LETTER — **Communicated by Aapo Hyvarinen**

Learning and Inference in Sparse Coding Models With Langevin Dynamics

Michael Y.-S. Fang
michaelfang@berkeley.edu
Department of Physics, University of California, Berkeley, Berkeley, CA 94720, U.S.A., and Redwood Center for Theoretical Neuroscience, University of California, Berkeley, Berkeley, CA 94720, U.S.A.

Mayur Mudigonda
mudigonda@berkeley.edu
Redwood Center for Theoretical Neuroscience, University of California, Berkeley, Berkeley, CA 94720, U.S.A.

Ryan Zarcone
ryanzarcone@gmail.com
Redwood Center for Theoretical Neuroscience, University of California, Berkeley, Berkeley, CA 94720, U.S.A., and Biophysics Graduate Group, University of California, Berkeley, Berkeley, CA 94720, U.S.A.

Amir Khosrowshahi
khosra@gmail.com
Redwood Center for Theoretical Neuroscience, University of California, Berkeley, Berkeley, CA 94720, U.S.A., and Intel Corporation, Santa Clara, CA 95054, U.S.A.

Bruno A. Olshausen
baolshausen@berkeley.edu
Redwood Center for Theoretical Neuroscience, University of California, Berkeley, Berkeley, CA, U.S.A., and Helen Wills Neuroscience Institute and School of Optometry, University of California, Berkeley, Berkeley, CA 94720, U.S.A.

We describe a stochastic, dynamical system capable of inference and learning in a probabilistic latent variable model. The most challenging problem in such models—sampling the posterior distribution over latent variables—is proposed to be solved by harnessing natural sources of stochasticity inherent in electronic and neural systems. We demonstrate this idea for a sparse coding model by deriving a continuous-time equation for inferring its latent variables via Langevin dynamics. The model parameters are learned by simultaneously evolving according to another continuous-time equation, thus bypassing the need for digital accumulators or a global clock. Moreover, we show that Langevin dynamics lead to

Neural Computation **34, 1676–1700** (2022)
https://doi.org/10.1162/neco_a_01505

an efficient procedure for sampling from the posterior distribution in the L_0 sparse regime, where latent variables are encouraged to be set to zero as opposed to having a small L_1 norm. This allows the model to properly incorporate the notion of sparsity rather than having to resort to a relaxed version of sparsity to make optimization tractable. Simulations of the proposed dynamical system on both synthetic and natural image data sets demonstrate that the model is capable of probabilistically correct inference, enabling learning of the dictionary as well as parameters of the prior.

1 Introduction

Latent variable models such as sparse coding (Olshausen & Field, 1997) and Boltzmann machines (Hinton & Sejnowski, 1983; Ackley, Hinton, & Sejnowski, 1985) have been shown to be powerful and flexible tools in machine learning. However, training such models properly requires sampling from probability distributions over the latent variables. Typically, instead of sampling, a maximum a posteriori (MAP) estimate or other heuristics are used since most sampling algorithms are laboriously slow and have convergence guarantees only under limited conditions. The time cost in large part comes from simulating stochastic dynamics of state transitions on deterministic, discrete-logic-based hardware, requiring random number generation and fine sampling intervals to avoid discretization errors. These limitations have hindered the ability of latent variables models to learn complex structures in data, since adapting the parameters in a more complex, structured model, such as a hierarchical probabilistic model (Lee & Mumford, 2003), necessitates sampling under the posterior distribution.

This letter proposes a solution to this problem based on utilizing the intrinsic sources of stochasticity that exist in any physical system. Our central thesis is that rather than forcing a deterministic, discrete-logic-based system to simulate stochastic dynamics on continuous variables, a more sensible and efficient solution is to exploit physics to directly implement stochastic, analog computation. In the same way that the analog VLSI retina implements filtering via lateral inhibition in a resistive grid (Mead & Mahowald, 1988), resulting in orders of magnitude greater computational efficiency than digital simulation, we envision the development of analog circuits that perform the necessary computations and stochastic dynamics for probabilistic inference and learning in complex latent variable models. A recent successful example of this approach is the work of Borders et al. (2019), who used the intrinsic probabilistic behavior of nanoscale magnetic tunneling junctions to sample from the binary state variables of a Boltzmann machine. Another example is the use of stochastic logic circuits to perform fast Bayesian inference for perception and reasoning tasks (Mansinghka & Jonas, 2014; Mansinghka, Jonas, & Tenenbaum, 2008).

Additionally, in neuroscience, it has been hypothesized that seemingly random fluctuations in neural activity can be interpreted as a process for sampling from posterior distributions (Hoyer & Hyvärinen, 2003; Berkes, Orbán, Lengyel, & Fiser, 2011; Orbán, Berkes, Fiser, & Lengyel, 2016; Echeveste, Aitchison, Hennequin, & Lengyel, 2020). Our goal here is to demonstrate, through derivation and simulation of a dynamical system of equations, the viability of such an approach for probabilistic inference and learning in a latent variable model. In an online appendix, we point the way to a potential circuit implementation.

Beyond the difficulties associated with sampling, learning the parameters of a probabilistic model requires averaging the samples or other quantities computed from them. One direct way of doing this is to accumulate these quantities followed by a parameter update (see Figure 1b). However, this requires a digital accumulator, and the interfacing between analog and digital hardware is often a bottleneck for sampling. For example, in recent work by Roques-Carmes et al. (2019), the limiting component for a photonic sampler was identified as the photodetector. Here we propose a novel, fully analog framework in which the update of parameters occurs simultaneously alongside the sampling of latent variables through continuous time dynamics (see Figure 1c). Rather than waiting for the collection of samples for each discrete parameter update, the effective accumulation of samples is achieved by simply having a longer time constant.

To study this analog learning and inference framework, we apply it to the sparse coding model, a simple yet expressive probabilistic model with an explicit prior over the latent variables (Tibshirani, 1996; Hastie, Tibshirani, & Friedman, 2009). The sparse coding model is of interest in both neuroscience and engineering as it provides an account of the neural representation of natural images in visual cortex (Olshausen & Field, 1997), and it has proven useful in computer vision (Wright et al., 2010; Wang et al., 2015) and signal compression (Donoho, 2006). However current implementations of sparse coding are slow due to the optimization required to infer the latent variables for each data sample, and learning is inefficient since only a single such point estimate of the latent variables is used to make a dictionary update (see Figure 1a). In section 2, we derive a a fully continuous-time sparse coding model by making use of fast Langevin dynamics to sample latent variables and slower dynamics to co-evolve the dictionary based on these samples, as in Figure 1c.

Sampling with Langevin dynamics is well studied in both theory (Bussi & Parrinello, 2007) and in application to Bayesian learning (Welling & Teh, 2011). However, to our knowledge, this is the first fully analog approach to simultaneous inference and learning for sparse coding. Prior sampling-based approaches utilized a mixture-of-gaussians model and employed discrete Gibbs sampling over the mixture variables (Olshausen & Millman, 2000) or a method for preselecting parts of the space to sample via MCMC (Shelton, Sheikh, Berkes, Bornschein, & Lücke, 2011).

An additional advantage of Langevin dynamics is that it leads us to a simple procedure for sampling from the posterior when using an L_0 sparse prior that explicitly encourages latent variables to be set to zero rather than simply taking on small values (also known as a spike and slab prior). Normally such priors are avoided as finding the optimal sparse representation of a signal requires solving a combinatorial search problem. Instead, sparsity is enforced by imposing an L_1 cost function on the latent variables, which is used as a proxy for L_0 since it allows for convex optimization. However, in probabilistic terms, the L_1 cost corresponds to a Laplacian prior, which only weakly captures the notion of sparsity. We show in section 3 how Langevin sparse coding releases us from this restriction. By simple thresholding of a continuous variable undergoing Langevin dynamics, we obtain samples from the posterior using an L_0 sparse prior.

