Received: 27 April 2023 | Revised: 1 August 2023 | Accepted: 23 August 2023

DOI: 10.1002/sta4.625

ORIGINAL ARTICLE

WILEY

Persistently trained, diffusion-assisted energy-based models

Xinwei Zhang[1] | Zhiqiang Tan[2] | Zhijian Ou[3]

[1]Department of Biostatistics, New York University, New York, New York, USA

[2]Department of Statistics, Rutgers University, Piscataway, New Jersey, USA

[3]Department of Electronic Engineering, Tsinghua University, Beijing, China

Correspondence
Zhiqiang Tan, Department of Statistics, Rutgers University, Piscataway, NJ, 08854, USA.
Email: ztan@stat.rutgers.edu

Zhijian Ou, Department of Electronic Engineering, Tsinghua University, Beijing, 100084, China.
Email: ozj@tsinghua.edu.cn

Maximum likelihood (ML) learning for energy-based models (EBMs) is challenging, partly due to nonconvergence of Markov chain Monte Carlo. Several variations of ML learning have been proposed, but existing methods all fail to achieve both post-training image generation and proper density estimation. We propose to introduce diffusion data and learn a joint EBM, called diffusion-assisted EBMs, through persistent training (i.e. using persistent contrastive divergence) with an enhanced sampling algorithm to properly sample from complex, multimodal distributions. We present results from a 2D illustrative experiment and image experiments and demonstrate that for the first time for image data, persistently trained EBMs can *simultaneously* achieve long-run stability, post-training image generation and superior out-of-distribution detection.

KEYWORDS
energy-based models (EBMs), out-of-distribution detection, score-based diffusion models

1 | INTRODUCTION

Energy-based models (EBMs) parameterise an unnormalised data density or equivalently an energy function, defined as the negative log density up to an additive constant. There has been persistent and ongoing interest in developing effective modeling and training techniques for learning EBMs from complex data such as natural images. A partial list of examples include LeCun et al. (2006), Xie et al. (2016), Du and Mordatch (2019), Nijkamp et al. (2019) and Grathwohl et al. (2020). Apart from maximum likelihood (ML) learning, various other training principles are available, including score matching (Hyvärinen, 2005; Saremi et al., 2018), noise-contrastive estimation (Gutmann & Hyvärinen, 2012) and f-divergence minimisation (Yu et al., 2020). See Table 1 in Grathwohl et al. (2021) for a comparison.

We aim to advance ML learning for EBMs. Several variations of ML learning have been proposed, and impressive performances have been reported (Du & Mordatch, 2019; Gao et al., 2021). See Section 2 for a discussion of initialisation schemes for Markov chain Monte Carlo (MCMC) during training, including persistent, data, noise and hybrid initialisations. However, training EBMs remains challenging and complicated by conflicting ideas. As shown in Table 1, existing training methods all fail to achieve at least one of the desired learning properties, defined as follows.

- Long-run stability (long-run): Long-run MCMC samples generated (after training) using the learned energy starting from *real images* remain realistic.
- Post-training sampling (post): MCMC samples generated (after training) using the learned energy starting from *random noises* are realistic.
- Global energy estimation: The learned energy function is globally aligned between different modes separated by low-density regions. A learned energy may lead to long-run stability but *only* be locally meaningful (i.e. accurate near local modes). See Figure 1.

For the noise initialisation method, the learned short-run MCMC can be used to generate realistic images from random noises similarly in GAN (Goodfellow et al., 2014) or Glow (Kingma & Dhariwal, 2018), but the learned energy functions seem to be invalid due to the lack of

TABLE 1 Comparison of training methods for EBMs.

	Long-run	Post	Global energy
EBM Pers.	✓	×	×
EBM CD	×	×	×
EBM Noise	×	✓	×
EBM Hybrid	×	✓	×
Dif. Rec. Lik. (DRL)	✓/×[a]	✓	×
DA-EBM (ours)	✓	✓	✓

Abbreviations: DRL, diffusion recovery likelihood; EBM, energy-based model.
[a]Mixed results are obtained in our experiments.

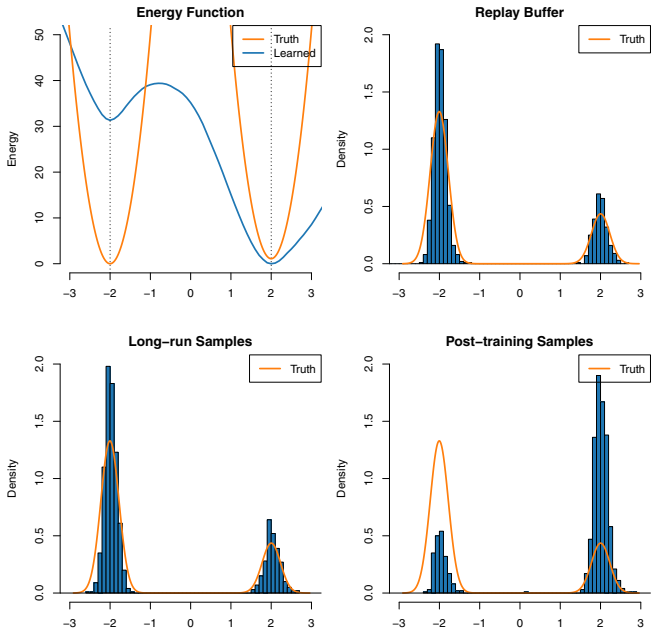


FIGURE 1 Results from 1D example trained by persistent-initialised energy-based model (EBM).

long-run stability (Nijkamp et al., 2020). Persistent training can be implemented using certain training strategies to produce stable long-run samples from real data (Nijkamp et al., 2020), but this approach has not been successful in generating new realistic images from random noises after training.

Current applications of EBMs to out-of-distribution (OOD) detection (Du & Mordatch, 2019; Grathwohl et al., 2020) leave much room for improvement due to the OOD reversal phenomenon, where higher likelihoods are assigned to OOD observations than in-distribution (InD) observations. Such phenomena are observed in both flow-based deep generative modeling (Choi et al., 2019; Nalisnick et al., 2019) and usual and joint EBMs (Grathwohl et al., 2020). A possible explanation and remedy were proposed in Nalisnick et al. (2019), using the concept of typicality (Cover & Thomas, 2006). An alternative hypothesis underlying our work is that the OOD reversal may occur because the estimated energies (or log-likelihoods) are locally meaningful but not globally aligned, so that higher likelihoods may be assigned to OOD data points near one mode than InD data points near another mode. See the 1D example in Section 3.1.

In this work, we make three main contributions. First, we identify and explain, for the first time, the phenomenon that for complex, multimodal data distributions, persistent training using an MCMC sampler which suffers local mixing may only learn local energy functions, which are locally meaningful but globally misaligned. Second, motivated by this understanding and inspired by recent success of score-based diffusion modeling, we propose to introduce diffusion data through a forward diffusion process and learn a joint EBM, called diffusion-assisted EBMs (DA-EBMs), from both the original training data and diffusion data at different time steps. We pursue persistent training while incorporating an enhanced sampling algorithm to overcome local mixing (Tan, 2017). See Section 4 for a comparison of our method with score-based diffusion modeling (Ho et al., 2020; Sohl-Dickstein et al., 2015; Song & Ermon, 2019; Song et al., 2021) and diffusion recovery likelihood (DRL) (Gao et al., 2021). Third, we present results from a 2D illustrative experiment and image experiments on MNIST-type data and demonstrate that, for the first time for image data, persistently trained EBMs can *simultaneously* achieve long-run stability, post-training image generation and superior OOD

detection, indicating that the learned energy is globally more meaningful than previously obtained. Our code is available online (https://github.com/xinweizhang/daebm).

2 | BACKGROUND: EBMS

An EBM is defined in the form

$$p_\theta(x) = \frac{\exp(-U_\theta(x))}{Z(\theta)}, \tag{1}$$

where $U_\theta(x)$ is an energy (or potential) function, for example, specified as a neural network with parameter θ and $Z(\theta) = \int \exp(-U_\theta(x))dx$ is the normalising constant. Traditionally, EBMs are used for unsupervised image modeling, where x denotes an image configuration (Du & Mordatch, 2019; LeCun et al., 2006; Nijkamp et al., 2019; Xie et al., 2016). Recently, EBMs are also considered in supervised or semisupervised settings, where x is a pair of an image configuration and a class label (Grathwohl et al., 2020; Song & Ou, 2018). Such models are called joint EBMs.

Statistically, EBMs can be trained by ML. Given training data $\mathcal{D} = \{z_i\}_{i=1}^n$, the gradient of the log-likelihood $l(\theta) = \frac{1}{n}\sum_{i=1}^n \log p_\theta(z_i)$ is

$$\frac{\partial}{\partial\theta} l(\theta) = -\mathbb{E}_{\hat{p}}\left\{\frac{\partial}{\partial\theta} U_\theta(x)\right\} + \mathbb{E}_{p_\theta}\left\{\frac{\partial}{\partial\theta} U_\theta(x)\right\}, \tag{2}$$

where $\hat{p}$ denotes the empirical distribution on $\mathcal{D}$ and $\mathbb{E}_q(\cdot)$ denotes the expectation with respect to q. The ML estimator of θ is obtained as a solution to $\frac{\partial}{\partial\theta} l(\theta) = 0$.

However, a challenge is that the expectation $\mathbb{E}_{p_\theta}\{\frac{\partial}{\partial\theta} U_\theta(x)\}$ in (2) is usually difficult to evaluate. This issue can be addressed through stochastic approximation (SA), which provides a principled and flexible approach for numerically solving intractable equations (Benveniste et al., 1990; Robbins & Monro, 1951).

Given current value θ_0 and replay-buffer $\mathcal{B}$ (which stores current synthetic data), an SA iteration for ML estimation (ML-SA) performs the following operations.

- Sampling: Draw x_1 from $\mathcal{D}$, $\tilde{x}_0$ from $\mathcal{B}$ and $\tilde{x}_1$ from a Markov transition kernel $K_{\theta_0}(\tilde{x}_0, \cdot)$, where $K_\theta(\cdot,\cdot)$ is assumed to leave p_θ invariant (among other technical conditions); reset $\tilde{x}_0$ to $\tilde{x}_1$ in $\mathcal{B}$.
- Updating: Update θ by gradient ascent as

$$\theta_1 = \theta_0 + \gamma\left\{-\frac{\partial}{\partial\theta} U_\theta(x_1) + \frac{\partial}{\partial\theta} U_\theta(\tilde{x}_1)\right\}|_{\theta=\theta_0}, \tag{3}$$

where γ is a learning rate.

The two operations are also called synthesis and analysis (Xie et al., 2016). The Markov transition from $\tilde{x}_0$ to $\tilde{x}_1$ can be defined by multiple (more elementary) sampling steps such as (4) below. Moreover, multiple observations can be allowed in each iteration by drawing a mini-batch of real observations from $\mathcal{D}$ and running multiple parallel chains to draw new synthetic observations and returning them to $\mathcal{B}$.

There are two important aspects of the sampling operation, where different choices may lead to variations related to but distinct from the ML-SA algorithm. One is the choice of the Markov transition $K_\theta(\cdot,\cdot)$, represented by a MCMC algorithm.

A popular MCMC algorithm is Langevin sampling. Given current observation $\tilde{x}_0$, a new observation is proposed as

$$\tilde{x}_{1/2} = \tilde{x}_0 - (\sigma^2/2)\nabla_x U_{\theta_0}(\tilde{x}_0) + \sigma\varepsilon, \tag{4}$$

where $\varepsilon \sim \mathrm{N}(0, I)$ is a Gaussian noise and σ is a step size. The next observation $\tilde{x}_1$ may directly be the proposal $\tilde{x}_{1/2}$, in which case the Markov transition from $\tilde{x}_0$ to $\tilde{x}_1$ does not strictly leave p_{θ_0} invariant, but the sampling bias may usually be small for $\sigma \approx 0$. To allow large σ, a correction can be achieved by accepting or rejecting the proposal $\tilde{x}_{1/2}$, that is, setting $\tilde{x}_1 = \tilde{x}_{1/2}$ or $\tilde{x}_0$, with the Metropolis–Hastings probability. Langevin sampling with rejection is known as the Metropolis-Adjusted Langevin Algorithm (MALA) (Besag, 1994; Roberts & Tweedie, 1996).

The other choice in the sampling operation is the initialisation scheme, that is, the choice $\tilde{x}_0$ used in $K_{\theta_0}(\tilde{x}_0, \cdot)$ above. We summarise existing initialisation schemes below. Note that the initialisation here refers to how the starting value is chosen for the Markov transition during training, not how the starting value is chosen in long-run MCMC sampling after training or in post-training image generation.

- Persistent initialisation: Markov chains are started from past synthetic observations in the previous training iteration. This is the rule prescribed in the ML-SA algorithm and also known as persistent contrastive divergence (PCD) (Tieleman, 2008). Nijkamp et al. (2020) discussed various tuning choices such as the Langevin step size σ and learning rate γ to obtain stable long-run samples for persistent training.
- Data initialisation: Hinton (2002) proposed contrastive divergence (CD), where Markov chains are initialised by real training data and run for several steps to obtain synthetic data. Gao et al. (2021) used Markov chains initialised by noise-added real data to train EBMs through recovery likelihood.
- Noise initialisation: Nijkamp et al. (2019, 2020) studied training EBMs with short-run MCMC, where Markov chains are always started from random noises and run for a fixed number of Langevin steps (4).
- Hybrid initialisation: Several works employed a hybrid of persistent initialisation and noise initialisation, that is, initialising Markov chains either by past synthetic observations at a certain rate (e.g. 95%) or in the remaining time by random noises (Du & Mordatch, 2019; Grathwohl et al., 2020).
- Ancillary generator: Because MCMC sampling may be computationally costly and inefficient, several works proposed training an ancillary generator to initialise Markov chains to reduce MCMC steps needed or even directly generate synthetic samples to avoid MCMC (Dai et al., 2019; Grathwohl et al., 2021; Kim & Bengio, 2016; Song & Ou, 2018; Xie et al., 2018).

We point out that from our experiment results (Sections 5 and 6), hybrid initialisation (although previously treated as a close variation of PCD) behaves similarly to noise initialisation rather than to persistent initialisation.

