better solution is obtained if we spatially offset the patch models so that no two patches are fully aligned. One such arrangement is shown in Figure 16. Here, the image is tiled by K grids of patch models. In each grid, the patches are nonoverlapping and cover all pixels in the image. Across different grids, the patch boundaries are spatially offset horizontally or vertically by half the patch size so that no two patches are fully aligned. This model allows finer translational invariance. For instance, with $K = 4$ and a patch size of 16×16, the patch boundaries are offset by 8 pixels; thus, it is invariant to

translations of 8 pixels or multiples thereof. It should be noted that although we colored all the patch models belonging to one grid with the same color in Figure 16, patch models belonging to the same grid are in no way more related than patch models belonging to different grids.

Thus, the image is covered with partially overlapping appearance RBMs (and possibly corresponding shape models), arranged such that each pixel is covered by exactly K RBMs. Figure 16 shows a field of masked RBMs, with two of the overlapping appearance RBMs highlighted. The set of mask variables now forms a mask image with a value for each image pixel indicating which of the K overlapping models it is explained by (bear in mind that the same value of K represents different superpixels across the image, since at different positions in the image, we have different superpixels in the kth grid). It should be noted that this model is a mixture rather than a product of appearance RBMs, in contrast, for instance, to the field of experts model of Roth and Black (2005).

Inference is done in the same way as at the patch level, with the one difference being that the patch models competing for a particular pixel are no longer aligned. This introduces long-range dependencies between spatially separated patches, so that inference has to be performed on the entire image simultaneously. While this makes perfect sense from a probabilistic point of view (in the general case, one has to take the whole image into account to understand part of it), the result is slower learning and inference.

Figure 17 is the equivalent of Figure 6 for full images. It shows the reconstruction of an image (i.e., the image generated using the hidden states inferred from the original image) using various numbers of layers and a uniform mask model. As for the experiments depicted in Figure 6, we used the same number of hidden variables for each value of K (4 bits per pixel). The RBMs used in the appearance model with $K = 4$ thus have only 128 hidden units, whereas those used in the model with $K = 1$ have 1024 hidden units. The patch size is 16×16 pixels for all K.

Both shape models discussed in the previous section (the softmax as well as the occlusion model) can be used at the image level. Figure 18 shows that using such a shape model yields more coherent regions for the mask image without significant loss in reconstruction accuracy. The occlusion model leads to a particularly appealing interpretation at the image level: each patch model can be thought of as an independent expert modeling shape and appearance of an image patch. It consists of an appearance RBM that determines the color—or, more generally, texture—of a patch and a binary RBM that determines its shape, as is illustrated in Figure 16. An image is generated by covering it fully with such patches in an occluding manner. This generative process bears some resemblance to the dead-leaves model (Lee, Mumford, & Huang, 2001), although there are important differences (e.g., in the current formulation of our model, the number of occluding objects covering an image is fixed and their maximum size is restricted while we allow for complicated and diverse shapes and appearances of individual

Figure 17: Reconstructions of an image under a field of masked RBMs with differing numbers of appearance layers. Each appearance layer is represented by a grid of nonoverlapping beta RBMs. In each case, the reconstruction uses 4 bits per pixel. The reconstruction quality is highest for $K = 4$, indicating a good trade-off between representing appearance and shape of objects in the image. In the mask images, all superpixels belonging to the same grid have the same color.

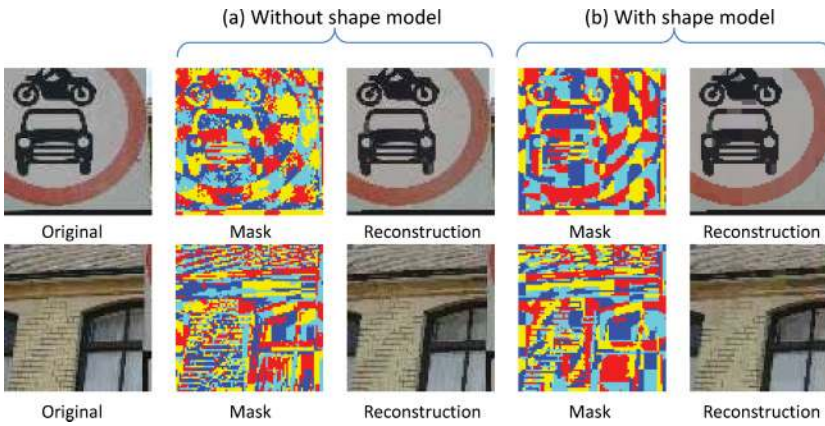

Figure 18: Field of masked RBMs with and without RBM softmax shape model. (a) Without a shape model, the mask gives the best reconstruction RMSE but is not coherent. (b) With a shape model, the inferred mask is much more coherent, while preserving thin structures like brick patterns. Reconstruction quality is very close to the one of *a*.

objects). In particular, and perhaps surprisingly, inference with the occlusion model can still be performed efficiently for full images: even though each image is explained by a potentially large number of patches, each individual patch overlaps with only a small number of neighbors (e.g., for $K = 4$ and the global patch layout shown in Figure 16, each patch overlaps with eight neighbors). Thus, instead of determining a global depth order of all patches (which would clearly be infeasible), it is sufficient to infer the depth of each patch relative to its neighbors. The depth of a particular patch given a fixed relative order of its neighbors can be determined following the principles described for image patches in section 4.3; the full local ordering of all patches covering the image is determined in an iterative manner by considering each patch in turn (see the appendix for details).

6.1 Evaluation on Shape Data Set. Although the reconstruction of test data gives some information about the quality of a generative model, it has severe shortcomings. We thus repeat some experiments done at the patch level to show how the main properties of the algorithm have been preserved despite operating at the image level.

We start by assessing the validity of our model on toy data. We focus our attention on three components:

- The allocation of objects to masked RBMs. Namely, are objects fully captured by the RBM they are centered on? Are RBMs explaining only parts of objects?
- How robust is the depth inference between overlapping objects?
- How good are the shape and appearance models learned using entire images?

For this purpose, we trained our field of masked RBMs with $K = 4$ on 100 80×80 images (each image composed of 144 overlapping 16×16 patches) composed of five different shapes with varying colors placed randomly in an overlapping fashion against a uniform background (see Figure 19, left, for an example; note that shapes were aligned with the patch grid). We allowed 20 hidden units for the shape model.

After training, we verified whether the shape model had indeed learned about the shapes comprising the images by sampling from the binary RBM directly. A selection of random samples is shown in Figure 21. Indeed, even though most shapes are only partially visible in the training images (and have varying colors), the shape model has recovered the five template shapes correctly. Figure 19 (right), shows the segmentation inferred with the fully trained model for the image shown on the left. Yellow outlines show the boundaries of objects captured by each masked RBM (patch model). These boundaries indeed reflect the shapes comprising the image (note that the background is segmented in a largely arbitrary manner). Segmentation is obviously not a very difficult task given the image at hand. More interesting is the simultaneously inferred relative depth of the different image

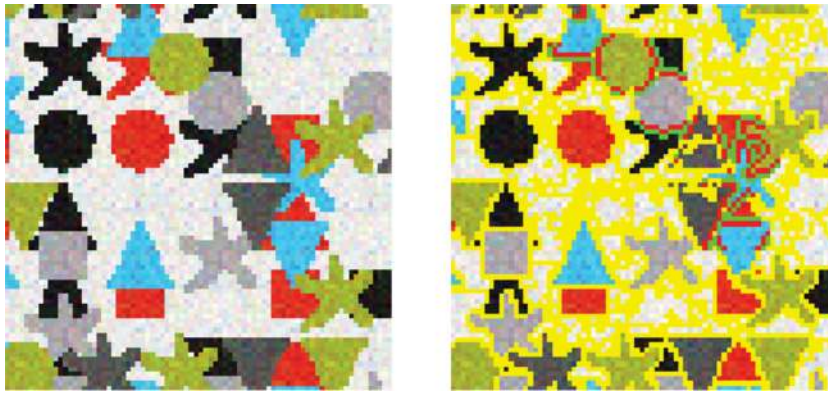

Figure 19: (Left) Training RGB image for the field of masked RBM. There are five different shapes and five different colors. No two overlapping shapes have the same color. Most of the shapes are explained by only one superpixel. (Right) Inferred segmentation and depths. Areas explained by a given superpixel are delimited by yellow lines. In several cases, the inferred relative depth is also displayed: the red line is on the inside of the shape, and the green line is on its outside. Therefore, when two shapes overlap (e.g., the green circle and the blue triangle at the top of the image), the red line of the object in the back is cut by the red and green lines of the object in the front (in this case, the blue triangle is in the back). The model inferred the correct depth ordering for all the shapes.

regions and the latent representation inferred for each patch model. The relative depths are shown for a subset of segmentation boundaries, which are double-marked with red and green lines. The red side of the boundary points toward the region that has been inferred to be in front and the green side toward the one that is inferred to be in the back. Figure 20 further shows the inferred latent shape and appearance for two of the patch models representing the image (indicated by the blue squares). In both cases, the true shapes (a gray star and a red triangle) are barely visible in the image (see also Figure 19). Nevertheless, the model correctly infers the appearance and, importantly, completes the partially occluded shape (see Figure 20). It is this ability to correctly complete occluded shapes that drives the depth inference.

