DIFFUSION-MODEL-ASSISTED SUPERVISED LEARNING OF GENERATIVE MODELS FOR DENSITY ESTIMATION

Yanfang Liu,[1] Minglei Yang,[2] Zezhong Zhang,[1] Feng Bao,[3] Yanzhao Cao,[4] & Guannan Zhang[1,*]

[1]*Computer Science and Mathematics Division, Oak Ridge National Laboratory, Oak Ridge, Tennessee 37831, USA*

[2]*Fusion Energy Science Division, Oak Ridge National Laboratory, Oak Ridge, Tennessee 37831, USA*

[3]*Department of Mathematics, Florida State University, Tallahassee, Florida 32306, USA*

[4]*Department of Mathematics and Statistics, Auburn University, Auburn, Alabama 36849, USA*

*Address all correspondence to: Guannan Zhang, Computer Science and Mathematics Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA, E-mail: zhangg@ornl.gov

Original Manuscript Submitted: 10/20/2023; Final Draft Received: 1/7/2024

We present a supervised learning framework of training generative models for density estimation. Generative models, including generative adversarial networks (GANs), normalizing flows, and variational auto-encoders (VAEs), are usually considered as unsupervised learning models, because labeled data are usually unavailable for training. Despite the success of the generative models, there are several issues with the unsupervised training, e.g., requirement of reversible architectures, vanishing gradients, and training instability. To enable supervised learning in generative models, we utilize the score-based diffusion model to generate labeled data. Unlike existing diffusion models that train neural networks to learn the score function, we develop a training-free score estimation method. This approach uses mini-batch-based Monte Carlo estimators to directly approximate the score function at any spatial-temporal location in solving an ordinary differential equation (ODE), corresponding to the reverse-time stochastic differential equation (SDE). This approach can offer both high accuracy and substantial time savings in neural network training. Once the labeled data are generated, we can train a simple, fully connected neural network to learn the generative model in the supervised manner. Compared with existing normalizing flow models, our method does not require the use of reversible neural networks and avoids the computation of the Jacobian matrix. Compared with existing diffusion models, our method does not need to solve the reverse-time SDE to generate new samples. As a result, the sampling efficiency is significantly improved. We demonstrate the performance of our method by applying it to a set of 2D datasets as well as real data from the University of California Irvine (UCI) repository.

KEY WORDS: *score-based diffusion models, density estimation, curse of dimensionality, generative models, supervised learning*

2689–3967/24/$35.00

1. INTRODUCTION

Density estimation involves the approximation of the probability density function (PDF) of a given set of observation data points. The overarching goal is to characterize the underlying structure of the observation data. Generative models belong to a class of machine learning models designed to model the underlying probability distribution of a given dataset, enabling the generation of new samples that are statistically similar to the original data. Many methods for generative models have been proposed over the past decade, including variational auto-encoders (VAEs) (Kingma and Welling, 2014), generative adversarial networks (GANs) (Goodfellow et al., 2014), normalizing flows (Kobyzev et al., 2020), and diffusion models (Yang et al., 2023). Generative models have been successfully used in a variety of applications, including image synthesis (Cai et al., 2020; Goodfellow et al., 2014; Ho et al., 2022; Meng et al., 2022; Song and Ermon, 2019), image denoising (Ho et al., 2020; Ledig et al., 2017; Luo and Hu, 2021; Sohl-Dickstein et al., 2015), anomaly detection (Papamakarios et al., 2017; Schlegl et al., 2017), and natural language processing (Austin et al., 2021; Hoogeboom et al., 2021; Li et al., 2022; Ma et al., 2019; Savinov et al., 2022; Yu et al., 2022). The key idea behind recent generative models is to exploit the superior expressive power and flexibility of deep neural networks to detect and characterize complicated geometries embedded in the possibly high-dimensional observational data.

Generative models for density estimation are usually considered as unsupervised learning, primarily due to the absence of labeled data. Various unsupervised loss functions have been defined to train the underlying neural network models in generative models, including adversarial loss for GANs (Goodfellow et al., 2014), the maximum likelihood loss for normalizing flows (Kobyzev et al., 2020), and the score matching losses for diffusion models (Hyvärinen, 2005; Song et al., 2019; Vincent, 2011). Despite the success of the generative models, there are several issues resulting from the unsupervised training approach. For example, the training of GANs may suffer from mode collapse, vanishing gradients, and training instability (Salimans et al., 2016). The maximum likelihood loss used in normalizing flows requires efficient computation of the determinant of the Jacobian matrix, which places severe restrictions on networks' architectures. If labeled data can be created based on the observational data without complicated training, the generator in the generative models, e.g., the decoder in VAEs or normalizing flows, can be trained in a supervised manner, which can circumvent the issues in unsupervised training.

In this work, we propose a supervised learning framework of training generative models for density estimation. The key idea is to use the score-based diffusion model to generate labeled data and then use the simple mean squared error (MSE) loss to train the generative model. A diffusion model can transport the standard Gaussian distribution to a complex target data distribution through a reverse-time diffusion process in the form of a stochastic differential equation (SDE), and the score function in the drift term guides the reverse-time SDE towards the data distribution. Since the standard Gaussian distribution is independent of the target data distribution, the information of the data distribution is fully stored in the score function. Thus, we use the reverse-time SDE to generate the labeled data. Unlike existing diffusion models that train neural networks to learn the score function (Bao et al., 2023; Shi et al., 2022; Song et al., 2021), we develop a training-free score estimation that uses mini-batch-based Monte Carlo estimators to directly approximate the score function at any spatial-temporal location in solving the reverse-time SDE. Numerical examples in Section 4 demonstrate that the training-free score estimation approach offers sufficient accuracy and saves significant computing cost on training neural networks in the meantime. Once the labeled data are generated, we can train a simple

fully connected neural network to learn the generator in the supervised manner. Compared with existing normalizing flow models, our method does not require the use of reversible neural networks and avoids the computation of the Jacobian matrix. Compared with existing diffusion models, our method does not need to solve the reverse-time SDE to generate new samples. This way, the sampling efficiency is significantly improved.

