Supplementary Materials for Conjugate Energy-Based Models

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

$$E_\theta^{\text{RBM}}(x, z) = -\langle x^\top \theta_{xz}, z\rangle - \langle x, \theta_x\rangle - \langle z, \theta_z\rangle, \tag{29}$$

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,

$$E_\theta^{\text{EFH}}(x, z) = -\langle t_x(x)^\top \theta_{xz}, t_z(z)\rangle - \langle t_x(x), \theta_x\rangle - \langle t_z(z), \theta_z\rangle, \tag{30}$$

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:

$$\text{Likelihood} \qquad p_\theta(x|z) = \exp\left\{\langle t_x(x), \tilde{\theta}_x\rangle - A(\tilde{\theta}_x)\right\}, \qquad \tilde{\theta}_x = \theta_x + \theta_{xz} t_z(z), \tag{31}$$

$$\text{Posterior} \qquad p_\theta(z|x) = \exp\left\{\langle t_z(z), \tilde{\theta}_z\rangle - B(\tilde{\theta}_z)\right\}, \qquad \tilde{\theta}_z = \theta_z + \theta_{xz} t_x(x), \tag{32}$$

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 17, we recover the energy function for an EFH by setting

$$t_\theta(x) = [t_x(x)^\top \theta_{xz},\ \langle \theta_x, t_x(x)\rangle], \qquad \eta(z) = [t_z(z),\ 1], \qquad E_\theta(z) = -\langle t_z(z), \theta_z\rangle. \tag{33}$$

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.

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

Table 4. 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. Derivation of Prior and Likelihood in a CEBM

B.1. Prior

$$p_{\theta,\lambda}(z) = \int dx \ \frac{1}{Z_{\theta,\lambda}} \exp\{-E_{\theta,\lambda}(x, z)\} \tag{34}$$

$$= \frac{1}{Z_{\theta,\lambda}} \int dx \ \exp\{-E_{\theta,\lambda}(x, z)\} \tag{35}$$

$$= \frac{1}{Z_{\theta,\lambda}} \int dx \ \exp\{\langle t_\theta(x), \eta(z)\rangle - E_\lambda(z)\} \tag{36}$$

$$= \frac{\exp\{-E_\lambda(z)\}}{Z_{\theta,\lambda}} \int dx \ \exp\{\langle t_\theta(x), \eta(z)\rangle\} \tag{37}$$

B.2. Likelihood

$$p_{\theta,\lambda}(x|z) = \frac{p_{\theta,\lambda}(x, z)}{p_{\theta,\lambda}(z)} \tag{38}$$

$$= \frac{\frac{1}{Z_{\theta,\lambda}} \exp\{-E_{\theta,\lambda}(x, z)\}}{\frac{\exp\{-E_\lambda(z)\}}{Z_{\theta,\lambda}} \int dx \ \exp\{\langle t_\theta(x), \eta(z)\rangle\}} \tag{39}$$

$$= \frac{\frac{\exp\{-E_\lambda(z)\}}{Z_{\theta,\lambda}} \exp\{\langle t_\theta(x), \eta(z)\rangle\}}{\frac{\exp\{-E_\lambda(z)\}}{Z_{\theta,\lambda}} \int dx \ \exp\{\langle t_\theta(x), \eta(z)\rangle\}} \tag{40}$$

$$= \frac{\exp\{\langle t_\theta(x), \eta(z)\rangle\}}{\int dx \ \exp\{\langle t_\theta(x), \eta(z)\rangle\}} \tag{41}$$

C. Training Details

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 & 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 & 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.

D. Model Architectures

Table 5, Table 7, and Table 6 show the architectures used for CEBM, VAE, and IGEBM, respectively.

Table 5. 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 6. 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 7. 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 8. 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.

