Discrete Sparse Coding

Georgios Exarchakis

georgios.exarchakis@uol.de

Jörg Lücke

joerg.luecke@uol.de

Machine Learning Lab

Cluster of Excellence Hearing4all and Department for Medical Physics and Acoustics

Carl-von-Ossietzky University Oldenburg, 26111 Oldenburg, Germany

Keywords: Sparse Coding, Generative Models, Probabilistic Modeling, Approximate Inference, Expectation Maximization

Abstract

Sparse Coding algorithms with continuous latent variables have been the subject of a large number of studies. However, discrete latent spaces for sparse coding have been largely ignored. In this work, we study sparse coding with latents described by discrete

instead of continuous prior distributions. We consider the general case in which the latents (while being sparse) can take on any value of a finite set of possible values; and in which we learn the prior probability of any value from data. The studied approach can be applied to any data generated by discrete causes; and it can be applied as an approximation of continuous causes. As the prior probabilities are learned, the approach then allows for estimating the prior shape without assuming specific functional forms. To efficiently train the parameters of our probabilistic generative model, we apply a truncated EM approach (Expectation Truncation) that we modify to work with a general discrete prior. We evaluate the performance of the algorithm by applying it to a variety of tasks: (A) We use artificial data to verify that the algorithm can recover the generating parameters from a random initialization. (B) We use image patches of natural images and discuss the role of the prior for the extraction of image components. (C) We use extra-cellular recordings of neurons to present a novel method of analysis for spiking neurons that includes an intuitive discretization strategy. And (D), we apply the algorithm on the task of encoding audio waveforms of human speech. The diverse set of numerical experiments presented in this work suggests that discrete sparse coding algorithms can scale efficiently to work with realistic datasets and provide novel statistical quantities to describe the structure of the data.

1 Introduction

Sparse Coding (SC; Olshausen and Field, 1997) was proposed as the neural coding strategy of simple cells in the primary visual cortex of the mammalian brain, and it has

since become a prominent information encoding paradigm on a diverse set of applications. The arguments in favor of sparsity stem from multiple research directions: classical computer vision results (Field, 1994), observed sparsity in brain recordings (Hubel and Wiesel, 1977), and the idea that the generative process of natural data consists of sparsely presented structural elements (Olshausen and Field, 1997; Field, 1994). In this work we focus on the latter idea, i.e., that it is common to perceive natural datasets as a large set of distinct structural elements that appear infrequently. Pursuing distinct selectivity of features in the data, as opposed to obfuscated overlapping responsibilities in earlier Gaussian approaches (Hancock et al., 1992), SC has commonly been associated with heavy tailed prior distributions. However, it is often argued that the SC principle encourages discrete distributions or distributions with a discrete component (Rehn and Sommer, 2007; Titsias and Lázaro-Gredilla, 2011; Goodfellow et al., 2012; Sheikh et al., 2014) since continuous distributions do not send a clear "yes" or "no" signal for the features that constitute a datapoint. Similarly, it is frequently pointed out in the closely related field of compressive sensing (see, e.g., Donoho, 2006; Eldar and Kutyniok, 2012; Sparrer and Fischer, 2014) that "hard" sparsity (in the form of an l_0 sparsity penalty) is preferable to softer sparsity as it would be reflected by continuous prior distributions.

Previous investigations of sparse coding with hard sparsity have been constrained in different ways. Most frequently a prior with hand-set parameters was used, and the prior remained unchanged during learning (e.g. Haft et al., 2004; Rehn and Sommer, 2007). For priors over discrete latents, the values the latents can take on are constrained – most frequently to take on the values zero or one (e.g. Haft et al., 2004; Henniges

et al., 2010; Bornschein et al., 2013). The aim of this study is the derivation of a sparse coding algorithm for discrete latents without any of these constraints, i.e., an algorithm applicable for sparse discrete latents that can take on any (finite) set of values. Furthermore, we aim at learning the prior probability of any of these discrete values. Such a general algorithm for discrete sparse latents can be applied in essentially two ways: (A) To optimally infer parameters for data generated by sparse discrete latents; (B) As discrete approximation of data generated by a continuous sparse prior. The advantage of the former type of application is the ability to directly model the discrete nature of the generating process and to infer its parameters including prior parameters (while the approach is not limited to any particular values such as zero and one). The advantage of the latter type of application is the absence of assumptions on the prior shape. The generality of the approach allows a latent to take on a specific value with any prior probability. While representing a discretized version of an underlying continuous distribution, the generality will allow for learning any prior shape. Any continuous sparse coding approach, in contrast, has to assume a specific functional form of the prior, be it a Cauchy or Laplace prior (Olshausen and Field, 1997, and many more), Student-t (Berkes et al., 2008) or another heavy-tail prior.

Overcoming the computational demand of training a general discrete sparse coding model, which goes much beyond the complexity of earlier approaches, will be a main challenge of this study. All non-Gaussian encodings of hidden variables typically pose difficulties in Machine Learning. While efficient approaches have been developed for approaches such as independent component analysis (Bell and Sejnowski, 1997; Bingham and Hyvärinen, 2000) or non-negative matrix factorization (Lee and Seung, 1999),

we usually face severe analytical intractabilities if data noise is taken into account. For typical sparse coding models, we are therefore forced to apply approximation schemes (e.g. Olshausen and Field, 1997; Lee et al., 2007; Berkes et al., 2008; Mairal et al., 2010) to obtain efficient learning algorithms for parameter optimization. Several techniques have been used to overcome that problem (Aharon et al., 2006) based on either sophisticated point estimates of the posterior mode or sampling based methods (e.g. Berkes et al., 2008). Each of these methods offers its own set of advantages and disadvantages. Methods based on point estimates tend to be computationally efficient by avoiding the intricacies of dealing with uncertainty in the posterior, for instance. Sampling based methods, on the other hand, offer a more advanced description of the posterior but usually at a cost of either computational complexity or convergence speed. In the case of discrete hidden variables, it is straight-forward to derive exact analytical solutions for the optimization of model parameters within the expectation maximization (EM) approach (e.g. Haft et al., 2004; Henniges et al., 2010, for binary latents) but such exact solutions scale very poorly with the number of latent dimensions. In order to overcome poor scalability during learning and inference for sparse coding with binary latents, factored or truncated variational approximations to a-posterior distributions have been used (Haft et al., 2004; Bingham et al., 2009; Henniges et al., 2010). Like in the continuous case, also sampling offers itself as a well-established and efficient approach (see, e.g., Zhou et al., 2009, for a 'hard' sparsity model or Griffiths and Ghahramani, 2011, for a non-parametric approach with binary latents). In practice, however, deterministic factored or truncated approaches are frequently preferred (Haft et al., 2004; Zhou et al., 2009; Titsias and Lázaro-Gredilla, 2011; Sheikh et al., 2014) presumably due to their

computational benefits in high dimensional hidden spaces. For discrete latents, truncated approximations to intractable posteriors (Lücke and Eggert, 2010; Puertas et al., 2010; Exarchakis et al., 2012; Henniges et al., 2014) have represented an alternative to sampling and factored variational methods. Like sampling (but unlike factored variational methods), truncated approximations do not make the assumption of *a-posteriori* independence. Like factored approaches (but unlike sampling), truncated approximations have been shown to be very efficient also in spaces with a very large number of hidden variables (Shelton et al., 2011; Sheikh et al., 2014). Truncated approaches can be expected to represent very accurate approximations if posterior masses are concentrated on relatively few states, which makes them well suited for our purposes.

To demonstrate the newly derived approach and its capabilities, we use different types of data in order to validate the approach and to demonstrate its different types of applicability. First, we demonstrate the effectiveness of the training scheme on artificial data to better expose the intricacies of the learning procedure. We continue by testing the model with different sets of discrete values on natural images and aim at inferring prior shapes with a minimal scientific bias, in the course of this work we are also verifying the validity of the model by replicating and confirming preliminary earlier results (Henniges et al., 2010; Exarchakis et al., 2012) as special cases of our approach. Furthermore, we perform an analysis of data captured through extra-cellular recordings of spiking neurons using discrete latent values that account for background activity as well as potential decays that occur in spike trains. Common methods of analysis of extra-cellular recordings use intricate pipelines for spike detection and identification and often rely on Gaussian priors to characterize a spike even though the spike is per-

ceived as a discrete quantity. Here, we propose a model that explicitly takes into account the discrete nature of spikes as well as their varying amplitudes, which are due to spike trains, as well as potential overlaps with spikes of nearby neurons. Finally, we apply the model on a feature extraction task from human speech using the raw waveform. The discrete prior we learn in this case can be taken to model an underlying continuous prior without the requirement to make assumptions on the prior shape. On the other hand, as speech makes use of resonances in the vocal tract, we can also expect a certain degree of discreteness in the underlying generation process which would also motivate the application of a model with discrete latents.

In Sec. 2 we will define the model and derive a learning algorithm based on truncated EM. In Sec. 3 we apply the model to artificial data (Sec. 3.1), to image patches (Sec. 3.2), to extra cellular neural recordings (Sec. 3.3), and to auditory data (Sec. 3.4). Secs. 3.1 ad 3.2 are examples of discrete hidden causes, and Secs. 3.3 and 3.4 are examples for how the generality of the approach can be used to learn prior shapes for presumably rather continuous latents. Sec. 4 discusses the model, algorithm and numerical results.

2 Model Definition

Consider a set, $\mathcal{Y}$, of N independent datapoints $\vec{y}^{(n)}$, with $n = 1, \ldots, N$, where $\vec{y}^{(n)} \in \mathbb{R}^D$. For these data the studied learning algorithm seeks parameters $\Theta^* = \{W^*, \sigma^*, \vec{\pi}^*\}$ that maximize the data log-likelihood:

$$L\left(\mathcal{Y}|\Theta\right) = \log \prod_{n=1}^{N} p\left(\vec{y}^{(n)}|\Theta\right) = \sum_{n=1}^{N} \log p\left(\vec{y}^{(n)}|\Theta\right)$$

Sparse coding models are latent variable models and therefore the likelihood is defined as a function of unobserved random variables as follows

$$L\left(\mathcal{Y}|\Theta\right) = \sum_{n=1}^{N} \log p\left(\vec{y}^{(n)}|\Theta\right) = \sum_{n=1}^{N} \log \sum_{\vec{s}} p\left(\vec{y}^{(n)}|\vec{s},\Theta\right) p\left(\vec{s}|\Theta\right) \tag{1}$$

where the latent variables $\vec{s}$ are taken to have discrete values, and where the sum $\sum_{\vec{s}}$ goes over all possible vectors $\vec{s}$, i.e., over all possible combinations of discrete states. Let $\vec{s}$ be of length H, i.e. $\vec{s} = (s_1, \ldots, s_H)^T$, where each element s_h can take on one of K discrete values $\phi_k \in \mathbb{R}$, i.e. $\vec{s}_h \in \Phi = \{\phi_1, \ldots, \phi_K\}$. For such latents, we can define the following prior:

$$p(\vec{s}\,|\,\Theta) \;=\; \prod_{h=1}^{H}\prod_{k=1}^{K} \pi_k^{\delta(\phi_k = s_h)}, \quad \text{with} \quad \sum_{k=1}^{K} \pi_k = 1, \tag{2}$$

where $\delta\left(\phi_k = s_h\right)$ is an indicator function which is one if and only if $\phi_k = s_h$ and zero otherwise. As for standard sparse coding, Equation 2 assumes independent and identical distributions for the latents s_h. The prior will be used to model sparse activity by demanding one of the values in $\Phi = \{\phi_1, \ldots, \phi_K\}$ to be zero and the corresponding probability to be relatively high. We will refer to the set of possible values Φ as a *configuration*. An example is to choose configuration $\Phi = \{0, 1\}$, which reduces (using $\pi = \pi_2$ and $(1-\pi) = \pi_1$) the prior (2) to the Bernoulli prior $p(\vec{s}\,|\,\Theta) \;=\; \prod_{h=1}^{H} \pi^{s_h} \left(1-\pi\right)^{1-s_h}$ (as used for binary sparse coding, Haft et al., 2004; Henniges et al., 2010; Bornschein et al., 2013). The notation used in (2) is similar to a categorical distribution but applies for latents with any values ϕ_k with any probabilities π_k. Its form will be convenient for later derivations.

Having defined the prior (2), we assume the observed variables $\vec{y} = (y_1, \ldots, y_D)^T$ to be generated as in standard sparse coding, i.e., we assume Gaussian noise with the

mean set by a linear superposition of the latents:

$$p\left(\vec{y}\,|\,\vec{s},\Theta\right) = \mathcal{N}\left(\vec{y};W\vec{s},\sigma^2\mathbb{1}\right) \tag{3}$$

with an isotropic covariance, $\sigma^2\mathbb{1}$, and mean $W\vec{s}$. We call the data model defined by (2) and (3) the discrete sparse coding (DSC) data model.

