Proposal of a Score Based Approach to Sampling Using Monte Carlo Estimation of Score and Oracle Access to Target Density

Anonymous Author(s)
Affiliation
Address
email

Abstract

Score based approaches to sampling have shown much success as a generative algorithm to produce new samples from a target density given a pool of initial samples. In this work, we consider if we have no initial samples from the target density, but rather 0th and 1st order oracle access to the log likelihood. Such problems may arise in Bayesian posterior sampling, or in training a network from data. Using this knowledge alone, we propose a Monte Carlo method to estimate the score empirically as a particular expectation of a random variable. Using this estimator, we can then run a discrete version of the backward flow SDE to produce samples from the target density. This approach has the benefit of not relying on a pool of initial samples from the target density, and it does not rely on a neural network or other black box model to estimate the score.

1 Introduction

The goal of score based sampling approaches is to produce samples θ_0 from some probability measure P_0 over $\mathbb{R}^d$ with density p_0. Of interest in this paper is the situation where we know the form of the target density up to normalization $p_0(\theta) \propto e^{f(\theta)-\frac{1}{2}\theta^T\theta}$ and we can query the function $f(\theta)$ as well as its gradient $\nabla f(\theta)$ at will. This may arise in two problems of interest. First is Bayesian posterior sampling, and second as a method to train networks from data. In Bayesian posteriors, θ may represent the parameters of a probability model and we have observed data $X_1, \cdots, X_n$. With knowledge of the likelihood $p(X_1, \cdots, X_n|\theta) = e^{f(X_1,\cdots,X_n,\theta)}$ and a normal prior on θ, the posterior is proportional to $p(\theta|X_1, \cdots, X_n) \propto e^{f(X_1,\cdots,X_n,\theta)-\frac{1}{2}\theta^T\theta}$.

Second, in training a single layer neural network we have n covariates $X_1, \cdots, X_n \in \mathbb{R}^d$ and response values $Y_1, \cdots, Y_n \in \mathbb{R}$. Using greedy algorithms [8], the problem of training a single layer network reduces to training a single neuron on the residuals of previously trained neurons. To train a single neuron with $\tanh$ activation, we want to find a weight vector $\theta \in \mathbb{R}^d$ to minimize $\min_\theta \sum_{i=1}^n (Y_i - \tanh(X_i^T\theta))^2 + \frac{1}{2}\theta^T\theta$ which is similar to the simpler objective $\max_\theta \sum_i Y_i \tanh(X_i^T\theta) - \frac{1}{2}\theta^T\theta$. This is a non-convex optimization problem, so instead of following an approach such as gradient descent we would like to sample the density $p(\theta) \propto e^{\alpha \sum_i Y_i \tanh(X_i^T\theta)-\frac{1}{2}\theta^T\theta}$. Points drawn from such a density will be near the minimizer of the non-convex objective. Regardless of the application in question, with knowledge of the functional form of the target density we wish to apply a score based approach to sampling.

Underlying all score based approaches to sampling is the dual relationship between the "forward flow" and "backward flow" stochastic differential equations (SDE). Consider the forward flow SDE

$$d\theta_t = f(\theta_t, t)dt + g(t)dW_t, \theta_0 \sim P_0, \tag{1}$$

Under review at the NeurIPS 2022 Workshop on Score-Based Methods. Do not distribute.

where the marginal distribution of θ_t at time t is denoted P_t with density p_t. Paired with this forward flow SDE is a backward flow SDE [1]

$$d\theta_t = [f(\theta_t, t) - g(t)^2 \nabla \log p_t(\theta_t)]dt + g(t)dW_t, \tag{2}$$

where we run time in reverse from $t = T$ to $t = 0$. The forward flow SDE takes $\theta_0 \sim P_0$ to $\theta_T \sim P_T$, while the backward flow SDE takes $\theta_T \sim P_T$ to $\theta_0 \sim P_0$. If we can sample P_T directly and implement the backward flow SDE then we can produce samples from P_0.

Crucial to the backward flow SDE is the score function, $\nabla \log p_t(\theta)$, which is not known and must be estimated. Popular generative approaches have a large pool of initial samples drawn from P_0. They then train a model such as a neural network to approximate the score at different time scales t by minimizing an objective function using the initial samples and noisy perturbations as training data [12, 13]. Various related methods such as [3], [4] propose improved methods to estimate the score, but are also ultimately generative and thus rely on initial samples. [5] applies score based diffusions as an approach to improve annealed importance sampling. It is not a generative approach as it does not rely on initial samples from the target density, but it ultimately trains a neural network to estimate the score by minimizing relative entropy. In this paper, we desire a method to estimate the score without any initial samples from P_0 and without appealing to a complex secondary model such as a neural network to estimate the score.