6.2 Natural Images

6.2.1 Inference on Natural Images: Interpretation as a Superpixel Algorithm. The field of masked RBMs learns to represent an image as a number of regions, each explained in terms of an appearance and a shape. These regions can be thought of as superpixels, although they differ from previous kinds of superpixels in that they are not required to be contiguous but

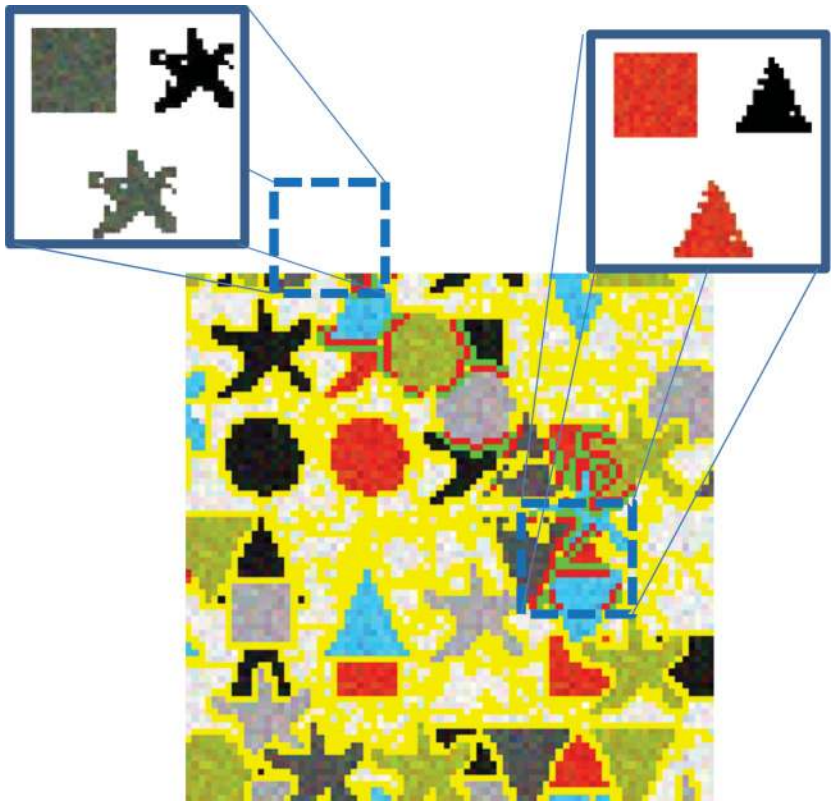

Figure 20: Inference of the shape and appearance of two barely visible objects (one being occluded by another object, the other sitting at the edge of the image). For these two objects, the inferred latent representations are shown in the corresponding insets. The appearance and shape are shown separately in the top left and top right panel of each inset. The bottom panel shows the combined shape and appearance. The inferred shapes are very close to the true underlying shape.

merely constrained to lie within the boundary of a patch. Also, they have high-order shape priors that have the potential to capture complex shapes, such as digits or letters. Such noncontiguity makes particular sense when dealing with occlusion, since the same superpixel can be used to represent parts of an object on either side of a narrow occlusion.

This behavior is illustrated in Figure 22, which shows the equivalent of Figures 19 and 20 for the natural image in Figure 17. It shows the segmentation inferred by a field of masked RBMs with occlusion shape model (see section D.3 in appendix D for details on how the model has been trained)

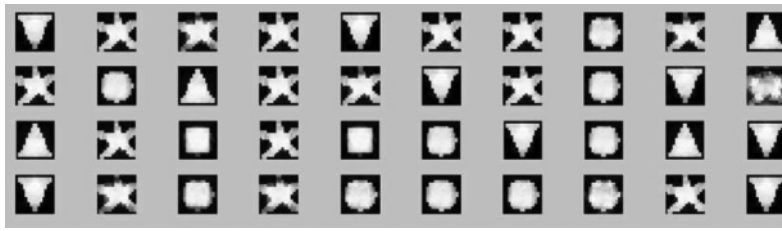

Figure 21: Samples generated from the shape model learned using the training image from Figure 19, after running a Gibbs sampler for 5000 steps. The images shown are the probabilities of the binary visible units given the binary states of the hidden units. Though most of the shapes in the training data are partially occluded, the model learns to generate complete shapes. One can see that the model has some difficulties distinguishing the square from the circle.

together with the corresponding latent representation of all 1218 patch models (of size 16×16 pixels) covering the image. For each superpixel, the combined latent shape and appearance are shown (cf. Figure 20). For the toy data considered in the previous section, the model was confident with respect to the shapes composing the image and with respect to their relative depth. In contrast, for real images such as the one considered in Figure 22, there is considerably more uncertainty as to what a suitable decomposition of the image would be. Not only are relevant regions typically significantly larger than the extent of the individual patch model, but there is also an enormous variability of shapes in natural images. With only the very local information available to the model, a decomposition in terms of high-level components of the scene cannot necessarily be expected. Nevertheless, the decomposition of the image that is inferred by the model appears largely sensible: in particular, it has a tendency to explain the image in terms of small shapes, especially thin horizontal and vertical structures, that appear in front of larger homogeneous backgrounds. This is very noticeable when focusing, for instance, on the representation of the various signs in the image ("Except for access," "ral Service," "TY Ltd," and the "no parking" sign), where the letters have largely been separated out and are placed in front of mostly contiguous background superpixels. Note that due to the explicit representation of occlusions, superpixels in the rear do not have to model the cut-out shape of foreground superpixels (even though there are some counterexamples, e.g., the "x" and "c" in "Except" are being explained in terms of a black background of unspecific shape behind a light gray foreground that has the letter shape cut out). Other examples are the frames of signs and windows that have predominantly been explained in terms of thin horizontal and vertical structures with often larger superpixels in the rear. To facilitate the mapping between the two representations, we have color-coded superpixels in both subfigures, representing letters in


Except for
access
al service

Figure 22: Inferred segmentation and latent representation of superpixels. (Top) Inferred segmentation of the test image shown in Figure 17. Areas explained by different superpixels are separated by yellow lines. Some superpixels are color-coded, as explained below. (Middle) Inferred latent representation of all superpixels covering the image (there are 1218 partially overlapping superpixels of size 16×16 pixels laid out, as explained in Figure 16). Each grid cell corresponds to one superpixel. Superpixels are arranged according to their relative position in the image. Note that superpixels that are horizontally or vertically adjacent in the grid overlap in the image by 16×8 and 8×16 pixels, respectively, while diagonally adjacent superpixels overlap by 8×8 pixels, as shown in the close-up in Figure 16. Superpixels that are separated by one cell in the grid are adjacent (nonoverlapping) in the image. For each superpixel, the inferred latent appearance is masked by the inferred latent shape as for the toy data in Figure 20. The model has a tendency to explain the image in terms of small shapes, especially thin horizontal and vertical structures, that appear in front of larger homogeneous backgrounds. This is reflected, for example, by the latent representation of the various signs in the image, for which the letters have largely been separated out correctly (superpixels are color-coded in red) and are inferred to be in front of mostly contiguous background superpixels (color-coded in blue) and also in the representation of the frames of signs and windows, which have predominantly been explained in terms of thin horizontal and vertical structures (some superpixels are color-coded in green) with typically larger superpixels in the rear. Note also how the set of noncontiguous image regions corresponding to the bricks in the wall is represented by superimposing a fine grid of mortar (superpixels colored in purple) onto a small number of larger brick-colored (latent) shapes. (Bottom) Reconstruction of part of the image from the inferred mask (left) and the inferred latent shapes (right). To illustrate that "background" superpixels are filled in beneath the foreground structure, we reconstruct the image using the inferred appearances, shapes, and depths for each superpixel, ignoring all superpixels corresponding to letters (color-coded in red in the top and middle panels). For the left-hand figure, we use only the visible parts of the shapes of the superpixels (corresponding to the mask represented by the segmentation outline in the top panel). This means that pixels belonging to the letters are missing in this reconstruction (highlighted in blue). In the right-hand figure, we reconstruct the image using the inferred latent shapes, as shown in the middle panel. These inferred shapes are larger than the visible parts of the superpixels, so they partially occlude each other. Since the letters had been inferred to be in the foreground, the model was able to largely fill in the missing pixels in the background superpixels. Accordingly, many fewer pixels are missing in the reconstructions (there is some noise from sampling the unobserved parts of the shapes). Note also that some structure in the background arising from shading of the sign that is partially occluded by the letters has been completed in a meaningful manner (purple arrow).

Figure 23: Structure inpainting in images. The figure demonstrates the value of the shape model on a simple structure inpainting task that requires knowledge about the shape of boundaries in natural images. (Left) Input image. 80×80 pixels with seven regions of "unobserved" (missing) pixels (unobserved pixels are colored in blue; the average size of each region is more than 26 pixels). Regions of unobserved pixels were chosen so as to overlap with region boundaries in the image. Full inference was run on the input image for 200 iterations (inferring the mask as well as shape and appearance fantasies and the depth order; each iteration corresponds to one full update of all latent variables) treating pixels in the blue regions as unobserved. (Middle) At the end of the inference, the unobserved pixels were filled in using the inferred latent shape and appearance fantasies. Note that the model's fills in the unobserved parts of the image largely correctly, continuing the boundaries of image regions in a plausible manner. This relies on the shape model's having acquired knowledge about plausible region shapes during learning. (Right) Pixelwise average of the reconstructions obtained during the last 100 iterations of inference. Taking such as average is not a good way of doing inpainting since it ignores correlations between neighboring pixels. We show it here to give some indication of uncertainty in the reconstructions (inference is done by Gibbs sampling, making possible changes in the inferred latent shapes, appearances, and depths from one iteration to the next).

red, superpixels representing the background of the signs in blue, and some of the superpixels explaining window frames in green.

The nature of this decomposition is the result of training the field of masked RBMs on a large data set of natural images (see section D.3 for some examples of the training data). Many of the training images images are efficiently explained in terms of thin structures in front of larger "background" patches. Furthermore, thin horizontal and vertical structures are especially frequent in natural images, and accordingly, the models' preference for separating these into "foreground" patches is particularly robust.

