CONJUGATE ENERGY-BASED MODELS

Hao Wu*, **Babak Esmaeili***, **& Jan-Willem van de Meent**
Northeastern University
{wu.hao10,esmaeili.b,j.vandemeent}@northeastern.edu

Michael Wick
Oracle Labs
michael.wick@oracle.com

Jean-Baptiste Tristan
Boston College
tristanj@bc.edu

ABSTRACT

In this paper, we propose conjugate energy-based models (CEBMs), a new class of energy-based models that define a joint density over data and latent variables. The joint density of a CEBM decomposes into an intractable distribution over data and a tractable posterior over latent variables. CEBMs have similar use cases as variational autoencoders, in the sense that they learn an unsupervised mapping from data to latent variables. However, these models omit a generator network, which allows them to learn more flexible notions of similarity between data points. Our experiments demonstrate that conjugate EBMs achieve competitive results in terms of image modelling, predictive power of latent space, and out-of-domain detection on a variety of datasets.

1 INTRODUCTION

Deep generative models approximate a data distribution by combining a prior over latent variables with a neural generator, which maps latent variables to points on a data manifold. It is common to evaluate these models in terms of their ability to generate realistic examples, or their estimated densities for unseen data. However, an arguably more practical use case for these models is unsupervised representation learning. If a generator can faithfully represent the data in terms of a lower-dimensional set of latent variables, then we hope that these variables will encode a set of meaningful factors of variation that will be relevant to a broad range of downstream tasks.

Guiding a model towards a semantically meaningful representation requires some form of inductive bias. A large body of work on variational autoencoders (VAEs, Kingma & Welling (2013); Rezende et al. (2014)) has explored the use of priors as inductive biases. Relatively mild biases in the form of conditional independence are common in the literature on disentangled representations (Higgins et al., 2016; Kim & Mnih, 2018; Chen et al., 2018; Esmaeili et al., 2019). More generally, recent work has shown that defining priors that reflect the structure of the underlying data will lead to representations that are easier to interpret and generalize better. Examples include priors that represent objects in an image (Eslami et al., 2016; Lin et al., 2020b; Engelcke et al., 2019; Crawford & Pineau, 2019b), or moving objects in video (Crawford & Pineau, 2019a; Kosiorek et al., 2018; Wu et al., 2020; Lin et al., 2020a).

Despite steady progress, work on disentangled representations and structured VAEs still predominantly considers synthetic data sets. The likelihood model in VAEs is typically a neural generator that is optimized to accurately reconstruct all the examples in the training set. For complex natural scenes, learning a model that can produce pixel-perfect reconstructions poses fundamental challenges, given the combinatorial explosion of possible inputs. This is not only a problem for gen-

*Equal contribution.

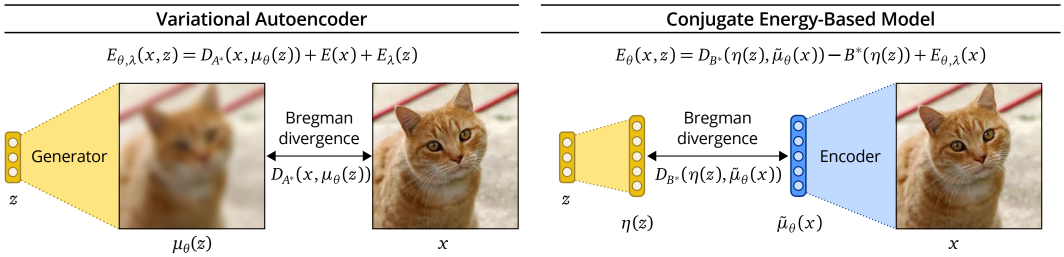


Figure 1: Comparison between a VAE and a CEBM. A VAE with a Gaussian or Bernoulli likelihood has an energy that can be expressed in terms of a Bregman divergence in data space $D_{A^*}(x, \mu_\theta(z))$ between an image x and the reconstruction from the generator network $\mu_\theta(z)$. The energy function in a CEBM can be expressed in terms of a Bregman divergence in the latent space $D_{B^*}(\eta(z), \tilde{\mu}_\theta(x))$ between a vector of natural parameters $\eta(z)$ and the output of an encoder network $\tilde{\mu}_\theta(x)$.

eration, but also from the perspective of the learned representation; a VAE must encode all factors of variation that give rise to large deviations in pixel space, regardless of whether these factors are semantically meaningful (e.g. presence and locations of objects) or not (e.g. shadows of objects in the background of the image).

In this paper, we consider energy-based models (EBMs) as an alternative to VAEs for learning representations in an unsupervised manner. The general idea of using EBMs for this purpose is by no means new; it has a long history in the context of restricted Boltzmann machines (RBMs) and related models (Smolensky, 1986; Hinton, 2002; Welling et al., 2004). Our motivation in the present work is to design a class of EBMs that retain the desirable features of VAEs, but employ a discriminative energy function to model data at an intermediate level of representation that does not necessarily encode all features of an image at the pixel level.

Concretely, we propose conjugate EBMs (CEBMs), a new family of energy-based models in which the energy function defines a neural exponential family. While the normalizer of CEBMs is intractable, we can nonetheless compute the posterior in closed form when we pair the likelihood with an appropriate conjugate bias term. Thus, the neural sufficient statistics in a CEBM fully determine the marginal likelihood and the encoder, thereby side-stepping the need for a generator (Figure 1).

We evaluate CEBMs in experiments that test how well learned representations agree with class labels (which are not used during training). We show that CEBMs learn a latent space in which neighbors are more likely to belong to the same class, which translates to increased performance in few-label downstream classification tasks.

2 Background

2.1 Conjugate Exponential Families

An exponential family is a set of distributions whose probability density can be expressed as

$$p(x \mid \eta) = h(x) \exp\big\{\langle t(x), \eta \rangle - A(\eta)\big\}, \tag{1}$$

where $h : \mathcal{X} \to \mathbb{R}^+$ is a base measure, $\eta \in \mathcal{H} \subseteq \mathbb{R}^K$ is a vector of natural parameters, $t : \mathcal{X} \to \mathbb{R}^K$ is a vector of sufficient statistics, and $A : \mathcal{H} \to \mathbb{R}$ is the log normalizer (or cumulant function),

$$A(\eta) = \log Z(\eta) = \int dx\, h(x) \exp\big\{\langle t(x), \eta \rangle\big\}. \tag{2}$$

If a likelihood belongs to an exponential family, then there exists a conjugate prior that is itself an exponential family

$$p(\eta \mid \lambda, \nu) = \exp\big\{\langle \eta, \lambda \rangle - A(\eta)\nu - B(\lambda, \nu)\big\}. \tag{3}$$

The convenient property of conjugate exponential families is that both the marginal $p(x \mid \lambda, \nu)$ and the posterior $p(\eta \mid x, \lambda, \nu)$ are tractable. If we define $\tilde{\lambda}(x) = \lambda + t(x)$, $\tilde{\nu} = \nu + 1$, then the posterior and marginal are

$$p(\eta \mid x, \lambda, \nu) = p(\eta \mid \tilde{\lambda}(x), \tilde{\nu}), \qquad p(x \mid \lambda, \nu) = h(x)\, \exp\big\{B(\tilde{\lambda}(x), \tilde{\nu}) - B(\lambda, \nu)\big\}. \tag{4}$$

2.2 Legendre Duality in Exponential Families

Two convex functions $A : \mathcal{H} \to \mathbb{R}^+$ and $A^* : \mathcal{M} \to \mathbb{R}^+$ on spaces $\mathcal{H} \subseteq \mathbb{R}^K$ and $\mathcal{M} \subseteq \mathbb{R}^K$ are conjugate duals when

$$A^*(\mu) := \sup_{\eta \in \mathcal{H}} \big\{ \langle \mu, \eta \rangle - A(\eta) \big\}. \tag{5}$$

When A is a function of Legendre type (see Rockafellar (1970)), the gradients of these functions define a bijection between conjugate spaces by mapping points to their corresponding suprema

$$\eta(\mu) = \nabla A^*(\mu), \qquad\qquad \mu(\eta) = \nabla A(\eta), \tag{6}$$

such that we can express $A^*(\mu)$ at the supremum as

$$A^*(\mu) = \langle \mu, \eta(\mu) \rangle - A(\eta(\mu)). \tag{7}$$

The log normalizer $A(\eta)$ of an exponential family is of Legendre type when the family is regular and minimal ($\mathcal{H}$ is an open set and sufficient statistics $t(x)$ are linearly independent; see Wainwright & Jordan (2008) for details). We refer to $\mathcal{M}$ as the mean parameter space, since we can express any $\mu \in \mathcal{M}$ as the expected value of the sufficient statistics $\mu(\eta) = \mathbb{E}_{p(x|\eta)}[t(x)]$.