We will specifically consider the forward flow SDE as the Ornstein–Uhlenbeck process with drift $f(\theta, t) = -\theta$ and constant diffusion $g(t) = \sqrt{2}$

$$d\theta_t = -\theta_t dt + \sqrt{2} dW_t, \tag{3}$$

the invariant measure is standard normal and the conditional measure can be expressed as

$$\theta_t = e^{-t}\theta_0 + \sqrt{1 - e^{-2t}} Z, Z \sim N(0, I). \tag{4}$$

Thus the density $p_t(\theta_t)$ up to normalization can be expressed as a convolution

$$p_t(\theta_t) \propto \int p_0(\theta_0) p_t(\theta_t|\theta_0) d\theta_0, \tag{5}$$

and the score, which removes any dependence on normalizing constants, can be expressed as

$$\nabla_{\theta_t} \log p_t(\theta_t) = \frac{\int p_0(\theta_0) \nabla_{\theta_t} p_t(\theta_t|\theta_0) d\theta_0}{\int p_0(\theta_0) p_t(\theta_t|\theta_0) d\theta_0}. \tag{6}$$

With some manipulations, we can express the numerator and denominator as expectations over normal random variables, and thus can approximate each via averaging the integrand at normally drawn points, i.e. a Monte Carlo (MC) estimator.

2 Results

Applying the derivative in (6) to our knowledge of $p(\theta_t|\theta_0)$ from (4), the expression for the score is equivalent to the ratio of two normal expectations (see appendix for derivation)

$$\nabla \log p_t(\theta) = -\theta + \frac{e^{-t}}{\sqrt{1 - e^{-2t}}} \frac{E[U e^{f(\sqrt{1-e^{-2t}}U + e^{-t}\theta)}]}{E[e^{f(\sqrt{1-e^{-2t}}U + e^{-t}\theta)}]} \quad U \sim N(0, I), \tag{7}$$

and further applying integration by parts, the score is also equivalently written as

$$\nabla \log p_t(\theta) = -\theta + e^{-t} \frac{E[\nabla f(\sqrt{1 - e^{-2t}}U + e^{-t}\theta) e^{f(\sqrt{1-e^{-2t}}U + e^{-t}\theta)}]}{E[e^{f(\sqrt{1-e^{-2t}}U + e^{-t}\theta)}]} \quad U \sim N(0, I). \tag{8}$$

In order to compute expressions (7) and (8), we appeal to a MC estimator. Say we draw K points $U_1, \cdots, U_K \sim N(0, I)$ from a standard normal, and approximate the score in one of two ways

$$\hat{s}_1(\theta, t) = -\theta + \frac{e^{-t}}{\sqrt{1 - e^{-2t}}} \sum_{k=1}^{K} U_k \frac{e^{f(\sqrt{1-e^{-2t}}U_k + e^{-t}\theta)}}{\sum_{j=1}^{K} e^{f(\sqrt{1-e^{-2t}}U_j + e^{-t}\theta)}} \tag{9}$$

$$\hat{s}_2(\theta, t) = -\theta + e^{-t} \sum_{k=1}^{K} \nabla f(\sqrt{1 - e^{-2t}}U_k + e^{-t}\theta) \frac{e^{f(\sqrt{1-e^{-2t}}U_k + e^{-t}\theta)}}{\sum_{j=1}^{K} e^{f(\sqrt{1-e^{-2t}}U_j + e^{-t}\theta)}}. \tag{10}$$

If we define the weights w_k then the above are weighted averages of the sampled integrands and this is a form of importance sampling

$$w_k = \frac{e^{f(\sqrt{1-e^{-2t}}U_k + e^{-t}\theta)}}{\sum_{j=1}^{K} e^{f(\sqrt{1-e^{-2t}}U_j + e^{-t}\theta)}}. \tag{11}$$

For small t, say $t < 0.1$, we conjecture that $\hat{s}_2(\theta, t)$ is the better estimator as it will have lower variance since $\frac{e^{-t}}{\sqrt{1-e^{-2t}}}$ can grow large as $t \to 0$. Additionally, as $t \to 0$, $\nabla f(\sqrt{1-e^{-2t}}U + e^{-t}\theta) \to \nabla f(\theta)$ and estimate $\hat{s}_2(\theta, t)$ approaches $-\theta + \nabla f(\theta)$ for small t, that is the score of the target density.

