Exponential-Family Harmoniums with Neural Sufficient Statistics

Azwar Abdulsalam and Joseph G. Makin

Elmore Family School of Electrical and Computer Engineering, Purdue University
465 Northwestern Ave.
West Lafayette, IN 47907
azwar.abdulsalam1729@gmail.com, jgmakin@purdue.edu

Abstract

Exponential-family harmoniums (EFHs) generalize the restricted Boltzmann machine beyond Bernoulli random variables to other exponential families. Here we show how to extend the EFH beyond standard exponential families (Poisson, Gaussian, etc.), by allowing the sufficient statistics for the hidden units to be arbitrary functions of the observed data, parameterized by deep neural networks. This rules out the standard sampling scheme, block Gibbs sampling, so we replace it with a form of Langevin dynamics within Gibbs, inspired by a recent method for training Gaussian restricted Boltzmann machines (GRBMs). With Gibbs-Langevin, the GRBM can successfully model small data sets like MNIST and CelebA-32, but struggles with CIFAR-10, and cannot scale to larger images because it lacks convolutions. In contrast, our neural-network EFHs (NN-EFHs) generate high-quality samples from CIFAR-10 and scale well to CelebA-HQ. On these datasets, the NN-EFH achieves FID scores that are 25–50% lower than a standard energy-based model with a similar neural-network architecture and the same number of parameters; and competitive with noise-conditional score networks, which utilize more complex neural networks (U-nets) and require considerably more sampling steps.

Introduction

A basic trade-off in the design of generative models is between facility of generation and facility of inference. For example, (vanilla) variational autoencoders (Rezende, Mohamed, and Wierstra 2014; Kingma and Welling 2014) correspond to directed graphical models with two vertices, so generation is trivial because it involves only an "ancestral pass" (from parent to child) through the graph. The generative process can still be made highly expressive without much computational overhead by retaining standard parametric distributions (e.g., normal) but letting their parameters (mean, variance, etc.) depend on the latent variables through deep neural networks. There is, however, a price to paid for these highly expressive generative processes: inference must be approximate, since Bayes rule will not be computable (it will involve either an intractable integral or an infeasible summation). VAEs therefore employ a separate "recognition model" for inference, that is, for approximating the distribution over latent variables ($\hat{\boldsymbol{z}}$), given the observations ($\boldsymbol{x}$), under the generative model. The same is true of diffusion models (Sohl-Dickstein et al. 2015), although the generative and recognition models can be more faithfully aligned by extending the directed graph over hundreds of vertices. There is, in turn, a cost in generation: it requires a large number of steps (Song and Ermon 2019; Ho, Jain, and Abbeel 2020), perhaps in the form of an ODE solver (Song et al. 2021)), distillation (Yin et al. 2023), or some approximation. Or again, the recognition model can be dispensed with altogether if the map from simply-distributed latent variables to the data is designed to be deterministic and invertible (Bell and Sejnowski 1995; Dinh, Krueger, and Bengio 2015; Rezende and Mohamed 2015). But in practice the latent variables in such models—which must have the same dimension as the observed data—learn to encoder visual rather than semantic features of the (e.g.) images on which they are trained (Kirichenko, Izmailov, and Wilson 2020).

One long-standing alternative is define the generative model directly in terms of the posterior distribution, $\hat{p}(\hat{\boldsymbol{z}}|\hat{\boldsymbol{x}};\boldsymbol{\theta})$, as well as the "emission" distribution, $\hat{p}(\hat{\boldsymbol{x}}|\hat{\boldsymbol{z}};\boldsymbol{\theta})$. (Circumflexes indicate models throughout.) A joint distribution is implied by these specifications, but can itself be determined only up to an intractable normalizer. Thus, inference is easy, but sampling from the joint (generation) is hard. It generally takes the form of block Gibbs sampling, i.e. repeated iterations of sampling first from the posterior and then from the emission distribution. In the best known example of this approach, both emission and posterior are defined to be products over Bernoulli probability mass functions; the resulting undirected graphical model is the restricted Boltzmann machine (RBM) (Smolensky 1986).

Even with simple emissions and posteriors, as in the RBM, this architecture can model arbitrarily complex *marginal* distributions in the observation space (Le Roux and Bengio 2008). Furthermore, the emission and posterior can be generalized to any exponential-family distributions (or combinations thereof) (Welling, Rosen-Zvi, and Hinton 2004), in which case they are known as exponential-family harmoniums (EFHs). This is particularly important for the visible units, since not all data are well represented by (or can be coerced into) binary vectors. On the other hand, it can be shown that certain non-pathological distributions cannot

be represented *efficiently* by RBMs (and by extension EFHs) with a realistic number of hidden units and magnitudes of the weights (Martens et al. 2013). And as they stand, EFHs cannot incorporate computational structures that are known to be efficient at computing features from images, like multilayer convolutional neural networks.

Here we propose to augment EFHs with just such computational structures, in particular allowing the sufficient statistics of the posterior to depend on the inputs through deep neural networks. The price is that the emission is no longer a known parametric distribution, so standard sampling techniques will no longer work. But sampling was never one of the EFHs strengths, so this is perhaps a small price to pay. And it is still possible through a form of Langevin-within-Gibbs that has recently been proposed for Gaussian-Bernoulli RBMs (GRBMs) (Liao et al. 2022). Below we show that this technique allows us to train EFHs with neural sufficient statistics on, and subsequently generate high-quality samples from, complex datasets like CIFAR-10 and CelebA-HQ. The model outperforms the GRBM, as well as a standard energy-based model (without latent variables) defined by a similar neural-network architecture.

The Exponential-Family Harmonium

We derive the exponential-family harmonium from a slightly different perspective than that of Welling and colleagues (Welling, Rosen-Zvi, and Hinton 2004). This allows us to show that the resulting joint distribution is (under some mild conditions) the most generic possible. In particular, when the conditional distributions (posterior and emission) are exponential families,

$$\hat{p}(\hat{\boldsymbol{z}}|\hat{\boldsymbol{x}};\boldsymbol{\theta}) = h(\hat{\boldsymbol{z}})\exp\big\{\boldsymbol{\eta}(\hat{\boldsymbol{x}})^{\mathrm{T}}\boldsymbol{T}(\hat{\boldsymbol{z}}) - A(\boldsymbol{\eta}(\hat{\boldsymbol{x}}))\big\},$$
$$\hat{p}(\hat{\boldsymbol{x}}|\hat{\boldsymbol{z}};\boldsymbol{\theta}) = k(\hat{\boldsymbol{x}})\exp\big\{\boldsymbol{\zeta}(\hat{\boldsymbol{z}})^{\mathrm{T}}\boldsymbol{U}(\hat{\boldsymbol{x}}) - B(\boldsymbol{\zeta}(\hat{\boldsymbol{z}}))\big\},$$

then the only term in the joint that involves both hidden (latent) and visible (observed) vectors must be a bilinear form in the sufficient statistics, $\boldsymbol{U}(\hat{\boldsymbol{x}})^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}\boldsymbol{T}(\hat{\boldsymbol{z}})$, for some matrix $\mathbf{W}$.