Given a set of datapoints $\vec{y}^{(1)},\ldots,\vec{y}^{(N)}$ and the DSC data model, we now seek parameters $\Theta = (\vec{\pi},W,\sigma)$ that maximize the likelihood (1). We derive parameter update equations using Expectation Maximization in its free-energy formulation (Neal and Hinton, 1998). In our case, exact EM update equations can be derived in closed-form but the E-step scales with the number of hidden states $\mathcal{O}(K^H)$, making the algorithm computationally intractable for large H.

In order to derive computationally tractable approximations for parameter optimization, we approximate the intractable a-posteriori probabilities $p(\vec{s}\,|\,\vec{y},\Theta)$ by a truncated distribution:

$$p(\vec{s}\,|\,\vec{y}^{(n)},\Theta) \;\approx\; q^{(n)}(\vec{s};\Theta) \;=\; \frac{p(\vec{s}\,|\,\vec{y}^{(n)},\Theta)}{\sum_{\vec{s}'\in\mathcal{K}^{(n)}} p(\vec{s}'\,|\,\vec{y}^{(n)},\Theta)}\,\delta(\vec{s}\in\mathcal{K}^{(n)}), \tag{4}$$

where $\mathcal{K}^{(n)}$ is a subset of the set of all states, $\mathcal{K}^{(n)}\subseteq\{\phi_1,\ldots,\phi_K\}^H$, and $\delta(\vec{s}\in\mathcal{K}^{(n)})$ is again an indicator function (one if $\vec{s}\in\mathcal{K}^{(n)}$ and zero otherwise).

While truncated approximations have been shown to represent efficient approximation of high accuracy for a number of sparse coding generative models (Lücke and Eggert, 2010; Bornschein et al., 2013; Henniges et al., 2014), they have so far only been applied to binary sparse coefficients. Here, we will generalize the application of truncated distributions to sparse latents with any (finite) number of discrete states.

Considering (4), we can first note that the assumptions for applying Expectation

Truncation (ET; Lücke and Eggert, 2010) are fulfilled for the DSC model (2) and (3) such that we can derive a tractable free-energy given by:

$$\mathcal{F}\left(q,\Theta\right)=\sum_{n\in\mathcal{M}}\left[\sum_{\vec{s}}q^{(n)}\left(\vec{s};\Theta^{\mathrm{old}}\right)\left(\log p\left(\vec{y}^{(n)},\vec{s}\,|\,\Theta\right)\right)\right]+H\left(q\right) \tag{5}$$

where $q^{(n)}\left(\vec{s};\Theta^{\mathrm{old}}\right)$ is given in (4) and where $H(q)$ is the Shannon entropy. Notice that the summation over datapoints is no longer over the index set $\{1,\ldots,N\}$ but over a subset $\mathcal{M}$ of those datapoints that are best explained by the model. Since we use a truncated posterior distribution we expect that we do not explain well the entire dataset but rather a subset of it of size $\sum_{\vec{s}'\in\mathcal{K}^{(n)}}p(\vec{s}'|\Theta)/\sum_{\vec{s}}p(\vec{s}|\Theta)$. To populate $\mathcal{M}$ we use the datapoints with the highest value for $\sum_{\vec{s}\in\mathcal{K}^{(n)}}p\left(\vec{s},\vec{y}^{(n)}|\Theta^{old}\right)$. It can be shown for a large class of generative models (including the DSC model) (Lücke and Eggert, 2010), that maximizing the free-energy (5) then approximately maximizes the likelihood for the full dataset.

To get the optimal parameters for the model $\Theta^{*}=\{\vec{\pi}^{*},W^{*},\sigma^{*}\}$ we take the gradient of the free energy and seek the values of the parameters that set it to 0:

$$\begin{aligned}\nabla\mathcal{F}\left(q,\Theta\right)&=\nabla\sum_{n\in\mathcal{M}}\left[\left\langle\log p\left(\vec{y}^{(n)}|\vec{s},\Theta\right)\right\rangle_{q^{(n)}}+\left\langle\log p\left(\vec{s}|\Theta\right)\right\rangle_{q^{(n)}}\right]\\&=\nabla\sum_{n\in\mathcal{M}}\left[\left\langle-\frac{D}{2}\log\left(2\pi\sigma^{2}\right)-\frac{\sigma^{2}}{2}\|\vec{y}^{(n)}-W\vec{s}\|_{2}^{2}\right\rangle_{q^{(n)}}\right.\\&\qquad\left.+\left\langle\sum_{h,k}\delta\left(\phi_{k},s_{h}\right)\log\pi_{k}\right\rangle_{q^{(n)}}\right]=0\,,\end{aligned}$$

where we denote with $\left\langle g\left(\vec{s}\right)\right\rangle_{q(n)}$ the expectation value of a function $g\left(\vec{s}\right)$ under the distribution $q^{(n)}\left(\vec{s};\Theta^{old}\right)$. For W and σ the results are

$$\nabla_{W}\mathcal{F}\left(q,\Theta\right)=0\Leftrightarrow W^{*}=\left(\sum_{n\in\mathcal{M}}\vec{y}^{(n)}\left\langle\vec{s}^{T}\right\rangle_{q^{(n)}}\right)\left(\sum_{n\in\mathcal{M}}\left\langle\vec{s}\vec{s}^{T}\right\rangle_{q^{(n)}}\right)^{-1} \tag{6}$$

$$\nabla_{\sigma}\mathcal{F}\left(q,\Theta\right)=0\Leftrightarrow\sigma^{*}=\sqrt{\frac{1}{|\mathcal{M}|D}\left\langle\sum_{n\in\mathcal{M}}\|\vec{y}^{(n)}-W\vec{s}\|_{2}^{2}\right\rangle_{q^{(n)}}}\tag{7}$$

where $|\mathcal{M}|$ is the size of the set $\mathcal{M}$.

The prior parameter π_k can be obtained in the same way if one introduces the constraint to the free energy of having $\sum_k \pi_k = 1$ to maintain the normalized prior during the gradient procedure.

$$\nabla_{\pi_k}\mathcal{F}\left(q,\Theta\right)=0\Leftrightarrow\pi_{k}^{*}=\frac{\left\langle\sum_{h}\delta\left(\vec{s_{h}},k\right)\right\rangle_{q^{(n)}}}{\left\langle\sum_{k,h}\delta\left(\vec{s_{h}},k\right)\right\rangle_{q^{(n)}}}\tag{8}$$

The parameter update equations (6), (7), and (8) require the computation of expectation values $\left\langle g\left(\vec{s}\right)\right\rangle_{q(n)}$ (the E-step). By inserting the truncated distribution (4) we obtain:

$$\left\langle g\left(\vec{s}\right)\right\rangle_{q^{(n)}}=\sum_{\vec{s}}q^{(n)}\left(\vec{s}\right)\;g\left(\vec{s}\right)=\frac{\sum_{\vec{s}\in\mathcal{K}^{(n)}}p\left(\vec{s},\vec{y}^{(n)}|\Theta^{old}\right)g\left(\vec{s}\right)}{\sum_{\vec{s}\in\mathcal{K}^{(n)}}p\left(\vec{s},\vec{y}^{(n)}|\Theta^{old}\right)}\tag{9}$$

where $g\left(\vec{s}\right)$ is a function of the hidden variable $\vec{s}$ (see parameter updates above). As can be observed, the expectation values are now computationally tractable if $|\mathcal{K}^{(n)}|$ is sufficiently small. At the same time, we can expect approximations with high accuracy if $\mathcal{K}^{(n)}$ contains the hidden variables $\vec{s}$ with the large majority of the posterior mass.

In order to select appropriate states $\mathcal{K}^{(n)}$ for a datapoint $\vec{y}$ we use the joint of each datapoint and the singleton posterior variables, i.e. variables that have only one non-zero dimension, to identify the features that are most likely to have contributed to the datapoint and only include those preselected states as in the posterior estimation. More formally, we define

$$\mathcal{K}^{(n)}=\left\{\vec{s}\,|\,\forall i\notin I^{(n)}:s_{i}=0\;\text{ and }\;\|\vec{s}\|_{0}\leq\gamma\right\}$$

where $\|\cdot\|_0$ is the non-zero counting norm and where $I^{(n)}$ is an index set that contains the indices of the H' basis functions that are most likely to have generated the datapoint $\vec{y}(n)$. The index set $I^{(n)}$ is in turn defined using a selection (or scoring) function. For our purposes, we here choose a selection function of the following form:

$$\mathcal{S}_h\left(\vec{y}^{(n)}\right) = \max_{\phi\in\Phi}\left\{p\left(s_h = \phi, s_{\not h} = 0, \vec{y}^{(n)}|\Theta^{old}\right)\right\}$$

where $s_{\not h} := \{s_i : i \in \{1,\ldots,H\} \setminus \{h\}\}$. $\mathcal{S}_h$ gives a high value for the index h if the generative field in the h-th column of W contains common structure with the datapoint $\vec{y}^{(n)}$ regardless of the discrete scaling that the model provides. In other words, the selection function uses the best matching discrete value for each generative field as for a comparison with the other generative fields. The H' fields with the largest values of $\mathcal{S}_h\left(\vec{y}^{(n)}\right)$ are then used to construct the set of states in $\mathcal{K}^{(n)}$. States can be selected in different ways including scalar products (Shelton et al., 2011). Joint probabilities are preferable as they are defined by the generative model itself (including current values for noise and priors), which has motivated our choice above. Using appropriate approximation parameters γ and H', the sets $\mathcal{K}^{(n)}$ can contain sufficiently many states to realize a very accurate approximation but sufficiently few states in order to warrant sufficiently efficient scalability with H. Crucially, H' can maintain a small value for many types of data while H increases.

The M-step equations (6) to (8) together with approximate E-step equations (9) using the truncated distributions (4) represent a learning algorithm for approximate maximization of the data likelihood under the discrete sparse coding model (Equations 2 and 3). We will refer to this algorithm as the *Discrete Sparse Coding* (DSC) algorithm.

3 Numerical Experiments

We test the DSC algorithm on four different types of data: artificial data, natural images, extra-cellular neuronal recordings, and audio data of human speech. The artificial data are generated using the DSC generative model and they are used to confirm the ability of the DSC algorithm to learn the parameters of the generative model. The other three types of data are commonly encountered in real world scientific tasks. There we show that the DSC algorithm is capable of extracting interesting structure from the data while using discrete latents and small sets of parameters. Notably, the developed algorithm will enable learning of the K parameters of the prior distribution (alongside noise and generative fields).

Similarly to the vast majority of generative models trained with variants of the EM algorithm the DSC algorithm can converge to a local optimum solution since EM is in general a gradient approach. In the case of artificial data, where the ground truth is known, we have observed that an annealing schedule can make a global optimum convergence more likely. In the absence of an annealing schedule the algorithm converges to qualitatively similar parameter sets, and often the global optimum, quite early. Since we do observe an improvement in convergence properties though we wish to utilize the same techniques in other datasets, where the ground truth is unknown. In order to identify the necessary annealing parameters, we increase them sufficiently to maintain the highest level of variability of the parameters without breaking the algorithm initially and proceed to slowly decay the annealing parameters to the point of no annealing in the first half of the training time. It is unlikely that the exact same annealing scheme have the same effect on a different dataset, however, the procedure to identifying the

annealing parameters remains the same across all applications.

In the definition of a DSC algorithm we encounter the choice of latent space dimensionality as in many other feature learning algorithms we encounter the choice of latent space dimensionality. In the case of artificial data the dimensionality is already known is set to the ground truth for learning to ensure correct extraction of the parameter set. In the case of realistic datasets we tend to use an overcomplete latent representation to illustrate the ability of the algorithm to extract a rich dictionary from the data.

3.1 Artificial Data

We used a linear bars test (Hoyer, 2002; Henniges et al., 2010) to evaluate the ability of the algorithm to recover optimal solutions for the likelihood. We generated $N = 1\,000$ datapoints using a DSC data model configuration with states $\Phi = \{-2, -1, 0, 1, 2\}$ and parameters $\vec{\pi} = (0.025, 0.075, 0.8, 0.075, 0.025)$ for the prior respectively. For the parameters $W \in \{0, 10\}^{H \times D}$, we used $H = 10$ different dictionary elements of $D = 25$ observed dimensions. To simplify the visualization of such high dimensional observations, we choose each dictionary element to resemble a distinct vertical or horizontal bar when reshaped to a 5×5 image, with the value of the bar pixels to be equal to 10 and the background 0, see Figure 1 **C**. The resulting datapoints were generated as linear superposition of the basis functions, scaled by a corresponding sample from the prior. Following the DSC generative model, we also added samples of a mean free Gaussian noise with a standard deviation of $\sigma = 2$ to the data, example datapoints can be seen in Figure 1 **A**.

