Supervised Score-Based Modeling by Gradient Boosting

Changyuan Zhao[1,2], **Hongyang Du**[3*], **Guangyuan Liu**[1,4], **Dusit Niyato**[1]

[1]College of Computing and Data Science, Nanyang Technological University
[2]CNRS@CREATE, 1 Create Way, # 08-01 Create Tower, Singapore 138602
[3]Department of Electrical and Electronic Engineering, University of Hong Kong
[4]The Energy Research Institute @ NTU, Interdisciplinary Graduate Program
zhao0441@e.ntu.edu.sg, duhy@eee.hku.hk, liug0022@e.ntu.edu.sg, dniyato@ntu.edu.sg

Abstract

Score-based generative models can effectively learn the distribution of data by estimating the gradient of the distribution. Due to the multi-step denoising characteristic, researchers have recently considered combining score-based generative models with the gradient boosting algorithm, a multi-step supervised learning algorithm, to solve supervised learning tasks. However, existing generative model algorithms are often limited by the stochastic nature of the models and the long inference time, impacting prediction performances. Therefore, we propose a Supervised Score-based Model (SSM), which can be viewed as a gradient boosting algorithm combining score matching. We provide a theoretical analysis of learning and sampling for SSM to balance inference time and prediction accuracy. Via the ablation experiment in selected examples, we demonstrate the outstanding performances of the proposed techniques. Additionally, we compare our model with other probabilistic models, including Natural Gradient Boosting (NGboost), Classification and Regression Diffusion Models (CARD), Diffusion Boosted Trees (DBT), and non-probabilistic gradient boosting models. The experimental results show that our model outperforms existing models in both accuracy and inference time.

Introduction

Generative artificial intelligence (GAI) models aim to create highly realistic simulations or reproductions of data by learning from the characteristics of the original data. Recently, score-based models (diffusion models) have become the most powerful families of GAI due to their excellent generation performance (Yang et al. 2023; Dhariwal and Nichol 2021). In these models, a series of noises is first added to the original data through a forward diffusion process. The model then learns the added noise and continuously denoises it to generate new data that follows the distribution of the original dataset. At present, many researchers have applied score-based models to generation tasks in many fields, including image generation, audio generation, video generation, etc., all with excellent performance (Rombach et al. 2021; Kong et al. 2020; Ho et al. 2022).

Supervised learning is another important research area in machine learning different from generative learning. In supervised learning, a model is trained on labeled data with the goal of learning a mapping from inputs to outputs, allowing it to predict the output for unseen data accurately (Cunningham, Cord, and Delany 2008). Regression and classification problems are two basic types of supervised learning tasks, which aim to train a model for prediction. For regression problems, this prediction result is the target output corresponding to the input data, while for classification problems, it is the accurate label. The learning performance of this data relationship is closely related to the learning ability of the sophisticated structure of neural network models (He et al. 2016). Particularly, the Gradient Boosting Machine (GBM) is a supervised learning model that combines several weak learners into a strong learner (Bentéjac, Csörgő, and Martínez-Muñoz 2021). The sequential weak learners reduce errors from previous iterations to obtain accurate predictions, especially with large and complex datasets. Recently, GBM algorithms such as XGBoost (Chen and Guestrin 2016) and LightGBM (Ke et al. 2017) have demonstrated state-of-the-art performance in tabular competitions on platforms, including Kaggle.

Due to the multi-step denoising characteristic, researchers have recently considered combining score-based generative models with the gradient boosting algorithm to solve supervised learning tasks (Han and Zhou 2024; Beltran-Velez et al. 2024). However, these previous methods are mainly based on diffusion models, whose random denoising process may lead to unstable prediction results and reduce the robustness. Moreover, diffusion models often require longer denoising steps, increasing model inference time. If accelerated algorithms such as DDIM (Song, Meng, and Ermon 2020) are used, the quality of the generated results will usually decrease, resulting in lower prediction accuracy.

Therefore, in this paper, our goal is to propose a score-based gradient boosting model for supervised learning to enable quick and accurate inference. To achieve this goal, we consider the score-based generative models via denoising score matching (Song and Ermon 2019) since its inference process does not have a specific number of denoising steps. Based on this, we propose **Supervised Score-based Model (SSM)**, a score-based model by gradient boosting.

We summarize our main contributions as follows: 1) We utilize noise-free Langevin Dynamics to establish a connection between score-based generative models and GBM al-

*Corresponding Authors

gorithms. 2) we present SSM to predict the maximum log-likelihood estimation point, which estimates errors between the target and input data through a score network. 3) For the training process and inference process, we provide a theoretical analysis to balance inference time and prediction accuracy. 4) Experiments on regression and classification tasks show that SSM can achieve better performance than existing methods and significantly shorten the inference time.

Related Work

Score-based generative models have demonstrated powerful learning capabilities for complex data distribution in many fields (Croitoru et al. 2023; Du et al. 2023; Liu et al. 2024). There are generally two categories of such models: diffusion-based and score matching-based models (Yang et al. 2023). Diffusion-based models consist of a forward diffusion process, which adds Gaussian noises into the original data, making it converge to the standard normal distribution, and a denoising process, which removes the added noises from a random variable following normal distribution (Ho, Jain, and Abbeel 2020; Nichol and Dhariwal 2021). The score matching-based model can be regarded as an energy-based model (Song and Kingma 2021), which aims to learn explicit probability distributions of data by a score function. Denoising score matching is a state-of-the-art approach for score estimation, which also estimates the score function from the disturbed dataset and then denoises through Langevin dynamics (Song and Ermon 2019). From a general perspective, these two types of models can also be viewed as the discretization of stochastic differential equations (Song et al. 2020b).

The GBM is a powerful ensemble learning technique that builds models sequentially, with each new model aiming to correct the errors of its predecessors (Friedman 2001). By combining the predictions of multiple weak learners, GBM creates a strong predictive model that can handle complex data patterns and interactions (Natekin and Knoll 2013). XGBoost, LightGBM, and NGBoost are typical decision trees-based gradient boosting techniques that extend the foundational principles of GBM to enhance efficiency and performance. XGBoost and LightGBM focus on computational speed and scalability (Chen and Guestrin 2016; Ke et al. 2017), while NGBoost introduces probabilistic predictions, offering a novel approach to uncertainty estimation in gradient boosting models (Duan et al. 2020).