To derive the form of the joint distribution, we note that the ratio of the conditionals is also the ratio of the marginals:

$$\frac{\hat{p}(\hat{\boldsymbol{x}}|\hat{\boldsymbol{z}};\boldsymbol{\theta})}{\hat{p}(\hat{\boldsymbol{z}}|\hat{\boldsymbol{x}};\boldsymbol{\theta})} = \frac{\hat{p}(\hat{\boldsymbol{x}};\boldsymbol{\theta})}{\hat{p}(\hat{\boldsymbol{z}};\boldsymbol{\theta})} = \frac{k(\hat{\boldsymbol{x}})e^{A(\boldsymbol{\eta}(\hat{\boldsymbol{x}}))}}{h(\hat{\boldsymbol{z}})e^{B(\boldsymbol{\zeta}(\hat{\boldsymbol{z}}))}}e^{\boldsymbol{\zeta}(\hat{\boldsymbol{z}})^{\mathrm{T}}\boldsymbol{U}(\hat{\boldsymbol{x}}) - \boldsymbol{\eta}(\hat{\boldsymbol{x}})^{\mathrm{T}}\boldsymbol{T}(\hat{\boldsymbol{z}})}.$$

We know an additional fact about this ratio: it must factor entirely into terms that refer to at most one of $\hat{\boldsymbol{z}}$ or $\hat{\boldsymbol{x}}$. This condition is satisfied for the terms occurring in the quotient. For it to hold also for the exponential term, it is necessary that

$$\boldsymbol{\zeta}(\hat{\boldsymbol{z}})^{\mathrm{T}}\boldsymbol{U}(\hat{\boldsymbol{x}}) - \boldsymbol{\eta}(\hat{\boldsymbol{x}})^{\mathrm{T}}\boldsymbol{T}(\hat{\boldsymbol{z}}) = \mu(\hat{\boldsymbol{z}}) - \nu(\hat{\boldsymbol{x}}), \qquad (1)$$

for some functions μ and ν. It can be shown (see the proof in the Appendix) that under some mild conditions, this requires each distribution's natural parameters to be an affine function of the other distribution's sufficient statistics,

$$\begin{aligned}\boldsymbol{\eta}(\hat{\boldsymbol{x}}) &= \boldsymbol{b}_{\hat{z}} + \mathbf{W}\boldsymbol{U}(\hat{\boldsymbol{x}})\\ \boldsymbol{\zeta}(\hat{\boldsymbol{z}}) &= \boldsymbol{b}_{\hat{x}} + \mathbf{W}^{\mathrm{T}}\boldsymbol{T}(\hat{\boldsymbol{z}}),\end{aligned} \qquad (2)$$

$$\hat{p}(\hat{\boldsymbol{z}}|\hat{\boldsymbol{x}};\boldsymbol{\theta}) = \mathrm{Bern}(\mathbf{W}\boldsymbol{U}(\hat{\boldsymbol{x}},\boldsymbol{\phi}) + \boldsymbol{b}_{\hat{z}})$$

$$\hat{p}(\hat{\boldsymbol{x}}|\hat{\boldsymbol{z}};\boldsymbol{\theta}) \propto \exp\big\{-\|\hat{\boldsymbol{x}} - \boldsymbol{\mu}\|^2 + \boldsymbol{U}(\hat{\boldsymbol{x}},\boldsymbol{\phi})^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}\hat{\boldsymbol{z}}\big\}$$

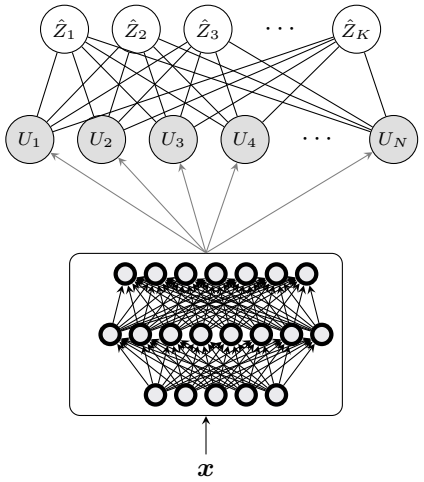

Figure 1: The NN-EFH.

with a shared, albeit transposed, linear transformation $\mathbf{W}$. Therefore, the marginal distributions are (up to the proportionality constants)

$$\hat{p}(\hat{\boldsymbol{z}};\boldsymbol{\theta}) \propto h(\hat{\boldsymbol{z}})e^{\boldsymbol{b}_{\hat{z}}{}^{\mathrm{T}}\boldsymbol{T}(\hat{\boldsymbol{z}}) + B(\boldsymbol{b}_{\hat{x}} + \mathbf{W}^{\mathrm{T}}\boldsymbol{T}(\hat{\boldsymbol{z}}))},$$
$$\hat{p}(\hat{\boldsymbol{x}};\boldsymbol{\theta}) \propto k(\hat{\boldsymbol{x}})e^{\boldsymbol{b}_{\hat{x}}{}^{\mathrm{T}}\boldsymbol{U}(\hat{\boldsymbol{x}}) + A(\boldsymbol{b}_{\hat{z}} + \mathbf{W}\boldsymbol{U}(\hat{\boldsymbol{x}}))};$$

and the conditional distributions are

$$\begin{aligned}\hat{p}(\hat{\boldsymbol{z}}|\hat{\boldsymbol{x}};\boldsymbol{\theta}) &= h(\hat{\boldsymbol{z}})e^{(\boldsymbol{b}_{\hat{z}} + \mathbf{W}\boldsymbol{U}(\hat{\boldsymbol{x}}))^{\mathrm{T}}\boldsymbol{T}(\hat{\boldsymbol{z}}) - A(\boldsymbol{b}_{\hat{z}} + \mathbf{W}\boldsymbol{U}(\hat{\boldsymbol{x}}))},\\ \hat{p}(\hat{\boldsymbol{x}}|\hat{\boldsymbol{z}};\boldsymbol{\theta}) &= k(\hat{\boldsymbol{x}})e^{\left(\boldsymbol{b}_{\hat{x}} + \mathbf{W}^{\mathrm{T}}\boldsymbol{T}(\hat{\boldsymbol{z}})\right)^{\mathrm{T}}\boldsymbol{U}(\hat{\boldsymbol{x}}) - B(\boldsymbol{b}_{\hat{x}} + \mathbf{W}^{\mathrm{T}}\boldsymbol{T}(\hat{\boldsymbol{z}}))}.\end{aligned} \qquad (3)$$

Multiplying a conditional by the appropriate marginal yields the joint distribution:

$$\begin{aligned}\hat{p}(\hat{\boldsymbol{z}},\hat{\boldsymbol{x}};\boldsymbol{\theta}) &= \hat{p}(\hat{\boldsymbol{z}}|\hat{\boldsymbol{x}};\boldsymbol{\theta})\hat{p}(\hat{\boldsymbol{x}};\boldsymbol{\theta})\\ &\propto h(\hat{\boldsymbol{z}})k(\hat{\boldsymbol{x}})e^{\boldsymbol{b}_{\hat{x}}{}^{\mathrm{T}}\boldsymbol{U}(\hat{\boldsymbol{x}}) + \boldsymbol{b}_{\hat{z}}{}^{\mathrm{T}}\boldsymbol{T}(\hat{\boldsymbol{z}}) + \boldsymbol{U}(\hat{\boldsymbol{x}})^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}\boldsymbol{T}(\hat{\boldsymbol{z}})}.\end{aligned}$$

Thus the joint takes the form of a Boltzmann distribution with energy

$$\begin{aligned}E(\hat{\boldsymbol{z}},\hat{\boldsymbol{x}},\boldsymbol{\theta}) = &-\log(h(\hat{\boldsymbol{z}})k(\hat{\boldsymbol{x}})) - \boldsymbol{b}_{\hat{x}}{}^{\mathrm{T}}\boldsymbol{U}(\hat{\boldsymbol{x}}) - \boldsymbol{b}_{\hat{z}}{}^{\mathrm{T}}\boldsymbol{T}(\hat{\boldsymbol{z}})\\ &- \boldsymbol{U}(\hat{\boldsymbol{x}})^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}\boldsymbol{T}(\hat{\boldsymbol{z}}).\end{aligned} \qquad (4)$$

The Sufficient Statistics

The sufficient statistics for the emission distribution, $\boldsymbol{U}(\hat{\boldsymbol{x}})$, determine which features of the input the EFH is sensitive to. In the RBM, for example, the emission is a product over Bernoullis, so its sufficient statistics are $\boldsymbol{U}(\hat{\boldsymbol{x}}) = \hat{\boldsymbol{x}}$. This means that no hidden unit can encode information about, e.g., pairwise correlations, $x_i x_j$. Although the marginal distributions (over either latent or observed variables) can be

arbitrarily complex (see above), we suspect that this limitation of the posterior damages the efficiency of the RBM. Furthermore, for any EFH, so long as the emission is defined to be a product over standard exponential families, the sufficient statistics $\boldsymbol{U}(\hat{\boldsymbol{x}})$ cannot contain interaction terms.

