

# TENSOR TRAIN BASED SAMPLING ALGORITHMS FOR APPROXIMATING REGULARIZED WASSERSTEIN PROXIMAL OPERATORS

FUQUN HAN, STANLEY OSHER, AND WUCHEN LI

**ABSTRACT.** We present a tensor train (TT) based algorithm designed for sampling from a target distribution and employ TT approximation to capture the high-dimensional probability density evolution of overdamped Langevin dynamics. This involves utilizing the regularized Wasserstein proximal operator, which exhibits a simple kernel integration formulation, i.e., the softmax formula of the traditional proximal operator. The integration, performed in  $\mathbb{R}^d$ , poses a challenge in practical scenarios, making the algorithm practically implementable only with the aid of TT approximation. In the specific context of Gaussian distributions, we rigorously establish the unbiasedness and linear convergence of our sampling algorithm towards the target distribution. To assess the effectiveness of our proposed methods, we apply them to various scenarios, including Gaussian families, Gaussian mixtures, bimodal distributions, and Bayesian inverse problems in numerical examples. The sampling algorithm exhibits superior accuracy and faster convergence when compared to classical Langevin dynamics-type sampling algorithms.

**Keywords:** tensor train, sampling, wasserstein proximal, bayesian inverse problem

## 1. INTRODUCTION

Over the past decade, obtaining samples from a known and potentially complicated distribution has become increasingly vital in the fields of data science, computational mathematics, and engineering. Sampling algorithms play a central role in many critical real-world applications, including finding global optimizers for a high-dimensional function [20], obtaining samples from the latent space in generative modeling [31], and solving Bayesian inverse problems to estimate the posterior distribution [32]. Efficient and reliable sampling algorithms are essential for the success of the aforementioned applications.

Given the significance of sampling from a known distribution, numerous intriguing algorithms have been proposed and analyzed. Among them, Markov Chain Monte Carlo (MCMC) type algorithms have been the most popular due to their simple formulation and intrinsic diffusion resulting from the adoption of Gaussian noise, which is desirable. Representative MCMC type algorithms include Metropolis random walk [23], hit and run [2] which are zero-order methods, and unadjusted Langevin [26], Hamiltonian Monte Carlo [21], Metropolis-adjusted Langevin algorithms [38], which are first-order methods using Langevin diffusions. For a more in-depth review and details for Monte Carlo type algorithms, one may refer to [3] and references therein. Langevin-type MCMC methods usually involve evaluating the gradient of the potential function and adding a Gaussian noise to achieve diffusion. Several theoretical results showed that they can converge to the stationary distribution under proper assumptions [10, 11].

Classical Langevin dynamics-type sampling algorithms may exhibit slow convergence, particularly for complex and high-dimensional distributions. As an alternative approach, generating diffusion using the score function, defined as the logarithm of the density function, offers a promising direction for sampling algorithms. In [31], the author demonstrates that score-based diffusion minimizes the Kullback–Leibler divergence (KL divergence) between the target and generated distribution, a perspective that can also be interpreted as an entropy-regularized optimal transport problem [16]. Score-based diffusion has already showcased impressive numerical performance in various domains, including image generation [7], medical inverse problems [4], and importance sampling [9]. Despite these successes, a notable challenge in many score

---

F. Han and S. Osher are partially funded by AFOSR MURI FA9550-18-502 and ONR N00014-20-1-2787.

W. Li's work is partially supported by AFOSR YIP award No.FA9550-23-1-008, NSF DMS-2245097, and NSF RTG: 2038080.

function based algorithms lies in efficiently approximating the score function, stemming from the inherent complexity associated with solving the density function.

Recently, in [33], a new score function based sampling algorithm via the backward Wasserstein proximal operator (BRWP) is proposed. In that work, the authors approximate the score function by considering a regularized Wasserstein proximal, which can be shown to have a kernel formula through the Hopf-Cole transform. With the explicit form for the computation of the score function, the sampling algorithm becomes deterministic and relatively robust. Numerical experiments show that the evolution of samples exhibits a highly structured manner and a reliable convergence is guaranteed for several interesting scenarios. However, as the author pointed out in [33], the algorithm is biased due to the discretization of the particle evolution equation and is not easily scalable to high-dimensional spatial domains.

To address the curse of dimensionality inherent in high-dimensional sampling problems, many strategies have been employed. These approaches encompass the investigation of low-rank structures [5, 36], the utilization of accelerated gradient flow techniques [19, 37], and the incorporation of deterministic sparse quadrature rules tailored for special cases [30, 12]. In this work, we shall leverage efficient and delicate tensor algorithms to improve both theoretical results and the numerical performance of BRWP. Tensor train approximation, which can represent a broad class of functions with storage complexity  $\mathcal{O}(d)$  and allows many algebraic computations of order  $\mathcal{O}(d)$ , where  $d$  is the dimension, has become popular in recent years. Tensor train approximations and algorithms, also called matrix product space [35], are populated in [25] and find applications in neural network representation [34], achieving image super-resolution [8], high-dimensional PDE simulation [28], generative modeling [15], and so on. In this paper, we fully harness the power of tensor train approximation to derive proper tensor sampling algorithms that are computationally efficient and accurate for high-dimensional distributions.

