Supplementary Material

A M-Step Equation Derivations

The optimal parameters that maximize the data log-likelihood under the generative model can be sought by Expectation Maximization (EM) algorithm (see eg., [16]), which iteratively optimizes a lower bound $\mathcal{F}(\Theta, q)$ of the likelihood w.r.t. the parameters Θ and a distribution q:

$$\mathcal{L}(\Theta) \geq \mathcal{F}(\Theta, q_{\Theta'}) = \sum_{n=1}^{N} \sum_{s} q_n(\mathbf{s}|\Theta') \log \frac{p(y^{(n)}, \mathbf{s}|\Theta)}{q_n(\mathbf{s}|\Theta')} \tag{19}$$

$$= \langle \log p(\mathbf{y}, \mathbf{s} \mid \Theta) \rangle_{q(\mathbf{s}|\Theta')} + \mathrm{H}[q(\mathbf{s}|\Theta')]. \tag{20}$$

Each iteration consists of an E-step and an M-step. The E-step optimizes the lower bound w.r.t. to the distributions $q_n(s \mid \Theta)$ by setting them equal to the posterior distributions $q_n(s \mid \Theta) \leftarrow p(s \mid y^{(n)}, \Theta)$ while keeping the parameters Θ fixed, denoted by Θ'. The M-step then optimizes $\mathcal{F}(\Theta, q_{\Theta'})$ w.r.t. the parameters Θ keeping the distributions $q_n(s \mid \Theta')$ fixed. If we are given many samples of s for the posterior then we wish to find:

$$\Theta^{(t+1)} = \mathrm{argmax}_{\Theta} \mathcal{F}(\Theta, q_{\Theta^{(t)}}). \tag{21}$$

This is maximised with the maximum likelihood estimate:

$$\Theta^{(t+1)} = \mathrm{argmax}_{\Theta} \langle \log p(\mathbf{y}, \mathbf{s} \mid \Theta) \rangle_{q(\mathbf{s}|\Theta^{(t)})}. \tag{22}$$

To keep the derivation focused, we present a simple derivation of the update equations only for a single element of W. The other parameters are similarly derived and are not covered here. For pedagogical purposes we first derive an update equation *without* a max rule, then we show how this rule should be modified when the max rule is used. Assuming the data $y^{(n)}$ is distributed as follows:

$$y^{(n)} = w s^{(n)} + \varepsilon \tag{23}$$

where $\varepsilon \sim \mathcal{N}(\mu = 0; \sigma^2)$. for w. This gives the conditional probability as:

$$p(y^{(n)} \mid s^{(n)}, w) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{y^{(n)} - w s^{(n)}}{\sigma}\right)^2} \tag{24}$$

In log space this is a quadratic function:

$$\log p(y^{(n)} \mid s^{(n)}, w) = c - \log \sigma - \frac{1}{2}\left(\frac{y^{(n)} - w s^{(n)}}{\sigma}\right)^2 \tag{25}$$

and is summed over all datapoints n. The maximum likelihood solution differentiates this sum with respect to w (this function is linear in σ and when differentiated σ can be discarded) to find the maximum:

$$\frac{d}{dw}\left[\sum_{n}\left(y^{(n)} - s^{(n)} w\right)^2\right] = 0. \tag{26}$$

From which the maximum is given by:

$$w = \frac{\sum_n s^{(n)} w^{(n)}}{\sum_n s^{(n)2}}. \tag{27}$$

However we care about finding the ML solution for the max rule:

$$y^{(n)} = \max_h \left\{ W_h s_h^{(n)} \right\} + \varepsilon \tag{28}$$

If the new estimates of W_h do not change significantly then the simple derivation for w will apply to W_h, but only the data for which W_h is the maximum will be used. The data is going to vary over: the number of images N, the number of samples per image K, and we will estimate W_{hd} per latent dimension h and observed dimension (or pixel) d. This leads to:

$$W_{hd} = \frac{\sum_n^N \sum_k^K \delta(\text{h is max}) s_{hn}^{(k)} y_d^{(n)}}{\sum_n^N \sum_k^K \delta(\text{h is max})\ {s_{hn}^{(k)}}^2} \tag{29}$$

which corresponds to the results given in equation 9 of the main paper. $\delta(\text{h is max})$ is used to identify the index for which $W_{hd} s_{hn}^k$ is the maximal cause of the data, if it is not the maximal cause, then $\delta()$ returns 0, and the term does not contribute to the sum.