We can relax this restriction if we are willing to let go of standard exponential families for the emission. In particular, we will retain a product over Bernoullis for the posterior distribution,

$$\hat{p}(\hat{\boldsymbol{z}}|\hat{\boldsymbol{x}};\boldsymbol{\theta}) = \text{Bern}(\mathbf{W}\boldsymbol{U}(\hat{\boldsymbol{x}},\boldsymbol{\phi}) + \boldsymbol{b}_{\hat{z}}), \tag{5}$$

but allow the joint energy to have the form

$$E(\hat{\boldsymbol{z}},\hat{\boldsymbol{x}},\boldsymbol{\theta}) = \|\hat{\boldsymbol{x}} - \boldsymbol{\mu}\|^2 - \boldsymbol{U}(\hat{\boldsymbol{x}},\boldsymbol{\phi})^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}\hat{\boldsymbol{z}} - \boldsymbol{b}_{\hat{z}}{}^{\mathrm{T}}\hat{\boldsymbol{z}}, \tag{6}$$

where the sufficient statistics $\boldsymbol{U}(\hat{\boldsymbol{x}},\boldsymbol{\phi})$ are now allowed to be any deep neural network (Fig. 1). In particular, by letting $\boldsymbol{U}(\hat{\boldsymbol{x}},\boldsymbol{\phi})$ be a convolutional neural network, we allow the latent variables (under the posterior distribution) to be directly sensitive to the two-dimensional, translation-invariant features of natural images. We also include a quadratic term, $\|\hat{\boldsymbol{x}} - \boldsymbol{\mu}\|^2$, to encourage the energy to be convex far from the data, in order to speed convergence of the Langevin dynamics (Cheng et al. 2020). Here $\boldsymbol{\mu}$ is a learned vector of parameters.

Learning

To fit the marginal distribution over visible units, $\hat{p}(\hat{\boldsymbol{x}};\boldsymbol{\theta})$, to an observed distribution of data, $p(\hat{\boldsymbol{x}})$, we descend the gradient of the relative entropy of these distributions (the standard loss):

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\theta}}\,\mathrm{D}_{\mathrm{KL}}\{p(\boldsymbol{X})\|\hat{p}(\boldsymbol{X};\boldsymbol{\theta})\} &= \mathbb{E}_{\hat{\boldsymbol{Z}},\boldsymbol{X}}\left[-\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\theta}}\log\hat{p}\Big(\hat{\boldsymbol{Z}},\boldsymbol{X};\boldsymbol{\theta}\Big)\right] \\ &= \mathbb{E}_{\hat{\boldsymbol{Z}},\boldsymbol{X}}\left[\frac{\mathrm{d}E}{\mathrm{d}\boldsymbol{\theta}}(\hat{\boldsymbol{Z}},\boldsymbol{X},\boldsymbol{\theta})\right] - \mathbb{E}_{\hat{\boldsymbol{Z}},\hat{\boldsymbol{X}}}\left[\frac{\mathrm{d}E}{\mathrm{d}\boldsymbol{\theta}}(\hat{\boldsymbol{Z}},\hat{\boldsymbol{X}},\boldsymbol{\theta})\right].\end{aligned} \tag{7}$$

The final equality is well known (Hinton 2002). Its apparently simplicity belies the fact that the second term requires samples from the model joint—which, as we have lately discussed, are expensive to generate. We return to the sampling procedure below.

In the standard EFH, with energy given by Eq. 4, the parameters are $\boldsymbol{\theta} = \{\boldsymbol{b}_{\hat{z}}, \boldsymbol{b}_{\hat{x}}, \mathbf{W}\}$, and Eq. 7 takes a particularly elegant form as a difference of expected vectors or outer products. For our proposed energy, Eq. 6, the parameters are $\boldsymbol{\theta} = \{\boldsymbol{b}_{\hat{z}}, \boldsymbol{\mu}, \mathbf{W}, \boldsymbol{\phi}\}$. The derivative with respect to the last, $\boldsymbol{\phi}$, in particular can be computed with automatic differentiation through the sufficient-statistics network, $\boldsymbol{U}(\hat{\boldsymbol{x}},\boldsymbol{\phi})$.

Sampling

If the emission and posterior are chosen to be standard exponential families, then sampling is straightforward. The natural parameters can be computed with Eq. 2, converted into moment parameters with the inverse link function, and samples drawn from the distributions (Bernoulli, Poisson, Gaussian, etc.) with standard procedures. This is the case, for example, in RBMs and GRBMs. In our model, exact sampling is feasible for the posterior distribution, Eq. 5, but not for the emission distribution: since its sufficient statistics are a complex neural network, it does not correspond to any known exponential-family distribution.

Instead, we turn to Markov chain Monte Carlo (MCMC) to sample the emission, in particular to Langevin dynamics (Neal 2011). Sampling from the model's *joint* distribution still requires block Gibbs sampling, so the overall scheme is a form of "Langevin within Gibbs." More precisely, to draw a sample from the model, we make M pairs of alternating draws from the posterior and the emission (see Algorithm 1). Draws from the posterior follow the standard procedure for sampling from Bernoulli distributions. Sampling from the emission takes L steps of Langevin dynamics:

$$\hat{\boldsymbol{x}}^{(l+1)} = \hat{\boldsymbol{x}}^{(l)} - \epsilon\frac{\partial E}{\partial\hat{\boldsymbol{x}}}\Big(\hat{\boldsymbol{z}}_m, \hat{\boldsymbol{x}}^{(l)}, \boldsymbol{\theta}\Big) + \sqrt{2\epsilon T}\,\hat{\boldsymbol{y}}^{(l)}, \tag{8}$$

with $\hat{\boldsymbol{y}}^{(l)} \sim \mathcal{N}(\mathbf{0},\ \mathbf{I})$, ϵ the step size, and T the temperature. The energy in Eq. 8 is the joint energy of Eq. 6, evaluated at the vector of latent variables, $\hat{\boldsymbol{z}}_m$, produced by the preceding draw from the posterior. This energy has the same gradient as the energy of the emission, $\hat{p}(\hat{\boldsymbol{x}}|\hat{\boldsymbol{z}};\boldsymbol{\theta})$, since it differs only by a normalizer that is constant in $\hat{\boldsymbol{x}}$.

This still leaves some choices for the sampler. In theory, after a sufficient number of steps, samples from an ergodic Markov chain will eventually be drawn from the (unique) stationary distribution, independent of the location of the ini-

Algorithm 1: Gibbs-Langevin Training

Input: Data $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_N$, number of Gibbs steps M, number of Langevin steps L, step size ϵ, temperature T, learning rate η, initial parameters $\boldsymbol{\theta} = (\mathbf{W}, \boldsymbol{\phi})$

for $i = 1,\ldots,I$ **do**
  ▷ in practice, minibatches rather than single samples
  $\boldsymbol{z} \sim \hat{p}(\hat{\boldsymbol{z}}|\boldsymbol{x};\boldsymbol{\theta}) = \text{Bern}(\mathbf{W}\boldsymbol{U}(\boldsymbol{x},\boldsymbol{\phi}) + \boldsymbol{b}_{\hat{z}})$
  $\hat{\boldsymbol{x}}, \hat{\boldsymbol{z}} \leftarrow \textsc{TrainingSampler}(\boldsymbol{\theta}, M, L, \epsilon, T)$
  $\boldsymbol{\theta} \leftarrow \textsc{ParameterUpdate}(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{z}, \hat{\boldsymbol{x}}, \hat{\boldsymbol{z}}, \eta)$
end for

