From score matching to score-based generative modeling

Maxim Kondracki, Eustache Le Bihan

January 12, 2024

Abstract

In this report, we explore the modeling of complex probability distributions in machine learning via score matching techniques. Based on three main papers ([1] [3] [5]), we explore various approaches for handling score matching, focusing on its ability to circumvent the normalization challenges. Through a series of analyses, we present methods such as Denoising Score Matching, Sliced Score Matching, and Langevin dynamics, providing theoretical and practical perspectives. Additionally, we conducted retraining of the Noise Conditional Score Networks (NCSN) on our specific dataset to confirm certain findings and insights.

1 Introduction: estimating probability distributions

In the realm of machine learning, there has been intense effort in modeling probability distributions. Indeed, such modeling should enable sampling from these distributions to generate data but also evaluate the probability of existing data coming from those distributions. Methods consisting in modeling probability distributions aim to provide a family of parametrized probability distributions p_θ. The objective would then be to minimize the distance between p_θ and p_{data}, in other words match p_θ to p_{data}.

Classical Bayesian machine learning methods would circumvent the question of modeling the distribution by using reasonable models of the data, meaning closed-form distributions, and assigning prior belief to these models. Yet, such methods do not allow us to model complex distributions. Modern approaches would rather use deep neural networks (DNN), leveraging their ability to model complex and high-dimensional distributions.

Such DNNs would typically output real values from high-dimensional data, modeling an unnormalized and not necessarily positive function f_θ. This concern is solved by applying the following transformation: $\frac{e^{f_\theta}}{Z_\theta}$ with $Z_{\boldsymbol{\theta}}$ a normalizing constant computed with a high-dimensional integral $Z_{\boldsymbol{\theta}} = \int e^{f_{\boldsymbol{\theta}}(\mathbf{x})} \mathrm{d}\mathbf{x}$, available in closed form for simple models (example of Gaussian) but intractable in the general case.

This difficulty has been tackled by either approximating the normalizing constant or using restricted DNNs such as autoregressive models (year 2000) or normalizing flow models (year 2014), where $Z_{\boldsymbol{\theta}}$ becomes tractable by construction. Nevertheless, restricting DNNs in this way limits their flexibility by restricting the family of probability distributions that can be modeled. Another popular alternative has been to model the generation process directly with so-called Generative Adversarial Networks (GAN), but such models only respond to one of our previously stated objectives: they cannot evaluate the probability for existing data.

In this report, we will explore another approach: score matching. Such methods relies on the Stein score $s_\theta(\mathbf{x}) = \nabla_{\mathbf{x}} \log p_{\boldsymbol{\theta}}(\mathbf{x}) = \nabla_{\mathbf{x}} f_{\boldsymbol{\theta}}(\mathbf{x})$ to learn p_θ, freeing from the normalizing constant while still enabling sampling and probability evaluation.

2 Score matching

2.1 Why do we need normalization ?

Consider modeling a probability distribution with a neural network using a sigmoid activation function. This function produces values between 0 and 1, making it suitable for representing probabilities. When training the model with a log-likelihood maximization approach, as in Bayesian machine learning, without normalization, it tends to output 1 for all input data. This highlights the importance of normalization in the training process.

2.2 Some intuition: energy based models

As explained in introduction, the energy based approach aim to circumvent the issue with making sure that a DNN modeling a probability distribution respect its fundamental requirements: positivity and suming to one. Energy-based models define probabilities indirectly by outputing E_θ:

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \frac{1}{Z_{\boldsymbol{\theta}}} e^{-E_{\boldsymbol{\theta}}(\boldsymbol{x})} \tag{1}$$

Employing the exponential function for positivity, unlike the square function, captures more probability variations as it operates as a bijection from $\mathbb{R}$ to $\mathbb{R}^+$. Additionally, common distributions, aligning with principles like maximum entropy and the second law of thermodynamics, can be expressed in this form and are prevalent in statistical physics.

