Guiding Energy-based Models via Contrastive Latent Variables

Hankook Lee[1] **Jongheon Jeong**[1] **Sejun Park**[2] **Jinwoo Shin**[1]
[1]Korea Advanced Institute of Science and Technology (KAIST)
[2]Korea University

Abstract

An energy-based model (EBM) is a popular generative framework that offers both explicit density and architectural flexibility, but training them is difficult since it is often unstable and time-consuming. In recent years, various training techniques have been developed, *e.g.*, better divergence measures or stabilization in MCMC sampling, but there often exists a large gap between EBMs and other generative frameworks like GANs in terms of generation quality. In this paper, we propose a novel and effective framework for improving EBMs via contrastive representation learning (CRL). To be specific, we consider representations learned by contrastive methods as the true underlying latent variable. This *contrastive latent variable* could guide EBMs to understand the data structure better, so it can improve and accelerate EBM training significantly. To enable the joint training of EBM and CRL, we also design a new class of latent-variable EBMs for learning the joint density of data and the contrastive latent variable. Our experimental results demonstrate that our scheme achieves lower FID scores, compared to prior-art EBM methods (e.g., additionally using variational autoencoders or diffusion techniques), even with significantly faster and more memory-efficient training.

1 Introduction

Energy-based models (EBMs) [1, 2], whose density is proportional to the exponential negative energy, *i.e.*, $p_\theta(\mathbf{x}) \propto \exp(-E_\theta(\mathbf{x}))$, have recently gained much attention due to their attractive properties: EBMs can naturally provide the explicit (unnormalized) density unlike GANs [3], and, they are much less restrictive in architectural designs than other explicit density models such as autoregressive [4, 5] and flow-based models [6, 7]. Despite the attractive properties, training EBMs has remained challenging; *e.g.*, it often suffers from the training instability due to the intractable sampling and the absence of the normalizing constant. There have recently developed various techniques for improving the stability [8–12]. To further improve EBMs, there are several recent attempts to incorporate other generative models into EBM training, *e.g.*, VAEs [13], flow-based models [14, 15], or diffusion techniques [16]. However, they often require a high computational cost for training such an extra generative model, or there still exists a large gap between EBMs and state-of-the-art generative frameworks like GANs [17] or score-based models [18].

Instead of utilizing extra expensive generative models, in this paper, we ask whether EBMs can be improved by other unsupervised techniques of low cost. To this end, we are inspired by recent advances in unsupervised representation learning literature [19–21], especially by the fact that the discriminative representations can be obtained much easier than generative modeling. In particular, we primarily focus on contrastive representation learning [22, 19, 23] since it can learn instance discriminability, which has been shown to be effective in not only representation learning, but also training GANs [24, 17] and out-of-distribution detection [25].

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

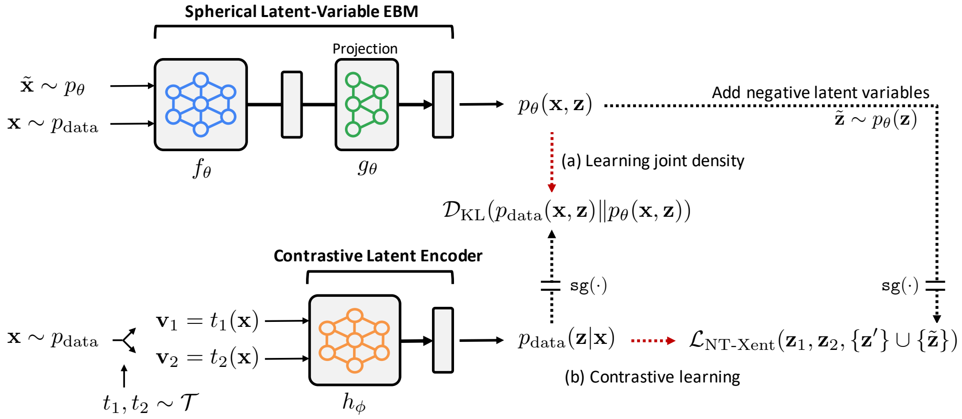

Spherical Latent-Variable EBM
Projection
$\tilde{\mathbf{x}} \sim p_\theta$
$\mathbf{x} \sim p_{\text{data}}$
f_θ
g_θ
$p_\theta(\mathbf{x}, \mathbf{z})$
Add negative latent variables
$\tilde{\mathbf{z}} \sim p_\theta(\mathbf{z})$
(a) Learning joint density
$\mathcal{D}_{\text{KL}}(p_{\text{data}}(\mathbf{x}, \mathbf{z}) \| p_\theta(\mathbf{x}, \mathbf{z}))$
Contrastive Latent Encoder
$\text{sg}(\cdot)$
$\text{sg}(\cdot)$
$\mathbf{x} \sim p_{\text{data}}$
$\mathbf{v}_1 = t_1(\mathbf{x})$
$\mathbf{v}_2 = t_2(\mathbf{x})$
$t_1, t_2 \sim \mathcal{T}$
h_ϕ
$p_{\text{data}}(\mathbf{z}|\mathbf{x})$
$\mathcal{L}_{\text{NT-Xent}}(\mathbf{z}_1, \mathbf{z}_2, \{\mathbf{z}'\} \cup \{\tilde{\mathbf{z}}\})$
(b) Contrastive learning

Figure 1: Illustration of the proposed Contrastive Latent-guided Energy Learning (CLEL) framework. (a) Our spherical latent-variable EBM (f_θ, g_θ) learns the joint data distribution $p_{\text{data}}(\mathbf{x}, \mathbf{z})$ generated by our contrastive latent encoder h_ϕ. (b) The encoder h_ϕ is trained by contrastive learning with additional negative variables $\tilde{\mathbf{z}} \sim p_\theta(\tilde{\mathbf{z}})$.

In this paper, we propose *Contrastive Latent-guided Energy Learning (CLEL)*, a simple yet effective framework for improving EBMs via contrastive representation learning (CRL). Our CLEL consists of two components, which are illustrated in Figure 1.