Using the generated datapoints, we recovered the ground truth parameters by train-

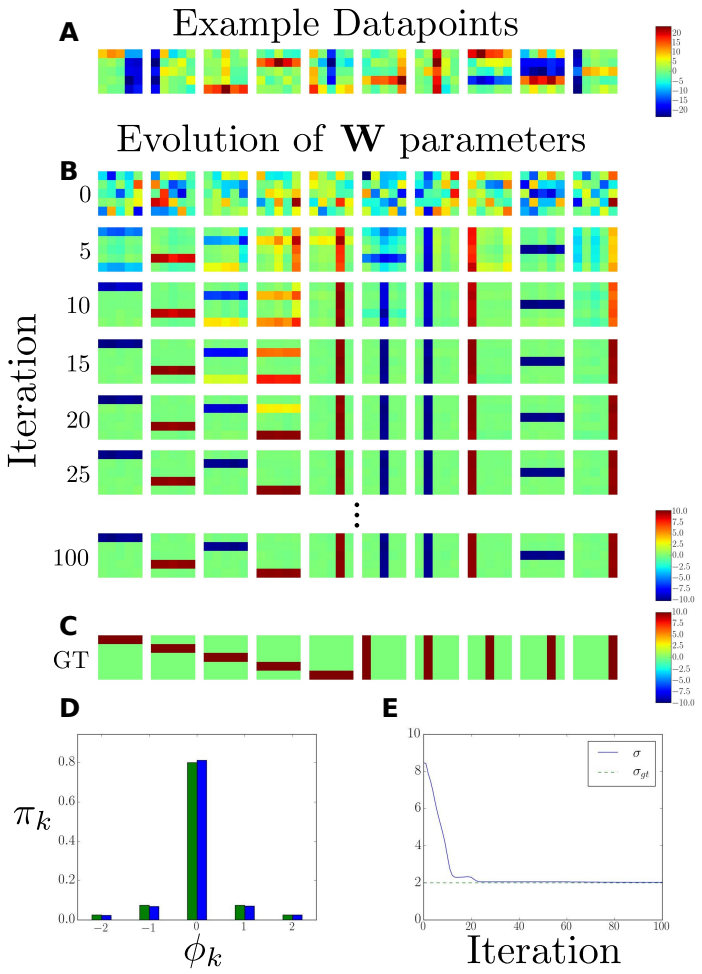

Figure 1: Results from training on artificial images using a DSC model with a configuration $\Phi = \{-2, -1, 0, 1, 2\}$. **A** Example datapoints sampled from the generative model. **B** The evolution of the dictionary over iterations. Iteration 0 shows the initial values and iteration 100 the dictionary after convergence, no interesting changes occur after iteration 25. **C** The ground truth values for the dictionary. **D** The learned prior parameters (green) compared to the ground truth prior parameters (blue). **E** The evolution of the model standard deviation (solid line) compared to the ground truth(dashed line). Notice that due to the symmetric state configuration the learned dictionary has identical structure with the ground truth but not necessarily the same sign.

ing the model as described in Section 2. To ensure that the maximum likelihood parameters are the same as the generating ones we train the model with the same latent variable dimensionality $H = 10$ as in the generating process. We initialized the standard deviation of the noise model with the standard deviation of the observed variables σ_y, the parameters W with the mean datapoint plus Gaussian noise with standard deviation of $\sigma_y/4$, and the prior parameters where initialized such that $p(\vec{s}_h = 0) = (H-1)/H$, and $p(\vec{s}_h \neq 0)$ was drawn from a uniformly random distribution and scaled to satisfy the constraint that $\sum_h p(\vec{s}_h) = 1$. The approximation parameters for the truncated approximation scheme were $H' = 7$, and $\gamma = 5$.

We ran the DSC algorithm for 100 iterations using an annealing scheme described in (Ueda and Nakano, 1998; Sahani, 1999) with the value of the β parameter to be equal to 2 for the first 10 iterations and linearly decreased to 1, no annealing, by iteration 40. Furthermore, to avoid early rejection of datapoints we used all the datapoints for training for the first 60 iterations and then proceeded to decrease the number of training datapoints linearly to $|M|$ by iteration 90.

After convergence of the algorithm, the learned parameters for σ and $\vec{\pi}$ were observed to match the generating parameters with high accuracy, see Figure 1 **D**, and **E**. For the parameters W, we don't recover the exact ground truth parameters, see Figure 1 **B**, and **C**. The reason is that in this configuration of the model, and all symmetric ones, there are multiple maximum likelihood solutions since it is equiprobable for a dictionary element to contribute with either sign. Furthermore, we noticed that some configurations of the algorithm are more likely to converge to locally optimal solutions than others.

These results show that we can successfully learn a correct dictionary for the data while at the same time learning a value that parametrizes uncertainty of the discrete coefficients and the scale of the isometric noise of the observed space.

3.2 Image Patches

Sparse Coding (Olshausen and Field, 1996) was originally proposed as a sensory coding model for simple cell receptive fields in the primary visual cortex which was able to learn biologically plausible filters from natural image patches. Since then there has been a lot of effort in improving the original SC model, including approaches using alternatives to the originally suggested prior distributions. While by far most work kept focusing on continuous priors, discrete priors in the form of Bernoulli priors for binary latents have been investigated previously (Haft et al., 2004, Henniges et al. 2010, and also compare non-parametric Bayesian approaches such as Griffiths and Ghahramani, 2011). Furthermore, preliminary work for this study has investigated symmetric priors for three states (-1,0,1) (Exarchakis et al., 2012). We will use results of earlier approaches (Henniges et al., 2010; Exarchakis et al., 2012) and of their application to image patches as a further verification of the DSC algorithm before we proceed to the more general case for this data domain.

The data. The data set we used for DSC were a selection of images with no artificial structures taken from the van Hateren image data set (van Hateren and van der Schaaf, 1998). We randomly selected $N = 200\,000$ image patches of size 16×16, thus setting the dimensionality of the data to $D = 256$ dimensions. As preprocessing, we first

whitened the data using PCA-whitening and then we rotated the whitened data back to the original coordinate space using the set of highest principle components that corresponded 95% of the data variance, this technique is commonly referred to as zero-phase component whitening or ZCA (Bell and Sejnowski, 1997).

Algorithm details. As in most SC variants, we were concerned with introducing an algorithm that is overcomplete in the absolute number of dimensions. It is worth noting at this point that the dimensionality of the model is not invariant of the model structure so for different configurations the size of the hidden space should also change in order to achieve the same level of accuracy. However, it is clearer to expose this behavior if we use models with constrained dimensionality and we do that by fixing the number of hidden dimension for all tasks to $H = 300$. To maintain similar results across different configurations of the model we use the same training scheme. We ran the DSC algorithm for 200 iterations. To avoid local optima we again used deterministic annealing, as described in (Ueda and Nakano, 1998; Sahani, 1999), with an initial temperature for $T = 2$ that is decreased linearly to $T = 1$ between iterations 10 and 80. Furthermore, in order not to reject any datapoints early in training we used the full data set for the first 20 iterations and linearly decreased it to the set of best explained datapoints $\mathcal{M}$ between iteration 20 and 60. In all cases, we used the same approximation parameters $H' = 8$ and $\gamma = 5$ to maintain a comparable effect of the approximation on the results.

Discrete Sparse Coding with binary latents

The binary configuration of DSC (bDSC), with $\Phi = \{0, 1\}$, which recovers the Binary Sparse Coding (BSC) algorithm shows emergence of Gabor-like receptive fields as ex-

pected from (Henniges et al., 2010). The achievements highlighted in (Henniges et al., 2010) were primarily the high dimensional scaling of the latent space, inference of sparsity (a notable difference to Haft et al., 2004), and the recovery of image filters with statistics more familiar to those of primates (Ringach, 2002) than earlier algorithms (Olshausen and Field, 1997), even though later work reportedly improved on that further (Bornschein et al., 2013) (also compare Lücke, 2007, 2009; Rehn and Sommer, 2007; Zylberberg et al., 2011).

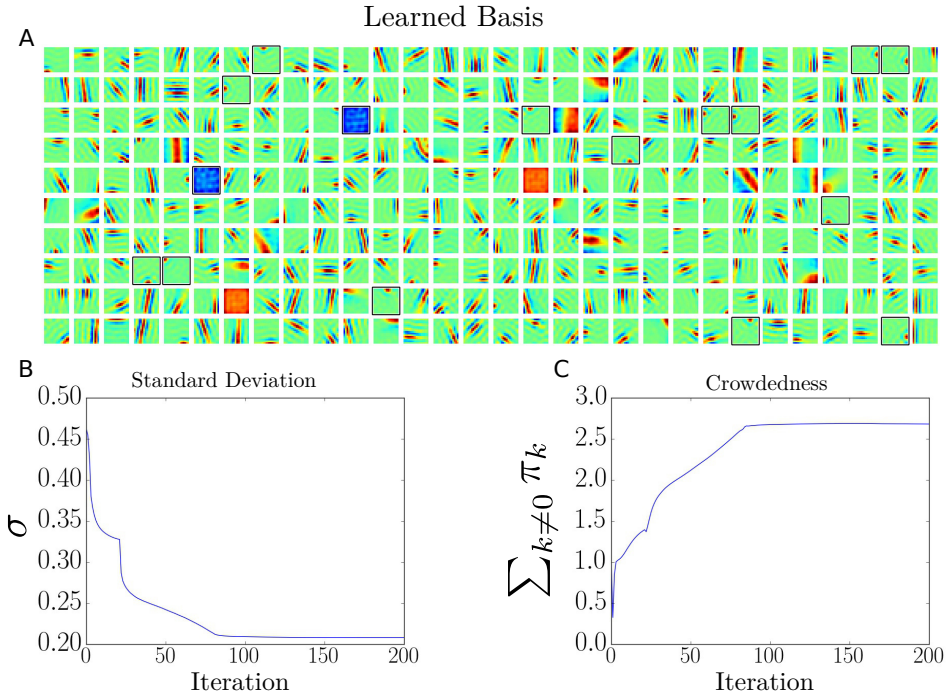

Figure 2: Results from training on natural images using the binary DSC model, $\Phi = \{0, 1\}$. **A** Learned dictionary elements. The framed dictionary elements are considered as center surround. **B** Model uncertainty parameter over EM iterations. **C** Average number of non zero coefficients "crowdedness" over EM iterations.

By applying bDSC we reproduce earlier results, i.e., we recover a dictionary with

Gabor-like and center surround filters, in a high dimensional latent space ($H = 300$), Figure 2 summarizes the obtained results. The work in (Henniges et al., 2010) showed results with an even higher number of observed and latent dimensions, however, the binary configuration of DSC has the same algorithmic complexity as BSC and it is trivial to show that DSC can scale to the same size. Here, we chose the lower dimensional observed and latent spaces to facilitate later comparison to the computationally more demanding DSC applications with more latent states. Also note that (Henniges et al., 2010) used a difference-of-Gaussian preprocessing instead of ZCA whitening chosen here and this may have an effect on the resulting parameters (compare Bornschein et al., 2013).

Discrete Sparse Coding with ternary latents

The next more complex DSC configuration we tried is the ternary case (tDSC) in which we use the configuration $\Phi = \{-1, 0, 1\}$. Unlike bDSC, tDSC is symmetric in the state space and therefore shares more features with popular SC algorithms which utilize symmetric priors. However, in this work we study symmetry only in terms of the states (i.e., $-1, 0, 1$) but allow different prior probabilities for each of these states (unlike Exarchakis et al., 2012, which assumed the same probabilities for states -1, and 1). The approximation parameters and training schedule were set to be identical to the bDSC in order to facilitate the comparison of the two configurations.

As the results in Figure 3 **C** show, tDSC converges to an almost symmetric prior, even with non-symmetric initializations of the prior probability of non zero states. For the DSC data model with configuration $\Phi = \{-1, 0, 1\}$ this means that any generative

field is similarly likely as its negative version.

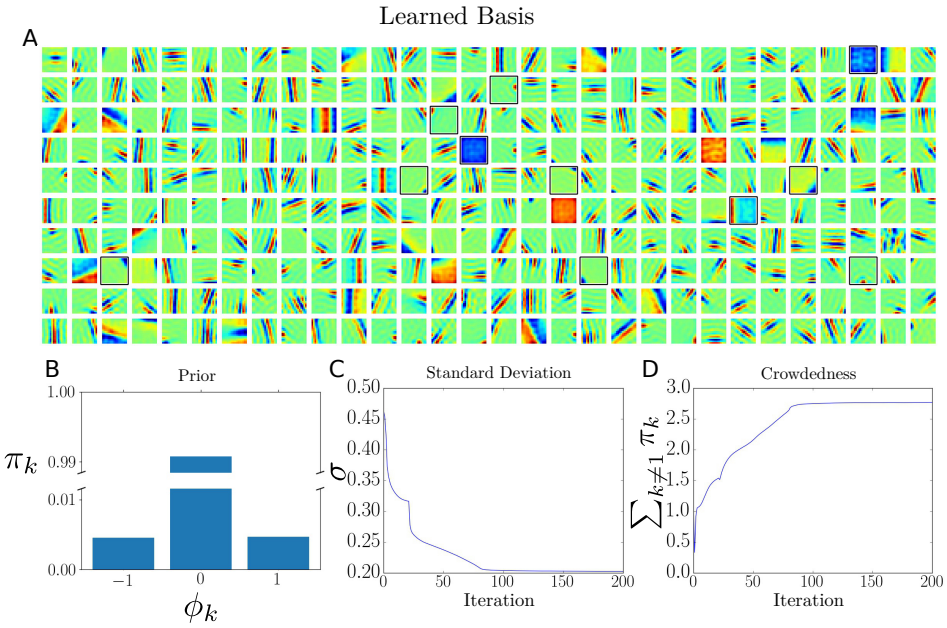

Figure 3: Results from training on natural images using the ternary DSC model ,$\Phi = \{-1, 0, 1\}$. **A** Learned dictionary elements. The framed dictionary elements are considered as center surround. **B** Learned prior parameters. **C** Model uncertainty parameter over EM iterations. **D** Average number of non zero coefficients "crowdedness" over EM iterations.