procedure $\textsc{TrainingSampler}(\boldsymbol{\theta}, M, L, \epsilon, T)$
  $\hat{\boldsymbol{x}} \sim \mathcal{N}(\mathbf{0},\ \mathbf{I})$
  $\hat{\boldsymbol{z}} \sim \hat{p}(\hat{\boldsymbol{z}}|\hat{\boldsymbol{x}};\boldsymbol{\theta}) = \text{Bern}(\mathbf{W}\boldsymbol{U}(\hat{\boldsymbol{x}},\boldsymbol{\phi}) + \boldsymbol{b}_{\hat{z}})$
  for $m = 1,\ldots,M$ **do**
    for $l = 1,\ldots,L$ **do**
      $\hat{\boldsymbol{y}} \sim \mathcal{N}(\mathbf{0},\ \mathbf{I})$
      $\hat{\boldsymbol{x}} \leftarrow \hat{\boldsymbol{x}} - \epsilon\frac{\partial E}{\partial\hat{\boldsymbol{x}}}(\hat{\boldsymbol{z}},\hat{\boldsymbol{x}},\boldsymbol{\theta}) + \sqrt{2\epsilon T}\,\hat{\boldsymbol{y}}$
    end for
    $\hat{\boldsymbol{z}} \sim \hat{p}(\hat{\boldsymbol{z}}|\hat{\boldsymbol{x}};\boldsymbol{\theta}) = \text{Bern}(\mathbf{W}\boldsymbol{U}(\hat{\boldsymbol{x}},\boldsymbol{\phi}) + \boldsymbol{b}_{\hat{z}})$
  end for
  return $\hat{\boldsymbol{x}}, \hat{\boldsymbol{z}}$
end procedure

procedure $\textsc{ParameterUpdate}(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{z}, \hat{\boldsymbol{x}}, \hat{\boldsymbol{z}}, \eta)$
  ▷ in practice, use AdaM
  return $\boldsymbol{\theta} - \eta\big(\frac{\mathrm{d}E}{\mathrm{d}\boldsymbol{\theta}}(\boldsymbol{z},\boldsymbol{x},\boldsymbol{\theta}) - \frac{\mathrm{d}E}{\mathrm{d}\boldsymbol{\theta}}(\hat{\boldsymbol{z}},\hat{\boldsymbol{x}},\boldsymbol{\theta})\big)$
end procedure

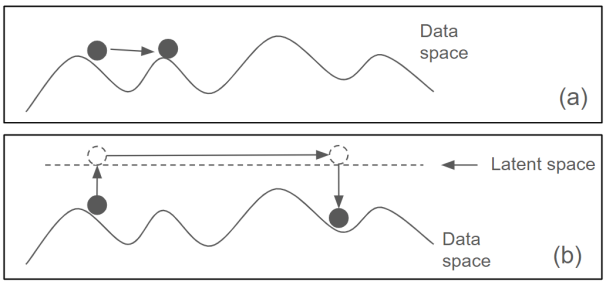

Figure 2: Conceptual illustration of the differences between LD and Gibbs-Langevin. (a) Langevin dynamics exploring the data space using gradient information. (b) Gibbs-Langevin exploring the space using a combination of data and latent space.

Algorithm 2: Gibbs-Langevin Testing

procedure TESTSAMPLER($\boldsymbol{x}, \boldsymbol{\theta}, M, L, \epsilon, T$)
  ▷ $\boldsymbol{x}$ from the data distribution
  $\hat{\boldsymbol{x}} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\hat{\boldsymbol{z}} \sim \hat{p}(\hat{\boldsymbol{z}}|\boldsymbol{x}; \boldsymbol{\theta}) = \text{Bern}(\mathbf{W}\boldsymbol{U}(\boldsymbol{x}, \boldsymbol{\phi}) + \boldsymbol{b}_{\hat{z}})$
  for $m = 1, \ldots, M$ **do**
    for $l = 1, \ldots, L$ **do**
      $\hat{\boldsymbol{y}} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
      $\hat{\boldsymbol{x}} \leftarrow \hat{\boldsymbol{x}} - \epsilon \frac{\partial E}{\partial \hat{\boldsymbol{x}}}(\hat{\boldsymbol{z}}, \hat{\boldsymbol{x}}, \boldsymbol{\theta}) + \sqrt{2\epsilon T}\hat{\boldsymbol{y}}$
    end for
    $\hat{\boldsymbol{z}} \sim \hat{p}(\hat{\boldsymbol{z}}|\hat{\boldsymbol{x}}; \boldsymbol{\theta}) = \text{Bern}(\mathbf{W}\boldsymbol{U}(\hat{\boldsymbol{x}}, \boldsymbol{\phi}) + \boldsymbol{b}_{\hat{z}})$
  end for
  return $\hat{\boldsymbol{x}}, \hat{\boldsymbol{z}}$
end procedure

tial sample. That implies that the Gibbs chain as well as the Langevin dynamics could be initialized anywhere. In practice, Langevin dynamics can be extremely slow to converge (Hinton 1999; Nijkamp et al. 2019, 2020). In their recent work on GRBMs, Liao and colleagues (Liao et al. 2022) propose allowing the Markov chains for the Langevin dynamics to "persist" across Gibbs steps—but not across weight updates—and we adopt that procedure here. During training, then, both the Gibbs sampler and the very first ($m = 1$) Langevin dynamics are initialized at noise. After that, the Langevin sampler is initialized at the current state of both the latent and observed vectors, although it only updates the latter (see again Algorithm 1). We found that this method of persistent Langevin dynamics within Gibbs sampling yields better results compared to a Langevin dynamics in which, at each Gibbs iteration, the Langevin chain is initialized from noise.

At test time (i.e., for generation of images), in the spirit of contrastive divergence (Hinton 2002), we initialize the Gibbs sampler at data rather than noise (Algorithm 2). Since the model is assumed to be well-trained at this point, samples from the model should be near the data distribution, so data initialization should significantly shorten the burn-in phase of the Markov chain. In our experiments, we found that good samples can be generated even when initializing the Gibbs sampler at noise, but that data initialization tended to produce the best results. Note, however, that the first Langevin chain is still initialized at noise, as during training.

Technically, Eq. 8 is only guaranteed to generate samples from the stationary distribution in the limit of very small steps, ϵ, so a Metropolis adjustment is required to correct the transitions of the Markov chain. However, in our model, as in others' (Hinton 1999; Nijkamp et al. 2019, 2020; Du and Mordatch 2019; Du et al. 2021), the acceptance probabilities computed under Metropolis-Hastings are very small, leading to a large number of rejected samples and intolerably long Langevin dynamics. Following the literature, we simply omit the Metropolis adjustment, accepting all samples, which we find works well in practice.

We hypothesize the superior performance of Gibbs-Langevin compared to baseline models trained solely with Langevin dynamics can be attributed to the significantly larger space explored by Gibbs-Langevin, particularly during the initial training phase. As illustrated in Fig. 2a, Langevin dynamics explores the data space by utilizing gradient information. However, in regions of the energy landscape where the gradient signal is weak (flat energy landscape)—especially during the initial training phase—exploration is limited. In contrast, Gibbs-Langevin dynamics leverages the latent space to make jumps (as shown in Fig. 2b), and when combined with the gradient-based sampling of Langevin dynamics, it enables exploration of a much larger space. To verify this, we analyzed the average distance traveled between the starting point and the end of the chain for both Langevin and Gibbs-Langevin dynamics. For Langevin dynamics, we performed 300 steps, while for Gibbs-Langevin, we used 5 Gibbs steps and 60 Langevin dynamics steps. In the case of Gibbs-Langevin, the distance was approximately 10 times greater than that observed with Langevin dynamics, confirming our hypothesis.

Related Work

Attempts to put convolutions into RBMs date back at least to Lee and colleagues (Lee et al. 2009). This approach builds a single-layer convolution into the RBM. Multilayer convolutions are then achieved by extending this RBM into a deep belief network (DBN), i.e., training a second RBM on the latent variables inferred by the first RBM, and so on. However, no attempt was made to generate images, and the model is limited to convolutions.

A more closely related idea is to incorporate an inverse autoregressive flow (IAF) (Kingma et al. 2016) into the energy function of the RBM (Liu, Xie, and Wang 2020). Since the flow is invertible, sampling in the observation space is still possible, and the energy gradient is computable simply via its Jacobian determinant. This is an intriguing approach and has some advantages over our own, chiefly obviating Langevin dynamics. On the other hand, the IAF limits the available neural-network architectures since it must retain invertibility; and the authors were apparently unable to generate images from their model.