For larger t, it would seem $\hat{s}_1(\theta, t)$ will have lower variance since U_k is less variable than the gradient of the target $\nabla f(U_k)$. Additionally, as $t \to \infty$ we have the weights w_k approach a θ independent value, so the expectation is some finite vector while $\frac{e^{-t}}{\sqrt{1-e^{-2t}}} \to 0$ thus the score approaches $-\theta$, the score of a standard normal.

The algorithm to generate a sample from P_0 is as follows. Pick a terminal time $T > 0$, step size $\delta > 0$, and MC sample size $K \in \mathbb{N}$. Initialize $\theta_T \sim N(0, I)$ and $t = T$. While $t > 0$, in each iteration draw K values $U_1, \cdots, U_K \sim N(0, I)$ and compute weights w_k. For large times $t > 0.1$ estimate score as $\hat{s}(\theta_t, t) = \hat{s}_1(\theta, t)$, otherwise set $\hat{s}(\theta_t, t) = \hat{s}_2(\theta, t)$. Update the point as $\theta_{t-\delta} = \theta_t - \delta(-\theta_t - 2\hat{s}(\theta_t, t)) + \sqrt{2\delta}Z, \quad Z \sim N(0, I)$ and decrease $t = t - \delta$. Finally, return θ_0.

The backward flow SDE bears resemblance to the Follmer drift as discussed in [11] and [6]. The Follmer drift is the solution to the Schrodinger Bridge problem with initial measure $\theta_0 \sim \delta_0$, a point mass at 0, and measure P_1 at time 1 with density $p_1(\theta) \propto e^{f(\theta) - \frac{1}{2}\theta^T\theta}$. The resulting SDE for the Follmer drift in this context is

$$d\theta_t = \frac{E[\nabla f(\theta_t + \sqrt{1-t}U)e^{f(\theta_t + \sqrt{1-t}U)}]}{E[e^{f(\theta_t + \sqrt{1-t}U)}]}dt + dW_t, \quad \theta_0 = 0, U \sim N(0, I). \tag{12}$$

We see much resemblance between (12) and our expression (8). If we change notation, $\tilde{t} = e^{-2t} \in [e^{-2T}, 1]$, then the drift we must estimate in our backward flow SDE is of the form

$$(\sqrt{\tilde{t}})\frac{E[\nabla f(\theta + \sqrt{1-\tilde{t}}U)e^{f(\theta + \sqrt{1-\tilde{t}}U)}]}{E[e^{f(\theta + \sqrt{1-\tilde{t}}U)}]} \quad U \sim N(0, I). \tag{13}$$

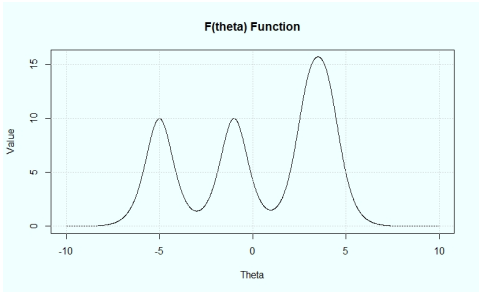
The expression (13) is better suited for a MC estimator than (12) due to the $\sqrt{\tilde{t}}$ in front of the expression which would lower the variance. When $\tilde{t}$ is small estimating the drift in (12) empirically would be difficult due to high variance, but the appearance of $\sqrt{\tilde{t}}$ (or e^{-t} in our notation) outside the fraction brings down the variance of our estimator (10). Conceptually, the Follmer drift starts all points at $\theta_0 = 0$ and gives exact samples at time $t = 1$, while the backward SDE injects initial randomness with $\theta_T \sim N(0, I)$ and has a larger time span $T > 1$. These differences may be what makes the score easier to estimate empirically than the Follmer drift.

One could propose more complex methods such Stein's method [2],[10] or multi-modal sampling methods such as Simulated Tempering [7] to estimate the quantities in (7) and (8), as these represent expectations over un-normalized densities (the denominator is the normalizing constant). However, estimating these values is itself a sub problem of sampling the un-normalized density p_0 in the first place, thus if we employ sufficiently complex methods to sample the sub problems we may have simply employed said methods on the original problem. Additionally, for large $t >> 0$ and small $t \approx 0$ the simple estimators will have small variance, thus it is the middle region of t where it may require more involved estimation of quantities (7) and (8) to keep the estimates accurate.