Discrete Sparse Coding with multiple positive latents

We now use a DSC configuration with a greater number of discrete states, and use $\Phi = \{0, 1, 2, 3, 4\}$ to investigate prior probability structures that are not elucidated by the bDSC and tDSC models. In particular, we are interested in identifying patterns in the slope of the prior as defined by a uniform discretization of their scale. Once more, the algorithmic details regarding this run can be viewed at the beginning of section 3.2 and they remain the same across all configurations. The only difference across the three

different tests is the configurations of the algorithms.

The learned prior at convergence is monotonically decreasing with the increasing values of the states suggesting that states of higher value have an auxiliary character. Furthermore, the shape of the prior distributions reinforces the argument for unimodal distributions. The increased number of states also shows a decreased scale for the noise model at convergence suggesting, once more, that an increased number of states provides a better fit for the data, compare to Figure 3 and Figure 2.

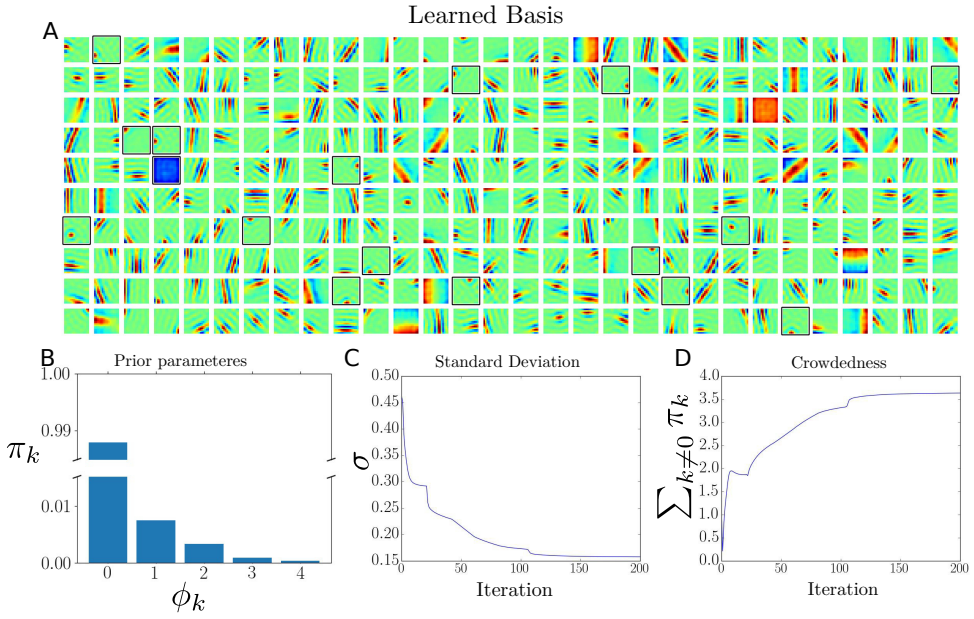

Figure 4: Results from training on natural images using the DSC model with a a configuration $\Phi = \{0, 1, 2, 3, 4\}$. **A** Learned dictionary elements. The framed dictionary elements are considered as center surround. **B** Prior parameters at convergence. **C** Model uncertainty parameter over EM iterations. **D** Average number of non zero coefficients "crowdedness" over EM iterations.

When compared with the learned dictionaries of the two earlier configurations, Fig-

ures 2 **A** and 3 **A**, the dictionary we learn for a higher number of positive discrete states, Figure 4 **A**, appears to are more localized, i.e. the dictionary elements resemble wavelets with a smaller support. This observation suggests that each of the learned generative fields is responsible for more fine detail structure in an image patch than a generative field by the binary and ternary configurations. Additionally, in Figure 4 we see that the sparsity decreases when compared with the ternary or binary configuration. Meaning that more generative fields are used on average to optimally, in likelihood terms, recreate an image. Which is to be expected since having each generative field being responsible for a smaller part of the image means that more generative fields would now be necessary on average to explain an image.

Discrete Sparse Coding with assymetric and non-uniform latents

For our last experiment, we used a configuration that allows for positive and negative values but the chosen prior values are not symmetrical around zero (unlike tDSC). Furthermore, we use a scaling of values that tiles the prior space with non uniform increments. The contrast response of pattern-sensitive neurons, e.g., in the visual system, has been observed to be non-linear (Heeger, 1992; Carandini and Heeger, 2012) with their range of responses being asymmetric with regard to the spontaneous response (Kandel et al., 1991). As a result, the distribution of contrast discrimination levels is non-uniform (Legge and Foley, 1980; Watson and Solomon, 1997). On the technical side, the observed non-linear and non-uniform contrast encodings have been used to provide technical improvements for the JPEG image encoding (Daly, 1990; Malo et al., 2000; Taubman and Marcellin, 2002; Malo et al., 2006). Here, we model assymetric

and non-uniform encoding using DSC with a configuration $\Phi = \{-1, 0, 2, 6\}$, which we will refer to as assymetric DSC (aDSC).

Again we maintain the same data set, same number of generative fields, and same training procedure as used for the experiments above. Figure 5 show the learned parameters after convergence. Again we observed many Gabor-like and some more globular generative fields (Fig. 5A). The learned probabilities of the prior values (Fig. 5B) are notably assigning more mass to positive values than to negative values. The prior is thus neither positive only nor symmetric around zero.

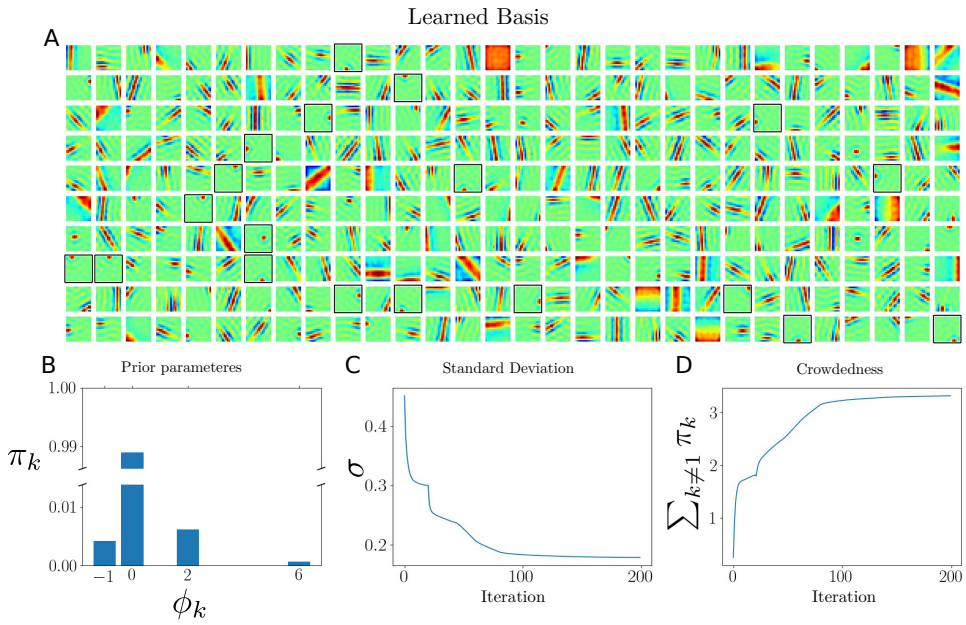

Figure 5: Results from training on natural images using the DSC model with a a configuration $\Phi = \{-1, 0, 2, 6\}$. **A** Learned dictionary elements. The framed dictionary elements are considered as center surround. **B** Prior parameters at convergence. **C** Model uncertainty parameter over EM iterations. **D** Average number of non zero coefficients "crowdedness" over EM iterations.

Quantitative Comparison of Generative Fields

Finally, let us quantitatively compare the properties of the image models learned using the different configuration of prior values above. Figure 6, shows comparisons of different measures of the learned model parameters. As any of the experiments on image patches (especially those for dDSC and aDSC) required extensive computational resource, Figure 6 does only take into account the experiments of Figures 2 to 5 and no repeated runs. However, repeated experiments for bDSC in this and previous studies (Henniges et al., 2010) show only small variations of individual runs compared to the differences between the four configurations as analyzed in Figure 6.

Figure 6**A** compares the variances inferred by the different configurations (in decreasing order). The variance decreases with increasing number different discrete values (size of Φ). If a generative field can be scaled by different non-zero factors it can be better reconstruct any given data point. Crowdedness (inverse sparsity) increases with increasing numbers of values (Figure 6**B**). In other words, more fields are used on average if many factors are available of any generative field. One explanation is that it becomes easier, e.g., for dDSC to add fields with small factors for better reconstruction. Furthermore, we observed a tendency to more localized generative fields for aDSC and dDSC, such that more fields are required to reconstruct extended structures.

To analyze the learned generative fields more closely, we matched them using Gabor functions (compare Ringach, 2002) and difference of Gaussians (DoG) functions. Based on the matching error, we then determined the number of DoG fields (Bornschein et al., 2013; Dai et al., 2013), which well correspond to the 'globular' fields described by Ringach (2002). We used the generative fields directly as we are here primarily in-

terested in the comparison between the models, and as the classification into Gabor-like and globular fields has not been observed to change when estimated receptive fields instead of generative fields were used (compare Bornschein et al., 2013; Dai et al., 2013). Figure 6**C** shows strong differences among the different configurations and unlike Figure 6**A,B** no monotonous dependency on the number of configuration values can be observed. Notably, the asymmetric prior (aDSC), which can be motivated by neurophysiological experiments shows the highest percentage of globular fields. High percentages of globular fields are, on the other hand, often observed *in vivo* (Ringach, 2002; Usrey et al., 2003; Niell and Stryker, 2008). The observed percentages are lower than, e.g., those measured by Ringach (2002) but we here investigate a relatively small number of fields (compare Bornschein et al., 2013, who observe increased percentages with overcompleteness). Also note that non-linear superposition properties of data components are likely to play a role for visual data (Bornschein et al., 2013). Finally, we measured the localization of generative fields (Figure 6**D**). We used the matched Gabor functions or DoG functions and computed the approximate area of the patch they covered[1]. The aDSC configuration does notably result in the most localized generative fields. Still it is using on average fewer components than dDSC for reconstruction. The tDSC configuration results in the least localized generative fields.

[1]We first used the classification into Gabor-like of DoG-like fields. Then we used $(2\sigma_x\, 2\sigma_y)$ as the area covered by a Gabor field with σ_x and σ_y parameterizing the Gaussian envelop of the matched Gabor. For a DoG field we used $(2\sigma_{\mathrm{out}})^2$ with σ_{out} being the larger of standard deviations of the two Gaussians of the DoG function.