2.3 Bregman Divergences and Exponential Families

A Bregman divergence for a function $F : \mathcal{M} \to \mathbb{R}$ that is continuously-differentiable and strictly convex on a closed set $\mathcal{M}$ has the form

$$D_F(\mu', \mu) = F(\mu') - F(\mu) - \langle \mu' - \mu, \nabla F(\mu) \rangle. \tag{8}$$

Well-known special cases of Bregman divergences include the squared distance ($F(\mu) = \langle \mu, \mu \rangle$) and the Kullback-Leiber (KL) divergence ($F(\mu) = \sum_k \mu_k \log \mu_k$).

Any Bregman divergence can be associated with an exponential family and vice versa, where $F(\mu) = A^*(\mu)$ is the conjugate dual of $A(\eta)$ (see Banerjee et al. (2005)). To see this, we re-express the log density of a (regular and minimal) exponential family using the substitution $\mu = \nabla A(\eta)$[1],

$$\begin{aligned} \log p(x \mid \eta) &= \langle t(x), \eta \rangle - A(\eta), \\ &= \big(\langle \mu, \eta \rangle - A(\eta) \big) + \langle t(x) - \mu, \eta \rangle, \\ &= A^*(\mu) + \langle t(x) - \mu, \nabla A^*(\mu) \rangle, \\ &= -D_{A^*}(t(x), \mu) + A^*(t(x)). \end{aligned} \tag{9}$$

In other words, the log density of an exponential family can be expressed in terms of a bias term $A^*(t(x))$, and a notion of agreement in the form of a Bregman divergence $D_{A^*}(t(x), \mu)$ between the sufficient statistics $t(x)$ and the mean parameters μ. We will make use of this property to provide an interpretation of both CEBMs and VAEs in terms of Bregman divergences.

3 Conjugate Energy-Based Models

In this paper we are interested in learning a probabilistic model that defines a joint distribution $p_{\theta,\lambda}(x, z)$ over high-dimensional data $x \in \mathbb{R}^D$ and a lower-dimensional set of latent variables $z \in \mathbb{R}^K$. The intuition that guides our work is that we would like to measure agreement between latent variables and data at a high level of representation, rather than at the level of individual pixels, where it may be more difficult to distinguish informative features from noise. To this end, we will explore energy-based models as an alternative to VAEs.

Concretely, we propose to consider models of the form

$$p_{\theta,\lambda}(x, z) = \frac{1}{Z_{\theta,\lambda}} \exp\big\{ - E_{\theta,\lambda}(x, z) \big\}, \qquad E_{\theta,\lambda}(x, z) = -\langle t_\theta(x), \eta(z) \rangle + E_\lambda(z). \tag{10}$$

[1] We here omit the base measure $h(x)$ for notational simplicity.

In this energy function, θ are the weights of a network $t_\theta : \mathbb{R}^D \to \mathbb{R}^H$, which plays the role of an encoder by mapping high-dimensional data to a lower-dimensional vector of neural sufficient statistics. The function $\eta : \mathbb{R}^K \to \mathbb{R}^H$ maps latent variables to a vector of natural parameters in the same space as the neural sufficient statistics. The function $E_\lambda : \mathbb{R}^K \to \mathbb{R}$ serves as an inductive bias, with hyperparameters λ, that plays a role analogous to the prior.

We will consider a bias $E_\lambda(z)$ in form of a tractable exponential family with sufficient statistics $\eta(z)$

$$E_\lambda(z) = -\log p(z \mid \lambda) = -\langle \eta(z), \lambda \rangle + B(\lambda). \tag{11}$$

We can then express the energy function as

$$E_{\theta,\lambda}(x, z) = -\langle \lambda + t_\theta(x), \eta(z) \rangle + B(\lambda). \tag{12}$$

This form of the energy function has a convenient property: It corresponds to a model $p_{\theta,\lambda}(x, z)$ in which the posterior $p_{\theta,\lambda}(z \mid x)$ is tractable. To see this, we make a substitution $\tilde{\lambda}_\theta(x) = \lambda + t_\theta(x)$ analogous to the one in Equation 4, which allows us to express the energy as

$$E_{\theta,\lambda}(x, z) = -\langle \eta(z), \tilde{\lambda}_\theta(x) \rangle + B(\tilde{\lambda}_\theta(x)) + E_{\theta,\lambda}(x), \quad E_{\theta,\lambda}(x) = -B(\tilde{\lambda}_\theta(x)) + B(\lambda). \tag{13}$$

We see that we can factorize the corresponding density $p_{\theta,\lambda}(x, z) = p_{\theta,\lambda}(x)\, p_{\theta,\lambda}(z \mid x)$, which yields a posterior and marginal that are analogous the distributions in Equation 4

$$p_{\theta,\lambda}(z \mid x) = p(z \mid \tilde{\lambda}_\theta(x)), \qquad p_{\theta,\lambda}(x) = \frac{1}{Z_{\theta,\lambda}} \exp\left\{ B\big(\tilde{\lambda}_\theta(x)\big) - B\big(\lambda\big) \right\}. \tag{14}$$

In other words, the joint density of this model factorizes into a tractable posterior $p_{\theta,\lambda}(z \mid x)$ and an intractable energy-based marginal likelihood $p_{\theta,\lambda}(x \mid \lambda)$. This posterior is conjugate, in the sense that it is in the same exponential family as the bias. For this reason, we refer to this class of models as conjugate energy-based models (CEBMs).

4 RELATIONSHIP TO VAES

CEBMs differ from VAEs in that they lack a generator network. Instead, the density is fully specified by the encoder network $t_\theta(x)$, which defines a notion of agreement $\langle \tilde{\lambda}_\theta(x), \eta(z) \rangle$ between data and latent variables in the latent space. As with other exponential families, we can make this notion of agreement explicit by expressing the conjugate posterior in terms of a Bregman divergence using the decomposition in Equation 9

$$E_{\theta,\lambda}(x, z) = D_{B*}(\eta(z), \tilde{\mu}_\theta(x)) - B^*(\eta(z)) + E_{\theta,\lambda}(x). \tag{15}$$

Here $B^*(\mu)$ is the conjugate dual of the the log normalizer $B(\lambda)$, and we use $\tilde{\mu}_\theta(x) = \mu(\tilde{\lambda}_\theta(x))$ as a shorthand for the mean-space posterior parameters. We see that maximizing the density corresponds to minimizing a Bregman divergence in the space of sufficient statistics of the bias.

In Figure 1, we compare CEBMs to VAE in terms of the energy function for the log density of the generative model. In making this comparison, we have to keep in mind that these models are trained using different methods, and that VAEs have a tractable density $p_\theta(x, z)$. That said, the objectives in both models maximize the marginal likelihood, so we believe that it is instructive to write down the corresponding Bregman divergence in the VAE likelihood. This likelihood is typically a Gaussian with known variance, or a Bernoulli distribution (when modeling binarized images). Both distributions have sufficient statistics $t(x) = x$. Once again omitting the base measure $h(x)$ for expediency, we can express the log density of a VAE as an energy

$$E_{\theta,\lambda}(x, z) = -\langle x, \eta_\theta(z) \rangle + A(\eta_\theta(z)) - \log p_\lambda(z) = D_{A^*}(x, \mu_\theta(z)) - A^*(x) - \log p_\lambda(z) \tag{16}$$

Here $A^*(x)$ is the conjugate dual of the log normalizer $A(\eta)$, and we use $\eta_\theta(z)$ and $\mu_\theta(z)$ to refer to the output of the generator network in the natural-parameter and the mean-parameter space respectively. To reduce clutter and accommodate the case where a base measure $h(x)$ is needed (e.g. that of a Gaussian likelihood with known variance), we will introduce the additional shorthands

$$E(x) = -A(x) - \log h(x), \qquad E_\lambda(z) = -\log p_\lambda(z). \tag{17}$$

Then the energy function of a VAE has the form $E_{\theta,\lambda}(x, z) = D_{A^*}(x, \mu_\theta(z)) + E(x) + E_\lambda(z)$. Like that of a CEBM, the energy function of a VAE contains a Bregman divergence, as well as two terms that depend only on x and z. However, whereas the Bregman divergence in CEBM is defined in the mean-parameter space of the latent variables, that of a VAE is computed in the data space.