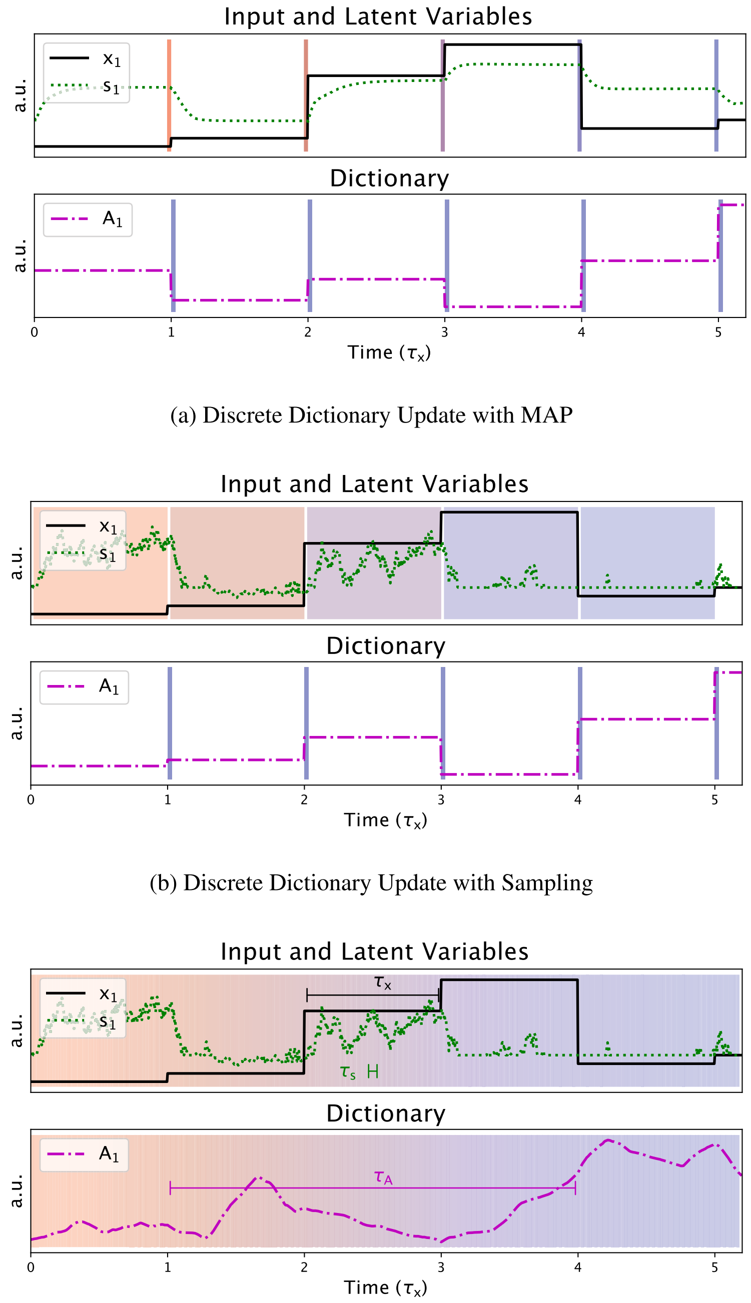
In section 4.1, we demonstrate the efficacy of this model for correct inference and learning using a synthetic data set. Furthermore in section 4.2 we demonstrate that this approach allows for learning the size of the dictionary, which was attempted in previous work using variational approximation of the posterior (Berkes, Turner, & Sahani, 2008). Then in section 4.3, we fit our L_0-sparse coding model to the van Hateren data set (van Hateren & van der Schaaf, 1998) of natural images. In addition to learning the dictionary elements, we provide an estimate for the sparsity of natural images.

To summarize, we present the following main contributions:

1. A theoretical formulation of simultaneous dynamics for sampling from latent variables and learning model parameters
2. Langevin sparse coding (LSC), a continuous-time, probabilistic model for simultaneous inference and learning in a sparse coding model
3. An efficient procedure for sampling from the posterior with an L_0-sparse prior
4. Learning not only the dictionary for representing natural images but also other parameters of the model such as the sparsity level and size of the dictionary

2 Langevin Sparse Coding

Sparse coding is a simple yet efficient algorithm for learning structure in data by finding a "dictionary" to describe patterns contained in the data. While it is formulated as a probabilistic latent-variable model, it is often approximated in practice by finding point estimates for the latent variables rather than sampling from their posterior distribution. As a result, it is difficult to make rigorous claims about the relation between the learned dictionary and the statistics of the data, and it is problematic to adapt other parameters of the model such as the degree of sparsity or overcompleteness of the dictionary. More broadly, it has hindered the advancement of



(a) Discrete Dictionary Update with MAP

(b) Discrete Dictionary Update with Sampling

(c) Continuous, Simultaneous Dictionary Update with Sampling sparse coding into a more powerful generative modeling framework (e.g., by incorporating hierarchical structure) since there is no principled way to learn the parameters of such models without sampling from the posterior.

In this section, we introduce Langevin sparse coding (LSC), which efficiently samples the latent variables of a sparse coding model and allows simultaneous, continuous updates of dictionary elements along with the latent variables. This last property is important in making the LSC framework amenable for fully analog implementation. We begin with a review of the canonical approach of discrete sparse coding (DSC). Next, we introduce simultaneous-update sparse coding (SSC) in which dictionary updates are made continuously and concurrent with the dynamics of the coefficients. Finally, we present LSC, where we demonstrate that the inherent noise to analog systems can be used to perform sampling.

2.1 Probabilistic Model. Sparse coding assumes that the data, $\mathbf{x} \in \mathbb{R}^D$, are described as a linear combination of elements from a dictionary $A \in \mathbb{R}^{D\times K}$ with additive gaussian noise $\mathbf{n} \in \mathbb{R}^D$:

$$\mathbf{x} = A\,\mathbf{s} + \mathbf{n}, \tag{2.1}$$

where $n_i \overset{iid}{\sim} N(0, \sigma^2)$. The coefficients $\mathbf{s} \in \mathbb{R}^K$ are latent variables that are assumed to be sparsely distributed, so that any given data point should be well approximated using a small number of columns of the dictionary. Sparsity is enforced by the choice of prior, typically chosen to be factorial::

$$p_s(\mathbf{s}) = \Pi_{i=1}^K p_s(s_i), \tag{2.2}$$

$$p_s(s_i) \propto \exp(-\lambda\, C(s_i)), \tag{2.3}$$

Figure 1: Illustration of three approaches to learning latent variable models. (a) In the standard approach, data $\mathbf{x}$ are presented at regular intervals (upper plot, black trace). A MAP estimate of latent variables is calculated via gradient descent or other iterative algorithm (green trace). The resulting estimate is used for a discrete update to the dictionary A (lower plot). The blue vertical bars illustrate the computational inefficiency where only a single point estimate of the coefficients is used to make a dictionary update. (b) In a sampling-based approach, for each data interval, multiple samples from the posterior are averaged for a dictionary update. The colored regions in the top panel show that many samples are collected to approximate the posterior distribution. However, the discrete dictionary updates (at corresponding vertical bands) make a fully analog implementation difficult. (c) Rather than waiting for the accumulation of samples, the dictionary A is updated simultaneously alongside the latent variables $\mathbf{s}$. The slow timescale of the dictionary compared to the latent variables $\tau_A \gg \tau_s$ allows for effective averaging. (Learning rates shown are purely for illustrative purposes.)

where the form of C is chosen so that $p_s(s_i)$ is peaked at $s_i = 0$ and with heavy tails away from zero. (Note that nonfactorial priors are also possible; see e.g., Garrigues & Olshausen, 2010.)

The posterior over the latent variables in this model may be written in exponential form,

$$p(\mathbf{s} \mid \mathbf{x}, A) \propto \exp(-E(A, \mathbf{s}, \mathbf{x})), \tag{2.4}$$

with the energy function $E(A, \mathbf{s}, \mathbf{x})$ given by

$$E(A, \mathbf{s}, \mathbf{x}) = \frac{\|\mathbf{x} - A\,\mathbf{s}\|_2^2}{2\sigma^2} + \lambda \sum_i C(s_i). \tag{2.5}$$

Thus, inferring a good (highly probable) interpretation of a given data sample, $\mathbf{x}$, corresponds to finding a set of latent variables, $\mathbf{s}$, with low energy, E.

