3rd Symposium on Advances in Approximate Bayesian Inference, 2020 1–14

Conjugate Energy-Based Models

Anonymous Authors
Anonymous Institution

Abstract

Energy-Based Models (EBMs) define an unnormalized distribution in terms of an energy function. In this paper, we develop a new class of conjugate EBMs that represent a joint distribution over data and latent variables. The joint distribution of these models decomposes into an intractable distribution over data and a tractable posterior over latent variables. Conjugate EBMs have similar use cases as variational autoencoders, in the sense that they learn an unsupervised mapping from data to latent variables, but omit a generator, 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-distribution 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. One core use case of deep generative models is to learn representations for high-dimensional data in an unsupervised manner. A commonly used class of methods is Variational Autoencoders (VAEs, Kingma and Welling (2013); Rezende et al. (2014). Since VAEs learn an explicit mapping in the form of an encoder that approximates the posterior distribution over latent variables, it is natural to define the prior to add useful inductive biases. For example, using conditional independence in the definition of a prior is widely used to learn disentangled representations (Higgins et al., 2016; Kim and Mnih, 2018; Esmaeili et al., 2019). More generally, we can define priors that reflect structure of the underlying model; Examples include priors that represent objects in an image (Eslami et al., 2016; Lin et al., 2020b; Engelcke et al., 2019; Crawford and Pineau, 2019b), or moving objects in video (Crawford and Pineau, 2019a; Kosiorek et al., 2018; Wu et al., 2020; Lin et al., 2020a).

VAEs define a likelihood model that is typically a neural generator that is optimized to accurately reconstruct the training examples, which will pose challenges on learning semantically meaningful representations. As an example, consider the problem of scene understanding. Since a VAE is trained to reconstruct all the pixels in an image, it will learn a pixel-perfect representation of the background of the scene, which can be irrelevant features in terms of capture representative objects in the foreground.

In this paper, we consider energy-based models (EBMs, Du and Mordatch (2019)) as such an alternative to deep generative models. EBMs define an unnormalized density in terms of an energy function, which is a discriminative network rather than a generative network. Since EBMs do not attempt to reconstruct the data, we hope to learn more flexible densities that represents factors of variation at varying levels of precision.

© A. Authors.

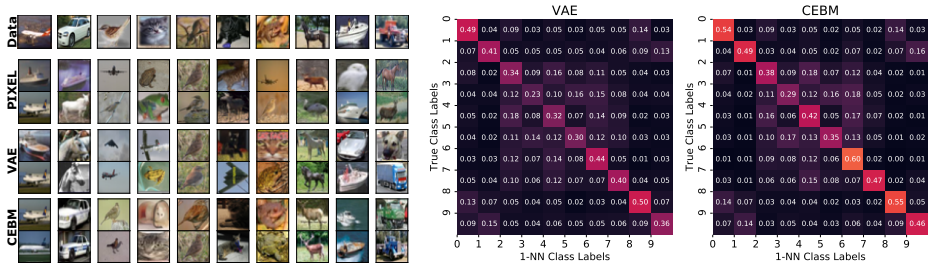

Figure 1: (*Left*) Samples from CIFAR-10 along with the top 2-nearest-neighbors in pixel space, the latent space of a VAE, and the latent space of a CEBM. (*Right*) Confusion matrices of 1-nearest-neighbor classification on CIFAR-10 based on L2 distance in the latent space. On average, CEBM representations more closely align with class labels compared to VAE.

More specifically, we propose Conjugate EBMs (CEBMs), a new family of deep probabilistic models that combine a prior over latent variables with an energy-based likelihood. In CEBMs, the energy function defines a vector of sufficient statistics in a neural exponential family. While the normalizer of this family is intractable, we can nonetheless compute the posterior in closed form when we pair the likelihood with an appropriate conjugate prior. As a result, the energy function in a CEBM fully determines both the marginal likelihood and the encoder, hereby side-stepping the need for a generator.

In our experiments, we evaluate the representations learned by CEBM using class labels as a proxy for the primary factors of variation in a dataset. We show that CEBMs learn a notion of similarity that aligns more closely with class labels in terms of the nearest neighbors in latent space (Figure 1). Moreover, we show that the representations learned by CEBMs (in an unsupervised manner) can achieve a competitive performance in few-shot learning and out-of-distribution detection tasks.