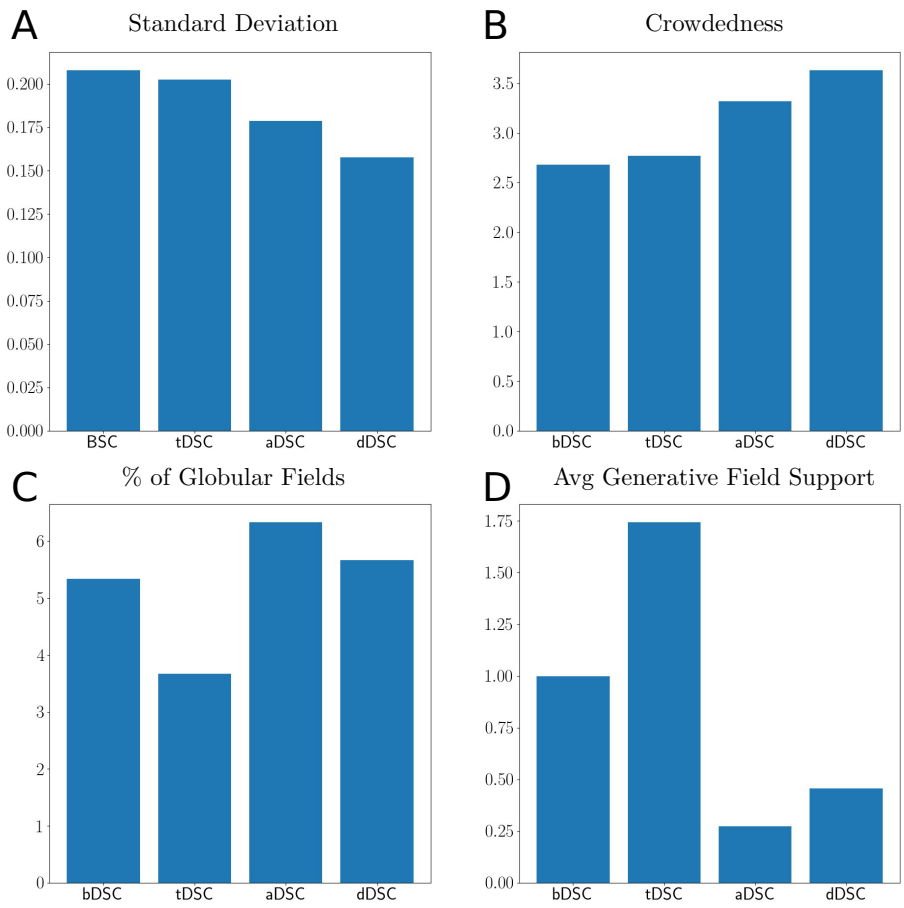

Figure 6: Quantitative analysis and comparison of components extracted from image patches. **A** Estimated noise σ using discrete sparse coding with different configurations (ordered in descending order). bDSC shows the highest level of noise followed by tDSC and aDSC. dDSC shows the lowest noise level, i.e., the best average reconstruction. **B** Crowdedness (inverse sparsity) in terms of the average number of components used to reconstruct an image patch. bDSC uses the least components followed by tDSC and aDSC while dDSC uses most components. **C** Fraction of globular fields. tDSC shows the lowest number of globular fields and aDSC the highest number. **D** Localization of fields (surface size of generative field support - relative to bDSC). The tDSC fields are localized the least while aDSC shows the highest average localization.

3.3 Analysis of Neuronal Recordings

Information in the brain is widely considered to be processed in the form of rapid changes of membrane potential across neurons, commonly know as action potentials or spikes. This activity is often viewed as a natural form of discretization of continuous sensory stimuli for later processing in the cortex.

A cost effective way to study the behavior of these neurons and the spike generating process is to perform extra-cellular (EC) electrode recordings. However, when one observes the data obtained from an extra-cellular recording one sees various forms of noise either structured, for instance spikes from remote neurons, or unstructured, such as sensor noise. In this setting, we expect the DSC algorithm to provide interesting insights on the analysis of neural data. Using, different configurations one can either explain overlapping spikes, or as we attempt to show here, use the discrete scaling inherent to the algorithm to explain background spikes, i.e. spikes of remote neurons, and high scaling to explain relevant/near spikes, or the amplitude decay of spike trains using multiple high values.

In this work we will present a study of neural data using the DSC algorithm. To be concise, we will focus on a single configuration of DSC that we believe best elucidates most of the features of the algorithm.

Dataset. We used the dataset described in (Henze et al., 2000, 2009). The dataset contains simultaneous intra-cellular and extra-cellular recordings from hippocampus region CA1 of anesthetized rats. We took the first EC channel of recording d533101, sampled at 10 kHz, and band-pass filtered it in the range of $400 - 4000$ Hz and then

we sequentially extracted $2ms$ patches of the filtered signal with an overlap of 50%. We used those patches as the training datapoints for our algorithm. We also use the intra-cellular (IC) recording provided by the dataset to better illustrate the properties of the uncertainty involved in EC recordings.

Training. We used a DSC configuration with 4 discrete states, $\Phi = \{0, 1, 6, 8\}$, to describe the structure of the data. This configuration was selected using the intuition that spikes of distant neurons will have roughly the same shape as spikes of the relevant neuron but at a smaller scale and therefore correspond to state 1 and the states 6, and 8 will explain features of the relevant neuron or nearby neurons for which we allow some variation in strength. Note that a configuration of $\Phi = \{0, 0.5, 3, 4\}$ would be equivalent because of the unnormalized columns of W. To choose the best model configuration we could use the variance at convergence as a selection criterion, however, it is useful to make assumptions on a configuration by observing the data. The number of hidden variables, $H = 40$, was selected to be slightly higher than the number of observed variables, $D = 20$, which in turn correspond to $2ms$ of recording sampled at 10 kHz. The approximation parameters for the ET algorithm were set to $H' = 6$ and $\gamma = 4$.

We initialize the noise scale σ as the mean standard deviation of the observed variables, the columns of W using the mean of the datapoints plus a Gaussian noise with standard deviation $\sigma/4$, and the parameters $\vec{\pi}$ we initialized such that $p(s_h = 0) = (H - 1)/H$ and $p(s_h \neq 0)$ is sampled from a uniformly random distribution under the constraint that $\sum_h p(s_h) = 1$.

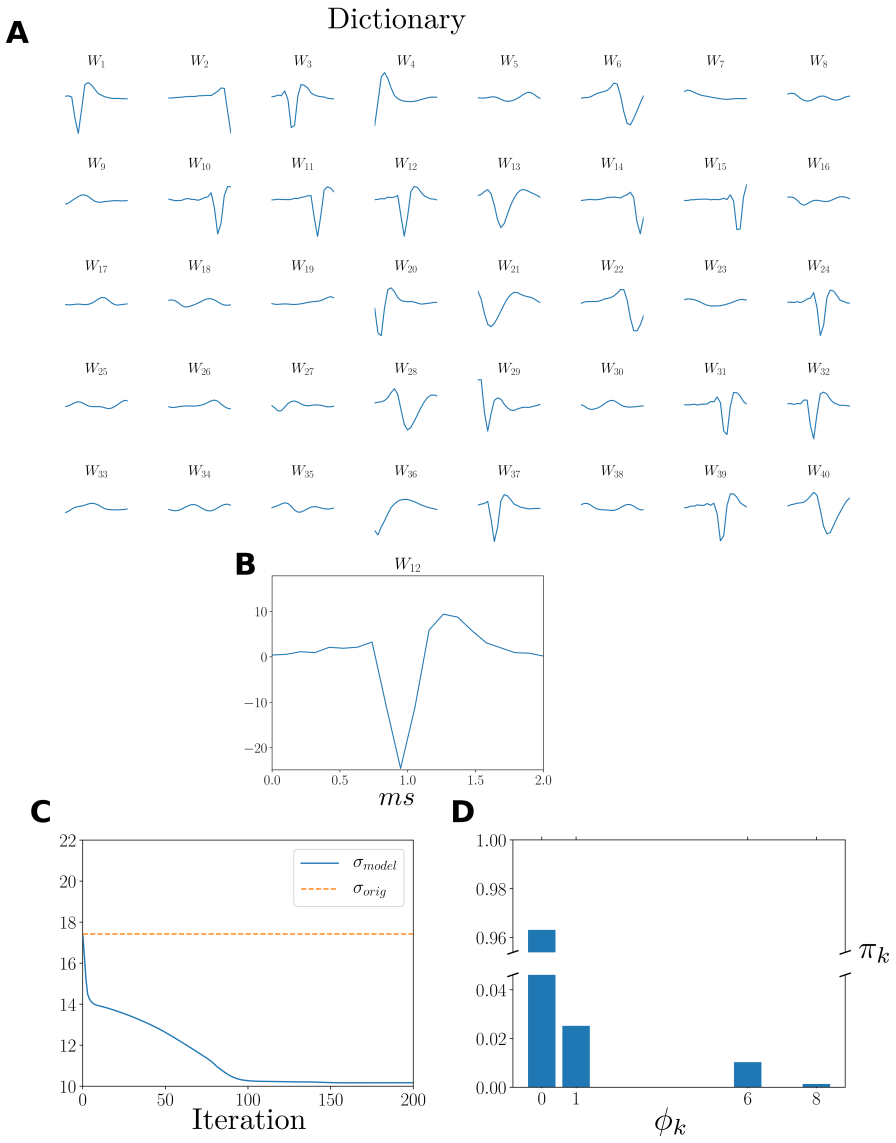

Figure 7: **A** The learned dictionary. Some of the basis develop as extra-cellular recordings of spikes similar to those seen in earlier literature. We also discover components that can only be attributed to structured noise, e.g. from distant neurons. **B** $\vec{W}_{12}$, On the x-axis you see the duration in ms and y-axis the voltage in mV. **C** The evolution of the model σ next to the original data std (dashed). **D** The learned prior parameters $\vec{\pi}$

We let the algorithm run for 200 EM iterations using a deterministic annealing schedule (Ueda and Nakano, 1998; Sahani, 1999) with $\beta = 2$ for the first 10 itera-

tions and proceed to linearly decreasing it to $\beta = 1$ by iteration 80. Furthermore, in order to avoid early rejection of interesting datapoints we force the algorithm to learn on all datapoints for the first 60 iterations and then decrease the number of datapoints to $|\mathcal{M}|$ by iteration 100, always maintaining the datapoints with the highest value for $\sum_{\vec{s}\in\mathcal{K}^{(n)}} p(\vec{y}_n, \vec{s})$, see Section 2.

In Figure 7 **A**, we see the dictionary as it was formed at convergence. There we notice potential shifts similar to the ones reported in (Henze et al., 2000) for the extracellular recordings but also other elements that are more similar to finer details of potential changes. Such a decomposition of activity into district subspaces has been shown to improve classification in many tasks and it could prove useful in identifying spiking activity of different neurons in spike sorting systems. One should also take into account that since there is no built-in temporal invariance in the model and there is no spike alignment previously performed in the data, we sometimes observe similar features to appear shifted across the time axis. These temporal shifts emerge as the DSC algorithm addresses temporal alignment by populating the dictionary with time shifted elements. Figure 7 **C** shows the evolution of the model noise scale σ compared to the total standard deviation of the original signal. As expected, once our model accounts for the spikes, of near or distant neurons, the noise in the signal becomes smaller. It is worth noting at this point the correlation of σ_{orig} with the presence of spikes in the signal. Figure 7 **D** shows the learned prior. The result suggests that most spikes are active with a coefficient of 1 suggesting that they belong to the background noise (modeling distant spike events received by the electrode), then the most active coefficient is 6 suggesting that the dictionary element describes firing scaled down, perhaps due to a spike burst,

and the lowest probability latent state is 8 which was intended to model spikes at their highest intensity.

To illustrate how well we were able to fit the data we reconstructed the extra-cellular signal using the latent variables values with highest (approximate) posterior probability), see Figure 9. More precisely, for each datapoint $\vec{y}_n$ we use the $\vec{s}^* \in \mathcal{K}^{(n)}$ that has the highest value for the truncated posterior $q_n(\vec{s})$ and we reconstruct the datapoint using the mean of the noise model $\hat{y}_n = W\vec{s}^*$. Since the datapoints were selected as consecutive patches of the original recording with a 50% overlap it was necessary to find a sensible way to appropriately reconstruct the overlapping region. We used the reconstruction contributed by the latent vector with the highest truncated posterior to determine the reconstruction at the overlapping region, i.e. $\hat{y}_n^+ = W^+\vec{s}^*$ with $\vec{s}^* = \mathrm{argmax}_{\vec{s}^*}\{q_n(\vec{s}^*), q_{n+1}(\vec{s}^*)\}$ where $\vec{y}_n^+$ is the last 50% of the vector $\vec{y}$ and W^+ the corresponding part of the rows of the W matrix. In Figure 9 **A**, we present the reconstruction (red line) of the original extra-cellular signal (blue line). The decomposition of the reconstruction in terms of generative fields (corresponding to the inferred states $\vec{s}^*$ is visualized in Figure 9 **B**, **C**, **D**.

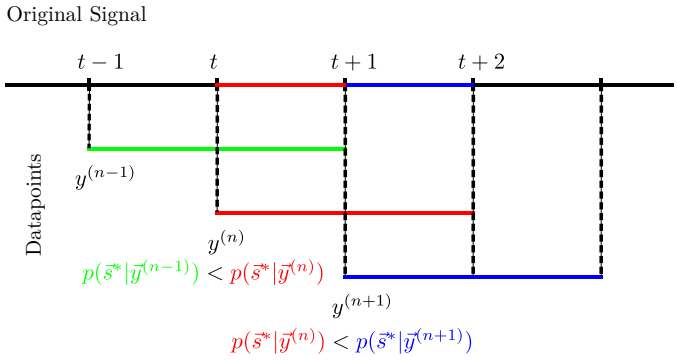

Figure 8: Graphical representation of our treatment of a time series signal. We separate the time series in segments and each datapoint for DSC is a patch of the time series that starts at the first observation of each segment and covers two consecutive segments. To reconstruct a datapoint we use $W\bar{s}^*$, where $\bar{s}^*$ is the MAP vector for each datapoint. To reconstruct a segment that appears in more than one datapoint we use the reconstructed values of the datapoint with the highest approximate posterior.