The goal of learning in this model is to find a dictionary, A, that provides the best fit to the data. This is accomplished by solving for the maximum likelihood estimator (MLE) of the dictionary,

$$A^* = \arg\max_A \left\langle \log p(\mathbf{x}|A) \right\rangle_{\mathbf{x}\sim\mathcal{D}}, \tag{2.6}$$

where $\langle\cdot\rangle_{\mathbf{x}\sim\mathcal{D}}$ denotes expectation over the data set $\mathcal{D}$ (e.g., natural images). The MLE can be found through gradient ascent, where the gradient is given by

$$\nabla_A \langle \log p(\mathbf{x}|A)\rangle_{\mathbf{x}\sim\mathcal{D}} = \left\langle \left\langle -\nabla_A E(A, \mathbf{s}, \mathbf{x}) \right\rangle_{\mathbf{s}|\mathbf{x}} \right\rangle_{\mathbf{x}\sim\mathcal{D}} \tag{2.7}$$

$$= \left\langle \left\langle (\mathbf{x} - A\,\mathbf{s})\,\mathbf{s}^T \right\rangle_{\mathbf{s}|\mathbf{x}} \right\rangle_{\mathbf{x}\sim\mathcal{D}}, \tag{2.8}$$

where $\langle\cdot\rangle_{\mathbf{s}|\mathbf{x}}$ denotes expectation with respect to the posterior distribution $p(\mathbf{s} \mid \mathbf{x}, A)$ (see Lewicki & Olshausen, 1999, for a derivation). Thus, adapting the dictionary to the data requires, for each data sample $\mathbf{x}$, sampling from the posterior over $\mathbf{s}$ and computing the correlation between the residual, $\mathbf{x} - A\,\mathbf{s}$, and $\mathbf{s}$. The dictionary A would then be incrementally updated according to this correlation (see equation 2.8). Equilibrium is reached when $\left\langle \langle \hat{\mathbf{x}}(\mathbf{s})\mathbf{s}^T\rangle_{\mathbf{s}|\mathbf{x}} \right\rangle_{\mathbf{x}\sim\mathcal{D}} = \left\langle \mathbf{x}\langle \mathbf{s}^T\rangle_{\mathbf{s}|\mathbf{x}} \right\rangle_{\mathbf{x}\sim\mathcal{D}}$, with $\hat{\mathbf{x}}(\mathbf{s}) = A\,\mathbf{s}$.

Beyond learning the dictionary, one can adapt other parameters of the model such as σ and λ also via gradient descent. The gradients for these parameters are

$$\nabla_\sigma \langle \log p(\mathbf{x}|A)\rangle_{\mathbf{x}\sim\mathcal{D}} \propto \frac{1}{D} \left\langle \left\langle |\mathbf{x} - A\,\mathbf{s}|^2 \right\rangle_{\mathbf{s}|\mathbf{x}} \right\rangle_{\mathbf{x}\sim\mathcal{D}} - \sigma^2, \tag{2.9}$$

$$\nabla_\lambda \langle \log p(\mathbf{x}|A)\rangle_{\mathbf{x}\sim\mathcal{D}} \propto \frac{1}{K} \left\langle \left\langle \sum_i^K C(s_i) \right\rangle_{\mathbf{s}|\mathbf{x}} \right\rangle_{\mathbf{x}\sim\mathcal{D}} - \langle C(s)\rangle_{p_s(s)}. \tag{2.10}$$

Adapting these parameters similarly requires computing averages under the posterior distribution for each data sample. Note that when the sparse coding model objective is formulated purely in terms of its energy function (see equation 2.5), which is typically the case, then there is no principled way to adapt these parameters to the data. The probabilistic framework makes it possible, so long as it is tractable to sample from the posterior distribution.

2.2 Discrete Sparse Coding. In practice, the expectation over the data in equation 2.8 is approximated via stochastic gradient descent (SGD). For a batch of data of size N, $\{\mathbf{x}_n\}_{n=1\ldots N}$, the update rule is

$$\Delta A = \eta \, \frac{1}{N} \sum_{n=1}^{N} \left\langle (\mathbf{x}_n - A\,\mathbf{s}_n)\,\mathbf{s}_n^T \right\rangle_{\mathbf{s}_n|\mathbf{x}_n}, \tag{2.11}$$

where η specifies the learning rate. However, the expectation over $\mathbf{s}_n$ is usually considered intractable, and so in practice it is approximated by the maximum a posteriori (MAP) estimator of $\mathbf{s}_n$,

$$\mathbf{s}_n^* = \arg\min_{\mathbf{s}_n} E(A, \mathbf{s}_n, \mathbf{x}_n). \tag{2.12}$$

Solving via gradient descent yields the iterative update equation

$$\Delta \mathbf{s}_n \propto -\nabla_{\mathbf{s}} E(A, \mathbf{s}_n, \mathbf{x}_n) \tag{2.13}$$

$$= -\frac{1}{\sigma^2} A^T(\mathbf{x}_n - A\mathbf{s}_n) - \lambda\, C'(\mathbf{s}_n), \tag{2.14}$$

where C' is the derivative of cost function C above equation 2.5 and operates elementwise on $\mathbf{s}_n$. For each $\mathbf{x}_n$, equation 2.14 is iteratively evaluated until it converges to a solution. In order to make this a convex optimization, the cost function C is typically taken to be the L_1 norm, corresponding to a Laplacian prior $p_s(\mathbf{s})$. Gradient descent does not generally constitute the most efficient method for finding the MAP estimate, but we use it here as a step toward the development of LSC below.

The price we pay for approximating the expectation $\langle\,\cdot\,\rangle_{\mathbf{s}_n|\mathbf{x}_n}$ in equation 2.11 with a single MAP estimate is that it now becomes necessary to normalize the dictionary elements $A = (\mathbf{A}_1, \ldots, \mathbf{A}_K)$ after each update via

$$\mathbf{A}_i \leftarrow \frac{\mathbf{A}_i}{\|\mathbf{A}_i\|_2} \equiv \hat{\mathbf{A}}_i. \tag{2.15}$$

This is necessary because the MAP estimator $\mathbf{s}^*$ will consistently underestimate $\mathbf{s}$ such that it is biased toward zero (due to the sparse prior). As a result, each $\mathbf{A}_i$ will grow without bound unless normalized. (As we shall see, this no longer becomes necessary when we sample from the posterior.)

Both updates ΔA and $\Delta \mathbf{s}_n$ can be expressed more efficiently through gradient descent on a batch energy function:

$$E(A, S, X) \equiv \sum_{n=1}^{N} E(A, \mathbf{s}_n, \mathbf{x}_n) \tag{2.16}$$

$$= \frac{\|AS - X\|_{2,2}^2}{2\sigma^2} + \lambda \|S\|_{1,1}. \tag{2.17}$$

We have defined batch matrices $S \in \mathbb{R}^{K\times N}$ and $X \in \mathbb{R}^{D\times N}$. Above, $\|\cdot\|_{p,q}$ refer to the $L_{(p,q)}$ matrix norm, defined by

$$\|A\|_{p,q} = \left(\sum_j \left(\sum_i |a_{ij}|^p \right)^{\frac{q}{p}} \right)^{\frac{1}{q}}. \tag{2.18}$$

With the batch energy defined, the update rules are

$$S \leftarrow S - \eta_S \nabla_S E(A, S, X), \tag{2.19}$$

$$A \leftarrow A - \eta_A \nabla_A E(A, S, X), \tag{2.20}$$

$$A \leftarrow \text{Norm}(A), \tag{2.21}$$

where the Norm() operation corresponds to the normalization of equation 2.15.

To coordinate the updates of S and A, a nested loop must be used (see algorithm 1). The inner loop approximates the MAP estimator S^* while the outer loop finds the MLE of A.

A closely related cousin of DSC, the locally competitive algorithm (LCA; Rozell, Johnson, Baraniuk, & Olshausen, 2008), computes the MAP estimate by following dynamics that descend the energy E in a more efficient manner. Instead of doing direct gradient descent (see equation 2.14), $\mathbf{s}$ is taken to be a monotonically increasing, nonlinear function of another variable $\mathbf{u}$ that follows the gradient with respect to $\mathbf{s}$:

$$\Delta \mathbf{u}_n \propto -\nabla_{\mathbf{s}} E(A, \mathbf{s}_n, \mathbf{x}_n) \tag{2.22}$$

$$\mathbf{s}_n = g(\mathbf{u}_n), \tag{2.23}$$

where g operates elementwise on $\mathbf{u}$ and is determined by the choice of cost function C. For an L1 cost, g is a signed ReLU function with threshold λ:

$$g(u_i) = \begin{cases} 0 & |u_i| < \lambda \\ \text{sign}\,(u_i)(|u_i| - \lambda) & |u_i| \geq u_0. \end{cases} \tag{2.24}$$

Algorithm 1: Algorithm for Discrete Sparse Coding (DSC).

```
1: for $k \leftarrow 1$ to $N_A$ do
2:     $X \leftarrow$ SampleBatch()
3:     for $n \leftarrow 1$ to $N_s$ do
4:         $S \leftarrow S - \eta_S \cdot \nabla_S E(A, S, X)$
5:     end for
6:     $S^* \leftarrow S$
7:     $A \leftarrow A - \eta_A \cdot \nabla_A E(A, S^*, X)$
8:     $A \leftarrow \text{Norm}(A)$
9: end for
```

Note: Line 6 was included purely to emphasize S^* as a MAP estimate.

Other than this difference in the dynamics for MAP inference, which falls purely within the inner loop (line 4) of algorithm 1, both DSC and LCA update the dictionary based on a single MAP estimate and thus suffer the same inefficiency as depicted in Figure 1a.