2. Background

2.1. Energy-Based Models

An EBM (LeCun et al., 2006) on a space of configuration $x \in \mathbb{R}^D$ defines a Gibbs-Boltzmann distribution as $p_\theta(x) = \exp\{-E_\theta(x)\}/Z_\theta$. $E_\theta(x) : \mathbb{R}^D \to \mathbb{R}$ is the energy function and maps each configuration to a scalar value. The distribution $p_\theta(x)$ can only be evaluated up to an unknown constant of proportionality since the normalizing constant $Z_\theta = \int dx \exp\{-E_\theta(x)\}$ is intractable in general. To train an energy-based model, we maximize the expected log-likelihood as

$$\mathcal{L}_\theta = \mathbb{E}_{p_{\text{data}}(x)}[\log p_\theta(x)] = \mathbb{E}_{p_{\text{data}}(x)}[-E_\theta(x)] - \log Z_\theta. \tag{1}$$

We can express the gradient of $\mathcal{L}_\theta$ as $\nabla_\theta \mathcal{L}_\theta = -\mathbb{E}_{p_{\text{data}}(x)}[\nabla_\theta E_\theta(x)] + \mathbb{E}_{p_\theta(x')}[\nabla_\theta E_\theta(x')]$.

Estimating this term however requires samples from the model itself which is not straight forward. A common method for sampling from the model $x' \sim p_\theta(x')$ is Stochastic Gradient Langevin Dynamics (SGLD) (Welling and Teh, 2011), which initializes a sample $x'_0 \sim p_0(x')$

and then performs a sequence of gradient updates with additional injected noise ϵ

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

Recent work on EBMs has shown that it is possible to use deep neural network as energy functions, and that it is possible to use Langevin methods to generate realistic samples from learned models (Nijkamp et al., 2019a,b; Du and Mordatch, 2019; Xie et al., 2016). Work by Grathwohl et al. (2019) interprets classifiers through the lens of energy-based models. Liu and Abbeel (2020) propose a similar model where the objective combines a discriminative conditional and a generative conditional conditional distribution.

2.2. Conjugate Exponential Families

An exponential family is a set of distributions whose probability density function or probability mass function can be expressed in the form

$$p(x \mid \eta) = h(x) \exp\big\{t(x)^\top \eta - A(\eta)\big\},$$

where $h(\cdot)$ is a vector of natural parameters η, a vector of sufficient statistics $t(\cdot)$, and a log normalizer $A(\cdot)$. If a likelihood belongs to an exponential family, then there exists a conjugate prior with the form $p(\eta \mid \lambda) = h_0(\eta) \exp\big\{t_0(\eta)^\top \lambda - A_0(\lambda)\big\}$. A prior is conjugate to a likelihood when its vector of sufficient statistics comprises the natural parameters and the log-normalizer of the likelihood $t_0(\eta) = \big[\eta, -A(\eta)\big]$, $\lambda = \big[\lambda_1, \lambda_2\big]$. The convenient property of conjugate exponential families is that both the marginal likelihood $p(x \mid \lambda)$ and the posterior $p(\eta \mid x, \lambda)$ are tractable. The reason is that the joint probability has the form

$$p(x, \eta \mid \lambda) = p(x \mid \eta)\, p(\eta \mid \lambda) = \exp\big\{\eta^\top\big(\lambda_1 + t(x)\big) - A(\eta)(\lambda_2 + 1) - A_0(\lambda)\big\}.$$

If we substitute $\tilde{\lambda}_1 = \lambda_1 + t(x)$ and $\tilde{\lambda}_2 = \lambda_2 + 1$, we can equivalently factorize this joint as

$$p(x, \eta \mid \lambda) = p(\eta \mid x, \lambda)\, p(x \mid \lambda) = p(\eta \mid \tilde{\lambda})\ \exp\big\{A_0(\tilde{\lambda}) - A_0(\lambda)\big\}. \tag{2}$$