Most recently, Liao and colleagues have proposed learn-

ing the variance of the input dimensions with a Gaussian-Bernoulli RBM (Liao et al. 2022). Our implementation of Langevin within Gibbs was inspired by theirs. But this model is limited to second-order statistics (indeed, only the diagonal of the covariance matrix), and as we see below is unable to capture the fine details of complex distributions.

Methods

We model the sufficient statistic using a neural network with an architecture similar to that described by Nijkamp and colleagues (Nijkamp et al. 2019, 2020): three convolutional layers interspersed with self-attention layers. We use this same neural network in an energy-based model (EBM) as a baseline for comparison. In particular, in the EBM, the model energy is computed from the penultimate layer by inner product with a (learned) vector of weights. For our NN-EFH, in contrast, the penultimate layer provides the sufficient statistics for the emission density. We additionally normalize the final energy by the batch size, a step we have found to help stabilize the training process. For MNIST and CIFAR-10, the final layer contains 1024 units; for CelebA, 4096. The NN-EFH additionally has a hidden layer ($\hat{\boldsymbol{z}}$), with sizes 2048 and 8192, respectively.

We train the NN-EFH to descend the gradient in Eq. 7 with Langevin-within-Gibbs, as lately described. In particular, we employ $L = 60$ steps of Langevin dynamics within $M = 5$ steps of Gibbs sampling. We train the baseline EBM using the standard MLE/min-KL loss. In our experience, the best results for EBMs trained with Langevin dynamics on complex datasets are achieved with a step size of $\epsilon = 1$ and temperature $T = 5\text{e}{-5}$, and we accordingly used these parameters in our Langevin-with-Gibbs when training the NN-EFH and for the (vanilla) Langevin dynamics for the EBM. To obtain samples from the EBM during training, we employ Langevin dynamics with the same parameters.

The recent implementation of the GRBM proposed by Liao and colleagues (Liao et al. 2022) also provides another useful point of comparison, since it is trained with the same Langevin-within-Gibbs scheme, and differs only in the expressivity of the sufficient statistics. For training GRBMs in this work, we use their source code and choices for hyperparameters.

All models are trained using stochastic gradient descent with the Adam optimizer on V100 GPUs for 50,000 iterations with a batch size of 64.

Results

Quality of Generated Images. We begin with MNIST (32x32). Fig. 3a shows digits generated from an NN-EFH. The model successfully learns the distribution and effectively captures the strokes, curves, and typical shapes without suffering from any kind of mode collapse. Model performance is comparable with the GRBM (Liao et al. 2022) and EBMs.

Similar appraisals can be made of images generated from NN-EFHs trained on Fashion MNIST (Fig. 3b) and the Oxford Flowers dataset (Fig. 3c). We draw attention to the latter in particular. Nijkamp et al. (2020) trained energy-based

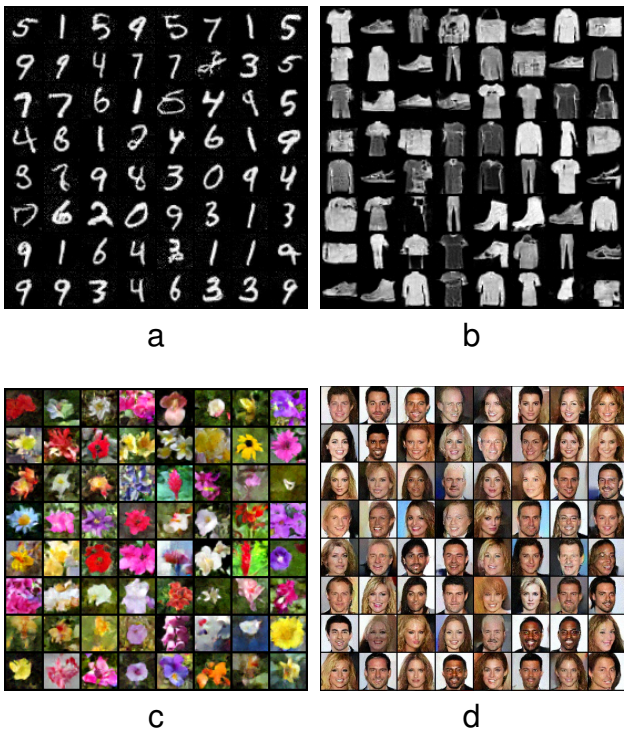

Figure 3: Samples generated from the NN-EFH. (a) MNIST, (b) FMNIST, (c) Oxford Flowers, (d) CelebA-HQ (64x64).

Model	FID
EBM base	22.7
NN-EFH	**11.3**
DCGAN (Radford, Metz, and Chintala 2016)	**12.5**
VAE (Kingma and Welling 2014)	38.76
Glow (Kingma and Dhariwal 2018)	23.32

Table 1: Frechet Inception distance on CelebA

models by gradient descent of the marginal cross entropy—the same loss we use for the NN-EFH (Eq. 7)—using (short-run) Langevin dynamics to draw samples from the model, and with an identical neural-network architecture to ours. Their generated flowers are noticeably blurrier (cf. Figure 7, second column, *op. cit.*). This very strongly suggests that the introduction of latent variables and the Gibbs-Langevin sampling scheme together yield a superior approach. FID scores are not reported in that work, so we reproduce their EBMs below and quantify the superiority of the NN-EFH.

Next we turn to CelebA-HQ (64x64). The dataset is a high-quality version of the CelebA dataset, consisting of 30,000 high-resolution celebrity images, refined and processed to ensure higher visual quality. Randomly selected samples generated from the NN-EFH are shown in Fig. 3d. The samples are high-quality and capture fine detail; the latter in particular appears difficult for the GRBM (see Fig. 4 in (Liao et al. 2022)). We also note the diversity of faces and of backgrounds, which likewise is noticeably absent from

Model	FID
EBM	42.3
JEM (Grathwohl et al. 2020)	38.40
IGEBM (Du and Mordatch 2019)	38.2
FlowCE (Gao et al. 2020)	37.3
NCSN [21] (Song and Ermon 2019)	25.2
Glow (Kingma and Dhariwal 2018)	48.9
NT-EBM	48.01
GRBM (Liao et al. 2022) (Gibbs-Langevin)	164
NN-EFH	**32.1**

Table 2: Frechet Inception distance on CIFAR-10

samples generated by the GRBM.

To quantify the quality of the celebrity faces generated by the NN-EFH, we compare their FID score (Heusel et al. 2017) against similarly sized recent models (Table 1). The NN-EFH outperforms an EBM with an identical neural-network architecture, as well as a vanilla VAE and Glow; it is similar to DCGAN.

Finally, we train all three models on CIFAR-10 (60,000 32x32 color images spread across 10 classes, each representing different objects such as animals and vehicles). Samples generated from each of three different models are shown in Fig. 4. The GRBM (Fig. 4a) fails to generate realistic samples from this dataset, likely due to its greater complexity and diversity compared to MNIST and CelebA. (We note that Liao et al. did not report results for CIFAR-10.) The EBM (Fig. 4b) and the NN-EFH (Fig. 4c) both generate realistic and diverse samples. To determine the relative quality of these samples, we again compute FID scores. Table 2 shows that the NN-EFH achieves significantly better scores than both of the other models. This is (to our knowledge) the first time an RBM/EFH has generated realistic samples from CIFAR-10. Indeed, its FID scores are competitive with noise-conditional score matching (Song and Ermon 2019).

Sample Diversity. The ability to generate high-quality images is a necessary but not sufficient condition for being a good generative model. It is also necessary to show that the samples are not overly similar to the training set. We focus here on CIFAR-10.

More precisely, we investigate whether the initialization of the Gibbs sampler at data at test time yokes the generated images to the initial images. Fig. 5a shows images from eight randomly chosen Gibbs-Langevin runs that were initialized at the images shown in the leftmost column. Each subsequent column shows the chain after 1 more Gibbs step. The noise introduced in initializing the first Langevin chain (see again Algorithm 2) clearly lifts the samples off the manifold of images, to which they subsequently return over the course of sampling—but not to the original image from the data distribution. Indeed, samples can even move from one category to another—as in the second row, where a sampler initialized at a truck yields a horse—suggesting that the sampler can mix across modes of the distribution.

