Guiding Energy-based Models via Contrastive Latent Variables

Hankook Lee¹ Jongheon Jeong¹ Sejun Park² Jinwoo Shin¹
¹Korea Advanced Institute of Science and Technology (KAIST)
²Korea 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., \( p_\theta(x) \propto \exp(-E_\theta(x)) \), 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].
We demonstrate the effectiveness of the proposed framework through extensive experiments. For example, our spherical latent-variable EBM (\(f_\theta, g_\theta\)) learns the joint data distribution \(p_{\text{data}}(x, z)\) generated by our contrastive latent encoder \(h_\phi\). (b) The encoder \(h_\phi\) is trained by contrastive learning with additional negative variables \(z \sim p_\theta(\bar{z})\).

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 \(p_{\text{data}}(z|x)\). Specifically, we train an encoder \(h_\phi\) via CRL, and treat the encoded representation \(z := h_\phi(x)\) as the true latent variable given data \(x\), i.e., \(z \sim p_{\text{data}}(z|x)\). 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., \(\|z\|_2 = 1\), 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 \(p_\theta(x, z)\) for modeling the joint distribution \(p_{\text{data}}(x, z)\) generated by the contrastive latent encoder. Since the latent variables are spherical, we separate the output vector \(f := f_\theta(x)\) into its norm \(||f||_2\) and direction \(f/||f||_2\) for modeling \(p_\theta(x)\) and \(p_\theta(z|x)\), respectively. We found that this separation technique reduces the conflict between \(p_\theta(x)\) and \(p_\theta(z|x)\) optimizations, which makes training stable. In addition, we treat the latent variables drawn from our EBM, \(\bar{z} \sim p_\theta(z)\), 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\(\rightarrow\)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) \(p_\theta(x) \propto \exp(-E_\theta(x))\) to approximate a complex underlying data distribution \(p_{\text{data}}(x)\). 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 \(z \sim p_{\text{data}}(z|x)\) of \(x\), generated by contrastive learning, as the true underlying latent variable. Namely, we model the joint distribution \(p_{\text{data}}(x, z) = p_{\text{data}}(x)p_{\text{data}}(z|x)\) via a latent-variable EBM \(p_\theta(x, z)\). 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 \( \mathbb{R}^d \), defined as follows: for data \( x \in \mathbb{R}^d \), \( p_\theta(x) = \exp(-E_\theta(x))/Z_\theta \) where \( Z_\theta = \int_{\mathbb{R}^d} \exp(-E_\theta(x)) dx \) denotes the normalizing constant, called the partition function. An important application of EBMs is to find a parameter \( \theta \) such that \( p_\theta \) is close to \( p_{\text{data}} \). A popular method for finding such \( \theta \) is to minimize Kullback–Leibler (KL) divergence between \( p_{\text{data}} \) and \( p_\theta \) via gradient descent:

\[
\nabla_\theta \mathbb{D}_{\text{KL}}(p_{\text{data}} \parallel p_\theta) = \mathbb{E}_{x \sim p_{\text{data}}} [\nabla_\theta E_\theta(x)] - \mathbb{E}_{x \sim p_\theta} [\nabla_\theta E_\theta(x)].
\]

(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]:

\[
\mathcal{L}_{\text{NT-Xent}}(z, z_+, z_-; \tau) = -\log \frac{\exp(\text{sim}(z, z_+)/\tau)}{\exp(\text{sim}(z, z_+)/\tau) + \sum_{z_-} \exp(\text{sim}(z, z_-)/\tau)},
\]

(2)

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

2.2 CLEL: Contrastive Latent-guided Energy Learning

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

**Spherical latent-variable energy-based models.** We use a DNN \( f_\theta : \mathbb{R}^d \rightarrow \mathbb{R}^d \) parameterized by \( \theta \) for modeling \( p_\theta(x, z) \). Following that the latent variable \( z \sim p_{\text{data}}(z|x) \) is on the unit sphere, we utilize the directional information \( f_\theta(x/||f_\theta(x)||_2) \) for modeling \( p_\theta(x, z) \), while the remaining information \( ||f_\theta(x)||_2 \) is used for modeling \( p_\theta(x) \). Overall, we define the joint energy \( E_\theta(x, z) \) by