This shows that the posterior is in the same exponential family as the prior, and that we can express the marginal likelihood using the log normalizer $A_0(\cdot)$

$$p(\eta \mid x, \lambda) = p(\eta \mid \tilde{\lambda}), \qquad p(x \mid \lambda) = \exp\big\{A_0(\tilde{\lambda}) - A_0(\lambda)\big\}. \tag{3}$$

3. Conjugate Energy-Based Models

In this paper we are interested in learning a probabilistic model that defines a joint distribution $p_\theta(x, z)$ over high-dimensional data $x \in \mathbb{R}^D$ and a lower-dimensional set of latent variables $z \in \mathbb{R}^K$. Because most work on deep generative models has focused on images, we will restrict ourselves to this data modality.

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 models that learn a generator network. Concretely, we propose to consider models of the form

$$p_\theta(x, z) = \frac{1}{Z_\theta}\ \exp\big\{-E_\theta(x, z)\big\}, \qquad E_\theta(x, z) = -t_\theta(x)^\top \eta_\theta(z) - b_\theta(z), \tag{4}$$

where θ are the parameters of our model and Z_θ is the partition function. In the energy function, $\eta_\theta : \mathbb{R}^K \to \mathbb{R}^H$ plays a role similar to that of a generator, which maps latent variables to a vector of natural parameters of dimension H. The function $t_\theta : \mathbb{R}^D \to \mathbb{R}^H$ plays the role of an encoder, which maps data to a vector of sufficient statistics. The function $b_\theta : \mathbb{R}^K \to \mathbb{R}$ serves as an inductive bias that plays a role analogous to the prior. The generative model in a VAE can be recovered as a special case (Appendix B). We now define the bias $b_\theta(z) = \log p_\theta(z \mid \lambda)$ in form of a tractable exponential family with $\eta(z)$

$$b_\theta(z) = \eta_\theta(z)^\top \lambda - A(\lambda). \tag{5}$$

We can then express the energy function as

$$E_\theta(x, z) = -t_\theta(x)^\top \eta_\theta(z) - b_\theta(z) = -\big(\lambda + t_\theta(x)\big)^\top \eta_\theta(z) + A(\lambda). \tag{6}$$

This form of the energy function has a very convenient property: It corresponds to a model $p_\theta(x, z)$ in which the posterior $p_\theta(z \mid x)$ is tractable. To see this, we can make a substitution $\tilde{\lambda} = \lambda + t_\theta(x)$ analogous to the one in Equation 2, which allows us to express the energy as

$$E_\theta(x, z) = -\big(\eta_\theta(z)^\top \tilde{\lambda} - A(\tilde{\lambda})\big) - \big(A(\tilde{\lambda}) - A(\lambda)\big)$$

We now see that we can factorize the he corresponding density $p_\theta(x, z)$ as

$$p_\theta(x, z \mid \lambda) = p_\theta(x \mid \lambda)\, p_\theta(z \mid x, \lambda). \tag{7}$$

which yields a posterior and a marginal that are analogous to the distributions in Equation 3

$$p_\theta(z \mid x, \lambda) = p(z \mid \lambda + t_\theta(x)), \quad p_\theta(x \mid \lambda) = \frac{1}{Z_\theta} \exp\big\{A\big(\lambda + t_\theta(x)\big) - A\big(\lambda\big)\big\}. \tag{8}$$

The posterior $p_\theta(z \mid x, \lambda)$ 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).

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 $(\lambda + t_\theta(x))^\top \eta(z)$ between data and latent variables in a low-dimensional feature space.

In addition to having a tractable posterior, CEBMs have the convenient property that the marginal likelihood $p_\theta(x \mid \lambda)$ itself can be expressed as an energy-based model that is defined in terms of the log normalizer $A(\cdot)$ and the encoder network $t_\theta(x)$. This means that we can train CEBMs by maximizing the marginal likelihood $p_\theta(x \mid \lambda)$ using persistent contrastive divergence (Tieleman, 2008) , in the same way that other EBMs are trained (Nijkamp et al., 2019a; Du and Mordatch, 2019; Grathwohl et al., 2019).