B Gibbs Sampler

Our likelihood can be expressed as follows:

$$\log p(y_d|s_h, \mathbf{s}_{H\setminus h}, \theta) \;\; = m(s_h) = \begin{cases} m_1(s_h) & s < P_{\delta(1)} \\ m_2(s_h) & P_{\delta(1)} \leq s < P_{\delta(2)} \\ m_3(s_h) & P_{\delta(2)} \leq s < P_{\delta(3)} \\ \vdots & \vdots \\ m_{D+1}(s_h) & P_{\delta(D)} \leq s, \end{cases} \tag{30}$$

and the summation of all these segments of the peicewise function of our likelihood (12) is expressed with:

$$m_i(s_h) = \sum_{j=1}^{j-1} r_{\delta(j')}(s_h) + \sum_{u=1}^{D} l_{\delta(u)}(s_h) \qquad \text{for} \quad 1 \leq i \leq D+1. \tag{31}$$

The left and right sides of each transition point P_d can be formulated as

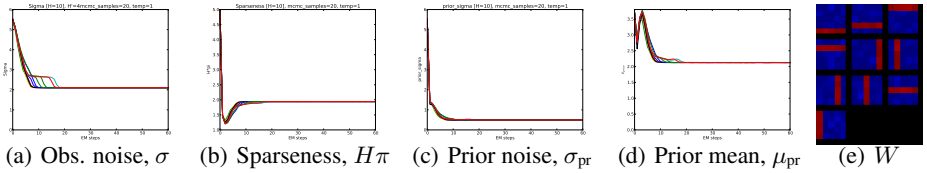
$$l_d(s_h) = -\frac{1}{2}\log(2\pi) - \log(\sigma) + \frac{1}{2\sigma^2}(y_d - \max_{h'\setminus h}\{W_{dh'}s'_h\}) \tag{32}$$

$$r_d(s_h) = -\frac{1}{2}\log(2\pi) - \log(\sigma) + \frac{1}{2\sigma^2}(y_d - W_{dh}s_h)^2, \tag{33}$$

and with this calculation, the operation in Eq. 31 can be elegantly expressed by computing three coefficients for each segment $m_i(s_h)$.

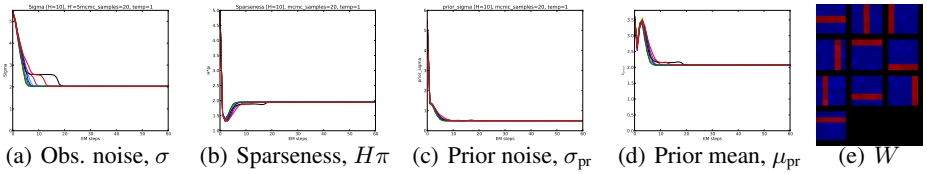
C Experiments: Artificial Data

We ran several sets of experiments on the exact same ground-truth dataset described and displayed in 4, but varied the number of latents H' we select, considering the range $H' \in (4, 10)$ where $H = 10$. All experiments recover ground-truth parameters $\Theta = (\pi, \mu_{\text{pr}}, \sigma_{\text{pr}}, W, \sigma)$. Our algorithm reliably converges to globally optimal solutions of all parameters in all runs of the experiments with 30 EM iterations. Shown are are a few examples.



(a) Obs. noise, σ (b) Sparseness, $H\pi$ (c) Prior noise, σ_{pr} (d) Prior mean, μ_{pr} (e) W

Figure 6: Parameter recovery results for $H' = 4$.



(a) Obs. noise, σ (b) Sparseness, $H\pi$ (c) Prior noise, σ_{pr} (d) Prior mean, μ_{pr} (e) W

Figure 7: Parameter recovery results for $H' = 5$.

D Experiments: Natural Image Patches

Here we show further results from the experiment on $N = 50,000$ preprocessed and channel split natural image patches of 16×16 pixel.

First we relate the generative fields that are learned from image patches to their corresponding receptive fields as measured in biological systems. In order to do so we carried out several reverse correlation experiments, with the aim of identifying the relationship between the generative fields and the reverse-correlation fields. To calculate the reverse correlation for a single latent variable we calculate the average activation for a particular image (since the code is sparse, most of the time activations will be zero). The images are then averaged together, weighted by the average activation.