To further illustrate the value of the shape model, we show the behavior of the model on a simple structure inpainting task in Figure 23. In several places, image pixels overlapping with region boundaries were removed and treated as unobserved during inference (there are seven such "unobserved" areas with an average size of more than 26 pixels; see Figure 23,

left panel). The learned shape prior allows the model to continue region boundaries across the unobserved parts of the image, giving rise to a plausible reconstruction of the removed pixels (see Figure 23, middle panel). Inference is done by sampling, and there is some uncertainty with respect to the correct reconstruction. This is reflected in the mean reconstruction (see Figure 23, right panel) for which some of the filled-in boundaries are slightly blurred. The model is, however, relatively confident in most cases. Note that the ability of the model to perform such a task crucially depends on the shape model.

6.2.2 Generating From the Field of Masked RBM. The field of masked RBMs defines a generative model of natural images and it is possible to draw samples from this model. Figure 24 shows images of size 80×80 pixels generated from a field of masked RBMs trained on natural images (the same model as the one used in the inference experiments in the previous section, see section D.3 for details). Samples are obtained by first sampling shape and appearance independently for each of the 144 patch models covering the image and then composing them according to a random depth order (as pointed out above, this generative process bears some resemblance to the "dead-leaves model", Lee et al. 2001).

The generated images contain many regions arising from partially overlapping 16×16 pixel square patches. This is to be expected considering that the training data contain large homogeneous regions that are well explained in terms of such almost completely filled superpixels (see also the discussion of the inferred latent representation in the previous section). In addition, the samples contain many regions with smooth, nonrectangular boundaries that cannot be explained in this manner. These reflect the shapes of boundaries found in natural images that have been learned by the shape model.

These characteristics of the samples (and also the nature of the latent representation inferred for real images discussed in the preceding section) suggest that the field of masked RBMs does indeed learn a sensible representation of natural images. At the same time, however, they also indicate one structural deficit of the model: individual patch models are assumed to be independent of each other. This is not necessarily a problem when performing inference since, in this case, the relevant longer-range dependencies are prescribed by the observed data (in fact, this independence assumption helps keeping inference tractable). Yet when generating from the model, this means that shapes and appearances of neighboring patches are not required to be consistent with each other, giving rise to the more or less random patchwork of shapes and appearances observed in Figure 24. This makes it very unlikely that the model will generate images with homogeneous regions larger than the size of individual patches or regions that have smooth boundaries extending across multiple patches.

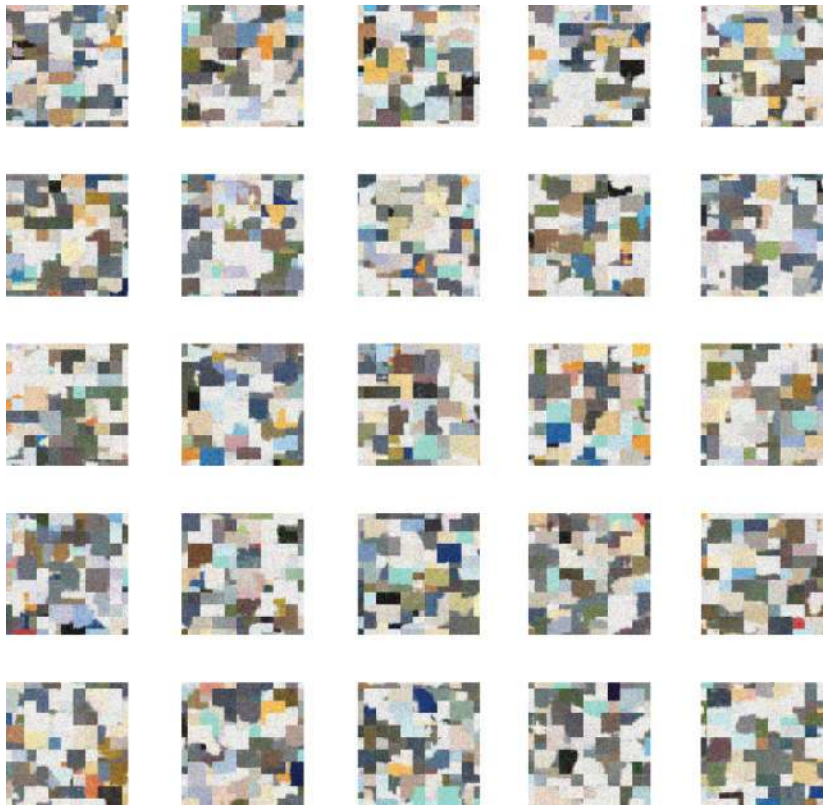

Figure 24: Samples from the field of masked RBMs trained on natural images. Samples from a field of masked RBMs trained on natural images as described in Section D.3. The sample images are of size 80×80 pixels and are obtained by sampling shape, appearance, and depth independently for each of the 144 superpixels and then composing these patches according to their relative depth. The layout of the superpixels is as described in Figure 16.

From a generative point of view, this is certainly a drawback of the model. However, the field of masked RBMs itself applied hierarchically does provide an elegant solution to this problem, which we outline in section 8.

7 Conclusion

The contributions of this article are as follows. First, we provided an empirical comparison of a range of RBMs able to model continuous data, showing that properly modeling the variance dramatically improves the quality of

the model. We then introduced the masked RBM, a generative model that works with the assumption that natural image patches are composed of objects occluding each other. In this model, each object is factored into an appearance and a shape, over which we made no prior assumptions. This proved to be a much more accurate model of image patches than the standard RBM while still allowing for efficient inference. We demonstrated how it was able to infer the depth of objects in natural scenes using only learned visual cues. We also showed that properly dealing with occlusion was essential for a good latent representation of objects. Finally, composing the masked RBMs into a field, we were able to extend our model to large images while retaining the properties observed at the patch level.

We believe the abilities to deal with occlusion, to model generic shapes and appearances, and the applicability to large images are central to a generative model suitable for a broad range of images. Inspired by previous work that dealt with a subset of these properties, we provided a unified, comprehensive probabilistic framework that, while powerful, remains computationally tractable (though still expensive). We hope that this will encourage the community to build richer, more powerful models, with the ultimate goal of approaching the capacity of the human visual system.

8 Future Work: The Deep Segmentation Network

We have shown how a field of masked RBMs is able to decompose an image into superpixels and model the shape and appearance of each superpixel using separate sets of hidden variables, even under occlusion (see Figure 22 for an example). The next stage of this research is to learn how these superpixels fit together into object parts and how object parts go together to form objects. To do this, we can follow the approach of deep belief nets and combine multiple fields of masked RBMs in a hierarchical model, which we call a deep segmentation network (DSN). The idea is to treat the superpixels learned by the first field of masked RBMs (see Figure 22) as input "pixels" for a higher-level field of masked RBMs. For example, the superpixels learned in the previous section are associated with patches laid out on a regular 8×8 grid. Hence, we can construct a new "image" one-eighth the size of the original image where the "pixels" are 512 bit feature vectors (384 shape + 128 appearance) rather than RGB values. We can train a second-level field of masked RBMs on a set of such images, where the appearance models are now binary RBMs, as shown in Figure 25. The overlapping patches of the second level cover multiple first-level superpixels and hence learn how the shape and appearance of nearby superpixels go together. Mask images will also be inferred for the second level, leading to second-level superpixels that merge a number of first-level superpixels. This process can be repeated by adding levels to the DSN until the entire image belongs to a single superpixel. This formulation gives rise to a tree-structured hierarchy in which each lower-level node (pixel) is connected to

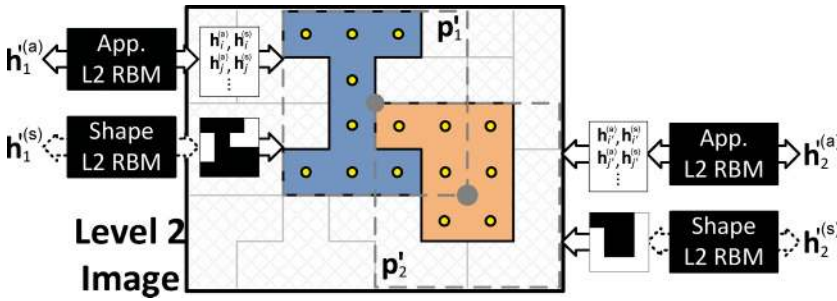

Figure 25: Second level of a DSN. The second level of a DSN is a field of masked RBMs of the same structure as the first level (Figure 16) but where the input image "pixels" are the feature vectors of the first level superpixels ($\mathbf{h}^{(a)}$, $\mathbf{h}^{(s)}$).

exactly one node in the next level. This hierarchy is, however, not fixed: since the mask determines to which superpixel pixels are associated, DSNs define an image-dependent parse tree of the input image, similar to Dynamic Trees (Williams & Adams, 1999; Storkey & Williams, 2003). However, DSNs are able to define richer and more complex priors over such parse trees than was possible with DTs. Preliminary results show that using deeper DSNs leads to meaningful higher-level superpixels while increasing accuracy on a segmentation task. We believe this is due to the capacity of the higher layers to capture longer-range dependencies, allowing parts, entire objects, and object context to be captured.

Deeper DSNs will require very large image training sets in order to learn about the range and variability of objects in natural images. Large-scale training of deep DSNs is a significant research and engineering challenge that will require extensive parallelization, in combination with novel methods for learning from vast image data sets. In the future, we will pursue this goal, with the aim of learning generative models that start to capture the daunting complexity of natural images.