4. Inductive Biases

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

$$p_\theta(z) = \frac{\exp\{-b_\theta(z)\}}{Z_\theta} \int dx\ \exp\{t_\theta(x)^\top \eta(z)\}.$$

Figure 2: CEBM Samples of MNIST, F-MNIST, SVHN and CIFAR-10.

Models	MNIST				Fashion-MNIST				CIFAR-10				SVHN			
	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
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**	**52**	12	**25**	**48**	**70**
CEBMM	**67**	**91**	**97**	**98**	**52**	**71**	**80**	**85**	16	28	**42**	51	10	17	39	60

Table 1: Few-Shot classification accuracy. We pre-train 5 unsupervised models (rows) on MNIST, Fashion-MNIST, CIFAR10, SVHN. Then we train logistic classifiers using 1, 10, 100 examples per class (i.e. shots) and the full training dataset. We report the testing classification accuracy, where CEBM outperforms.

In principle the bias 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. In this paper, we will constrain ourselves to a Spherical Gaussian and a Mixture of Gaussians. We provide derivations for both cases in Appendix A.

5. 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 train using both inductive biases and will refer to them as CEBM (Spherical Gaussian) and CEBMM (Mixutre of Gaussian) for the rest of Section. See Appendix C and D for architecture and training details.

5.1. Samples and Latent Space

We begin with a qualitative evaluation by visualizing samples. Figures 2 shows samples from CEBMs with uniform noise initialization an 500 SGLD steps.

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. Figure 1 shows that 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. Quantitatively, in Appendix F, we observe a stronger alignment between classes and the latent representation in CEBMs, which is reflected in higher numbers on the diagonal of the matrix.

	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	9	61	57	1	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
CEBMM	**56**	**56**	**92**	56	80	95	**55**	30	62	**40**	23	**62**

Table 2: 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).

5.2. Few-shot classification

To evaluate performance in settings where few labels are available, we train a logistic classifier using $1, 10, 100$ examples per class, as well as the full training dataset. We compare CEBMs against the IGEBM Du and Mordatch (2019), a standard VAE with the spherical Gaussian prior, and the GMM-VAE Tomczak and Welling (2018) where the prior is a mixture of Gaussians. As discussed in Section 2, IGEBM does not have an explicit representation. In order to compare against IGEBM, we remove the last layer (which outputs the energy) and use the resulting intermediate representation as the latent code.

We report the classification accuracy on the test set in Table 5. We can see that that CEBMs overall 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 IGEBM which suggest that the inductive biases in CEBMs can lead to increased performance in downstream tasks.

5.3. Out-of-Distribution Detection

EBMs have formed the basis for encouraging results in out-of-distribution (OOD) detection Du and Mordatch (2019); Grathwohl et al. (2019). In Table 2, we report the area under the receiver-operator curve (AUROC) using two score functions: $\log p_\theta(x)$ and a gradient-based function proposed by Grathwohl et al. (2019). CEBMs results for OOD detection in most cases improve upon VAE and IGEBM baselines.

6. Conclusion

We introduced CEBMs; a new family of energy-based models that define a joint energy function over both the data and latent variables. The joint distribution factorizes into a tractable posterior and a marginal likelihood, imposing an inductive bias on the latent space. 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.

References

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, pages 3412–3420, 2019b.

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, pages 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*, pages 2525–2534. PMLR, 2019.

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.

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.

Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International Conference on Machine Learning*, pages 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*, pages 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.

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*, pages 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*, pages 5232–5242, 2019b.

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

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

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

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)*, pages 681–688, 2011.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3-4):229–256, 1992. ISSN 0885-6125. doi: 10.1007/BF00992696.

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*, pages 10205–10215. 2020.