The proposed sampling algorithm presents several distinctive features. Firstly, by embracing tensor train approximation for high-dimensional integration kernels, the algorithm generates samples that align more accurately with theoretical results than the Monte Carlo integration used in BRWP. Secondly, the paper introduces a delicate choice of covariance matrix for initial density estimation through the discretization of the density function on a high-dimensional mesh, leading to an unbiased sampling algorithm. Lastly, diffusion which is generated by a deterministic approximation procedure of the score function endows the proposed method with great robustness for highly ill-posed Bayesian inverse problems.

The paper is structured as follows. Section 2 provides an insightful review of the score function based sampling algorithm focusing on the BRWP proposed in [33] and emphasizing the kernel representation of the solution for the regularized Wasserstein proximal operator. Section 3 will introduce the fundamental concept of tensor train approximation and will present several important motivations for the adoption of tensor train approximation for our objective. Section 4 intricately details the proposed tensor train based sampling algorithm, offering comprehensive insights into its implementation and convergence analysis across critical scenarios. Finally, Section 5 presents a series of numerical experiments to validate the proposed sampling algorithm's claimed features and provide a robust empirical foundation for its efficacy in real-world applications, such as Bayesian inverse problems.

## 2. PROBLEM STATEMENT AND SCORE-BASED SAMPLING ALGORITHM

In this section, our main goal is to present the definition of the score function and outline an approximation problem for a regularized Wasserstein proximal operator. The kernel formula for this operator is key to the derivation of our sampling algorithm. Following this, we provide a brief description of the Backward Regularized Wasserstein Proximal Scheme (BRWP) proposed in [33], which serves as motivation for the current work.

Our specific objective is to acquire a series of samples  $\{x_{k,j}\}_{j=1}^N$  from a distribution  $\Pi_k$ , approximating the target distribution  $\Pi^*$  with a density function

$$\rho^*(x) = \frac{1}{Z} \exp(-V(x)), \quad (1)$$

where  $V(x)$  is a known continuously differentiable potential function, and  $Z = \int_{\mathbb{R}^d} \exp(-V(y)) dy < +\infty$  is a normalization constant.

Score-based diffusion concerns the evolution of particles with density function  $\rho$ , following the Langevin stochastic differential equation

$$dX(t) = -\nabla V(X(t))dt + \sqrt{2/\beta}dW(t),, \quad X(0) = X_0,, \quad (2)$$

where  $\beta > 0$  is a constant,  $W(t)$  is the standard Wiener process in  $\mathbb{R}^d$ , and  $X_0$  is the initial set of particles.

With the help of Louisville's equation [22], we derive the scored-based particle evolution equation whose trajectories have the same marginal distribution as (2)

$$\frac{dX}{dt} = -\nabla V(X) - \beta^{-1}\nabla \log \rho(t, X),, \quad (3)$$

where  $\nabla \log \rho(t, X)$  is the score function associated with density  $\rho$ .

To compute the density function, it is known that  $\rho$  will evolve following the Fokker-Planck equation

$$\frac{\partial \rho}{\partial t} = \nabla \cdot (\nabla V(x)\rho) + \beta^{-1}\Delta \rho, \quad \rho(0, x) = \rho_0(x), \quad (4)$$

for the initial density  $\rho_0$ .

Approximating the terminal density  $\rho_T := \rho(T, \cdot)$  with terminal time  $T$  is challenging due to non-linearity and high dimensions. To address this, we consider a kernel formula to approximate a regularized Wasserstein operator. Specifically, we first recall the Wasserstein proximal with linear energy

$$\rho_T = \arg \min_q \left[ \frac{1}{2T} W(\rho_0, q)^2 + \int_{\mathbb{R}^d} V(x)q(x)dx \right], \quad (5)$$

where  $W(\rho_0, q)$  is the Wasserstein-2 distance between  $\rho_0$  and  $q$ . Additionally, by the Benamou-Brenier formula, the Wasserstein-2 distance can be expressed as an optimal transport problem, leading to

$$\frac{W(\rho_0, q)^2}{2T} = \inf_{\rho, v, \rho_T} \int_0^T \int_{\mathbb{R}^d} \frac{1}{2} \|v(t, x)\|_2^2 \rho(t, x) dx dt,$$

where the minimizer is taken over all vector fields  $v : [0, T] \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ , density functions  $\rho : [0, T] \times \mathbb{R}^d \rightarrow \mathbb{R}$ , such that