5 INDUCTIVE BIASES

CEBMs have a property that is somewhat counter-intuitive. While the posterior $p_{\theta,\lambda}(z \mid x)$ in this class of models is tractable, the prior is in general not tractable. In particular, although the bias $-E_\lambda(z)$ is the logarithm of a tractable exponential family, it is not the case that $p_{\theta,\lambda}(z) = \exp\{-E_\lambda(z)\}$. Rather the prior $p_{\theta,\lambda}(z)$ has the form,

$$p_{\theta,\lambda}(z) = \frac{\exp\{-E_\lambda(z)\}}{Z_{\theta,\lambda}} \int dx \ \exp\{\langle t_\theta(x), \eta(z)\rangle\}.$$

In other words, $E_\lambda(z)$ defines an inductive bias, but this bias is different from the bias in a VAE (where the prior is always tractable by construction[2]), in the sense that it imposes only a soft constraint on the geometry of the latent space. In principle the bias in a CEBM can take the form of any exponential family distribution. Since products of exponential families are also in the exponential family, this covers a broad range of possible biases. For purposes of evaluation in this paper, we will constrain ourselves to two cases: a spherical Gaussian and a Mixture of Gaussians. We provide comprehensive derivation in Appendix C.

6 RELATED WORK

Energy-Based Latent Variable Models. The idea of using EBMs to jointly model data and latent variables has a long history in the machine learning literature. Examples of this class of models include restricted Boltzmann machines (Smolensky, 1986; Hinton, 2002), deep belief nets (DBNs, Hinton et al. (2006)), and deep Boltzmann machines (Salakhutdinov & Hinton, 2009). The idea of extending RBMs in exponential families and exploiting conjugacy to yield a tractable posterior is also not new and has been explored in Exponential Family Harmoniums (Welling et al., 2004). The models differ from CEBMs in that the they employ a bilinear interaction term $x^\top W z$, which ensures that both the likelihood $p(x \mid z)$ and $p(z \mid x)$ are tractable. In CEBMs, the corresponding term $t_\theta(x)^\top z$ is nonlinear, which means that the posterior is tractable, but the likelihood is not. We provide a detailed discussion regarding the connection to this class of models in Appendix A.

EBMs for Image Modelling. Recent work has shown that EBMs with convolutional energy functions can model distributions over images, given their competitive results in image generation (Xie et al., 2016; Nijkamp et al., 2019a;b; Du & Mordatch, 2019). This line of work typically focuses on generation and not on unsupervised representation learning as we do here. A line of work, which is similar to ours in spirit, employs EBMs as priors on the latent space of deep generative models (Pang et al., 2020; Aneja et al., 2020). These approaches, unlike our work, require a generator. Grathwohl et al. (2019); Liu & Abbeel (2020) have proposed to interpret a classifier as an EBM that defines a joint energy function on the data and labels. CEBMs with a discrete bias can interpreted as the unsupervised variant of this model class. Che et al. (2020) interpret a GAN as an EBM defined by both the generator and discriminator.

Training EBMs. While we here employ PCD for training, there is a large literature on alternative methods for training unnormalized models. An alternative class of methods is Noise Contrastive Estimation (NCE, (Gutmann & Hyvärinen, 2010)) in which we define a noise model and learn by discriminating between data and noise. An example of this is Gao et al. (2020), where the authors pretrain a flow that acts as the noise model for training EBMs with NCE. Another popular approach is score matching (SM, Hyvärinen & Dayan (2005); Vincent (2011); Song et al. (2020); Bao et al. (2020)), which matches the gradients of log probability density of the model and data distribution. Lastly, there are adversarial methods for avoiding MCMC sampling during MLE, where we introduce a variational distribution $q_\phi(x)$ that is trained alongside the model with a maximin objective (Grathwohl et al., 2021). We refer the readers to Song & Kingma (2021) for a more comprehensive study of training methods for EBMs. Note that because CEBMs can marginalize over z to get the marginal, all of the discussed methods are available for training, and therefore could be applied in the context of this work.

[2]The bias in a VAE contains the log prior $\log p_\lambda(z)$ and the log normalizer $A(\eta_\theta(z))$ of the likelihood. In a CEBM, by contrast, we omit the term $A_\theta(\eta(z)) = \log \int dx \exp\{\langle t_\theta(x), \eta(z)\rangle\}$, which is intractable, and hereby implicitly absorb it into its prior.

Figure 2: Samples from a CEBM trained on MNIST, Fashion-MNIST, SVHN and CIFAR-10.

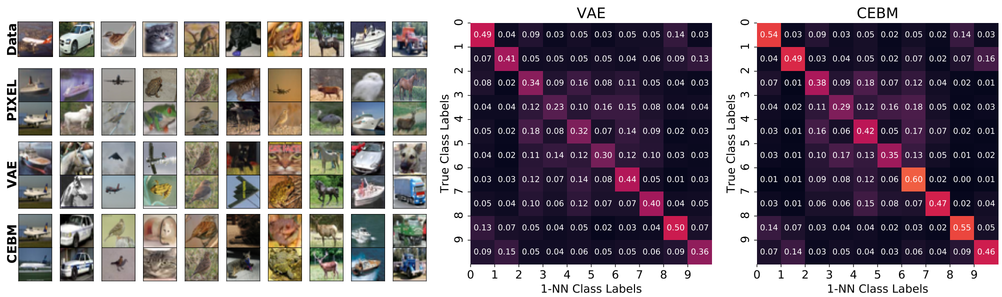


Figure 3: (*Left*) Samples from CIFAR-10 along with the top 2-nearest-neighbors in pixel space, the latent space of a VAE, and the latent space of a CEBM. (*Right*) Confusion matrices of 1-nearest-neighbor classification on CIFAR-10 based on L2 distance in the latent space.

7 Experiments

Our experiments evaluate to what extent CEBMs can learn representations that encode meaningful factors of variation, whilst discarding details about the input that we would consider noise. We begin with a qualitative evaluation by visualizing samples and then demonstrate that learned representations align with class structure, in the sense that nearest neighbors in the latent space are more likely to belong to the same class (Section 7.1). Next, we pre-train the models in an unsupervised manner, and use the learned representations for a few-label classification task (Section 7.2) and an out-of-distribution task (Appendix F).

We train CEBMs using presistent contarstive divergence (Tieleman, 2008; Du & Mordatch, 2019) where we sample from the models using Stochastic Gradient Langevin Dynamics (SGLD, (Welling & Teh, 2011)). We train our models using 60 SGLD steps, 90k gradient steps, batch size 128, Adam optimizer with learning rate 1e-4. We refer the readers to Appendix E & I for more training details.

7.1 Samples and Latent Space

We begin with a qualitative evaluation by visualizing samples from the model (see Figure 2). While generation is not our intended use case in this paper, such samples do serve as a useful diagnostic, in the sense that they allow us to visually inspect what characteristics of the data are captured by the learned representation. We initialize the samples with uniform noise and run 500 SGLD steps. We observe that the distribution over images is diverse and captures the main characteristics of the dataset. Sample quality is roughly on par with samples from other EBMs Nijkamp et al. (2019a), although it is possible to generate samples with higher visual quality using class-conditional EBMs Du & Mordatch (2019); Grathwohl et al. (2019); Liu & Abbeel (2020) (which assume access to labels).

To assess to what extent the representation in CEBMs aligns with classes in each dataset, we look at the agreement between the label for each data point and the label of its nearest neighbor in the latent space. In Figure 3, we show samples from CIFAR-10, along with the images that correspond to the nearest neighbors in pixel space, the latent space of a VAE, and the latent space of a CEBM. The distance in pixel space is a poor measure of similarity in this dataset, whereas proximity in the latent space is more likely to agree with class labels in both VAEs and CEBMs.