2.3 Stein score: solving the normalization issue

Stein score is defined as :

$$s_\theta(\mathbf{x}) = \nabla_{\mathbf{x}} \log p_{\boldsymbol{\theta}}(\mathbf{x}) \tag{2}$$

First, one might note the difference with traditional score function in statistics, where we take the gradient in regard of the parameters while here we do so in regard of the data. Secondly, let's note $f_\theta = -E_\theta$. It is obvious that $s_\theta(\mathbf{x}) = \nabla_{\mathbf{x}} f_{\boldsymbol{\theta}}(\mathbf{x})$, freeing from $Z_{\boldsymbol{\theta}}$.

While [2] and [5] proposed to use this score to learn non-normalized statistical models, meaning learn f_θ, [3] learns directly the score s_θ as a parameterized neural network.

2.4 Objective function

Matching p_θ to p_{data} now resolves in matching s_θ to $\nabla_{\mathbf{x}} \log p_{data}$, we hence define the objective function, described by [5] as *Explicit Score Matching (ESM)* but also known as *Fisher divergence*, and its associated estimator:

$$J_{ESM}(\boldsymbol{\theta}) = \frac{1}{2}\mathbb{E}_{p_{\boldsymbol{data}}(\boldsymbol{x})}[\|\nabla_{\mathbf{x}} \log p_{\boldsymbol{data}}(\boldsymbol{x}) - s_\theta(\boldsymbol{x})\|^2] \tag{3}$$

$$\hat{\boldsymbol{\theta}}_{ESM} = \arg\min_\theta J_{ESM}(\boldsymbol{\theta}) \tag{4}$$

Yet J_{ESM} is not easily computable as we do not have access to a closed form of $p_{\boldsymbol{data}}$. Hyvärinen [2] proved is his 2005 paper that such estimation can still be achieved via a proxy objective, later described by Vincent [5] as *Implicit Score Matching (ISM)*:

$$J_{ISM}(\boldsymbol{\theta}) = \mathbb{E}_{p_{\boldsymbol{data}}(\boldsymbol{x})}[Tr(\nabla_{\mathbf{x}} s_\theta(\boldsymbol{x})) + \frac{1}{2}\|s_\theta(\boldsymbol{x})\|^2] \tag{5}$$

In practice, for n i.i.d. samples $\mathbf{x}_1, ..., \mathbf{x}_n \in \mathbb{R}^d$, the expectation is estimated using a mean over all samples, with its associated estimator:

$$\hat{J}_{ISM}(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n}[Tr(\nabla_{\mathbf{x}} s_\theta(\mathbf{x}_i)) + \frac{1}{2}\|s_\theta(\mathbf{x}_i)\|^2] \tag{6}$$

$$\hat{\boldsymbol{\theta}}_{n,ISM} = \arg\min_\theta \hat{J}_{ISM}(\boldsymbol{\theta}) \tag{7}$$

[2] showed that $\hat{\boldsymbol{\theta}}_{n,ISM} \xrightarrow{p} \hat{\boldsymbol{\theta}}^*_{ESM}$ (meaning that sample size approaches infinity), where $\hat{\boldsymbol{\theta}}^*_{ESM}$ is such that $p_{\hat{\boldsymbol{\theta}}^*_{ESM}} = p_{data}$. This result justifies the relevance of such implementation.

Nevertheless, let's remember that our neural network $s_\theta : \mathbb{R}^d \to \mathbb{R}^d$ models a gradient vector field. For this reason, computing $Tr(\nabla_{\mathbf{x}} s_\theta(\mathbf{x}_i))$ for a given sample $\mathbf{x}_i$ would require d backpropagation passes to get each diagonal element of the hessian $\nabla_{\mathbf{x}} s_\theta(\mathbf{x}_i)$. For high dimensional data like images, for example the dataset *CelebA-HQ* where $d = 1024 * 1024$, such a computation would be too slow and thus makes this approach unscalable. Below we discuss two ways to deal with the scalability of the score estimation.

2.5 Denoising score matching

As shown by Vincent [5], denoising autoencoder training criterion is equivalent to matching the score (with respect to the data) of a specific energy based model. This method does not require the heavy computation of $Tr(\nabla_{\mathbf{x}} s_\theta(\mathbf{x}_i))$.