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho v) = 0, \quad \rho(0, x) = \rho_0(x), \quad \rho(T, x) = q(x).$$

Solving (5) is usually a challenging optimization problem. In [18], motivated by Schrödinger bridge systems, the authors introduce a regularized Wasserstein proximal operator by adding a Laplacian regularization term, leading to

$$\rho_T = \arg \min_{\rho} \inf_{v, q} \int_0^T \int_{\mathbb{R}^d} \frac{1}{2} \|v(t, x)\|_2^2 \rho(t, x) dx dt + \int_{\mathbb{R}^d} V(x)q(x) dx, \quad (6)$$

with

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho v) = \beta^{-1}\Delta \rho,, \quad \rho(0, x) = \rho_0(x), \quad \rho(T, x) = q(x), \quad (7)$$

where  $\rho(t, x)$  as the solution to the above regularized Wasserstein proximal operator approximates one step of the Fokker-Planck equation when  $T$  is small.

Introducing a Lagrange multiplier function  $\Phi$ , we find that solving  $\rho_T$  is equivalent to computing the solution of the coupled PDEs

$$\begin{cases} \partial_t \rho + \nabla_x \cdot (\rho \nabla_x \Phi) = \beta^{-1}\Delta_x \rho, \\ \partial_t \Phi + \frac{1}{2} \|\nabla_x \Phi\|^2 = -\beta^{-1}\Delta_x \Phi, \\ \rho(0, x) = \rho_0(x), \quad \Phi(T, x) = -V(x). \end{cases} \quad (8)$$

The coupled PDEs can be solved using the Hopf-Cole type transform, and then  $\rho(t, x)$  can be computed by solving a system of backward-forward heat equations

$$\rho(t, x) = \eta(t, x)\hat{\eta}(t, x), \quad (9)$$

where  $\hat{\eta}$  and  $\eta$  satisfy

$$\partial_t \hat{\eta}(t, x) = \beta^{-1} \Delta_x \hat{\eta}(t, x), \quad (10)$$

$$\partial_t \eta(t, x) = -\beta^{-1} \Delta_x \eta(t, x), \quad (11)$$

$$\eta(0, x)\hat{\eta}(0, x) = \rho_0(x), \quad \eta(T, x) = \exp(-\beta V(x)). \quad (12)$$

In summary, equation (9) provides a closed-form update to compute the density function by solving a system of backward-forward heat equations. Once  $\rho_T$  is known, by the Euler discretization for equation (3), the evolution of particles in the next iteration can be computed as

$$X_{k+1} = X_k - h(\nabla V(X_k) + \beta^{-1} \nabla \log(\rho_T(X_k))). \quad (13)$$

where  $h$  is the step size and  $\rho_T$  is the terminal density with initial distribution  $\Pi_k$ .

For a comprehensive understanding of our upcoming discussion, we recall the key steps in the BRWP algorithm developed in [33]. This algorithm estimates the initial density  $\rho_0$  by empirical distribution, computes (9) by convolution with a heat kernel as

$$\rho_T(x) = \int_{\mathbb{R}^d} \frac{\exp\left[-\frac{\beta}{2}\left(V(x) + \frac{\|x-y\|_2^2}{2T}\right)\right]}{\int_{\mathbb{R}^d} \exp\left[-\frac{\beta}{2}\left(V(z) + \frac{\|z-y\|_2^2}{2T}\right)\right] dz} \rho_0(y) dy \quad (14)$$

and generates new samples by (13). In particular, the integral in  $\mathbb{R}^d$  that appeared in (14) is estimated using Monte-Carlo integration.

### 3. REVIEW ON TENSOR TRAIN APPROXIMATION AND ALGORITHMS

In this section, we will provide a concise overview of the definition, algorithms, and convergence properties of tensor train (TT) approximation applied to several crucial classes of multivariate functions of interest. The general objective of TT approximation is to reduce the computational complexity and improve the robustness of the sampling algorithms that we will introduce in the subsequent sections.

The TT decomposition  $f_{TT}$  of a  $d$ -dimensional tensor  $f : \mathbb{R}^d \rightarrow \mathbb{R}$  is defined as follows:

$$f(x_1, \dots, x_d) \approx f_{TT}(x_1, \dots, x_d) = \sum_{\alpha_1=1}^{r_1} \cdots \sum_{\alpha_d=1}^{r_d} g_1(x_1, \alpha_1) g_2(\alpha_1, x_2, \alpha_2) g_3(\alpha_2, x_3, \alpha_3) \cdots g_d(\alpha_{d-1}, x_d), \quad (15)$$

where  $r = \max\{r_j\}_{j=1}^d$  is the rank of the decomposition, and the functions  $g_i$  are cores of the TT decomposition. We remark (15) can be generalized to functional approximation in  $\mathbb{R}^d$ .