Table 1: Average classification accuracy on the test set. We train a variety of deep generative models on MNIST, Fashion-MNIST, CIFAR-10, and SVHN in an unsupervised way. Then we use the learned latent representations to train logistic classifiers with 1, 10, 100 training examples per class, and the full training set. We train each classifier 10 times on randomly drawn training examples.

	MNIST				Fashion-MNIST				CIFAR-10				SVHN			
Models	1	10	100	*full*	1	10	100	*full*	1	10	100	*full*	1	10	100	*full*
VAE	42	85	92	95	41	63	72	81	16	22	31	38	**13**	13	16	36
GMM-VAE	53	86	93	97	49	68	79	84	**19**	23	33	39	**13**	14	23	56
BIGAN	33	67	85	91	46	65	75	81	18	**30**	**43**	52	11	20	42	56
IGEBM	63	89	95	97	50	**70**	79	83	16	26	33	42	10	16	35	49
CEBM	**67**	89	95	97	**52**	**70**	77	83	**19**	**30**	42	**53**	12	**25**	**48**	**70**
GMM-CEBM	**67**	**91**	**97**	**98**	**52**	**70**	**80**	**85**	16	29	42	52	10	17	39	60

In Figure 3 (right), we quantify this agreement by computing the fraction of neighbors in each class conditioned on the class of the original image. We see a stronger alignment between classes and the latent representation in CEBMs, which is reflected in higher numbers on the diagonal of the matrix. On average, a fraction of 0.38 of the nearest neighbors are in the same class in the VAE, whereas 0.45 of the neighbors are in the same class in the CEBM. This suggests that the representation in CEBMs should lead to higher performance in downstream tasks.

7.2 Few-label Classification

To evaluate performance in settings where few labels are available, we train logistic classifiers using the learned latent features (trained in an unsupervised manner) with $1, 10, 100$ training examples per class, as well as the full training set. We evaluate our model for both of the inductive biases discussed in Section 5 and Appendix C: the spherical Gaussian (CEBM) and the mixture of Gaussian (GMM-CEBM). We compare our models against the IGEBM Du & Mordatch (2019), a standard VAE with the spherical Gaussian prior, GMM-VAE (Tomczak & Welling, 2018) where the prior is a mixture of Gaussians (GMM), and BIGAN Donahue et al. (2016). IGEBM does not explicitly have latent representations. To resolve this, we extract the features from the last layer of the encoder for IGEBM.

We report the classification accuracy on the test set in Table 1. Overall CEBMs achieve a higher accuracy compared to VAEs in particular for CIFAR-10 and SVHN where the pixel distance is not good measure for similarity. Moreover, we observe that CEBMs outperform IGEBMs, which suggests that the inductive bias in a CEBM can increase performance for downstream tasks. The prformance of BIGANs is competitive, and we suspect that the reason is that, silimiar to CEBMs, BIGANs also do not define a likelihood that measure similarity at the pixel level. We also observe that the CEBM with the GMM inductive bias does not always ourtperform the one with standard Gaussainn inductive bias, wich we supect is due to GMM-VAE being more diffuclt to converge.

8 Conclusion

We introduced CEBMs; a new family of energy-based models that define a joint energy function over both the data and latent variables. This joint distribution factorizes into an intractable energy-based marginal likelihood, which can be trained using standard methods for EBMS, and a tractable posterior, which serves to maps input data to a low-dimensional latent representation. This factorization allows us to directly optimize the marginal likelihood of the data, while at the same time imposing an inductive bias on the latent space. Experimental results for this class of models are encouraging; we observe a closer agreement between unsupervised representations and class labels, which translates into improvements in downstream classification tasks relative to VAE-based baselines.

Acknowledgments

We would like to thank our reviewers for their thoughtful comments. This work was supported by the Intel Corporation, the 3M Corporation, NSF award 1835309, startup funds from Northeastern University, the Air Force Research Laboratory (AFRL), and DARPA.

References

Jyoti Aneja, Alexander Schwing, Jan Kautz, and Arash Vahdat. Ncp-vae: Variational autoencoders with noise contrastive priors. *arXiv preprint arXiv:2010.02917*, 2020.

Arindam Banerjee, Srujana Merugu, Inderjit S. Dhillon, and Joydeep Ghosh. Clustering with Bregman Divergences. *Journal of Machine Learning Research*, 6(58):1705–1749, 2005. ISSN 1533-7928.

Fan Bao, Chongxuan Li, Taufik Xu, Hang Su, Jun Zhu, and Bo Zhang. Bi-level Score Matching for Learning Energy-based Latent Variable Models. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Tong Che, Ruixiang Zhang, Jascha Sohl-Dickstein, Hugo Larochelle, Liam Paull, Yuan Cao, and Yoshua Bengio. Your GAN is Secretly an Energy-based Model and You Should Use Discriminator Driven Latent Sampling. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Ricky TQ Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *Advances in neural information processing systems*, pp. 2610–2620, 2018.

Eric Crawford and Joelle Pineau. Exploiting spatial invariance for scalable unsupervised object tracking. *arXiv preprint arXiv:1911.09033*, 2019a.

Eric Crawford and Joelle Pineau. Spatially invariant unsupervised object detection with convolutional neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 3412–3420, 2019b.

Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016.

Yilun Du and Igor Mordatch. Implicit generation and generalization in energy-based models. *arXiv preprint arXiv:1903.08689*, 2019.

Martin Engelcke, Adam R Kosiorek, Oiwi Parker Jones, and Ingmar Posner. Genesis: Generative scene inference and sampling with object-centric latent representations. *arXiv preprint arXiv:1907.13052*, 2019.

S. M. Ali Eslami, Nicolas Heess, Theophane Weber, Yuval Tassa, David Szepesvari, Koray Kavukcuoglu, and Geoffrey E. Hinton. Attend, infer, repeat: Fast scene understanding with generative models. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pp. 3233–3241, Red Hook, NY, USA, December 2016. Curran Associates Inc. ISBN 978-1-5108-3881-9.

Babak Esmaeili, Hao Wu, Sarthak Jain, Alican Bozkurt, Narayanaswamy Siddharth, Brooks Paige, Dana H Brooks, Jennifer Dy, and Jan-Willem Meent. Structured disentangled representations. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 2525–2534. PMLR, 2019.

Ruiqi Gao, Erik Nijkamp, Diederik P Kingma, Zhen Xu, Andrew M Dai, and Ying Nian Wu. Flow contrastive estimation of energy-based models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7518–7528, 2020.

Will Grathwohl, Kuan-Chieh Wang, Jörn-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. *arXiv preprint arXiv:1912.03263*, 2019.

Will Sussman Grathwohl, Jacob Jin Kelly, Milad Hashemi, Mohammad Norouzi, Kevin Swersky, and David Duvenaud. No {mcmc} for me: Amortized sampling for fast and stable training of energy-based models. In *International Conference on Learning Representations*, 2021.

Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 297–304. JMLR Workshop and Conference Proceedings, 2010.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016.

Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation*, 18(7):1527–1554, May 2006. ISSN 0899-7667. doi: 10.1162/neco.2006.18.7.1527.

Aapo Hyvärinen and Peter Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.

Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International Conference on Machine Learning*, pp. 2649–2658, 2018.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *International Conference on Learning Representations*, 2013.

Adam Kosiorek, Hyunjik Kim, Yee Whye Teh, and Ingmar Posner. Sequential attend, infer, repeat: Generative modelling of moving objects. In *Advances in Neural Information Processing Systems*, pp. 8606–8616, 2018.

Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and F Huang. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

Zhixuan Lin, Yi-Fu Wu, Skand Peri, Bofeng Fu, Jindong Jiang, and Sungjin Ahn. Improving generative imagination in object-centric world models. *arXiv preprint arXiv:2010.02054*, 2020a.

Zhixuan Lin, Yi-Fu Wu, Skand Vishwanath Peri, Weihao Sun, Gautam Singh, Fei Deng, Jindong Jiang, and Sungjin Ahn. SPACE: Unsupervised Object-Oriented Scene Representation via Spatial Attention and Decomposition. *arXiv:2001.02407 [cs, eess, stat]*, March 2020b.