A related, residual concern is whether the final sample might resemble, if not the initialization of the Gibbs sampler, some other image in the training set. Fig. 5b shows four randomly chosen generated images from the NN-EFH (leftmost column), along with their nearest neighbors in CIFAR-10 (columns 2–4) in terms of Euclidean distance in pixel space. The generated images resemble images in the training set only at the category level ("truck," "horse," etc.), but not in fine details. This is precisely the desired outcome.

Dataset	**CIFAR10**	**FMNIST**
Model	**Accuracy**	
$\boldsymbol{U}(\hat{\boldsymbol{x}})$	53%	71%
$\boldsymbol{\eta}(\hat{\boldsymbol{x}})$	47%	62%
EBM penultimate layer	29%	35%

Table 3: Linear evaluation accuracy on CIFAR10 and FMNIST datasets.

The Learned Features. After training an energy-based model (EBM) with latent variables, we sought to evaluate whether the learned latent variables encode useful information. One approach to testing this is to train a simple linear classifier on top of the learned features and to assess the accuracy across different datasets. For the NN-EFH, we used as features the sufficient statistics, $\boldsymbol{U}(\hat{\boldsymbol{x}})$, or the natural parameters for the latent variables, $\boldsymbol{\eta}(\hat{\boldsymbol{x}})$, on top of which the linear model was trained. As a baseline, we let the features be the penultimate layer of an EBM with a similar architecture, trained using short-run MCMC.

Table 3 shows the results for CIFAR-10 and FMNIST (chance is 10%). The NN-EFH evidently learns useful features, achieving respectable classification accuracies. Using the penultimate layer of the EBM for features cuts accuracies in half. These results highlight the potential of the NN-EFH model, which not only serves as a strong generative model but also learns meaningful latent representations. A promising direction for future research involves incorporating a contrastive loss into the existing training procedure to enhance the quality of latent representations (Kim and Ye 2022; Lee et al. 2023).

Discussion

The RBM (or EFH) represents a particular compromise between ease of inference and sample generation, in particular opting for exact inference while requiring MCMC to generate samples. Until now, this architecture has not been able to generate images from complex datasets like CIFAR-10. The premise of this study is that this is due in part to the absence of useful biases in the architecture, most obviously convolutions, which exploit the two-dimensional translation invariance of images—but also attention, which we have found to significantly improve generation from energy-based models. Such biases can be incorporated into the EFH, at the price of losing cheap conditional sampling of the observations given the latent variables—the "emission" distribution. We have argued that this is not such a steep price to pay, since generation from the joint or marginal distributions is already the limiting factor, and since Langevin dynamics can be used

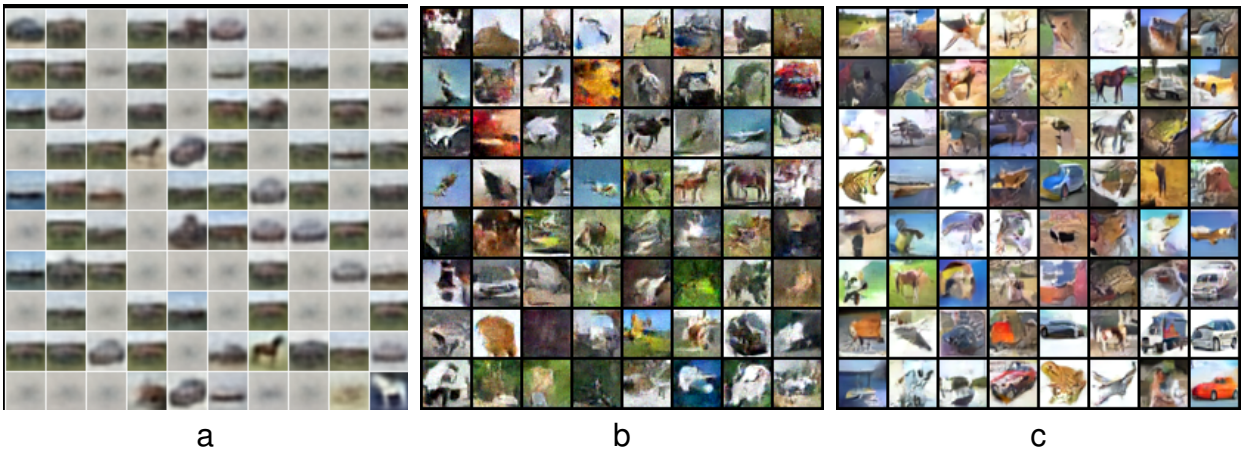

Figure 4: CIFAR-10 samples generated from (a) the GRBM, (b) an EBM, and (c) the NN-EFH.

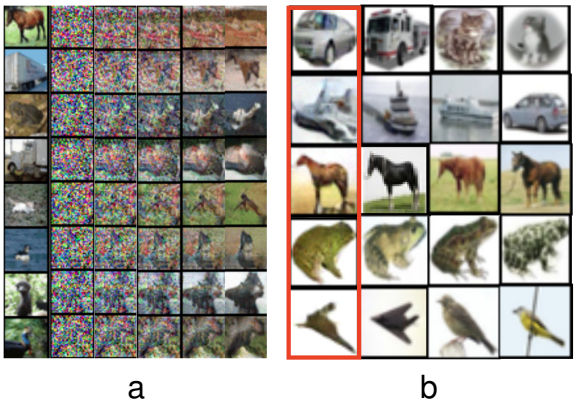

Figure 5: Sample diversity. (a) An illustration of the sampling process for Algorithm 2 for an NN-EFH trained on CIFAR-10. Each row corresponds to a different initial sample (first column); columns show samples after every full Gibbs step. Final samples (last column) do not strongly resemble the data initialization. (b) Nearest neighbors in the training set (columns 2–4) to the generated samples in column 1.

to sample the emission effectively. In particular, if the sufficient statistics for the latent variables are computed from the inputs with a deep neural network, the posterior (and consequently the joint) can be directly sensitive to complex features in those input. Furthermore, inference is still exact and cheap, since we do not alter the sufficient statistics for the posterior distribution—a product over Bernoulli distributions.

The resulting model can learn to generate from the standard datasets (MNIST, CelebA, CIFAR-10)—to our knowledge, the first time the RBM/EFH architecture has ever done so for the last in particular. We emphasize that we did not use a particularly large or complex neural network; the point of this study was to determine whether relaxing the limitations on the sufficient statistics of the emission could by itself yield superior models. Evidently, this is the case, since a GRBM trained with the same procedure cannot learn to generate from CIFAR-10, or (to our knowledge) capture fine detail in CelebA.

On CIFAR-10, the neural-network EFH also outperforms an energy-based model with the same neural-network architecture. The most obvious explanation is that the addition of latent variables improves model performance. But by itself, this amounts only to adding another layer to the neural network, with softplus nonlinearities (Martens et al. 2013). More interesting is the possibility that Gibbs sampling provides an advantage over Langevin dynamics alone.

In fact, the reverse also appears to be true. The sampling procedure, Langevin within Gibbs, was forced upon us by giving up a standard exponential family for the emission distribution. However, it may be beneficial *per se*. In the GRBM, standard techniques are available to sample from the emission density—it is a normal distribution. But Liao and colleagues (Liao et al. 2022) found that Langevin-within-Gibbs yields superior results for the GRBM. We consider this an important direction for future research.

Finally, since the network learns good models of moderately complex datasets, since inference is still exact, and since this posterior is sensitive to higher-order statistics in the input, the latent variables learned by this model may be useful for downstream tasks. They are, for example, more useful for classification than are features from a similarly stuctured EBM.