Appendix A: Inference and Learning in the Masked RBM

One of the strengths of RBMs is to have a factorial posterior distribution over the latent variables given the visible ones, making it extremely easy to perform inference. Unfortunately, this is not the case in our model, since even when the mask is known, the latent images are only partially observed, resulting in a nonfactorial posterior distribution. Furthermore, the mask is not known for natural images, and this needs to be inferred as well. This section explains in detail how to infer all of these variables using Gibbs sampling. The mask model we will consider here is the occlusion-based one, as the other two can be easily deduced from it. We recall that:

- Given a mask $\mathbf{m}$, the probability of a joint state $\{\mathbf{v}, \widehat{\mathbf{v}}_{1..K}, \mathbf{h}^{(a)}_{1..K}\}$ is equal to

$$P(\mathbf{v}, \widehat{\mathbf{v}}_{1..K}, \mathbf{h}^{(a)}_{1..K} | \mathbf{m}) = \left(\prod_i \delta(\widehat{v}_{m_i,i} = v_i) \right) \left(\prod_k \text{APP}(\widehat{\mathbf{v}}_k, \mathbf{h}^{(a)}_k) \right). \tag{A.1}$$

- The probability of a joint state $\{\mathbf{m}, \mathbf{s}_{1..K}, \mathbf{h}^{(s)}_{1..K}, \pi\}$ is

$$P(\mathbf{m}, \mathbf{s}_{1..K}, \mathbf{h}^{(s)}_{1..K}, \pi) \propto P(\pi) \left(\prod_i \delta(s_{m_i,i} = 1) \prod_{k:\pi(k)<\pi(m_i)} \delta(s_{k,i} = 0) \right) \times \left(\prod_k \text{SHAPE}(\mathbf{s}_k, \mathbf{h}^{(s)}_k) \right). \tag{A.2}$$

The right-hand side of equation A.2 is unnormalized due to configurations' violating the constraints ($s_{k,i} = 0$ for all k, for instance). When generating a mask from the occlusion mask model, this could be dealt with by simply rejecting such invalid shape tuples. This would, however, mean that the shapes are no longer truly marginally independent (this corresponds to a renormalization of equation A.2). In practice we therefore take a different approach: when generating from the occlusion model, we do not draw the shape for the rear-most layer from the shape RBM but rather assume that this layer's shape is always on everywhere it is visible (for all pixels that are not covered by any of the other preceding shapes). This can be thought of as drawing the rear-most shape from a special shape model that puts all probability mass at the fully filled shape, and the generative model remains thus well defined. In this view, equation A.2 does not include the term $\text{SHAPE}(\mathbf{s}_k, \mathbf{h}^{(s)}_k)$ for $k = \pi^{-1}(K)$ (i.e., for the rear-most shape) and it is normalized, giving rise to the directed edges in Figure 9.

Combining equations A.1 and A.2, we get:

$$P\left(\mathbf{v}, \widehat{\mathbf{v}}_{1..K}, \mathbf{h}^{(a)}_{1..K}, \mathbf{m}, \mathbf{s}_{1..K}, \mathbf{h}^{(s)}_{1..K}, \pi\right) \propto \left(\prod_k \text{APP}(\widehat{\mathbf{v}}_k, \mathbf{h}^{(a)}_k) \text{SHAPE}(\mathbf{s}_k, \mathbf{h}^{(s)}_k) \right) \times P(\pi) \left(\prod_i \delta(\widehat{v}_{m_i,i} = v_i) \delta(s_{m_i,i} = 1) \prod_{k:\pi(k)<\pi(m_i)} \delta(s_{k,i} = 0) \right). \tag{A.3}$$

A.1 Inference. The joint distribution defined by equation A.3 exhibits several properties:

1. Given the latent images $\widehat{\mathbf{v}}_{1..K}$, the distribution over the appearance hidden states $\mathbf{h}^{(a)}_{1..K}$ is factorial (APP is an RBM).

2. Given the latent shapes $\mathbf{s}_{1..K}$, the distribution over the shape hidden states $\mathbf{h}^{(s)}_{1..K}$ is factorial (SHAPE is an RBM).
3. Given the image patch $\mathbf{v}$, the hidden states $\mathbf{h}^{(a)}_{1..K}$, the hidden states $\mathbf{h}^{(s)}_{1..K}$, and the ordering π, the marginal distribution over the mask $\mathbf{m}$ (when integrating out the latent images $\widehat{\mathbf{v}}_{1..K}$ and the latent shapes $\widehat{\mathbf{s}}_{1..K}$) is factorial.
4. Given the image patch $\mathbf{v}$, the mask $\mathbf{m}$, and the hidden states $\mathbf{h}^{(a)}_{1..K}$, the distribution over the latent images $\widehat{\mathbf{v}}_{1..K}$ is factorial.
5. Given the mask $\mathbf{m}$, the hidden states $\mathbf{h}^{(s)}_{1..K}$, and the ordering π, the distribution over the latent shapes $\mathbf{s}_{1..K}$ is factorial.

Properties 1, 2, 4, and 5 are easily deduced from the form of equation A.3. Let us prove property 3. Given the image patch $\mathbf{v}$, the hidden states $\mathbf{h}^{(a)}_{1..K}$, the hidden states $\mathbf{h}^{(s)}_{1..K}$, and the ordering π, we have

$$P\left(\widehat{\mathbf{v}}_{1..K}, \mathbf{m}, \mathbf{s}_{1..K} | \mathbf{v}, \mathbf{h}^{(a)}_{1..K}, \mathbf{h}^{(s)}_{1..K}, \pi\right) \propto \left(\prod_k \text{APP}(\widehat{\mathbf{v}}_k | \mathbf{h}^{(a)}_k) \text{SHAPE}(\mathbf{s}_k | \mathbf{h}^{(s)}_k)\right)$$
$$\times \left(\prod_i \delta(\widehat{v}_{m_i,i} = v_i)\delta(s_{m_i,i} = 1) \prod_{k:\pi(k)<\pi(m_i)} \delta(s_{k,i} = 0)\right), \tag{A.4}$$

where $\text{APP}(\widehat{\mathbf{v}}_k | \mathbf{h}^{(a)}_k)$ (resp. $\text{SHAPE}(\mathbf{s}_k | \mathbf{h}^{(s)}_k)$) is the conditional probability of $\widehat{\mathbf{v}}_k$ (resp. $\mathbf{s}_k$) given $\mathbf{h}^{(a)}_k$ (resp. $\mathbf{h}^{(s)}_k$) under the appearance RBM (resp. shape RBM).

Thus, for the mask m_i to be equal to k, we need that:

- $\widehat{v}_{k,i} = v_i$.
- $s_{t,i} = 0$ if $\pi(t) < \pi(k)$.
- $s_{k,i} = 1$.

Since, in equation A.4, the distributions over $\widehat{v}_{k,i}$ and $s_{k,i}$ are factorial and do not depend on the value of m_k, the resulting conditional distribution on m_i is also factorial.

This suggests the following Gibbs sampling scheme to infer all the hidden variables given an image $\mathbf{v}$. Starting from a random mask $\mathbf{m}$, we iterate over the following steps:

1. Given the mask $\mathbf{m}$, we sample the unobserved parts of the latent images $\widehat{\mathbf{v}}_{1..K}$ using block Gibbs sampling (using properties 1 and 4).
2. Given the mask $\mathbf{m}$ and the ordering π, we sample the unobserved parts of the latent shapes $\mathbf{s}_{1..K}$ using block Gibbs sampling (using properties 2 and 5).
3. Given the latent images $\widehat{\mathbf{v}}_{1..K}$, we sample the appearance hidden units $\mathbf{h}^{(a)}_{1..K}$ (using property 1).

4. Given the latent shapes $\mathbf{s}_{1..K}$, we sample the shape hidden units $\mathbf{h}^{(s)}_{1..K}$ (using property 2).
5. Given the appearance hidden units $\mathbf{h}^{(a)}_{1..K}$, the shape hidden units $\mathbf{h}^{(s)}_{1..K}$, the image patch $\mathbf{v}$ and the ordering π, we sample a new mask $\mathbf{m}$ (using property 3).
6. Given the mask, infer the depth ordering as explained in section B.1 in appendix B.

This process is repeated until convergence of the mask. The sampling procedure directly implies that the mask may be different each time. However, in all our experiments, it consistently matched the structure of the shapes in the images.

A.2 Learning. We shall now see how learning of the parameters $W^{(s)}$ and θ can be achieved using the above inference procedure. We need to compute the gradient of the log probability of an image patch $\mathbf{v}$ with respect to the parameters, that is,

$$\frac{\partial \log p(\mathbf{v})}{\partial \theta} = \frac{\partial}{\partial \theta} \log \sum_{\mathbf{m}, \widehat{\mathbf{v}}_{1..K}, \mathbf{h}^{(a)}_{1..K}, \mathbf{s}_{1..K}, \mathbf{h}^{(s)}_{1..K}, \pi} P\left(\mathbf{v}, \mathbf{m}, \widehat{\mathbf{v}}_{1..K}, \mathbf{h}^{(a)}_{1..K}, \mathbf{s}_{1..K}, \mathbf{h}^{(s)}_{1..K}, \pi\right), \tag{A.5}$$