Hao Liu and Pieter Abbeel. Hybrid discriminative-generative training via contrastive learning. *arXiv preprint arXiv:2007.09070*, 2020.

Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

Erik Nijkamp, Mitch Hill, Tian Han, Song-Chun Zhu, and Ying Nian Wu. On the anatomy of mcmc-based maximum likelihood learning of energy-based models. *arXiv*, pp. arXiv–1903, 2019a.

Erik Nijkamp, Mitch Hill, Song-Chun Zhu, and Ying Nian Wu. Learning non-convergent non-persistent short-run mcmc toward energy-based model. In *Advances in Neural Information Processing Systems*, pp. 5232–5242, 2019b.

Bo Pang, Tian Han, Erik Nijkamp, Song-Chun Zhu, and Ying Nian Wu. Learning latent space energy-based prior model. *Advances in Neural Information Processing Systems*, 33, 2020.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In Eric P. Xing and Tony Jebara (eds.), *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pp. 1278–1286, Bejing, China, June 2014. PMLR.

R Tyrrell Rockafellar. *Convex analysis*, volume 36. Princeton university press, 1970.

Ruslan Salakhutdinov and Geoffrey Hinton. Deep boltzmann machines. In *Artificial intelligence and statistics*, pp. 448–455, 2009.

Paul Smolensky. Information processing in dynamical systems: Foundations of harmony theory. Technical report, Colorado Univ at Boulder Dept of Computer Science, 1986.

Yang Song and Diederik P Kingma. How to train your energy-based models. *arXiv preprint arXiv:2101.03288*, 2021.

Yang Song, Sahaj Garg, Jiaxin Shi, and Stefano Ermon. Sliced score matching: A scalable approach to density and score estimation. In *Uncertainty in Artificial Intelligence*, pp. 574–584. PMLR, 2020.

Tijmen Tieleman. Training restricted boltzmann machines using approximations to the likelihood gradient. In *Proceedings of the 25th international conference on Machine learning*, pp. 1064–1071, 2008.

Jakub Tomczak and Max Welling. Vae with a vampprior. In *International Conference on Artificial Intelligence and Statistics*, pp. 1214–1223, 2018.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

Martin J Wainwright and Michael I Jordan. Graphical Models, Exponential Families, and Variational Inference. *Foundations and Trends in Machine Learning*, 1(1–2):1–305, 2008. doi: 10/bpnwrm.

Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 681–688, 2011.

Max Welling, Michal Rosen-zvi, and Geoffrey E Hinton. Exponential family harmoniums with an application to information retrieval. In L. Saul, Y. Weiss, and L. Bottou (eds.), *Advances in Neural Information Processing Systems*, volume 17, pp. 1481–1488. MIT Press, 2004.

Hao Wu, Heiko Zimmermann, Eli Sennesh, Tuan Anh Le, and Jan-Willem van de Meent. Amortized population gibbs samplers with neural sufficient statistics. In *Proceedings of the International Conference on Machine Learning*, pp. 10205–10215, 2020.

Jianwen Xie, Yang Lu, Song-Chun Zhu, and Yingnian Wu. A theory of generative convnet. In *International Conference on Machine Learning*, pp. 2635–2644, 2016.

A CONNECTION TO EXPONENTIAL FAMILY HARMONIUMS

As mentioned in Section 6, there is a long history of incorporating latent variables in EBMs, particularly in the context of restricted Boltzmann machines (RBMs) (Smolensky, 1986; Hinton, 2002), deep belief nets (Hinton et al., 2006), and deep Boltzmann machines (Salakhutdinov & Hinton, 2009). Moreover, the idea of formulating EBMs into the exponential family is also not new; Welling et al. (2004) proposed a new class of models called Exponential Family Harmoniums (EFHs) by extending RBMs into the exponential family. In this Section, we discuss the connection between our approach to these models. Concretely, we show that EFHs can be recovered a special case of CEBMs.

For observed variable x and latent variable z, the energy of an RBM is defined as

$$E_\theta^{\text{RBM}}(x,z) = -\langle x^\top \theta_{xz}, z\rangle - \langle x, \theta_x\rangle - \langle z, \theta_z\rangle, \tag{18}$$

where $\theta_x \in \mathbb{R}^D$, $\theta_z \in \mathbb{R}^K$, and $\theta_{xz} \in \mathbb{R}^{D\times K}$. In RBMs, the conditional distributions $p_\theta(x|z)$ and $p_\theta(z|x)$ are both tractable which means that during contrastive divergence, we can sample $x \sim p_\theta(x)$ using Gibbs sampling.

EFHs extend these models into the exponential family by incorporating the sufficient statistics of x and z in the energy,

$$E_\theta^{\text{EFH}}(x,z) = -\langle t_x(x)^\top \theta_{xz}, t_z(z)\rangle - \langle t_x(x), \theta_x\rangle - \langle t_z(z), \theta_z\rangle, \tag{19}$$

where $t_x(\cdot)$ and $t_z(\cdot)$ are the sufficient statistics for variables x and z respectively. Welling et al. (2004) show that this energy function yields the following conditional distributions:

$$\text{Likelihood} \qquad p_\theta(x|z) = \exp\left\{\langle t_x(x), \tilde{\theta}_x\rangle - A(\tilde{\theta}_x)\right\}, \qquad \tilde{\theta}_x = \theta_x + \theta_{xz} t_z(z), \tag{20}$$

$$\text{Posterior} \qquad p_\theta(z|x) = \exp\left\{\langle t_z(z), \tilde{\theta}_z\rangle - B(\tilde{\theta}_z)\right\}, \qquad \tilde{\theta}_z = \theta_z + \theta_{xz} t_x(x), \tag{21}$$

where $\tilde{\theta}_x$ and $\tilde{\theta}_z$ are the canonical parameters, and $A(\cdot)$ and $B(\cdot)$ are the log normalizer of the models $p_\theta(x|z)$ and $p_\theta(z|x)$ respectively. Given that both conditional distributions are tractable, EFHs have the same advantage as RBMs: We can use a Gibbs sampler for sampling $x \sim p_\theta(x)$.

CEBMs can be considered an extension of EFHs. In Equation 10, we recover the energy function for an EFH by setting

$$t_\theta(x) = [t_x(x)^\top \theta_{xz},\ \langle \theta_x, t_x(x)\rangle], \qquad \eta(z) = [t_z(z),\ 1], \qquad E_\theta(z) = -\langle t_z(z), \theta_z\rangle. \tag{22}$$

Perhaps the most crucial difference between CEBMs and EFHs (and other RBM-based models) is the non-linearity relationship between the observed and latent variables. The non-linearity in $t_\theta(\cdot)$ has the benefit of providing the flexibility to learn more complex structures in the data. This modelling choice however comes with a cost. In CEBMs, while the posterior is still tractable, the likelihood model is not. As a consequence, we lose the ability to use Gibbs sampling to sample $x \sim p_\theta(x)$. However, given that our motivation here is not to generate high quality samples at test time but to learn good representations representations, we believe giving up the ability to easily sample x in order to learn more complex structures while keeping the posterior tractable is an appropriate trade-off.