- **Contrastive latent encoder.** Our key idea is to consider representations learned by CRL as an underlying latent variable distribution $p_{\text{data}}(\mathbf{z}|\mathbf{x})$. Specifically, we train an encoder h_ϕ via CRL, and treat the encoded representation $\mathbf{z} := h_\phi(\mathbf{x})$ as the true latent variable given data $\mathbf{x}$, *i.e.*, $\mathbf{z} \sim p_{\text{data}}(\cdot|\mathbf{x})$. This latent variable could guide EBMs to understand the underlying data structure more quickly and accelerate training since the latent variable contains semantic information of the data thanks to CRL. Here, we assume the latent variables are spherical, *i.e.*, $\|\mathbf{z}\|_2 = 1$, since recent CRL methods [23, 19] use the cosine distance on the latent space.
- **Spherical latent-variable EBM.** We introduce a new class of latent-variable EBMs $p_\theta(\mathbf{x}, \mathbf{z})$ for modeling the joint distribution $p_{\text{data}}(\mathbf{x}, \mathbf{z})$ generated by the contrastive latent encoder. Since the latent variables are spherical, we separate the output vector $f := f_\theta(\mathbf{x})$ into its norm $\|f\|_2$ and direction $f/\|f\|_2$ for modeling $p_\theta(\mathbf{x})$ and $p_\theta(\mathbf{z}|\mathbf{x})$, respectively. We found that this separation technique reduces the conflict between $p_\theta(\mathbf{x})$ and $p_\theta(\mathbf{z}|\mathbf{x})$ optimizations, which makes training stable. In addition, we treat the latent variables drawn from our EBM, $\tilde{\mathbf{z}} \sim p_\theta(\mathbf{z})$, as additional negatives in CRL, which further improves our CLEL. Namely, CRL guides EBM and vice versa.

We demonstrate the effectiveness of the proposed framework through extensive experiments. For example, our EBM achieves 8.61 FID under unconditional CIFAR-10 generation, which is lower than those of existing EBM models. Here, we remark that utilizing CRL into our EMB training increases training time by only 10% in our experiments (*e.g.*, 38→41 GPU hours). This enables us to achieve the lower FID score even with significantly less computational resources than the prior EBMs that utilize VAEs [13] or diffusion-based recovery likelihood [16]. We remark that our idea is not limited to contrastive representation learning and we show EBMs can be also improved by other representation learning methods like BYOL [20] or MAE [21].

2 Method

Our goal is to learn an energy-based model (EBM) $p_\theta(\mathbf{x}) \propto \exp(-E_\theta(\mathbf{x}))$ to approximate a complex underlying data distribution $p_{\text{data}}(\mathbf{x})$. In this work, we propose Contrastive Latent-guided Energy Learning (CLEL), a simple yet effective framework for improving EBMs via contrastive representation learning. Our main intuition is that directly incorporating with *meaningful semantic information* of data could improve EBMs. To this end, we consider the (random) representation $\mathbf{z} \sim p_{\text{data}}(\mathbf{z}|\mathbf{x})$ of $\mathbf{x}$, generated by contrastive learning, as the true underlying latent variable. Namely, we model the joint distribution $p_{\text{data}}(\mathbf{x}, \mathbf{z}) = p_{\text{data}}(\mathbf{x}) p_{\text{data}}(\mathbf{z}|\mathbf{x})$ via a latent-variable EBM $p_\theta(\mathbf{x}, \mathbf{z})$. We first briefly describe the concepts of EBMs and CRL in Section 2.1, and then introduce our framework in Section 2.2. Our framework is illustrated in Figure 1.

2.1 Background

An **energy-based models (EBM)** is a probability distribution on $\mathbb{R}^{d_{\mathbf{x}}}$, defined as follows: for data $\mathbf{x} \in \mathbb{R}^{d_{\mathbf{x}}}$, $p_\theta(\mathbf{x}) = \exp(-E_\theta(\mathbf{x}))/Z_\theta$ where $Z_\theta = \int_{\mathbb{R}^{d_{\mathbf{x}}}} \exp(-E_\theta(\mathbf{x}))d\mathbf{x}$ denotes the normalizing constant, called the *partition function*. An important application of EBMs is to find a parameter θ such that p_θ is close to p_{data}. A popular method for finding such θ is to minimize Kullback–Leibler (KL) divergence between p_{data} and p_θ via gradient descent:

$$\nabla_\theta D_{\text{KL}}(p_{\text{data}} \| p_\theta) = \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}[\nabla_\theta E_\theta(\mathbf{x})] - \mathbb{E}_{\tilde{\mathbf{x}}\sim p_\theta}[\nabla_\theta E_\theta(\tilde{\mathbf{x}})]. \tag{1}$$

Since this gradient computation (1) is NP-hard in general [26], it is often approximated via Markov chain Monte Carlo (MCMC) methods. In this work, we use the stochastic gradient Langevin dynamics (SGLD) [27], a gradient-based MCMC method for approximate sampling.

Contrastive representation learning (CRL) aims to learn a meaningful representation by minimizing distance between similar (*i.e.*, positive) samples, and maximizing distance between dissimilar (*i.e.*, negative) samples on the representation space. To this end, we use the following objective [19]:

$$\mathcal{L}_{\text{NT-Xent}}(\mathbf{z}, \mathbf{z}_+, \{\mathbf{z}_-\}; \tau) = -\log \frac{\exp(\text{sim}(\mathbf{z}, \mathbf{z}_+)/\tau)}{\exp(\text{sim}(\mathbf{z}, \mathbf{z}_+)/\tau) + \sum_{\mathbf{z}_-} \exp(\text{sim}(\mathbf{z}, \mathbf{z}_-)/\tau)}, \tag{2}$$

where $\mathbf{z} = h_\phi(\mathbf{x})$ is the representation, $h_\phi : \mathbb{R}^{d_{\mathbf{x}}} \to \mathbb{R}^{d_{\mathbf{z}}}$ be a ϕ-parameterized encoder, $\text{sim}(\mathbf{u}, \mathbf{v})$ is the cosine similarity, and $(\mathbf{x}, \mathbf{x}_+)$ and $(\mathbf{x}, \mathbf{x}_-)$ are positive and negative pairs, respectively.