The data point $\mathbf{x}$ is first perturbed by a noise leading to a corrupted data point $\tilde{\mathbf{x}}$ which follows the distribution $q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})$. Instead of estimating directly the distribution $q_\sigma(\mathbf{x})$, we try to estimate the corrupted density $q_\sigma(\tilde{\mathbf{x}}) = \int q_\sigma(\tilde{\mathbf{x}}|\mathbf{x}) p_{\text{data}}(\mathbf{x})\, d\mathbf{x}$

$$J_{\text{DSM}_{q_\sigma}}(\theta) = \frac{1}{2}\mathbb{E}_{q_\sigma(\tilde{\mathbf{x}},\mathbf{x})}\left[\|s_\theta(\tilde{\mathbf{x}}) - \nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})\|^2\right] \tag{8}$$

The intuition is that following the gradient of the log density at some corrupted point $\tilde{\mathbf{x}}$ gives us the direction towards the clean sample $\mathbf{x}$. Indeed we try to minimize the error between our estimated density of the corrupted data point $q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})$ and the gradient of the score defined as :

$$\nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}}|\mathbf{x}) = \frac{1}{\sigma^2}(\mathbf{x} - \tilde{\mathbf{x}}) \tag{9}$$

The direction $\frac{1}{\sigma^2}(\mathbf{x} - \tilde{\mathbf{x}})$ corresponds to going from corrupted $\tilde{\mathbf{x}}$ to clean $\mathbf{x}$. Vincent shows in his paper [5] that the optimal score network $s_{\theta^*}(\mathbf{x})$ that minimizes Eq. (8) satisfies $s_{\theta^*}(\mathbf{x}) = \nabla_x \log q_\sigma(\mathbf{x})$ almost surely. This result is very important since it avoid us to compute the trace of the gradient of the network.

2.6 Sliced score matching

[4] introduced a method to deal with such scaling issues: sliced score matching. The intuition behind it is that if the problem stems from high dimension, one may try to reduce it to a one-dimensional one. Since our score function is a vector field, a one-dimensional approach would be to make it a scalar field. This is done using projections of our score function on random vectors $\mathbf{v}$ with a given density $p_{\mathbf{v}}$. This yields the *Sliced Fisher Divergence*:

$$L(\boldsymbol{\theta}, p_{\mathbf{v}}) = \frac{1}{2}\mathbb{E}_{p_{\mathbf{v}}}\mathbb{E}_{p_{\boldsymbol{data}}(\boldsymbol{x})}[\|\mathbf{v}^T s_\theta(\boldsymbol{x}) - \mathbf{v}^T \nabla_{\mathbf{x}} \log p_{\boldsymbol{data}}(\boldsymbol{x})\|^2] \tag{10}$$

Following the idea of [2] with *Implicit Score Matching*, [4] show that this objective amounts the proxy objective, and its associated estimator:

$$J_{SlicedSM}(\boldsymbol{\theta}) = \mathbb{E}_{p_{\mathbf{v}}}\mathbb{E}_{p_{\boldsymbol{data}}(\boldsymbol{x})}[\mathbf{v}^T \nabla_{\mathbf{x}} s_\theta(x)\mathbf{v} + \frac{1}{2}(\mathbf{v}^T s_\theta(\boldsymbol{x}))^2] \tag{11}$$

$$\hat{\boldsymbol{\theta}}_{SlicedSM} = \arg\min_\theta J_{SlicedSM}(\boldsymbol{\theta}) \tag{12}$$

By writing $\mathbf{v}^T \nabla_{\mathbf{x}} s_\theta(x)\mathbf{v} = \mathbf{v}^T \nabla_{\mathbf{x}}(\mathbf{v}^T s_\theta(x))$, one can see that $\mathbf{v}^T s_\theta(x)$ can be computed in one forward pass by adding one neuron at the end of the computational graph, and that $\nabla_{\mathbf{x}}(\mathbf{v}^T s_\theta(x))$ is then obtained by one backpropagation of this computational graph. Now that we can compute $\mathbf{v}^T \nabla_{\mathbf{x}} s_\theta(x)\mathbf{v} + \frac{1}{2}(\mathbf{v}^T s_\theta(\boldsymbol{x}))^2$ for a given $\mathbf{x}$ and $\mathbf{v}$, the expectations would then be estimated by averaging over all samples and random directions. Yet, [4] goes even further by noticing that if $p_{\mathbf{v}}$ is multivariate standard normal or multivariate Rademacher distributions, $\mathbb{E}_{p_{\mathbf{v}}}[(\mathbf{v}^T s_\theta(\boldsymbol{x}))^2] = \|s_\theta(\boldsymbol{x})\|_2^2$, allowing to avoid approximating this term by averaging and thus reduce the variance of the estimation. This way, for samples $\mathbf{x}_1, ..., \mathbf{x}_n \in \mathbb{R}^d$ and m random directions drawn from $p_{\mathbf{v}}$ for each $\mathbf{x}_i$, we get the so-called *Sliced Score Matching Variance Reduction* estimation:

$$\hat{J}_{SSM-VR}(\boldsymbol{\theta}) = \frac{1}{n}\frac{1}{m}\sum_{i=1}^{n}[\sum_{j=1}^{m}\mathbf{v}_{ij}^{T}\nabla_{\mathbf{x}}s_\theta(\mathbf{x}_i)\mathbf{v}_{ij} + \frac{1}{2}\|s_\theta(\mathbf{x_i})\|_2^2] \tag{13}$$

We experimentally validated the scalability of our solution by employing a simple model with two hidden layers of size 10×dimension. A model with more parameters is necessary as the data dimension increases to effectively learn s_θ. We measured the computation time for both the vanilla *ISM* objective (eq. 5) and *SSM-VR* objective (eq. 13). Notably, for our score model and data in dimension 1000, the execution time is approximately 610 times greater, providing qualitative evidence of the scalability of SSM-VR (see Fig. 7).

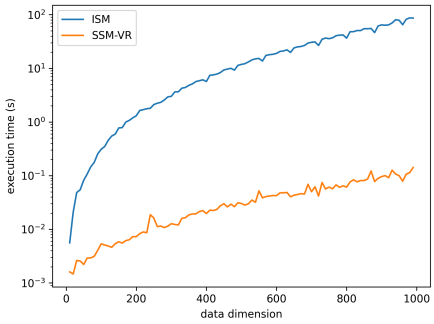

Figure 1: **Execution time:** ISM compared to SSM-VR.

2.7 Langevin dynamics: sampling from score-based models

2.7.1 Principle

We extensively discussed the theory and practice of modeling the Stein score. However, it's crucial to maintain focus on our primary goal: generating samples from a complex, high-dimensional distribution. Langevin dynamics offers a method to achieve this by sampling from p_{data} using a learned model of the score function $s_\theta(\mathbf{x}) = \nabla_{\mathbf{x}} \log p_{\boldsymbol{\theta}}(\mathbf{x})$. The concept is straightforward: generate random points from a prior distribution and then follow directions provided by our learned vector field $\nabla_{\mathbf{x}} \log p_{\boldsymbol{\theta}}$. Yet, a vanilla implementation may cause points to collapse in regions with the lowest divergence. To address this, we incorporate centered standard Gaussian noise, resulting in the algorithm (for a prior distribution-generated point $\tilde{\mathbf{x}}_0$):

$$\tilde{\mathbf{x}}_t = \tilde{\mathbf{x}}_{t-1} + \frac{\epsilon}{2}s_\theta(\tilde{\mathbf{x}}_{t-1}) + \sqrt{\epsilon}\,\mathbf{z}_t,\ \mathbf{z}_t \sim \mathcal{N}(0,\, I) \tag{14}$$

[6] showed that by replacing s_θ by its exact value $\nabla_{\mathbf{x}} \log p_{data}$ and for $\epsilon \to 0$ and $t \to \infty$, $\tilde{\mathbf{x}}_t \sim p_{data}$, confirming the intuition. In practice, we simply take ϵ and t respectively small and big enough.