Recently, via setting input data in supervised learning as conditions, conditional score-based models have been utilized to solve supervised learning problems (Amit et al. 2021; Rahman et al. 2023; Beltran-Velez et al. 2024). (Zimmermann et al. 2021) proposed a score-based generative classifier for classification tasks. This model predicts classification results through maximum likelihood estimation of different label generation distributions and target values instead of direct prediction. In (Han, Zheng, and Zhou 2022), the authors proposed CARD, a score-based model for classification and regression. They emphasized its ability to solve uncertainty in supervised learning and compared its performance with existing Bayesian neural networks (Blundell et al. 2015; Tomczak et al. 2021). The DBT model, a diffusion boosting paradigm, connects the denoising diffusion generative model and GBM via decision trees. Through experiments on real-world regression tasks, this approach demonstrates the potential of integrating GBM with score-based generative models (Han and Zhou 2024).

Background

Gradient Boosting Machine

The GBM is a supervised learning framework that combines several weak learners into strong learners in an iterative way (Friedman 2001). In GBM, a boosted model $F(x)$ is a weighted linear combination of m weak learners, which can be formulated as

$$F(x) = F_m(x) = F_0(x) + \sum_{k=1}^{m} \alpha_k h_k(x), \tag{1}$$

where $F_0(x)$ is the initial prediction, α_k is the weight coefficient, and $h_k(x)$ represents weak learner. Let $l(y, F(x))$ represent a measure of data error at the observation (y, x) for the differentiable loss function l. To train a GBM, the goal is to explore a function F that minimizes the expected loss $L(F(x)) = E_{(x,y)} l(y, F(x))$ where the expectation is taken over the input dataset of (y, x). Since the loss function l is differentiable, the optimal solution $F^*(x)$ can be represented as

$$F^*(x) = F_0(x) + \sum_{k=1}^{m} \alpha_k \cdot (-g_k(x)), \tag{2}$$

where $g_k(x) = \nabla_{F_{k-1}(x)} L(F_{k-1}(x))$ is the gradient at optimization step k. Therefore, weak learners are trained to estimate the negative gradient term to approximate the optimal solution. Given a trained GBM and input, the predicted result can be obtained via the iterative equation Eq. (1).

Langevin Dynamics

Langevin dynamics is a mathematical model that describes the dynamics of molecular systems. For a continuously differentiable probability distribution $p(x)$ where $x \in \mathbb{R}^d$, Langevin dynamics uses the score function, i.e., log-likelihood estimate $\nabla_x \log p(x)$, iteratively to obtain samples that fit the distribution $p(x)$ (Welling and Teh 2011). Given a certain step size $\epsilon > 0$ and any prior distribution $\pi(x)$, the Langevin dynamics can be expressed as

$$x_t = x_{t-1} + \frac{\epsilon}{2} \nabla_x \log p(x_{t-1}) + \sqrt{\epsilon} z_t, \tag{3}$$

where $t \geq 0$, $x_0 \sim \pi(x)$, and $z_t \sim \mathcal{N}(0, 1)$. As a Markov chain Monte Carlo (MCMC) technique (Robert, Casella, and Casella 1999), the Langevin equation takes gradient steps based on the score function and also injects Gaussian noise to capture the distribution $p(x)$ instead of the maximum point of the log-likelihood estimate $\log p(x)$. When $\epsilon \to 0$ and $t \to \infty$, the distribution of x_t will converge to the target distribution $p(x)$ under some regularity conditions (Roberts and Tweedie 1996).

Denoising Score Matching for Score Estimation

Since the Langevin equation in Eq. (3) only relies on the score function $\log p(x)$, estimating the score of the target distribution is a crucial step. Recent studies (Hyvärinen and Dayan 2005; Song et al. 2020a; Vincent 2011) have explored the use of the score network $s_\theta : \mathbb{R}^d \rightarrow \mathbb{R}^d$, a neural network parameterized by θ, to estimate the score function.

Denoising Score Matching Denoising score matching (Vincent 2011) is a method to estimate the score function by adding noise to the original data and then estimating the perturbed data distribution. Specifically, perturbing the data with a specified noise can represent the perturbed distribution for point x as $q(\widetilde{x}|x)$. Then, the entire perturbed distribution is $q(\widetilde{x}) = \int q(\widetilde{x}|x)p(x)dx$. To estimate the perturbed distribution, the loss function can be represented as

$$J'(\theta) = \frac{1}{2}\mathbb{E}_{q(\widetilde{x}|x)p(x)}[\|s_\theta(\widetilde{x}) - \nabla_{\widetilde{x}} \log q(\widetilde{x}|x)\|_2^2]. \quad (4)$$

When the added noise is small enough, e.g., $q(x) \approx p(x)$, the optimal score network $s_{\theta^*}(x)$, where $\theta^* = \arg\min_\theta J'(\theta)$, can represent the score of original data distribution, i.e., $s_{\theta^*}(x) = \nabla_x \log q(x) \approx \nabla_x \log p(x)$ (Vincent 2011). However, denoising score matching may not accurately estimate the score in areas with low data density due to insufficient training data (Song and Ermon 2019). This issue can be solved by adding larger noise, but it will not satisfy the condition that the added noise is sufficiently small, which can lead to another kind of error.

Noise Conditional Score Network Noise Conditional Score Network (NCSN) is a well-designed score network that uses various levels of noise while simultaneously estimating scores using a single neural network (Song and Ermon 2019). In NCSN, the noises are a series of normal distributions whose variances $\{\sigma_i\}_{i=1}^L$ are a positive geometric sequence, i.e., $\frac{\sigma_1}{\sigma_2} = \cdots = \frac{\sigma_{L-1}}{\sigma_L} > 1$. Based on the noise levels $\{\sigma_i\}_{i=1}^L$, the perturbed data distribution is $q_\sigma(\widetilde{x}) = \int p(x)\mathcal{N}(\widetilde{x}|x, \sigma^2 I)dx$. NCSN aims to jointly estimate the scores of all perturbed data distribution, i.e., $\forall \sigma \in \{\sigma_i\}_{i=1}^L : s_\theta(x, \sigma) \approx \nabla_x \log q_\sigma(x)$, where $s_\theta(x, \sigma) \in \mathbb{R}^d$. For a specific noise level σ, the objective in Eq. (4) can be transferred into,

$$l(\theta; \sigma) = \frac{1}{2}\mathbb{E}_{p(x)}\mathbb{E}_{\widetilde{x}\sim\mathcal{N}(x,\sigma^2 I)}[\|s_\theta(\widetilde{x}, \sigma) + \frac{\widetilde{x} - x}{\sigma^2}\|_2^2]. \quad (5)$$

Then for all noise levels $\{\sigma_i\}_{i=1}^L$, the loss is,

$$\mathcal{L}(\theta; \{\sigma_i\}_{i=1}^L) = \frac{1}{L}\Sigma_{i=1}^L \lambda(\sigma_i) l(\theta; \sigma_i), \quad (6)$$

where $\lambda(\sigma_i)$ is a coefficient depending on σ_i. After being trained, NCSN generates samples using annealed Langevin dynamics (Song and Ermon 2019) where the denoising process occurs level by level according to the noise level.