3 | PROPOSED METHOD

In spite of recent progress, learning EBMs for complex data like natural images remains challenging with various dilemmas (see Table 1). There are two aims in our investigation: (i) to further study the behaviour of persistent training and (ii) to develop a new method by leveraging diffusion data for persistent training to achieve satisfactory learning outcomes in several aspects *simultaneously*, including long-run stability, post-training image generation and OOD detection.

3.1 | Diagnosis: Local mixing and local energy

We present a simple but informative 1D example. The training data consist of 750 observations from $\mathrm{N}(-2, .1^2)$ and 250 observations from $\mathrm{N}(2, .1^2)$. The EBM is specified by $U_\theta(x) = (x/10)^2 + \theta^\top h(x)$, where $h(x)$ consists of ReLU basis functions centred at the equi-spaced knots by 0.1 from -4 to 4. Figure 1 presents the results from persistent training using MALA (see Appendix V.1 for more details). We observe the following patterns. (i) The replay-buffer data resemble the training data, including 3:1 proportions near the two modes -2 and 2. (ii) The long-run sample (MALA starting from real data) also resemble the training data. (iii) The post-training sample (MALA starting from standard Gaussian noises) shows two local modes at -2 and 2 but with proportions different from 3:1 as in the training data. (iv) The learned energy function exhibits two local modes about -2 and 2 but is globally misaligned. The estimated energy at the local mode -2 is substantially higher than at the other mode 2 and hence is also higher than (e.g. at $x = 1$, an OOD point near no real data). More troubling is that the estimated energies near -2 and 2 may reverse the direction of relative magnitudes from different training runs, as shown in Appendix V.1. These results not only confirm the previous findings about long-run stability for persistent training (Nijkamp et al., 2020) but also reveal a new phenomenon that the learned energies from persistent training appear to be locally meaningful but globally misaligned. Hence, application of such learned energies to OOD detection is problematic.

Motivated by the 1D example, we provide some new theoretical understanding of persistent training of EBMs with MCMC sampling which enables only local mixing for multimodal distributions with modes separated by low-density (i.e. high-energy) barriers. Our discussion is heuristic but highlights the main ideas which can be exploited to develop formal analysis. In the limit of the persistent training process (assumed to exist), let $\hat{\theta}$ be a limit value of the network parameter θ and $\hat{q}$ be a limit distribution for the synthetic data (which can be represented by the empirical distribution of the replay-buffer). Then we expect that $(\hat{\theta}, \hat{q})$ satisfy the following stationarity conditions:

- Sampling stationarity: $\hat{q}$ is invariant under the Markov transition $K_{\hat{\theta}}(\cdot, \cdot)$, that is,

$$\int K_{\hat{\theta}}(x,\cdot)\hat{q}(x)\,dx = \hat{q}(\cdot). \tag{5}$$

- Parameter stationarity: $\hat{\theta}$ is a stationary point of the expected gradient, that is,

$$0 = -\mathbb{E}_{\hat{p}}\left\{\frac{\partial}{\partial\theta}U_{\hat{\theta}}(x)\right\} + \mathbb{E}_{\hat{q}}\left\{\frac{\partial}{\partial\theta}U_{\hat{\theta}}(x)\right\}. \tag{6}$$

Conversely, if the training process is initialised by any network parameter $\hat{\theta}$ and replay-buffer distribution $\hat{q}$ satisfying (5) and (6), then the network parameter and replay-buffer distribution are expected to stay as $\hat{\theta}$ and $\hat{q}$, respectively, during persistent training. Therefore, any pair $(\hat{\theta},\hat{q})$ satisfying (5) and (6) can potentially be the limit values from the training process, unless additional constraints are introduced.

From our numerical experiments (Sections 5 and 6) as well as the 1D example, we postulate that persistent training involves the following mechanisms through (5) and (6)

- Equation (5) dictates the network parameter $\hat{\theta}$ such that the transition kernel $K_{\hat{\theta}}$ (depending on the energy function $U_{\hat{\theta}}$) leaves the replay-buffer distribution $\hat{q}$ invariant.
- Equation (6) induces moment matching between $\hat{p}$ and $\hat{q}$ such that the replay-buffer distribution $\hat{q}$ resembles the real data distribution $\hat{p}$.

Note that if an MCMC sampler suffers local mixing for a target distribution with separated modes by low-density barriers, then there is no unique invariant distribution. In this setting, the relative weights between the modes may be arbitrary for samples from Langevin dynamics as discussed in Song and Ermon (2019), Section 3.2.2. See Appendix II for further discussion. To see how this sampling deficiency affects energy estimation, our interpretation of (5) and (6) proceeds as follows. The network parameter $\hat{\theta}$ is learned such that the transition kernel $K_{\hat{\theta}}$ leaves invariant the replay-buffer distribution $\hat{q}$ (which resembles the data distribution $\hat{p}$). Given sufficient data and model capacity, it can be assumed that $\hat{q} \approx p^*$, where p^* is the population version of $\hat{p}$. Then $\hat{\theta}$ can be any parameter value such that the global invariance holds approximately:

$$\int K_{\hat{\theta}}(x,\cdot)p^*(x)\,dx \approx p^*(\cdot). \tag{7}$$

In Appendix I, we show that if the transition kernel $K_{\hat{\theta}}$ enables only local mixing with low-density barriers in p^*, then (7) can be satisfied nonuniquely, provided $U_{\hat{\theta}}$ is a *local energy function* which matches the (global) energy function for p^* locally in separate regions up to possibly different additive constants. Nonuniqueness of invariant distributions for sampling can be translated into nonuniqueness of energy functions learned. An example of such local energy functions is the learned energy in Figure 1. See Appendix I for a formal discussion of local energy functions and Song and Ermon (2019), Section 3.2.1, for a related discussion about inaccurate score estimation in low-density regions.

In summary, we provide both numerical evidence and theoretical explanation for the phenomenon that in the presence of separated modes by low-density barriers, persistent training using an MCMC sampler which suffers local mixing may only learn local energy functions, which are locally meaningful but globally misaligned. Stable long-run MCMC samples from real data using such learned energy functions may be obtained, but this alone does not imply the (global) validity of the learned energies.

3.2 | Diffusion-assisted EBMs

From Section 3.1, we see that proper learning of energy functions for multimodal data requires strategies to overcome local mixing in MCMC sampling from multimodal model distributions. Better global sampling will likely make the learned energy function more globally aligned. A direct approach is to tackle this problem only within the sampling operation by exploiting *enhanced sampling* techniques such as serial tempering (Geyer & Thompson, 1995; Marinari & Parisi, 1992) and parallel tempering (Geyer, 1991; Swendsen & Wang, 1986) with tempered distributions (see Appendix III).

However, a drawback of this approach is that samples from the tempered distributions do not contribute to updating network parameters (because no empirical data are modeled by the tempered distributions).

Alternatively, we realise that diffusion data created with multiple noise levels can also be used as auxiliary distributions to bridge different modes in multimodal data or 'fill low-density regions', as noted in Song and Ermon (2019) for score estimation. We propose diffusion-assisted EBMs and develop an effective algorithm for ML learning. Instead of using tempered distributions, our approach exploits diffusion data and their model distributions for both sampling and parameter learning in an interdependent manner. The diffusion data are used together with the original

data to learn EBMs at multiple noise levels, and the learned densities of diffusion data are then used, similarly as tempered distributions, to facilitate enhanced sampling.

First, we construct diffused real data as in Ho et al. (2020). For an observation z in the training set $\mathcal{D}$ and $t = 1,...,T$, let $z^{(t)} = \sqrt{\alpha_t} z^{(t-1)} + \sqrt{1-\alpha_t}\varepsilon^{(t)}$, where $z^{(0)} = z$, $\varepsilon^{(t)} \sim \mathrm{N}(0, I)$ independently over t, and $\alpha_1, ..., \alpha_T \in (0,1)$ are prespecified noise variances such that $z^{(T)}$ is approximately distributed as $\mathrm{N}(0, I)$. Equivalently, $z^{(t)}$ can be directly generated from z as

$$z^{(t)} \sim \mathrm{N}\left(\sqrt{\bar{\alpha}_t} z, (1-\bar{\alpha}_t) I\right), \tag{8}$$

where $\bar{\alpha}_t = \prod_{j=1}^{t} \alpha_j$ for $t \geq 1$ and $\bar{\alpha}_0 = 1$. We say that $z^{(t)}$ is a diffusion observation at time t given z, and the (marginal) distribution of $z^{(t)}$ is the diffusion data distribution at time t if z is randomly drawn from $\mathcal{D}$.

We propose to model the original data and diffusion data simultaneously through a joint EBM:

$$p_\theta(x,t) = \frac{\exp(-U_\theta(x,t))}{Z(\theta)}, \tag{9}$$

where $U_\theta(x,t)$ is an energy function in (x,t) jointly and $Z(\theta) = \int \sum_{t=0}^{T} \exp(-U_\theta(x,t))\mathrm{d}x$. The pair (x,t) encodes that x is treated as a diffusion observation at time t. For any fixed t, the diffusion data at time t are modeled by the conditional distribution of x given t under (9), which is an EBM with energy function $U_\theta(x,t)$ in x only:

$$p_\theta(x|t) = \frac{\exp(-U_\theta(x,t))}{Z_t(\theta)}, \tag{10}$$

where $Z_t(\theta) = \int \exp(-U_\theta(x,t))\mathrm{d}x$. In particular, $p_\theta(x|0)$ at time-0 represents an EBM for the original data, whereas $p_\theta(x|T)$ at time-T represents an EBM for a data distribution close to a Gaussian noise. Hence, model (9) combines individual EBMs for original and diffusion data at different time steps into a compact, joint model.

We pursue persistent training for the joint EBM (9). Training Algorithm 1 essentially follows ML-SA in Section 2. Lines 3–6 are the sampling operation, and line 7 is the parameter updating operation. However, we incorporate an enhanced sampling algorithm, called MALA within Gibbs mixture sampling (MGMS), to achieve joint sampling from the model distribution $p_\theta(x,t)$ in an efficient manner (Algorithm 2). In Algorithm 2, lines 2–6 perform sampling from the conditional distribution $p_\theta(x|\tilde{t}_0)$ given the current time label $\tilde{t}_0$ using MALA, starting from the current configuration $\tilde{x}_0$. Line 7 performs exact sampling from the conditional distribution $p_\theta(t|\tilde{x}_1)$ given the new configuration $\tilde{x}_1$. The same MGMS can also be used to implement post-training sampling, as shown in Algorithm 3.

The MGMS algorithm can be derived as an instance of labeled mixture sampling in Tan (2017). The two operations are called, respectively, Markov move and global jump, which draws $\tilde{t}_1$ given $\tilde{x}_1$ independently of $\tilde{t}_0$.

These two operations illustrate how MGMS may achieve proper sampling from a multimodal distribution: moving through auxiliary distributions and jumping to different modes in the original distribution. Moreover, our learning Algorithm 1 can be seen to extend self-adjusted mixture sampling in Tan (2017) for handling nonintercept parameters in learning EBMs. See Appendix III for further discussion.

We comment on some additional features of Algorithm 2. The Langevin step size σ_t is allowed to depend on the time label t to accommodate different variations in the diffusion data distributions. We employ MALA instead of Langevin dynamics without rejection, to ensure sampling convergence for relatively large σ_t. Moreover, the acceptance rates from the Metropolis–Hastings step can be used to adjust step sizes dynamically to automate tuning. See Appendix V.3.

Algorithm 1 ML learning for diffusion-assisted EBMs

1: **Input:** Real data $\mathcal{D}$, replay-buffer $\mathcal{B}$, initial value θ, learning rate γ, Langevin steps L, step sizes $\{\sigma_t\}_{t=0}^{T}$.
2: **repeat**
3: **Draw** x_0 from $\mathcal{D}$ and $t_1 \sim \mathrm{Unif}(0, 1, \ldots, T)$;
4: **Generate** $x_1 = x_0^{(t_1)}$ by (8) with $z = x_0$ and $t = t_1$;
5: **Draw** $(\tilde{x}_0, \tilde{t}_0)$ from $\mathcal{B}$;
6: **Sample** $(\tilde{x}_1, \tilde{t}_1)$ given $(\tilde{x}_0, \tilde{t}_0)$ by Algorithm 2 and **reset** $(\tilde{x}_0, \tilde{t}_0)$ to $(\tilde{x}_1, \tilde{t}_1)$ in $\mathcal{B}$;
7: **Update** using gradient ascent $\theta \leftarrow \theta + \gamma\{-\frac{\partial}{\partial\theta} U_\theta(x_1, t_1) + \frac{\partial}{\partial\theta} U_\theta(\tilde{x}_1, \tilde{t}_1)\}$.
8: **until** Converged
9: **Return:** ML estimator θ.

Algorithm 2 MALA within Gibbs mixture sampling (MGMS)

1: **Input:** Current observation $(\tilde{x}_0, \tilde{t}_0)$, Langevin steps L, step sizes $\{\sigma_t\}_{t=0}^{T}$.
2: **for** $l = 1$ **to** L **do**
3: **Draw** $\varepsilon \sim \mathrm{N}(0, I)$ and propose $\tilde{x}_{1/2} = \tilde{x}_0 - \frac{\sigma_{\tilde{t}_0}^2}{2}\nabla_x U_\theta(\tilde{x}_0, \tilde{t}_0) + \sigma_{\tilde{t}_0}\varepsilon$;
4: **Accept** $\tilde{x}_1 = \tilde{x}_{1/2}$ with probability $\min(1, r)$ or **reject** $\tilde{x}_{1/2}$ and set $\tilde{x}_1 = \tilde{x}_0$, where $r = \exp(-H_1 + H_0)$, where

$$H_1 = U_\theta\left(\tilde{x}_{1/2}, \tilde{t}_0\right) + \frac{1}{2\sigma_{\tilde{t}_0}^2}\left\|\tilde{x}_0 - \tilde{x}_{1/2} + \frac{\sigma_{\tilde{t}_0}^2}{2}\nabla_x U_\theta\left(\tilde{x}_{1/2}, \tilde{t}_0\right)\right\|_2^2,$$

$$H_0 = U_\theta\left(\tilde{x}_0, \tilde{t}_0\right) + \frac{1}{2\sigma_{\tilde{t}_0}^2}\left\|\tilde{x}_{1/2} - \tilde{x}_0 + \frac{\sigma_{\tilde{t}_0}^2}{2}\nabla_x U_\theta\left(\tilde{x}_0, \tilde{t}_0\right)\right\|_2^2.$$

5: **Reset** $\tilde{x}_0 = \tilde{x}_1$.
6: **end for**
7: **Draw** $\tilde{t}_1$ given $\tilde{x}_1$ from Multinomial with class probabilities $p_\theta(t|\tilde{x}_1) = \frac{\exp(-U_\theta(\tilde{x}_1, t))}{\sum_{k=0}^{T}\exp(-U_\theta(\tilde{x}_1, k))}$.
8: **Return:** New observation $(\tilde{x}_1, \tilde{t}_1)$.