\[
E_\theta(x, z) := \frac{1}{2} ||f_\theta(x)||^2_2 - \beta g_\theta \left( \frac{f_\theta(x)}{||f_\theta(x)||_2} \right)^T z
\]

where \( \beta \geq 0 \) is a hyperparameter and \( g_\theta : \mathbb{S}^{d-1} \rightarrow \mathbb{S}^{d-1} \) is a directional projection MLP. Note that the marginal energy \( E_\theta(x) \) only depends on \( ||f_\theta(x)||_2 \) since \( \int_{\mathbb{S}^{d-1}} \exp(\beta g_\theta(f_\theta(x)/||f_\theta(x)||_2)^T z) dz \) is independent of \( x \) due to the symmetry.

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

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

(3)

\[
\mathcal{L}_{\text{LE}}(\mathcal{B}, \tilde{\mathcal{B}}; \phi, \tau) = \frac{1}{2hn} \sum_{i=1}^{n} \sum_{j=1,2} \mathcal{L}_{\text{NT-Xent}}(z_j^{(i)}, z_{3-j}^{(i)}, \{z_l^{(i)}\}_{k \neq i,l \in \{1,2\}} \cup \{\tilde{z}^{(i)}\}_{i=1}^n; \tau),
\]

(4)

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

---

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

2Here, \( \tilde{z} \) is unnecessary for \( D_{\text{KL}}(p_{\text{data}} \parallel p_\theta) \) since \( \mathbb{E}_{z \sim p_\theta(x)} \mathbb{E}_{z \sim p_\theta(x, \tilde{z})} [\nabla_\theta E_\theta(x, \tilde{z})] = \nabla_\theta E_\theta(x) \).
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 out-of-distribution detection, especially when OOD samples are drawn from different domains, e.g., SVHN [39], CIFAR-10, and Texture [40], respectively. The unconditional 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 \( f \). As a result, we achieve 8.61 FID on CIFAR-10, which is lower than that of the prior-art EBM based on diffusion learning (CRL) with a larger batch, more channels at lower layers in our EBM.

### Conditional generation using \( p_{\theta}(x|z) \)

Even without explicit conditional training, our latent-variable EBMs naturally can provide the latent-conditional density \( p_{\theta}(x|z) \). 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


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


A Training procedure of CLEL

Algorithm 1: Contrastive Latent-guided Energy Learning (CLEL)

Require: a latent-variable EBM \( f_\theta \), a latent encoder \( h_\phi \), an augmentation distribution \( \mathcal{T} \), hyperparameters \( \alpha, \beta, \tau > 0 \), and the stop-gradient operation \( \text{sg}(.) \).

1: for \# training iterations do
2: // Construct batches \( B \) and \( \tilde{B} \)
3: Sample \( \{x^{(i)}\}_{i=1}^n \sim p_{\text{data}}(x) \)
4: Sample \( \{\tilde{x}^{(i)}\}_{i=1}^n \sim p_{\text{data}}(x) \) using stochastic gradient Langevin dynamics (SGLD)
5: \( z^{(i)} \leftarrow \text{sg}(h_\phi(t^{(i)}(x^{(i)}))/||h_\phi(t^{(i)}(x^{(i)}))||_2), t^{(i)} \sim \mathcal{T} \)
6: \( \tilde{z}^{(i)} \leftarrow \text{sg}(g_\theta(f_\theta(\tilde{x}^{(i)}))/||f_\theta(\tilde{x}^{(i)}))||_2) \)
7: // Compute the EBM loss, \( \mathcal{L}_{EBM} \)
8: \( \mathcal{L}_{EBM} \leftarrow \frac{1}{n} \sum_{i=1}^n E_\theta(x^{(i)}, z^{(i)}) - E_\theta(\tilde{x}^{(i)}) + \alpha \cdot (E_\theta(x^{(i)})^2 + E_\theta(\tilde{x}^{(i)})^2) \)
9: // Compute the encoder loss, \( \mathcal{L}_{LE} \)
10: \( z_i^{(i)} \leftarrow h_\phi(t_j^{(i)}(x^{(i)})), t_j^{(i)} \sim \mathcal{T} \)
11: \( \mathcal{L}_{LE} \leftarrow \frac{1}{2\tau} \sum_{i=1}^n \sum_{j=1,2} \mathcal{L}_{\text{NT-Xent}}(z_i^{(i)}, z_{3-j}^{(i)}; \{z_j^{(k)}\}_{k \neq i, j} \cup \{\tilde{z}^{(i)}\}_{i=1}^n; \tau) \)
12: Update \( \theta \) and \( \phi \) to minimize \( \mathcal{L}_{EBM} + \mathcal{L}_{LE} \)
13: end for