The rest of this paper is organized as follows. In Section 2, we briefly introduce the density estimation problem under consideration. In Section 3, we provide a comprehensive discussion of the proposed method. Finally in Section 4, we demonstrate the performance of our method by applying it to a set of 2D datasets as well as real data from the University of California Irvine (UCI) repository.

2. PROBLEM SETTING

We are interested in learning how to generate an unrestricted number of samples of a target d-dimensional random variable, denoted by

$$X \in \mathbb{R}^d \text{ and } X \sim p(x), \tag{1}$$

where p is the PDF of X. We aim to achieve this from a given dataset, denoted by $\mathcal{X} = \{x_1, x_2, \ldots, x_J\} \subset \mathbb{R}^d$, and from its PDF p. To this end, we aim at building a parameterized generative model, denoted by

$$X = F(Y; \theta) \text{ with } Y \in \mathbb{R}^d, \tag{2}$$

which is a transport model that maps a reference variable Y (usually following a standard Gaussian distribution) to the target random variable X. Once the optimal value for the parameter θ is obtained by training the model, we can generate samples of X by drawing samples of Y and pushing them through the trained generative model.

This problem has been extensively studied in the machine learning community using normalizing flow models (Creswell et al., 2018; Dinh et al., 2016; Grathwohl et al., 2018; Guo et al., 2022; Kobyzev et al., 2020; Papamakarios et al., 2017; Rezende and Mohamed, 2015), where $F(y; \theta)$ is defined as a bijective function. One major challenge for training the generative model $F(y; \theta)$ is the lack of labeled training data; i.e., there is no corresponding sample of Y for each sample $x_j \in \mathcal{X}$. As a result, the model $F(y; \theta)$ cannot be trained via the simplest supervised learning using the MSE loss. Instead, an indirect loss defined by the change of variables formula, i.e., $p(x) = p(F^{-1}(x))|\det \mathbf{D}(F^{-1}(x))|$, is extensively used to train normalizing flow models in an unsupervised manner. The drawback of this training approach is that specially designed reversible architectures for $F(y; \theta)$ need to be used to efficiently perform backpropagation through the computation of $|\det \mathbf{D}(F^{-1}(x))|$. In this paper, we address this challenge by using score-based diffusion models to generate labeled data such that the generative model $F(y; \theta)$ can be trained in a supervised manner and F^{-1} is no longer needed in the training.

3. METHODOLOGY

In this section, we present in detail the proposed method. We briefly review the score-based diffusion model in Section 3.1. In Section 3.2, we introduce the training-free score estimation approach that is needed for generating the labeled data. The main algorithm for the supervised learning of the generative model $F(y; \theta)$ is presented in Section 3.3.

3.1 The Score-Based Diffusion Model

In this subsection, we will provide a brief overview of score-based diffusion models [see Song et al. (2019, 2021) for more details]. The model under consideration consists of a forward SDE and a reverse-time SDE defined in a standard temporal domain $[0, 1]$. The forward SDE, defined by

$$dZ_t = b(t)Z_t dt + \sigma(t)dW_t \text{ with } Z_0 = X \text{ and } Z_1 = Y, \tag{3}$$

is used to map the target random variable X (i.e., the initial state Z_0) to the standard Gaussian random variable $Y \sim \mathcal{N}(0, \mathbf{I}_d)$ (i.e., the terminal state Z_1). There is a number of choices for the drift and diffusion coefficients in Eq. (3) to ensure that the terminal state Z_1 follows $\mathcal{N}(0, \mathbf{I}_d)$ [see Song et al. (2021), Vincent (2011), Ho et al. (2020), and Lu et al. (2022) for details]. In this work, we choose $b(t)$ and $\sigma(t)$ in Eq. (3) as

$$b(t) = \frac{d(\log \alpha_t)}{dt} \quad \text{and} \quad \sigma^2(t) = \frac{d\beta_t^2}{dt} - 2\frac{d(\log \alpha_t)}{dt}\beta_t^2, \tag{4}$$

where the two processes α_t and β_t are defined by

$$\alpha_t = 1 - t, \quad \beta_t^2 = t \quad \text{for} \quad t \in [0, 1]. \tag{5}$$

Because the forward SDE in Eq. (3) is linear and driven by an additive noise, the conditional PDF $Q(Z_t|Z_0)$ for any fixed Z_0 is a Gaussian. In fact, the choice of the $b(t)$ and $\sigma^2(t)$ ensures the conditional distribution

$$Q(Z_t|Z_0) = \mathcal{N}(\alpha_t Z_0, \beta_t^2 \mathbf{I}_d), \tag{6}$$

is also a Gaussian distribution. More importantly, we have $Q(Z_1|Z_0) = \mathcal{N}(0, \mathbf{I}_d)$ for any fixed Z_0, which means the forward SDE in Eq. (3) equipped with $b(t)$ and $\sigma^2(t)$ in Eq. (4) can transport any distribution of Z_0 to the standard normal distribution within the time interval $[0, 1]$. The corresponding reverse-time SDE is defined by

$$dZ_t = \left[b(t)Z_t - \sigma^2(t)S(Z_t, t)\right]dt + \sigma(t)d\overleftarrow{W}_t \text{ with } Z_0 = X \text{ and } Z_1 = Y, \tag{7}$$

where $\overleftarrow{W}_t$ is the backward Brownian motion and $S(Z_t, t)$ is the score function. If the score function is defined by

$$S(Z_t, t) := \nabla_z \log Q(Z_t), \tag{8}$$

where $Q(Z_t)$ is the PDF of Z_t in Eq. (3), then the reverse-time SDE maps the standard Gaussian random variable Y to the target random variable X. Therefore, if we can evaluate the score function for any (Z_t, t), then we can directly generate samples of X by pushing samples of Y through the reverse-time SDE. Thus, the central task in training diffusion models is appropriately approximating the score function. There are several established approaches, including score matching (Hyvärinen, 2005), denoising score matching (Vincent, 2011), and sliced score matching (Song et al., 2019), etc. Recent advances in diffusion models focus on explorations of using neural networks to approximate the score function. Compared to the normalizing flow models, a notable drawback of learning the score function is its inefficiency in sampling since generating one sample of X requires solving the reverse-time SDE. To alleviate this challenge, we use the score-based diffusion model as a data labeling approach, which will enable supervised learning of the generative model of interest.