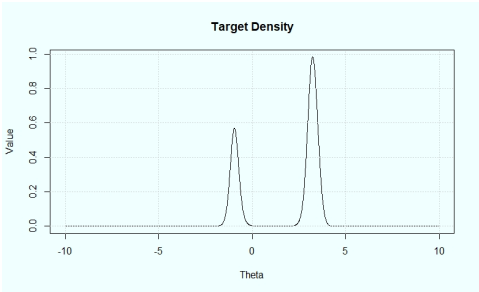
3 Example

The authors are interested in sampling as an approach to train single layer neural networks one neuron at a time in greedy algorithms. That is, with n covariates $X_i \in \mathbb{R}^d$ and response values $Y_i \in \mathbb{R}$, we would like to sample $\theta \sim e^{\alpha \sum_{i=1}^{n} Y_i \tanh(X_i^T\theta) - \frac{1}{2}\theta^T\theta}$ as our target density. To create a toy one dimensional example with similar properties and which exhibits multi modality, consider the function

$$f(\theta) = 100\sum_{i=1}^{4}(\tanh(\theta + 0.05 - \mu_i) - \tanh(\theta - 0.05 - \mu_i)), \mu_1 = -5, \mu_2 = -1, \mu_3 = 3, \mu_4 = 4.$$

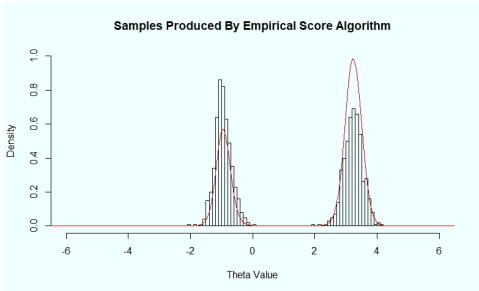


(a) Target function $f(\theta)$

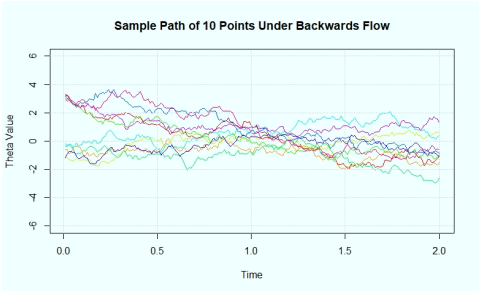


(b) Target density

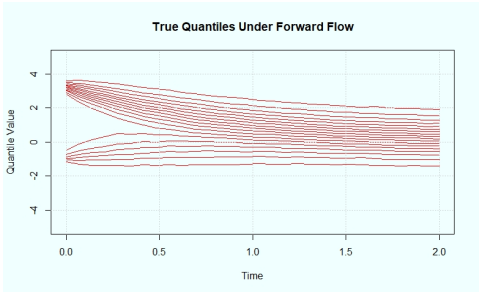
Figure 1: Graphical representations of target density



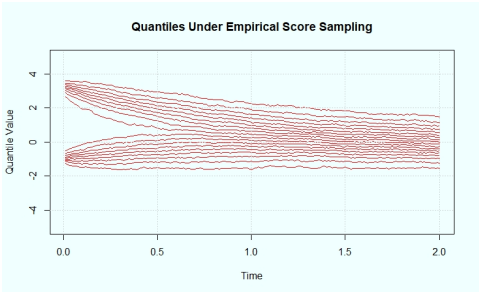
(a) Samples produced by backward flow



(b) Example sample paths



(c) True quantiles under forward flow



(d) Empirical quantiles under backward flow

Figure 2: Output of backward flow with MC estimation of score

The target function $f(\theta)$ and density $e^{f(\theta)-\frac{1}{2}\theta^2}$ are shown in Figure 1. We then run the backward sampling algorithm on 1000 points in parallel with $T = 2, \delta = 0.01, K = 1000$. The results are shown in Figure 2. All points converge to one of the two modes, however the left mode is over sampled thus the algorithm did not converge to the exact target measure. For such a simple model, we are able to simulate the forward flow and compute the quantiles of a number of paths in parallel. These quantiles match closely with those under the backward flow paths.