In Figure 10, we show the difference between the reconstructed time series and the original. From the result, we observe that the model does very well at explaining background activity, however, on the locations of some action potentials it appears that there is still relatively high uncertainty. Potentially, the relative frequency of spikes to background is very low and therefore making the spikes a rare event and not captured very well by the model. One could improve on that by either creating higher overlap between consecutive patches allowing the dictionary to explain more details on the potential axis rather than the time axis or to use some spike sorting preprocessing routine, such as spike detection (Quiroga et al., 2004). In Figure 10 **C**, we see the corresponding IC recording. Note that only two spikes belong to the targeted neuron.

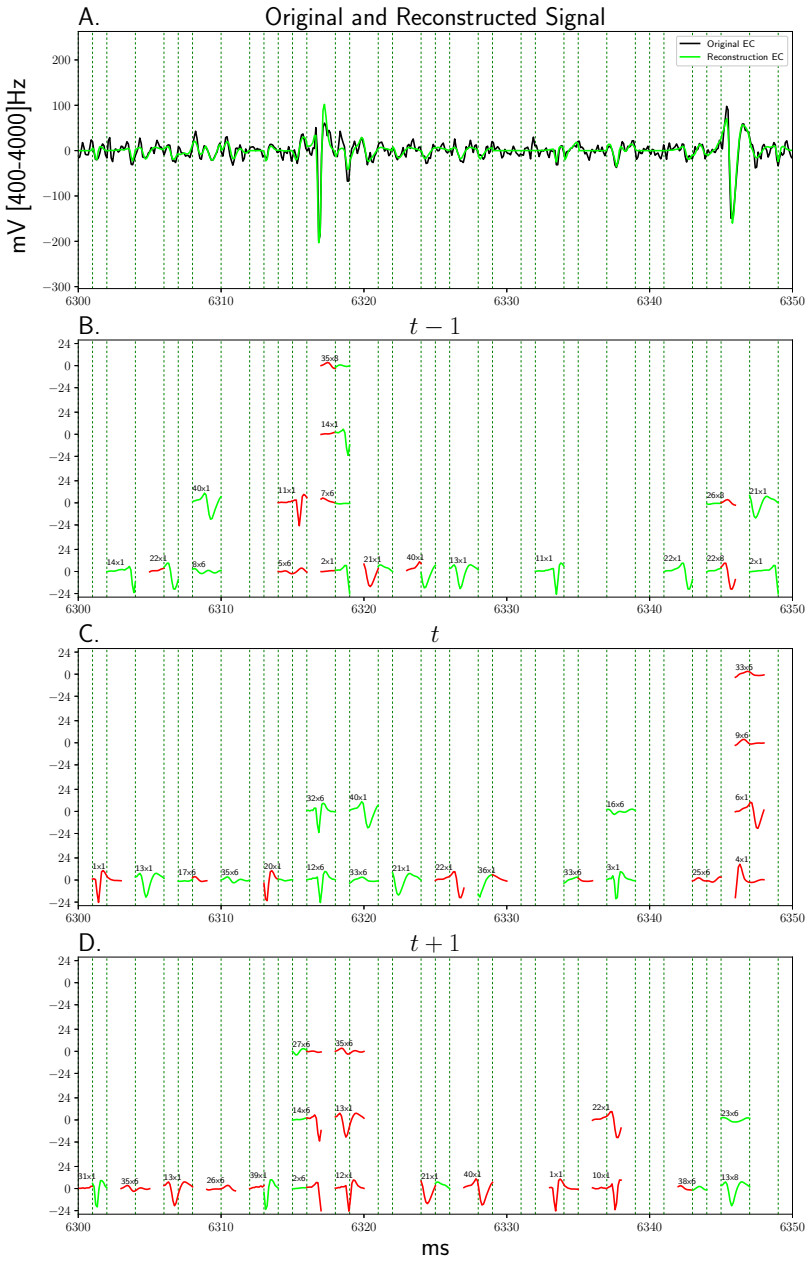

Figure 9: **A** Reconstruction results of an EC recording. **B**-**D** Dictionary elements used to reconstruct the signal. The time-axis are aligned - the plots **B**-**D** represent three consecutive datapoints with 50% overlap. The text above each line denotes the element id times the scaling factor. The green(red) segments of the elements were used(cut out) to reconstruct the corresponding part of the time series.

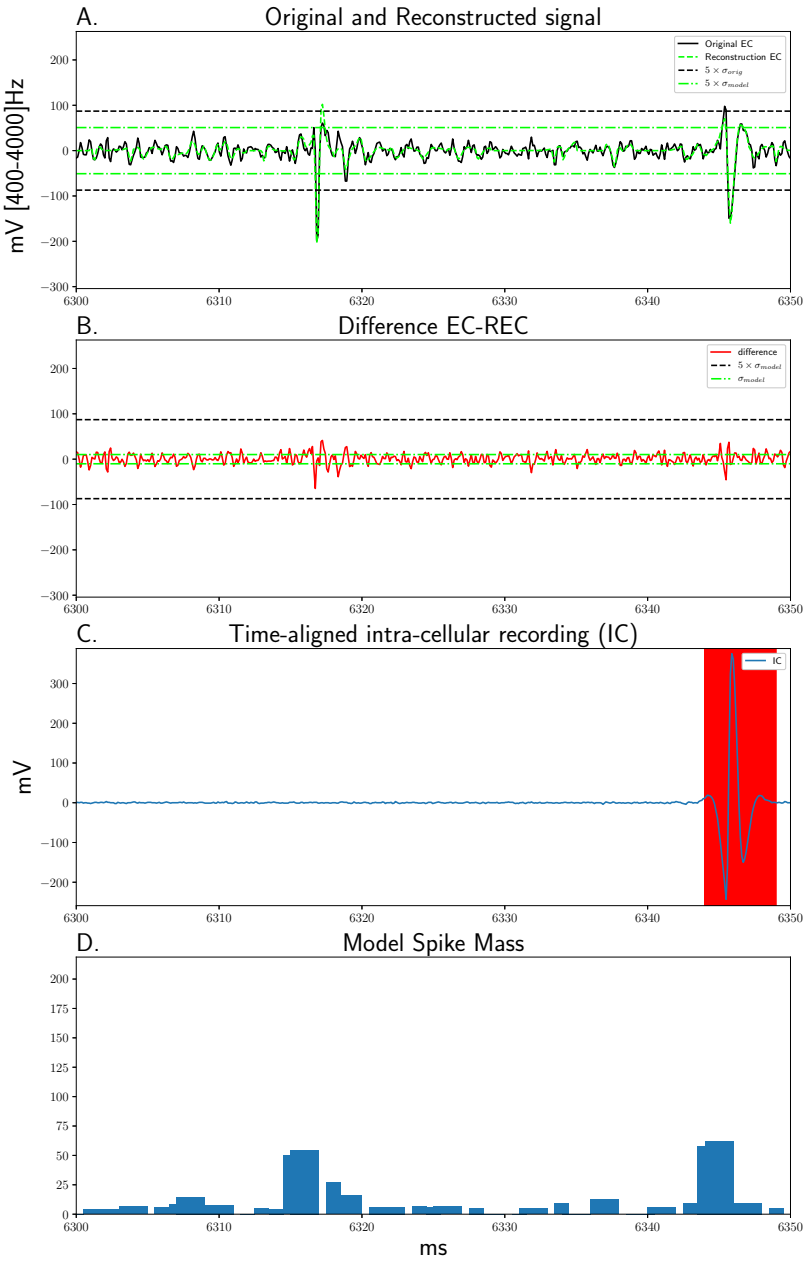

Figure 10: **A** Reconstruction results of an EC recording. **B** The Difference between the reconstructed and the original signal. **C** Time-aligned IC recording - only two of the three clear spikes in **A** correspond to a spike from the targeted neuron. **D** The energy contained in each reconstructed segment estimated using $\sum_h s_h \frac{1}{T} \sum_d |W_{dh}|$.

Part of a spike sorting task is spike identification. Spike identification is usually

performed using a threshold over the signal that is taken in a rather ad-hoc way to be at 5 times the standard deviation of the signal, e.g. see $5 \times \sigma_{orig}$ in Figure 10 **A** (Quiroga et al., 2004). In Figure 10 **D**, we propose an alternative based on the DSC model. The barplot shows the sum of the l_1 norm of all active dictionary elements at that point scaled by the corresponding latent, i.e. $\sum_h s_h \frac{1}{T} \sum_d |W_{dh}|$ for $s_h \in \vec{s}^*$ where $\vec{s}^*$ is the highest posterior latent vector. We expect this quantity to be a better spike detection measure since it is invariant of noise in the signal. For instance, if the neuron was spiking more frequently the threshold $5 \times \sigma_{orig}$ we see in Figure 10 **A** would increase but $5 \times \sigma_{model}$ would remain the same because the spikes would be explained by the latent $\vec{s}^*$. That means the threshold, $5 \times \sigma_{orig}$, applied in the signal varies with the neuron firing rate but any threshold imposed on Figure 10 **D** would only be affected very mildly by variations in the neuron firing rate.

3.4 Audio Data

For our final experimental setting we applied the algorithm to audio data of human speech. We used the TIMIT database (Garofolo et al., 1993) to extract $N = 100\,000$ datapoints $\vec{y}^{(n)}$ in the form of $D = 60$-dimensional consecutive waveforms with an overlap of 50%. We used $H = 100$ hidden variables to describe the data under the DSC generative model with a configuration $\Phi = \{-2, -1, 0, 1, 2\}$.

For the training, we initialize each column of the dictionary matrix $W \in \mathbb{R}^{D \times H}$ using a non silent datapoint, defined as a datapoint with a norm greater than 1, i.e. $\|\vec{y}^{(n)}\|_1 > 1$. The prior parameter $\vec{\pi}$ was initialized such that $p\left(s_h = 0\right) = \left(H - 1\right)/H$ and the probabilities of the non zero states were randomly drawn from a uniform dis-

tribution under the constraint that $\sum_h p(s_h) = 1$. The scale of the noise model σ was initialized as the average standard deviation of each observed variable.

We ran the DSC algorithm for 200 iteration and verified convergence by a stability check of the parameters over EM iterations. During the run we used an annealing schedule described in (Ueda and Nakano, 1998) with the annealing parameter starting at $\beta = 10$ and decaying it linearly to 1 by iteration 50. We also avoided datapoint cutting until iteration 20 and then proceeded to linearly decrease the datapoints to $|M|$ by iteration 80 as per the algorithm description in section 2.

After convergence we noticed that the learned dictionary, see Figure 11, is composed of both temporally localized components and global components. The dictionary components are frequently constrained to a single frequency although frequency mixing is not unlikely. The prior emerges to be symmetric around zero even thought no such constraint was imposed by the model and we also see a considerable decrease in the scale of the noise model which suggests a good fit.

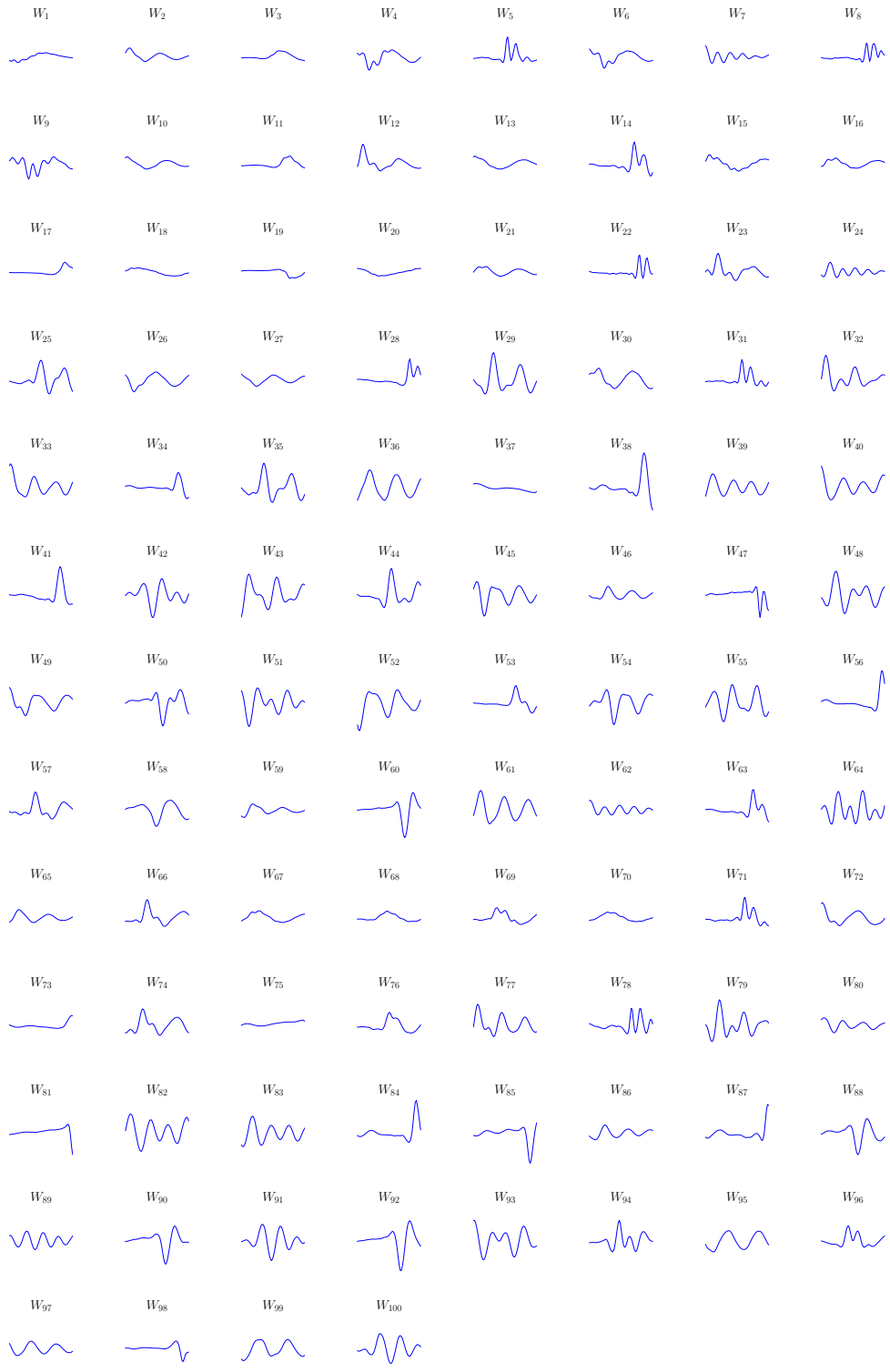

Figure 11: Columns of dictionary matrix, W, after convergence of the algorithm.