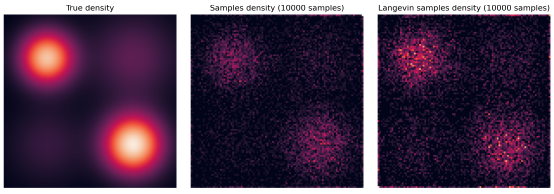

Figure 2: **Langevin dynamics:** generating samples from a GMM using a score MLP model trained with *SSM-VR*

2.7.2 Limitations: low data density regions

Gradient information conveys local details, yet in score matching, the goal is to minimize the Fisher divergence (3) by assessing its value on a training dataset. Consequently, it becomes apparent that in regions with low data density, characterized by fewer points in the training set, this approximation tends to be inaccurate. [3] underscores this challenge, highlighting its limitation in learning complex data distributions such as the *CIFAR-10* dataset. We experimentally validated this observation on a mixture of four Gaussians, training a three-hidden-layer MLP with 128 neurons using *SSM-VR* to learn the score. We computed the true score using *PyTorch* automatic differentiation and obtained the score estimation error as a scalar field through Euclidean distances (see 3).

In low-density areas, Langevin dynamics struggle to reach probable points, hampering the method's effectiveness for learning high-dimensional data like images

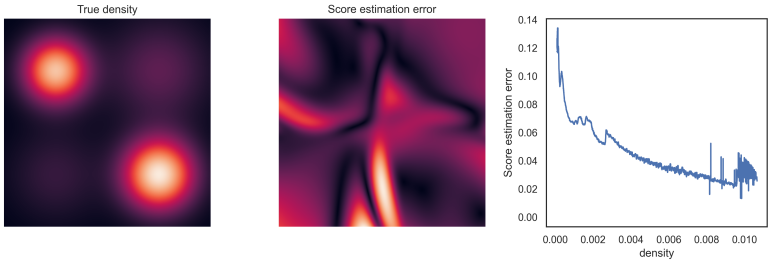

Figure 3: **Score estimation error:** in low-density data regions, score estimation errors surge. We bin error values based on regularly spaced density bins, calculating mean density and error values for each.

2.7.3 Solution: annealed langevin dynamics

We experimentally validated the impact of data density on score estimation error, particularly highlighting the detrimental effects of Langevin dynamics sampling. Annealed Langevin dynamics, inspired by principles from score matching and denoising diffusion probability models [1], involves perturbing the data distribution with noise of known variance. The idea is to perturb data distribution with noise with known variance and to learn the noise conditional score of the perturbed density $s_\theta(\mathbf{x}, \sigma)$. This way, we can first generate random samples that will likely fall in low density regions, follow Langevin dynamics using high variance that will give better approximation of the score and then when reaching regions with higher probability gradualy reduce the noise. Experimentally, we trained the noise conditional score model for 10 standard deviation values. As depicted in Figure 4, except for the outlier $\sigma = 1$, the score estimation error is generally lowest in low-density regions for higher σ (of the unperturbed distribution). It is noteworthy that the training of the noise conditional score model introduces high uncertainties.

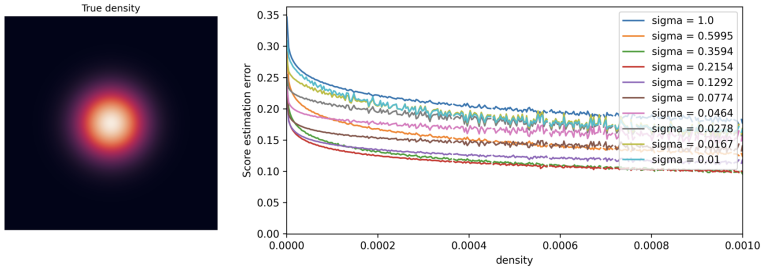

Figure 4: **Score estimation error:** evaluation for different noise conditional score networks

3 Generative modeling using score matching

3.1 Noise Conditional Score Networks

The introduction of Gaussian noise serves to populate the sparser areas of the original, unperturbed data distribution, thereby providing a more robust training signal for score matching. As shown before, the estimation of the score matching is greatly improved in dense regions. Moreover, employing varying levels of noise leads to a progression of disturbed distributions that gradually align with the true data distribution. This approach enhances the efficiency of annealed Langevin dynamics across multimodal distributions.

The NCSM first apply different noise intensities to the data, and concurrently computes scores through a noise-level condition score network. During the inference, the samples are generated using annealed Langevin dynamics, we start with scores associated with higher noise levels and progressively decrease the noise intensity. This section evaluates the Noise Conditional Score Network (NCSN) from Yang Song's research, focusing on the MNIST setup tested on our custom dataset.