2.2 CLEL: Contrastive Latent-guided Energy Learning

Contrastive latent encoder. To construct a meaningful latent distribution $p_{\text{data}}(\mathbf{z}|\mathbf{x})$ for improving EBMs, we use a latent encoder h_ϕ trained by contrastive learning. Formally, we define $p_{\text{data}}(\mathbf{z}|\mathbf{x})$ by $p_{\text{data}}(\mathbf{z}|\mathbf{x}) := \mathbb{P}_{t\sim\mathcal{T}}(\mathbf{z} = h_\phi(t(\mathbf{x}))/\|h_\phi(t(\mathbf{x}))\|_2)$ where $\mathcal{T}$ is a random augmentation distribution.

Spherical latent-variable energy-based models. We use a DNN $f_\theta : \mathbb{R}^{d_{\mathbf{x}}} \to \mathbb{R}^{d_{\mathbf{z}}}$ parameterized by θ for modeling $p_\theta(\mathbf{x}, \mathbf{z})$. Following that the latent variable $\mathbf{z} \sim p_{\text{data}}(\mathbf{z}|\mathbf{x})$ is on the unit sphere, we utilize the directional information $f_\theta(\mathbf{x})/\|f_\theta(\mathbf{x})\|_2$ for modeling $p_\theta(\mathbf{z}|\mathbf{x})$, while the remaining information $\|f_\theta(\mathbf{x})\|_2$ is used for modeling $p_\theta(\mathbf{x})$.[1] Overall, we define the joint energy $E_\theta(\mathbf{x}, \mathbf{z})$ by $E_\theta(\mathbf{x}, \mathbf{z}) := \frac{1}{2}\|f_\theta(\mathbf{x})\|_2^2 - \beta g_\theta\left(\frac{f_\theta(\mathbf{x})}{\|f_\theta(\mathbf{x})\|_2}\right)^\top \mathbf{z}$ where $\beta \geq 0$ is a hyperparameter and $g_\theta : \mathbb{S}^{d_{\mathbf{z}}-1} \to \mathbb{S}^{d_{\mathbf{z}}-1}$ is a directional projection MLP. Note that the marginal energy $E_\theta(\mathbf{x})$ only depends on $\|f_\theta(\mathbf{x})\|_2$ since $\int_{\mathbb{S}^{d-1}} \exp(\beta g_\theta\left(f_\theta(\mathbf{x})/\|f_\theta(\mathbf{x})\|_2\right)^\top \mathbf{z})d\mathbf{z}$ is independent of $\mathbf{x}$ due to the symmetry.

Training. Let $\{\mathbf{x}^{(i)}\}_{i=1}^n$ be a random mini-batch of training samples. We first generate n samples $\{\tilde{\mathbf{x}}^{(i)}\}_{i=1}^n \sim p_\theta(\mathbf{x})$ using the current EBM E_θ via SGLD. We then draw latent variables from p_{data} and p_θ, *i.e.*, $\mathbf{z}^{(i)} \sim p_{\text{data}}(\mathbf{z}|\mathbf{x}^{(i)})$ and $\tilde{\mathbf{z}}^{(i)} \sim p_\theta(\mathbf{z}|\tilde{\mathbf{x}}^{(i)})$ for all i. For the latter case, we simply use the mode of $p_\theta(\mathbf{z}|\mathbf{x}^{(i)})$ instead of sampling, namely, $\tilde{\mathbf{z}}^{(i)} := g_\theta(f_\theta(\mathbf{x})/\|f_\theta(\mathbf{x})\|_2)$. Let $\mathcal{B} := \{(\mathbf{x}^{(i)}, \mathbf{z}^{(i)})\}_{i=1}^n$ and $\tilde{\mathcal{B}} := \{(\tilde{\mathbf{x}}^{(i)}, \tilde{\mathbf{z}}^{(i)})\}_{i=1}^n$ be real and generated mini-batches, respectively. We optimize θ and ϕ via minimizing KL divergence (1) and contrastive learning (2), respectively. Formally, we minimize the following losses simultaneously (see Appendix A for CLEL training pseudocode):[2]

$$\mathcal{L}_{\text{EBM}}(\mathcal{B}, \tilde{\mathcal{B}}; \theta, \alpha, \beta) = \frac{1}{n}\sum_{i=1}^n E_\theta(\mathbf{x}^{(i)}, \mathbf{z}^{(i)}) - E_\theta(\tilde{\mathbf{x}}^{(i)}) + \alpha \cdot (E_\theta(\mathbf{x}^{(i)})^2 + E_\theta(\tilde{\mathbf{x}}^{(i)})^2), \tag{3}$$

$$\mathcal{L}_{\text{LE}}(\mathcal{B}, \tilde{\mathcal{B}}; \phi, \tau) = \frac{1}{2n}\sum_{i=1}^n \sum_{j=1,2} \mathcal{L}_{\text{NT-Xent}}\left(\mathbf{z}_j^{(i)}, \mathbf{z}_{3-j}^{(i)}, \{\mathbf{z}_l^{(k)}\}_{k\neq i, l\in\{1,2\}} \cup \{\tilde{\mathbf{z}}^{(i)}\}_{i=1}^n; \tau\right), \tag{4}$$

where $\mathbf{z}_j^{(i)} = h_\phi(t_j^{(i)}(\mathbf{x}^{(i)}))$ is the representation for contrastive learning with augmentations $t_j^{(i)} \sim \mathcal{T}$, and α is a hyperparameter for energy regularization to prevent divergence [8]. Remark that we here utilize $\{\tilde{\mathbf{z}}^{(i)}\}_{i=1}^n$ as additional negative latent variables for contrastive learning.

[1] We empirically found that this norm-direction separation stabilizes the latent-variable EBM training.