The joint EBM (9) has the same form as in Grathwohl et al. (2020), although a time step t is involved instead of a class label y. The two models are formulated for apparently different purposes. The training algorithms are also different in the initialisation scheme, persistent initialisation here versus hybrid initialisation in Grathwohl et al. (2020) and in how MCMC sampling is handled. Grathwohl et al. (2020) factorised the joint EBM density as $p_\theta(x, y) = p_\theta(y|x)p_\theta(x)$ and apply Langevin sampling (with a constant step size) to the marginal density $p_\theta(x)$. This method is unsuitable in our setting, because the diffusion data exhibit different variations at different time steps. See a related discussion of mixture importance sampling in Appendix III.

Algorithm 3 Posttraining sampling with MGMS

1: **Input:** Random noise and time label, $(\tilde{x}_0, \tilde{t}_0) = (\varepsilon, T)$, stopping criterion M.
2: **repeat**
3: **Sample** $(\tilde{x}_1, \tilde{t}_1)$ given $(\tilde{x}_0, \tilde{t}_0)$ by Algorithm 2 and **reset** $(\tilde{x}_0, \tilde{t}_0)$ to $(\tilde{x}_1, \tilde{t}_1)$;
4: **until** $\tilde{t}_0$ reaches time 0 for M times
5: **Return:** Synthetic observation $\tilde{x}_0$ at time 0.

4 | RELATED WORK

There is a vast and growing literature on EBMs and related topics. In addition to earlier discussions in Sections 1–3, we discuss here how our approach is compared with score-based diffusion modeling, mainly the representative works (Ho et al., 2020; Sohl-Dickstein et al., 2015; Song & Ermon, 2019; Song et al., 2021), and with the DRL for EBM learning (Gao et al., 2021).

Our approach and the score-based approach differ in how to use these diffusion data for learning from the original data. Denote as p_t^* the population density of the diffusion data distribution at time t. The score-based approach postulates a score-based model $s_\theta(x, t)$ for the score $\nabla_x \log p_t^*(x)$ and then employs denoising score matching and extensions to implement training (Hyvärinen, 2005; Vincent, 2011). The log-density (or log-likelihood) $\log p_0^*(x)$ for the original data is not directly parameterised but can be approximated by solving the probability flow ODE based on the learned score (Song et al., 2021). From our experience, this method for likelihood calculation is computationally costly (see Appendix III). More importantly, the calculated likelihoods from the score-based approach are observed to suffer the OOD reversal in our experiments on MNIST-type images.

By comparison, our approach involves EBM modeling and ML learning. Hence, the energy function (or the negative log-likelihood up to a constant) is analytically available for the learned EBMs. The learned energy functions are globally more meaningful than those obtained by previous methods, as shown by the superior performance for OOD detection in our experiments. The MGMS algorithm from our training algorithm can also be used to generate new images from noises based on the learned EBMs (Algorithm 3).

For diffusion data as in Ho et al. (2020), the approach of Gao et al. (2021) postulates marginal EBMs in the form (10), rewritten as $p_\theta(x^{(t)}) = \exp(-U_\theta(x^{(t)}, t))/Z_t(\theta)$, and then derives conditional EBMs $p_\theta(x^{(t-1)}|x^{(t)})$, where $x^{(t)}$ denotes a diffusion observation at time t. The training algorithm of Gao et al. (2021) proceeds similarly as training marginal EBMs $p_\theta(x^{(t-1)})$ but in each iteration applies a fixed number of Langevin sampling steps for $p_\theta(x^{(t-1)}|x^{(t)})$, initialised by the diffusion observation $x^{(t)}$ (being conditioned on). This is an instance of data initialisation (or a

variation of CD) as discussed in Section 3.1. For post-training image generation, the same number of Langevin sampling steps are applied to sample from $p_\theta(x^{(t-1)}|x^{(t)})$ with the initial value set to the final observation drawn at time t, iteratively from $t = T$ to 1. At time T, the initial value is a random noise. This is reminiscent of image generation using learned short-run MCMC (Nijkamp et al., 2019), although in a sequential manner as in backward diffusion.

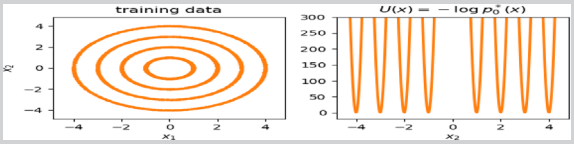
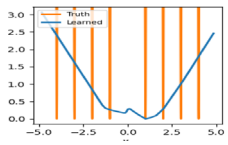
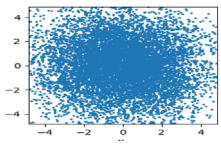
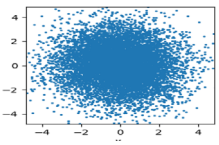
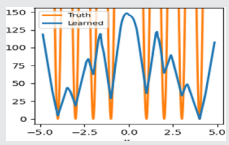
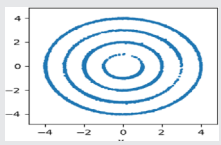
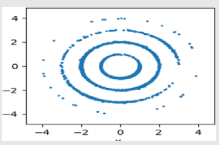
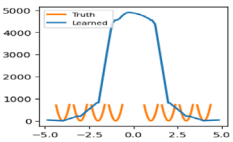
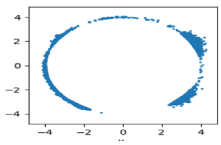
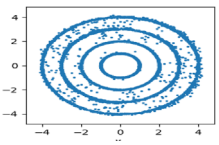
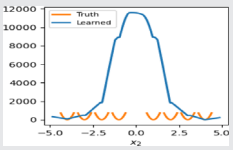
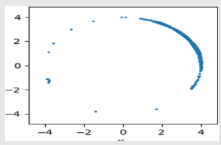
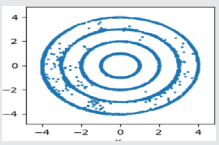
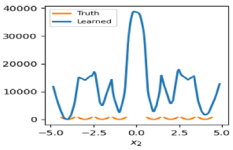
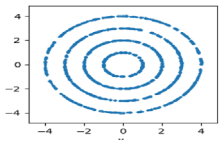
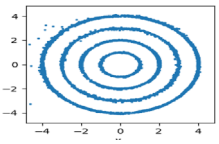
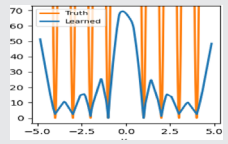
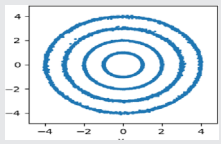
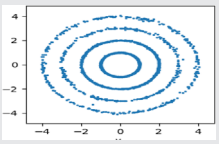
The learned energy functions in the recovery likelihood approach appear to be globally misaligned and suffer the OOD reversal in our experiments.

In the Appendix IV, we show that the recovery likelihood approach can be equivalently formulated as training via CD 'bivariate' EBMs for the pairs of observations $(x^{(t-1)}, x^{(t)})$, although it is originally presented in terms of training conditional EBMs $p_\theta(x^{(t-1)}|x^{(t)})$. Hence, our approach differs from Gao et al. (2021) in maintaining persistent training while combining marginal (univariate) EBMs into a joint EBM and incorporating enhanced MCMC sampling. These choices help to improve the global alignment of the learned energy functions as seen in our experiments.

5 | ILLUSTRATIVE EXPERIMENT

We provide a 2D example of four rings, where the data distribution exhibits multimodality and singularity. Similar examples are reported in Nijkamp et al. (2019) and Gao et al. (2021). We compare EBMs trained with different initialisation schemes (Section 2) and DRL (Gao et al., 2021) and our method (DA-EBM). See Appendix V.2 for further details of the experiment.

TABLE 2 Results from four-ring example.

Train. data info.			
method	**Energy**	**Long-run**	**Post**
EBM CD			
EBM Pers.			
EBM Noise			
EBM Hybrid (5% noise)			
DRL Data Init.			
DA-EBM Pers. Init.			

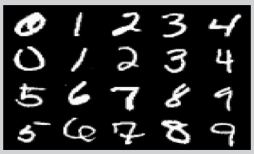
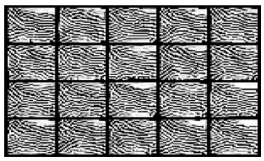
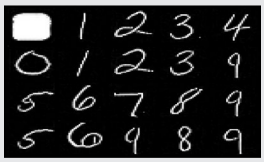
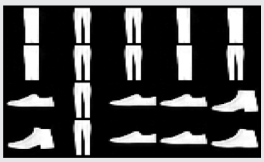
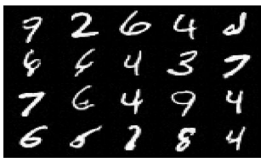
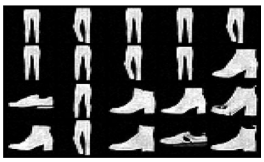
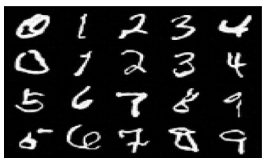
Abbreviation: EBM, energy-based model.

In Table 2, we plot learned energy functions (along the line $x_1 = 0$ and, for comparison, anchored to have a minimum 0), long-run samples and post-training samples. An energy function is negative log-density up to a constant.

For DRL and DA-EBM, we plot learned energy functions at time 0 in Table 2 and at other times in Appendix Table S1.

Table 2 confirms the properties of EBM methods in Table 1. (i) Among the first four rows for usual EBMs, only persistent training gives long-run stable samples. For persistent training, the learned energy has local modes at the four rings but is globally misaligned. For example, the learned energy at $x_2 = -2$ (where data are present) is incorrectly higher than that at $x_2 = -3.5$ (where data are absent). Post-training sampling (from standard Gaussian noises) can barely reach the outmost ring. (ii) The noise initialisation (short-run MCMC) and hybrid initialisation give similar results. The learned energy functions have local modes at the rings, but the modes are almost invisible due to the large ranges of energy values, which makes the gradient $\nabla_x U_\theta(x)$ strong to drive Langevin dynamics for data generation. Post-training sampling can generate realistic samples, but long-run sampling is unstable. (iii) Compared with noise and hybrid initialisations, DRL also learns an energy function which has a large range and is globally misaligned. The energy landscape has deeper local modes at the rings, and hence, long-run sampling can be stable. (iv) Among all methods, our method gives an energy function which best approximates the truth. The energy values are properly aligned across the local modes at the four rings. Long-run sampling is stable. The post-training sampling result shows that our enhanced MCMC sampling can move across modes, with help from exploring all diffusion data distributions.

TABLE 3 MNIST and FashionMNIST results.

	MNIST		FashionMNIST	
Train. data (long-run starting points)				
method	Long-run	Post	Long-run	Post
EBM Pers.				
EBM Noise				
EBM Hybrid (5% noise)				
DRL Data Init.				
DA-EBM Pers. Init.				

Abbreviations: DRL, diffusion recovery likelihood; EBM, energy-based model.

6 | IMAGE EXPERIMENTS

We study the performances of several existing methods and ours on the two grayscale image datasets, MNIST and FashionMNIST. In addition to EBM training methods as in Section 5, we also include Glow (Kingma & Dhariwal, 2018) and continuous-time DDPM (cDDPM) (Song et al., 2021) when investigating OOD detection. We do not report inception scores, because differences in the sampling results are visually clear between different methods. See Appendix V.3 for further details of the experiment.

6.1 | Long-run stability and image generation

Table 3 presents the results of long-run sampling and image generation on MNIST and FashionMNIST, with supplemental plots provided in Appendix V.3. The *exact* starting points of long-run sampling are shown in the first row.

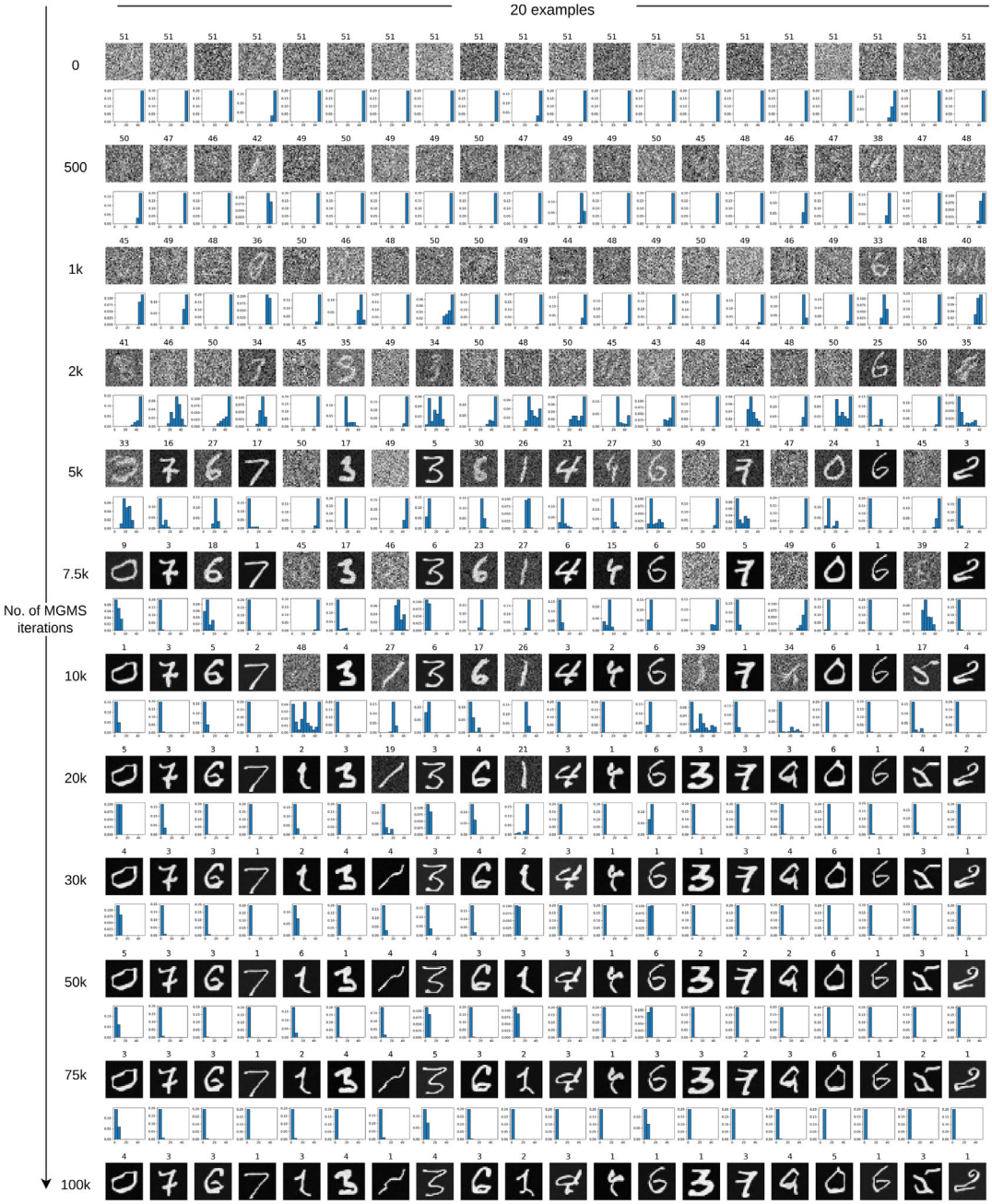


FIGURE 2 Examples of post-training image generation through MALA within Gibbs mixture sampling (MGMS) (Algorithm 2).