The TT representation of a high-dimensional tensor enables efficient computation of various algorithms, including addition, Hadamard product, and matrix-vector products, especially when the rank  $r$  is relatively small. Assuming the number of nodal points in each dimension is  $n$ , then the complexity of several frequently used algorithms for tensors in TT format is summarized in the following table [25]:

Operation	Rounding	Addition	Hadamard Product	Matrix-Vector Product
Complexity	$\mathcal{O}(dnr^3)$	$\mathcal{O}(dnr^3)$	$\mathcal{O}(dnr^3)$	$\mathcal{O}(dn^2r^4)$

To use TT algorithms effectively, it is crucial to determine the class of functions for which the TT representation (15) exists with a relatively small rank  $r$ . Since the TT decomposition can be viewed as a recursive singular value decomposition, we recall the following theorem from [14] based on SVD if  $f$  is a  $d$ -dimensional tensor.

**Theorem 3.1.** [14] *Let  $f \in H^k(\mathbb{R}^d)$  for some fixed  $k > 0$  and  $0 < \epsilon < 1$ . Then, the overall truncation error of the TT decomposition for  $f$  with ranks  $r \leq \epsilon^{-1/k}$  is given by*

$$\|f - f_{TT}\|_{L^2(\mathbb{R}^d)} \leq \sqrt{d}\epsilon,$$

and the storage cost for TT representation will be  $\epsilon^{-d/k}$ .

The above theorem is applicable since, as we will see later, for sampling problems, functions we approximate in TT format are the exponential of potential functions or simply the heat kernel, which is typically very smooth. In such cases,  $k$  can be relatively large, resulting in small ranks for TT approximation and low storage complexity for a fixed  $d$ .

Moreover, it is noteworthy that the estimate in Theorem 3.1 remains slightly unsatisfactory, considering that the storage cost and approximation error still depend on  $\sqrt{d}$  or  $\epsilon^{-d/k}$ , which might be expensive for high-dimensional scenarios. For an alternative approximation result, specifically when  $f$  represents the density function of a Gaussian distribution, we refer to [29] which presents a convergence theorem that is independent of dimension and relies on the off-diagonal ranks of the variance matrix.

The next issue we need to address is the algorithm to obtain a proper TT decomposition efficiently for a given tensor. Based on our numerical experience, we found that the TT crossing algorithm in [24] is the most efficient and stable one for our purpose. For a discretized tensor defined on  $(\mathbb{R}^n)^d$ , where  $n$  is the number of nodal points in each dimension, the general idea of TT crossing-type algorithms is to unfold the tensor into a matrix  $A_1$  of the form  $\mathbb{R}^n \times \mathbb{R}^{n(d-1)}$ , called the unfolding matrix. The standard crossing approximation is then applied on  $A_1$  to have

$$A_1 \approx C\hat{A}^{-1}R,$$

with

$$C = A_1(:, I_1), \quad R = A_1(J_1, :),$$

where  $I_1$  and  $J_1$  are small index sets chosen by certain energy minimization algorithms. This process is repeated for each dimension. Notably, crossing algorithms only require access to a small number of entries of the original tensor, and the full tensor is never formed in the algorithm.

For the accuracy of the TT crossing algorithm, the following error bound of TT approximation in the Frobenius norm for  $f$  discretized on  $n$  points in each dimension is established in [27].

**Theorem 3.2.** [27] *Suppose a tensor  $f$  can be approximated by a tensor train with maximum rank  $r$  and error  $\epsilon$ . Using the crossing algorithm with tolerance  $\epsilon$ , we can find  $f_{TT}$  with rank at most  $r$  such that*

$$\|f - f_{TT}\|_F \leq \frac{(3\kappa)^{\log_2 d} - 1}{3\kappa - 1}(r + 1)\epsilon,$$

where  $\kappa$  is the condition number of the unfolding matrix of  $f$ .

**3.1. A Straightforward Tensor Train Algorithm to Improve BRWP.** In this subsection section, we initially propose a straightforward sampling method that leverages TT algorithms discussed in the preceding subsection. This method is designed to compute high-dimensional integrals appearing in (14). The error analysis of this algorithm offers valuable insights into the power of TT approximations in approximating the kernel formula, paving the way for the development of the new algorithm to be presented in the next section.

In the algorithm described in [33], for each iteration, it requires the computation of the density  $\rho_T$  using the following formula

$$\rho_T(x_j) = \frac{1}{N} \sum_{i=1}^N \frac{\exp[-\beta/2 (V(x_i) + \|x_i - x_j\|_2^2/(2T))]}{\int_{\mathbb{R}^d} \exp[-\beta/2 (V(z) + \|x_i - z\|_2^2/(2T))] dz}, \quad (16)$$

for each particle located at  $x_j$ .