Since this cannot be computed exactly, we shall use an EM procedure (Dempster, Laird, & Rubin, 1977). We first derive a variational lower bound of $\log p(\mathbf{v})$:

$$\begin{aligned} \log p(\mathbf{v}) \geq \sum_{\mathbf{m}, \widehat{\mathbf{v}}_{1..K}, \mathbf{s}_{1..K}, \pi} & Q(\mathbf{m}, \widehat{\mathbf{v}}_{1..K}, \mathbf{s}_{1..K}, \pi | \mathbf{v}) \\ & \log \sum_{\mathbf{h}^{(a)}_{1..K}, \mathbf{h}^{(s)}_{1..K}} P\left(\mathbf{v}, \mathbf{m}, \widehat{\mathbf{v}}_{1..K}, \mathbf{h}^{(a)}_{1..K} \mathbf{s}_{1..K}, \mathbf{h}^{(s)}_{1..K}, \pi\right) \\ & - H[Q(\mathbf{m}, \widehat{\mathbf{v}}_{1..K}, \mathbf{s}_{1..K}, \pi | \mathbf{v})] \end{aligned}$$

for any function Q. The bound is tight when $Q(\mathbf{m}, \widehat{\mathbf{v}}_{1..K}, \mathbf{s}_{1..K}, \pi | \mathbf{v})$ is the true posterior distribution. Since we cannot compute the sum over all masks, all latent images, all latent shapes, and all orderings, we will replace it by a sample from the posterior distribution. Therefore, the gradient direction we follow is

$$\Delta\theta \propto \frac{\partial}{\partial \theta} \log \sum_{\mathbf{h}^{(a)}_{1..K}, \mathbf{h}^{(s)}_{1..K}} P\left(\mathbf{v}, \widetilde{\mathbf{m}}, \widetilde{\widehat{\mathbf{v}}}_{1..K}, \mathbf{h}^{(a)}_{1..K}, \widetilde{\mathbf{s}}_{1..K}, \mathbf{h}^{(s)}_{1..K}, \widetilde{\pi}\right), \tag{A.6}$$

where $\widetilde{\mathbf{m}}$, $\widetilde{\mathbf{v}}_{1..K}$, $\widetilde{\mathbf{s}}_{1..K}$, and $\widetilde{\pi}$ are samples from the posterior distribution (obtained using the method described in section A.1). Using more than one sample would reduce noise at the expense of extra computation. In our experiments, we used a single sample and found that learning worked well.

Appendix B: Depth Inference in the Occlusion Model

B.1 Depth Inference for Image Patches. In order to infer the depth variable π given a mask $\mathbf{m}$, we consider each possible ordering of the K layers explicitly. The mask $\mathbf{m}$, together with a particular occlusion order π, defines which shape pixels $s_{k,i}$ are observed and which are unobserved. This is illustrated in Figure 8. The likelihood of a particular ordering π is then simply given as the likelihood of all the partially observed shapes $\mathbf{s}_k$ under the shape model:

$$P(\pi|\mathbf{m}) \propto \prod_{k=1}^{K} \sum_{\{s_{k,i}: i \in U_{\pi,k}(\mathbf{m})\}} \sum_{\mathbf{h}_k^{(s)}} \text{SHAPE}(\mathbf{s}_k, \mathbf{h}_k^{(s)}). \tag{B.1}$$

Here, $U_{\pi,k}(\mathbf{m})$ is the set of all unobserved pixels for shape k given the mask $\mathbf{m}$ and the ordering π. The set of unobserved pixels $U_{\pi,k}(\mathbf{m})$ will vary among different orderings π, and this is what drives the depth inference.

In practice, the sum over unobserved pixels and over the latent variables $\mathbf{h}_k^{(s)}$ cannot be computed exactly. We therefore replace the first sum by sampling the unobserved pixels $\{s_{k,i} : i \in U_{\pi,k}(\mathbf{m})\}$ conditioned on the observed shape pixels for each k and π. Sampling can be done efficiently using several iterations of block Gibbs sampling. This results in "completed" shape images $\hat{\mathbf{s}}_k^{\pi}$ for which the unnormalized probability under the shape model can be computed efficiently,[5]

$$p(\hat{\mathbf{s}}_k^{\pi}) = \sum_{\mathbf{h}} \text{SHAPE}(\hat{\mathbf{s}}_k^{\pi}, \mathbf{h}) \tag{B.2}$$

$$\propto \exp\left(\mathbf{b}^T \hat{\mathbf{s}}_k^{\pi}\right) \prod_j \left[1 + \exp\left((\hat{\mathbf{s}}_k^{\pi})^T W_{\cdot j}\right)\right]. \tag{B.3}$$

[5] It should be noted that equation B.2 is not an unbiased estimate of the unnormalized log probability. Overall this estimator might give rise to a slight preference for depth orderings with fewer unobserved shape pixels. Nevertheless, in our experiments we found the estimator to work well. An unbiased estimator can also be constructed: Let $\mathbf{s}_O$ denote the observed shape pixels and $\mathbf{s}_U$ the unobserved ones (for a given mask, depth ordering, and layer). In this notation equation B.2 corresponds to $\hat{Z} = \tilde{p}(\hat{\mathbf{s}}_U, \mathbf{s}_O)$ where $\hat{\mathbf{s}}_U \sim p(\mathbf{s}_U|\mathbf{s}_O)$ through multiple iterations of Gibbs sampling ($\tilde{p}(\mathbf{s})$ is the unnormalized log probability after summing out $\mathbf{h}$). We obtain an unbiased estimator by considering $\hat{Z}' = \tilde{p}(\hat{\mathbf{s}}_U, \mathbf{s}_O)/p(\hat{\mathbf{s}}_U|\hat{\mathbf{h}})$ where $\hat{\mathbf{s}}_U \sim p(\mathbf{s}_U|\hat{\mathbf{h}})$ and $\hat{\mathbf{h}} \sim p(\mathbf{h}|\mathbf{s}_O)$.

We then obtain

$$P(\pi|m, \hat{\mathbf{s}}^{\pi}_{1..K}) \propto \prod_{k=1}^{K} \text{SHAPE}(\hat{\mathbf{s}}^{\pi}_{k}). \tag{B.4}$$

Note that the completed shape images are different for different π; for plausible orderings, the shape model will be able to "fill in" the unobserved pixels to give rise to a shape with a high likelihood, which leads to a high probability of the respective ordering. It should further be noted that although considering each possible ordering π explicitly might seem expensive (the number of possible orderings is factorial in K), this remains feasible in practice for $K \leq 4$. Given a depth ordering π and the latent states of the K shape RBMs $\{\mathbf{h}^{(s)}_{k}\}_{k=1...K}$, the conditional probability of the mask is given as

$$\begin{aligned} P(m_i = t|\mathbf{h}^{(s)}_{1..K}, \pi) &\propto \text{SHAPE}\big(s_{t,i} = 1|\mathbf{h}^{(s)}_{t}\big) \\ &\times \prod_{k:\pi(k)<\pi(t)} \big[1 - \text{SHAPE}\big(s_{k,i} = 1|\mathbf{h}^{(s)}_{k}\big)\big]. \end{aligned} \tag{B.5}$$

This probability can be combined with the signal from the appearance models as described in appendix A. The shape in the rear-most layer is largely determined by the preceding layers. For this reason, and as explained in appendix A, we treat the rear-most shape in a special manner. During depth inference, this means that we ignore the likelihood of the rear-most shape when computing the probability of a particular depth ordering π using equation B.4: $P(\pi|m, \hat{\mathbf{s}}^{\pi}_{1..K}) \propto \prod_{k:\pi(k)\neq K} \text{SHAPE}(\hat{\mathbf{s}}^{\pi}_{k})$. Note that the product here no longer includes a term for the rear-most layer. Similarly, equation B.5 becomes $P(m_i = t|\mathbf{h}^{(s)}_{1..K}, \pi) = \prod_{k:\pi(k)<\pi(t)}[1 - \text{SHAPE}(s_{k,i} = 1|\mathbf{h}^{(s)}_{k})]$ if t is the rear-most layer (i.e., if $t = \pi^{-1}(K)$) and the proportionality $\propto$ in equation B.5 becomes an equality for all other values of t.

B.2 Depth Inference for Images. Depth inference at the image level, given a mask image, is performed by determining local depth orderings of overlapping patches. For this purpose, each patch is considered in turn and its depth relative to its neighbors is determined, keeping the ordering of its neighbors fixed. For instance, for the experiments with 16×16 pixel patches and $K = 4$, each patch model overlaps partially with eight neighboring patches (so that each pixel is covered by four competing patch models). Thus, for any given patch and a fixed ordering of its eight neighbors, nine different relative depths need to be considered. Each of these relative depths gives rise to a set of unobserved pixels, not only for the patch considered but also for its neighbors. The probability of the different relative depths can be computed in essentially the same way as described in section B.1

(approximating the sum over unobserved pixels by a sample and then efficiently computing the unnormalized log probability of the completed shape).

Note that for each neighboring patch, the set of unobserved pixels depends only on whether the patch under consideration is in front of or behind that neighbor; this considerably reduces the number of "shape completions" that need to be considered (two completions per neighboring patch and $N+1$ for the central patch, where N is the number of neighbors).

In practice, given a mask, we perform one full sweep through the set of patch models, updating the relative depth (and the latent shapes) of each patch with respect to its neighbors once in a random order. Given the resulting depth ordering and the latent states of the shape models, the mask can then be updated as in the patch case (cf. equation B.5 above).

Appendix C: Computing the Log Probability of Image Patches Under the Masked RBM

Due to the number of latent variables involved in the masked RBM, it is impossible to compute the exact log probability of natural image patches under this model. We may, however, derive a variational lower bound that would allow us to quantify the gains provided by the mask. However, one must bear in mind that all the techniques presented in this section require the use of AIS to yield an estimate, which is unusable in our setting (see section 5.3.1). Nevertheless, we believe them to be of interest if the limitations of AIS may be overcome.