2.3 Simultaneous (Update) Sparse Coding.

We note that the DSC algorithm requires the alternating update of the dictionary elements and coefficients. Typically this necessitates a digital clock for synchronization and is a major challenge toward fully analog implementation. In this section, we present an asychronous framework, simultaneous (update) sparse coding (SSC), where both the dictionary and coefficients are updated simultaneously.

Rather than updating the dictionary A at the end of the loop when S has converged to the MAP estimator S^*, SSC updates A continuously and concurrent with S. In search of dynamics amenable to analog computation, we take the step sizes to be infinitesimally small and arrive at the following set of differential equations,

$$\tau_S \dot{S} = -\nabla_S E(A, S, X(t)), \tag{2.25}$$

$$\tau_A \dot{A} = -\nabla_A E(A, S, X(t)), \tag{2.26}$$

while still enforcing the normalization constraint on A (equation 2.15). Here, we take $X(t)$ to be updated synchronously at regular intervals of τ_X. At each update, a new batch of samples is drawn.

To compare SSC and DSC, consider the simulation for SSC using the Euler method in algorithm 2.

Algorithm 2: Euler Method Simulation of SSC with Step Size of Δt and Regular Interval Input of X.

```
for $t \leftarrow 1$ to $t_{\max}/\Delta t$ do
    $dS \leftarrow \frac{\partial E}{\partial S}(A, S, X(t))$
    $dA \leftarrow \frac{\partial E}{\partial A}(A, S, X(t))$
    $S \leftarrow S - \frac{\Delta t}{\tau_S} \cdot dS$
    $A \leftarrow A - \frac{\Delta t}{\tau_A} \cdot dA$
    $A \leftarrow \mathrm{Norm}(A)$
end for
```

In a comparison of algorithms 1 and 2, the timescales τ can be related to the learning rates, η, and the number of iterations N_S. We stress an important difference between the two is that SSC is fully described through a set of coupled differential equations and requires no control structure (i.e., a nested for loop). This is especially desirable for analog implementation as a global clock is no longer a necessary. Furthermore, there is no longer a need for synchronous, regular input of the data X. While not explored here, dynamic input such as videos can be naturally processed without any frame-by-frame synchronization.

2.4 Sampling via Langevin Dynamics. Consider a time-varying system described by coordinates $\mathbf{u}(t)$ with energy $E(\mathbf{u})$. It can be modeled by Langevin dynamics according to the following stochastic differential equation:

$$\dot{\mathbf{u}} = -\nabla E(\mathbf{u}) + \sqrt{2T}\xi(t), \tag{2.27}$$

where $\xi(t)$ is independent gaussian white noise with $\langle \xi(t)\xi(t')^T \rangle = \mathbf{I}\delta(t - t')$. Under these dynamics, the distribution of $p(\mathbf{u}(t))$ over time will asymptotically converge to

$$p^{(\infty)}(\mathbf{u}) \propto e^{-E(\mathbf{u})/T}. \tag{2.28}$$

This relation suggests that we change the dynamics of SSC, equation 2.25, by injecting noise to $\dot{S}$:

$$\tau_S \dot{S} = -\nabla_S E(A, S, X) + \sqrt{2T\tau_S}\xi(t). \tag{2.29}$$

Note that under the scaling of $t \to t/\tau_S$, we have $\langle \xi(t/\tau_S)\xi(t'/\tau_S)^T \rangle = \mathbf{I}\delta(\tau_S^{-1}(t - t')) = \tau_S \mathbf{I}\delta(t - t') = \langle \sqrt{\tau_S}\xi(t)\sqrt{\tau_S}\xi(t')^T \rangle$. This necessitates the somewhat unexpected scaling factor of τ_S.

Following the above dynamics, for fixed A and input X, S will sample from the posterior distribution:

$$p_{S|X}(S(t)|X, A) \propto e^{-E(A,S,X)/T}. \tag{2.30}$$

This is a remarkable result: *By simply injecting noise into the continuous-time dynamics normally used for MAP inference in sparse coding, we obtain a dynamical system that naturally samples from the desired posterior distribution (see equation 2.4).* With $T = 0$, we recover the SSC dynamics above (see equations 2.25 and 2.26) where S converges to the MAP estimate.

A useful property of equation 2.29 is that the equilibrium distribution is independent of the time constant τ_S. By taking $\tau_A \gg \tau_S$, the assumption that A is fixed with respect to the dynamics of S can be upheld. Conversely, because S evolves much faster than A, the dynamics of A are well approximated by

$$\tau_A \dot{A} = -\langle \nabla_A E(A, S, X) \rangle_{S|A,X}. \tag{2.31}$$

This is the exact mean gradient that we originally sought to calculate (see equation 2.7).

In summary, we have derived a new method for inference and learning in a sparse coding model, Langevin sparse coding (LSC), as specified by the continuous, coupled dynamics of equations 2.29 and 2.31, which achieves the desired property illustrated in Figure 1c. Importantly, our aim in doing this is not simply to produce another MCMC algorithm but rather to move toward a physical realization that naturally implements these dynamics (an example is described in appendix C).

3 L_0-Sparse Prior

Since the goal of sparse coding is to represent each data item using a small number of nonzero latent variables, the prior should ideally have a sharp peak at zero in order to encourage many latent variables to be set to zero. In this case, the cost term C within the energy function, equation 2.5, would resemble an L_0 cost that rewards coefficients for being strictly zero (as opposed to being nonzero and merely small in amplitude). However such cost functions are not used in practice because they are not amenable to gradient-based or convex optimization methods for computing the MAP estimate. Instead, the L_1 cost is usually adopted as a proxy for L_0 as it has been shown to yield equivalent solutions under certain conditions (Tropp, 2006). However from the perspective of a probabilistic model, the L_1 cost corresponds to a Laplacian prior that only weakly expresses the notion of sparsity. In fact, the Laplacian is the maximum entropy distribution for a real-valued variable of a given mean absolute value. Here we show that

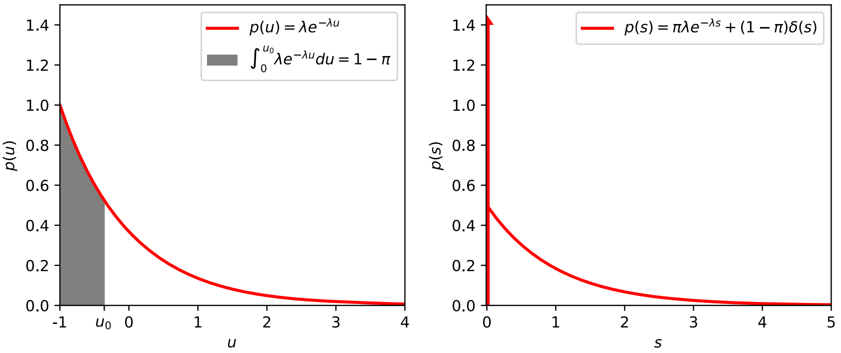


Figure 2: L_0-sparse prior. (Left) The exponential distribution $p(u)$. With the change of variable $s = f(u)$ via the application of a soft-thresholding function, we obtain the desired L_0-like distribution $p_0(s)$ shown in the right panel (shown for the region $s \geq 0$). The threshold parameter u_0 is chosen so that the probability weight of the delta function, $1 - \pi$, is equal to the shaded region in the left panel. These plots show the resulting distributions for $\lambda = 1, \pi = 0.5$.

the use of L_0-sparse priors becomes tractable in our sampling-based setting, and we develop a modified LSC formulation that enables efficient sampling from the posterior.

Consider the following prior consisting of a mixture of a delta function and Laplacian distribution (also known as a spike-and-slab prior; Mitchell & Beauchamp, 1988):

$$p_0(s) = \pi \, \lambda e^{-\lambda s} + (1 - \pi)\delta(s). \tag{3.1}$$

With π as the probability of being "active," $1 - \pi$ quantifies the L_0 sparsity, or how likely s is to be zero. When s is in the active state, it is exponentially distributed with mean $1/\lambda$ (see the right panel of Figure 2). Note that here and in what follows, we assume the latent variables to be nonnegative as opposed to allowing them to go positive or negative as is typically the case in sparse coding models.