B Training details

Architectures. For the spherical latent-variable energy-based model (EBM) \( f_\theta \), 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. For the projection \( g_\theta \), we use a 2-layer MLP with a output dimension of 128, the leakyReLU activation, and no bias, i.e., \( g_\theta(u) = W_2 \sigma(W_1 u) \in \mathbb{R}^{128} \). For the latent encoder \( h_\phi \), we simply use the CIFAR variant of ResNet-18 [41], followed by a 2-layer MLP with a output dimension of 128.

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

<table>
<thead>
<tr>
<th>Input</th>
<th>Small ((3,32,32))</th>
<th>Base ((3,32,32))</th>
<th>Large ((3,32,16,3,16,3,8,8))</th>
</tr>
</thead>
<tbody>
<tr>
<td>EBM ( f_\theta(x) )</td>
<td>Conv(3 \times 3, 64) (\times 1)</td>
<td>Conv(3 \times 3, 128)</td>
<td>Conv(3 \times 3, 256)</td>
</tr>
<tr>
<td></td>
<td>ResBlock(64)</td>
<td>ResBlock(128)</td>
<td>ResBlock(256)</td>
</tr>
<tr>
<td></td>
<td>AvgPool(2 \times 2) (\times 1)</td>
<td>AvgPool(2 \times 2)</td>
<td>AvgPool(2 \times 2)</td>
</tr>
<tr>
<td></td>
<td>ResBlock(64)</td>
<td>ResBlock(128)</td>
<td>ResBlock(256)</td>
</tr>
<tr>
<td></td>
<td>AvgPool(2 \times 2) (\times 1)</td>
<td>AvgPool(2 \times 2)</td>
<td>AvgPool(2 \times 2)</td>
</tr>
<tr>
<td></td>
<td>ResBlock(128)</td>
<td>ResBlock(256)</td>
<td>ResBlock(256)</td>
</tr>
<tr>
<td></td>
<td>AvgPool(2 \times 2) (\times 1)</td>
<td>AvgPool(2 \times 2)</td>
<td>AvgPool(2 \times 2)</td>
</tr>
<tr>
<td></td>
<td>ResBlock(128)</td>
<td>ResBlock(256)</td>
<td>ResBlock(256)</td>
</tr>
<tr>
<td></td>
<td>GlobalAvgPool</td>
<td>GlobalAvgPool</td>
<td>GlobalAvgPool</td>
</tr>
<tr>
<td></td>
<td>MLP({128, 2048, 128})</td>
<td>MLP({256, 2048, 128})</td>
<td>Concat (\rightarrow) MLP({768, 2048, 128})</td>
</tr>
</tbody>
</table>

Training. For the EBM parameter \( \theta \), 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 \( \phi \), we use SGD optimizer with a learning rate of \(3 \times 10^{-2}\), a weight decay of \(5 \times 10^{-4}\), 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 $T$, we follow Chen et al. [19], i.e., $T$ includes random cropping, flipping, color jitting, and color dropping. For hyperparameters, we use $\alpha = 1$ following Du and Mordatch [8], and $\beta = 0.01$ (see Appendix F for $\beta$-sensitivity experiments). For our large model, we use a large batch size of 256 only for learning the contrastive encoder $h_\phi$. 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 $\tilde{x} \sim p_0$. Following Du et al. [10], we apply a random augmentation $t \sim T$ 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

![Generated Samples](image.png)

(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

![Diagram](image)