C.1 Uniform Mask Model. We begin with the uniform mask model case as this will simplify the equations. From there, it is relatively straightforward to move to the more complex mask models:

$$\begin{aligned}
\log p(\mathbf{v}) &= \log \sum_{\mathbf{m},\widehat{\mathbf{v}}_{1..K},\mathbf{h}^{(a)}_{1..K}} P\left(\mathbf{v}, \mathbf{m}, \widehat{\mathbf{v}}_{1..K}, \mathbf{h}^{(a)}_{1..K}\right) \\
&= \log \sum_{\mathbf{m},\widehat{\mathbf{v}}_{1..K},\mathbf{h}^{(a)}_{1..K}} P\left(\mathbf{v}, \mathbf{m}, \widehat{\mathbf{v}}_{1..K}, \mathbf{h}^{(a)}_{1..K}\right) \frac{Q(\mathbf{m}|\mathbf{v})}{Q(\mathbf{m}|\mathbf{v})} \\
&\geq \sum_{\mathbf{m}} Q(\mathbf{m}|\mathbf{v}) \log P(\mathbf{m}) \sum_{\widehat{\mathbf{v}}_{1..K},\mathbf{h}^{(a)}_{1..K}} P\left(\mathbf{v}, \widehat{\mathbf{v}}_{1..K}, \mathbf{h}^{(a)}_{1..K}|\mathbf{m}\right) \\
&\quad - \sum_{\mathbf{m}} Q(\mathbf{m}|\mathbf{v}) \log Q(\mathbf{m}|\mathbf{v}) \qquad \text{(C.1)}
\end{aligned}$$

for any function Q, using Jensen's inequality. Let us first rewrite the sum inside the logarithm:

$$P\left(\mathbf{v}, \widehat{\mathbf{v}}_{1..K}, \mathbf{h}^{(a)}_{1..K}|\mathbf{m}\right) = P\left(\widehat{\mathbf{v}}_{1..K}, \mathbf{h}^{(a)}_{1..K}|\mathbf{m}\right) P\left(\mathbf{v}|\widehat{\mathbf{v}}_{1..K}, \mathbf{h}^{(a)}_{1..K}, \mathbf{m}\right).$$

The second term enforces the constraints described in equation A.1: all configurations that do not match $\widehat{\mathbf{v}}_{m_i,i} = v_i$ for all i have zero probability. Therefore, we need only to compute the sum over the configurations satisfying these constraints. Since these constraints are independent of the $\mathbf{h}^{(a)}_k$, we have

$$P\left(\mathbf{v}|\widehat{\mathbf{v}}_{1..K}, \mathbf{h}^{(a)}_{1..K}, \mathbf{m}\right) = P\left(\mathbf{v}|\widehat{\mathbf{v}}_{1..K}, \mathbf{m}\right),$$

and this distribution is fully concentrated on one point (given the latent images and the mask, there is only one valid image). Furthermore, we have

$$P\left(\widehat{\mathbf{v}}_{1..K}, \mathbf{h}^{(a)}_{1..K}|\mathbf{m}\right) = \prod_{k=1}^{K} P\left(\widehat{\mathbf{v}}_k, \mathbf{h}^{(a)}_k|\mathbf{m}\right),$$

yielding

$$\sum_{\widehat{\mathbf{v}}_{1..K}, \mathbf{h}^{(a)}_{1..K}} P(\mathbf{m}) P\left(\mathbf{v}, \widehat{\mathbf{v}}_{1..K}, \mathbf{h}^{(a)}_{1..K}|\mathbf{m}\right) = P(\mathbf{m}) \prod_{k=1}^{K} \sum_{\widehat{\mathbf{v}}_k \in C_k, \mathbf{h}^{(a)}_k} P\left(\widehat{\mathbf{v}}_k, \mathbf{h}^{(a)}_k|\mathbf{m}\right)$$

$$\sum_{\widehat{\mathbf{v}}_{1..K}, \mathbf{h}^{(a)}_{1..K}} P(\mathbf{m}) P\left(\mathbf{v}, \widehat{\mathbf{v}}_{1..K}, \mathbf{h}^{(a)}_{1..K}|\mathbf{m}\right) = P(\mathbf{m}) \prod_{k=1}^{K} \sum_{\widehat{\mathbf{v}}_k \in C_k} P\left(\widehat{\mathbf{v}}_k|\mathbf{m}\right), \tag{C.2}$$

where C_k is the set of $\widehat{\mathbf{v}}_k$ matching the constraints imposed by $\mathbf{v}$ and $\mathbf{m}$ (as defined in equation A.1). We recall that the set C_k is the set of all $\widehat{\mathbf{v}}_k$ such that $\widehat{\mathbf{v}}_{k,i} = v_i$ if $m_i = k$. Therefore, we need to sum the probabilities of all visible vectors with a subset of the units being fixed. This can be done using AIS (Salakhutdinov & Murray, 2008). Indeed, the conditional distribution over a subset of the visible units given the rest of the other visible units is also an RBM (conditioning on some visible units only modifies the biases of the hidden layer). Given the strong constraint imposed by the observed pixels, the resulting RBM is likely to have a very peaked distribution, making its partition function easy to approximate.

Now that we know how to compute $\sum_{\widehat{\mathbf{v}}_{1..K}, \mathbf{h}^{(a)}_{1..K}} P(\mathbf{v}, \mathbf{m}, \widehat{\mathbf{v}}_{1..K}, \mathbf{h}^{(a)}_{1..K})$ for a given $\mathbf{m}$, we need to find the optimal subset of masks to consider (that is, the distribution $Q(\mathbf{m}|\mathbf{v})$).

Let us denote $p_i = P(\mathbf{v}, \mathbf{m}^i)$ for a certain mask configuration $\mathbf{m}^i$ and $q_i = Q(\mathbf{m}^i|\mathbf{v})$. We need to optimize the quantity $D = \sum_i q_i \log p_i - \sum_i q_i \log q_i$

over the q_i's, subject to the constraint $\sum_i q_i = 1$. The optimal solution is given by $q_i = \dfrac{p_i}{\sum_i p_i}$, yielding

$$D = \log \sum_i p_i. \tag{C.3}$$

We therefore need to find the $\mathbf{m}^i$'s yielding the maximal p_i's. Since $p_i = P(\mathbf{v}, \mathbf{m}^i) = P(\mathbf{v})P(\mathbf{m}^i|\mathbf{v})$, we need to find the modes of the posterior distribution of $\mathbf{m}$ given $\mathbf{v}$. Due to the very constrained nature of the mask, the probability mass is heavily concentrated around a small number of modes, making it possible to achieve a tight bound over the log probability of an image patch with few masks.

A simpler explanation of this approximation is that we have replaced the quantity $p(\mathbf{v}) = \sum_{\mathbf{m}} p(\mathbf{v}, \mathbf{m})$ by a sum over a subset of the masks. It then becomes clear that this subset needs to include the masks $\mathbf{m}$ for which the quantity $p(\mathbf{v}, \mathbf{m})$ is maximized.

To find the modes of $P(\mathbf{m}|\mathbf{v})$, we first do a few iterations (typically 20) of sampling as described in section A.1 and then replace the third sampling step by a maximization step for a few more iterations (typically 10). Maximization should not be performed from the beginning as this often results in finding a poor local optimum.

C.2 Nonuniform Mask Model. In the case of a nonuniform (occlusion-based) mask model, we have

$$\begin{aligned}
\log p(\mathbf{v}) &= \log \sum_{\mathbf{m}, \widehat{\mathbf{v}}_{1..K}, \mathbf{h}^{(a)}_{1..K}, \mathbf{s}_{1..K}, \mathbf{h}^{(s)}_{1..K}, \pi} P\left(\mathbf{v}, \mathbf{m}, \widehat{\mathbf{v}}_{1..K}, \mathbf{h}^{(a)}_{1..K}, \mathbf{s}_{1..K}, \mathbf{h}^{(s)}_{1..K}, \pi\right) \\
&= \log \sum_{\mathbf{m}, \widehat{\mathbf{v}}_{1..K}, \mathbf{h}^{(a)}_{1..K}, \mathbf{s}_{1..K}, \mathbf{h}^{(s)}_{1..K}, \pi} P\big(\mathbf{v}, \mathbf{m}, \widehat{\mathbf{v}}_{1..K}, \mathbf{h}^{(a)}_{1..K}, \mathbf{s}_{1..K}, \mathbf{h}^{(s)}_{1..K}, \pi\big) \frac{Q(\mathbf{m}|\mathbf{v})}{Q(\mathbf{m}|\mathbf{v})} \\
&\geq \sum_{\mathbf{m}} Q(\mathbf{m}|\mathbf{v}) \log P(\mathbf{m}) \sum_{\widehat{\mathbf{v}}_{1..K}, \mathbf{h}^{(a)}_{1..K}, \mathbf{s}_{1..K}, \mathbf{h}^{(s)}_{1..K}, \pi} \\
&\qquad \times P\left(\mathbf{v}, \widehat{\mathbf{v}}_{1..K}, \mathbf{h}^{(a)}_{1..K}, \mathbf{s}_{1..K}, \mathbf{h}^{(s)}_{1..K}, \pi|\mathbf{m}\right) \\
&\qquad - \sum_{\mathbf{m}} Q(\mathbf{m}|\mathbf{v}) \log Q(\mathbf{m}|\mathbf{v}).
\end{aligned} \tag{C.4}$$

Given $\mathbf{m}$, the latent variables may be split in two sets as follows:

$$\begin{aligned}
&P\big(\mathbf{v}, \widehat{\mathbf{v}}_{1..K}, \mathbf{h}^{(a)}_{1..K}, \mathbf{s}_{1..K}, \mathbf{h}^{(s)}_{1..K}, \pi|\mathbf{m}\big) \\
&\quad = P\big(\mathbf{v}, \widehat{\mathbf{v}}_{1..K}, \mathbf{h}^{(a)}_{1..K}|\mathbf{m}\big) P\big(\mathbf{s}_{1..K}, \mathbf{h}^{(s)}_{1..K}, \pi|\mathbf{m}\big),
\end{aligned} \tag{C.5}$$

and, following the same reasoning as in section C.1, we could compute $P(\mathbf{s}_{1..K}, \mathbf{h}_{1..K}^{(s)}, \pi|\mathbf{m})$ using AIS.