Supervised Score-based Model via Denoising Score Matching and Gradient Boosting

In supervised learning, the training set $\mathcal{D}$ comprises multiple input-target pairs, i.e., $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, where $x_i \in \mathbb{R}^m$ and $y_i \in \mathbb{R}^d$. The model $f : \mathbb{R}^m \rightarrow \mathbb{R}^d$ trained on this training set aims to predict the response variable y given an input variable x, i.e., $f(x) \approx y$. For a regression problem, the response variable y is a continuous variable, whereas it is a categorical variable for classification.

For supervised learning problems, given an input x_i, unlike estimating the distribution of output, we pay more attention to predicting the value of y_i, or a special single-point distribution, i.e. $p(y) = \mathbb{I}_{y_i}(y)$, where $\mathbb{I}(\cdot)$ is an indicator function. It is meaningless to estimate the score function of a single point distribution since its score function is only defined at value y_i. To address this issue, denoising score matching provides a feasible method via perturbing the single point distribution with noises. Specifically, when adding a Gaussian noise with mean 0 and variance σ, the perturbed distribution will become a normal distribution with mean y_i and variance σ, i.e., $q(\widetilde{y}) = \int p(y)\mathcal{N}(\widetilde{y}|y, \sigma^2 I)dy = \mathcal{N}(y_i, \sigma^2)$, whose score function is well defined. Therefore, to estimate the value or the single point distribution of y_i, we aim to train an NCSN to estimate the score function of input-target pairs $(x_i, y_i) \in \mathcal{D}$ which uses an input variable x_i as another condition, i.e., $s_\theta(y, \sigma_j, x_i) \in \mathbb{R}^d$, where $\sigma_j \in \{\sigma_j\}_{j=1}^L$. For a given $(x_i, y_i) \in \mathcal{D}$, $s_\theta(y, \sigma_j, x_i)$ denotes the score function of y_i, which is the score network of our framework SSM.

Accordingly, the objective in our framework is,

$$l(\theta; \sigma, x, y) = \frac{1}{2}\mathbb{E}_{p(y)}\mathbb{E}_{\widetilde{y}\sim\mathcal{N}(y,\sigma^2 I)}[\|s_\theta(\widetilde{y}, \sigma, x) + \frac{\widetilde{y} - y}{\sigma^2}\|_2^2]. \quad (7)$$

Then for all noise levels $\{\sigma_i\}_{i=1}^L$ and dataset $\mathcal{D}$, the loss is,

$$\mathcal{L}(\theta; \{\sigma_i\}_{i=1}^L) = \frac{1}{L}\Sigma_{i=1}^L \lambda(\sigma_i)\mathbb{E}_{(x,y)\in\mathcal{D}} l(\theta; \sigma_i, x, y). \quad (8)$$

Due to the multi-step denoising characteristic of score-based generative models and GBM models, analyzing the relationship between the two is more helpful in deploying score-based generative models in supervised learning tasks. Additionally, the choice of model parameters greatly affects the performance of the score-based generative model and the GBM model (Bentéjac, Csörgő, and Martínez-Muñoz 2021). As discussed in (Song and Ermon 2020), we need to design many parameters to ensure the effectiveness of training and inference, including (i) the choices of noise scales $\{\sigma_i\}_{i=1}^L$; (ii) the step size ϵ in Langevin dynamics; (iii) the inference steps t in Langevin equation. Therefore, in the following, we provide theoretical analysis to ensure the performance of the SSM on regression and classification problems.

Gradient Boosting

Firstly, we consider the connection between SSM and GBM. For a given input, the denoising score match of SSM provides a perturbed distribution in the solution space whose maximum log-likelihood estimation point is the target value. To predict the maximum log-likelihood estimation point, we can set the loss function l in GBM as the negative log-likelihood estimation, i.e., $-log\ q(F(x))$, where $q(\widetilde{x}|x)$ is the perturbed distribution. Thus, the optimal solution in Eq.

(2) will be the maximum log-likelihood estimation point of the perturbed distribution. Particularly, the iterative equation of GBM in Eq. (1) can be converted to a noise-free Langevin equation with the above loss function, i.e.,

$$F_k(x) = F_{k-1}(x) + \beta_k \cdot \nabla log\, q(F_{k-1}(x)). \tag{9}$$

Moreover, we plan to train one conditional score network with different noise levels as the conditions, similar to the uniform score network in the score-based generative model. The different noise level conditions represent various weak learners in GBM. Specifically, given an input x_I and a noise level σ_j, and a trained score network $s_\theta(y, \sigma_j, x_I)$, the converted noise-free Langevin equation is

$$y_t = y_{t-1} + \alpha_j \cdot s_\theta(y_{t-1}, \sigma_j, x_I), \tag{10}$$

where $\alpha_j = \epsilon \cdot \sigma_j^2/\sigma_L^2$ is the step size.

Next, we analyze the convergence results of the noise-free Langevin equation. Assume that we have a well-trained score network that can estimate the gradient accurately. For a prediction pair (x_I, y_I), the noise-free Langevin equation is,

$$y_t = y_{t-1} - r_L \cdot (y_{t-1} - y_I), \tag{11}$$

where $r_L = \epsilon/\sigma_L^2$ is called the refinement rate. Similar to GBM, it iteratively reduces the difference between y_t and the target y_I, and then lets y_t converge to y_I as $t \to \infty$. Simultaneously, the estimation of score network $s_\theta(y, \sigma_j, x_I)$ is the error estimation weighted by $\frac{1}{\sigma_i}$. We call the inference algorithm following Eq. (11) as *error refinement*.