[2] Here, $\tilde{\mathbf{z}}$ is unnecessary for $D_{\text{KL}}(p_{\text{data}}\|p_\theta)$ since $\mathbb{E}_{\tilde{\mathbf{z}}\sim p_\theta(\tilde{\mathbf{z}}|\tilde{\mathbf{x}})}[\nabla_\theta E_\theta(\tilde{\mathbf{x}}, \tilde{\mathbf{z}})] = \nabla_\theta E_\theta(\tilde{\mathbf{x}})$.

Table 1: (a,b) unconditional generation and (c) out-of-distribution (OOD) detection results. † denotes EBMs that utilize auxiliary generators, and ‡ denotes hybrid discriminative-generative models. The training time and GPU memory footprint are based on single RTX3090 GPU of 24G memory. Underline is based on our estimation.

(a) FID scores

Method	**FID**
CIFAR-10	
Short-run EBM [9]	44.50
JEM[‡] [28]	38.40
IGEBM [8]	38.20
FlowCE[†] [14]	37.30
VERA[†‡] [29]	27.50
Improved CD [10]	25.10
BiDVL [30]	20.75
GEBM[†] [31]	19.31
CF-EBM [32]	16.71
CoopFlow[†] [15]	15.80
CLEL-Base (Ours)	15.27
VAEBM[†] [13]	12.19
EBM-Diffusion [16]	9.58
CLEL-Large (Ours)	**8.61**
ImageNet 32×32	
IGEBM [8]	62.23
PixelCNN [4]	40.51
Improved CD [10]	32.48
CF-EBM [32]	26.31
CLEL-Base (Ours)	**22.16**

(b) FID improvements via different configurations (CIFAR-10)

Method	**Params (M)**	**Time**	**Memory**	**FID**
Baseline w/o CLEL	6.96	38h	6G	23.50
+ CLEL (**Base**)	6.96	41h	7G	15.27
+ multi-scale architecture	19.29	74h	8G	12.46
+ CRL with a batch size of 256	19.29	76h	10G	11.65
+ more channels (**Large**)	30.70	133h	11G	**8.61**
EBM-Diffusion ($N = 2$ blocks)	9.06	163h	10G	17.34
EBM-Diffusion ($N = 8$ blocks)	34.83	652h	40G	9.58
VAEBM	135.88	414h	129G	12.19

(c) Out-of-distribution detection (CIFAR-10 → OOD data)

Method	**SVHN**	**Textures**	**CIFAR10 Interp.**	**CIFAR100**	**CelebA**
PixelCNN++ [33]	0.32	0.33	0.71	0.63	-
GLOW [34]	0.24	0.27	0.51	0.55	0.57
NVAE [35]	0.42	-	0.64	0.56	0.68
IGEBM [8]	0.63	0.48	0.70	0.50	0.70
VAEBM [13]	0.83	-	0.70	0.62	0.77
Improved CD [10]	0.91	0.88	0.65	**0.83**	-
CLEL-Base (Ours)	**0.9848**	**0.9437**	**0.7248**	0.7161	**0.7717**
JEM[‡] [28]	0.67	0.60	0.65	0.67	0.75
VERA[†‡] [29]	0.83	-	0.86	0.73	0.33

3 Experiments

Training details. We provide all the implementation details are described in Appendidx B.

Unconditional image generation. An important application of EBMs is to generate images using the energy function $E_\theta(\mathbf{x})$. To this end, we train our CLEL framework on CIFAR-10 [36] and ImageNet 32×32 [37, 38] under the unsupervised setting. Table 1a shows the FID scores of our CLEL and other EBMs for unconditional generation on CIFAR-10 and ImageNet 32×32, respectively. The unconditionally generated samples are provided in Appendix C. We first find that CLEL outperforms previous EBMs under both CIFAR-10 and ImageNet 32×32 datasets. As shown in Table 1b, our method can benefit from a multi-scale architecture as Du et al. [10] did, contrastive representation learning (CRL) with a larger batch, more channels at lower layers in our EBM f_θ. As a result, we achieve 8.61 FID on CIFAR-10, which is lower than that of the prior-art EBM based on diffusion recovery likelihood, EBM-Diffusion [16], even with 5× faster and 4× more memory-efficient training (when using the similar number of parameters for EBMs). Then, we narrow the gap between EBMs and state-of-the-art frameworks like GANs without help from other generative models.

Out-of-distribution detection. EBMs can be also used for detecting out-of-distribution (OOD) samples. For the OOD sample detection, previous EBM-based approaches often use the (marginal) unnormalized likelihood $p_\theta(\mathbf{x}) \propto \exp(-E_\theta(\mathbf{x}))$. In contrast, our CLEL is capable of modeling the joint density $p_\theta(\mathbf{x}, \mathbf{z}) \propto E_\theta(\mathbf{x}, \mathbf{z})$. Using this capability, we propose an energy-based OOD detection score: given $\mathbf{x}$,

$$s(\mathbf{x}) := \frac{1}{2}\|f_\theta(\mathbf{x})\|_2^2 - \beta g_\theta\left(\frac{f_\theta(\mathbf{x})}{\|f_\theta(\mathbf{x})\|_2}\right)^\top \frac{h_\phi(\mathbf{x})}{\|h_\phi(\mathbf{x})\|_2}. \tag{5}$$

We found that the second term in (5) helps to detect the semantic difference between in- and out-of-distribution samples. Table 1c shows our CLEL's superiority over other explicit density models in OOD detection, especially when OOD samples are drawn from different domains, *e.g.*, SVHN [39] and Texture [40] datasets.

Conditional generation using $p_\theta(\mathbf{x}|\mathbf{z})$. Even without explicit conditional training, our latent-variable EBMs naturally can provide the latent-conditional density $p_\theta(\mathbf{x}|\mathbf{z})$. We verify its effectiveness under conditional sampling (Appendix D), and compositional sampling (Appendix E).