Unfortunately, AIS does not work well in these models due to the low variance of the conditional distributions, resulting in confident but wrong estimates of the partition function.

Appendix D: Experimental Procedure

In this section, we describe in greater detail how the model was trained.

D.1 Pretraining the Appearance RBM. In our model, patches are of size 16×16, which means that the appearance RBM has 768 ($16 \times 16 \times 3$) visible units. We used a beta RBM of the form described in section 2.6, with 128 hidden units. In the first phase, the RBM was trained without using a mask and using stochastic approximation (Tieleman, 2008). We performed a few tens of thousands of parameter updates. Once the filters started converging, we continued training in the masked context (with a uniform mask model) with $K = 2$. The inferred masks were kept between epochs, and only one iteration of mask update was run for each patch. To compute this mask update, the unobserved pixels of the latent patches were initialized to the mean of the observed ones (for each color channel), and one up-down pass was realized to update these unobserved values. Once this was done, we sampled $\mathbf{h}$ given $\mathbf{v}$ to update the mask. Eventually we completed this training with $K = 4$.

This part of training does not critically depend on the number of parameter updates for each phase. If it is too low, the whole procedure will be slower, as inferring the mask is more expensive for higher values of K. If it is too high, the next phase will unlearn what has been previously learned, again slowing down the pretraining, but for a similar final result.

Also, using only one iteration to reinfer the mask proved to be enough to yield accurate results.

D.2 Training the Occlusion-Based Shape Model for Image Patches. The shape model for image patches was trained in two phases. In the first phase we pretrained the shape model directly on binary mask patches. For this purpose we inferred the mask ($K = 3$) for a large set of natural image patches (16×16 pixels RGB patches) using the uniform model as mask prior. For each patch, we performed 100 mask iterations. From each patch, we thus obtained three binary mask patches (due to the lack of a shape prior, many of these mask patches were very noisy). We then trained a binary RBM (384 hidden units, 256 visible units) directly on 95000 binary mask patches. Training was performed with stochastic approximation (Tieleman, 2008) with a small learning rate of 0.0005, weight decay 0.0002, no momentum, and mini-batches of size 100. Training was performed for 10,000 epochs.

The parameters of this binary RBM served as initialization for training of the shape model in the context of the full model. Pretraining took about 3.5 days using our Matlab implementation on a single-core machine.

In the second phase, we trained the shape model in the context of the full model (masked RBM with $K = 3$). The parameters of the shape RBM were initialized with the parameters obtained from phase 1. We used a training set of 21,000 RGB patches grouped into mini-batches of size 60. Learning was performed in alternation with inference. For each patch, we performed two iterations of full inference in the model (this includes the update of the appearance fantasies, the depth, the shape fantasies, and the mask) before updating the model parameters. Inference was performed as described in the main text. During inference in the mask model, we used 10 iterations of masked Gibbs sampling to update the shape fantasies. Before sampling, unobserved pixels in the shape fantasies were initialized with their state from the previous cycle. To prevent the model from hallucinating shapes into unused layers (which would slow down learning), we forced such layers to be in front of all visible layers and thus to be empty. Learning was performed using CD-10 with a learning rate of 0.001, a weight decay of 0.0002, and a momentum of 0.5. Training in the full model was performed for 550 epochs and took approximately two weeks using our unoptimized Matlab implementation on a single-core machine.

D.3 Training the Occlusion-Based Shape Model for Natural Images. As for image patches, the shape model for natural images (i.e., for the field of masked RBMs) was trained in two phases.

For pretraining we inferred the mask for natural images of size 80×80 pixels (RGB) extracted from the MSRC data set (see Figure 26) with a field of masked RBMs, using the uniform model as mask prior, and running 100 iterations of mask inference. From each image, we obtained 144 binary mask patches (using the superpixel layout described in the main text, each 80×80 pixel image is covered by four layers of 6×6 superpixels of size 16×16 pixels). We randomly selected 95,000 binary mask patches (excluding any mask patches from superpixels not fully overlapping with the images) and used those as training data for a binary RBM (256 visible units, 384 hidden units). Training was performed for 10,000 epochs using stochastic approximation, with a learning rate of 0.0005, no momentum, a weight decay of 0.0002, and mini-batches of size 100. The parameters of this binary RBM were used to initialize the shape model for training in the context of the full model.

We subsequently trained the occlusion-based shape model in the context of a field of masked RBMs, initializing the binary RBM for the shape model with the parameters obtained in phase 1. Our training set consisted of 1000 RGB images, and our "batches" consisted of individual images (144 superpixels are associated with each image). We alternated inference and the update of the model parameters. Two iterations of full inference

Figure 26: Examples of 80 × 80 pixels training images for field of masked RBMs.

(update of the appearance fantasies, shape fantasies, relative depth for all superpixels, as well as of the mask) were performed for each image before computing the gradient and updating the parameters. Inference in the mask model was performed in parallel for patches that did not share neighbors (i.e., for patches that were independent conditioned on the mask and the remaining nonoverlapping patches), and such sets of independent patches were treated sequentially but in a random order. Ten steps of masked Gibbs sampling were performed to update the shape fantasies. Completely unobserved superpixels were forced to be in front (i.e., their shape fantasies were required to be completely off) in order to prevent unconstrained hallucinations by the model. We used CD-15 for training, with a learning rate of 0.0025, weight decay of 0.0002, and momentum of 0.5. For the superpixel layout described in the main text, some superpixels are overlapping with the image boundaries: they are always only partially unobserved. To prevent the model from learning from largely unconstrained shapes (its own hallucinations), we did not include shapes from superpixels into the gradient that overlapped with the image to less than 25%. Training was run for 100 iterations and took approximately three weeks using our unoptimized Matlab implementation on a single-core machine.

Appendix E: Additional Analysis of the Model for Natural Image Patches

Inferring relative depth based on the information provided to the model in the experiment shown in Figure 15—using only very local shape information (from small, 16 × 16 image patches)—is a highly ambiguous problem in many cases, not just for our model but equally so for a human observer. Accordingly, the confidence of the model with respect to the relative depth of the regions in a patch can vary significantly between patches. For the

examples shown in Figure 16, the model is rather confident with respect to the inferred depth for patches 1, 2, 4, and 5 but considerably less confident for patch 3 (inference is performed by sampling from the posterior distribution; Figure 15 shows the most likely depth ordering under the model for the five patches).

To evaluate the behavior of the model on a larger data set and demonstrate how learning of a shape prior can drive depth inference, we ran depth inference on 73 three-region mask patches, similar to patch 3 in Figure 15, extracted from the segmentation images provided with the Berkeley segmentation database.[6] Depth inference was run for 8000 iterations, and the inferred depth after each iteration was recorded. For each patch, we determined which of the three mask regions was most frequently sampled to be the front-most region and which of the remaining two layers was most frequently chosen to be the middle layer. For the preferred middle-layer region, we then determined, for each patch, the average shape fantasy associated with that region being the middle layer. The results are shown in Figures 27a and 27b.

Although there is some variability, the model has a clear tendency to explain the mask patches in terms of extended shapes overlapping each other, in particular in terms of roughly horizontal or vertical shapes. This is consistent with the results shown in Figure 15 and a very plausible behavior given the training data in which regions of such shapes occur frequently (note that these shapes also feature prominently in the samples shown in Figure 12). This behavior is also in rough agreement with the judgment of human observers. We showed the same 73 patches to five subjects and asked them to indicate, for each patch, which of the three regions they thought to be in front. The depth inferred by the model was consistent with the majority of human observers in 44 of 73 cases—60% of the patches, a considerably higher percentage than expected if the model selected the front-most region randomly (a random choice of the front most region in this task would correspond to an agreement of 33%). At the same time, human subjects were in agreement with each other only for 32/73 patches (44%), highlighting the general difficulty and ambiguity of this task. Note that these results cannot be explained by a simple bias of the model to place smaller regions in the front since for 37 of 73 (51%) of the test patches, the region that was inferred to be in front by the model was in fact the largest of the three regions, while the smallest region was inferred to be in front only in 17/73 (23%) cases.

[6]We used mask patches (i.e., patches for which the segmentation had already been provided) in order to separate depth inference from the segmentation problem. As explained in the main text the segmentation of a patch can be affected, for example, by matting or shading, which the appearance model does not currently handle well.

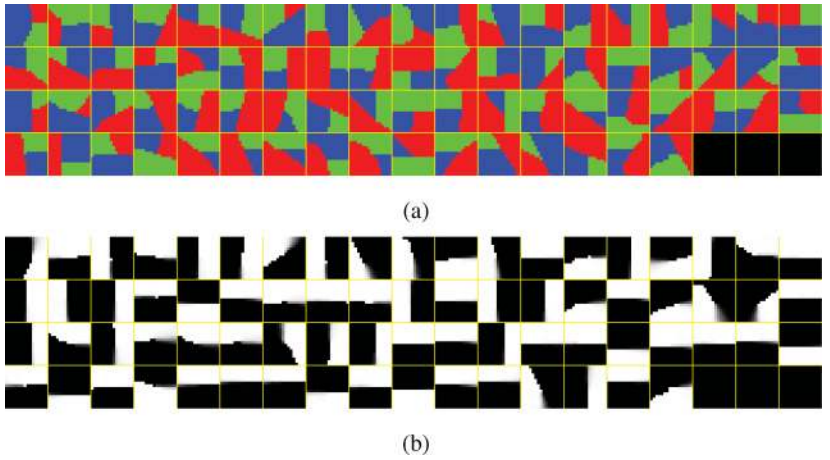

Figure 27: Depth inference for mask patches (a) Mask patches with three regions extracted from segmentation images from the Berkeley Segmentation Database. Each region is colored according to the depth inferred by the model as in Figure 15: red, front; green, middle; blue, back. (b) Average shape fantasy for the middle layer for each of the mask patches and the associated preferred ordering shown in *a*. Although there is some variability, the model tends to explain the mask patches in terms of extended shapes overlapping each other, in many cases consistent with human judgement.