To develop an efficient sampling strategy, we first define auxiliary variables $\mathbf{u}$ such that each u_i independently follows an exponential distribution:

$$p_U(u_i) = \lambda \, e^{-\lambda \, u_i}. \tag{3.2}$$

We then take the latent variables $\mathbf{s}$ to be given by $s_i = f(u_i)$ where f is a biased ReLU function:

$$s_i = f(u_i) = \begin{cases} 0 & u_i < u_0 \\ u_i - u_0 & u_i \geq u_0 \end{cases} \tag{3.3}$$

for some positive u_0. We can show that s_i is then distributed according to the prior $p_0(s)$ by marginalizing the joint distribution $p(s, u)$ over u as follows:

$$\begin{aligned}
p_S(s) &= \int_{-\infty}^{\infty} p(s|u)\, p_U(u) du \\
&= \int_{0}^{u_0} \delta(s)\, p_U(u) du + \int_{u_0}^{\infty} \delta(s - (u - u_0))\, p_U(u) du \\
&= \delta(s) \int_{0}^{u_0} p_U(u) du + p_U(s + u_0) \\
&= \delta(s)\, [1 - e^{-\lambda u_0}] + \lambda e^{-\lambda s}\, e^{-\lambda u_0} \\
&= [1 - \pi]\, \delta(s) + \pi\, \lambda e^{-\lambda s} \;\equiv\; p_0(s), \qquad (3.4)
\end{aligned}$$

with $\pi = e^{-\lambda u_0}$. The relation between $p(u)$, u_0 and $p(s)$ is illustrated in Figure 2.

To derive the Langevin dynamics for sampling from the posterior using the L_0-sparse prior above, we first rewrite the energy function in terms of $\mathbf{u}$:

$$E(A, \mathbf{u}, \mathbf{x}) = \frac{1}{2} \frac{\|\mathbf{x} - A f(|\mathbf{u}|)\|_2^2}{\sigma^2} + \lambda \|\mathbf{u}\|_1. \qquad (3.5)$$

We then let $\mathbf{u}$ follow Langevin dynamics governed by this energy function. Note that we can allow the u_i to move freely between positive and negative values and then use only their absolute value in evaluating the energy. This essentially reflects the dynamics about the origin, which avoids the problems associated with having an infinite energy barrier at $u_i = 0$. Letting $|\mathbf{u}|$ denote the elementwise absolute value of $\mathbf{u}$, the distribution of $|\mathbf{u}|$ will converge to

$$\begin{aligned}
p(|\mathbf{u}|\,|\,\mathbf{x}) &\propto \exp\left(-\|A\, f(|\mathbf{u}|) - \mathbf{x}\|_2^2/\sigma^2 - \lambda \|\mathbf{u}\|_1\right) \qquad (3.6) \\
&\propto p(\mathbf{x}|f(|\mathbf{u}|))\, p_U(|\mathbf{u}|) \qquad (3.7) \\
&= p(\mathbf{x}|\mathbf{s})\, p_0(\mathbf{s}). \qquad (3.8)
\end{aligned}$$

Thus, we obtain a second remarkable result: *By following Langevin dynamics on the energy in equation 3.5 with* $\mathbf{s} = f(|\mathbf{u}|)$, *we obtain samples from the posterior* $p(\mathbf{s}|\mathbf{x})$ *given by combining the likelihood with the* L_0*-sparse prior* $p_0(\mathbf{s})$. This is significant, because a MAP-estimate-based approach would be impossible with such a prior since the posterior will always have its maximum at $\mathbf{s} = 0$ regardless of the likelihood.

Applying the LSC equations 2.29 and 2.31 using the energy in equation 3.5, we obtain the following coupled stochastic differential equations for inference and learning in L_0-LSC:

$$\tau_u \dot{\mathbf{u}} = -A^T (A\,\mathbf{s} - \mathbf{x})\Theta(|\mathbf{u}| - \mathbf{u_0}) - \lambda\,\mathrm{sign}\,(\mathbf{u}) + \sqrt{2}\xi(t), \tag{3.9}$$

$$\mathbf{s} = f(|\mathbf{u}|), \tag{3.10}$$

$$\tau_A \dot{A} = -(A\,\mathbf{s} - \mathbf{x})\mathbf{s}^T, \tag{3.11}$$

where $\Theta(u)$ is the Heaviside function and $\xi(t)$ is independent gaussian white noise. Importantly, we can also learn u_0, and therefore the activation probability, π, via the dynamics

$$\dot{u}_0 \propto \left\langle \left\langle -\frac{\partial E}{\partial u_0} \right\rangle_{\mathbf{s}|\mathbf{x}} \right\rangle_{X \sim \mathcal{D}} \tag{3.12}$$

$$= \left\langle \left\langle A^T(A\mathbf{s} - \mathbf{x}) \cdot \mathbf{1}(\mathbf{s} > 0) \right\rangle_{\mathbf{s}|\mathbf{x}} \right\rangle_{\mathbf{x} \sim \mathcal{D}} . \tag{3.13}$$

4 Results

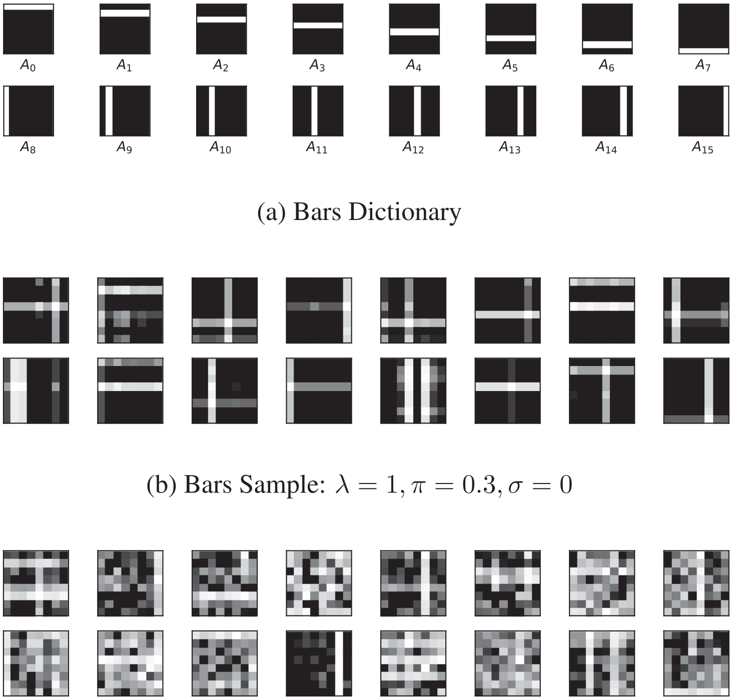
To study the efficacy of L_0-LSC, we first apply it to an artificial data set consisting of images of bars in different orientations. This provides a useful test case for evaluation since the causes that generate the data are known. We then turn to a data set of natural scenes where the ground truth is unknown.

4.1 Inference on Bars Data Set. For the Bars data set, samples are generated from a dictionary A consisting of vertical and horizontal lines (see Figure 3a). We compare results obtained on this data set against DSC as well as another method for training sparse coding, the locally competitive algorithm (LCA; Rozell et al., 2008).

We synthetically generate data as a linear combination of the dictionary with additive gaussian noise (see equation 2.1) where, $n_i \sim N(0, \sigma^2)$ and the coefficients are distributed according to L_0 zero-inflated exponential prior (Beckett et al., 2014; see equation 3.1). A sample drawn from this model without noise and with noise is shown in Figures 3b and 3c.

When trained on this data set, all three algorithms were successful at learning the correct dictionary. However, L_0-LSC can better capture the posterior distribution than either DSC or LCA due to the fact that it directly enforces L_0 sparsity. In both DSC and LCA, the sparsity is controlled by adjusting the parameter λ. However, the relationship between λ and L_0 sparsity (see Figure 4a) is rather indirect and no analytic expression is known. On the other hand, in L_0-LSC, a specific level of L_0-sparsity can be directly enforced by setting $u_0 = -\lambda^{-1}\log(\pi)$.

Moreover, the activation probability π can be learned by LSC without any guesswork or parameter search (see equation 3.12). Specifically,



(a) Bars Dictionary

(b) Bars Sample: $\lambda = 1, \pi = 0.3, \sigma = 0$

(c) Bars Sample: $\lambda = 1, \pi = 0.3, \sigma = 0.5$

Figure 3: The synthetic Bars data set used as a toy problem. (a) The dictionary is the collection of vertical and horizontal lines. (b) An example of a sample drawn from the data set. (c) Another sample with noise introduced.

simultaneous to the evolution of A, $\mathbf{u}$, the threshold parameter u_0 is treated as a variable evolves through gradient descent, $\dot{u}_0 \propto \nabla_{u_0} E$.