3.2 Training-Free Score Estimation

We now derive the analytical formula of the score function and its approximation scheme. Substituting $Q(Z_t) = \int_{\mathbb{R}^d} Q(Z_t, Z_0)dZ_0 = \int_{\mathbb{R}^d} Q(Z_t|Z_0)Q(Z_0)dZ_0$ into Eq. (8) and exploiting the fact in Eq. (6), we can rewrite the score function as

$$\begin{aligned} S(Z_t, t) &= \nabla_z \log\left(\int_{\mathbb{R}^d} Q(Z_t|Z_0)Q(Z_0)dZ_0\right) \\ &= \frac{1}{\int_{\mathbb{R}^d} Q(Z_t|Z_0')Q(Z_0')dZ_0'} \int_{\mathbb{R}^d} -\frac{Z_t - \alpha_t Z_0}{\beta_t^2} Q(Z_t|Z_0)Q(Z_0)dZ_0 \\ &= \int_{\mathbb{R}^d} -\frac{Z_t - \alpha_t Z_0}{\beta_t^2} w_t(Z_t, Z_0)Q(Z_0)dZ_0, \end{aligned} \tag{9}$$

where the weight function $w_t(Z_t, Z_0)$ is defined by

$$w_t(Z_t, Z_0) := \frac{Q(Z_t|Z_0)}{\int_{\mathbb{R}^d} Q(Z_t|Z_0')Q(Z_0')dZ_0'}, \tag{10}$$

satisfying that $\int_{\mathbb{R}^d} w_t(Z_t, Z_0)Q(Z_0)dZ_0 = 1$.

The integrals/expectations in Eq. (9) can be approximated by Monte Carlo estimators using the available samples in $\mathcal{X} = \{x_1, x_2, \ldots, x_J\} \subset \mathbb{R}^d$. According to the definition of the reverse-time SDE in Eq. (7), the samples in $\mathcal{X}$ are also samples from $Q(Z_0)$. Thus, the integral in Eq. (9) can be estimated by

$$S(Z_t, t) \approx \bar{S}(Z_t, t) := \sum_{n=1}^{N} -\frac{Z_t - \alpha_t x_{j_n}}{\beta_t^2} \bar{w}_t(Z_t, x_{j_n}), \tag{11}$$

using a mini-batch of the dataset $\mathcal{X}$ with batch size $N \leq J$, denoted by $\{x_{j_n}\}_{n=1}^N$, where the weight $w_t(Z_t, x_{j_n})$ is calculated by

$$w_t(Z_t, x_{j_n}) \approx \bar{w}_t(Z_t, x_{j_n}) := \frac{Q(Z_t|x_{j_n})}{\sum_{m=1}^{N} Q(Z_t|x_{j_m})}, \tag{12}$$

and $Q(Z_t|x_{j_n})$ is the Gaussian distribution given in Eq. (6). This means $w_t(Z_t, Z_0)$ can be estimated by the normalized probability density values $\{Q(Z_t|x_{j_n})\}_{n=1}^N$. In practice, the size of the mini-batch $\{x_{j_n}\}_{n=1}^N$ can be flexibly adjusted to balance the efficiency and accuracy.

3.3 Supervised Learning of the Generative Model

In this subsection, we describe how to use the score approximation scheme given in Section 3.2 to generate labeled data and use such data to train the generative model of interest. Due to the stochastic nature of the reverse-time SDE in Eq. (7), the relationship between the initial state Z_0 and the terminal state Z_1 is not deterministic or smooth, as shown in Fig. 1. Thus, we cannot directly use Eq. (7) to generate labeled data. Instead, we use the corresponding ordinary differential equation (ODE), defined by

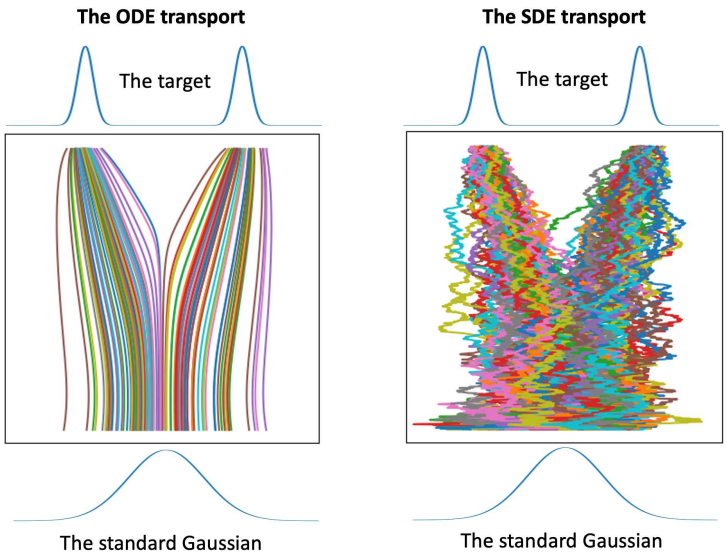

FIG. 1: Illustration of the trajectories of the ODE model in Eq. (13) and the SDE in Eq. (7) using a simple one-dimensional example. We observe that the ODE model creates a much smoother function relationship between the initial state and the terminal state, which indicates that the ODE model is more suitable for generating the labeled data for supervised learning of the generative model of interest.

$$dZ_t = \left[b(t)Z_t - \frac{1}{2}\sigma^2(t)S(Z_t,t) \right] dt \quad \text{with} \quad Z_0 = X \text{ and } Z_1 = Y, \tag{13}$$

whose trajectories share the same marginal PDFs as the reverse-time SDE in Eq. (7). An illustration of the trajectories of the SDE and the ODE is given in Fig. 1. We observe that ODE defines a much smoother function relationship between the initial state and the terminal state than that defined by the SDE. Thus, we use the ODE in Eq. (13) to generate labeled data.