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

Appendix A. Derivation for Two Cases of Inductive Biases

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 $z \in \mathbb{R}^K$,

$$b_\theta(z) = \sum_k \big(\eta(z_k)^\top \lambda - A(\lambda)\big),$$

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

$$\lambda = \left(\frac{\mu}{\sigma^2}, -\frac{1}{2\sigma^2}\right) = \left(0, -\frac{1}{2}\right). \tag{9}$$

The marginal likelihood of the CEBM is then

$$p_\theta(x \mid \lambda) = \frac{1}{Z_\theta} \exp\Big\{\sum_k \big(A(\tilde{\lambda}_k) - A(\lambda)\big)\Big\}, \tag{10}$$

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

$$A(\lambda_k) = -\frac{\lambda_{k,1}^2}{4\lambda_{k,2}} - \frac{1}{2}\log(-\lambda_{k,2}).$$

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. As an inductive bias that is amenable to uncovering this structure, we will consider a bias in the form of a mixture of L Gaussians,

$$b_\theta(y, z) = \sum_{k,l} I[y = l]\big(\eta(z_k)^\top \lambda_{l,k} - A(\lambda_{l,k})\big).$$

Here $z \in \mathbb{R}^K$ is a vector of features and $y \in \{1, \ldots, L\}$ 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 $\tilde{\lambda}_{l,k} = \lambda_{l,k} + t_{\theta,l,k}(x)$ to refer to the posterior parameters, then we obtain an energy

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

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

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

which then allows us to compute the marginal

$$p_\theta(x \mid \lambda) = \sum_y p_\theta(x, y \mid \lambda). \tag{11}$$

We optimize this marginal with respect hyperaparameters λ as well as the weights θ.

Appendix B. Recovering VAE as a Special Case

B.1. Variational Autoencoders

Variational autoencoders are a widely used class of deep generative models Kingma and Welling (2013); Rezende et al. (2014). A VAE defines a joint distribution $p_\theta(x, z)$ over data x and latent variables z; it combines an unstructured prior (e.g. a spherical Gaussian) with a likelihood that is parameterized by an expressive neural network, often referred to as a decoder. An inference model, also known as an encoder, approximates the posterior $p_\theta(z \mid x)$ by mapping each data point x onto latent variables z. These models are trained by maximizing the stochastic evidence lower bound (ELBO) defined as

$$\mathcal{L}(\phi, \theta) = \mathbb{E}_{p_{\text{data}}(x)\, q_\phi(z|x)} \left[\log \frac{p_\theta(x, z)}{q_\phi(z \mid x)} \right] \tag{12}$$

When the $q_\phi(z|x)$ is reparameterizable, we can compute Monte Carlo estimates of the gradient of this objective using pathwise derivatives. Non-reparameterizable cases, such as models with discrete variables, require likelihood-ratio estimators Williams (1992).

Despite their successes, VAEs have limitations. By maximizing the ELBO, we favor encoder-decoder pair that perfectly reconstruct their input, so there is nothing preventing the VAE from mapping similar inputs to similar encoding, even when they might be semantically different. Likewise, examples that might be very dissimilar in pixel space because of noise or benign transformations might end up with very different latent representations.

B.2. VAE is a Special Case

Recall that we define the energy function in a form as

$$E_\theta(x, z) = -t_\theta(x)^\top \eta_\theta(z) - b_\theta(z). \tag{13}$$

In this setting, $\eta_\theta(z)$ is the generator network that maps low-dimensional latent variables to a high-dimensional vector of natural parameters. The function $t(x)$ is a known mapping of data to the sufficient statistics of a Gaussian or Bernoulli likelihood.

The bias $b_\theta(z)$ contains the terms in the log density $\log p_\theta(x, z)$ that only depend on z. In a VAE there are two such terms. The first is the log prior $\log p_\theta(z)$. The second is the log normalizer of $A(\eta_\theta(z))$ for the likelihood, which can be computed in closed form because Gaussian and Bernoulli distributions are in the exponential family. Combining these terms yields an energy function for the bias,

$$b_\theta(z) = \log p_\theta(z) - A(\eta_\theta(z)). \tag{14}$$

Appendix C. 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. For IGEBMs, we add one extra MLP as its final layer which outputs a scalar value. VAEs use the same encoder network as the CEBMs, and use a decoder network in form of an MLP followed by 4-layer CNN.

Table Table C, Table Table C, and Table Table C show the architectures used for CEBM, VAE, and IGEBM, respectively.

(a) MNIST and Fashion-MNIST.

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

(b) CIFAR10 and SVHN.

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

Table 3: Architecture of CEBM.