Figure 4b shows the convergence of model parameter π to match (approximately) the actual level of sparsity in the data. To further characterize the coefficients, the distributions of the nonnegative coefficients of the three algorithms were also plotted in Figure 5. Using a fixed dictionary, the algorithm was run either to infer the MAP estimate (DSC and LCA) or to sample from the posterior (L_0-LSC). This was done with a correctly learned dictionary (see Figure 3a) as well as a random dictionary (i.e., uncorrelated gaussian noise). In addition to having the correct L_0-sparsity, L_0-LSC correctly samples the posterior, which when averaged over the data matches the desired prior (see Figure 5c), as expected from theory. This is in contrast to nonstochastic algorithms where the inferred latent variable distribution

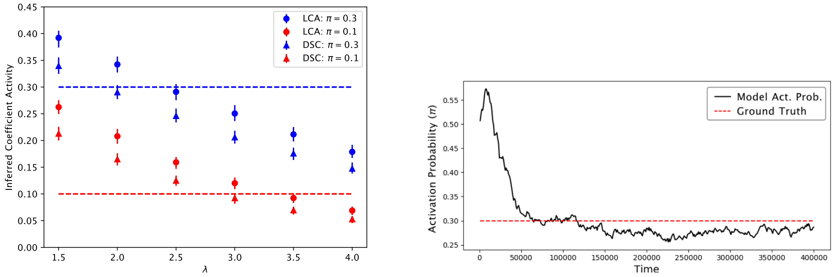


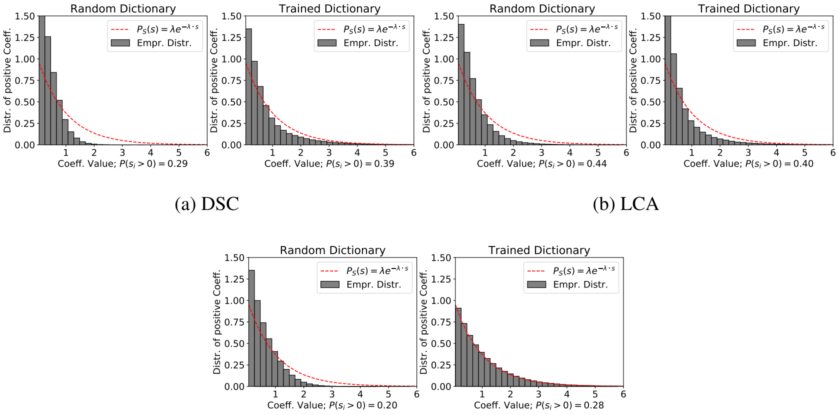
Figure 4: (a) LCA and DSC are trained on data generated with activation probability $\pi = 0.3$ (blue) and $\pi = 0.1$ (red). For both, a sweep in sparsity parameter λ is made. While a correspondence between λ and π exists, there is no analytic expression to automatically adapt these parameters to the data. Even with data of known sparsity, it is impossible to select the correct parameter λ to use. (b) With L_0-sparse LSC, the activation probability π is directly related to the parameter $u_0 = -\lambda^{-1} \log \pi$ and can be learned directly without a parameter search.

often exhibits a more pronounced peak at zero compared to the prior. A more quantitative analysis is provided in appendix B.

4.2 Learning the Dictionary Norm. For traditional sparse coding models such as DSC and LCA, which update the dictionary based on a single MAP estimate for each data item, it is necessary to normalize the dictionary elements after each update. However if the update is based on samples from the posterior, as specified in equation 2.8, then this is no longer necessary. As a result, when using LSC, there is no need for normalization. Instead, the dictionary element norms $\|\mathbf{A}_i\|$ will automatically grow or shrink as needed to optimize the model log likelihood.

The adaptive norm property can also be used to automatically select for the size of the dictionary. For data of dimension D, we consider a dictionary of size $K = \Omega \times D$ to have an (over)completeness of Ω. A $2\times$ overcomplete model was trained using the LSC algorithm using a fixed activation probability π, without normalizing the dictionary A. The resulting learned dictionary is shown in Figure 6b. In previous work by Berkes et al. (2008), annealed importance sampling (AIS; Neal, 2001) was used to approximate the marginal likelihood in order to find the optimal dictionary elements. However, L_0-LSC, without additional procedures, can be used to effectively do the same through attenuation of unnecessary dictionary elements. The learned dictionary contains exactly the Bars dictionary and the extra elements decay to nearly zero, as shown in Figure 6a.

When both $\|\mathbf{A}_i\|$ and π are being learned, a more stable solution is to have duplicated dictionary elements with reduced activity. This is shown in Figure 7a with a duplicated dictionary but halved activity (see Figure 7b).

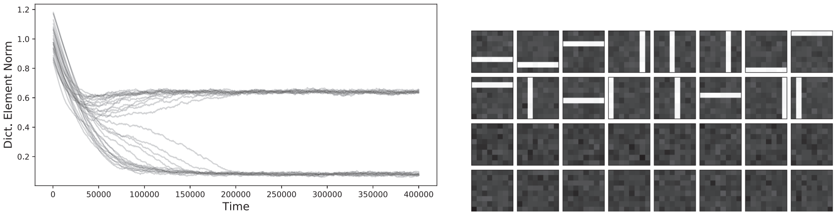


(a) DSC

(b) LCA

(c) L_0-LSC

Figure 5: The distribution of nonzero coefficients of each of the three algorithms. The dotted red line shows the prior of coefficients used in generating the data set. The left panel of each subfigure shows the empirical distribution when each algorithm is run with random dictionaries. The right panel shows the the distribution with learned dictionaries. Only L_0-LSC, with the correctly trained dictionary, achieves the distribution matching the prior.

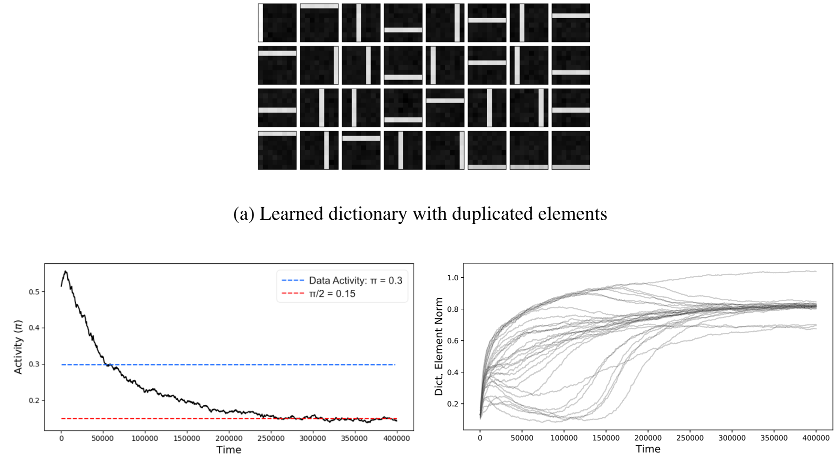


(a) Evolution of dictionary norms

(b) Unnecessary dictionary elements vanish.

Figure 6: Learning the dictionary size. (a) Dictionary norms bifurcate, with half decaying to nearly zero. (b) The remaining elements contain exactly one copy of the dictionary elements used to generate the data.

4.3 Natural Image Patches. We ran the L_0-LSC algorithm on a data set of 8×8 image patches of whitened natural scenes from the Van Hateren data set (van Hateren & Schaaf, 1998; Olshausen, 2013). First, the model activity was fixed at $\pi = 0.5$, and we used L_0-LSC to learn a 4× overcomplete dictionary ($K = 4 \times 64 = 256$). We can see in Figure 8 that a little more than half of the dictionary was utilized. The unused dictionary elements had a comparatively insignificant norm. In contrast to prior efforts to determine



(a) Learned dictionary with duplicated elements

(b) Activation probability π learned by twice overcomplete model

(c) Evolution of dictionary norms

Figure 7: LSC is used to learn both the dictionary size and activation probability of the same 2× overcomplete model. (a) The learned dictionary now contains duplicated elements. (b) But the activation probability π is half of the actual value used in generating the data.