4 Concluding remarks

This paper presents a score based sampling approach using Monte Carlo estimation of the score and knowledge of the functional form of the target density. This method does not rely on any initial samples from the target density, nor on a model such as a neural network to estimate the score function. Much work remains to be done on the the analysis of such a method, specifically on high dimensional target densities of interest. The MC estimators may be of high variance, and theoretical guarantees on this variance must be provided. Furthermore, the estimates of the score will never be perfect, they will have some standard error around the true value, thus analysis of the backward flow SDE under discretization and approximate drift such as the analysis in [9] will be also be required to guarantee convergence to the correct target measure.

References

[1] Brian DO Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

[2] Wilson Ye Chen, Alessandro Barp, François-Xavier Briol, Jackson Gorham, Mark Girolami, Lester Mackey, and Chris Oates. Stein point markov chain monte carlo. In *International Conference on Machine Learning*, pages 1011–1021. PMLR, 2019.

[3] Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling. *Advances in Neural Information Processing Systems*, 34:17695–17709, 2021.

[4] Tim Dockhorn, Arash Vahdat, and Karsten Kreis. Score-Based Generative Modeling with Critically-Damped Langevin Diffusion. *International Conference on Learning Representations (ICLR)*, 2022.

[5] Arnaud Doucet, Will Grathwohl, Alexander GDG Matthews, and Heiko Strathmann. Score-based diffusion meets annealed importance sampling. *arXiv preprint arXiv:2208.07698*, 2022.

[6] Ronen Eldan and James R Lee. Regularization under diffusion and anticoncentration of the information content. *Duke Mathematical Journal*, 167(5):969–993, 2018.

[7] Rong Ge, Holden Lee, and Andrej Risteski. Simulated tempering langevin monte carlo ii: An improved proof using soft markov chain decomposition. *arXiv preprint arXiv:1812.00793*, 2018.

[8] Cong Huang. *Risk of penalized least squares, greedy selection and L1-penalization for flexible function libraries*. Yale University, 2008.

[9] Holden Lee, Jianfeng Lu, and Yixin Tan. Convergence for score-based generative modeling with polynomial complexity. *arXiv preprint arXiv:2206.06227*, 2022.

[10] Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. *Advances in neural information processing systems*, 29, 2016.

[11] Maxim Raginsky. Sampling using diffusion processes, from langevin to schrödinger, Sep 2021.

[12] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019.

[13] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *International Conference on Learning Representations (ICLR)*, 2020.

5 Appendix

Here we derive the form of the expectations that define the score. Define $\sigma_t = \sqrt{1-e^{-2t}}$, we have

$$
\begin{aligned}
\nabla_{\theta_t} \log p_t(\theta_t) &= \frac{\int p_0(\theta_0) \nabla_{\theta_t} p_t(\theta_t|\theta_0) d\theta_0}{\int p_0(\theta_0) p_t(\theta_t|\theta_0) d\theta_0} \\
&= \frac{\int e^{f(\theta_0) - \frac{1}{2}\theta_0^T\theta_0} \nabla_{\theta_t} e^{-\frac{1}{2\sigma_t^2}(\theta_t - e^{-t}\theta_0)^T(\theta_t - e^{-t}\theta_0)} d\theta_0}{\int e^{f(\theta_0) - \frac{1}{2}\theta_0^T\theta_0} e^{-\frac{1}{2\sigma_t^2}(\theta_t - e^{-t}\theta_0)^T(\theta_t - e^{-t}\theta_0)} d\theta_0} \\
&= \frac{\int e^{f(\theta_0)} \left(-\frac{\theta_t - e^{-t}\theta_0}{\sigma_t^2}\right) e^{-\frac{1}{2}\theta_0^T\theta_0 - \frac{1}{2\sigma_t^2}(\theta_t - e^{-t}\theta_0)^T(\theta_t - e^{-t}\theta_0)} d\theta_0}{\int e^{f(\theta_0)} e^{-\frac{1}{2}\theta_0^T\theta_0 - \frac{1}{2\sigma_t^2}(\theta_t - e^{-t}\theta_0)^T(\theta_t - e^{-t}\theta_0)} d\theta_0}
\end{aligned}
$$

we now perform some completing the square on the θ_0 terms in the exponent to express it as a normal density over θ_0. Note by cancellation on the top and bottom of the fraction, we can ignore any constant terms in the integral and only focus on those terms which relate to θ_0.