Specifically, we first draw M random samples of Y, denoted by $\mathcal{Y} = \{y_1, \ldots, y_M\}$ from the standard Gaussian distribution. For $m = 1, \ldots, M$, we solve the ODE in Eq. (13) from $t = 1$ to $t = 0$ and collect the state $Z_0|y_m$, where the score function is computed using Eqs. (11) and (12), and the dataset $\mathcal{X} = \{x_1, \ldots, x_J\}$. The labeled training dataset is denoted by

$$\mathcal{D}_{\text{train}} := \{(y_m, x_m) : x_m = Z_0|y_m, \quad \text{for} \quad m = 1, \ldots, M\}, \tag{14}$$

where x_m is obtained by solving the ODE in Eq. (13). We remark that the $\{x_m\}_{m=1}^M$ in the generated labeled training set $\mathcal{D}_{\text{train}}$ is not a subset of the original training set $\mathcal{X}$, but $\{x_m\}_{m=1}^M$ follows the same distribution as the target random variable X when both the number of time steps for solving the reverse-time SDE and the number of training samples in $\mathcal{X}$ go to infinity. An illustration comparing $\mathcal{X}$ and $\mathcal{D}_{\text{train}}$ is given in Fig. 2. Moreover, the number of samples in $\mathcal{D}_{\text{train}}$ can be much larger than the size of $\mathcal{X}$, which could help improve the stability and alleviate overfitting in training the final generator $F(\cdot;\theta)$. After obtaining $\mathcal{D}_{\text{train}}$ we can use it to train the generative model $F(y;\theta)$ in Eq. (2) using supervised learning with the MSE loss.

Our method is summarized in Algorithm 1. Compared to the existing normalizing flow models and diffusion models, our method has two significant advantages in performing density estimation tasks. First, it does not require one to know $F^{-1}(x;\theta)$; hence it does not require the computation of $|\det \mathbf{D}(F^{-1}(x))|$ in the training process. This enables us to use simpler neural

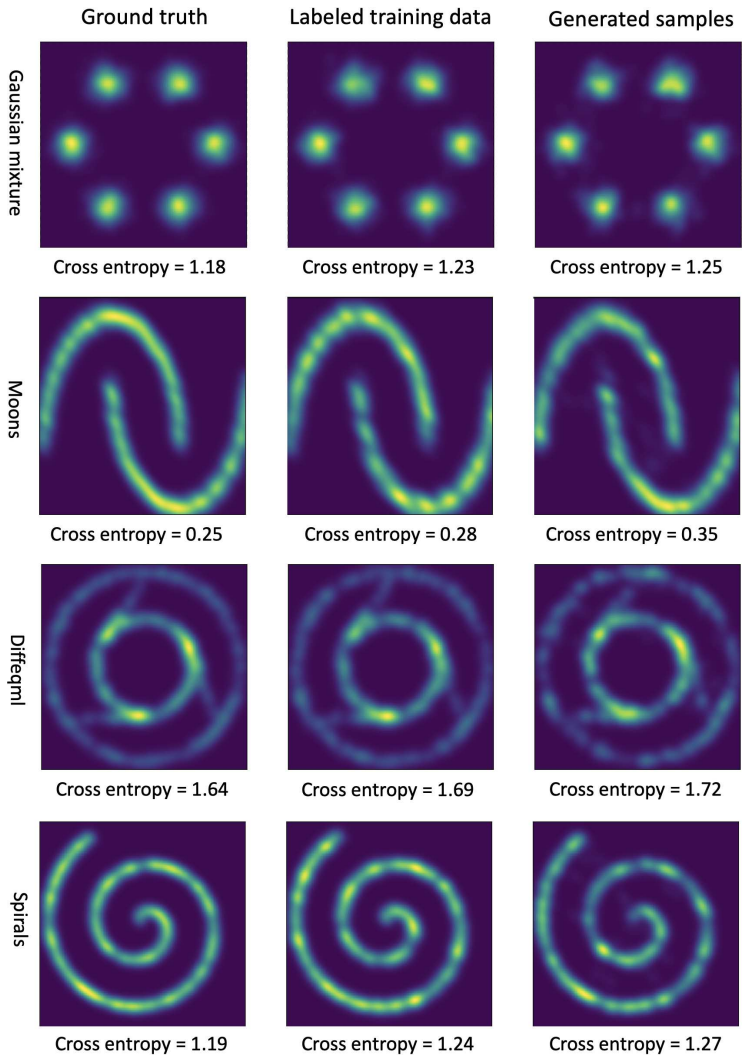

FIG. 2: Results on four 2D toy datasets. The left column shows the ground-truth distribution, i.e., the dataset $\mathcal{X}$; the middle column shows the generated labeled data by solving the ODE in Eq. (13); and the right column shows the samples generated by the trained generative model $F(y;\theta)$. The cross entropy is computed using a density function obtained by kernel density estimation with another 2000 samples.

network architectures to define $F(y;\theta)$, resulting in a more straightforward training procedure compared to the training of a normalizing flow model. Second, after $F(y;\theta)$ is trained, our method does not require solving the reverse-time SDE or ODE to generate samples of X. As a result, it significantly enhances the sampling efficiency in comparison with the diffusion model.