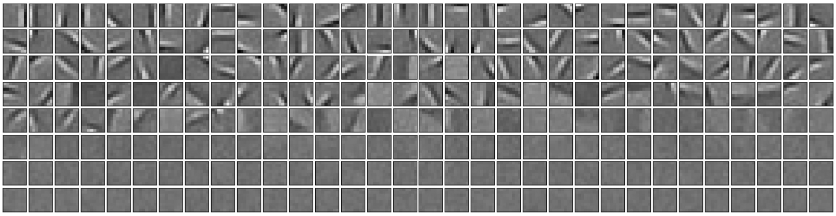
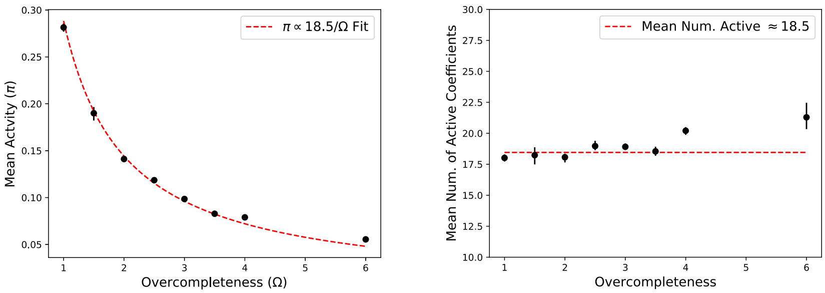


Figure 8: With activity fixed ($\pi = 0.5$), only a fraction of the total dictionary elements have significant norm; the rest vanish. The dictionary elements are sorted by their respective norms.

the optimal number of dictionary elements based on approximating the log likelihood (Berkes et al., 2008), this result emerges directly from dictionary learning in Langevin sparse coding.

Then, unfixing π, we allow the activity to be learned. Repeating the experiment at different levels of overcompleteness Ω, a correspondence between the activity and overcompleteness is plotted in Figure 9a. This relationship happens to be very well modeled by $\pi \propto \Omega^{-1}$. As a consequence,



(a) Mean activity at different levels of overcompleteness

(b) Mean number of active coefficients

Figure 9: (a) Using LSC to learn dictionaries for natural scenes at different levels of completeness Ω, the relationship $\pi \propto 1/\Omega$ is obtained. (b) This implies that the mean number of dictionary elements used to code each image is constant irrespective of the total number of dictionary elements learned. Error bars on both plots denote the 10% to 90% range.

the expected number of active dictionary elements, $\pi \times K = \pi \times \Omega \times D$ stays nearly constant irrespective of the overcompleteness Ω.

5 Discussion

Our main contribution in this letter is to show that by using Langevin dynamics to sample from posterior distributions, we obtain a set of continuous-time equations over analog state variables that enable probabilistically correct inference and learning in a latent variable model. While the use of Langevin dynamics for sampling in probabilistic models per se is not new (Cheng, Chatterji, Bartlett, & Jordan, 2018), our emphasis here is to show how these dynamics play out in the case of the sparse coding model and to point the way toward their efficient implementation in analog, electronic circuits that harness natural sources of stochasticity, for which we provide an example in appendix C. The basic operations involve computing inner products, thresholding, lateral inhibition, and thresholding, in addition to injection of a gaussian noise source. The first four of of these are shared with LCA, for which there already exist examples of both efficient analog implementations (Shapero, Charles, Rozell, & Hasler, 2012; Sheridan et al., 2017), and digital implementation using spiking neurons (Davies et al., 2021). In the latter case, LCA was shown to achieve the highest efficiency gains. The only additional component required for implementing LSC or L_0-LSC beyond these existing implementations is the injection of a gaussian noise source. This would seem quite natural since noise is intrinsic to any physical system; however, shaping the noise to be gaussian and i.i.d., and whether this is strictly required, remain important issues to resolve.

Finding efficient implementations is key to making probabilistic models tractable and scalable to practical problems of interest such as image analysis. Indeed, latent variable models such as Boltzmann machines are often considered intractable due to the inner loop required to sample over hidden unit states conditioned on input data. For this reason, practitioners often turn to approximations such as restricted Boltzmann machines (RBMs; (Hinton & Salakhutdinov, 2006)) or variational inference (VAEs; Kingma & Welling, 2013) so as to make the problem tractable by eliminating "explaining away"—that is, dependencies among hidden units conditioned on the data. But for most problems of interest in perception, explaining away is key (Olshausen, 2014). So doing away with explaining away in the interest of making the problem tractable simply dodges the very problem that needs to be solved. Here we show that there is an alternative approach that tackles sampling from posteriors head-on and makes it tractable via dynamics that could be naturally realized in a physical system.

An important next step will be to improve the efficiency of sampling by developing richer dynamical models. It is well known that the first-order Langevin dynamics we have used here can be slow to mix and reach equilibrium (Hennequin, Aitchison, & Lengyel, 2014). Adding higher-order terms to the dynamics such as momentum or even third-order terms has been shown to dramatically improve mixing time (Mou, Ma, Wainwright, Bartlett, & Jordan, 2021), and it has even been proposed that the balanced excitatory and inhibitory recurrent networks in cortex could serve such a function (Hennequin et al., 2014; Echeveste et al., 2020). The model we have proposed here could be modified along similar lines, and indeed this is a topic of ongoing work. Yet another route is to harness recent improvements in Hamiltonian Monte Carlo (Sohl-Dickstein, Mudigonda, & DeWeese, 2014).

With an efficient sampler in place, it becomes possible to adapt parameters of a sparse coding model beyond the dictionary, such as the level of sparsity or overcompleteness, which has not been possible in previous MAP-estimate-based approaches. Furthermore, through application of a threshold function to the stochastic dynamics, *we demonstrate that inference with an L_0-sparse prior, which has been avoided in most approaches by using L_1 as a proxy, can be readily computed and implemented* (see section 3). As shown in section 4.1, L_0-LSC is better at sampling from the posterior distribution as well as capable of learning the activation probability π of the latent variables $\mathbf{s}$. In applying the model to natural images (see section 4.3), we found that the mean number of dictionary elements used to encode an image is mostly invariant to the total dictionary size. This runs counter to previous results (Olshausen, 2013) showing that, on average, the number of elements required for reconstructing a given image decreases with larger dictionaries in which the elements take on more specific and diverse shapes. This discrepancy could possibly be reconciled by the fact that the previous work utilized MAP estimates rather than sampling, and so the learning was biased accordingly. Nonetheless, it is still intriguing that the mean number of dictionary elements in our case was near constant, suggesting that overcompleteness is an under used degree of freedom. However, another likely culprit is the assumption of a factorial prior, and it may be that an overcomplete dictionary loses its explanatory power under such a prior. Thus, it will be important to consider group sparse coding or other approaches for modeling statistical dependencies among latent variables (Garrigues & Olshausen, 2007, 2010) in order to fully realize the gains from overcompleteness.

Finally, another contribution of this work is to show how both learning and inference can be mapped to simultaneous dynamics at two different timescales. An underlying assumption in all implementations of probabilistic models on digital systems is the notion of a global clock. But the global clock is an impossibility for neural systems of any significant complexity. Our work presents an alternative approach to computing sparse coding, which allows for simultaneous updates of both latent variables and model parameters such as the dictionary elements. This type of concurrent dynamics removes the need of any such global clock.

More generally, the mixed timescale analog sampling framework on which LSC is based opens the way to learning richer generative models that capture dependencies among latent variables via horizontal connections (Garrigues & Olshausen, 2007) or top-down priors (Boutin, Franciosini, Ruffier, & Perrinet, 2020). And this goes beyond just sparse coding. In the future, we hope to develop analogous procedures for learning other latent variable models such as Boltzmann machines and hierarchical Bayesian models (Lee & Mumford, 2003).

References

Ackley, D. H., Hinton, G. E., & Sejnowski, T. J. (1985). A learning algorithm for Boltzmann machines. *Cognitive Science*, *9*(1), 147–169. 10.1207/s15516709cog0901_7

Beckett, S., Jee, J., Ncube, T., Pompilus, S., Washington, Q., Singh, A., & Pal, N. (2014). Zero-inflated Poisson (zip) distribution: Parameter estimation and applications to model data from natural calamities. *Involve: A Journal of Mathematics*, *7*(6), 751–767. 10.2140/involve.2014.7.751

Berkes, P., Orbán, G., Lengyel, M., & Fiser, J. (2011). Spontaneous cortical activity reveals hallmarks of an optimal internal model of the environment. *Science*, *331*(6013), 83–87. 10.1126/science.1195870, PubMed: 21212356

Berkes, P., Turner, R., & Sahani, M. (2008). On sparsity and overcompleteness in image models. In D. Koller, D. Schuurmans, Y. Bengio, & L. Bottou (Eds.), *Advances in neural information processing systems, 21* (pp. 89–96). Red Hook, NY: Curran.