We observe that the bottleneck of computation in (16) at each iteration lies in the normalization term

$$N(x) := \int_{\mathbb{R}^d} \exp \left[ -\frac{\beta}{2} \left( V(z) + \frac{\|x - z\|_2^2}{2T} \right) \right] dz, \quad (17)$$

which involves integration in  $\mathbb{R}^d$ . Monte Carlo integration through random sampling can only provide a solution that converges at a rate of  $N^{-1/2}$ , where  $N$  is the number of random points used in the integration. It is important to note that the  $N^{-1/2}$  error bound does not imply that the computational complexity to achieve a fixed degree of accuracy is the same for any dimension, as the complexity of evaluating a  $d$ -dimensional function  $f$  will also depend on  $d$ . For certain scenarios, as we will present in the following table 3.1, the convergence could be extremely slow for some interesting high-dimensional integrals that arise in sampling problems.

In this case, we propose to apply TT approximation to improve both the accuracy and efficiency of the computation of the normalization term in (17). We first consider  $n$  quadrature points  $\{z_j\}_{j=1}^n \in [-L, L]$  in each dimension. The mesh formulated by  $\otimes\{z_j\}$  in  $[-L, L]^d$  is denoted as  $\mathcal{Z}_{d,n}$ . The integration then becomes

$$N(x) \approx K_d \mathcal{T}(\exp(-\beta V/2))(x), \quad (18)$$

where  $\mathcal{T}(f)$  is the tensor train approximation using the crossing algorithm for  $f$  on the mesh  $\mathcal{Z}_{d,n}$ , and  $K_d$  is defined as

$$K_d(x, y) = K_1(x, y) \otimes \cdots \otimes K_1(x, y), \quad (19)$$

with

$$(K_1(x, y))_{j,k} = w_j \exp \left( -\beta \frac{\|z_j - z_k\|_2^2}{4T} \right),$$

and  $w_j$  being quadrature weights for  $z_j$ . The error term  $\epsilon$  in the approximation (18) consists of the truncation of integration onto  $[-L, L]^d$ , which is small due to the exponential decay of  $\exp(-\beta\|x - z\|_2^2/4T)$  for  $z$ , and the quadrature error, which will be of order  $n^{-n}$  if  $z_j$  is Legendre quadrature points.

Then, the computation of the density function  $\rho_T$  becomes:

$$\rho_T(x_j) = \int_{\mathbb{R}^d} G(x_j, y) \rho_0(y) dy, \quad G(x, y) = \mathcal{T}(\exp(-\beta V/2))(x_j) \frac{K_d(x_j, y)}{N(y)}. \quad (20)$$

Combining Theorems 3.1 and 3.2, we observe that the score function

$$\nabla \log(\rho_T)(x_j) = \frac{\nabla \rho_T(x_j)}{\rho_T(x_j)}$$

can be computed efficiently and accurately when  $\exp(-\beta V(x)/2)$  is sufficiently smooth and has relatively small TT ranks. Before we proceed, we would like to provide an example to demonstrate the improvement of accuracy and efficiency of approximating the integral (17) for a special case where  $\rho_0 = 1$  and  $N(y) = 1$ . In this case, the function  $\nabla \log(\rho_T)$  can be explicitly written out for some cases by computing the proximal of  $V(x)$  directly. We pick a challenging non-convex potential function  $V(x) = |x|_{1/2}^2$ , the computational time and accuracy using TT integration and MC integration are summarized in the following tables.

	TT Integration	MC Integration with	
		10 <sup>3</sup> Samples	10 <sup>6</sup> Samples
Error	0.099	0.224	0.190
Time	5	< 1	85

TABLE 1. Relative error and time for different numbers of samples in seconds.

$d$	TT Integration	MC Integration
10	0.27	31.48
100	6.29	277.81
200	16.53	2186.5

TABLE 2. Computational time in seconds among different dimensions  $d$

In summary, TT integration demonstrates substantial advantages in terms of accuracy and efficiency when compared to MC integration, especially for some high-dimensional integrations.

Given the improved accuracy and efficiency of using the tensor train to compute integral as in (18), we present the following theorem that rigorously demonstrates the improvement in the accuracy of the sampling algorithm. This is done for a representative scenario where both  $\Pi_k$  and  $\Pi^*$  are Gaussian distributions. For the notional sake, we will write  $f(n, d) \lesssim g(n, d)$  when  $f(n, d) \leq Cg(n, d)$  for a positive constant  $C$  independent of  $n$  and  $d$ .