Acknowledgments

We thank Chris Williams for his support and help and Iain Murray for insightful comments. N.H. is supported by an Engineering and Physical Sciences Research Council/Medical Research Council scholarship from the Neuroinformatics and Computational Neuroscience Doctoral Training Centre at the University of Edinburgh.

References

Ackley, D. H., Hinton, G. E., & Sejnowski, T. J. (1985). A learning algorithm for Boltzmann machines. *Cognitive Science*, *9*, 147–169.

Bienenstock, E., Geman, S., & Potter, D. (1997). Compositionality, MDL priors, and object recognition. In M. C. Mozer, M. I. Jordan, & T. Petsche (Eds.), *Neural information processing systems* (pp. 838–844). Cambridge, MA: MIT Press.

Bouchard, G., & Triggs, B. (2005). Hierarchical part-based visual object categorization. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* (Vol. 1, pp. 710–715). Washington, DC: IEEE Computer Society.

Bouman, C., & Shapiro, M. (1994). A multiscale random field model for Bayesian image segmentation. *3*, 162–177.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological), 39*, 1–38.

Fidler, S., & Leonardis, A. (2007). Towards scalable representations of object categories: Learning a hierarchy of parts. In *2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Los Alamitos, CA: IEEE Computer Society.

Freund, Y., & Haussler, D. (1994). *Unsupervised learning of distributions on binary vectors using two layer networks* (Tech. Rep. UCSC-CRL-94-25). Santa Cruz: University of California.

Frey, B. J., & Jojic, N. (2003). Learning appearance and transparency manifolds of occluding objects in layers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Los Alamitos, CA: IEEE Computer Society Press.

Guo, C.-E., Zhu, S.-C., & Wu, Y. N. (2003). Modeling visual patterns by integrating descriptive and generative methods. *Int. J. Comput. Vision, 53*, 5–29.

Guo, C.-E., Zhu, S.-C., & Wu, Y. N. (2007). Primal sketch: Integrating structure and texture. *Comput. Vis. Image Underst., 106*, 5–19. doi:http://dx.doi.org/10.1016/j.cviu.2005.09.004.

Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural Computation, 14*, 1771–1800.

Hinton, G. E., Ghahramani, Z., & Teh, Y. W. (2000). Learning to parse images. *Advances in neural information processing systems, 12*. In S. A. Solla, T. K. Leen, & K.-R. Müller (Eds.), (pp. 463–469). Cambridge, MA: MIT Press.

Hinton, G. E., Osindero, S., & Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural Comp., 18*, 1527–1554. Available online at http://neco.mitpress.org/cgi/content/abstract/18/7/1527.

Hyvärinen, A., Hoyer, P. O., & Inki, M. O. (2001). Topographic independent component analysis. *Neural Comput., 13*, 1527–1558.

Jin, Y., & Geman, S. (2006). Context and hierarchy in a probabilistic image model. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 2145–2152). Washington, DC: IEEE Computer Society. doi:http://dx.doi.org/10.1109/CVPR.2006.86.

Kannan, A., Jojic, N., & Frey, B. J. (2005). Generative model for layers of appearance and deformation. In *Proceedings of the Tenth Annual Worishop on Artificial Intelligence and Statistics*. N.p.: Society for Artificial Intelligence and Statistics.

Kannan, A., Winn, J. M., & Rother, C. (2006). Clustering appearance and shape by learning jigsaws. In B. Schölkopf, J. Platt, & T. Hoffman (Eds.), *Advances in neural information proceesing systems, 19*. Cambridge, MA: MIT Press.

Karklin, Y., & Lewicki, M. (2009). Emergence of complex cell properties by learning to generalize in natural scenes. *Nature, 457*, 83–86.

Larochelle, H., Erhan, D., Courville, A., Bergstra, J., & Bengio, Y. (2007). An empirical evaluation of deep architectures on problems with many factors of variation. In Z. Ghahramani (Ed.), *Twenty-Fourth International Conference on Machine Learning (ICML'2007)* (pp. 473–480). Available online at http://www.machinelearning.org/proceedings/icml2007/papers/331.pdf.

Lee, A., Mumford, D., & Huang, J. (2001). Occlusion models for natural images: A statistical study of a scale-invariant dead leaves model. *International Journal of Computer Vision, 41*, 35–59.

Lee, H., Ekanadham, C., & Ng, A. (2008). Sparse deep belief net model for visual area V2. In J. Platt, D. Koller, Y. Singer, & S. Roweis (Eds.), *Advances in neural information processing systems 20*. Cambridge, MA: MIT Press.

Lee, H., Grosse, R., Ranganath, R., & Ng, A. Y. (2009). Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning* (pp. 609–616). New York: ACM. doi:http://doi.acm.org/10.1145/1553374.1553453.

Lewicki, M., & Olshausen, B. (1999). A probabilistic framework for the adaptation and comparison of image codes. *J. Opt. Soc. Am. A, 16*, 1587–1601.

Luettgen, M., & Willsky, A. (1995). Likelihood calculation for a class of multiscale stochastic-models, with application to texture-discrimination, *4*, 194–207.

Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In Z. Ghahramani (Ed.), *Twenty-seventh International Conference on Machine Learning*. New York: ACM.

Olshausen, B. A., & Field, D. J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature, 381*(6583), 607–609, doi:10.1038/381607a0.

Ommer, B., & Buhmann, J. M. (2010). Learning the compositional nature of visual object categories for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 32*, 501–516, doi:http://doi.ieeecomputersociety.org/10.1109/TPAMI.2009.22.

Osindero, S., & Hinton, G. E. (2008). Modeling image patches with a directed hierarchy of Markov random field. In J. C. Platt, D. Köller, Y. Singer, & S. Roweis (Eds.), *Neural information processing systems, 20*. Cambridge, MA: MIT Press.

Raina, R., Madhavan, A., & Ng, A. Y. (2009). Large-scale deep unsupervised learning using graphics processors. In *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning* (pp. 873–880). New York: ACM, doi:http://doi.acm.org/10.1145/1553374.1553486.

Roth, S., & Black, M. J. (2005). Fields of experts: A framework for learning image priors. *IEEE Conference on Computer Vision and Pattern Recognition, 2*, 860–867.

Salakhutdinov, R. (2009). Learning in Markov random fields using tempered transitions. In Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, A. Culota (Eds.), *Advances in neural information processing systems, 22*. Cambridge, MA: MIT Press.

Salakhutdinov, R., & Murray, I. (2008). On the quantitative analysis of deep belief networks. In *Proceedings of the 25th Annual International Conference on Machine Learning*. Madison, WI: Omnipress.

Smolensky, P. (1986). Information processing in dynamical systems: Foundations of harmony theory. In D. E. Rumelhart & J. L. McClelland (Eds.), *Parallel distributed processing* (Vol. 1, pp. 194–281). Cambridge, MA: MIT Press.

Storkey, A. J., & Williams, C. K. I. (2003). Image modeling with position-encoding dynamic trees. *IEEE Trans. Pattern Anal. Mach. Intell., 25*, 859–871.

Tieleman, T. (2008). Training restricted Boltzmann machines using approximations to the likelihood gradient. In *Proceedings of the International Conference on Machine Learning* (Vol. 25). New York: ACM.

Todorovic, S., & Ahuja, N. (2008). Unsupervised category modeling, recognition, and segmentation in images. *IEEE Trans. Pattern Anal. Mach. Intell., 30*, 2158–2174, doi:http://dx.doi.org/10.1109/TPAMI.2008.24.

Tu, Z., Chen, X., Yuille, A. L., & Zhu, S.-C. (2005). Image parsing: Unifying segmentation, detection, and recognition. *Int. J. Comput. Vision, 63*, 113–140, doi:http://dx.doi.org/10.1007/s11263-005-6642-x.

Welling, M., Rosen-Zvi, M., & Hinton, G. E. (2005). Exponential family harmoniums with an application to information retrieval. In L. K. Saul, Y. Weiss, & L. Bottou (Eds.), *Advances in neural information processing systems, 17*. Cambridge, MA: MIT Press.

Williams, C., & Adams, N. (1999). DTs: Dynamic trees. In M. Kearns, S. Solla, and D. Cohn (Eds.), *Advances in neural information processing systems, 11* (pp. 634–640). Cambridge, MA: MIT Press.

Williams, C. K. I., & Titsias, M. K. (2004). Greedy learning of multiple objects in images using robust statistics and factorial learning. *Neural Comput., 16*, 1039–1062, doi:http://dx.doi.org/10.1162/089976604773135096.

Winn, J. M., & Jojic, N. (2005). LOCUS: Learning object classes with unsupervised segmentation. In *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision* (Vol. 1, pp. 756–763). IEEE Computer Society. doi:http://dx.doi.org/10.1109/ICCV.2005.148.

Zhu, L. L., Lin, C., Huang, H., Chen, Y., & Yuille, A. (2008). Unsupervised structure learning: Hierarchical recursive composition, suspicious coincidence and competitive exclusion. In *ECCV '08: Proceedings of the 10th European Conference on Computer Vision* (pp. 759–773). Berlin: Springer-Verlag. doi:http://dx.doi.org/10.1007/978-3-540-88688-4_56.

Zhu, S., & Mumford, D. (2006). A stochastic grammar of images. *Found. Trends. Comput. Graph. Vis., 2*, 259–362, doi:http://dx.doi.org/10.1561/0600000018.

Received January 20, 2010; accepted August 11, 2010.