(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 LeakyReLU. stride 1. padding 1	FC. 128 ReLU
4×4 conv. 64 LeakyReLU. stride 2. padding 1	FC. $3 \times 3 \times 32$ ReLU
4×4 conv. 32 LeakyReLU. stride 2. padding 1	4×4 upconv. 32 LeakyReLU. stride 2. padding 1
4×4 conv. 32 LeakyReLU. stride 2. padding 1	4×4 upconv. 64 LeakyReLU. stride 2. padding 1
FC. 128 ReLU	4×4 upconv. 64 LeakyReLU. stride 2. padding 0
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 LeakyReLU. stride 1. padding 1	FC. 128 ReLU
4×4 conv. 128 LeakyReLU. stride 2. padding 1	FC. $4 \times 4 \times 512$ ReLU
4×4 conv. 256 LeakyReLU. stride 2. padding 1	4×4 upconv. 32 LeakyReLU. stride 2. padding 1
4×4 conv. 512 LeakyReLU. stride 2. padding 1	4×4 upconv. 64 LeakyReLU. stride 2. padding 1
FC. 128 ReLU	3×3 upconv. 64 LeakyReLU. stride 2. padding 1
FC. 2×128	3×3 upconv. 1 stride 1. padding 1

Table 4: Architecture of VAE.

(*a*) MNIST and Fashion-MNIST.

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

(*b*) CIFAR10 and SVHN.

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

Table 5: Architecture of IGEBM

Appendix D. Training Details of Persistent Contrastive Divergence

Optimization. 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 L2 regularize energy magnitudes (proposed by Du and Mordatch (2019)), where the coefficient of the L2 regularization term is 0.1. We empirically found that the training would become unstable without this regularization. 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 and Mordatch (2019), we found it useful to add some noise to the image before encoding. In our experiments, we used Gaussian noise with $\sigma^2 = 0.03$. For the mixture models (CEBMM and GMM-VAE), we used 50 mixtures.

Training Stability. As observed in previous work Du and Mordatch (2019); Grathwohl et al. (2019), training EBMs can be a challenging task that often requires a thorough hyperparameter search. We found that the choices of activation function, learning rate, number of SGLD steps, and regularization will all affect training stability. Models regularly diverge during training, and it is difficult to perform diagnostics given that $\log p_\theta(x)$ cannot be computed. As suggested by Nijkamp et al. (2019a), we found checking the difference in energy between data and model samples to be helpful for verifying stable training. We also note that in general, we 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 is having more difficulty to converge when the latent space is more disjoint.

Appendix E. Persistent Constrastive Divergence

Algorithm 1: Persistent Contrastive Divergence

Input: $p_{\text{data}}(\cdot)$, θ, α, T
$\mathcal{B} \leftarrow \{x_b \sim \mathcal{U}$ **for** $b = 1 \ldots$ buffer-size$\}$;
while *not converged* **do**
  $x \sim p_{\text{data}}(x)$;
  $x' \sim \mathcal{B}$ with 95% probability and $\mathcal{U}$ otherwise;
  for $t = 1 \ldots T$ **do**
    $\epsilon \sim \mathcal{N}(0, \alpha)$;
    $x' \leftarrow x' - \frac{\alpha}{2} \nabla_x E_\theta(x') + \epsilon$;
  end
  $\Delta_\theta \leftarrow \nabla_\theta E_\theta(x) - \nabla_\theta E_\theta(x')$;
  $\theta \leftarrow \text{Adam}(\theta, \Delta_\theta)$;
  $\mathcal{B}[x'] \leftarrow x'$;
end
Output: θ

Appendix F. Confusion Matrices on 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