Analysis. We conduct ablation experiments to validate the contributions of CLEL's components and compatibility with other representation learning methods instead of contrastive one (Appendix F).

References

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

[2] Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton. Restricted boltzmann machines for collaborative filtering. In *International Conference on Machine Learning*, 2007.

[3] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, 2014.

[4] Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International Conference on Machine Learning*, 2016.

[5] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[6] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning*, 2015.

[7] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. In *International Conference on Learning Representations*, 2017.

[8] Yilun Du and Igor Mordatch. Implicit generation and generalization in energy-based models. In *Advances in Neural Information Processing Systems*, 2019.

[9] 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*, 2019.

[10] Yilun Du, Shuang Li, Joshua Tenenbaum, and Igor Mordatch. Improved contrastive divergence training of energy based models. In *International Conference on Machine Learning*, 2021.

[11] Lantao Yu, Yang Song, Jiaming Song, and Stefano Ermon. Training deep energy-based models with f-divergence minimization. In *International Conference on Machine Learning*, 2020.

[12] Lantao Yu, Jiaming Song, Yang Song, and Stefano Ermon. Pseudo-spherical contrastive divergence. In *Advances in Neural Information Processing Systems*, 2021.

[13] Zhisheng Xiao, Karsten Kreis, Jan Kautz, and Arash Vahdat. Vaebm: A symbiosis between variational autoencoders and energy-based models. In *International Conference on Learning Representations*, 2021.

[14] 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*, pages 7518–7528, 2020.

[15] Jianwen Xie, Yaxuan Zhu, Jun Li, and Ping Li. A tale of two flows: Cooperative learning of langevin flow and normalizing flow toward energy-based model. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=31d5RLCUuXC.

[16] Ruiqi Gao, Yang Song, Ben Poole, Ying Nian Wu, and Diederik P Kingma. Learning energy-based models by diffusion recovery likelihood. In *International Conference on Learning Representations*, 2021.

[17] Minguk Kang, Woohyeon Shim, Minsu Cho, and Jaesik Park. Rebooting acgan: Auxiliary classifier gans with stable training. In *Advances in Neural Information Processing Systems*, 2021.

[18] Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. In *Advances in Neural Information Processing Systems*, 2021.

[19] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, 2020.

[20] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap Your Own Latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

[21] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners, 2021.

[22] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[23] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[24] Jongheon Jeong and Jinwoo Shin. Training {gan}s with stronger augmentations via contrastive discriminator. In *International Conference on Learning Representations*, 2021.

[25] Jihoon Tack, Sangwoo Mo, Jongheon Jeong, and Jinwoo Shin. Csi: Novelty detection via contrastive learning on distributionally shifted instances. In *Advances in Neural Information Processing Systems*, 2020.

[26] Mark Jerrum and Alistair Sinclair. Polynomial-time approximation algorithms for the ising model. *SIAM Journal on Computing*, 22(5):1087–1116, 1993.

[27] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient Langevin dynamics. In *International Conference on Machine Learning*, 2011.

[28] Will Grathwohl, Kuan-Chieh Wang, Joern-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. In *International Conference on Learning Representations*, 2019.

[29] 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.

[30] Ge Kan, Jinhu Lü, Tian Wang, Baochang Zhang, Aichun Zhu, Lei Huang, Guodong Guo, and Hichem Snoussi. Bi-level doubly variational learning for energy-based latent variable models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[31] Michael Arbel, Liang Zhou, and Arthur Gretton. Generalized energy based models. In *International Conference on Learning Representations*, 2021.

[32] Yang Zhao, Jianwen Xie, and Ping Li. Learning energy-based generative models via coarse-to-fine expanding and sampling. In *International Conference on Learning Representations*, 2021.

[33] Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *arXiv preprint arXiv:1701.05517*, 2017.

[34] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in neural information processing systems*, 2018.

[35] Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. In *Advances in Neural Information Processing Systems*, 2020.

[36] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[37] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[38] Patryk Chrabaszcz, Ilya Loshchilov, and Frank Hutter. A downsampled variant of imagenet as an alternative to the cifar datasets. *arXiv preprint arXiv:1707.08819*, 2017.

[39] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

[40] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3606–3613, 2014.

[41] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[42] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018.

[43] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[44] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. *arXiv preprint arXiv:2011.10566*, 2020.

[45] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*, 2017.

[46] Maximilian Seitzer. pytorch-fid: FID Score for PyTorch. `https://github.com/mseitzer/pytorch-fid`, August 2020. Version 0.2.1.

[47] Yilun Du, Shuang Li, and Igor Mordatch. Compositional visual generation with energy based models. In *Advances in Neural Information Processing Systems*, 2020.

[48] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *International Conference on Computer Vision*, December 2015.

A Training procedure of CLEL

Algorithm 1 Contrastive Latent-guided Energy Learning (CLEL)