3.2 Dataset

For this segment, we employed a custom dataset designed to replicate various shapes (triangle, bar, concentric circles, disc, square) in a format close to MNIST (28x28 pixels with varying pixel intensities). By initially generating uniform random values across the entire image and subsequently overlaying a mask representing the desired shape, we created diverse sample following the same distribution for each geometric figure. Evaluating the model on this novel dataset sould provide deeper insights into its performance and characteristics.

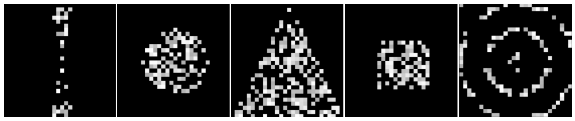

Figure 5: Samples of the dataset

3.3 Results

After implementing the network structure proposed by Song and applying a DSM loss on our tailored dataset, we achieved the outcomes illustrated in Figure 6, produced through the annealed Langevin method. The score network, when provided with an input $\boldsymbol{x}$ and a specific noise level σ_i, yields an estimated directional vector. As shown in equation (9), the aim of the network is to estimate the gradient of the score given a certain amount of noise. Since we start from a uniform data point density, we overcome the problem of low-density estimation.

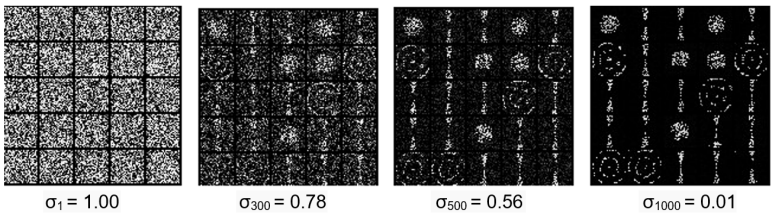

Figure 6: Annealed Langevin sampling from NCSN trained on our dataset

3.4 Sensitivity to disbalanced data

Here's an additional illustration showcasing the model's image generation capabilities on an imbalanced dataset that contains fewer triangles and concentric circles. In the previous outcomes, the model accurately predicted the distribution of shapes, ensuring that no points were generated outside the mask set during the construction of the original dataset. Despite the limited presence of discs and squares in the dataset, the model effortlessly produces precise shapes (see left figure). However, when concentric circles and triangles are in the minority, the model encounters difficulties in identifying the correct density function (see right figure).

Concentric circles and triangles exhibit complex distribution that require more samples for accurate representation. Their distribution is more dispersed compared to squares and circles, which have their maximum concentration centered in the middle of the image. If the model hasn't seen diverse instances of these shapes during training, it may struggle to generate them accurately.

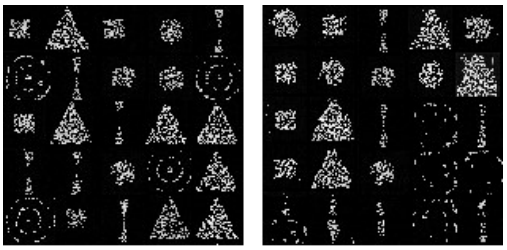

Figure 7: Annealed Langevin sampling from NCSN trained on a disbalanced version of our dataset. On the right Triangles and concentric circles are 13% of the total dataset size. On the left Triangles and concentric circles are 87% of the total dataset size. The rest is homogeneously distributed.

4 Conclusion

This sequential examination of the phases leading to the creation of NCSNs underscores the potency of this tool. Also, the class-independent nature of NCSNs signifies a paradigm shift in generative modeling, allowing for a more flexible and unsupervised approach to understanding and generating data. Presently, Yang Song is advancing his research on diffusion models, an extension of our investigation and a state-of-the-art performance in the realm of data generation.

References

[1] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020.

[2] Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6:695–709, 2005.

[3] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *CoRR*, abs/1907.05600, 2019.

[4] Yang Song, Sahaj Garg, Jiaxin Shi, and Stefano Ermon. Sliced score matching: A scalable approach to density and score estimation. *CoRR*, abs/1905.07088, 2019.

[5] Pascal Vincent. A connection between score matching and denoising autoencoders. Technical report, University of Montreal, 2010.

[6] Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, page 681–688, Madison, WI, USA, 2011. Omnipress.