As explained below, the qualitative results in Table 3 substantiate the effectiveness of our proposed method. Popular metrics such as the inception score or Frechet inception distance (FID) may not be entirely suitable to MNIST-type grayscale data. To compensate, we provide quantitative results about OOD detection in Table 5.

From Table 3, the long-run samples from hybrid-initialised EBM collapse into mostly digit 1 when starting with images other than 1. This is similar to the collapse of long-run samples to the outmost ring in Table 2. The long-run samples from persistent-initialised EBM and DRL are stable, but the former exhibit significant distortions whereas those from DRL have some arbitrary sprinkles added. In contrast, the long-run samples from DA-EBM remain close to the starting images.

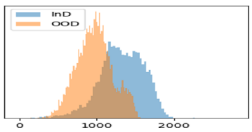
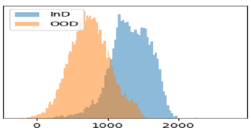
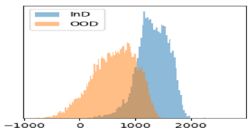
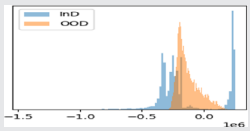
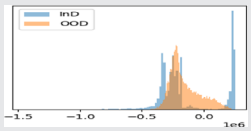
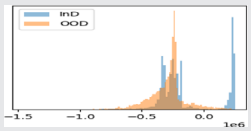
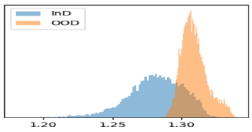
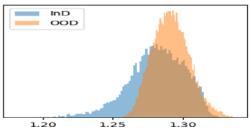
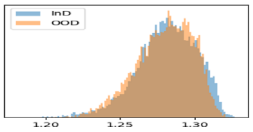
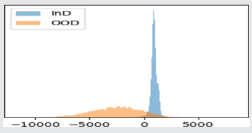
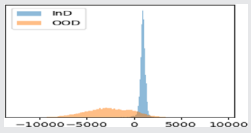
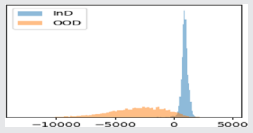
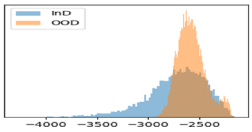
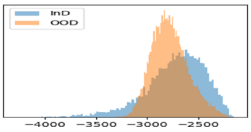
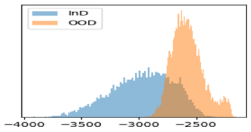
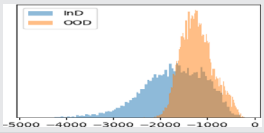
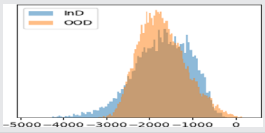
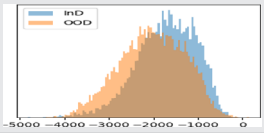
For image generation, EBM with hybrid initialisation seems to generate samples of good quality. Persistently trained EBM fail to yield satisfactory images. The samples from DRL are distorted and deviate from realistic digits. The samples from DA-EBM, despite being persistently trained, are close to realistic digits. Hence, DA-EBM is the only method which performs satisfactorily in both long-run stability (from real images) and image generation (from random noises).

In Figure 2, we illustrate the image generation process using a collection of 20 images. Each column shows an individual sampling process where MGMS (Algorithm 2) starts from Gaussian noise with the time label (or noise level) $t = 51$. For visualisation, the sampling process is captured at 12 predetermined sampling iterations, with images and their associated time labels depicted in rows. Moreover, we include histograms of time labels between two consecutive sampling rows, indicating the distribution of time labels obtained between the two sampling iterations.

Interestingly, there is a noticeable trend in the image label distributions, from approximately $t = 50$ decreasing towards $t = 0$, as the number of MGMS sampling iterations increases. Meanwhile, each image gradually transmutes from noise to recognizable digits. Once an image crystallises into a digit-like image, the shape appears resistent to significant changes.

In theory, the design of DA-EBM would yield a nearly uniform time label distribution spanning from 0 to 51 during the post-training sampling. However, the results in Figure 2 appear to deviate from this theoretical expectation. This discrepancy may be mainly due to the 'cold' nature of the real image distribution: We observe that a substantial gap exists in energy functions between the distributions of clear images and diffused images. This gap causes the time label to predominantly stay near 0 in the global jump phase of the MGMS algorithm, because a large energy

TABLE 4 OOD results.

Method	MNIST	EMNIST	KMNIST
EBM Pers.			
EBM Hybrid (5% noise)			
DRL			
DA-EBM			
GLOW			
cDDPM			

Abbreviations: DA-EBM, diffusion-assisted EBM; DRL, diffusion recovery likelihood; EBM, energy-based model; OOD, out-of-distribution.

TABLE 5 OOD AUROC results.

Method	Mnist	EMnist	KMnist
EBM PERS.	0.83	0.92	0.93
EBM HYBRID	0.37	0.45	0.68
DRL	0.09	0.35	0.52
DA-EBM	**0.93**	**0.93**	**0.98**
GLOW	0.35	0.60	0.09
cDDPM	0.27	0.52	0.63

Abbreviations: AUROC, area under ROC; DA-EBM, diffusion-assisted EBM; EBM, energy-based model; OOD, out-of-distribution.

difference (in hundreds or more) invariably leads to classifying an image to a time label close to 0 once it becomes a realistic digit. Implications of this phenomenon and potential improvement can be investigated in future work.

6.2 | OOD detection

For OOD detection, we use FashionMNIST as the in-distribution (InD) dataset and consider MNIST, EMNIST (Letters) and KMNIST (Japanese characters) as OOD datasets. All methods are trained on the training split of FashionMNIST, and OOD metrics are evaluated on the test split of each dataset. All methods use estimated log-densities (i.e. log-likelihoods) as the scores to predict InD and OOD data. Table 4 presents the histograms of log-likelihoods, and Table 5 gives the AUROC (area under ROC) results. Additional results can be found in Appendix V.3.

Glow, cDDPM and DRL all exhibit the OOD reversal. The learned models assign higher (or similar) likelihoods to the OOD data than to InD data. For the hybrid-initialised EBM, the log-likelihoods of OOD data cover several modes of those of InD data. This pattern was also seen in Table 2 of Grathwohl et al. (2020). In addition, the ranges of these log-likelihoods are of 10^6 magnitude, reminiscent of the unreasonable ranges of energy values in Table 2.

Persistently trained EBM and DA-EBM are the only two methods which assign likelihoods in a proper direction (InD vs. OOD). But our learned likelihoods achieve better separation between InD and OOD data. The AUROC values from DA-EBM not only improve upon the usual EBMs by margins at least 10%, 1% and 5% for the three OOD datasets but also are much higher than those from Glow and cDDPM. From the AUC-PR Table S7 in Appendix, DA-EBM also outperforms (with over 10% margins) the best in Table 1 of Elflein et al. (2021) among EBM and other methods.

7 | CONCLUSION

We propose diffusion-assisted EBMs and develop a persistent training algorithm. The diffusion data are exploited for two benefits. First, the diffusion data help to bridge different modes in the original data such that the density estimates between different modes can be appropriately aligned. Second, the diffusion data also help to connect images and random noises to make post-training sampling through MCMC possible. Our work opens the possibility of simultaneously achieving proper density estimation and post-training sampling. See Appendix VI for a discussion about the current limitations of our method. It is desirable to further develop our method and conduct experiments on more complex image datasets and diverse tasks.

AUTHOR CONTRIBUTIONS

Xinwei Zhang designed the study, wrote the original draft, developed the code, and performed experiments. Zhiqiang Tan designed and supervised the study, edited the paper, and provided computing resources. Zhijian Ou designed the study, edited the paper, and provided computing resources.

DATA AVAILABILITY STATEMENT

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

ORCID

Xinwei Zhang https://orcid.org/0000-0003-1176-3815
Zhiqiang Tan https://orcid.org/0000-0003-1780-6839

REFERENCES

Benveniste, A., Priouret, P., & Métivier, M. (1990). *Adaptive algorithms and stochastic approximations*: Springer.

Besag, J. E. (1994). Comments on "representations of knowledge in complex systems" by U. Grenander and M.I. Miller. *Journal of the Royal Statistical Society: Series B*, *56*, 549–581.

Choi, H., Jang, E., & Alemi, A. A. (2019). WAIC, but why? Generative ensembles for robust anomaly detection. arXiv preprint arXiv:1810.01392v4.

Cover, T. M., & Thomas, J. A. (2006). *Elements of information theory*: Wiley.

Dai, B., Liu, Z., Dai, H., He, N., Gretton, A., Song, L., & Schuurmans, D. (2019). Exponential family estimation via adversarial dynamics embedding. In *Advances in Neural Information Processing Systems*, pp. 32.

Du, Y., & Mordatch, I. (2019). Implicit generation and modeling with energy based models. In *Advances in Neural Information Processing Systems*, pp. 32.

Elflein, S., Charpentier, B., Zügner, D., & Günnemann, S. (2021). On out-of-distribution detection with energy-based models. In *ICML 2021 Workshop on Uncertainty and Robustness in Deep Learning*.

Gao, R., Song, Y., Poole, B., Wu, Y. N., & Kingma, D. P. (2021). Learning energy-based models by diffusion recovery likelihood. In *International Conference on Learning Representations*.

Geyer, C. J. (1991). Markov chain Monte Carlo maximum likelihood. In *Computing Science and Statistics: Proceedings of 23rd Symposium on the Interface Interface Foundation, Fairfax Station*, pp. 156–163.

Geyer, C. J., & Thompson, E. A. (1995). Annealing Markov chain Monte Carlo with applications to ancestral inference. *Journal of the American Statistical Association*, *90*, 909–920.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pp. 27.

Grathwohl, W., Wang, K.-C., Jacobsen, J.-H., Duvenaud, D., Norouzi, M., & Swersky, K. (2020). Your classifier is secretly an energy based model and you should treat it like one. In *International Conference on Learning Representations*.

Grathwohl, W. S., Kelly, J. J., Hashemi, M., Norouzi, M., Swersky, K., & Duvenaud, D. (2021). No MCMC for me: Amortized sampling for fast and stable training of energy-based models. In *International Conference on Learning Representations*.

Gutmann, M. U., & Hyvärinen, A. (2012). Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of Machine Learning Research*, *13*, 307–361.

Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural Computation*, *14*, 1771–1800.

Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, pp. 6840–6851.

Hyvärinen, A. (2005). Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, *6*, 695–709.

Kim, T., & Bengio, Y. (2016). Deep directed generative models with energy-based probability estimation. arXiv preprint arXiv:1606.03439.

Kingma, D. P., & Dhariwal, P. (2018). Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, pp. 31.

LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., & Huang, F. (2006). A tutorial on energy-based learning. *Predicting Structured Data*, *1*, 191–246.

Marinari, E., & Parisi, G. (1992). Simulated tempering: A new Monte Carlo scheme. *Europhysics Letters*, *19*, 451.

Nalisnick, E., Matsukawa, A., Teh, Y. W., Gorur, D., & Lakshminarayanan, B. (2019). Do deep generative models know what they don't know? In *International Conference on Learning Representations*.

Nalisnick, E., Matsukawa, A., Teh, Y. W., & Lakshminarayanan, B. (2019). Detecting out-of-distribution inputs to deep generative models using typicality. arXiv preprint arXiv:1906.02994.

Nijkamp, E., Hill, M., Han, T., Zhu, S.-C., & Wu, Y. N. (2020). On the anatomy of MCMC-based maximum likelihood learning of energy-based models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 5272–5280.

Nijkamp, E., Hill, M., Zhu, S.-C., & Wu, Y. N. (2019). Learning non-convergent non-persistent short-run MCMC toward energy-based model. In *Advances in Neural Information Processing Systems*, pp. 32.

Robbins, H., & Monro, S. (1951). A stochastic approximation method. *Annals of Mathematical Statistics*, *22*, 400–407.

Roberts, G. O., & Tweedie, R. L. (1996). Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli*, *2*, 341–363.

Saremi, S., Mehrjou, A., Schölkopf, B., & Hyvärinen, A. (2018). Deep energy estimator networks. arXiv preprint arXiv:1805.08306.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., & Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pp. 2256–2265.

Song, Y., & Ermon, S. (2019). Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, pp. 32.

Song, Y., & Ou, Z. (2018). Learning neural random fields with inclusive auxiliary generators. arXiv preprint arXiv:1806.00271v4.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., & Poole, B. (2021). Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*.