Require: a latent-variable EBM (f_θ, g_θ), a latent encoder h_ϕ, an augmentation distribution $\mathcal{T}$, hyperparameters $\alpha, \beta, \tau > 0$, and the stop-gradient operation $\texttt{sg}(\cdot)$.

```
for # training iterations do
    // Construct batches $\mathcal{B}$ and $\tilde{\mathcal{B}}$
    Sample $\{\mathbf{x}^{(i)}\}_{i=1}^n \sim p_{\text{data}}(\mathbf{x})$
    Sample $\{\tilde{\mathbf{x}}^{(i)}\}_{i=1}^n \sim p_\theta(\mathbf{x})$ using stochastic gradient Langevin dynamics (SGLD)
    $\mathbf{z}^{(i)} \leftarrow \texttt{sg}\left(h_\phi(t^{(i)}(\mathbf{x}^{(i)}))/\|h_\phi(t^{(i)}(\mathbf{x}^{(i)}))\|_2\right), t^{(i)} \sim \mathcal{T}$
    $\tilde{\mathbf{z}}^{(i)} \leftarrow \texttt{sg}\left(g_\theta(f_\theta(\tilde{\mathbf{x}}^{(i)})/\|f_\theta(\tilde{\mathbf{x}}^{(i)})\|_2)\right)$
    // Compute the EBM loss, $\mathcal{L}_{\text{EBM}}$
    $\mathcal{L}_{\text{EBM}} \leftarrow \frac{1}{n}\sum_{i=1}^n E_\theta(\mathbf{x}^{(i)}, \mathbf{z}^{(i)}) - E_\theta(\tilde{\mathbf{x}}^{(i)}) + \alpha \cdot (E_\theta(\mathbf{x}^{(i)})^2 + E_\theta(\tilde{\mathbf{x}}^{(i)})^2)$
    // Compute the encoder loss, $\mathcal{L}_{\text{LE}}$
    $\mathbf{z}_j^{(i)} \leftarrow h_\phi(t_j^{(i)}(\mathbf{x}^{(i)})), t_j^{(i)} \sim \mathcal{T}$
    $\mathcal{L}_{\text{LE}} \leftarrow \frac{1}{2n}\sum_{i=1}^n \sum_{j=1,2} \mathcal{L}_{\text{NT-Xent}}\left(\mathbf{z}_j^{(i)}, \mathbf{z}_{3-j}^{(i)}, \{\mathbf{z}_l^{(k)}\}_{k\neq i, l\in\{1,2\}} \cup \{\tilde{\mathbf{z}}^{(i)}\}_{i=1}^n; \tau\right)$
    Update $\theta$ and $\phi$ to minimize $\mathcal{L}_{\text{EBM}} + \mathcal{L}_{\text{LE}}$
end for
```

B Training details

Architectures. For the spherical latent-variable energy-based model (EBM) f_θ, we use the 8-block ResNet [41] architectures following Du and Mordatch [8]. The details of the (a) small, (b) base, and (c) large ResNets are described in Table 2. We append a 2-layer MLP with a output dimension of 128 to the ResNet, *i.e.*, $f_\theta : \mathbb{R}^{3\times 32\times 32} \to \mathbb{R}^{128}$. Note that we use the small model for ablation experiments in Appendix F. To stabilize training, we apply spectral normalization [42] to all convolutional layers. For the projection g_θ, we use a 2-layer MLP with a output dimension of 128, the leaky-ReLU activation, and no bias, *i.e.*, $g_\theta(\mathbf{u}) = W_2\sigma(W_1\mathbf{u}) \in \mathbb{R}^{128}$. For the latent encoder h_ϕ, we simpy use the CIFAR variant of ResNet-18 [41], followed by a 2-layer MLP with a output dimension of 128.

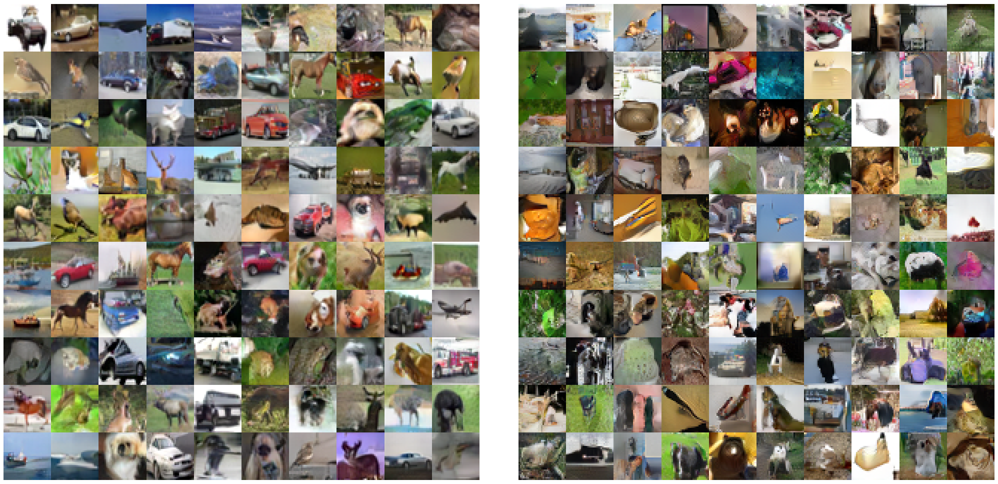
Table 2: Our EBM f_θ architectures. For our large model, we build three independent ResNets and resize an input image $\mathbf{x} \in \mathbb{R}^{3\times 32\times 32}$ to three resolutions: 32×32, 16×16, and 8×8. We use each ResNet for each resolution image, concatenate their output features, and then compute the final output feature $f_\theta(\mathbf{x}) \in \mathbb{R}^{128}$ using single MLP.

	Small		Base		Large		
Input	$(3, 32, 32)$		$(3, 32, 32)$		$(3, 32, 32), (3, 16, 16), (3, 8, 8)$		
EBM $f_\theta(\mathbf{x})$	Conv($3\times 3, 64$)		Conv($3\times 3, 128$)		Conv($3\times 3, 256$)		$\times 3$
	ResBlock(64)	$\times 1$	ResBlock(128)	$\times 2$	ResBlock(256)	$\times 2$	
	AvgPool(2×2)		AvgPool(2×2)		AvgPool(2×2)		
	ResBlock(64)	$\times 1$	ResBlock(128)	$\times 2$	ResBlock(256)	$\times 2$	
	AvgPool(2×2)		AvgPool(2×2)		AvgPool(2×2)		
	ResBlock(128)	$\times 1$	ResBlock(256)	$\times 2$	ResBlock(256)	$\times 2$	
	AvgPool(2×2)		AvgPool(2×2)		AvgPool(2×2)		
	ResBlock(128)	$\times 1$	ResBlock(256)	$\times 2$	ResBlock(256)	$\times 2$	
	GlobalAvgPool		GlobalAvgPool		GlobalAvgPool		
	MLP($128, 2048, 128$)		MLP($256, 2048, 128$)		Concat $\to$ MLP($768, 2048, 128$)		

Training. For the EBM parameter θ, we use Adam optimizer [43] with $\beta_1 = 0$, $\beta_2 = 0.999$, and a learning rate of 10^{-4}. We use the linear learning rate warmup for the first 2k training iterations. For the encoder parameter ϕ, we use SGD optimizer with a learning rate of 3×10^{-2}, a weight decay of 5×10^{-4}, and a momentum of 0.9 as described in Chen and He [44]. For all experiments, we train our models 100k iterations with a batch size of 64, unless otherwise stated. For data augmentation $\mathcal{T}$, we follow Chen et al. [19], *i.e.*, $\mathcal{T}$ includes random cropping, flipping, color jittering, and color dropping. For hyperparameters, we use $\alpha = 1$ following Du and Mordatch [8], and $\beta = 0.01$ (see Appendix F for β-sensitivity experiments). For our large model, we use a large batch size of 256 only for learning the contrastive encoder h_ϕ. After training, we utilize exponential moving average (EMA) models for evaluation.