B DERIVATION OF PRIOR AND LIKELIHOOD IN A CEBM

B.1 PRIOR

$$p_{\theta,\lambda}(z) = \int dx\ \frac{1}{Z_{\theta,\lambda}} \exp\{-E_{\theta,\lambda}(x,z)\} \tag{23}$$

$$= \frac{1}{Z_{\theta,\lambda}} \int dx\ \exp\{-E_{\theta,\lambda}(x,z)\} \tag{24}$$

$$= \frac{1}{Z_{\theta,\lambda}} \int dx\ \exp\{\langle t_\theta(x), \eta(z)\rangle - E_\lambda(z)\} \tag{25}$$

$$= \frac{\exp\{-E_\lambda(z)\}}{Z_{\theta,\lambda}} \int dx\ \exp\{\langle t_\theta(x), \eta(z)\rangle\} \tag{26}$$

Energy Type	Model	Energy
$E_\theta(x)$	IGEBM (Du & Mordatch, 2019)	$f_\theta(x)$
$E_\theta(x, y)$	JEM (Grathwohl et al., 2019) HDGE (Liu & Abbeel, 2020)	$-f_\theta(x)[y]$
$E_\theta(x, z)$	RBM (Smolensky, 1986) EFH (Welling et al., 2004) VAE (Kingma & Welling, 2013) GAN (Che et al., 2020) CEBM (this paper)	$-\langle x^\top \theta_{x,z}, z\rangle - \langle x, \theta_x\rangle - \langle z, \theta_z\rangle$ $-\langle t(x)^\top \theta_{x,z}, t(z)\rangle - \langle t(x), \theta_x\rangle - \langle t(z), \theta_z\rangle$ $-\langle x, \mu_\theta(z)\rangle + A(\eta_\theta(z)) + E(z)$ $D_\theta(x) + E(z)$ $-\langle t_\theta(x), \eta(z)\rangle + E(z)$
$E_\theta(x, y, z)$	GMM-VAE (Tomczak & Welling, 2018) GMM-CEBM (this paper)	$-\langle x, \mu_\theta(z, y)\rangle + A(\eta_\theta(z, y)) + E(z, y)$ $-\langle t_\theta(x), \eta(y, z)\rangle + E(y, z)$

Table 2: Comparison of energies in generative models. The functions $f_\theta(\cdot)$, $\eta_\theta(\cdot)$, and $t_\theta(\cdot)$ are typically deep neural networks (DNNs). In EBMs defined on only the data space (type $E_\theta(x)$) such as IGEBM, the DNN outputs a scalar value $f_\theta(x) : \mathbb{R}^D \to \mathbb{R}$. In EBMs defined on the data space as well as labels (type $E_\theta(x, y)$) such as JEM, the DNN outputs a vector of length L corresponding to the number of classes $f_\theta(x) : \mathbb{R}^D \to \mathbb{R}^L$. In GAN, $D_\theta(x)$ refers to the discriminator.

B.2 Likelihood

$$
\begin{aligned}
p_{\theta,\lambda}(x|z) &= \frac{p_{\theta,\lambda}(x, z)}{p_{\theta,\lambda}(z)} && (27)\\
&= \frac{\frac{1}{Z_{\theta,\lambda}} \exp\{-E_{\theta,\lambda}(x, z)\}}{\frac{\exp\{-E_\lambda(z)\}}{Z_{\theta,\lambda}} \int dx\ \exp\{\langle t_\theta(x), \eta(z)\rangle\}} && (28)\\
&= \frac{\frac{\exp\{-E_\lambda(z)\}}{Z_{\theta,\lambda}} \exp\{\langle t_\theta(x), \eta(z)\rangle\}}{\frac{\exp\{-E_\lambda(z)\}}{Z_{\theta,\lambda}} \int dx\ \exp\{\langle t_\theta(x), \eta(z)\rangle\}} && (29)\\
&= \frac{\exp\{\langle t_\theta(x), \eta(z)\rangle\}}{\int dx\ \exp\{\langle t_\theta(x), \eta(z)\rangle\}} && (30)
\end{aligned}
$$

C Two Special Cases for Inductive Biases

In this work we will consider two special cases for inductive bias:

1. Spherical Gaussian. As a bias that is analogous to the standard prior in VAEs, we consider a spherical Gaussian with fixed hyperparameters $(\mu, \sigma) = (0, 1)$ for each dimension of $z \in \mathbb{R}^K$,

$$E_\lambda(z) = -\sum_k \big(\langle \eta(z_k), \lambda\rangle - B(\lambda)\big).$$

Each term has sufficient statistics $\eta(z_k) = (z_k, z_k^2)$ and natural parameters

$$\lambda = \left(\frac{\mu}{\sigma^2}, -\frac{1}{2\sigma^2}\right) = \left(0, -\frac{1}{2}\right).$$

The marginal likelihood of the CEBM is then

$$p_{\theta,\lambda}(x) = \frac{1}{Z_{\theta,\lambda}} \exp\Big\{\sum_k \big(B(\tilde{\lambda}_{\theta,k}(x)) - B(\lambda)\big)\Big\},$$

where $\tilde{\lambda}_{\theta,k}(x) = \lambda + t_{\theta,k}(x)$ and the log normalizer is

$$B(\tilde{\lambda}_k) = -\frac{\tilde{\lambda}_{k,1}^2}{4\tilde{\lambda}_{k,2}} - \frac{1}{2}\log(-2\tilde{\lambda}_{k,2}).$$

2. Mixture of Gaussians. In our experiments, we will consider datasets that are normally used for classification. These datasets, by design, exhibit multimodal structure that we would like to see reflected in the learned representation.In order to design a model that is amenable to uncovering this structure, we will extend the energy function in Equation 10 to contain a mixture component y

$$E_{\theta,\lambda}(x, y, z) = -\langle t_\theta(x), \eta(y, z)\rangle + E_\lambda(y, z).$$

As an inductive bias, we will consider a bias in the form of a mixture of L Gaussians,

$$E_\lambda(y, z) = -\sum_{k,l} I[y = l]\big(\langle \eta(z_k), \lambda_{l,k}\rangle - B(\lambda_{l,k})\big).$$

Here $z \in \mathbb{R}^K$ is a vector of features and $y \in \{1, \dots, L\}$ is a categorical assignment variable. The bias for each component l is a spherical Gaussian with hyperparameters $\lambda_{l,k}$ for each dimension k. Again, using the notation $\tilde{\lambda}_{\theta,l,k} = \lambda_{l,k} + t_{\theta,l,k}(x)$ to refer to the posterior parameters, then we obtain an energy

$$E_{\theta,\lambda}(x, y, z) = -\sum_{k,l} I[y = l]\big(\langle \eta(z_k), \tilde{\lambda}_{l,k}\rangle - B(\lambda_{l,k})\big).$$

We then define a joint probability over data x and the assignment y in terms the log normalizer $B(\cdot)$,

$$p_{\theta,\lambda}(x, y) = \frac{1}{Z_{\theta,\lambda}} \exp\Big\{\sum_{k,l} I[y = l]\big(B(\tilde{\lambda}_{l,k}) - B(\lambda_{l,k})\big)\Big\},$$

which then allows us to compute the marginal $p_{\theta,\lambda}$ by summing over y. We optimize this marginal with respect hyperaparameters $\lambda_{l,k}$ as well as the weights θ.

D Background on Energy-Based Models

An EBM (LeCun et al., 2006) defines a probability density for $x \in \mathbb{R}^D$ via the Gibbs-Boltzmann distribution

$$p_\theta(x) = \frac{\exp\{-E_\theta(x)\}}{Z_\theta}, \qquad Z_\theta = \int dx\ \exp\{-E_\theta(x)\}.$$

The function $E_\theta : \mathbb{R}^D \to \mathbb{R}$ is called the energy function which maps each configuration to a scalar value, the energy of the configuration. This type of model is widely used in statistical physics, for example in Ising models. The distribution can only be evaluated up to an unknown constant of proportionality, since computing the normalizing constant Z_θ (also known as the partition function) requires an intractable integral with respect to all possible inputs x.

Our goal is to learn a model $p_\theta(x)$ that is close to the true data distribution $p_{\text{data}}(x)$. A common approach for achieving this objective is to minimize the Kullback-Leibler divergence between the data distribution and the model, which is equivalent to maximizing the expected log-likelihood

$$\mathcal{L}(\theta) = \mathbb{E}_{p_{\text{data}}(x)}[\log p_\theta(x)] = \mathbb{E}_{p_{\text{data}}(x)}[-E_\theta(x)] - \log Z_\theta. \tag{31}$$

The key difficulty when performing maximum likelihood estimation is that computing the gradient of $\log Z_\theta$ is intractable. A common strategy is to express this gradient as an expectation with respect to $p_\theta(x)$,

$$\nabla \log Z_\theta = \mathbb{E}_{p_\theta(x')}\left[-\nabla_\theta E_\theta(x')\right], \tag{32}$$

which means that the gradient of $\mathcal{L}(\theta)$ has the form:

$$\nabla_\theta \mathcal{L}(\theta) = -\mathbb{E}_{p_{\text{data}}(x)}[\nabla_\theta E_\theta(x)] + \mathbb{E}_{p_\theta(x')}[\nabla_\theta E_\theta(x')].$$

This corresponds to *maximizing* the probability of samples $x \sim p_{\text{data}}(x)$ from the data distribution and *minimizing* the probability of samples $x' \sim p_\theta(x')$ from the learned model.