Technique 1. *(Error Refinement) In the inference stage, the denoising process follows the GBM framework and works as the noise-free Langevin equation (Eq.* (10)*) without the term in the original Langevin equation (Eq.* (3)*).*

From this perspective, adding a set of noises aims to fully train the score network to obtain more accurate estimates. In (Song and Ermon 2020), the authors proposed a strategy to guide the selection of noise where the score network can be more accurately estimated. This noise level is a geometric progression based on the "three sigma rule of thumb" (Grafarend 2006), which can ensure that the samples from $p_{\sigma_i}(x)$ will cover high density regions of $p_{\sigma_{i-1}}(x)$. Besides, for the initial noise scale, the authors suggest choosing σ_1 to be as large as the maximum Euclidean distance between all pairs of training data points, i.e., $\sigma_1 \geq \max_{y_i,y_j} \|y_i - y_j\|_2$. Score models based on geometric progression have achieved good performance in a variety of tasks (Yang et al. 2023). Therefore, we also adopt a geometric progression noise level and the technique for choosing the initial noise scale.

Refinement steps

It has been observed that a notorious shortcoming of NCSN and other score-based generative models is their relatively long inference time (Cao et al. 2024). In (Song and Ermon 2019), the refinement steps for different levels of noise are usually the same value. However, according to the *error refinement* equation (Eq. (11)), the final stage is crucial when deploying a hierarchical denoising process. When the last level of noise can be estimated accurately enough, as the denoising step size increases, we can be confident that the predicted value converges around the ground truth. On the contrary, even if the previous levels of noise are estimated accurately, convergence cannot be guaranteed if the last level is not accurate. Hence, the same refinement steps may waste inference time on denoising processes with large levels of noise. Additionally, even though the authors in (Song and Ermon 2020) analyzed how to select a suitable refinement step for different noise levels, it is based on the different inference process from the *error refinement* following Eq. (10). Therefore, we propose our strategy for step setting.

Give an input-target pair $(x_I, y_I) \in \mathcal{D}$, following the design of noise level (Song and Ermon 2020), the samples from $p_{\sigma_i}(y_I)$ will cover high density regions of $p_{\sigma_{i-1}}(y_I)$. Therefore, in the process of training, there will be more training samples from regions with low noise levels. Additionally, for a perturbed data $\widetilde{y}_I$, since the score network $s_\theta(\widetilde{y}_I, \sigma_i, x_I)$ estimates $(\widetilde{y}_I - y_I)/\sigma_i^2$, the estimation based on small noise level is more accurate, when the loss is small enough. Due to the *error refinement* continuously reducing the error between the current prediction and the ground truth, a simple idea is to select noise levels based on the current error. Based on this, when the error is reduced to high density areas of noise level σ_{i+1}, we believe the score estimated by the noise level σ_{i+1} to be more accurate than σ_i. For the current estimated prediction, we have the following proposition,

Proposition 1. *For an input-target pair $(x_I, y_I) \in \mathcal{D}$, let y_0 denote the initial prediction point. After t steps of denoising, the current estimated prediction y_t satisfies,*

$$y_t = (1 - r_L)^t \cdot (y_0 - y_I) + y_I, \tag{12}$$

where $r_L = \epsilon/\sigma_L^2$ is the refinement rate.

When $|1 - r_L| < 1$ and $t \to \infty$, we have $(1 - r_L)^t \cdot (y_0(\sigma_L) - y_I) \to 0$, i.e., $\lim_{t\to\infty} y_t = y_I$. Therefore, to ensure the stability and convergence of the *error refinement*, we need to demand $|1 - r_L| < 1$, i.e., the refinement rate satisfies $0 < r_L < 2$.

Given noise levels $\{\sigma_i\}_{i=1}^L$ and the initial prediction point $y_0(\sigma_i)$ for each noise level σ_i, let the t-th estimated error be $\|y_t - y_I\|_\infty$, i.e., the maximum value of the error in all dimensions. Based on this, we aim to define an end-signal $\beta(\sigma_i)$. When the error estimated by noise level σ_i less than this signal $\beta(\sigma_i)$, we jump into using the next noise level to denoise from the current estimated prediction. We call the jump step as the switch time denoted as t_{σ_i}. Therefore, we have the following,

Proposition 2. *Given a trained score network $s_\theta(y_t, \sigma_i, x_I)$, the end-signal $\beta(\sigma_i) \in \mathbb{R}$ satisfies,*

$$\sigma_i^2 \cdot \|s_\theta(y_{t_{\sigma_i}}, \sigma_i, x_I)\|_\infty < \beta(\sigma_i) \leq \sigma_i^2 \cdot \|s_\theta(y_{t_{\sigma_i}-1}, \sigma_i, x_I)\|_\infty, \tag{13}$$

where $i < L$. Moreover, when $1 < i < L$, the switch time t_{σ_i} satisfies,

$$t_{\sigma_i} \leq \log_{|1-r_L|}(\frac{\beta(\sigma_i)}{\beta(\sigma_{i-1})}). \tag{14}$$

Specially, when $i = 1$, the switch time t_{σ_i} satisfies,

$$t_{\sigma_1} \leq \log_{|1-r_L|}(\frac{\beta(\sigma_1)}{\sigma_1}). \tag{15}$$

According to Proposition 2, via comparing the value between the current output of the score network $s_\theta(y_t, \sigma_i, x_I)$ and end-signal $\beta(\sigma_i)$, we can then determine when to switch to the next level of noise for denoising. Additionally, Proposition 2 provides an upper bound estimation of switch time t_{σ_i}. This upper bound estimation is relevant to refinement rate r_L and end-signal $\beta(\sigma_i)$. Therefore, via selecting a suitable refinement rate and end-signal, we can effectively limit the number of denoising steps on the specified noise level to shorten the time cost.