Swendsen, R. H., & Wang, J.-S. (1986). Replica Monte Carlo simulation of spin-glasses. *Physical Review Letters*, *57*, 2607–2609.

Tan, Z. (2017). Optimally adjusted mixture sampling and locally weighted histogram analysis. *Journal of Computational and Graphical Statistics*, *26*, 54–65.

Tieleman, T. (2008). Training restricted Boltzmann machines using approximations to the likelihood gradient. In *International Conference on Machine Learning*, pp. 1064–1071.

Vincent, P. (2011). A connection between score matching and denoising autoencoders. *Neural Computation*, *23*, 1661–1674.

Xie, J., Lu, Y., Gao, R., Zhu, S.-C., & Wu, Y. N. (2018). Cooperative training of descriptor and generator networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *42*, 27–45.

Xie, J., Lu, Y., Zhu, S.-C., & Wu, Y. (2016). A theory of generative ConvNet. In *International Conference on Machine Learning*, pp. 2635–2644.

Yu, L., Song, Y., Song, J., & Ermon, S. (2020). Training deep energy-based models with f-divergence minimization. In *International Conference on Machine Learning*, pp. 10957–10967.

AUTHOR BIOGRAPHIES

Xinwei Zhang is a Postdoctoral Research Fellow in the Department of Biostatistics at New York University. He received his PhD in statistics from Rutgers University in 2023 and obtained an AM in statistics from Washington University in St. Louis in 2017. Before pursuing graduate studies, he received his BS degree in Mathematics and Applied Mathematics and BE degree in Computer Science from Renmin University of China in 2015. His research interests include statistical learning, multiclass classification, and energy-based models.

Zhiqiang Tan received a BS degree in Applied Mathematics from Tsinghua University and a PhD degree in Statistics from the University of Chicago. He is a Distinguished Professor in the Department of Statistics at Rutgers University. He received an NSF CAREER award and is a Fellow of the American Statistical Association, a Fellow of the Institute of Mathematical Statistics, and an Elected Member of the International Statistical Institute. He has published extensively on statistical theory, methods, and applications in leading statistical and interdisciplinary journals and conferences. His research interests include Monte Carlo methods, causal inference, statistical learning, and related areas.

Zhijian Ou is an associate professor with the Department of Electronic Engineering at Tsinghua University. He received his PhD degree from Tsinghua University in 2003. He currently serves as Associate Editor of "IEEE/ACM Transactions on Audio, Speech and Language Processing", Editorial Board Member of "Computer Speech and Language", member of the IEEE Speech and Language Processing Technical Committee, and he was General Chair of SLT 2021, EMNLP 2022 SereTOD workshop, and Tutorial Chair of INTERSPEECH 2020. He conducts research in the general area of speech and language processing (particularly dialogue systems, speech recognition) and machine intelligence (particularly with graphical models and deep learning).

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

How to cite this article: Zhang, X., Tan, Z., & Ou, Z. (2023). Persistently trained, diffusion-assisted energy-based models. *Stat*, *12*(1), e625. https://doi.org/10.1002/sta4.625

Supplementary Material for "Persistently Trained, Diffusion-assisted Energy-based Models"

Xinwei Zhang[1], Zhiqiang Tan[1], and Zhijian Ou[2]
[1]Department of Statistics, Rutgers University, Piscataway, New Jersey 08854, U.S.A
[2]Department of Electronic Engineering, Tsinghua University, Beijing, 100084, China.
xinwei.zhang@rutgers.edu, ztan@stat.rutgers.edu, ozj@tsinghua.edu.cn

We provide additional material to support the content of the paper.

I | LOCAL ENERGIES

To formalize the discussion in Section 3.1, we introduce the following concepts. For a (normalized) probability density $p(\cdot)$, say that $U(\cdot)$ is a *(global) energy function* for $p(\cdot)$ if

$$p(x) = \frac{\exp(-U(x))}{\int \exp(-U(x'))\,\mathrm{d}x'}.$$

Hence an energy function usually defined is a global energy function. Suppose that the support of $p(\cdot)$, $\{x : p(x) > 0\}$, is arbitrarily partitioned into 2 regions B_1 and B_2. Say that $\tilde{U}(\cdot)$ is a *local energy function* for $p(\cdot)$ with respect to (B_1, B_2) if

$$p(x) = \sum_{j=1}^{2} p(B_j) \frac{\exp(-\tilde{U}(x))}{\int_{B_j} \exp(-\tilde{U}(x'))\,\mathrm{d}x'} 1_{B_j}(x), \tag{S1}$$

where $p(B_j) = \int_{B_j} p(x)\,\mathrm{d}x$ and 1_{B_j} is the indicator function for B_j. Informally, a local energy function can be locally normalized and then properly combined into a given probability distribution. This definition suffices in the following discussion of multimodal distributions with modes separated by zero-density barriers. For nonzero- but low-density barriers, our discussion can be extended with more technical complexity, for example, allowing equality (S1) to hold approximately, as measured by some distance between the two sides of (S1).

Let p_0 be a mixture of two disjoint densities,

$$p_0(x) = \pi p_{01}(x) + (1 - \pi) p_{02}(x), \tag{S2}$$

where p_{01} and p_{02} are two disjoint (normalized) densities with supports B_1 and B_2 respectively ($B_1 \cap B_2 = \emptyset$) and $\pi \in (0, 1)$ is the relative weight for p_{01}. Then there exist an infinite collection of local energy functions with respect to (B_1, B_2), such that none of them is a global energy function for $p_0(\cdot)$. These local energy functions can be defined as $(-\log p_{01} + c_{01})1_{B_1} + (-\log p_{02} + c_{02})1_{B_2} + \infty 1_{(B_1 \cup B_2)^c}$ for arbitrary constants (c_{01}, c_{02}).

For a multimodal distribution with strictly separated modes, a local energy function (not necessarily a global energy function) can be used to achieve global invariance (S5) below, with an MCMC sampler such as random walk Metropolis sampling or Langevin sampling, which only enables local moves near individual modes. Let $\tilde{U}(\cdot)$ be a local energy function for $p_0(\cdot)$ as above, and let $K_{\tilde{U}}(\cdot, \cdot)$ be the transition kernel of an MCMC sampler, depending on $\tilde{U}(\cdot)$, such that (see Appendix II)

$$K_{\tilde{U}}(x, x') = 0 \text{ for } x \in B_1, x' \in B_2 \text{ or } x \in B_2, x' \in B_1, \tag{S3}$$

$$\textstyle\int_{B_j} K_{\tilde{U}}(x, x') p_{0j}(x)\,\mathrm{d}x = p_{0j}(x') \text{ for } x' \in B_j, j = 1, 2. \tag{S4}$$

Informally, the sampler only moves points within B_1 or B_2, and leaves p_{01} or p_{02} (locally) invariant on B_1 or B_2. Then the global invariance holds (see the proof later):

$$\textstyle\int K_{\tilde{U}}(x, x') p_0(x)\,\mathrm{d}x = p_0(x'),\ x' \in B_1 \cup B_2, \tag{S5}$$

that is, the sampler leaves the mixture distribution p_0 invariant on $B_1 \cup B_2$. Note that property (S5) does not imply that the Markov chain with kernel $K_{\tilde{U}}$ and any initial point converges to the target distribution p_0; the chain is restricted to B_j if the initial point is from B_j. But a key message here is that in the presence of separated modes, global invariance (S5) can be *non-uniquely* achieved by using any local energy function U with an MCMC sampler which enables only local mixing.

We give a direct proof of Eq. (S5). For $x' \in B_1$, the left-hand side of (S5) can be calculated as

$$\int K_{\tilde{U}}(x, x') p_0(x)\,\mathrm{d}x = \int_{B_1} K_{\tilde{U}}(x, x') p_0(x)\,\mathrm{d}x = \pi \int_{B_1} K_{\tilde{U}}(x, x') p_{01}(x)\,\mathrm{d}x = \pi p_{01}(x'),$$

where the first equality is from (S3), the second from (S2) and $B_1 \cap B_2 = \emptyset$, and the third from (S4). Similarly, for $x' \in B_2$, the left-hand side of (S5) can be shown to be $(1-\pi)p_{02}(x')$. Combining the two cases gives the desired equality (S5), due to (S2) and $B_1 \cap B_2 = \emptyset$.

II | LOCAL MIXING

We discuss the local mixing behavior of Langevin sampling in the presence of separated modes. For simplicity, we focus on the situation where two modes are strictly separated as two disjoint components of a mixture as in Appendix I. The main point can also be extended to separated modes by nonzero- but low-density barriers.

Consider a mixture of two disjoint densities as defined in (S2). Let $U_0(x) = -\log p_0(x) + c_0$ and $U_{0j}(x) = -\log p_{0j}(x) + c_{0j}$ for $j = 1, 2$, where c_0, c_{01}, c_{02} are three arbitrary (unknown) constants. Then the gradient of the energy function U_0 is

$$\nabla_x U_0(x) = \begin{cases} \nabla_x U_{01}(x), & x \in B_1, \\ \nabla_x U_{02}(x), & x \in B_2. \end{cases} \tag{S6}$$

From this relationship, if an initial point is from the support of p_{01}, then Langevin sampling treats p_{01} as the target (invariant) distribution. In this case, the Langevin chain only travels within the support of p_{01}. Similarly, if an initial point is from the support of p_{02}, then the Langevin chain only travels within the support of p_{02}. Therefore, cross-mode traveling is impossible (or almost impossible with low-density barriers), and the Langevin chain is always trapped in the region where the initial point is located. From another perspective, the Langevin chain using the energy function U_0 may admit a mixture of p_{01} and p_{02} with *any* relative weight, not just p_0, as an invariant distribution. The relative weight information is lost for Langevin sampling. See Song and Ermon (2019), Section 3.2.2., for a related discussion.

As discussed in Appendix I, a local energy function for p_0 can be defined as $\tilde{U}(x) = U_{01}(x)$ if $x \in B_1$ or $U_{02}(x)$ if $x \in B_2$ or ∞ otherwise. From the gradient expression (S6), Langevin sampling using $\tilde{U}$ as the energy function is indistinguishable from Langevin sampling using the (global) energy function U_0. This also explains why the global invariance (S5) is satisfied for Langevin sampling using any local energy function in the presence of separated modes.

III | ENHANCED SAMPLING

To overcome local mixing for multimodal distributions, various enhanced sampling algorithms can be used while introducing auxiliary distributions as mentioned in Section 3.2. For a distribution with energy function U, a popular scheme in practice as well as theoretical studies (Chen, Chen, Dong, Peng, & Wang 2019; Ge, Lee, & Risteski 2018) is to introduce tempered distributions with energies $\{\beta_j U : j = 1, \ldots, J\}$ for a decreasing sequence of inverse temperatures $1 = \beta_1 > \cdots > \beta_J \approx 0$. The Markov chains at higher temperatures (smaller β_j) may travel more easily between modes, which then provide "bridges" for the Markov chains at lower temperatures to explore those modes (even separated by low-density barriers). More generally, a heuristic guideline is that those auxiliary distributions should be easy to sample from and connected (or overlapped) with the original distribution, so as to help the sampling algorithm to explore the original distribution.

Return to the mixture distribution of two disjoint densities, as defined in (S2). In this case, introducing tempered distributions does not work because each tempered distribution remains a mixture of two disjoint densities. Alternatively, we may introduce another auxiliary distribution to fill the low-density regions in p_0. Let $p_1(x)$ be an "umbrella" unimodal density such that its support $\{x : p_1(x) > 0\}$ contains $B_1 \cup B_2$, and let $U_1(x) = -\log p_1(x) + c_1$, where c_1 is an arbitrary (unknown) constant. The difference of log-normalizing constants (or free energy difference) between U_0 and U_1 is $\delta = c_1 - c_0$. To see advantages of our sampling Algorithm 2 and a connection to our learning Algorithm 1, we compare several sampling algorithms in the special case of only one auxiliary distribution.

Simple importance sampling (SIS). The first is SIS with the design density taken to be p_1. By unimodality of p_1, we assume that Langevin sampling from p_1 achieves global (fast) mixing.

(i) Draw a sample $\{\tilde{x}_i\}_{i=1}^n$ from p_1 using MALA from an initial value $\tilde{x}_0$.

(ii) Draw a sample of some size k, $\{i_1, \ldots, i_k\}$, with replacement from $\{1, 2, \ldots, n\}$ with the probability of selecting i equal to

$$w_i = \frac{(p_0/p_1)(\tilde{x}_i)}{\sum_{l=1}^n (p_0/p_1)(\tilde{x}_l)} = \frac{\mathrm{e}^{-U_0(\tilde{x}_i)+U_1(\tilde{x}_i)}}{\sum_{l=1}^n \mathrm{e}^{-U_0(\tilde{x}_l)+U_1(\tilde{x}_l)}}.$$

Then $\{\tilde{x}_{i_1}, \ldots, \tilde{x}_{i_k}\}$ is a valid sample from p_0, in the sense that sample averages are consistent for the corresponding expectations under p_0 as $n \to \infty$. The weights w_i can be calculated without knowing δ, but the sample is only approximately unbiased for large n. The algorithm is non-iterative because increasing n requires redrawing indices from $\{1, \ldots, n\}$. For SIS to perform properly, the ratio $p_0(x)/p_1(x) \propto \mathrm{e}^{-U_0(x)+U_1(x)}$ need to have a finite variance under p_1, which can be difficult to assess in practice.

Mixture importance sampling (MIS). The second is MIS with the design density defined by (for example) the energy function $U_\bullet(x) = -\log(\mathrm{e}^{-U_0} + \mathrm{e}^{-U_1})$. The associated density function is a mixture of p_0 and p_1,

$$p_\bullet(x) \propto p_0(x) + \mathrm{e}^{-\delta} p_1(x),$$

where the relative weight for p_0 is $1/(1+\mathrm{e}^{-\delta})$ depending on δ. The mixture density $p_\bullet$ may be multimodal, but the two regions B_1 and B_2 are "connected" under $p_\bullet$ through probability mass from p_1 instead of completely separated in p_0, so that Langevin sampling from $p_\bullet$ may achieve global mixing. Given an initial value $\tilde{x}_0$, the MIS algorithm iterates for $i = 1, \ldots, n$ as follows.