Contrastive divergence methods Hinton (2002) compute a Monte Carlo estimate of this gradient. Estimating this gradient however requires samples from the model $x' \sim p_\theta(x')$, whereas this density is intractable. A common method for generating samples from EBMs is Stochastic Gradient Langevin

Dynamics (SGLD, Welling & Teh (2011)), which initializes a sample $x'_0 \sim p_0(x')$ and performs a sequence of gradient updates with additional injected noise ϵ,

$$x'_{i+1} = x'_i - \frac{\alpha}{2}\frac{\partial E_\theta(x')}{\partial x'} + \epsilon\,, \qquad\qquad \epsilon \sim N(0, \alpha). \tag{33}$$

SGLD is motivated as a discretization of a stochastic differential equation whose stationary distribution is equal to the posterior. It is correct in the limit $i \to \infty$ and $\alpha \to 0$, but in practice will have a bias.

The initialization x'_0 is crucial because it determines the number of steps needed to converge to a high-quality sample. For this reason, EBMs are commonly trained Nijkamp et al. (2019a); Du & Mordatch (2019); Grathwohl et al. (2019) using persistent contrastive divergence (PCD, Tieleman (2008)), which initializes some samples from a replay buffer $\mathcal{B}$ of previously generated samples.

E Training CEBMs

Hyperparameters. In CEBMs and VAEs, we choose the dimension of latent variables to be 128. For CEBMS, We found that the optimization becomes difficult with smaller dimensions. We train our models using 60 SGLD steps where we initialize samples from the replay buffer with 0.95 probability, and initialize from uniform noise with 0.05 probability. We train all the models with 90k gradient steps, batch size 128, Adam optimizer with learning rate 1e-4. When doing PCD, we used a reply buffer of size 5000. We set the α in the SGLD teps to be 0.075. Similar to Du & Mordatch (2019), we found it useful to add some noise to the image before encoding. In our experiments, we used Gaussian noise with $\sigma^2 = 0.03$. We used 50 GMM components for GMM-VAE and 10 GMM components for GMM-CEBM.

Training Stability. As observed in previous work Du & Mordatch (2019); Grathwohl et al. (2019), training EBMs is challenging and often requires a thorough hyperparameters search. We found that the choices of activation function, learning rate, number of SGLD steps, and L2 regularization (proposed by Du & Mordatch (2019)) will all affect training stability. Models regularly diverge during training, and it is difficult to perform diagnostics given that $\log p_{\theta,\lambda}(x)$ cannot be computed. As suggested by Nijkamp et al. (2019a), we found checking the difference in energy between data and model samples can help to verify training stability. In general we also observed a trade-off between sample quality and the predictive power of latent variables in our experiments. We leave investigation of the source of this trade-off to future work, but we suspect that this is because SGLD has more difficulty converging when the latent space is more disjoint.

Nijkamp et al. (2019a;b) performed a comprehensive analysis of convergence of PCD in recent EBMs for images, where they study a variety of factors such as MCMC chain initialization, network, and optimizer. They identify that a key factor for diagnosing these models is the difference between the energy of positive and negative samples. Many of these findings were helpful during the training and evaluation of EBMs in our work.

F Out-of-Distribution Detection

EBMs have formed the basis for encouraging results in out-of-distribution (OOD) detection Du & Mordatch (2019); Grathwohl et al. (2019). While not our focus in this paper, OOD detection is an additional benchmark that helps evaluate whether a learned model accurately characterizes the data distribution. In Table 3, we report results in terms of two metrics. The first is the area under the receiver-operator curve (AUROC) when thresholding the log marginal $\log p_{\theta,\lambda}(x)$. The second is the gradient-based score function proposed by Grathwohl et al. (2019). CEBMs results for OOD detection in most cases improve upon VAE and IGEBM baselines.

G Visulaization of Latent Space

Table 3: AUROC scores in OOD Detection. We use $\log p_\theta(\boldsymbol{x})$ and $\|\nabla_{\boldsymbol{x}} \log p_\theta(\boldsymbol{x})\|$ as score functions.The left block shows results of the models trained on F-MNIST and tested on MNIST, E-MNIST, Constant (C); The right block shows results of the models trained on CIFAR-10 and tested on SVHN, Texture and Constant (C).

	Fashion-MNIST						CIFAR-10					
	$\log p_\theta(\boldsymbol{x})$			$\|\nabla_{\boldsymbol{x}} \log p_\theta(\boldsymbol{x})\|$			$\log p_\theta(\boldsymbol{x})$			$\|\nabla_{\boldsymbol{x}} \log p_\theta(\boldsymbol{x})\|$		
	MNIST	E-MNIST	C	MNIST	E-MNIST	C	SVHN	Texture	C	SVHN	Texture	C
VAE	.50	.39	.09	.61	.57	.01	.42	**.58**	.41	.38	**.51**	.37
IGEBM	.35	.36	.90	.78	.82	.96	.45	.31	.64	.33	.17	**.62**
CEBM	.37	.34	.90	**.82**	**.89**	**.98**	.47	.32	**.66**	.31	.17	.54
GMM-CEBM	**.56**	**.56**	**.92**	.56	.80	.95	**.55**	.30	.62	**.40**	.23	**.62**

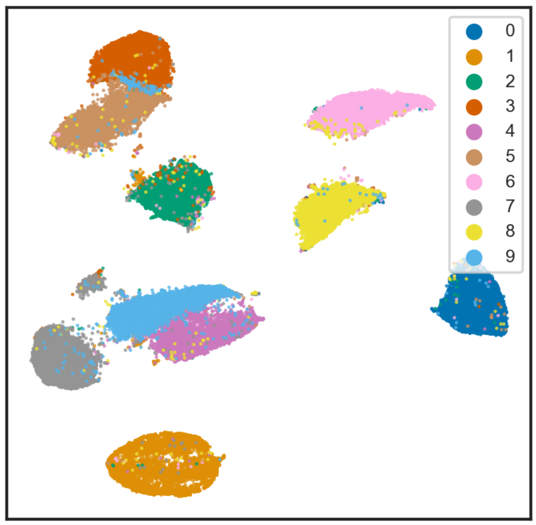
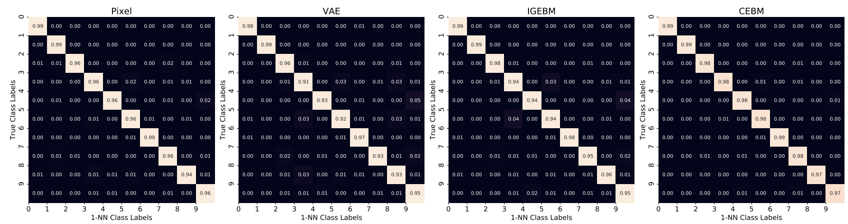


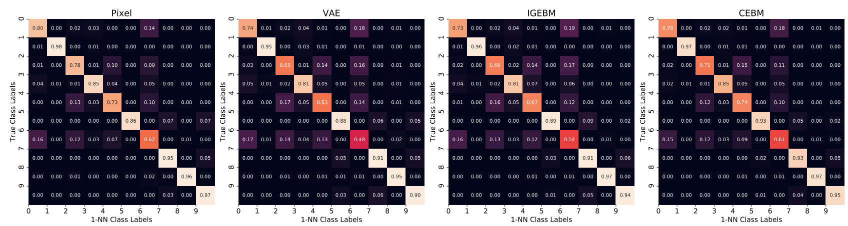
Figure 4: Latent space of a CEBM (the inferred means) visualized with UMAP McInnes et al. (2018) trained on MNIST.