$$
\begin{aligned}
& e^{f(\theta_0)-\frac{1}{2}\theta_0^T\theta_0} e^{-\frac{1}{2\sigma_t^2}(\theta_t-e^{-t}\theta_0)^T(\theta_t-e^{-t}\theta_0)} \\
&= e^{f(\theta_0)} e^{-\frac{1}{2}[\theta_0^T\theta_0+\frac{1}{\sigma_t^2}(\theta_t^T\theta_t-2e^{-t}\theta_t^T\theta_0+e^{-2t}\theta_0^T\theta_0)]} \\
&\propto e^{f(\theta_0)} e^{-\frac{1}{2}((1+\frac{e^{-2t}}{1-e^{-2t}})\theta_0^T\theta_0-2\frac{e^{-t}}{1-e^{2t}}\theta_t^T\theta_0)} \\
&\propto e^{f(\theta_0)} e^{-\frac{1}{2\sigma_t^2}(\theta_0-e^{-t}\theta_t)^T(\theta_0-e^{-t}\theta_t)}
\end{aligned}
$$

thus, the above function is proportional to $e^{f(\theta_0)}N(e^{-t}\theta_t, \sigma_t^2 I)$ and we can write the score as

$$
\frac{-E[e^{f(\theta_0)}\frac{\theta_t-e^{-t}\theta_0}{\sigma_t^2}]}{E[e^{f(\theta_0)}]} \quad \theta_0 \sim N(e^{-t}\theta_t, \sigma_t^2 I)
$$

then, consider if instead $U \sim N(0, I)$ and $\theta_0 = \sigma_t U + e^{-t}\theta_t$, we could write the above as

$$
\begin{aligned}
& \frac{-E[e^{f(\sigma_t U+e^{-t}\theta_t)}\frac{\theta_t-e^{-t}\sigma_t U-e^{-2t}\theta_t}{1-e^{-2t}}]}{E[e^{f(\sigma_t U+e^{-t}\theta_t)}]} \\
&= \frac{-E[e^{f(\sigma_t U+e^{-t}\theta_t)}(\theta_t - \frac{e^{-t}}{\sigma_t}U)]}{E[e^{f(\sigma_t U+e^{-t}\theta_t)}]} \\
&= -\theta_t + \frac{e^{-t}}{\sigma_t}\frac{E[Ue^{f(\sigma_t U+e^{-t}\theta_t)}]}{E[e^{f(\sigma_t U+e^{-t}\theta_t)}]}
\end{aligned}
$$

We can also apply integration by parts in the numerator to arrive at an equivalent expression,

$$
\begin{aligned}
\frac{e^{-t}}{\sigma_t}E[Ue^{f(\sigma_t U+e^{-t}\theta_t)}] &= \frac{e^{-t}}{\sigma_t}\int ue^{f(\sigma_t u+e^{-t}\theta_t)}\left(\frac{1}{2\pi}\right)^{\frac{d}{2}} e^{-\frac{1}{2}u^T u}du \\
&= -\frac{e^{-t}}{\sigma_t}\int e^{f(\sigma_t u+e^{-t}\theta_t)}\left(-u\left(\frac{1}{2\pi}\right)^{\frac{d}{2}} e^{-\frac{1}{2}u^T u}\right)du \\
&= -\frac{e^{-t}}{\sigma_t}\int e^{f(\sigma_t u+e^{-t}\theta_t)}\nabla_u\left(\left(\frac{1}{2\pi}\right)^{\frac{d}{2}} e^{-\frac{1}{2}u^T u}\right)du \\
&= \frac{e^{-t}}{\sigma_t}\int \nabla_u\left(e^{f(\sigma_t u+e^{-t}\theta_t)}\right)\left(\frac{1}{2\pi}\right)^{\frac{d}{2}} e^{-\frac{1}{2}u^T u}du \\
&= \frac{e^{-t}}{\sigma_t}\int \sigma_t\nabla f(\sigma_t u+e^{-t}\theta_t)e^{f(\sigma_t u+e^{-t}\theta_t)}\left(\frac{1}{2\pi}\right)^{\frac{d}{2}} e^{-\frac{1}{2}u^T u}du \\
&= e^{-t}E[\nabla f(\sigma_t U+e^{-t}\theta_t)e^{f(\sigma_t U+e^{-t}\theta_t)}]
\end{aligned}
$$

and thus can express the score as

$$
-\theta_t + e^{-t}\frac{E[\nabla f(\sigma_t U+e^{-t}\theta_t)e^{f(\sigma_t U+e^{-t}\theta_t)}]}{E[e^{f(\sigma_t U+e^{-t}\theta_t)}]}
$$