[Alternatively, the design density can be defined as $p_\bullet(x) = \frac{1}{2}p_0(x) + \frac{1}{2}p_1(x)$, which has (equal) relative weights independent of δ. But the associated energy function $U_\bullet(x) = -\log(\mathrm{e}^{-U_0} + \mathrm{e}^{-U_1+\delta})$ depends on δ. An MIS algorithm similar as below can be derived, but both steps (i)–(ii) requiring the value of δ.]

(i) Draw $\tilde{x}_i$ using MALA transition for target distribution $p_\bullet$ (with energy $U_\bullet$) from initial value $\tilde{x}_{i-1}$.

(ii) Draw $\tilde{s}_i = 1$ or 2 with probability $v(\tilde{x}_i)$ or $1 - v(\tilde{x}_i)$ respectively, where

$$v(x) = \frac{\frac{1}{1+\mathrm{e}^{-\delta}} p_0(x)}{p_\bullet(x)} = \frac{\mathrm{e}^{-U_0(x)}}{\mathrm{e}^{-U_0(x)} + \mathrm{e}^{-U_1(x)}},$$

which is independent of δ.

Then $\{\tilde{x}_i : \tilde{s}_i = 1, i = 1, \ldots, n\}$ is a valid sample from p_0. Compared with SIS, the weight $v(x)$ automatically has a finite variance under $p_\bullet$, because

$$\int \left\{\frac{p_0(x)}{p_\bullet(x)}\right\}^2 p_\bullet(x)\,\mathrm{d}x = (1+\mathrm{e}^{-\delta}) \int \frac{p_0^2(x)}{p_0(x) + \mathrm{e}^{-\delta} p_1(x)}\,\mathrm{d}x \le 1 + \mathrm{e}^{-\delta}.$$

Hence MIS can be more stable than SIS. On the other hand, the effectiveness of MALA sampling from the mixture $p_\bullet$ may be limited because p_0 and p_1 are spatially of different scales and hence different Langevin step sizes are desired for traversing the regions of p_0 and p_1.

Metropolis within Gibbs mixture sampling (MGMS). The third is MGMS or, more specifically, MALA within Gibbs mixture sampling, which constitutes a special case of our sampling Algorithm 2 with only one auxiliary distribution. Consider the joint distribution

$$p(x, s) = \begin{cases} \frac{1}{1+\mathrm{e}^{-\delta}} p_0(x), & s = 1, \\ \frac{1}{1+\mathrm{e}^{\delta}} p_1(x), & s = 2, \end{cases}$$

which is called a labeled mixture (Tan 2017) because each point x is paired with a label s. The marginal density of x under $p(x, s)$ is $p_\bullet(x)$. As an energy function of $p(x, s)$, let

$$U(x, s) = \begin{cases} U_0(x), & s = 1, \\ U_1(x), & s = 2. \end{cases}$$

Given initial values $(\tilde{x}_0, \tilde{s}_0)$, the MGMS algorithm iterates for $i = 1, \ldots, n$ as follows.

(i) If $\tilde{s}_{i-1} = 1$, draw $\tilde{x}_i$ using MALA transition for target distribution p_0 (with energy U_0) from initial value $\tilde{x}_{i-1}$. If $\tilde{s}_{i-1} = 2$, draw $\tilde{x}_i$ using MALA transition for target distribution p_1 (with energy U_1) from initial value $\tilde{x}_{i-1}$.

(ii) Draw $\tilde{s}_i \in \{1, 2\}$ as in step (ii) of MIS.

Then $\{\tilde{x}_i : \tilde{s}_i = 1, i = 1, \ldots, n\}$ is a valid sample from p_0. The two steps of MGMS perform sampling from the conditional distribution $p(x|s = \tilde{s}_{i-1})$ using MALA and then sampling exactly from $p(s|x = \tilde{x}_i)$, both under the joint distribution $p(x, s)$. These two steps are called Markov move and global jump, which draws $\tilde{s}_i$ independently of $\tilde{s}_{i-1}$ (Tan 2017). [Alternatively, a local-jump scheme can be used for drawing $\tilde{s}_i$ by Metropolis–Hastings sampling from $p(s|x = \tilde{x}_i)$ with initial value $\tilde{s}_{i-1}$; this is known as serial tempering (Geyer & Thompson 1995; Marinari & Parisi 1992).] Compared with MIS, the algorithm allows MALA sampling of x with different step sizes from p_0 and p_1 in a principled manner.

The performance of MGMS (as well as MIS) depends on the (unknown) value of δ or equivalently the relative weight for p_0, i.e., $p(s = 1) = 1/(1+\mathrm{e}^{-\delta})$. If δ is too negative (tending to $-\infty$), then the relative weight for p_0 is too large (making $p_\bullet$ close to p_0 and Langevin sampling suffer local mixing). If δ is too positive (tending to ∞), then the relative weight for p_0 is too small (making the realized sample size for p_0 small even for a large n).

Self-adjusted mixture sampling (SAMS). To address the preceding issue in MGMS, an adaptive approach is to iteratively shift the energy functions U_0 and U_1 to $U_0 + \zeta_0$ and $U_1 + \zeta_1$ for some additive constants ζ_0 and ζ_1 (while fixing initial U_0 and U_1), such that $\delta = 0$ or $p(s = 1) = p(s = 2) = 1/2$ based on the new energy functions. An SA algorithm using MGMS in the sampling step leads to SAMS (Tan 2017). [Alternatively, a local-jump instead of global-jump scheme can be used in the sampling step (Liang, Liu, & Carroll 2007).] Given initial values $(\tilde{x}_0, \tilde{s}_0)$ and $\zeta_0^{(0)} = \zeta_1^{(0)} = 0$, the SAMS algorithm iterates for $i = 1, \ldots, n$ as follows.

(i) If $\tilde{s}_{i-1} = 1$, draw $\tilde{x}_i$ using MALA with energy $U_0 + \zeta_0^{(i-1)}$ from initial value $\tilde{x}_{i-1}$. If $\tilde{s}_{i-1} = 2$, draw $\tilde{x}_i$ using MALA with energy $U_1 + \zeta_1^{(i-1)}$ from initial value $\tilde{x}_{i-1}$.

(ii) Draw $\tilde{s}_i = 1$ or 2 with probability $v(\tilde{x}_i)$ or $1 - v(\tilde{x}_i)$ respectively, where

$$v(x) = \frac{\mathrm{e}^{-U_0(x)-\zeta_0^{(i-1)}}}{\mathrm{e}^{-U_0(x)-\zeta_0^{(i-1)}} + \mathrm{e}^{-U_1(x)-\zeta_1^{(i-1)}}},$$

(iii) Update

$$\zeta_0^{(i)} = \zeta_0^{(i-1)} + \gamma(-\frac{1}{2} + 1\{\tilde{s}_i = 1\}),$$
$$\zeta_1^{(i)} = \zeta_1^{(i-1)} + \gamma(-\frac{1}{2} + 1\{\tilde{s}_i = 2\}),$$

where γ is a learning rate.

Remarkably, it can be verified that the SAMS algorithm is equivalent to our learning Algorithm 1 in the special case of learning the "intercept" parameter $\zeta = (\zeta_0, \zeta_1)$, where the model energy function is parameterized as

$$U_\zeta(x, s) = \begin{cases} U_0(x) + \zeta_0, & s = 1, \\ U_1(x) + \zeta_1, & s = 2, \end{cases}$$

with fixed initial energies (U_0, U_1), and the real-data distribution of (x, s) is assumed such that the marginal probabilities of $s = 1$ and 2 are both $\frac{1}{2}$. In this sense, the proposed Algorithm 1 can be seen to extend SAMS for learning non-intercept parameters in EBMs.

IV | REFORMULATION OF DIFFUSION RECOVERY LIKELIHOOD

We provide a reformulation of the recovery likelihood approach of Gao et al. (2021), as mentioned in Section 4. For simplicity, we consider only a single pair of observation $(x^{(t-1)}, x^{(t)})$, satisfying $x^{(t)} = \sqrt{\alpha_t} x^{(t-1)} + \sqrt{1-\alpha_t}\varepsilon^{(t)}$, where $\varepsilon^{(t)} \sim \mathrm{N}(0, I)$. Assume that $x^{(t-1)}$ satisfies an EBM with energy function $U_\theta(x^{(t-1)}, t-1)$, i.e.,

$$p_\theta(x^{(t-1)}) \propto \exp(-U_\theta(x^{(t-1)}, t-1)).$$

Then $x^{(t-1)}$ given $x^{(t)}$ satisfies a conditional EBM, with the conditional density

$$p_\theta(x^{(t-1)}|x^{(t)}) \propto \exp\left\{-U_\theta(x^{(t-1)}, t-1) - \frac{\|x^{(t)} - \sqrt{\alpha_t} x^{(t-1)}\|_2^2}{2(1-\alpha_t)}\right\},$$

up to a multiplicative constant, free of $x^{(t-1)}$ but depending on $x^{(t)}$. The gradient of $\log p_\theta(x^{(t-1)}|x^{(t)})$ can be shown to be

$$\frac{\partial}{\partial\theta} \log p_\theta(x^{(t-1)}|x^{(t)}) = -\frac{\partial}{\partial\theta} U_\theta(x^{(t-1)}, t-1) + \mathbb{E}_{p_\theta(z^{(t-1)}|x^{(t)})}\left\{\frac{\partial}{\partial\theta} U_\theta(z^{(t-1)}, t-1)\right\}, \tag{S7}$$

where $\mathbb{E}_{p_\theta(z^{(t-1)}|x^{(t)})}(\cdot)$ denotes the (conditional) expectation over $z^{(t-1)}$ with respect to $p_\theta(z^{(t-1)}|x^{(t)})$. Note that $x^{(t-1)}$ is the data point observed together with $x^{(t)}$, and $z^{(t-1)}$ is a general point. In the recovery likelihood approach, the gradient (S7) is approximated by

$$-\frac{\partial}{\partial\theta} U_\theta(x^{(t-1)}, t-1) + \frac{\partial}{\partial\theta} U_\theta(\tilde{x}^{(t-1)}, t-1), \tag{S8}$$

where $\tilde{x}^{(t-1)}$ is drawn by running (for example) L Langevin sampling steps targeting the conditional density $p_\theta(z^{(t-1)}|x^{(t)})$, with the initial value set to $x^{(t)}$.

For the reformulation, we note that the pair $(x^{(t-1)}, x^{(t)})$ satisfies a "bivariate" EBM, with the joint density

$$p_\theta(x^{(t-1)}, x^{(t)}) \propto \exp\left\{-U_\theta(x^{(t-1)}, t-1) - \frac{\|x^{(t)} - \sqrt{\alpha_t} x^{(t-1)}\|_2^2}{2(1-\alpha_t)}\right\},$$

up to a multiplicative constant, free of $(x^{(t-1)}, x^{(t)})$. The "bivariate" EBM $p_\theta(x^{(t-1)}, x^{(t)})$ is more fundamental than the conditional EBM $p_\theta(x^{(t-1)}|x^{(t)})$, because the latter is derived from the former, but not conversely. The gradient of $\log p_\theta(x^{(t-1)}, x^{(t)})$ can be shown to be

$$\frac{\partial}{\partial\theta} \log p_\theta(x^{(t-1)}, x^{(t)}) = -\frac{\partial}{\partial\theta} U_\theta(x^{(t-1)}, t-1) + \mathbb{E}_{p_\theta(z^{(t-1)}, z^{(t)})}\left\{\frac{\partial}{\partial\theta} U_\theta(z^{(t-1)}, t-1)\right\}, \tag{S9}$$

where $\mathbb{E}_{p_\theta(z^{(t-1)}, z^{(t)})}(\cdot)$ denotes the expectation with respect to $p_\theta(z^{(t-1)}, z^{(t)})$. Note that $(x^{(t-1)}, x^{(t)})$ is the observed pair, and $(z^{(t-1)}, z^{(t)})$ is a general pair of points. Consider training the bivariate EBM by CD (contrastive divergence). The expectation term in (S9) can be approximated by

$$-\frac{\partial}{\partial\theta} U_\theta(x^{(t-1)}, t-1) + \frac{\partial}{\partial\theta} U_\theta(\tilde{x}^{(t-1)}, t-1), \tag{S10}$$

where $(\tilde{x}^{(t-1)}, \tilde{x}^{(t)})$ is drawn from an Markov transition kernel targeting $p_\theta(z^{(t-1)}, z^{(t)})$, with the initial value set to the observed data point $(x^{(t-1)}, x^{(t)})$. Suppose that $(\tilde{x}^{(t-1)}, \tilde{x}^{(t)})$ is drawn as follows, given the initial value $(x^{(t-1)}, x^{(t)})$:

- Draw $\tilde{x}^{(t-1)}$ by running L Langevin sampling steps targeting the conditional density $p_\theta(z^{(t-1)}|x^{(t)})$, with the initial value set to $x^{(t)}$.
- Draw $\tilde{x}^{(t)}$ from $\mathrm{N}(\sqrt{\alpha_t}\tilde{x}^{(t-1)}, (1-\alpha_t)I)$.

The two operations constitute one step of Metropolis within Gibbs sampling (or MCMC within Gibbs sampling): sample $\tilde{x}^{(t-1)}$ given $x^{(t)}$ using Langevin sampling, and then sample $\tilde{x}^{(t)}$ given $\tilde{x}^{(t-1)}$. In fact, drawing $\tilde{x}^{(t)}$ given $\tilde{x}^{(t-1)}$ can be skipped, because the expectation term in (S9) involves only $z^{(t-1)}$, but not $z^{(t)}$. From the preceding discussion, the approximate gradient (S10) is the same as the approximate gradient (S8): not only the two expressions are of the same form, but also $\tilde{x}^{(t-1)}$ is generated in the same way given $x^{(t)}$. In this sense, the recovery likelihood approach of Gao et al. (2021) is equivalent to training the "bivariate" EBM $p_\theta(x^{(t-1)}, x^{(t)})$ via CD.