For example, the authors chose a geometric sequence with $\sigma_1 = 1$ and $\sigma_{10} = 0.01$, denoising step $T = 100$, and $\epsilon = 2 \times 10^{-5}$ for image generation in (Song and Ermon 2019). If we set $\beta(\sigma_i) = \gamma \cdot \sigma_i$, based on Eq. (14), we can approximately compute that $t_{\sigma_i} \approx 3 \ll 100 = T$. Even though the estimation does not consider errors caused by network training, it is still significantly less than the given number of denoising steps which can save a lot of inference time. In summary, we propose that

Technique 2. *(End-signal) By setting an end-signal $\beta(\sigma_i)$, the denoising process ends based on the estimated error rather than a fixed step size. Additionally, based on the selected refinement rate r_L and noise scales $\{\sigma_i\}_{i=1}^L$, the upper bound of switch time t_{σ_i} can be estimated when $i < L$.*

Refinement rate

As discussed above, the final prediction result depends directly on the last denoising process. It is determined by three parameters: 1) the last noise level σ_L; 2) step size parameter ϵ; 3) the number of the last refinement step. The first two parameters are also the components of the refinement rate, which affects the number of refinement steps of previous noise levels according to Proposition 2. Therefore, we need to specify the relationship between these parameters and the prediction error.

Proposition 3. *Let $e(y_t, x_I)$ represent the error between the estimated score and the ground truth score, i.e.,*

$$e(y_t, x_I) = s_\theta(y_t, \sigma_L, x_I) + \frac{y_t - y_I}{\sigma_L^2}.$$

After t times of denoising, the current estimated prediction y_t satisfies

$$y_t = (1-r_L)^t \cdot (y_0(\sigma_L) - y_I) + y_I + \sum_{i=0}^{t-1} (1-r_L)^i \cdot E_i(x_I), \tag{16}$$

where $E_i(x_I) = \alpha_L \cdot e(y_i, x_I)$ represents the neural network error.

To analyze the constraints of these parameters, we need the value y_t in Eq. (16) to converge as much as possible to ground truth y_I. However, when $t \to \infty$, y_t cannot converge to y_I due to the estimated error $E_i(\cdot)$. Therefore, considering the prediction error between current y_t and y_I, we have the following proposition.

Proposition 4. *Assume the last initial denoising point $y_0(\sigma_L)$ follows Technique 2, i.e., $\|y_0(\sigma_L) - y_I\|_\infty < \beta(\sigma_{L-1})$. Let $E \in \mathbb{R}$ be the upper bound of $E_i(x_I)$ for all $i \geq 0$ and $(x_I, y_I) \in \mathcal{D}$, i.e., $\|E_i(x_I)\|_2 < E$. For an end-signal $\beta(\sigma_L)$, when*

$$|1-r_L|^t \cdot \sqrt{d} \cdot \beta(\sigma_{L-1}) + E \cdot \frac{1 - |1-r_L|^t}{r_L} < \beta(\sigma_L), \tag{17}$$

where d is the number of dimensions of y_I. we can guarantee that $\|y_t - y_I\|_2 < \beta(\sigma_L)$.

We can use the loss value obtained during training to approximate the error E caused by the network. The end-signal $\beta(\sigma_L)$ can also be regarded as the error bound we hope to achieve iteratively. The end-signal can be set based on task accuracy requirements. After the end-signal is selected, we can choose all three parameters based on the condition in Eq. (17). Additionally, the first term $|1-r_L|^t \cdot \sqrt{d} \cdot \beta(\sigma_{L-1})$ decreases as t increases. On the contrary, the second term $E \cdot \frac{1-|1-r_L|^t}{r_L}$ increases as t increases. On the other hand, we can choose a suitable T where the second term has a higher order than the first term, i.e., $(|1-r_L|^T \cdot \sqrt{d} \cdot \beta(\sigma_{L-1}))/(E \cdot \frac{1-|1-r_L|^T}{r_L}) \ll 1$. When inference time $t \geq T$, we can ensure that the error of the current prediction result is dominated by the network training error. This approach can balance the error caused by the neural network and the error caused by iteration as much as possible. In this case, $t \geq \log_{|1-r_L|} \frac{E}{\sqrt{d} \cdot \beta(\sigma_{L-1}) \cdot r_L + E}$. In summary,

Technique 3. *(Refinement rate) By setting an end-signal $\beta(\sigma_L)$, choose suitable r_L and T in the sampling process following Eq. (17). Additionally, these parameters can make the second term have a higher order of magnitude than the first term in Eq. (17).*

Weight of Loss

The choice of coefficient $\lambda(\cdot)$ in Eq (8) can affect how well the network is trained for different noise levels. In (Song and Ermon 2019), the authors aimed to let the value of $\lambda(\sigma_i) l(\theta; \sigma_i)$ in Eq (6) at the same order of magnitude for all $\{\sigma_i\}_{i=1}^L$. To achieve this, they chose $\lambda(\sigma_i)$ as σ_i^2. Setting the same magnitude, the loss of the score network learns to a similar degree for each level of noise in each back-propagation and gradient descent.

When applying Techniques 2 and 3, the number of denoising steps for different noises is different, and the last step of denoising is the most important. Based on this, to obtain a better estimation of the score function for the last noise level, an intuitive idea is to enhance the training weight in the loss function (Caruana 1997). Therefore, we choose a different coefficient $\lambda(\sigma_i)$ in Eq (8), specifically taken as σ_i^k and $k < 2$. Since $\{\sigma_i\}_{i=1}^L$ is a geometric progression and $\sigma_i^2 l(\theta; \sigma_i) \propto 1$, $\sigma_i^k l(\theta; \sigma_i) \propto \sigma_i^{k-2}$ is also a geometric progression getting the maximum value when $i = L$, i.e., the last noise level. Based on this, we propose a coefficient selection strategy.