In Fig. 8**A** we show the generative fields that are learned with our method. Fig. 8**B** shows the receptive fields obtained by estimating the first order linear mapping from input to hidden units. The mapping is estimated by combining the preprocessing (a linear mapping with a kernel for pseudo-whitening) with the mapping obtained by reverse correlation using preprocessed patches. As can be observed by comparing Fig. 8**A** and **B**, the qualitative and quantitative shapes of generative and receptive fields are essentially equal. For the results in Fig. 5 we used the receptive field estimates in Fig. 8**B**.

For further analysis, we matched the fields with Gabor and DoG functions. to determine the number of Gabor-like and DoG fields. Afterwards, we plotted the distribution of Gabor shapes in an n_x/n_y plot as was first suggested in an experimental study [21]. We do not include the receptive fields classified as DoG fields for the plot because such center-surround fields cannot be matched with Gabor functions (Gabors do not parameterize DoGs except of the degenerate case of Gaussian functions). As a consequence, the cluster of field shapes close to origin (in [21] associated with globular fields) is not observed in the plot of Fig. 5.

As an alternative to estimating only the second part of the mapping by reverse correlation, we can also estimate the whole mapping using raw (unprocessed) image patches as samples. As a control, Fig. 8**C** shows the estimates obtained in this way. The colors of the image are significantly different as a result of the different range. A preprocessed image patch has a range from [-10..+10] with a large proportion of the image taking value zero. The original image patches have a range [0..1] and a non-zero mean value. While this is the main source of the differences, the basic shapes are again the same as for the fields in **A** and **B**.

Finally, regarding the n_x/n_y analysis, note that the type of preprocessing can have a significant impact on the Gabor statistics. This includes an impact of separate on- and off-channels (also compare [34] for a study of different channels within cortical layers).

[34] C. I. Yeh, D. Xing and R. M. Shapley. "Black" responses dominate macaque primary visual cortex V1. *The Journal of Neuroscience*, 29(38): 11753–60, 2009.