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 $p_\theta(x|z) \propto \exp(-E_\theta(x, z))$. 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 $x$, we draw the underlying latent variable $z \sim p_{\text{data}}(z|x)$ using our latent encoder $h_\phi$, and then perform SGLD sampling using our joint energy $E_\theta(x, z)$ 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 $\{x^{(i)}\}$, can we generate samples that contain the shared information in $\{x^{(i)}\}$? To this end, we first draw latent variables $z^{(i)} \sim p_{\text{data}}(z|x^{(i)})$ for all $i$, and then aggregate them by summation and normalization: $\bar{z} := \frac{\sum_i z^{(i)}}{\|\sum_i z^{(i)}\|_2}$. To demonstrate that samples generated from $p_\theta(x|\bar{z})$ contains the shared information in $\{x^{(i)}\}$, we collect the set of instances $\{x^{(i)}_y\}$ for each label $y$ in CIFAR-10, and check whether $\tilde{x}_y \sim p_\theta(\cdot|\bar{z}_y)$ has the same label $y$. Figure 3b shows the class-conditionally generated samples $\{\tilde{x}_y\}$ and Figure 3c presents the confusion matrix of predictions for $\{\tilde{x}_y\}$ computed by an external classifier $c$. Formally, each $(i, j)$-th entry is equal to $\mathbb{P}_{\tilde{x}_y}(c(\tilde{x}_y) = j)$. We found that $\tilde{x}_y$ 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

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 $E(x|c_1)$ and $E(x|c_2)$ that are conditional energies on concepts $c_1$ and $c_2$, respectively, one can construct a new energy conditioning on both concepts: $p_\theta(x|c_1 \text{ and } c_2) \propto \exp(-E(x|c_1) - E(x|c_2))$. As shown in Appendix D, our CLEL implicitly learns $E(x|z)$, and a latent variable $z$ 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 $64 \times 64$ [48]. For compositional sampling, we first acquire three attribute vectors $\bar{z}_a$ for $a \in A := \{\text{Young}, \text{Female}, \text{Smiling}\}$ as we did in Appendix D, then generate samples from a composition of conditional energies as follows:

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

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>
<thead>
<tr>
<th>Projection $g_\theta$</th>
<th>Negative $p_\theta(\tilde{z})$</th>
<th>FID↓</th>
<th>OOD↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) Baseline ($\beta = 0$)</td>
<td>42.46</td>
<td>0.8532</td>
<td></td>
</tr>
<tr>
<td>(b) MLP</td>
<td>36.29</td>
<td>0.8580</td>
<td></td>
</tr>
<tr>
<td>(c) MLP ✓</td>
<td>35.73</td>
<td>0.8723</td>
<td></td>
</tr>
<tr>
<td>(d) Identity ✓</td>
<td>86.02</td>
<td>0.8474</td>
<td></td>
</tr>
<tr>
<td>(e) Linear ✓</td>
<td>37.35</td>
<td>0.8540</td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Component ablation experiments.

<table>
<thead>
<tr>
<th>$\beta$</th>
<th>FID↓</th>
<th>OOD↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>42.46</td>
<td>0.8532</td>
</tr>
<tr>
<td>0.001</td>
<td>37.44</td>
<td>0.8485</td>
</tr>
<tr>
<td>0.01</td>
<td>35.73</td>
<td>0.8723</td>
</tr>
<tr>
<td>0.1</td>
<td>56.39</td>
<td>0.7559</td>
</tr>
</tbody>
</table>

Table 4: $\beta$ sensitivity.

<table>
<thead>
<tr>
<th>SSRL</th>
<th>FID↓</th>
<th>OOD↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>SimCLR</td>
<td>35.73</td>
<td>0.8723</td>
</tr>
<tr>
<td>BYOL</td>
<td>36.31</td>
<td>0.8792</td>
</tr>
<tr>
<td>MAE</td>
<td>37.67</td>
<td>0.8561</td>
</tr>
</tbody>
</table>

Table 5: Compatibility.

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 $p_\theta(z|x)$ to approximate $p_{data}(z|x)$ plays a crucial role for improving generation (see (a) vs. (b)). In addition, using generated latent variables $\tilde{z} \sim p_\theta(\cdot)$ 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 $g_\theta$ (i.e., (d)), our EBM failed to approximate $p_{data}(x)$, 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 $h_\phi$. 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.