Technique 4. *(Weigh of Loss) Choosing suitable coefficients $\lambda(\sigma_i)$ in the loss (Eq. (8)) increases the magnitude of the value $\lambda(\sigma_i) l(\theta; \sigma_i, x, y)$ as i increases, such as $\lambda(\sigma_i) = \sigma_i$.*

Algorithm 1: Training

Initialization: $\{\sigma_i\}_{i=1}^{L}$, training set $\mathcal{D}$, loss coefficient $\lambda(\sigma_i)$

```
repeat
    choose (x, y) ∈ D, σ ∈ {σ_i}_{i=1}^L, and ỹ ~ N(y, σ)
    Take gradient descent step on
    ∇_θ (1/2) λ(σ_i) ‖s_θ(ỹ, σ, x) + (ỹ − y)/σ²‖²_2
until
converged
```

$$\nabla_\theta \frac{1}{2}\lambda(\sigma_i)\|s_\theta(\widetilde{y},\sigma,x)+\frac{\widetilde{y}-y}{\sigma^2}\|_2^2$$

Algorithm 2: Inference

Initialization: $\{\sigma_i\}_{i=1}^{L}, \varepsilon, T, \{\beta(\sigma_i)\}_{i=1}^{L-1}, x_I$

```
Initialize y_0
for i ← 1 to L − 1 do
    α_i ← ε · σ_i² / σ_L²
    repeat
        y_t ← y_{t−1} + α_i s_θ(y_{t−1}, σ_i, x_I)
    until σ_i² · ‖s_θ(y_t, σ_i, x_I)‖_∞ < β(σ_i)
    y_0 ← y_T
end for
for t ← 1 to T do
    y_t ← y_{t−1} + ε s_θ(y_{t−1}, σ_L, x_I)
end for
return y_T
```

By employing Techniques 1–4 together, the complete training and inference procedure are shown as Alg. 1 and Alg. 2.

Experiment

Toy examples

To demonstrate the effectiveness of SSM and all improved techniques, we first perform experiments on 5 selected toy examples (linear regression, quadratic regression, log-log linear regression, log-log cubic regression, and sinusoidal regression) proposed in (Han, Zheng, and Zhou 2022). We add unbiased normal distribution noise to the original models to obtain noisy data for training, increasing model complexity. We aim to predict accurate regression results for pure input data in the inference process. To evaluate the impact of various technologies, we have designed an ablation experiment that considers three variables: the use of original Langevin dynamics, the implementation of the fast sampling technique as described in Technique 2, and the application of different coefficients $\lambda(\sigma_i)$ in the loss function, where L1 and L2 represent the degree of noise levels of 1 and 2, respectively. The experiment results with the Root Mean Squared Error (RMSE) and inference time about different model settings are shown in Table 2 and 3, respectively. The scatter plots for 5 examples of both true and generated data following the first setting, i.e., L1, w/o noise fast, are shown in Figure 1. We observe that while perturbed data is used, SSM can still fit the regression equation of the original data. The proposed fast sampling Technique 2 significantly reduces the inference time for all tasks. Additionally, since the fast sampling technique ensures the error bound of the current estimated state, it can even achieve the best performance on tasks except for log-log linear regression. Using the noise-free Langevin equation, i.e., w/o noise, performs better on most tasks, indicating that the selection of different coefficients also affects the performance of the model.

Regression

We further evaluate our model on 10 UCI regression tasks (Dua and Graff 2017). We employ the same experimental settings as those used in the CARD model (Han, Zheng, and Zhou 2022). We compare our model with NGboost (Duan et al. 2020), CARD (Han, Zheng, and Zhou 2022), DBT (Han and Zhou 2024), non-probabilistic GBM models, including CatBoost (Prokhorenkova et al. 2018) and XGBoost (Chen and Guestrin 2016), and GCDS (Zhou et al. 2023). The experiment results with RMSE are shown in Table 4. It is noticed that SSM outperforms existing approaches in RMSE. SSM (L2) obtains the best results in 8 out of 10 datasets. Although the RMSE of SSM (L1) is larger, it has a smaller standard deviation, which means the network is more stable. In addition, probabilistic models such as CARD predict the probability of distribution, while SSM predicts the points directly. This also leads to a lower standard deviation of SSM on most tasks.

Classification

For classification tasks, we compare our model with CARD on CIFAR-10 and CIFAR-100, focusing on both accuracy and the inference time (Krizhevsky 2009). Table 1 shows the comparison results of accuracy for CIFAR-10 and CIFAR-100 classification. We observe that SSM can achieve excellent performance for prediction accuracy with lower variance. This shows that our method has more stable prediction results when selecting different random number seeds. Additionally, SSM can significantly shorten the inference time while ensuring performance compared with CARD, which has the same network structure.

Model		CARD	SSM (L1)	SSM (L2)
CIFAR-10	Accuracy	90.93 ± 0.02	90.91 ± 0.00	$\mathbf{90.99 \pm 0.00}$
	Time	50.30 ± 0.30	19.63 ± 1.21	$\mathbf{15.43 \pm 1.12}$
CIFAR-100	Accuracy	71.42 ± 0.01	$\mathbf{71.51 \pm 0.00}$	71.38 ± 0.00
	Time	90.58 ± 0.12	32.30 ± 0.60	$\mathbf{27.16 \pm 0.19}$

Table 1: Accuracy (in %) and inference time (in second (s)) on CIFAR-10 and CIFAR-100.

Conclusion

In this paper, we proposed Supervised Score-based Model (SSM), score-based gradient boosting model using denoising score matching for supervised learning. First, we analyzed the connection between SSM and GBM and showed that the iterative equation of GBM can be converted to a noise-free Langevin equation. Then, we provided a theoretical analysis on learning and sampling for SSM to balance inference time and prediction accuracy with noise-free Langevin sampling. Furthermore, we showed the effectiveness of the proposed techniques on selected toy examples. Lastly, we compared SSM with existing GBM models and diffusion models, for several regression and classification tasks. The experimental results show that our model outperforms in both accuracy and inference time.

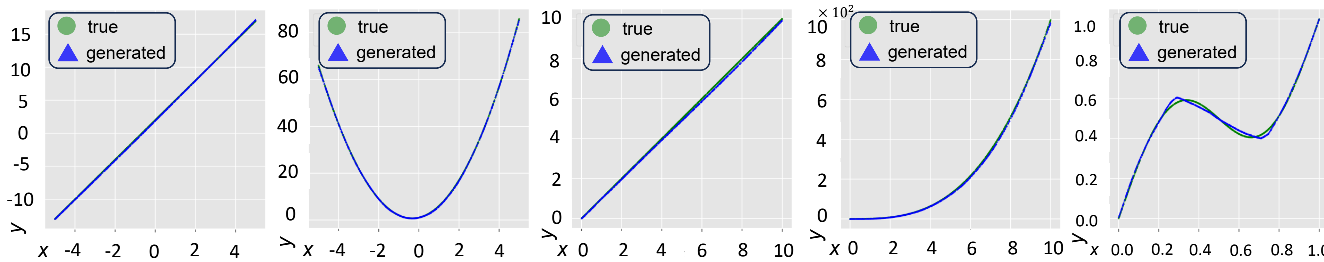

Figure 1: The scatter plots for toy examples. From left to right: linear regression, quadratic regression, log-log linear regression, log-log cubic regression, and sinusoidal regression. The green and blue points represent the true values and the prediction results generated by 1000 samples, respectively.