**Theorem 3.3.** *For BRWP in [33], let  $V(x) = x^\top x / (2\sigma^2)$ , and denote  $X_k$  and  $\tilde{X}_k$  be samples obtained at the  $k$ -th iteration with TT integration and MC iteration with means  $\mu_k$  and  $\tilde{\mu}_k$  respectively. Assuming all steps in the BRWP Algorithm are exact except the step of approximating  $N(x)$  in (17). If the complexities of the two numerical integration methods are of the same order, and let  $\mu_\infty = \lim_{k \rightarrow \infty} \mu_k$ ,  $\tilde{\mu}_\infty = \lim_{k \rightarrow \infty} \tilde{\mu}_k$ , we have*

$$|\mu_\infty - \mu^*| \lesssim \frac{d}{2^n}, \quad |\tilde{\mu}_\infty - \mu^*| \lesssim \frac{1}{d^{1/2n}},$$

where  $n$  represents the number of discretization points in each dimension for TT integration.

*Proof.* From [33] and our analysis in the next section, under exact arithmetic,  $\mu_\infty$  and  $\tilde{\mu}_\infty$  all converge to  $\mu^*$ . Hence, we shall examine the numerical errors generated by the two different integration methods when approximating  $N(y)$  defined in (17).

Let  $N_1(y)$  and  $N_2(y)$  be the approximations to  $N(y)$  by TT integration and MC integration, respectively. We have

$$|N_2(y) - N(y)| \lesssim n_{mc}^{-d/2},$$

where  $n_{mc}^d$  is the number of samples used in MC integration, involving  $\mathcal{O}(n_{mc}^d)$  flops.

For  $N_1$ , considering  $n$  quadrature points in each dimension and truncating the integration to  $[-L, L]^d$ , then the error between  $N_1(y)$  and  $N(y)$  consists of two parts: one from truncation of integration to a bounded domain denoted as  $E_1$ , and the other from the quadrature rule denoted as  $E_2$ .

For  $E_1$ , using the upper bound for the error function of the Gaussian distribution, we have

$$E_1 \lesssim d \frac{\exp(-\beta L_z^2 / (4\sigma^2))}{L_z}.$$

For  $E_2$ , recalling the classical error bound of the Gauss-Legendre quadrature, we derive

$$E_2 \leq d \frac{(2L_z)^{2n+1} (n!)^4 \|H_{2n}\|_\infty}{((2n)!)^3 (4\beta^{-1}\sigma^2)^{2n}},$$

where  $H_{2n}$  is the Hermitian polynomial coming from the  $n$ -th derivative of the Gaussian. Stirling's approximation for  $n!$  and the upper bound of Hermitian polynomials yield  $E_2 \lesssim \frac{d}{2^n}$ . Thus,  $|N_1(y) - N(y)| \lesssim \frac{d}{2^n}$ .

Assuming  $P$  and  $\tilde{P}$  are computed with the same order of flops, we have  $n_{mc} = (n^2 d)^{1/d}$ , and the error for MC integration becomes

$$|N_2(y) - N(y)| \lesssim \frac{1}{nd^{1/2}}.$$

Finally, recalling the iterative relation in [33], the numerical error in approximating the normalization term  $N(x)$  will propagate to all the remaining iterations. Hence, the error in the iteration at infinity will be bounded by the same term.  $\square$

To conclude, according to Theorem 3.3, if we employ TT integration to approximate the normalization term (17), the computational accuracy will experience a significant improvement while maintaining the same order of computational complexity compared with MC integration. This theorem underscores the necessity of applying TT approximation, and the introduced straightforward algorithm becomes particularly useful when  $T$  is extremely small, and an empirical distribution for  $\rho_0$  is acceptable. However, as revealed in [33], the aforementioned algorithm is still biased, meaning that the steady state of the generated distribution differs from the target distribution. Therefore, in the next section, we will propose an unbiased new sampling algorithm with the assistance of TT approximation.

## 4. TENSOR TRAIN BASED NOISE-FREE SAMPLING ALGORITHM (TT-BRWP)

In this section, we propose a new sampling algorithm with the help of a TT algorithm that utilizes a delicate choice of covariance matrix to enhance the accuracy of the BRWP proposed in [33]. We verify the convergence and mixing time of the proposed algorithm in a representative scenario that the distribution we would like to sample is a general Gaussian distribution.

We recall that to compute the terminal density function in BRWP, the empirical distribution is employed to estimate initial density  $\rho_0$ . However, this choice, along with the discretization of the particle evolution equation, leads to biased estimations of the sample mean and variance [33]. Given that, we choose to use the following density estimation which is only implementable when the TT algorithm is employed to approximate  $\rho_0$  as

$$\rho_0(y) = \frac{1}{M} \sum_{k=1}^M \frac{1}{|H|^{1/2} (2\pi)^{d/2}} \exp\left(-\frac{1}{2}(y-x_k)^\top H^{-1}(y-x_k)\right), \quad (21)$$

with a special choice of the covariance matrix  $H$ . In subsequent analysis, we will derive and justify the explicit choice of  $H$ , which depends on the sample covariance and parameters  $\beta, T$ .