V | EXPERIMENT DETAILS AND ADDITIONAL RESULTS

V.1 | Gaussian mixture 1D example

We train the EBM with persistent initialization and MALA sampling in the 1D Gaussian mixture example. The energy function is parameterized as $U_\theta(x) = (x/10)^2 + \theta^{\mathrm{T}} h_\theta(x)$, where the ReLU basis function $h_\theta(x)$ is

$$h_\theta(x) = ((x-\xi_1)_+, \ldots, (x-\xi_K)_+)^{\mathrm{T}},$$

where ξ_j's are the knots and $a_+ = a$ if $a \geq 0$ or 0 if $a < 0$. Equi-spaced knots from -4 to 4 by $.1$ are used, as mentioned in the main text. The prior quadratic term $(x/10)^2$ is needed to guarantee the energy function is well-defined, i.e., the corresponding density function is integrable. We train the model for 1,200 iterations with a starting learning rate $\gamma = 0.2$. Learning rate decays by a factor of 0.2 at the 600, 800, and 1,000 iteration milestones to ensure convergence. The Langevin step size σ is set to 0.1, and the number of Langevin steps L is set to 10 per training iteration. The replay buffer size is set to 1,000 (same as the training dataset).

We plot the estimated energy functions of five more independent runs in Figure S1. The estimated energies near -2 and 2 exhibit varying directions of relative magnitudes in different training runs. This shows the difficulty in learning a global energy function using persistent training of EBM in the presence of separated modes by low-density barriers.

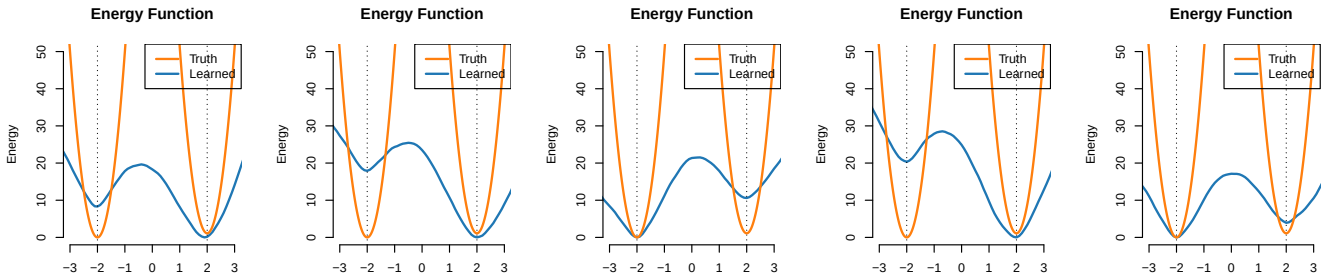


Figure S1 Learned energy functions from five independent runs by persistently training EBMs in 1D example

V.2 | Four-ring example

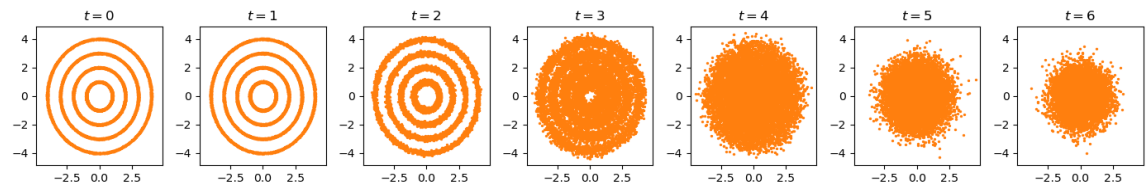
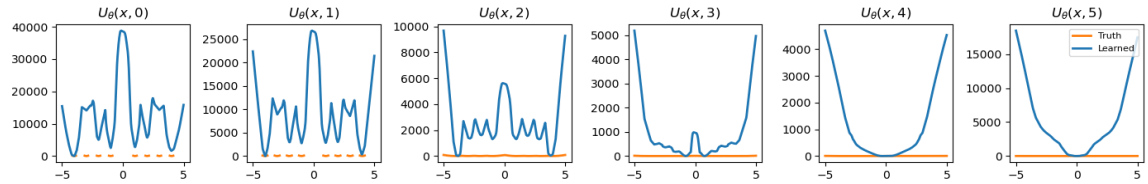
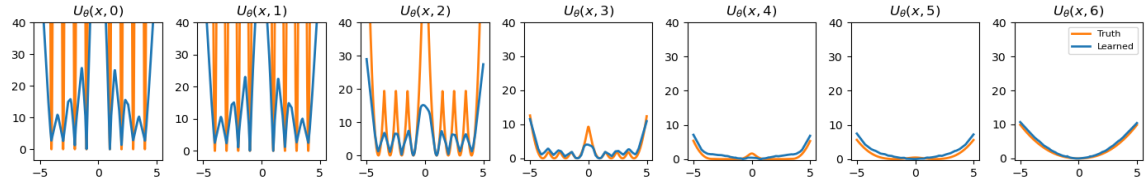
Training data. For a 2D vector $x = (x_1, x_2)$, suppose that the radius $r = \sqrt{x_1^2 + x_2^2}$ of a ring follows a one-sided truncated normal distribution to $(0, \infty)$, and the angle $\theta = \arctan(x_1/x_2)$ follows a uniform distribution over $[0, 2\pi]$. The means of the radii of the four rings are $1, 2, 3, 4$, and the standard deviations are 0.01. We draw a total of 50,000 points from the four rings with equal probability as the training set.

Network. We choose a four-layer MLP, which has 128 equal hidden neurons in each of the middle layers. EBMs trained with different initialization use ReLU as the activation function, while DA-EBM and DRL use the Softplus activation function. We use sinusoidal time embedding and set $T = 6$ for DA-EBM and DRL. Both methods use the same cumulative-sum diffusion scheduling described in Section V.3 with $\delta_1 = 0.01$ and $\delta_T = 0.3$.

Training and sampling. We train all methods with a multi-step learning rate decay schedule. DRL are trained with 500 epochs, and all other methods are trained with 200 epochs. The decay milestones are at 7/10, 8/10, and 9/10 of the total training epoch, and the decay factor is 0.1. The initial learning rate is set to 5×10^{-4}. The DRL training is strictly based on Algorithms 1 and 2 in Gao et al. (2021), without the additional variations in the image experiments (see Section V.3),

All EBM methods except our proposed DA-EBM use Langevin sampling without acceptance-rejection by implementation conventions in the literature. For EBM training, we consider two different Langevin step sizes, $\sigma = 0.005$ and $\sigma = 0.01$, and pick up the best result for illustration. For the DRL method, we set the step size $\sigma_t = b\sqrt{1 - \alpha_{t+1}^2}$ for $t = 0, \ldots, T-1$ with two possible choices $b = 0.02$ and $b = 0.01$, and present the best result. The step size of DA-EBM is dynamically adjusted depending on the average acceptance rate for each time label over each epoch. The adjustment method is described in Section V.3. The initial values are set to $\sigma_t = b\sqrt{1 - \alpha_t^2}$ for $t = 1, \ldots, T$ and $\sigma_0 = b\sqrt{1 - \alpha_1^2}$ with $b = 0.1$. During training per iteration, the number of Langevin steps L is set to $L = 200$ for hybrid initialization and $L = 4000$ for noise initialization due to training instability. For all other methods, L is set to $L = 40$.

Table S1 Additional results in the four-ring example

Method	Performance
Diffusion Data Info.	
DRL	
DA-EBM	

Long-run sampling and post-training sampling. After training, long-run samples are obtained after 100k Langevin steps, in parallel, starting from an independent draw of 10,000 points from the four-ring distribution, while post-training samples are obtained starting from 10,000 standard Gaussian noises. For post-training sampling, EBMs trained with noise and hybrid initialization use L and $20L$ Langevian steps. EBMs trained with data initialization and persistent initialization use 10,000 Langevian steps. DRL uses $6L$ Langevian steps in the sequential conditional sampling for post-training sampling. Our DA-EBM runs 10,000 parallel chains for a total of 250 MGMS sampling transitions (Algorithm 2), and time 0 samples are identified as post-training samples.

Additional results. We demonstrate the learned energy functions for all different time steps in Table S1. For DRL, the energy function at time 6, $U_\theta(x, 6)$, is assumed to be that of standard Gaussian. As can be seen, our DA-EBM learns the energy functions $U_\theta(x, t)$ that reasonably match with the true negative log densities (up to an additive constant) of diffusion data across all time steps $t = 0, \ldots, 6$, while DRL fails to do that.

Table S2 Model architecture for the experiments on the image data

(a) Network

3x3 Conv2D, 128
1 ResBlock, 128
Downsample 2×2
1 ResBlock, 256
Downsample 2×2
1 ResBlock, 256
Downsample 2×2
1 ResBlock, 256
SiLU, global sum
+ Dense(SiLU(temb))
SUM

(b) ResBlock

SiLU, 3×3 Conv2D
+ Dense(SiLU(temb))
SiLU, 3×3 Conv2D
+ Conv2D(input)

(c) Time embedding (temb)

Sinusoidal Embedding
Dense, SiLU
Dense

Table S3 Main differences in training implementations of different methods

	β_1 in Adam	lr γ	Normalization	Image Preprossessing	T	Number of Param.
EBM Pers.	0.0	2×10^{-5}	-	Gaussian (std=0.03)	-	4.7M
EBM Hybrid	0.0	2×10^{-5}	-	Gaussian (std=0.03)	-	4.7M
EBM Noise	0.0	5×10^{-6}	-	Gaussian (std=0.03)	-	4.7M
DRL	0.9	2×10^{-4}	Spectral	None	50	4.7M
DA-EBM	0.0	1×10^{-5}	-	Uniform	50	4.7M
Glow	0.9 [1]	5×10^{-4}	Actnorm	Gaussian (std=0.01)	-	29.5M
DDPM	0.0	2×10^{-4}	Group	Uniform	$[0, 999]$	20.1M

V.3 | Image experiments

Model architecture. We adopt a similar network architecture as in Gao et al. (2021), which is a variation of the WideResNet (Zagoruyko & Komodakis 2016) and agrees with the bottom-up part of the U-Net architecture used in Ho et al. (2020). We use uses the Sigmoid Linear Unit (SiLU) as the activation function. Time label t is encoded by Transformer sinusoidal positional embedding (Vaswani et al. 2017). We list the network structure in Table S2.

Training basics. We train all EBM methods with Adam optimizer (Kingma & Ba 2015) for 120k iterations, with a mini-batch size of 200. The first 6k iterations are set as the warm-up stage, during which the learning rate linearly increases from 0 to the desired learning rate. The last 48k iterations are set as the annealing stage, during which the learning rate linearly decreases to a small factor of its original level. The linear decay factor is set to 10^{-5}. We currently use Kaiming initialization (He, Zhang, Ren, & Sun 2015). Image is rescaled to range $[-1, 1]$, and additional Gaussian noise or uniform dequantization is added to stabilize training for different methods, as detailed in Table S3.

We find out that, empirically, persistent training does not work with normalization (e.g., batch/spectral normalization). Currently, we remove all normalizations in the network for EBM training methods, except that we keep the spectral normalization for the DRL method. Otherwise, the training fails to learn meaningful models or diverges easily.

DRL, Glow, cDDPM. DRL training is based on the released code[2] of the $T6$ setting in Gao et al. (2021). Note that the code involves some additional variations from the method described in Section 3.5 of the paper. First, in the code, the energy function of the second last diffusion time $U_\theta(x, T-1)$ is trained in the same way as using ML with noise initialization instead of using recovery likelihood for $x^{(T-1)}$ given $x^{(T)}$. Second, the Langevin

[1] Adamax is used instead of Adam.

[2] `https://github.com/ruiqigao/recovery_likelihood`

update for drawing from $p_\theta(x^{(t-1)}|x^{(t)})$ is modified as follows with the energy $U_\theta(x, t-1)$ scaled by σ^2_{t-1}:

$$\tilde{x}_l^{(t-1)} = \tilde{x}_{l-1}^{(t-1)} - \frac{\sigma^2_{t-1}}{2}\left\{\frac{\nabla_x U_\theta(\tilde{x}_{l-1}^{(t-1)}, t-1)}{\sigma^2_{t-1}} + \frac{x^{(t)} - \sqrt{\alpha_t}\tilde{x}_{l-1}^{(t-1)}}{1-\alpha_t}\right\} + \sigma_{t-1}\varepsilon, \tag{S11}$$

for $l = 1, \ldots, L$, with the initial value $\tilde{x}_0^{(t-1)} = x^{(t)}$ and the Langevin step size $\sigma_t = bc_t\sqrt{1-\alpha^2_{t+1}}$ for an additional factor c_t. This is not proper Langevin dynamics for the conditional EBM $p_\theta(x^{t-1})|x^{(t)})$ in Eq. (16) of Gao et al. (2021). Finally, a time-dependent weight factor is added to the gradient in updating the parameter. The effects of these additional variations are unclear, but we keep them in our training implementation; otherwise training tends to break down. The parameter b is set to 0.02 and other parameters remain as the default in the DRL code. For DRL, we keep the improper Langevin update as (S11) in sequential conditional sampling for post-training image generation, whereas we use the (proper) Langevin update for the learned energy $U_\theta(x, 0)$ without rescaling to investigate long-run sampling.

The Glow model used in the OOD experiments is trained using the GitHub repository, Glow-PyTorch[3]. We set the mini-batch size to 100 and use the additive coupling layer. Other parameters remain as the default training options in the repository. We also need to add Gaussian noise to the image for preprocessing. Otherwise, log-likelihoods evaluated on the test set get infinity or NaN values. The cDDPM (continuous-time DDPM) model is trained mainly based on the released code of Song et al. (2021), score_sde_pytorch [4]. We train the cDDPM for 600k iterations with the continuous-time objective function (7) in Song et al. (2021). The continuous time step avoids ad-hoc interpolation when evaluating the model density by solving the probability flow SDE. We remove the attention layer in the network structure and choose the network such that the bottom-up part agrees with our network structure in Table S2.

The main differences in training implementations for different methods are summarized in Table S3. Normalized density estimates are directly available from Glow, and can be obtained by solving a probability flow ODE for cDDPM.

Langevin sampling. The replay buffer size is set to 50,000. The number of Langevin steps L is set to $L = 80$ for the noise-initialized EBM and $L = 40$ for all other methods. EBMs trained with different initializations all use a Langevin step size 0.005. For our DA-EBM, the step sizes for different time labels are initialized to be 0.01. We turn on acceptance-rejection in the MGMS algorithm after the warm-up stage and adjust step sizes σ_t dynamically based on acceptance rates as follows.