Appendix

Enforcing consistency between exponential-family emission and posterior distributions. We showed in the main text that when the emission and posterior distributions are both exponential families, the natural parameters are constrained by Eq. 1. To simplify the presentation, we repeat the constraint here (with the vector-valued functions named

alphabetically):

$$\mu(\hat{\boldsymbol{z}}) - \nu(\hat{\boldsymbol{x}}) = \boldsymbol{\gamma}(\hat{\boldsymbol{z}})^{\mathrm{T}}\boldsymbol{\delta}(\hat{\boldsymbol{x}}) - \boldsymbol{\beta}(\hat{\boldsymbol{x}})^{\mathrm{T}}\boldsymbol{\alpha}(\hat{\boldsymbol{z}}). \tag{9}$$

It is intuitive that this equation constrains the natural parameters (here, $\boldsymbol{\gamma}(\hat{\boldsymbol{z}})$ and $\boldsymbol{\beta}(\hat{\boldsymbol{x}})$): no $\hat{\boldsymbol{z}}$-$\hat{\boldsymbol{x}}$ interaction terms appear on the left-hand side, so those generated on the right must cancel. This is particularly restrictive since the interactions are created only through inner products. For example, if $\delta(\hat{\boldsymbol{x}})$ contains only terms quadratic in the elements of $\hat{\boldsymbol{x}}$, then $\boldsymbol{\beta}(\hat{\boldsymbol{x}})$ must contain such terms as well, in order to cancel them (except in the trivial case where $\boldsymbol{\gamma}(\hat{\boldsymbol{z}})$ is constant).

Let all the functions be polynomials in $\hat{\boldsymbol{z}}$ and $\hat{\boldsymbol{x}}$ of maximum degree D, and define the monomial bases

$$\hat{\boldsymbol{x}} \coloneqq \left[\hat{x}_1, \hat{x}_2, \ldots, {\hat{x}_1}^2, \hat{x}_1\hat{x}_2, \hat{x}_1\hat{x}_3, \ldots, {\hat{x}_K}^D\right]^{\mathrm{T}}$$
$$\hat{\boldsymbol{z}} \coloneqq \left[\hat{z}_1, \hat{z}_2, \ldots, {\hat{z}_1}^2, \hat{z}_1\hat{z}_2, \hat{z}_1\hat{z}_3, \ldots, {\hat{z}_K}^D\right]^{\mathrm{T}}.$$

(Notice that we have omitted the constants from these bases.) For appropriately shaped matrices $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$, vectors $(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{d})$, and scalar (k), Eq. 9 is equivalent to the equation

$$\begin{aligned}
&\boldsymbol{m}^{\mathrm{T}}\hat{\boldsymbol{z}} - \boldsymbol{n}^{\mathrm{T}}\hat{\boldsymbol{x}} + k \\
&\quad = (\boldsymbol{c} + \mathbf{C}\hat{\boldsymbol{z}})^{\mathrm{T}}(\boldsymbol{d} + \mathbf{D}\hat{\boldsymbol{x}}) - (\boldsymbol{b} + \mathbf{B}\hat{\boldsymbol{x}})^{\mathrm{T}}(\boldsymbol{a} + \mathbf{A}\hat{\boldsymbol{z}}) \\
&\quad = \hat{\boldsymbol{z}}^{\mathrm{T}}\left(\mathbf{C}^{\mathrm{T}}\mathbf{D} - \mathbf{A}^{\mathrm{T}}\mathbf{B}\right)\hat{\boldsymbol{x}} + \hat{\boldsymbol{z}}^{\mathrm{T}}\left(\mathbf{C}^{\mathrm{T}}\boldsymbol{d} - \mathbf{A}^{\mathrm{T}}\boldsymbol{b}\right) \\
&\qquad + \left(\boldsymbol{c}^{\mathrm{T}}\mathbf{D} - \boldsymbol{a}^{\mathrm{T}}\mathbf{B}\right)\hat{\boldsymbol{x}} + \left(\boldsymbol{c}^{\mathrm{T}}\boldsymbol{d} - \boldsymbol{a}^{\mathrm{T}}\boldsymbol{b}\right)
\end{aligned}$$

holding for all values of $\hat{\boldsymbol{z}}$ and $\hat{\boldsymbol{x}}$. Therefore,

$$\begin{aligned}
k &= \boldsymbol{c}^{\mathrm{T}}\boldsymbol{d} - \boldsymbol{a}^{\mathrm{T}}\boldsymbol{b} \\
\boldsymbol{m} &= \mathbf{C}^{\mathrm{T}}\boldsymbol{d} - \mathbf{A}^{\mathrm{T}}\boldsymbol{b} \\
-\boldsymbol{n}^{\mathrm{T}} &= \boldsymbol{c}^{\mathrm{T}}\mathbf{D} - \boldsymbol{a}^{\mathrm{T}}\mathbf{B} \\
0 &= \mathbf{C}^{\mathrm{T}}\mathbf{D} - \mathbf{A}^{\mathrm{T}}\mathbf{B}.
\end{aligned} \tag{10}$$

We shall only make use of the last of these, Eq. 10.

Now if the sufficient statistics are not to be redundant (i.e., are to be minimal), then $\mathbf{A}$ and $\mathbf{D}$ must both have linearly independent rows (otherwise some elements of $\boldsymbol{\alpha}$ and $\boldsymbol{\delta}$ could be computed from other elements). For neural-network sufficient statistics $\boldsymbol{\delta}(\hat{\boldsymbol{x}})$, as in the main text, we consider a polynomial approximation. In this case D would be very large, and the size of the basis larger still (on the order of $K + D$ choose D), so even high-dimensional $\boldsymbol{\delta}(\hat{\boldsymbol{x}})$ will not violate the requirement. Under the assumption, then, that $\mathbf{A}$ and $\mathbf{D}$ have linearly independent rows, there exists a right pseudo-inverse for $\mathbf{A}$, call it $\mathbf{A}^{\dagger}$, such that $\mathbf{A}\mathbf{A}^{\dagger} = \mathbf{I}$; and a right pseudo-inverse for $\mathbf{D}$, call it $\mathbf{D}^{\dagger}$, such that $\mathbf{D}\mathbf{D}^{\dagger} = \mathbf{I}$. It follows immediately from Eq. 10 that

$$\begin{aligned}
&\mathbf{C}^{\mathrm{T}} = \mathbf{A}^{\mathrm{T}}\mathbf{B}\mathbf{D}^{\dagger}, && \mathbf{B} = \left(\mathbf{C}\mathbf{A}^{\dagger}\right)^{\mathrm{T}}\mathbf{D} \\
\implies &\left(\mathbf{C}\mathbf{A}^{\dagger}\right)^{\mathrm{T}} = \mathbf{B}\mathbf{D}^{\dagger} =: \mathbf{W} \\
\implies &\mathbf{C}^{\mathrm{T}} = \mathbf{A}^{\mathrm{T}}\mathbf{W}, && \mathbf{B} = \mathbf{W}\mathbf{D},
\end{aligned} \tag{11}$$

where on the second line we have defined a new matrix $\mathbf{W}$. This allows us to rewrite the functions $\boldsymbol{\gamma}(\hat{\boldsymbol{z}})$ and $\boldsymbol{\beta}(\hat{\boldsymbol{x}})$ in terms of $\boldsymbol{\alpha}(\hat{\boldsymbol{z}})$ and $\boldsymbol{\delta}(\hat{\boldsymbol{x}})$ (resp.):

$$\begin{aligned}
\boldsymbol{\gamma}(\hat{\boldsymbol{z}}) &= \mathbf{C}\hat{\boldsymbol{z}} + \boldsymbol{c} \\
&= \mathbf{W}\mathbf{D}\hat{\boldsymbol{x}} + \boldsymbol{b} \\
&= \mathbf{W}^{\mathrm{T}}(\mathbf{A}\hat{\boldsymbol{z}} + \boldsymbol{a}) + \left(\boldsymbol{c} - \mathbf{W}^{\mathrm{T}}\boldsymbol{a}\right) \\
&= \mathbf{W}^{\mathrm{T}}\boldsymbol{\alpha}(\hat{\boldsymbol{z}}) + \left(\boldsymbol{c} - \mathbf{W}^{\mathrm{T}}\boldsymbol{a}\right), \\
\boldsymbol{\beta}(\hat{\boldsymbol{x}}) &= \mathbf{B}\hat{\boldsymbol{x}} + \boldsymbol{b} \\
&= \mathbf{W}^{\mathrm{T}}\mathbf{A}\hat{\boldsymbol{z}} + \boldsymbol{c} \\
&= \mathbf{W}(\mathbf{D}\hat{\boldsymbol{x}} + \boldsymbol{d}) + (\boldsymbol{b} - \mathbf{W}\boldsymbol{d}) \\
&= \mathbf{W}\boldsymbol{\delta}(\hat{\boldsymbol{x}}) + (\boldsymbol{b} - \mathbf{W}\boldsymbol{d}).
\end{aligned}$$

In a word, $\boldsymbol{\gamma}(\hat{\boldsymbol{z}})$ is an affine function of $\boldsymbol{\alpha}(\hat{\boldsymbol{z}})$, and $\boldsymbol{\beta}(\hat{\boldsymbol{x}})$ is an affine function of $\boldsymbol{\delta}(\hat{\boldsymbol{x}})$; and the linear transformations are transposes of each other.