4. NUMERICAL EXPERIMENTS

We demonstrate the performance of the proposed method on several benchmark problems for density estimation. To solve the reverse-time ODE for generating the labeled data, we use the

Algorithm 1: Supervised learning of generative models

1: **Input**: the observation data $\mathcal{X}$, the diffusion coefficient $\sigma(t)$, the drift coefficient $b(t)$;
2: **Output**: the trained generative model $F(y;\theta)$;
3: Draw M samples $\mathcal{Y} = \{y_1, \ldots, y_M\}$ from the standard Gaussian distribution;
4: **for** $m = 1, \ldots, M$
5: Solve the ODE in Eq. (13) with the score function estimated by Eqs. (11) and (12);
6: Define a pair of labeled data (y_m, x_m) where $y_m = Z_1$ and $x_m = Z_0$ in Eq. (13);
7: **end**
8: Train the generative model $x = F(y;\theta)$ with the MSE loss.

explicit Euler scheme. The generative model $F(y;\theta)$ is defined by a fully-connected feed-forward neural network. Our method is implemented in Pytorch with GPU acceleration enabled. The source code is publicly available at https://github.com/mlmathphy/supervised_generative_model. The numerical results in this section can be precisely reproduced using the code on Github.

4.1 Density Estimation on Toy 2D Data

We use four two-dimensional datasets (Grathwohl et al., 2018) to demonstrate and visualize the performance of the proposed method. Each dataset has 1000 data, referred to as the ground truth in Fig. 2. We use a fully-connected neural network with four hidden layers to define $F(y;\theta)$, each of which has 100 neurons. One hundred times steps are used to discretize the reverse-time ODE in Eq. (13) to generate 1000 labeled data. The neural network is then trained with the MSE loss for 5000 epochs using the Adam optimizer with the learning rate chosen as 0.005.

The results are shown in Fig. 2. We observe that the labeled samples are not the same as the ground truth, but they accurately approximate the distribution of the ground truth. In fact, the reverse-time ODE in Eq. (13) can be viewed as a training-free version of the neural-ODE-based normalizing flow (Grathwohl et al., 2018), which can capture multimodal and discontinuous distributions. The samples generated by $F(y;\theta)$ also provide an accurate approximation to the ground truth, even though the accuracy is lower than the distribution of the labeled training data. There are scattered samples generated by $F(y;\theta)$ because the used neural networks cannot perfectly approximate the discontinuity among different modes of the target distribution.

4.2 Density Estimation on Real Data

We demonstrate the performance of our method using density estimation on four UCI datasets. The UCI Machine Learning Repository is a well-known resource for machine learning practitioners and researchers. It provides a collection of databases, domain theories, and data generators that are used by the machine learning community for empirical analysis of machine learning algorithms. In this work, we use four UCI datasets (that are commonly used to test density estimation algorithms) including POWER (six-dimensional, available at http://archive.ics.uci.edu/ml/datasets/Individual+household+electric+power+consumption), GAS (eight-dimensional, available at http://archive.ics.uci.edu/ml/datasets/Gas+sensor+array+under+dynamic+gas+mixtures), HEPMASS (11-dimensional, available at http://archive.ics.uci.edu/ml/datasets/

HEPMASS), and MINIBOONE (43-dimensional, available at http://archive.ics.uci.edu/ml/datasets/MiniBooNE+particle+identification). These datasets are taken from the UCI Machine Learning Repository. We follow the approach in Papamakarios et al. (2017) to preprocess each dataset. We normalize each dimension of the data by subtracting its mean and dividing its standard deviation, such that all the datasets used to test our method have zero mean and unit standard deviation. Discrete-valued dimensions and every attribute with a Pearson correlation coefficient greater than 0.98 were eliminated.

The reverse-time ODE in Eq. (13) is solved using the explicit Euler scheme with 500 time steps. All training data split into mini-batches of size $N = 5000$. We use a fully-connected neural network with one hidden layer to define $F(y; \theta)$. Specifically, the network for POWER has 256 hidden neurons, the network for GAS has 512 hidden neurons, the network for HEPMASS has 1024 neurons, and the network for MINIBOONE has 1400 hidden neurons. The number of hidden neurons is determined by a simple grid search using the MSE on a validation set (10% of the training set). The neural networks are trained using the Adam optimizer with 20,000 epochs with the learning rate being 0.01.

Figures 3–6 show the comparison among the ground truth data, the labeled data [from the ODE in Eq. (13)] and the generated samples from $F(y; \theta)$ using the following metrics:

- The 1D marginal PDF of a randomly selected dimension;
- The K-L divergences of all the 1D marginal distributions;
- The mean values of all the 1D marginal distributions;
- The standard deviations of all the 1D marginal distributions.

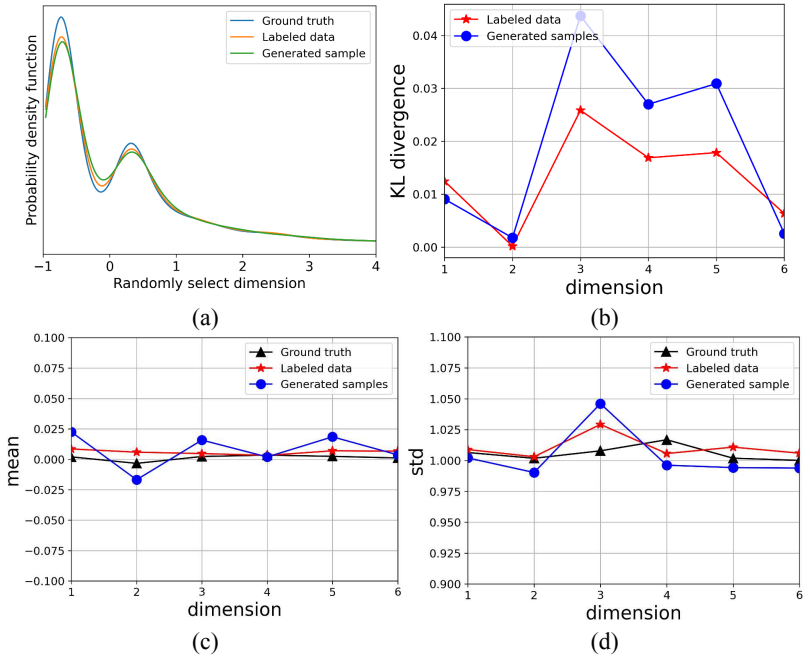

FIG. 3: Results on the POWER dataset. (a) The 1D marginal PDF of a randomly selected dimension; (b) the K-L divergences of all the 1D marginal distributions; (c) the mean values of all the 1D marginal distributions; (d) the standard deviations of all the 1D marginal distributions.