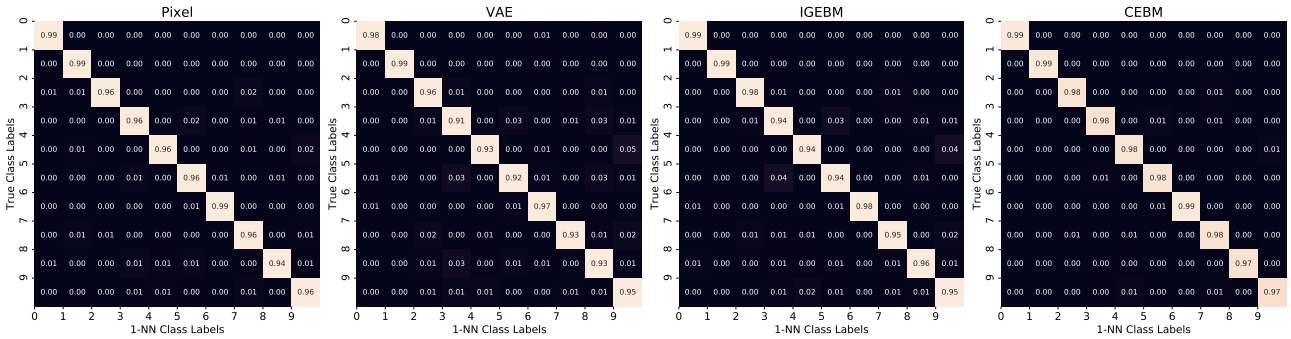
Discriminator
Input $28 \times 28 \times 1$ images
3×3 conv. 64 stride 1. padding 1. BN. LeakyReLU.
4×4 conv. 128 stride 2. padding 1. BN. LeakyReLU.
4×4 conv. 256 stride 2. padding 1. BN. LeakyReLU.
4×4 conv. 512 stride 2. padding 1. BN. LeakyReLU.
FC. 128 LeakyReLU.
256 FC 128 LeakyReLU. FC. 1. Sigmoid.

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.

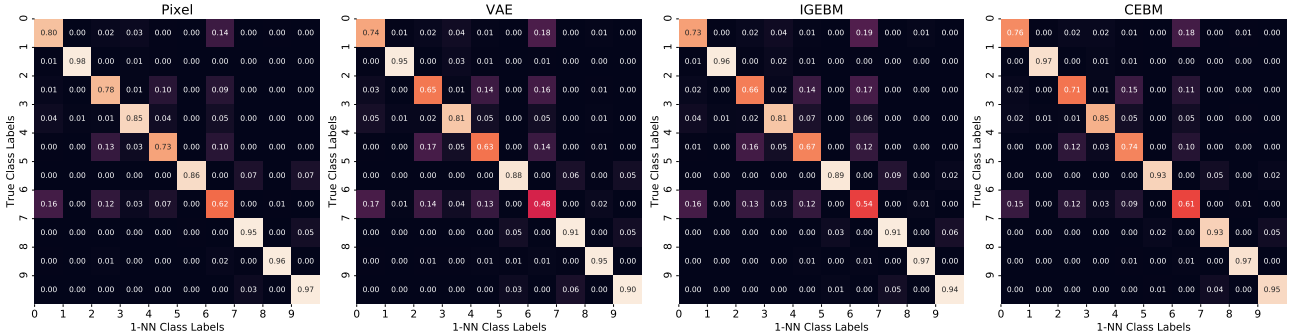
E. Additional Results

E.1. Confusion Matrices on 1-NN Classification

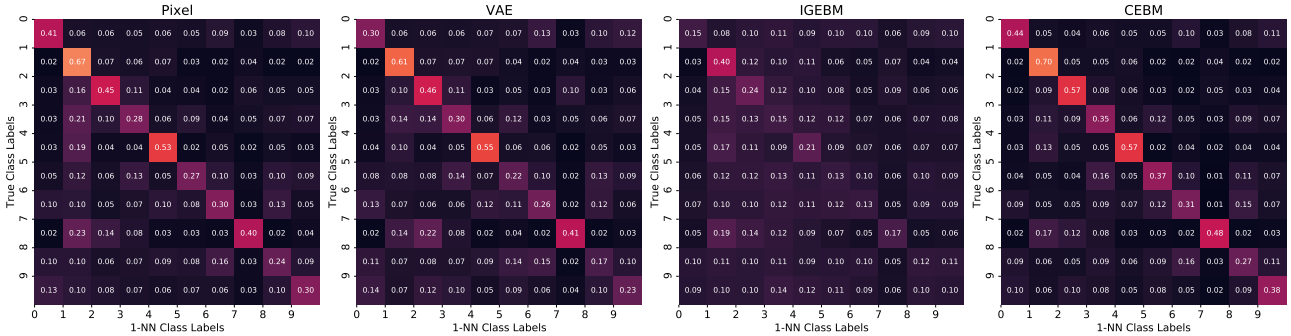
We perform 1-nearest-neighbor classification task for MNIST, Fashion-MNIST, SVHN, CIFAR10. We compute the L2 distance in the latent space of VAE, IGEBM and CEBM, and also in pixel space. We visualize the confusion matrices