We remark that, heuristically, solving  $\rho$  with our special choice of  $\rho_0$  can be considered as solving a preconditioned version of the Fokker-Planck equation satisfied by the density function. From our numerical experience and the following analysis of several representative scenarios, this preconditioned equation leads to a sampling algorithm with faster convergence and unbiased estimation. The more complete theoretical treatment will be addressed in future works.

Before the analysis, we first present the TT-BRWP algorithm as follows.

---

**Algorithm 1** Tensor train BRWP sampling algorithm (TT-BRWP)

---

- 1: **Input:** Given  $X_0 = \{x_{0,j}\}_{j=1}^M \in \mathbb{R}^d \sim \Pi_0$ .
  - 2: **Construction:** Solve backward heat equation by (18) and use TT crossing to obtain TT approximation for  $1/\eta_0$  in (9).
  - 3: **for**  $k = 1, 2, \dots$  Number of iterations **do**
  - 4:   **for**  $j = 1, 2, \dots$  Number of samples (parallel) **do**
  - 5:     **Construct:** Write  $\rho_{0,j}^*$  as in (21), which is a tensor train of rank 1.
  - 6:     **Compute**  $\hat{\eta}_0 = \rho_{0,j}^*/\eta_0$  by Hadamard product.
  - 7:     **Solve** forward heat equation with  $\hat{\eta}_0$  to get  $\hat{\eta}_T$  and  $\nabla \hat{\eta}_T$ .
  - 8:     **Compute**  $\rho_{T,j} = \eta_T \circ \hat{\eta}_T$  and  $\nabla \rho_{T,j} = \eta_T \circ \nabla \hat{\eta}_T$ .
  - 9:   **end for**
  - 10:   **Interpolate**  $\nabla \rho_T = \sum_j \nabla \rho_{T,j}$  and  $\rho_T = \sum_j \rho_{T,j}$  on  $X_k$  (interpolation on a gridded array).
  - 11:   **Compute** score function  $\nabla \log \rho_T = \nabla \rho_T / \rho_T$ .
  - 12:   **Solve** (3) using backward Euler scheme to have  $X_{k+1} = X_k - h(\nabla V(X_k) + \beta^{-1} \nabla \log(\rho_T(X_k)))$ .
  - 13: **end for**
  - 14: **Output:**  $X_k = \{x_{k,j}\}_{j=1}^M$  for  $k \in \mathbb{N}$ .
- 

In the following, section 4.1 and 4.2 will verify Algorithm 1 by computing its mixing time and deriving continuous analog to show the convergence of the generated samples to a desired stationary distribution when the underlying distribution is assumed to be a Gaussian. In section 4.3, details of numerical implementation will be addressed.

**4.1. Analysis of TT-BRWP for Gaussian Distribution.** We first focus on the verification of Algorithm 1 when the target distribution is a multivariate Gaussian distribution. We then derive the convergence and mixing time of the proposed algorithm.

Let us consider the case that the density function for the target distribution is

$$\rho^*(x) = \frac{1}{Z} \exp\left(-\frac{(x-\mu)^\top \Sigma^{-1}(x-\mu)}{2}\right)$$

which is a multivariate Gaussian distribution. Then the solution of the regularized Wasserstein proximal defined in (8) can be written as [18]

$$\rho_T(x) = \int_{\mathbb{R}^d} \rho_0(y) \frac{\exp\left(-\frac{\beta}{2}\left(\frac{\|x-y\|_2^2}{2T} + (x-\mu)^\top \Sigma^{-1}(x-\mu)\right)\right)}{N(y)} dy, \quad (22)$$

where

$$N(y) = \int_{\mathbb{R}^d} \exp\left(-\frac{\beta}{4}(z-\mu)^\top \Sigma^{-1}(z-\mu)\right) \exp\left(-\beta \frac{\|z-y\|_2^2}{4T}\right) dz. \quad (23)$$

Next, we initialize the algorithm by letting  $\Pi_0$  be a Gaussian distribution with mean  $\mu_0$  and covariance  $\Sigma_0$ , which is a practically common choice for real applications. Moreover, we assume that we have sufficiently many random samples from  $\Pi_0$  so that  $\rho_0$  in (21) can be approximated by a continuous convolution.