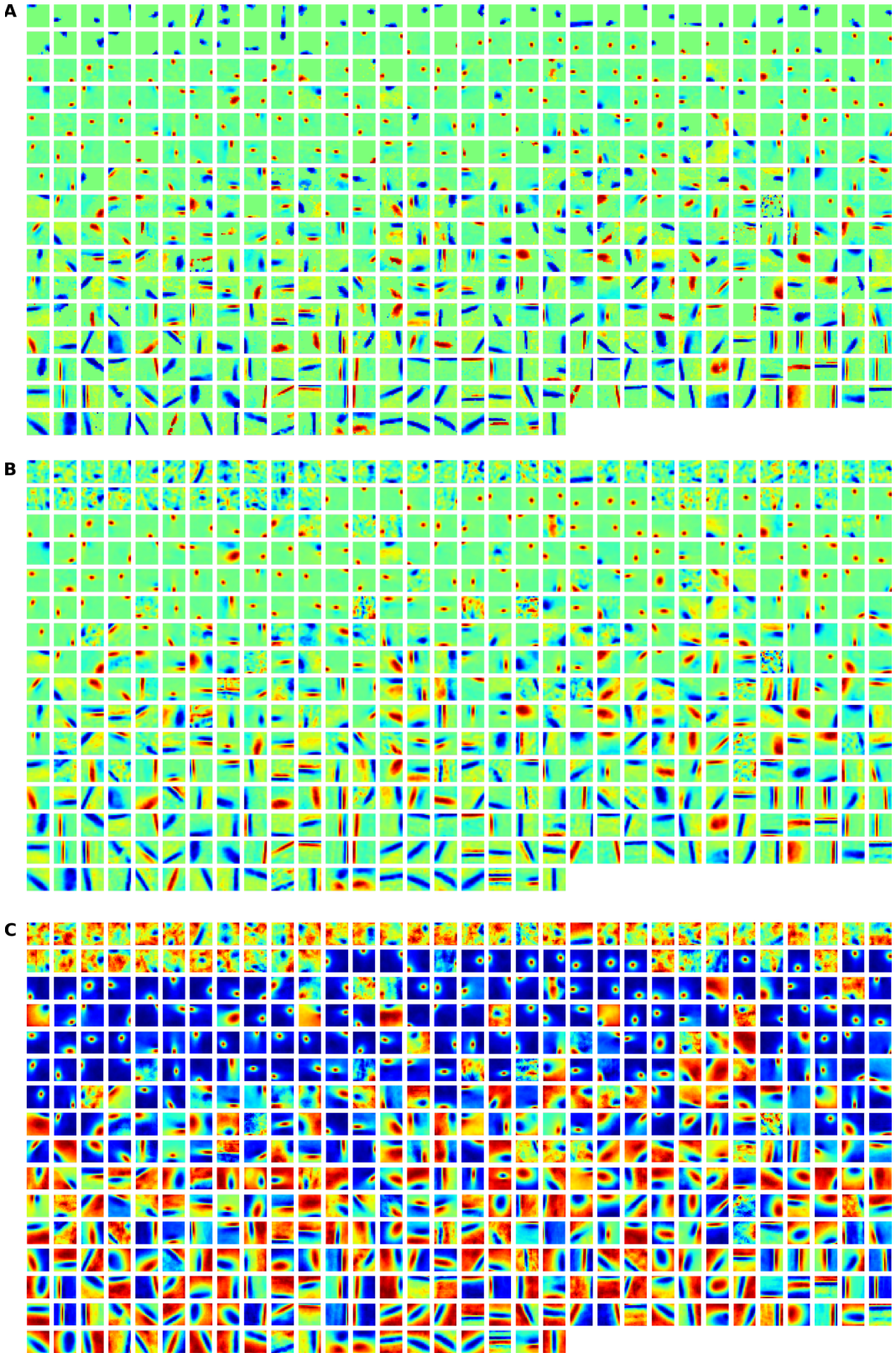


Figure 8: **A** Full set of $H = 500$ learned generative fields ($\mathbf{W}_h$). **B** Fields after reverse correlation with preprocessed input patches and **C** after reverse correlation with raw input patches.

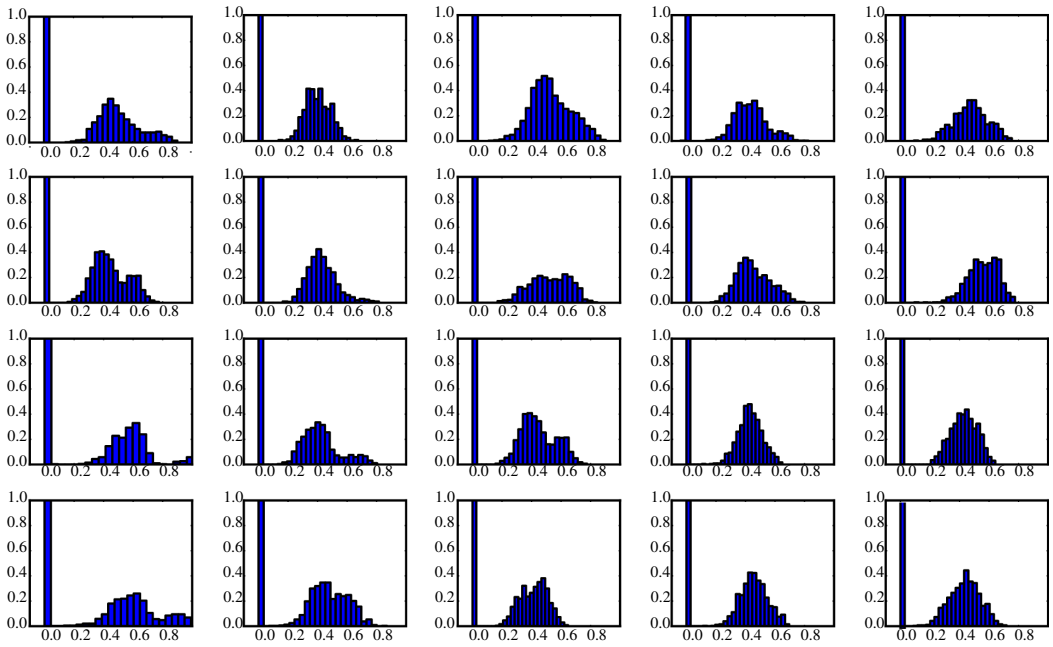
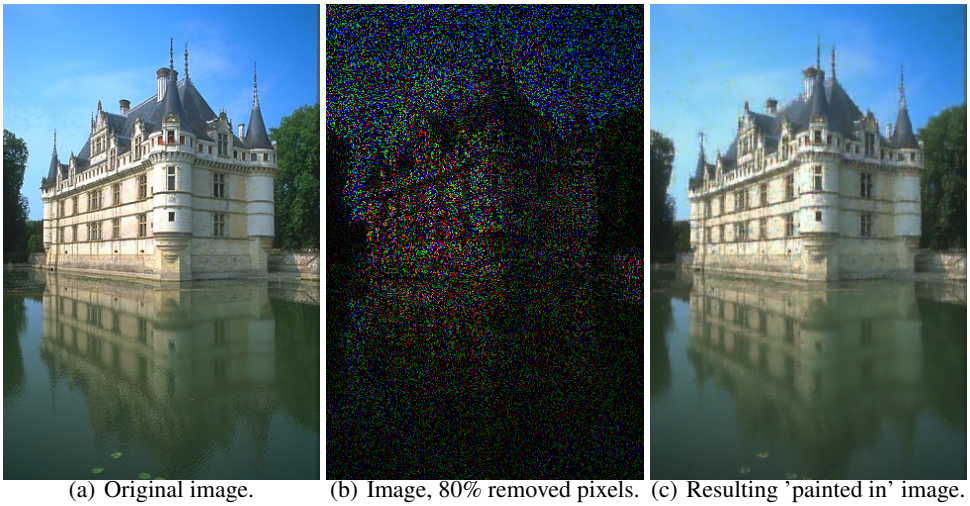