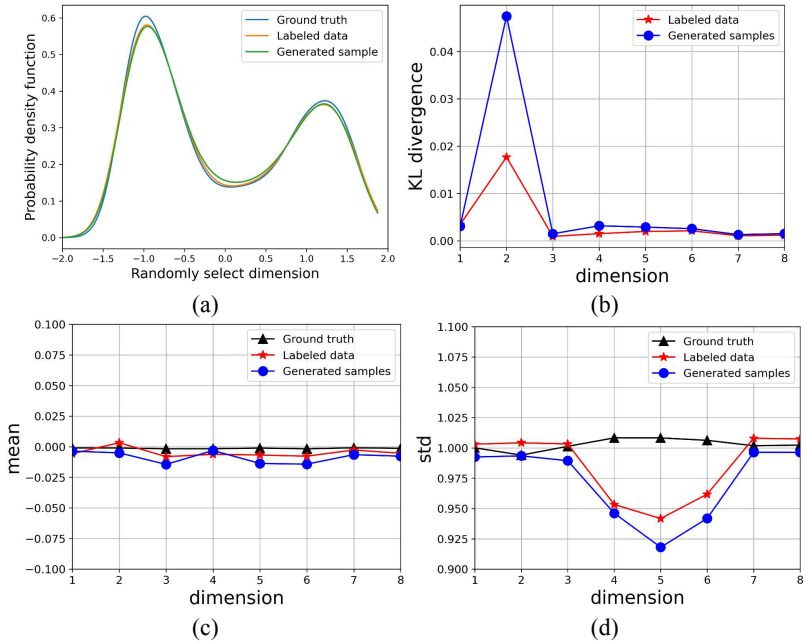

FIG. 4: Results on the GAS dataset. (a) The 1D marginal PDF of a randomly selected dimension; (b) the K-L divergences of all the 1D marginal distributions; (c) the mean values of all the 1D marginal distributions; (d) the standard deviations of all the 1D marginal distributions.

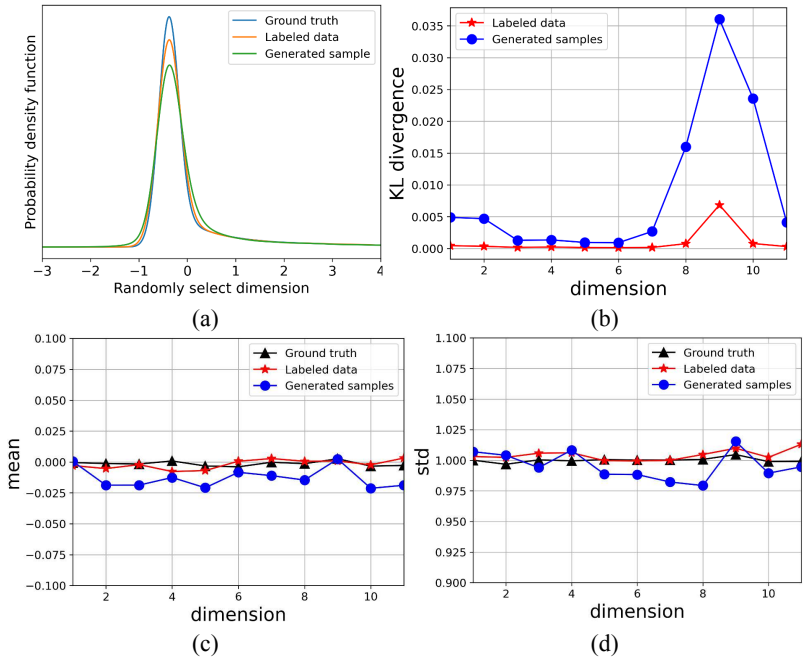

FIG. 5: Results on the HEPMASS dataset. (a) The 1D marginal PDF of a randomly selected dimension; (b) the K-L divergences of all the 1D marginal distributions; (c) the mean values of all the 1D marginal distributions; (d) the standard deviations of all the 1D marginal distributions.