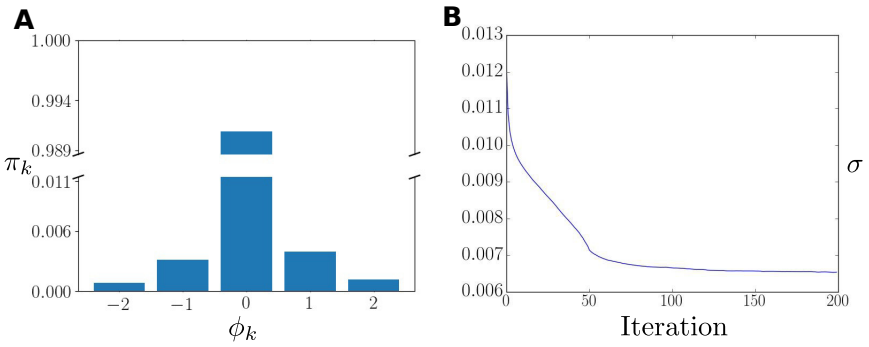

Figure 12: **A** prior parameters at convergence. **B** the evolution of the standard deviation of the model during ET algorithm iterations.

Similarly to the neural data analysis section 3.3, we used the reconstruction of a time series segment to evaluate how well we were able to fit the data. Once more, for each datapoint $\vec{y}_n$ we use the $\vec{s}^* \in \mathcal{K}^{(n)}$ that has the highest value for the truncated posterior $q_n(\vec{s})$ and we reconstruct the datapoint using the mean of the noise model $\hat{y}_n = W\vec{s}^*$. For the overlapping region we, again, use the reconstruction of the data point with the highest truncated posterior for $\vec{s}^*$. In Figure 13 **A**, we can see the reconstruction (red line) of the original waveform (blue line). The decomposition of the reconstruction can be seen in the following to subplots **B**, **C**, **D** over 3 consecutive datapoints $n-1, n, n+1$ respectively. The vertical lines are aligned in time across the four subplots and they represent the time limits for the reconstructed patches

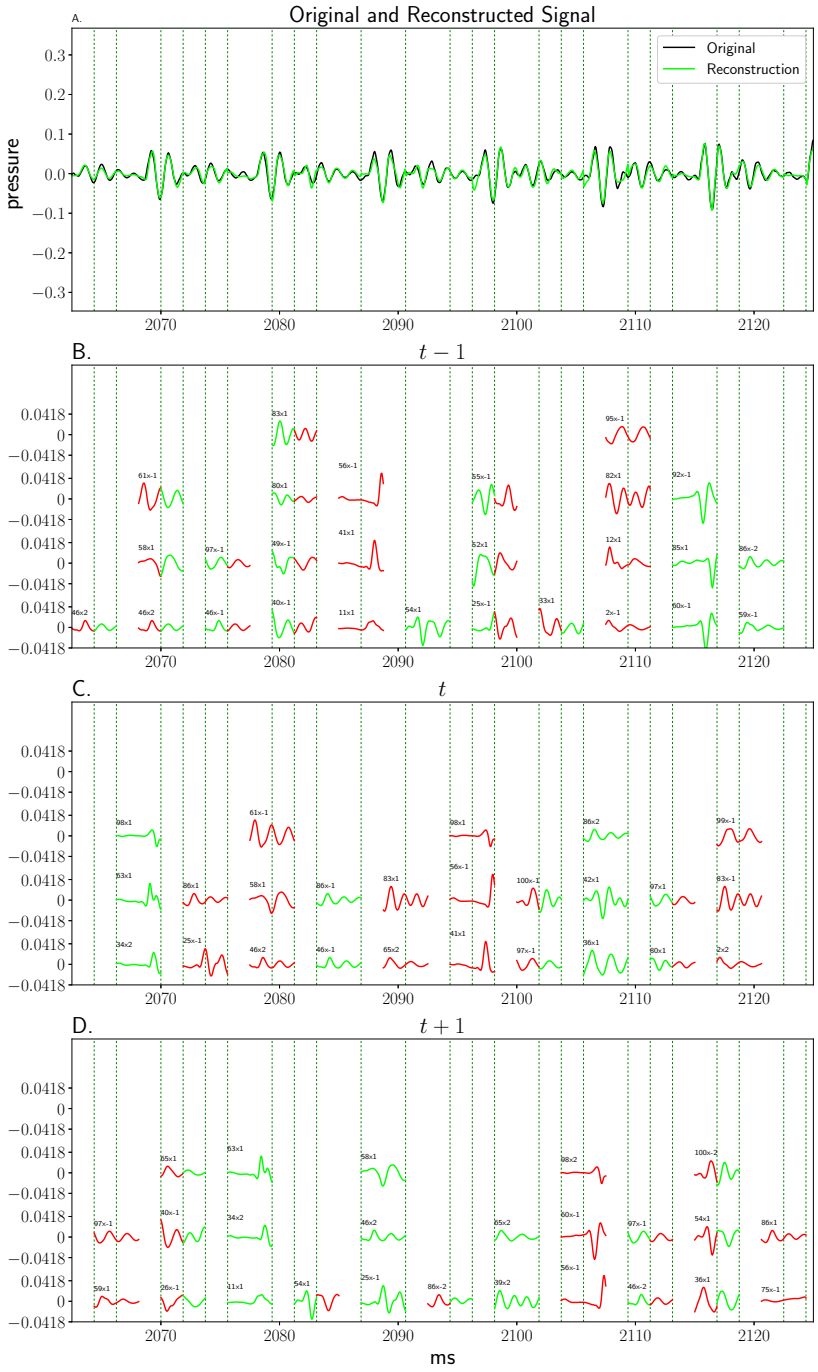

Figure 13: **A** Reconstruction results of an audio waveform. **B**-**D** Dictionary elements used to reconstruct the signal. The time-axis are aligned - the plots **B**-**D** represent three consecutive datapoints. The text above each line denotes the element id times the scaling factor. The green(red) segments of the elements were used(rejected) to reconstruct the corresponding part of the time series.

The red lines, in Figures 13 **B-D**, represent the reconstruction of the component with the highest truncated posterior from two consecutive datapoints - used to reconstruct the datapoint. The blue lines represent the component with the lower truncated posterior - rejected for the reconstruction.

4 Discussion

We have proposed a novel sparse coding algorithm with discrete latent variables, and we have shown that we are capable of efficiently learning model parameters for generative fields, noise and of the model's discrete prior distribution. Efficient learning was realized by adapting truncated approximations (ET; Lücke and Eggert, 2010) to work on latent spaces of multiple discrete states.

In this section we will discuss the interpretation of discrete latents in the Discrete Sparse Coding setting, the significance of varying discrete state spaces in image modeling, the properties that make a Discrete Sparse Coding algorithm relevant to spike analysis of neural data, and the efficiency of discrete sparse coding in fitting audio waveform.

Discrete latent variables for Sparse Coding. Sparse Coding algorithms were originally proposed as a method that deviates from traditional Gaussian encoding schemes to make encoding more selective to an axis and therefore implicitly forcing the features to be more descriptive of a given data structure. Constraining the hidden space to binary values provides an on/off encoding scheme that is selective to image structure aligned with standard Sparse Coding. Discretizing in an arbitrary domain, however, utilizes the

sparse coding principle to learn structure in the scale space of the data that would otherwise have been neglected or averaged out, for instance if we used continuous scaling values. The work presented in this paper shows that it is possible to efficiently learn a high dimensional discrete sparse coding model. Furthermore, we have shown that it is possible to learn a wider range of parameters than typical sparse coding algorithms such as the scale of the noise model and, more importantly here, parameters of a flexible prior.

Image Encoding. We have shown that our algorithm was able to scale to several real world high dimensional tasks. For image encoding, we have verified the functionality of our algorithm by first replicating previous (Henniges et al., 2010) and preliminary results (Exarchakis et al., 2012) using specific configurations of the DSC model. Furthermore, we have shown that scaling invariance in image encoding allows the filters to specialize in image structure rather than pixel intensity. Learning the prior parameters in different configurations of DSC has also shown us the distribution of the learned dictionary in scale space without constraints on the functional form like the ones commonly imposed by sparse coding algorithms with continuous latents (e.g., Laplace distribution/l_1 sparsity penalty). Identifying the appropriate shape for the prior of the SC latents has been a persistent research area in natural image statistics research (Olshausen and Millman, 2000; Hyvärinen et al., 2005; Berkes et al., 2008; Mohamed et al., 2010; Goodfellow et al., 2013; Sheikh et al., 2014). Since the prior of DSC does not define a density function and it can take arbitrary discrete values, we can use it to try and sketch the necessary qualities of a natural image prior.

The asymmetric prior of aDSC with unequally spaced values was explicitly motivated by neurophysiological data (Watson and Solomon, 1997; Legge and Foley, 1980). Asymmetric continuous priors pose considerable challenges for standard training schemes of sparse coding, and they have (to the knowledge of the authors) not been investigated for image patches. Notably, such a discrete prior (with learned probabilities as shown in Figure 5), shows the highest localization of generative fields which suggests that it can capture detailed image properties. Furthermore, it shows the highest number of 'globular' fields which were found to frequently occur in physiological data (Ringach, 2002). Our results may therefore be taken as motivating more research on asymmetric priors, with increasingly large-scale applications (larger number of fields, more discrete values, larger patch-sizes), in depth statistical analysis of fields and detailed comparison with a variety of physiological measurements. The configuration of dDSC uses five non-negative latent values and results in the best value for image patch reconstruction (Figure 6**A**). Notably, the learned probabilities support a heavy-tail distribution similarly to those used for standard sparse coding. The symmetric configuration of tDSC also results in an (approximately) symmetric distribution of prior mass for negative and positive values. Taken together, tDSC and dDSC may therefore be taken as support, e.g., for standard Laplace priors. Compared to aDSC, the results for tDSC and bDSC show less localized generative fields, and lower amounts of globular fields. Functionally, a future research direction may be the combination of DSC with model selection approaches in order to estimate their match to image patch data in a grounded way. This may also provide more evidence for encoding strategies observed in physiological data.

In general, it is important to note, however, that convergence to local optima has to

be considered for all sparse coding approaches. Also for DSC we have observed local optima including for artificial data, making it difficult to guarantee the optimality of the learned shapes. Also note, that the shapes emerge given the data distributions modeled by the DSC data model. While being general for discrete data, they do use standard assumptions of linear superposition and Gaussian noise model. These assumptions are shared with the large majority of sparse coding approaches but alternative models have been suggested in the past (Malo et al., 2006; Lücke and Sahani, 2008; Bornschein et al., 2013; Henniges et al., 2014; Frolov et al., 2016).

Discrete Latent variables for Neural Data Analysis. We used neural data to evaluate the performance of our algorithm due to their popular interpretation as sequences of discrete events. In this analysis, we showed that DSC can learn spike and sub-spike features that sufficiently describe neural recordings. Furthermore, carefully selecting the scale space makes it possible to discern physiological characteristics of temporal alignment, for instance whether a given spike is the initial event or a secondary spike in a spike burst. Notably, one of the most unique features of our algorithm, learning prior parameters, was very informative about the structure of extra-cellular (EC) recordings. The learned prior interprets EC recordings as being composed of a multitude of spiking patterns coming from a population of neurons around the targeted neuron. Furthermore, the fact that we can learn a Gaussian noise model distinct from the spiking activity provides a more clear separation of noise from spikes than those traditionally seen in spike sorting tasks (Quiroga et al., 2004).