Task \ Technique	RMSE ↓							
	$\lambda(\sigma_i) = \sigma_i$ (L1)				$\lambda(\sigma_i) = \sigma_i^2$ (L2)			
	w/o noise, fast	w/ noise, fast	w/ noise	w/o noise	w/ noise, fast	w/o noise	w/o noise, fast	w/ noise
Linear	**0.0667**	0.0668	0.1767	0.0773	0.0685	0.0822	0.0684	0.1764
Quadratic	0.1197	0.1198	0.1103	0.1296	0.0951	0.1026	**0.0950**	0.1139
Log-Log Linear	0.0783	0.0781	0.0857	**0.0654**	0.0793	0.0885	0.0794	0.0699
Log-Log Cubic	3.2704	3.2213	3.3903	3.9410	**1.9919**	2.9880	2.0050	3.2718
Sinusoidal	0.0096	0.0097	0.0098	0.0094	0.0035	0.0033	**0.0033**	0.0034

Table 2: RMSE on different ablation settings on toy examples. W/ and w/o noise represent whether the noise term is included in Langvin dynamics; fast denotes the use of Technique 2; L1 and L2 represent the degree of noise levels of 1 and 2, respectively.

Task \ Technique	Inference Time (s) ↓							
	$\lambda(\sigma_i) = \sigma_i$ (L1)				$\lambda(\sigma_i) = \sigma_i^2$ (L2)			
	w/o noise, fast	w/ noise, fast	w/ noise	w/o noise	w/ noise, fast	w/o noise	w/o noise, fast	w/ noise
Linear	0.0385	**0.0354**	0.1404	0.1114	0.0376	0.1106	0.0366	0.1404
Quadratic	**0.2743**	0.2774	1.0957	0.9622	0.3262	0.9631	0.3252	1.0909
Log-Log Linear	0.4151	0.4176	1.1151	0.9771	0.3425	0.9647	**0.3415**	1.0930
Log-Log Cubic	0.1311	0.1319	0.5741	0.5015	**0.0987**	0.4996	0.0998	0.5843
Sinusoidal	0.1774	**0.1766**	1.0863	0.9545	0.2103	0.9590	0.2066	1.1023

Table 3: Inference time on different ablation settings on toy examples. W/ and w/o noise represent whether the noise term is included in Langevin dynamics; fast denotes the use of Technique 2; L1 and L2 represent the degree of noise levels of 1 and 2, respectively.

Dataset	RMSE ↓							
	NGboost	CatBoost	XGBoost	GCDS	CARD	DBT	SSM (L1)	SSM (L2)
Boston	2.94 ± 0.53	2.19 ± 0.09	2.23 ± 0.04	2.75 ± 0.58	2.61 ± 0.63	2.73 ± 0.62	2.06 ± 0.19	$\mathbf{2.04 \pm 0.14}$
Concrete	5.06 ± 0.61	4.75 ± 0.16	4.62 ± 0.13	5.39 ± 0.55	4.77 ± 0.46	4.56 ± 0.50	5.88 ± 1.00	$\mathbf{3.20 \pm 1.00}$
Energy	0.46 ± 0.06	0.32 ± 0.02	$\mathbf{0.23 \pm 0.01}$	0.64 ± 0.09	0.52 ± 0.07	0.52 ± 0.07	0.36 ± 0.09	0.31 ± 0.02
Kin8nm	16.03 ± 0.04	9.31 ± 0.15	11.89 ± 0.17	8.88 ± 0.42	6.32 ± 0.18	7.04 ± 0.23	8.30 ± 0.10	$\mathbf{4.66 \pm 0.30}$
Naval	0.13 ± 0.00	0.13 ± 0.00	0.10 ± 0.00	0.14 ± 0.05	0.02 ± 0.00	0.07 ± 0.01	0.10 ± 0.00	$\mathbf{0.01 \pm 0.00}$
Power	3.79 ± 0.18	3.41 ± 0.03	$\mathbf{3.22 \pm 0.03}$	4.11 ± 0.16	3.93 ± 0.17	3.95 ± 0.16	4.25 ± 0.01	3.67 ± 0.33
Protein	4.33 ± 0.03	3.83 ± 0.02	$\mathbf{3.53 \pm 0.01}$	4.50 ± 0.02	3.73 ± 0.01	3.81 ± 0.04	5.20 ± 0.01	3.95 ± 0.43
Wine	0.63 ± 0.04	0.58 ± 0.02	0.57 ± 0.01	0.66 ± 0.04	0.63 ± 0.04	0.61 ± 0.04	0.68 ± 0.01	$\mathbf{0.46 \pm 0.21}$
Yacht	0.50 ± 0.20	0.44 ± 0.07	0.43 ± 0.08	0.79 ± 0.26	0.65 ± 0.25	1.08 ± 0.39	0.56 ± 0.16	$\mathbf{0.37 \pm 0.21}$
Year	$8.94 \pm$ NA	$8.94 \pm$ NA	$8.77 \pm$ NA	$9.20 \pm$ NA	$\mathbf{8.70 \pm}$ **NA**	$8.81 \pm$ NA	$10.14 \pm$ NA	$10.04 \pm$ NA
# best	0	0	3	0	1	0	0	6

Table 4: RMSE results for UCI regression tasks. For Kin8nm and Naval dataset, the results are multiplied by 100 to match the scale of others.

Acknowledgments

This research is supported by the National Research Foundation, Singapore, and Infocomm Media Development Authority under its Future Communications Research & Development Programme, Defence Science Organisation (DSO) National Laboratories under the AI Singapore Programme (FCP-NTU-RG-2022-010 and FCP-ASTAR-TG-2022-003), Singapore Ministry of Education (MOE) Tier 1 (RG87/22), the NTU Centre for Computational Technologies in Finance (NTU-CCTF), and Seitee Pte Ltd, and the National Research Foundation, Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme.

References

Amit, T.; Shaharbany, T.; Nachmani, E.; and Wolf, L. 2021. Segdiff: Image segmentation with diffusion probabilistic models. *arXiv preprint arXiv:2112.00390*.