SGLD sampling. For each training iteration, we use 60 SGLD steps with a step size of 100 for sampling $\tilde{\mathbf{x}} \sim p_\theta$. Following Du et al. [10], we apply a random augmentation $t \sim \mathcal{T}$ for every 60 steps. We also use a replay buffer with a size of 10000 and a resampling rate of 0.1% for maintaining diverse samples [8]. For evaluation, we generate 50k images by running 600 and 1200 SGLD steps from uniform noises for our base and large models, respectively, and then we evaluate their qualities using Fréchet Inception Distance (FID) scores [45, 46].

C Unconditionally generated samples

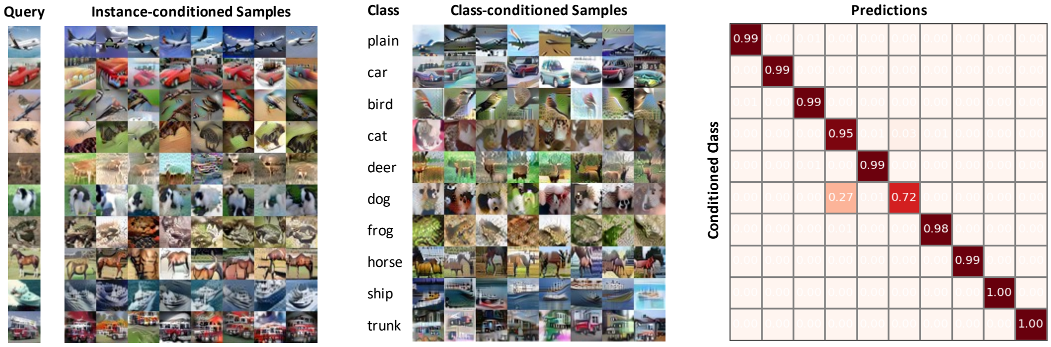


(a) CIFAR-10

(b) ImageNet 32×32

Figure 2: Unconditional generated samples from our EBMs on CIFAR-10 and ImageNet 32×32.

D Conditional sampling



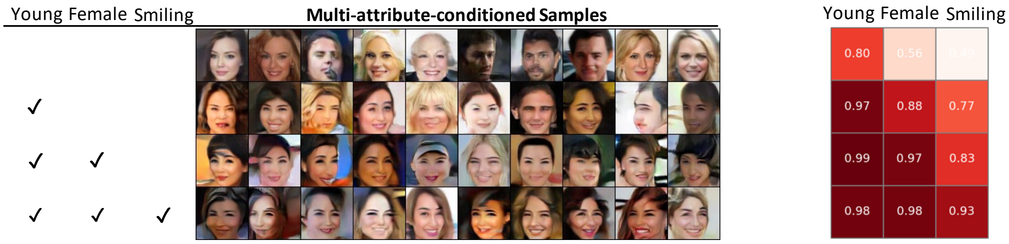
(a) Instance-conditional sampling (b) Class-conditional sampling (c) Confusion matrix

Figure 3: (a, b) Instance- and class-conditionally generated samples using our CLEL in CIFAR-10. (c) Confusion matrix for the class-conditionally generated samples computed by an external classifier.

One advantage of latent-variable EBMs is that they can offer the latent-conditional density $p_\theta(\mathbf{x}|\mathbf{z}) \propto \exp(-E_\theta(\mathbf{x},\mathbf{z}))$. Hence, our EBMs can enjoy the advantage even though CLEL does not explicitly train conditional models. To verify this, we first test *instance-conditional sampling*: given a real sample $\mathbf{x}$, we draw the underlying latent variable $\mathbf{z} \sim p_{\text{data}}(\mathbf{z}|\mathbf{x})$ using our latent encoder h_ϕ, and then perform SGLD sampling using our joint energy $E_\theta(\mathbf{x},\mathbf{z})$ defined in Section 2.2. We here use our CIFAR-10 model. As shown in Figure 3a, the instance-conditionally generated samples contain similar information (*e.g.*, color, shape, and background) to the given instance.

This successful result motivates us to extend the sampling procedure: given a set of instances $\{\mathbf{x}^{(i)}\}$, can we generate samples that contain the shared information in $\{\mathbf{x}^{(i)}\}$? To this end, we first draw latent variables $\mathbf{z}^{(i)} \sim p_{\text{data}}(\cdot|\mathbf{x}^{(i)})$ for all i, and then aggregate them by summation and normalization: $\bar{\mathbf{z}} := \sum_i \mathbf{z}^{(i)}/\|\sum_i \mathbf{z}^{(i)}\|_2$. To demonstrate that samples generated from $p_\theta(\mathbf{x}|\bar{\mathbf{z}})$ contains the shared information in $\{\mathbf{x}^{(i)}\}$, we collect the set of instances $\{\mathbf{x}_y^{(i)}\}$ for each label y in CIFAR-10, and check whether $\tilde{\mathbf{x}}_y \sim p_\theta(\cdot|\bar{\mathbf{z}}_y)$ has the same label y. Figure 3b shows the class-conditionally generated samples $\{\tilde{\mathbf{x}}_y\}$ and Figure 3c presents the confusion matrix of predictions for $\{\tilde{\mathbf{x}}_y\}$ computed by an external classifier c. Formally, each (i,j)-th entry is equal to $\mathbb{P}_{\tilde{\mathbf{x}}_i}(c(\tilde{\mathbf{x}}_i) = j)$. We found that $\tilde{\mathbf{x}}_y$ is likely to be predicted as the label y, except the case when y is dog: the generated dog images sometimes look like a semantically similar class, cat. These results verify that our EBM can generate samples conditioning on a instance or class label, even without explicit conditional training.