The average acceptance rate for each time label is calculated every 100 iterations during training, and the corresponding step size is then adjusted. Similar to the step size tuning in Appendix V.4 in Song and Tan (2021), we adjust the step size such that the acceptance rate of each time label is between 0.6 and 0.8 during the training. When the acceptance rate is smaller than 0.6, we decrease σ_t by a factor of $1/(1+2\tau)$, and when the acceptance rate is larger than 0.8, we increase σ_t by a factor of $(1+\tau)$, where τ is an adjustment value taken to be $\tau = 0.1$ in all our experiments. The increase and decrease factors are designed to be asymmetric. Otherwise, the step sizes fall in a fixed set of values determined by the initial value.

Long-run sampling and post-training sampling. We report 100k or 500k long-run samples, obtained after running 100k or 500k Langevin sampling steps, starting from real images, with the learned energy function. For DRL and DA-EBM, this is the learned energy function $U_{\hat{\theta}}(x, 0)$ at time 0. Post-training samples are obtained starting from standard Gaussian noises. EBMs trained with noise and hybrid initializations use L and $20L$ Langevian steps for post-training image generation, while EBM trained with persistent initialization uses 100k Langevian steps. DRL uses $50L$ Langevian steps in sequential conditional sampling. Our post-training image generation procedure is described below.

For post-training sampling, the Langevin step size is fixed to be the final step size during training. We start Markov chains from standard Gaussian noises and Repeatedly run the MGMS algorithm (Algorithm 2) until MCMC samples become realistic images. In this work, we regard a sampled configuration x as a realistic image when sampled configurations have been classified as time label $t = 0$ for 50 times in the Repeated MGMS runs (see Algorithm 3).

Diffusion scheduling. While DRL and cDDPM use original scheduling in their released codes, DA-EBM uses a cumulative-sum diffusion scheduling as follows. We set $\alpha_1, \ldots, \alpha_T$ such that the standard deviation $\sqrt{1-\alpha_t}$ in the diffusion process is increasing with respect to t as a cumulative sum. Specifically, we set $T = 50$ and set δ_i to increase from $\delta_1 = 0.0002$ to $\delta_T = 0.02$ with equal spaces. Then we determine α_t according to $\sqrt{1-\alpha_t} = \sum_{i=1}^t \delta_i$. The diffusion scheduling is not particularly optimized and can possibly be improved for better training performance.

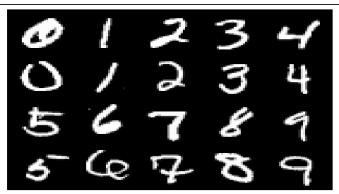
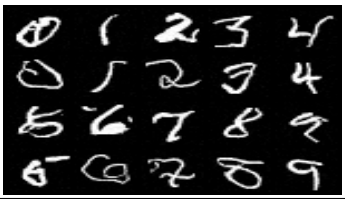
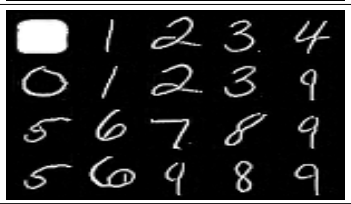
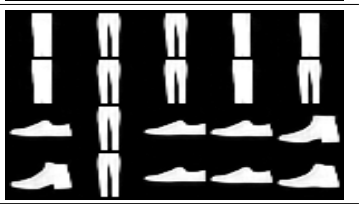
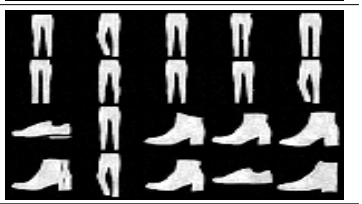
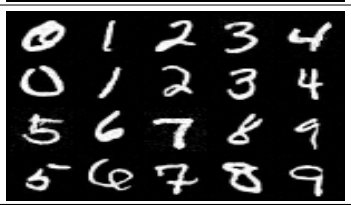
Additional sampling results. We show additional long-run samples obtained from 100k MCMC steps in Table S4. The results are similar to 500k long-run samples, but may be less obvious for visual inspection. Additional post-training samples of DA-EBM are presented in Table S5.

Additional OOD results. We plot the ROC curves and PR curves in Figure S2 and summarize the AUC-PR results in Table S7. A comparison of the computational cost of likelihood evaluation for DA-EBM and cDDPM is shown in Table S6. The computation time for cDDPM is huge compared to the almost negligible time for DA-EBM, which shows the advantage of learning EBMs for OOD detection.

[3] https://github.com/y0ast/Glow-PyTorch

[4] https://github.com/yang-song/score_sde_pytorch

Table S4 Results of (100k) long-run sampling on MNIST and FashionMNIST

	MNIST	Fashion-Mnist
Train. Data (Long-run starting points)		
Method	Long-run	Long-run
EBM PERS.		
EBM NOISE		
EBM HYBRID (5% noise)		
DRL		
DA-EBM		

VI | LIMITATIONS

Our post-training sampling is more costly than short-run EBM, hybrid-initialized EBM, and DRL. The average Langevin steps needed for generating 100 images are about 274,162 and 723,844 in Fashion-MNIST and MNIST experiments. To scale our method to more complex images and larger models, further investigation is needed. Nevertheless, we believe this work presents a necessary step towards improving persistent training of EBMs for complex data like natural images, to achieve post-sampling generation and proper density estimation simultaneously.

Table S5 Additional images generated by DA-EBM in post-training sampling

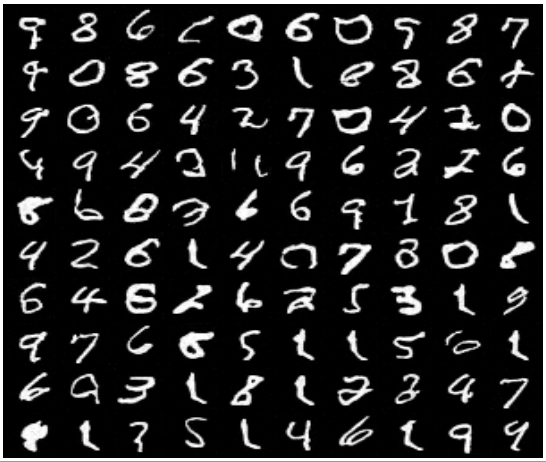
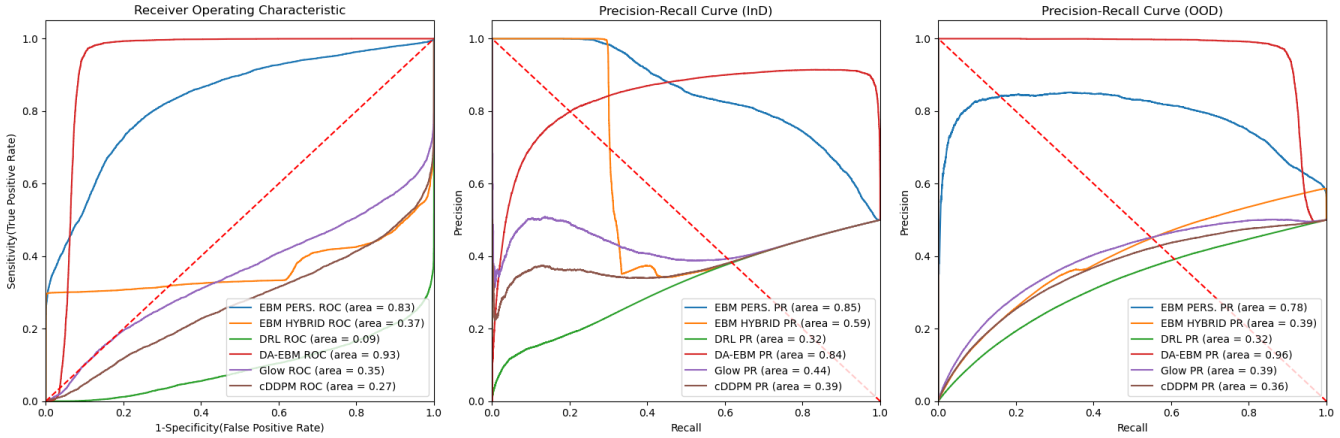
Method	MNIST	FashionMnist
DA-EBM		

Table S6 Elapsed time for likelihood evaluation on test split of datasets using a single NVIDIA GeForce GTX 1080Ti GPU

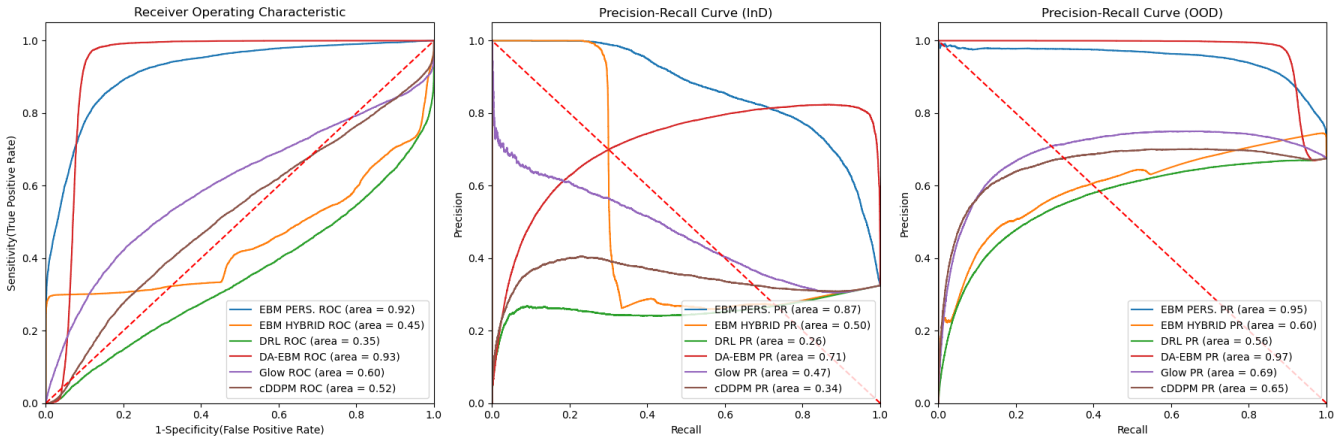
	Fashion-MNIST	MNIST	EMNIST	KMNIST
# of samples	10,000	10,000	14,800	10,000
DA-EBM	$\approx 2s$	$\approx 2s$	$\approx 4s$	$\approx 2s$
cDDPM	1h 49m 4s	2h 6m 49s	4h 30m 24s	2h 11m 17s

Table S7 OOD AUC-PR results. InD column indicates that InD examples are used as the positive class, while OOD column indicates that OOD examples are used as the positive class.

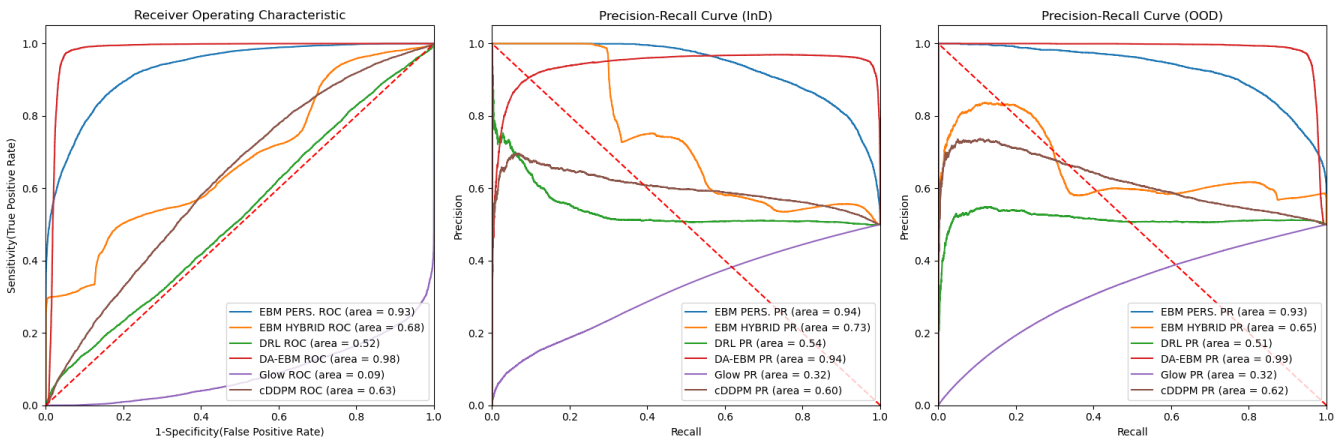
	InD			OOD		
Method	Mnist	EMnist	KMnist	Mnist	EMnist	KMnist
EBM PERS.	**0.85**	**0.87**	**0.94**	0.78	0.95	0.93
EBM HYBRID	0.59	0.50	0.73	0.39	0.60	0.65
DRL	0.32	0.26	0.54	0.32	0.56	0.51
DA-EBM	0.84	0.71	**0.94**	**0.96**	**0.97**	**0.99**
GLOW	0.44	0.47	0.32	0.39	0.69	0.62
cDDPM	0.39	0.34	0.60	0.36	0.65	0.32



(a) Fashion-MNIST (InD) vs MNIST(OOD)



(b) Fashion-MNIST (InD) vs EMNIST(OOD)



(c) Fashion-MNIST (InD) vs KMNIST(OOD)

Figure S2 ROC curves, PR curves

Supplement References

Chen, Y., Chen, J., Dong, J., Peng, J., & Wang, Z. (2019). Accelerating nonconvex learning via replica exchange Langevin diffusion. In *International conference on learning representations*.

Ge, R., Lee, H., & Risteski, A. (2018). Beyond log-concavity: Provable guarantees for sampling multi-modal distributions using simulated tempering Langevin Monte Carlo. In *Advances in neural information processing systems*.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the ieee international conference on computer vision* (pp. 1026–1034).

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In *International conference on learning representations*.

Liang, F., Liu, C., & Carroll, R. J. (2007). Stochastic approximation in monte carlo computation. *Journal of the American Statistical Association*, *102*, 305–320.

Song, Z., & Tan, Z. (2021). Hamiltonian-assisted metropolis sampling. *Journal of the American Statistical Association*, 1–19.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., … Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*.

Zagoruyko, S., & Komodakis, N. (2016). Wide residual networks. *arXiv preprint arXiv:1605.07146*.