Discrete Latent variables for Audio Data. The DSC algorithm was fitted to audio data successfully. The reconstruction has shown intelligible speech even though we did not use any hand crafted features of human speech suggesting that we were able to learn elementary short-time speech primitives. We did not extend our study beyond learning primitives and studying elementary reconstruction. Our results may motivate future research on audio compression using estimated discrete value, however. Any learned probabilities and unequally spaced discretization values may be of interest for efficient speech encoding, for instance. Speech reconstruction would than use averages of overlapping reconstructions as is usual, e.g., for inpainting methods. Such and other post-processing methods would warrant a smooth speech reconstruction but a detailed application would go beyond the purpose of this study.

Conclusion. To conclude we have derived, implemented and tested a novel sparse coding algorithm. Whenever it is reasonable to assume that the hidden variables are discrete, the studied approach offers itself to learn a statistical data model. We have shown applicability to data with discrete causes, and we have shown how the learned prior shapes can also be used as discretized versions for latents with presumably rather continuous latents. No other sparse coding model for discrete latents (other than binary) has previously been studied. Furthermore, our model covers the general class of (finite) discrete priors under the canonical sparse coding assumptions of iid and sparsely distributed latents. Discrete Sparse Coding can be used in latent variable tasks where the prior distribution is expected to be sparse and discrete and in latent variable task in which the structure of a sparse prior is not known. In the first case, the Discrete

Sparse Coding algorithm in its various configurations can fit any discrete tiling of the prior space that a given data set may necessitate. In the second case, the here studied approach can provide a proxy prior distribution for continuous latents while it does not impose constraints on the shape of the prior distribution.

Acknowledgment. We acknowledge funding by the DFG Cluster of Excellence EXC 1077/1 (Hearing4all) and by DFG grant LU 1196/5-1.

References

Aharon, M., Elad, M., and Bruckstein, A. (2006). K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. Signal Processing, IEEE Transactions on, 54(11):4311–22.

Bell, A. J. and Sejnowski, T. J. (1997). The "independent components" of natural scenes are edge filters. Vision Research, 37(23):3327–38.

Berkes, P., Turner, R., and Sahani, M. (2008). On sparsity and overcompleteness in image models. Advances in Neural Information Processing Systems, 21.

Bingham, E. and Hyvärinen, A. (2000). A fast fixed-point algorithm for independent component analysis of complex valued signals. International journal of neural systems, 10(01):1–8.

Bingham, E., Kabán, A., and Fortelius, M. (2009). The aspect bernoulli model: multiple causes of presences and absences. Pattern Analysis and Applications, 12(1):55–78.

Bornschein, J., Henniges, M., and Lücke, J. (2013). Are V1 simple cells optimized for visual occlusions? A comparative study. PLoS Computational Biology, 9(6):e1003062.

Carandini, M. and Heeger, D. J. (2012). Normalization as a canonical neural computation. Nature Reviews Neuroscience, 13:51–62.

Dai, Z., Exarchakis, G., and Lücke, J. (2013). What are the invariant occlusive components of image patches? a probabilistic generative approach. In Advances in Neural Information Processing Systems, pages 243–251.

Daly, S. J. (1990). Application of a noise-adaptive contrast sensitivity function to image data compression. Optical Engineering, 29(8):977–987.

Donoho, D. L. (2006). Compressed sensing. IEEE Transactions on information theory, 52(4):1289–1306.

Eldar, Y. C. and Kutyniok, G. (2012). Compressed sensing: theory and applications. Cambridge University Press.

Exarchakis, G., Henniges, M., Eggert, J., and Lücke, J. (2012). Ternary sparse coding. In Proceedings LVA/ICA, LNCS. Springer. in press.

Field, D. J. (1994). What is the goal of sensory coding? Neural Comput., 6(4):559–601.

Frolov, A. A., Húsek, D., and Polyakov, P. Y. (2016). Comparison of seven methods for boolean factor analysis and their evaluation by information gain. IEEE transactions on neural networks and learning systems, 27(3):538–550.

Garofolo, J. S., Lamel, L. F., Fisher, W. M., Fiscus, J. G., Pallett, D. S., and Dahlgren, N. L. (1993). {DARPA} {TIMIT} Acoustic Phonetic Continuous Speech Corpus {CDROM}.

Goodfellow, I., Courville, A. C., and Bengio, Y. (2012). Large-scale feature learning with spike-and-slab sparse coding. In ICML.

Goodfellow, I. J., Courville, A., and Bengio, Y. (2013). Scaling up spike-and-slab models for unsupervised feature learning. IEEE transactions on pattern analysis and machine intelligence, 35(8):1902–1914.

Griffiths, T. L. and Ghahramani, Z. (2011). The indian buffet process: An introduction and review. Journal of Machine Learning Research, 12(Apr):1185–1224.

Haft, M., Hofman, R., and Tresp, V. (2004). Generative binary codes. Pattern Anal Appl, 6:269–84.

Hancock, P. J. B., Baddeley, R. J., and Smith, L. (1992). The principal components of natural images. Network: Computation in Neural Systems, 3:61 – 70.

Heeger, D. J. (1992). Normalization of cell responses in cat striate cortex. Visual neuroscience, 9(02):181–197.

Henniges, M., Puertas, G., Bornschein, J., Eggert, J., and Lücke, J. (2010). Binary sparse coding. In Proceedings LVA/ICA, LNCS 6365, pages 450–57. Springer.

Henniges, M., Turner, R. E., Sahani, M., Eggert, J., and Lücke, J. (2014). Efficient occlusive components analysis. Journal of Machine Learning Research, 15:2689–2722.

Henze, D. A., Borhegyi, Z., Csicsvari, J., Mamiya, A., Harris, K. D., and Buzsáki, G. (2000). Intracellular features predicted by extracellular recordings in the hippocampus in vivo. Journal of neurophysiology, 84(1):390–400.

Henze, D. A., Borhegyi, Z., Csicsvari, J., Mamiya, A., Harris, K. D., and Buzsáki, G. (2009). Simultaneous intracellular and extracellular recordings from hippocampus region ca1 of anesthetized rats. CRCNS.org.

Hoyer, P. O. (2002). Non-negative sparse coding. In Neural Networks for Signal Processing XII: Proceedings of the IEEE Workshop on Neural Networks for Signal Processing, pages 557–65.

Hubel, D. H. and Wiesel, T. N. (1977). Functional architecture of macaque visual cortex. Proceedings of the Royal Society of London B, 198:1 – 59.

Hyvärinen, A., Hoyer, P. O., Hurri, J., and Gutmann, M. (2005). Statistical models of images and early vision. In Proceedings of the Int. Symposium on Adaptive Knowledge Representation and Reasoning (AKRR2005).

Kandel, E. R., Schwartz, J. H., and Jessell, R. M. (1991). Principles of neural science. Prentice-Hall International Inc.

Lee, D. D. and Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. Nature, 401(6755):788–91.

Lee, H., Battle, A., Raina, R., and Ng, A. (2007). Efficient sparse coding algorithms. In Advances in Neural Information Processing Systems, volume 20, pages 801–08.

Legge, G. E. and Foley, J. M. (1980). Contrast masking in human vision. JOSA, 70(12):1458–1471.

Lücke, J. (2007). A dynamical model for receptive field self-organization in V1 cortical columns. In Proc. International Conference on Artificial Neural Networks, LNCS 4669, pages 389 – 398. Springer.

Lücke, J. (2009). Receptive field self-organization in a model of the fine-structure in V1 cortical columns. Neural Computation, 21(10):2805–45.

Lücke, J. and Eggert, J. (2010). Expectation truncation and the benefits of preselection in training generative models. Journal of Machine Learning Research, 11:2855–900.

Lücke, J. and Sahani, M. (2008). Maximal causes for non-linear component extraction. Journal of Machine Learning Research, 9:1227–67.

Mairal, J., Bach, F., Ponce, J., and Sapiro, G. (2010). Online learning for matrix factorization and sparse coding. Journal of Machine Learning Research, 11.

Malo, J., Epifanio, I., Navarro, R., and Simoncelli, E. P. (2006). Nonlinear image representation for efficient perceptual coding. IEEE Transactions on Image Processing, 15(1):68–80.

Malo, J., Ferri, F., Albert, J., Soret, J., and Artigas, J. (2000). The role of perceptual contrast non-linearities in image transform quantization. Image and Vision Computing, 18(3):233–246.

Mohamed, S., Heller, K., and Ghahramani, Z. (2010). Sparse exponential family latent variable models. NIPS Workshop.

Neal, R. and Hinton, G. (1998). A view of the EM algorithm that justifies incremental, sparse, and other variants. In Jordan, M. I., editor, Learning in Graphical Models. Kluwer.

Niell, C. M. and Stryker, M. P. (2008). Highly selective receptive fields in mouse visual cortex. The Journal of Neuroscience, 28(30):7520–7536.

Olshausen, B. and Field, D. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Nature, 381:607–9.

Olshausen, B. and Millman, K. (2000). Learning sparse codes with a mixture-of-Gaussians prior. Advances in Neural Information Processing Systems, 12:841–847.

Olshausen, B. A. and Field, D. J. (1997). Sparse coding with an overcomplete basis set: A strategy employed by V1? Vision Research, 37(23):3311–3325.

Puertas, G., Bornschein, J., and Lücke, J. (2010). The maximal causes of natural scenes are edge filters. In Advances in Neural Information Processing Systems, volume 23, pages 1939–47.

Quiroga, R. Q., Nadasdy, Z., and Ben-Shaul, Y. (2004). Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering. Neural Comput., 16(8):1661–1687.

Rehn, M. and Sommer, F. T. (2007). A network that uses few active neurones to code visual input predicts the diverse shapes of cortical receptive fields. Journal of Computational Neuroscience, 22(2):135–46.

Ringach, D. L. (2002). Spatial structure and symmetry of simple-cell receptive fields in macaque primary visual cortex. Journal of Neurophysiology, 88:455–63.

Sahani, M. (1999). Latent variable models for neural data analysis. PhD thesis, Caltech.

Sheikh, A.-S., Shelton, J. A., and Lücke, J. (2014). A truncated EM approach for spike-and-slab sparse coding. Journal of Machine Learning Research, 15:2653–2687.

Shelton, J. A., Bornschein, J., Sheikh, A.-S., Berkes, P., and Lücke, J. (2011). Select and sample - a model of efficient neural inference and learning. Advances in Neural Information Processing Systems, 24.

Sparrer, S. and Fischer, R. F. (2014). Adapting compressed sensing algorithms to discrete sparse signals. In Smart Antennas (WSA), 2014 18th International ITG Workshop on, pages 1–8. VDE.

Taubman, D. S. and Marcellin, M. W. (2002). JPEG 2000: Image Compression Fundamentals, Standards and Practice. Kluwer Academic Publishers, Norwell, MA, USA.

Titsias, M. K. and Lázaro-Gredilla, M. (2011). Spike and slab variational inference for multi-task and multiple kernel learning. In Advances in Neural Information Processing Systems, volume 24.

Titsias, M. K. and Lázaro-Gredilla, M. (2011). Spike and slab variational inference for multi-task and multiple kernel learning. In Shawe-Taylor, J., Zemel, R. S., Bartlett, P. L., Pereira, F., and Weinberger, K. Q., editors, Advances in Neural Information Processing Systems 24, pages 2339–2347. Curran Associates, Inc.

Ueda, N. and Nakano, R. (1998). Deterministic annealing EM algorithm. Neural Networks, 11(2):271–82.

Usrey, W. M., Sceniak, M. P., Chapman, B., Nowak, L. G., Sanchez-vives, M. V., Mccormick, D. A., Cass, J., Stuit, S., Bex, P., Alais, D., Usrey, W. M., Sceniak, M. P., and Chapman, B. (2003). Receptive fields and response properties of neurons in layer 4 of ferret visual cortex. Journal of Neurophysiology, 89:1003–1015.

van Hateren, J. H. and van der Schaaf, A. (1998). Independent component filters of natural images compared with simple cells in primary visual cortex. Proceedings of the Royal Society of London B, 265:359–66.

Watson, A. B. and Solomon, J. A. (1997). Model of visual contrast gain control and pattern masking. JOSA A, 14(9):2379–2391.

Zhou, M., Chen, H., Ren, L., Sapiro, G., Carin, L., and Paisley, J. W. (2009). Non-parametric bayesian dictionary learning for sparse image representations. In Advances in neural information processing systems, pages 2295–2303.

Zylberberg, J., Murphy, J. T., and Deweese, M. R. (2011). A Sparse Coding Model with Synaptically Local Plasticity and Spiking Neurons Can Account for the Diverse Shapes of V1 Simple Cell Receptive Fields. PLoS Computational Biology, 7(10):e1002250.