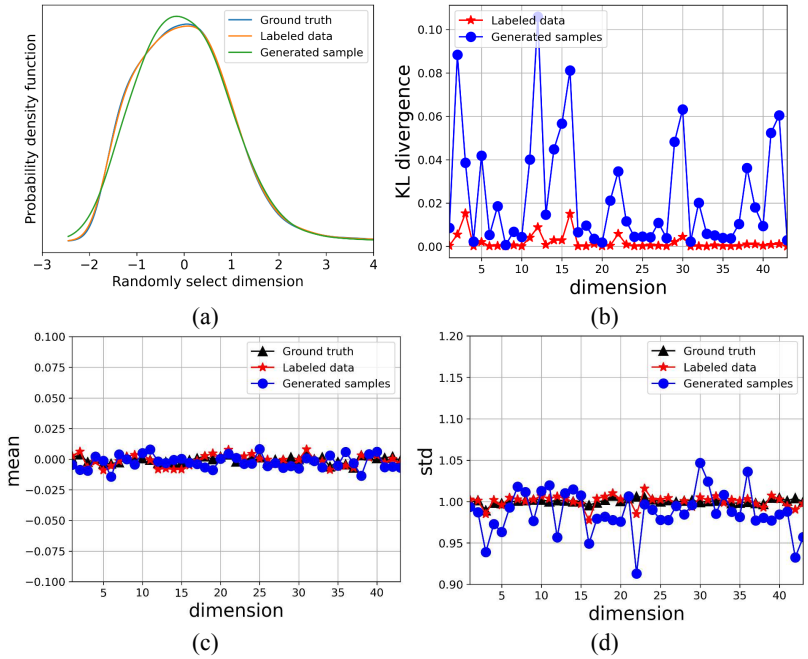

FIG. 6: Results on the MINIBOONE dataset. (a) The 1D marginal PDF of a randomly selected dimension; (b) the K-L divergences of all the 1D marginal distributions; (c) the mean values of all the 1D marginal distributions; (d) the standard deviations of all the 1D marginal distributions.

As expected, the labeled data and the generated samples can accurately approximate the distribution of the ground truth. Table 1 shows the computing costs of generating the labeled data, training the neural network $F(y;\theta)$, and using $F(y;\theta)$ to generate samples. The computational time is obtained by running our code on a workstation with Nvidia RTX A5000 GPU.

5. CONCLUSION

We introduced a supervised learning framework for training generative models for density estimation. Within this framework, we utilize the score-based diffusion model to generate labeled

TABLE 1: Wall-clock time of different stages of our method. Data labeling refers to the stage of solving the reverse-time ODE in Eq. (13); training F refers to the stage of using the labeled data to train the generative model $F(y;\theta)$; synthesizing 100K samples refers to using the trained model $F(y;\theta)$ to generate 100K new samples of X. We observe that our method features a promising efficiency in generating a large number of samples

Dataset	Data Labeling	Training F	Synthesizing 100K Samples
POWER	64.83 sec	182.51 sec	0.10 sec
GAS	85.78 sec	426.61 sec	0.23 sec
HEPMASS	109.12 sec	1940.87 sec	0.51 sec
MINIBOONE	408.36 sec	2220.79 sec	0.66 sec

data and employ simple, fully-connected neural networks to learn the generative model of interest. The key ingredient is the training-free score estimation that enables data labeling without training the score function. It is important to note that the current algorithm has only been successfully tested using a tabular dataset, and its performance in image/signal synthesis remains to be explored. Due to the slow convergence of the Monte Carlo estimation of the score function, the accuracy of the score estimation will become increasingly difficult as the dimension increases. On the other hand, this algorithm can be applied to a variety of Bayesian sampling problems in scientific and engineering applications, including parameter estimation of physical models, state estimation of dynamical systems (e.g., chemical reactions), and surrogate models for particle simulation in physics, all of which will be our future work on this topic.

ACKNOWLEDGMENTS

This material is based upon work supported by the U.S. Department of Energy (DOE), Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics program under the contract ERKJ387 at the Oak Ridge National Laboratory, which is operated by UT-Battelle, LLC, for the DOE under Contract No. DE-AC05-00OR22725. The author Feng Bao would also like to acknowledge the support from the U.S. National Science Foundation through project No. DMS-2142672 and the support from the DOE, Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics program under Grant No. DE-SC0022297, and the author Yanzhao Cao would also like to acknowledge the support from the DOE, Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics program under Grant No. DE-SC0022253.

This manuscript is authored by UT-Battelle, LLC, under contract No. DE-AC05-00OR22725 with the DOE. The U.S. government retains and the publisher, by accepting the article for publication, acknowledges that the US government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for U.S. government purposes. DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan.

REFERENCES

Austin, J., Johnson, D.D., Ho, J., Tarlow, D., and van den Berg, R., Structured Denoising Diffusion Models in Discrete State-Spaces, in *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021*, Virtual, pp. 17981–17993, December 6–14, 2021.

Bao, F., Zhang, Z., and Zhang, G., A Score-Based Nonlinear Filter for Data Assimilation, arXiv: 2306.09282, 2023.

Cai, R., Yang, G., Averbuch-Elor, H., Hao, Z., Belongie, S.J., Snavely, N., and Hariharan, B., Learning Gradient Fields for Shape Generation, *Computer Vision – ECCV 2020 – 16th European Conf.*, Glasgow, UK, August 23–28, 2020.

Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., and Bharath, A.A., Generative Adversarial Networks: An Overview, *IEEE Signal Process. Mag.*, vol. **35**, no. 1, pp. 53–65, 2018.

Dinh, L., Sohl-Dickstein, J., and Bengio, S., Density Estimation Using Real NVP, arXiv:1605.08803, 2016.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y., Generative Adversarial Nets, in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds., Vol. 27, Red Hook, NY: Curran Associates, Inc., 2014.

Grathwohl, W., Chen, R.T., Bettencourt, J., Sutskever, I., and Duvenaud, D., FFJORD: Free-Fform Continuous Dynamics for Scalable Reversible Generative Models, arXiv:1810.01367, 2018.

Guo, L., Wu, H., and Zhou, T., Normalizing Field Flows: Solving Forward and Inverse Stochastic Differential Equations Using Physics-Informed Flow Models, *J. Comput. Phys.*, vol. **461**, p. 111202, 2022.

Ho, J., Jain, A., and Abbeel, P., Denoising Diffusion Probabilistic Models, in *Advances in Neural Information Processing Systems*, Vol. 33, Red Hook, NY: Curran Associates, Inc., pp. 6840–6851, 2020.

Ho, J., Saharia, C., Chan, W., Fleet, D.J., Norouzi, M., and Salimans, T., Cascaded Diffusion Models for High Fidelity Image Generation, *J. Mach. Learn. Res.*, vol. **23**, pp. 1–33, 2022.