Borders, W. A., Pervaiz, A. Z., Fukami, S., Camsari, K. Y., Ohno, H., & Datta, S. (2019). Integer factorization using stochastic magnetic tunnel junctions. *Nature*, *573*(7774), 390–393. 10.1038/s41586-019-1557-9, PubMed: 31534247

Boutin, V., Franciosini, A., Ruffier, F., & Perrinet, L. (2020). Effect of top-down connections in hierarchical sparse coding. *Neural Computation*, *32*(11), 2279–2309. 10.1162/neco_a_01325, PubMed: 32946716

Bussi, G., & Parrinello, M. (2007). Accurate sampling using Langevin dynamics. *Physical Review E*, *75*(5), 056707. 10.1103/PhysRevE.75.056707

Cheng, X., Chatterji, N. S., Bartlett, P. L., & Jordan, M. I. (2018). Underdamped Langevin MCMC: A non-asymptotic analysis. In *Proceedings of the 31st Conference on Learning Theory* (pp. 300–323).

Davies, M., Wild, A., Orchard, G., Sandamirskaya, Y., Guerra, G. A. F., Joshi, P., . . . Risbud, S. R. (2021). Advancing neuromorphic computing with Loihi: A survey of results and outlook. In *Proceedings of the IEEE*, *109*(5), 911–934. 10.1109/JPROC.2021.3067593

Donoho, D. L. (2006). Compressed sensing. *IEEE Transactions on Information Theory*, *52*(4), 1289–1306. 10.1109/TIT.2006.871582

Echeveste, R., Aitchison, L., Hennequin, G., & Lengyel, M. (2020). Cortical-like dynamics in recurrent circuits optimized for sampling-based probabilistic inference. *Nature Neuroscience*, *23*(9), 1138–1149. 10.1038/s41593-020-0671-1, PubMed: 32778794

Garrigues, P., & Olshausen, B. A. (2007). Learning horizontal connections in a sparse coding model of natural images. In J. Platt, D. Koller, Y. Singer, & S. Roweis (Eds.), *Advances in neural information processing systems*, *21* (pp. 505–512). Cambridge, MA: MIT Press.

Garrigues, P., & Olshausen, B. A. (2010). Group sparse coding with a Laplacian scale mixture prior. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, & A. Culotta (Eds.), *Advances in neural information processing systems, 23* (pp. 676–684). Red Hook. NY: Curran.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction*. Berlin: Springer Science & Business Media.

Hennequin, G., Aitchison, L., & Lengyel, M. (2014). Fast sampling-based inference in balanced neuronal networks. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, & A. Culotta (Eds.), *Advances in neural information processing systems*, *27*. Red Hook, NY: Curran.

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, *313*(5786), 504–507. 10.1126/science.1127647, PubMed: 16873662

Hinton, G. E., & Sejnowski, T. J. (1983). Optimal perceptual inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Piscataway, NJ: IEEE.

Hoyer, P. O., & Hyvärinen, A. (2003). Interpreting neural response variability as Monte Carlo sampling of the posterior. In S. Becker, S. Thrun, & K. Overmayer (Eds.), *Advances in neural information processing systems, 16* (pp. 293–300). Cambridge, MA: MIT Press.

Kingma, D. P., & Welling, M. (2013). *Auto-encoding variational Bayes.* arXiv:1312.6114.

Krestinskaya, O., Choubey, B., & James, A. (2020). Memristive GAN in analog. *Scientific Reports*, *10*(1), 1–14. 10.1038/s41598-020-62676-7, PubMed: 31913322

Lee, T. S., & Mumford, D. (2003). Hierarchical Bayesian inference in the visual cortex. *JOSA A*, *20*(7), 1434–1448. 10.1364/JOSAA.20.001434, PubMed: 12868647

Lewicki, M. S., & Olshausen, B. A. (1999). Probabilistic framework for the adaptation and comparison of image codes. *JOSA A, 16*(7), 1587–1601. 10.1364/JOSAA.16.001587

Mansinghka, V., & Jonas, E. (2014). *Building fast Bayesian computing machines out of intentionally stochastic, digital parts.* arXiv:1402.4914.

Mansinghka, V. K., Jonas, E. M., & Tenenbaum, J. B. (2008). *Stochastic digital circuits for probabilistic inference* (Technical Report MITCSAIL-TR, 2069). Cambridge, MA: MIT.

Mead, C. A., & Mahowald, M. A. (1988). A silicon model of early visual processing. *Neural Networks*, *1*(1), 91–97. 10.1016/0893-6080(88)90024-X

Mitchell, T. J., & Beauchamp, J. J. (1988). Bayesian variable selection in linear regression. *Journal of the American Statistical Association*, *83*(404), 1023–1032. 10.1080/01621459.1988.10478694

Mou, W., Ma, Y.-A., Wainwright, M. J., Bartlett, P. L., & Jordan, M. I. (2021). High-order Langevin diffusion yields an accelerated MCMC algorithm. *J. Mach. Learn. Res.*, *22*(42), 1–41.

Neal, R. M. (2001). Annealed importance sampling. *Statistics and Computing*, *11*(2), 125–139. 10.1023/A:1008923215028

Olshausen, B. A. (2013). Highly overcomplete sparse coding. In *Proceedings of Human Vision and Electronic Imaging XVIII* (Vol. 8651, p. 86510S). Bellingham, WA: SPIE. 10.1117/12.2013504

Olshausen, B. A. (2014). Perception as an inference problem. In G. Mangun & M. Gazzaniga (Eds.), *The cognitive neurosciences*. Cambridge, MA: MIT Press.

Olshausen, B. A., & Field, D. J. (1997). Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, *37*(23), 3311–3325. 10.1016/S0042-6989(97)00169-7, PubMed: 9425546

Olshausen, B. A., & Millman, K. J. (2000). Learning sparse codes with a mixture-of-gaussians prior. In T. Leen, T. Dietterich, & V. Tresp (Eds.), *Advances in neural information processing systems, 13* (pp. 841–847). Cambridge, MA: MIT Press.

Orbán, G., Berkes, P., Fiser, J., & Lengyel, M. (2016). Neural variability and sampling-based probabilistic representations in the visual cortex. *Neuron*, *92*(2), 530–543.

Roques-Carmes, C., Shen, Y., Zanoci, C., Prabhu, M., Atieh, F., Jing, L., . . . Soljačić, M. (2019). Photonic recurrent Ising sampler. In *Proceedings of the Conference on Lasters and Electro-Optics*. Washington, DC: Optica Publishing Group.

Rozell, C. J., Johnson, D. H., Baraniuk, R. G., & Olshausen, B. A. (2008). Sparse coding via thresholding and local competition in neural circuits. *Neural Computation*, *20*(10), 2526–2563. 10.1162/neco.2008.03-07-486, PubMed: 18439138

Shapero, S., Charles, A. S., Rozell, C. J., & Hasler, P. (2012). Low power sparse approximation on reconfigurable analog hardware. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, *2*(3), 530–541. 10.1109/JETCAS.2012.2214615

Shelton, J. A., Sheikh, A. S., Berkes, P., Bornschein, J., & Lücke, J. (2011). Select and sample: A model of efficient neural inference and learning. In S. Solla, T. Leen, & K. R. Müller (Eds.), *Advances in neural information processing systems* (pp. 2618–2626). Red Hook, NY: Curran.

Sheridan, P. M., Cai, F., Du, C., Ma, W., Zhang, Z., & Lu, W. D. (2017). Sparse coding with memristor networks. *Nature Nanotechnology*, *12*(8), 784. 10.1038/nnano.2017.83, PubMed: 28530717

Sohl-Dickstein, J., Mudigonda, M., & DeWeese, M. (2014). Hamiltonian Monte Carlo without detailed balance. In *Proceedings of the International Conference on Machine Learning* (pp. 719–726).

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, *58*(1), 267–288. 10.1111/j.2517-6161.1996.tb02080.x

Tropp, J. A. (2006). Algorithms for simultaneous sparse approximation. Part II: Convex relaxation. *Signal Processing*, *86*(3), 589–602. 10.1016/j.sigpro.2005.05.031

van Hateren, J. H., & van der Schaaf, A. (1998). Independent component filters of natural images compared with simple cells in primary visual cortex. In *Proceedings of Biological Sciences*, *265*(1394), 359–366. 10.1098/rspb.1998.0303, PubMed: 9523437

Wang, Z., Yang, J., Zhang, H., Wang, Z., Huang, T. S., Liu, D., & Yang, Y. (2015). *Sparse coding and its applications in computer vision*. Singapore: World Scientific.

Welling, M., & Teh, Y. W. (2011). Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning* (pp. 681–688). Madison, WI: Omnipress.

Wright, J., Ma, Y., Mairal, J., Sapiro, G., Huang, T. S., & Yan, S. (2010). Sparse representation for computer vision and pattern recognition. In *Proceedings of the IEEE*, *98*(6), 1031–1044. 10.1109/JPROC.2010.2044470

Received May 5, 2021; accepted January 22, 2022.