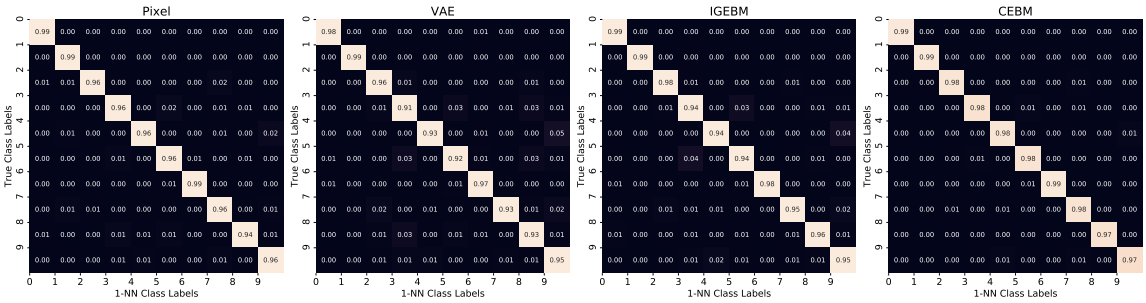

Figure 3: MNIST

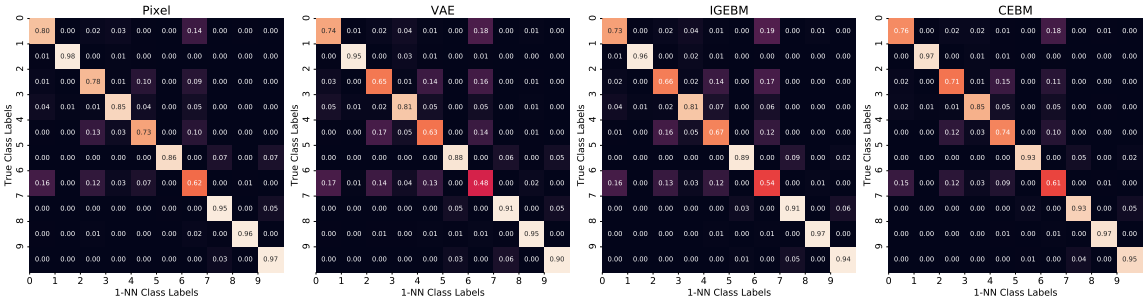

Figure 4: Fashion-MNIST

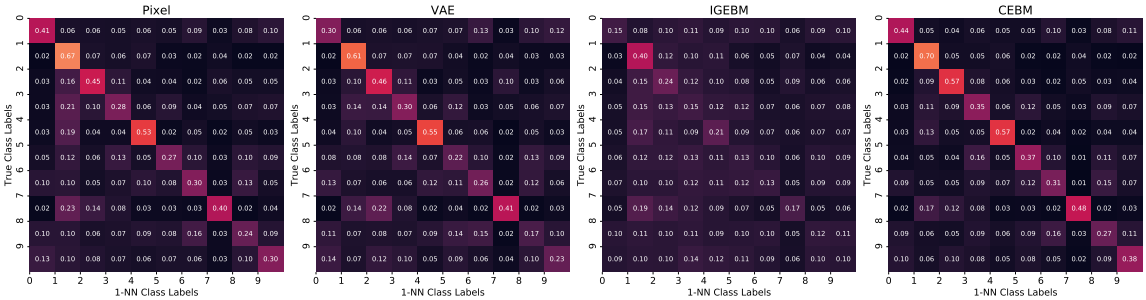

Figure 5: SVHN

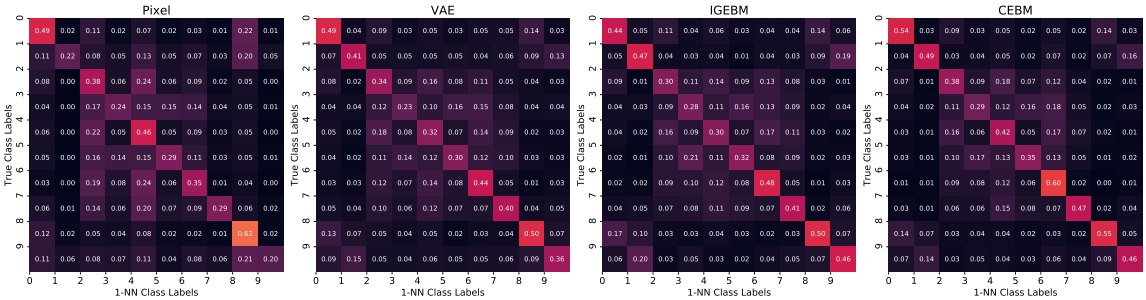

Figure 6: CIFAR10