Beltran-Velez, N.; Grande, A. A.; Nazaret, A.; Kucukelbir, A.; and Blei, D. 2024. Treeffuser: Probabilistic Predictions via Conditional Diffusions with Gradient-Boosted Trees. *arXiv preprint arXiv:2406.07658*.

Bentéjac, C.; Csörgő, A.; and Martínez-Muñoz, G. 2021. A comparative analysis of gradient boosting algorithms. *Artificial Intelligence Review*, 54: 1937–1967.

Blundell, C.; Cornebise, J.; Kavukcuoglu, K.; and Wierstra, D. 2015. Weight uncertainty in neural network. In *International conference on machine learning*, 1613–1622. PMLR.

Cao, H.; Tan, C.; Gao, Z.; Xu, Y.; Chen, G.; Heng, P.-A.; and Li, S. Z. 2024. A survey on generative diffusion models. *IEEE Transactions on Knowledge and Data Engineering*.

Caruana, R. 1997. Multitask learning. *Machine learning*, 28: 41–75.

Chen, T.; and Guestrin, C. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 785–794.

Croitoru, F.-A.; Hondru, V.; Ionescu, R. T.; and Shah, M. 2023. Diffusion models in vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Cunningham, P.; Cord, M.; and Delany, S. J. 2008. Supervised learning. In *Machine learning techniques for multimedia: case studies on organization and retrieval*, 21–49. Springer.

Dhariwal, P.; and Nichol, A. 2021. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34: 8780–8794.

Du, H.; Zhang, R.; Liu, Y.; Wang, J.; Lin, Y.; Li, Z.; Niyato, D.; Kang, J.; Xiong, Z.; Cui, S.; et al. 2023. Beyond deep reinforcement learning: A tutorial on generative diffusion models in network optimization. *arXiv preprint arXiv:2308.05384*.

Dua, D.; and Graff, C. 2017. UCI Machine Learning Repository.

Duan, T.; Anand, A.; Ding, D. Y.; Thai, K. K.; Basu, S.; Ng, A.; and Schuler, A. 2020. Ngboost: Natural gradient boosting for probabilistic prediction. In *International conference on machine learning*, 2690–2700. PMLR.

Friedman, J. H. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189–1232.

Grafarend, E. 2006. *Linear and Nonlinear Models: Fixed Effects, Random Effects, and Mixed Models*. Walter de Gruyter. ISBN 9783110162165.

Han, X.; Zheng, H.; and Zhou, M. 2022. Card: Classification and regression diffusion models. *Advances in Neural Information Processing Systems*, 35: 18100–18115.

Han, X.; and Zhou, M. 2024. Diffusion Boosted Trees. *arXiv preprint arXiv:2406.01813*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33: 6840–6851.

Ho, J.; Salimans, T.; Gritsenko, A.; Chan, W.; Norouzi, M.; and Fleet, D. J. 2022. Video diffusion models. *Advances in Neural Information Processing Systems*, 35: 8633–8646.

Hyvärinen, A.; and Dayan, P. 2005. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4).

Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; and Liu, T.-Y. 2017. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30.

Kong, Z.; Ping, W.; Huang, J.; Zhao, K.; and Catanzaro, B. 2020. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*.

Krizhevsky, A. 2009. Learning Multiple Layers of Features from Tiny Images.

Liu, G.; Van Huynh, N.; Du, H.; Hoang, D. T.; Niyato, D.; Zhu, K.; Kang, J.; Xiong, Z.; Jamalipour, A.; and Kim, D. I. 2024. Generative AI for Unmanned Vehicle Swarms: Challenges, Applications and Opportunities. *arXiv preprint arXiv:2402.18062*.

Natekin, A.; and Knoll, A. 2013. Gradient boosting machines, a tutorial. *Frontiers in neurorobotics*, 7: 21.

Nichol, A. Q.; and Dhariwal, P. 2021. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, 8162–8171. PMLR.

Prokhorenkova, L.; Gusev, G.; Vorobev, A.; Dorogush, A. V.; and Gulin, A. 2018. CatBoost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31.

Rahman, A.; Valanarasu, J. M. J.; Hacihaliloglu, I.; and Patel, V. M. 2023. Ambiguous medical image segmentation using diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11536–11546.

Robert, C. P.; Casella, G.; and Casella, G. 1999. *Monte Carlo statistical methods*, volume 2. Springer.

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

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2021. High-Resolution Image Synthesis with Latent Diffusion Models. arXiv:2112.10752.

Song, J.; Meng, C.; and Ermon, S. 2020. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.

Song, Y.; and Ermon, S. 2019. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32.

Song, Y.; and Ermon, S. 2020. Improved techniques for training score-based generative models. *Advances in neural information processing systems*, 33: 12438–12448.

Song, Y.; Garg, S.; Shi, J.; and Ermon, S. 2020a. Sliced score matching: A scalable approach to density and score estimation. In *Uncertainty in Artificial Intelligence*, 574–584. PMLR.

Song, Y.; and Kingma, D. P. 2021. How to train your energy-based models. *arXiv preprint arXiv:2101.03288*.

Song, Y.; Sohl-Dickstein, J.; Kingma, D. P.; Kumar, A.; Ermon, S.; and Poole, B. 2020b. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*.

Tomczak, M.; Swaroop, S.; Foong, A.; and Turner, R. 2021. Collapsed variational bounds for Bayesian neural networks. *Advances in Neural Information Processing Systems*, 34: 25412–25426.

Vincent, P. 2011. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7): 1661–1674.

Welling, M.; and Teh, Y. W. 2011. Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, 681–688. Citeseer.

Yang, L.; Zhang, Z.; Song, Y.; Hong, S.; Xu, R.; Zhao, Y.; Zhang, W.; Cui, B.; and Yang, M.-H. 2023. Diffusion models: A comprehensive survey of methods and applications. *ACM Computing Surveys*, 56(4): 1–39.

Zhou, X.; Jiao, Y.; Liu, J.; and Huang, J. 2023. A deep generative approach to conditional sampling. *Journal of the American Statistical Association*, 118(543): 1837–1848.

Zimmermann, R. S.; Schott, L.; Song, Y.; Dunn, B. A.; and Klindt, D. A. 2021. Score-based generative classifiers. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*.