Figure 9: Histogram of the latent activations of the 20 most often active dictionary elements. This corresponds to the prior over latent units that we have assumed in our model, thus supporting the consistency of the model.

E Experiments: Image Inpainting

As a sanity check we tested our model on a well-known task: image inpainting. Given an image (a) we can degrade the image by deleting 80% of the pixel values to create image (b). Using nothing other than image (b) we are able to train a model of image patches which can then be used to fill in the rest of the image pixels (c). Our images patches are 8×8 in size and split into 3 channels for red, green and blue pixels. We ensure that missing pixels never contribute to the inference, which is a cleaner and less computationally-demanding method than filling in missing pixels with best estimates, as is sometimes reported [5]. While drawing samples from the posterior $p(s \mid y, \Theta)$ we also kept track of the mean prediction: $\max_h\{w_{dh}s_h\}$. After seeing all posible image patches, we can use the mean prediction as our best estimate of the picture. Since we have ground truth we are able to evaluate the predictive accuracy. We obtained a peak signal-to-noise ratio of 26. This is comparable to other methods [5, 33], and we hope that this will reach state-of-the-art with further refinements to the model (for example allowing individual means for each spike and slab, or allowing different noise estimates for each pixel in the image patch). As image restoration is not the main thrust of this paper these extensions will be explored in further work.

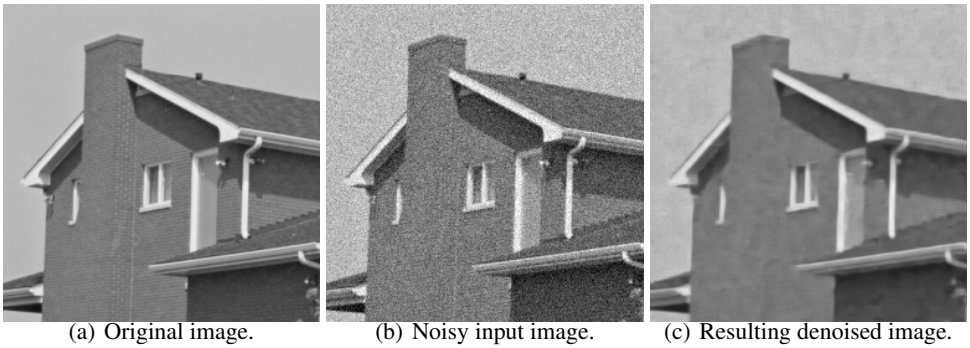


(a) Original image. (b) Image, 80% removed pixels. (c) Resulting 'painted in' image.

Figure 10: Application of the algorithm to image inpainting.

F Experiments: Image Denoising

Another possible use of our image model is denoising an image. Given an image (a) we add Gaussian noise (zero mean with standard deviation 25) to obtain image (b). Using only image (b) we train a model of natural patches which can then be used to infer what the original image was. This inference is shown in image (c). We get a Peak Signal-to-Noise ratio of 31 which is comparable to [5, 33].



(a) Original image. (b) Noisy input image. (c) Resulting denoised image.

Figure 11: Application of the algorithm to image denoising.