Hoogeboom, E., Nielsen, D., Jaini, P., Forré, P., and Welling, M., Argmax Flows and Multinomial Diffusion: Learning Categorical Distributions, in *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021*, Virtual, pp. 12454–12465, December 6–14, 2021.

Hyvärinen, A., Estimation of Non-Normalized Statistical Models by Score Matching, *J. Mach. Learn. Res.*, vol. **6**, no. 24, pp. 695–709, 2005.

Kingma, D.P. and Welling, M., Auto-Encoding Variational Bayes, in *2nd Int. Conf. on Learning Representations, ICLR 2014*, Banff, AB, Canada, April 14–16, 2014.

Kobyzev, I., Prince, S.J., and Brubaker, M.A., Normalizing Flows: An Introduction and Review of Current Methods, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. **43**, no. 11, pp. 3964–3979, 2020.

Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A.P., Tejani, A., Totz, J., Wang, Z., and Shi, W., Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network, in *2017 IEEE Conf. on Computer Vision and Pattern Recognition, CVPR 2017*, Honolulu, HI, USA, pp. 105–114, July 21–26, 2017.

Li, X.L., Thickstun, J., Gulrajani, I., Liang, P., and Hashimoto, T.B., Diffusion-LM Improves Controllable Text Generation, arXiv:2205.14217, 2022.

Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., and Zhu, J., DPM-Solver: A Fast ODE Solver for Diffusion Probabilistic Model Sampling in Around 10 Steps, in *Advances in Neural Information Processing Systems*, A.H. Oh, A. Agarwal, D. Belgrave, and K. Cho, Eds., Red Hook, NY: Curran Associates, Inc., 2022.

Luo, S. and Hu, W., Score-Based Point Cloud Denoising, in *2021 IEEE/CVF Int. Conf. on Computer Vision, ICCV 2021*, Montreal, QC, Canada, pp. 4563–4572, October 10–17, 2021.

Ma, X., Zhou, C., Li, X., Neubig, G., and Hovy, E.H., FlowSeq: Non-Autoregressive Conditional Sequence Generation with Generative Flow, in *Proc. of the 2019 Conf. on Empirical Methods in Natural Language Processing and the 9th Int. Joint Conf. on Natural Language Processing, EMNLP-IJCNLP 2019*, Hong Kong, China, pp. 4281–4291, November 3–7, 2019.

Meng, C., He, Y., Song, Y., Song, J., Wu, J., Zhu, J., and Ermon, S., SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations, in *The Tenth Int. Conf. on Learning Representations, ICLR 2022*, Virtual, April 25–29, 2022.

Papamakarios, G., Pavlakou, T., and Murray, I., Masked Autoregressive Flow for Density Estimation, *Adv. Neural Inf. Process. Syst.*, vol. **30**, pp. 1–10, 2017.

Rezende, D. and Mohamed, S., Variational Inference with Normalizing Flows, in *Int. Conf. on Machine Learning*, Lille, France, pp. 1530–1538, June 6–11, 2015.

Salimans, T., Goodfellow, I.J., Zaremba, W., Cheung, V., Radford, A., and Chen, X., Improved Techniques for Training GANs, in *Advances in Neural Information Processing Systems 29: Annual Conf. on Neural Information Processing Systems 2016*, Barcelona, Spain, pp. 2226–2234, December 5–10, 2016.

Savinov, N., Chung, J., Binkowski, M., Elsen, E., and van den Oord, A., Step-Unrolled Denoising Autoencoders for Text Generation, in *The Tenth Int. Conf. on Learning Representations, ICLR 2022*, Virtual, April 25–29, 2022.

Schlegl, T., Seeböck, P., Waldstein, S.M., Schmidt-Erfurth, U., and Langs, G., Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery, in *Information Processing in Medical Imaging – 25th Int. Conf., IPMI 2017*, Boone, NC, USA, pp. 146–157, June 25–30, 2017.

Shi, Y., De Bortoli, V., Deligiannidis, G., and Doucet, A., Conditional Simulation Using Diffusion Schrödinger Bridges, in *Proc. of the Thirty-Eighth Conf. on Uncertainty in Artificial Intelligence*, Eindhoven, Netherlands, pp. 1792–1802, August 1–5, 2022.

Sohl-Dickstein, J., Weiss, E.A., Maheswaranathan, N., and Ganguli, S., Deep Unsupervised Learning Using Nonequilibrium Thermodynamics in *JMLR Workshop Conf. Proc.*, vol. **37**, pp. 2256–2265, 2015.

Song, Y. and Ermon, S., Generative Modeling by Estimating Gradients of the Data Distribution, *Adv. Neural Inf. Process. Syst.*, vol. **32**, 2019.

Song, Y., Garg, S., Shi, J., and Ermon, S., Sliced Score Matching: A Scalable Approach to Density and Score Estimation, in *Proc. of the 35th Uncertainty in Artificial Intelligence Conf.*, Tel-Aviv, Israel, pp. 574–584, July 22–25, 2019.

Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., and Poole, B., Score-Based Generative Modeling through Stochastic Differential Equations, in *Int. Conf. on Learning Representations*, Vienna, Austria, May 3–7, 2021.

Vincent, P., A Connection between Score Matching and Denoising Autoencoders, *Neural Comput.*, vol. **23**, no. 7, pp. 1661–1674, 2011.

Yang, L., Zhang, Z., Song, Y., Hong, S., Xu, R., Zhao, Y., Zhang, W., Cui, B., and Yang, M.H., Diffusion Models: A Comprehensive Survey of Methods and Applications, *ACM Comput. Surv.*, vol. **56**, no. 4, p. 105, 2023.

Yu, P., Xie, S., Ma, X., Jia, B., Pang, B., Gao, R., Zhu, Y., Zhu, S., and Wu, Y.N., Latent Diffusion Energy-Based Model for Interpretable Text Modelling, of *Proc. Mach. Learn. Res.*, vol. **162**, pp. 25702–25720, 2022.