H Confusion Matrices for 1-NN Classification

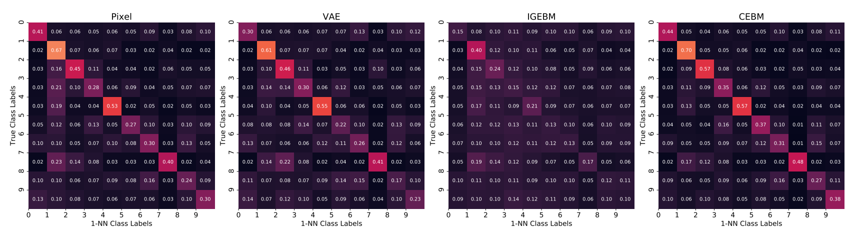
We perform 1-nearest-neighbor classification task for MNIST, Fashion-MNIST, SVHN, CIFAR10. We compute the L2 distance in the latent space of VAE, IGEBM and CEBM, and also in pixel space. We visualize the confusion matrices



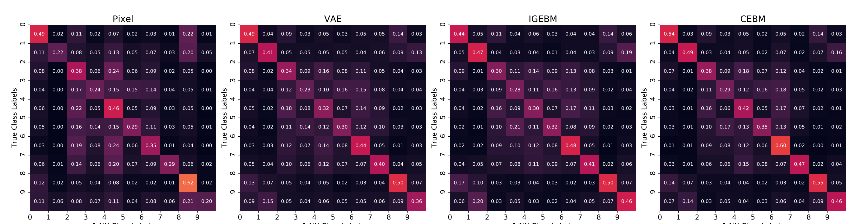
(a) MNIST



(b) Fashion-MNIST



(c) SVHN



(d) CIFAR10

I Model Architectures

CEBMs employ an encoder network $t_\theta(x)$ in the form of 4-layer CNN (which is proposed by Nijkamp et al. (2019a)), followed by an MLP output layer. We choose the dimension of latent variables to be 128 to be in CEBMs. We use the same architecture for the encoder of a VAE and the encoder of a BIGAN. Table 4, Table 5, Table 6, and Table 7 show the architectures used for CEBM, IGEBM, VAE and BIGAN, respectively.

Table 4: Architecture of CEBM and GMM-CEBM

(a) MNIST and Fashion-MNIST.

Encoder
Input $28 \times 28 \times 1$ images
3×3 conv. 64 stride 1. padding 1. Swish.
4×4 conv. 64 stride 2. padding 1. Swish.
4×4 conv. 32 stride 2. padding 1. Swish.
4×4 conv. 32 stride 2. padding 1. Swish.
FC. 128 Swish.
FC. 2×128

(b) CIFAR10 and SVHN.

Encoder
Input $32 \times 32 \times 3$ images
3×3 conv. 64 stride 1. padding 1. Swish.
4×4 conv. 128 stride 2. padding 1. Swish.
4×4 conv. 256 stride 2. padding 1. Swish.
4×4 conv. 512 stride 2. padding 1. Swish.
FC. 1024 Swish.
FC. 2×128

Table 5: Architecture of IGEBM

(a) MNIST and Fashion-MNIST.

Encoder
Input $28 \times 28 \times 1$ images
3×3 conv. 64 stride 1. padding 1. Swish.
4×4 conv. 64 stride 2. padding 1. Swish.
4×4 conv. 32 stride 2. padding 1. Swish.
4×4 conv. 32 stride 2. padding 1. Swish.
FC. 128 Swish.
FC. 128 Swish. FC. 1

(b) CIFAR10 and SVHN.

Encoder
Input $32 \times 32 \times 3$ images
3×3 conv. 64 stride 1. padding 1. Swish.
4×4 conv. 128 stride 2. padding 1. Swish.
4×4 conv. 256 stride 2. padding 1. Swish.
4×4 conv. 512 stride 2. padding 1. Swish.
FC. 1024 Swish
FC. 128 Swish. FC. 1

Table 6: Architecture of VAE and GMM-VAE

(a) MNIST and Fashion-MNIST.

Encoder	**Decoder**
Input $28 \times 28 \times 1$ images	Input $z \in \mathbb{R}^{128}$ latent variables
3×3 conv. 64 stride 1. padding 1. ReLU.	FC. 128 ReLU. FC. $3 \times 3 \times 32$ ReLU.
4×4 conv. 64 stride 2. padding 1. ReLU.	4×4 upconv. 32 stride 2. padding 1. ReLU.
4×4 conv. 32 stride 2. padding 1. ReLU.	4×4 upconv. 64 stride 2. padding 1. ReLU.
4×4 conv. 32 stride 2. padding 1. ReLU.	4×4 upconv. 64 stride 2. padding 0. ReLU.
FC. 128 ReLU. FC. 2×128.	3×3 upconv. 1 stride 1. padding 0

(b) CIFAR10 and SVHN.

Encoder	**Decoder**
Input $32 \times 32 \times 3$ images	Input $z \in \mathbb{R}^{128}$ latent variables
3×3 conv. 64 stride 1. padding 1. ReLU.	FC. 128 ReLU. FC. $4 \times 4 \times 512$ ReLU.
4×4 conv. 128 stride 2. padding 1. ReLU.	4×4 upconv. 32 stride 2. padding 1. ReLU.
4×4 conv. 256 stride 2. padding 1. ReLU.	4×4 upconv. 64 stride 2. padding 1. ReLU.
4×4 conv. 512 stride 2. padding 1. ReLU.	3×3 upconv. 64 stride 2. padding 1. ReLU.
FC. 1024 ReLU. FC. 2×128.	3×3 upconv. 1 stride 1. padding 1

Table 7: Architecture of BIGAN for MNIST and Fashion-MNIST.

(a) MNIST and Fashion-MNIST.

Discriminator
Input $28 \times 28 \times 1$ images
3×3 conv. 64 stride 1. padding 1. BN. LeakyReLU.
4×4 conv. 64 stride 2. padding 1. BN. LeakyReLU.
4×4 conv. 32 stride 2. padding 1. BN. LeakyReLU.
4×4 conv. 32 stride 2. padding 1. BN. LeakyReLU.
FC. 128 LeakyReLU.
256. FC 128 LeakyReLU. FC. 1. Sigmoid.

Generator	Encoder
Input $z \in \mathbb{R}^{128}$ latent variables	Input $28 \times 28 \times 1$ images
4×4 upconv. 64 stride 1. padding 1. BN. ReLU.	3×3 conv. 64 stride 1. padding 1. BN. LeakyReLU.
4×4 upconv. 64 stride 2. padding 1. BN. ReLU.	4×4 conv. 64 stride 2. padding 1. BN. LeakyReLU.
3×3 upconv. 32 stride 2. padding 1. BN. ReLU.	4×4 conv. 32 stride 2. padding 1. BN. LeakyReLU.
4×4 upconv. 32 stride 2. padding 1. BN. ReLU.	4×4 conv. 32 stride 2. padding 1. BN. LeakyReLU.
4×4 upconv. 1 stride 2. padding 1. Tanh.	FC. 128 LeakyReLU. FC. 2×128.

(b) CIFAR10 and SVHN.

Discriminator
Input $28 \times 28 \times 1$ images
3×3 conv. 64 stride 1. padding 1. BN. LeakyReLU.
4×4 conv. 128 stride 2. padding 1. BN. LeakyReLU.
4×4 conv. 256 stride 2. padding 1. BN. LeakyReLU.
4×4 conv. 512 stride 2. padding 1. BN. LeakyReLU.
FC. 128 LeakyReLU.
256 FC 128 LeakyReLU. FC. 1. Sigmoid.

Generator	Encoder
Input $z \in \mathbb{R}^{128}$ latent variables	Input $28 \times 28 \times 1$ images
4×4 upconv. 512 stride 2. padding 1. BN. ReLU.	3×3 conv. 64 stride 1. padding 1. BN. LeakyReLU.
4×4 upconv. 256 stride 2. padding 1. BN. ReLU.	4×4 conv. 128 stride 2. padding 1. BN. LeakyReLU.
4×4 upconv. 128 stride 2. padding 1. BN. ReLU.	4×4 conv. 256 stride 2. padding 1. BN. LeakyReLU.
4×4 upconv. 64 stride 2. padding 1. BN. ReLU.	4×4 conv. 512 stride 2. padding 1. BN. LeakyReLU.
4×4 upconv. 3 stride 2. padding 1. Tanh.	FC. 128 LeakyReLU. FC 2×128.