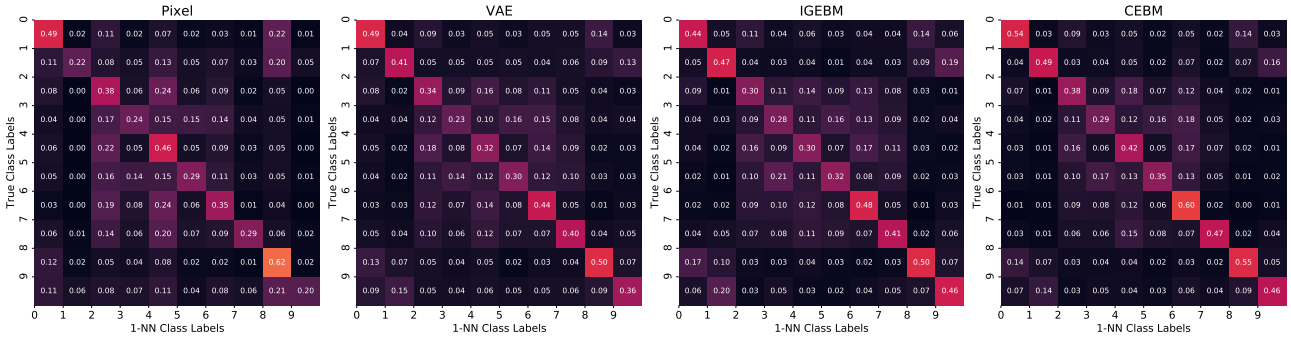
(a) MNIST



(b) Fashion-MNIST



(c) SVHN



(d) CIFAR10

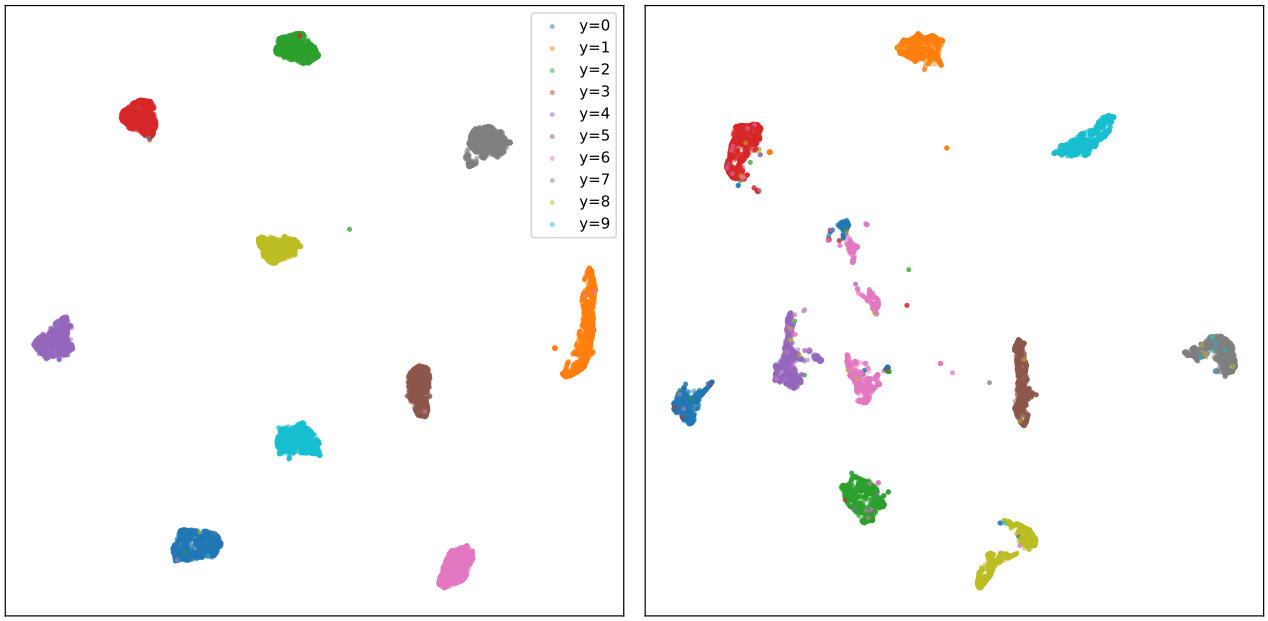


Figure 5. CEBMs latent space visualized with UMAP for MNIST (*Left*) and FashionMNIST (*Right*).