E Compositionality via latent variables



(a) Multi-attribute-conditional sampling

(b) Attribute predictions

Figure 4: Compositional generation results in CelebA. (a) Samples are generated by conditioning on checked attributes. (b) Attribute predictions of generated samples computed by an external classifier.

An intriguing property of EBMs is compositionality [47]: given two EBMs $E(\mathbf{x}|c_1)$ and $E(\mathbf{x}|c_2)$ that are conditional energies on concepts c_1 and c_2, respectively, one can construct a new energy conditioning on both concepts: $p_\theta(\mathbf{x}|c_1 \text{ and } c_2) \propto \exp(-E(\mathbf{x}|c_1) - E(\mathbf{x}|c_2))$. As shown in Appendix D, our CLEL implicitly learns $E(\mathbf{x}|\mathbf{z})$, and a latent variable $\mathbf{z}$ can be considered as a concept, *e.g.*, instance or class. Hence, in this section, we test compositionality of our model. To this end, we additionally train our CLEL in CelebA 64×64 [48]. For compositional sampling, we first acquire three attribute vectors $\bar{\mathbf{z}}_a$ for $a \in \mathcal{A} := \{\text{Young}, \text{Female}, \text{Smiling}\}$ as we did in Appendix D, then generate samples from a composition of conditional energies as follows:

$$E_\theta(\mathbf{x}|\mathcal{A}) := \frac{1}{2}\|f_\theta(\mathbf{x})\|_2 - \beta \sum_{a \in \mathcal{A}} \text{sim}(g_\theta(f_\theta(\mathbf{x})/\|f_\theta(\mathbf{x})\|_2), \bar{\mathbf{z}}_a), \tag{6}$$

where $\text{sim}(\cdot, \cdot)$ is the cosine similarity. Figure 4a and 4b show the generated samples conditioning on multiple attributes and their attribute prediction results computed by an external classifier, respectively. They verify our compositionality qualitatively and quantitatively. For example, almost generated faces conditioned by $\{\text{Young}, \text{Female}\}$ look young and female (see the third row in Figure 4.)

F Ablation study

Table 3: Component ablation experiments.

	Projection g_θ	Negative $p_\theta(\tilde{\mathbf{z}})$	FID↓	OOD↑
(a)	Baseline ($\beta = 0$)		42.46	0.8532
(b)	MLP		36.29	0.8580
(c)	MLP	✓	**35.73**	**0.8723**
(d)	Identity	✓	86.02	0.8474
(e)	Linear	✓	37.35	0.8540

Table 4: β sensitivity.

β	FID↓	OOD↑
0	42.46	0.8532
0.001	37.44	0.8485
0.01	**35.73**	**0.8723**
0.1	56.39	0.7559

Table 5: Compatibility.

SSRL	FID↓	OOD↑
	42.46	0.8532
SimCLR	**35.73**	0.8723
BYOL	36.31	**0.8792**
MAE	37.67	0.8561

Component analysis. To verify the importance of our CLEL's components, we conduct ablation experiments with training a smaller ResNet [41] in CIFAR-10 [36] for 50k training iterations. Then, we evaluate the quality of energy functions using FID and OOD detection scores. Here, we use SVHN [39] as the OOD dataset. Table 3 demonstrates the effectiveness of CLEL's components. First, we observe that learning $p_\theta(\mathbf{z}|\mathbf{x})$ to approximate $p_{\text{data}}(\mathbf{z}|\mathbf{x})$ plays a crucial role for improving generation (see (a) *vs.* (b)). In addition, using generated latent variables $\tilde{\mathbf{z}} \sim p_\theta(\cdot)$ as negatives for contrastive learning further improves not only generation, but also OOD detection performance (see (b) *vs.* (c)). We also empirically found that using an additional projection head is critical; without projection g_θ (*i.e.*, (d)), our EBM failed to approximate $p_{\text{data}}(\mathbf{x})$, but an additional projection head (*i.e.*, (c) or (e)) makes learning feasible. Hence, we use a 2-layer MLP (c) in all experiments since it is better than a simple linear function (e). We also test various $\beta \in \{0.1, 0.01, 0.001\}$ under this evaluation setup (see Table 4) and find $\beta = 0.01$ is the best.

Compatibility with other self-supervised representation learning methods. While we have mainly focused on utilizing contrastive representation learning (CRL), our framework CLEL is not limited to CRL for learning the latent encoder h_ϕ. To verify this compatibility, we replace SimCLR with other self-supervised representation learning (SSRL) methods, BYOL [20] and MAE [21]. Note that these methods have several advantages compared to SimCLR: e.g., BYOL does not require negative pairs, and MAE does not require heavy data augmentations. Table 5 implies that any SSRL methods can be used to improve EBMs under our framework, where the CRL method, SimCLR [19], is the best. We provide the implementation details for BYOL and MAE below.

BYOL for CLEL. Since BYOL also learns its representations on the unit sphere, the method can be directly incorporated with our CLEL framework.

MAE for CLEL. Since MAE's representations do not lie on the unit sphere, we incorporate MAE into our CLEL framework by the following procedure:

1. Pretrain a MAE framework and remove its MAE decoder. To this end, we simply use a publicly-available checkpoint of the ViT-tiny architecture.
2. Freeze the MAE encoder parameters and construct a learnable 2-layer MLP on the top of the encoder.
3. Train only the MLP via contrastive representation learning *without data augmentations* using our objective (4) for the latent encoder.