Acknowledgments

The authors received support from NIH 1R01DC021600-01, the Showalter Research Trust, and the Google Research Scholar Program. They would like to thank Renjie Liao for a helpful discussion on the GRBM.

References

Bell, A. J.; and Sejnowski, T. J. 1995. An Information-Maximization Approach to Blind Separation and Blind Deconvolution. *Neural Computation*, 7(6): 1129–1159.

Cheng, X.; Chatterji, N. S.; Abbasi-Yadkori, Y.; Bartlett, P. L.; and Jordan, M. I. 2020. Convergence Rates for Langevin Monte Carlo in the Nonconvex Setting. http://arxiv.org/abs/1805.01648. Accessed: 2024-12-06.

Dinh, L.; Krueger, D.; and Bengio, Y. 2015. NICE: Nonlinear independent components estimation. *International Conference on Learning Representations, ICLR - Workshop Track Proceedings*, 1(2): 1–13.

Du, Y.; Li, S.; Tenenbaum, J.; and Mordatch, I. 2021. Improved Contrastive Divergence Training of Energy-Based Models. *International Conference on Machine Learning*, 139: 2837–2848.

Du, Y.; and Mordatch, I. 2019. Implicit generation and modeling with energy-based models. In *Advances in Neural Information Processing Systems*, volume 32.

Gao, R.; Nijkamp, E.; Kingma, D. P.; Xu, Z.; Dai, A. M.; and Wu, Y. N. 2020. Flow contrastive estimation of energy-based models. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 7515–7525.

Grathwohl, W.; Wang, K.-C.; Jacobsen, J.-H.; Duvenaud, D.; Norouzi, M.; and Swersky, K. 2020. Your Classifier is Secretly an Energy Based Model and You Should Treat it Like One. In *International Conference on Learning Representations*, 1–23.

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. *Advances in Neural Information Processing Systems*, 2017-Decem(Nips): 6627–6638.

Hinton, G. E. 1999. Products of experts. In *9th International Conference on Artificial Neural Networks: ICANN '99*, volume 2, 1–6. London; Institution of Electrical Engineers; 1999. ISBN 0 85296 721 7.

Hinton, G. E. 2002. Training Products of Experts by Minimizing Contrastive Divergence. *Neural Computation*, 14: 1771–1800.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems 33*, 1–12.

Kim, B.; and Ye, J. C. 2022. Energy-Based Contrastive Learning of Visual Representations. *Advances in Neural Information Processing Systems*, 35(NeurIPS).

Kingma, D. P.; and Dhariwal, P. 2018. Glow: Generative Flow with Invertible 1x1 Convolutions. In *Neural Information Processing Systems*, 1–15.

Kingma, D. P.; Salimans, T.; Jozefowicz, R.; Chen, X.; Sutskever, I.; and Welling, M. 2016. Improved Variational Inference with Inverse Autoregressive Flow. *Advances in Neural Information Processing Systems*, 4743–4751.

Kingma, D. P.; and Welling, M. 2014. Auto-Encoding Variational Bayes. In *International Conference on Learning Representations*, 1–14.

Kirichenko, P.; Izmailov, P.; and Wilson, A. G. 2020. Why normalizing flows fail to detect out-of-distribution data. *Advances in Neural Information Processing Systems*, 2020-December(NeurIPS).

Le Roux, N.; and Bengio, Y. 2008. Representational Power of Restricted Boltzmann Machines and Deep Belief Networks. *Neural Computation*, 20(6): 1631–49.

Lee, H.; Grosse, R.; Ranganath, R.; and Ng, A. Y. 2009. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. *Proceedings of the 26th Annual International Conference on Machine Learning ICML 09*, 1–8.

Lee, H.; Jeong, J.; Park, S.; and Shin, J. 2023. Guiding Energy-based Models via Contrastive Latent Variables. In *International Conference on Learning Representations*, 1–18.

Liao, R.; Kornblith, S.; Ren, M.; Fleet, D. J.; and Hinton, G. E. 2022. Gaussian-Bernoulli RBMs Without Tears. http://arxiv.org/abs/2210.10318. Accessed: 2024-12-06.

Liu, Y.; Xie, D.; and Wang, X. 2020. Generalized Boltzmann machine with deep neural structure. *AISTATS 2019 - 22nd International Conference on Artificial Intelligence and Statistics*, 89.

Martens, J.; Chattopadhyay, A.; Pitassi, T.; and Zemel, R. S. 2013. On the Representational Efficiency of Restricted Boltzmann Machines. *Neural Information Processing Systems*, 1–21.

Neal, R. M. 2011. MCMC using Hamiltonian Dynamics. In Brooks, S.; Gelman, A.; Jones, G.; and Meng, X.-L., eds., *Handbook of Markov Chain Monte Carlo*, chapter 5, 113–162. CRC Press. ISBN 9781420079425.

Nijkamp, E.; Hill, M.; Han, T.; Zhu, S. C.; and Wu, Y. N. 2020. On the anatomy of MCMC-based maximum likelihood learning of energy-based models. In *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence*, 5272–5280. ISBN 9781577358350.

Nijkamp, E.; Hill, M.; Zhu, S. C.; and Wu, Y. N. 2019. Learning non-convergent non-persistent short-run MCMC toward energy-based model. In *Advances in Neural Information Processing Systems*, volume 32.

Radford, A.; Metz, L.; and Chintala, S. 2016. Unsupervised representation learning with deep convolutional generative adversarial networks. *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*, 1–16.

Rezende, D. J.; and Mohamed, S. 2015. Variational Inference with Normalizing Flows. In *International Conference in Machine Learning*, volume 37.

Rezende, D. J.; Mohamed, S.; and Wierstra, D. 2014. Stochastic backpropagation and approximate inference in deep generative models. *International Conference in Machine Learning*, 32: 1278–1286.

Smolensky, P. 1986. Information processing in dynamical systems: Foundations of harmony theory. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, chapter 6, 194–281. MIT Press.

Sohl-Dickstein, J.; Weiss, E. A.; Maheswaranathan, N.; and Ganguli, S. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. *32nd International Conference on Machine Learning, ICML 2015*, 3: 2246–2255.

Song, Y.; and Ermon, S. 2019. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32(NeurIPS).

Song, Y.; Sohl-Dickstein, J.; Kingma, D. P.; Kumar, A.; Ermon, S.; and Poole, B. 2021. Score-Based Generative Modeling through Stochastic Differential Equations. In *International Conference on Learning Representations*, 1–36.

Welling, M.; Rosen-Zvi, M.; and Hinton, G. E. 2004. Exponential Family Harmoniums with an Application to Information Retrieval. In *Advances in Neural Information Processing Systems 17: Proceedings of the 2004 Conference*, 1481–1488.

Yin, T.; Gharbi, M.; Zhang, R.; Shechtman, E.; Durand, F.; Freeman, W. T.; and Park, T. 2023. One-step Diffusion with Distribution Matching Distillation. http://arxiv.org/abs/2311.18828. Accessed: 2024-12-06.