Now, recall the fact that the convolution of two multivariate Gaussians with mean and covariance  $(\mu_1, \Sigma_1)$ ,  $(\mu_2, \Sigma_2)$  will still be a Gaussian with mean and covariance  $(\mu_1 + \mu_2, \Sigma_1 + \Sigma_2)$ . Hence, we have

$$N(y) \sim \exp\left(-\frac{\beta}{4}(y-\mu)^\top (\Sigma + T)^{-1}(y-\mu)\right), \quad \rho_0(y) \sim \exp\left(-\frac{1}{2}(y-\mu_0)^\top (H + \Sigma_0)^{-1}(y-\mu_0)\right),$$

where  $f(x) \sim g(x)$  denotes  $f(x) = Cg(x)$  for some constants  $C$  that are independent of  $x$ . Substituting the above two expressions into (22), we derive

$$\begin{aligned} \rho_T(x) &\sim \exp\left(-\beta \frac{(x-\mu)^\top \Sigma^{-1}(x-\mu)}{4}\right) \\ &\int_{\mathbb{R}^d} \exp\left(-\beta \frac{\|x-y\|_2^2/T - (y-\mu)^\top (\Sigma + T)^{-1}(y-\mu)}{4} - \frac{(y-\mu_0)^\top (H + \Sigma_0)^{-1}(y-\mu_0)}{2}\right) dy \\ &\sim \exp\left(-\frac{\beta}{2}(x-\mu_{0,T})^\top \Sigma_{0,T}^{-1}(x-\mu_{0,T})\right) \end{aligned} \quad (24)$$

where

$$\mu_{0,T} = K\mu + \frac{\beta}{2}K(H + \Sigma_0)^{-1}K\mu, \quad \Sigma_{0,T} = 2\beta^{-1}TK + K^\top(H_0 + \Sigma_0)^{-1}K, \quad (25)$$

and  $K = (I + T\Sigma^{-1})^{-1}$ . Consequently, we employ the discretization of particle evolution equation (13) to have

$$X_1 = X_0 - h\nabla V(X_0) - h\beta^{-1}\nabla \log \rho_T = (1 - h\Sigma^{-1} + h\beta^{-1}\Sigma_{k,T})X_0 - h\beta^{-1}\Sigma_{k,T}^{-1}\mu_{k,T}. \quad (26)$$

Hence, it will be clear that  $X_1$  and also all  $X_k$  will be with Gaussian distributions. Moreover, their means and covariances can be explicitly computed which is presented in the following Lemma similar to proposition 2 in [33].

**Lemma 4.1.** *Assuming  $V(x) = (x-\mu)^\top \Sigma^{-1}(x-\mu)/2$  and  $X_k$  is a Gaussian distribution with mean  $\mu_k$  and covariance  $\Sigma_k$ , then  $X_{k+1}$  will also be a Gaussian with mean and covariance as*

$$\mu_{k+1} = (I - h\Sigma^{-1} + h\beta^{-1}\Sigma_{k,T})\mu_k - h\beta^{-1}\Sigma_{k,T}^{-1}\mu_{k,T} \quad (27)$$

and

$$\Sigma_{k+1} = (I - h\Sigma^{-1} + h\beta^{-1}\Sigma_{k,T}^{-1})\Sigma_{k,T}(I - h\Sigma^{-1} + h\beta^{-1}\Sigma_{k,T}^{-1}) \quad (28)$$

where  $\mu_{k,T}$  and  $\Sigma_{k,T}$  are given in (25) by replacing  $\Sigma_0$  with  $\Sigma_k$ .

Now, we are ready to introduce our choice of the covariance matrix  $H_k$  in (21) at the  $k$ -th iteration, which will be

$$H_k = \beta^{-1}(I - T\Sigma_k^{-1})\Sigma_k(I + T\Sigma_k^{-1}) - \Sigma_k = \beta^{-1}(\Sigma_k - T^2\Sigma_k^{-1}) - \Sigma_k, \quad (29)$$

where  $\Sigma_k$  is the covariance matrix for particles  $X_k$  in the  $k$ -th iteration. The reason we choose  $H_k$  is from (28) as the steady state of covariance denoted as  $\Sigma_\infty$  will satisfy

$$I + h(\beta^{-1}\Sigma_{\infty,T}^{-1} - \Sigma^{-1}) = I \Rightarrow \Sigma - T^2\Sigma^{-1} = \Sigma_\infty - T^2\Sigma_\infty^{-1}. \quad (30)$$

Hence, in this scenario, the covariance of the target distribution  $\Sigma$  will also be a steady state for the evolution of  $X_k$ . We remark that to allow  $H_k$  to be positive definite, we shall choose  $\beta < 1$  and  $T$  to be sufficiently small.

To better understand the convergence of  $\Sigma_k$  and  $\mu_k$ , we first consider the simplified case where  $\Sigma_0$  and  $\Sigma$  commute, indicating they share the same eigenspace. Then from (28), we observe that  $\Sigma_k$  also commutes with  $\Sigma$  for all  $k$  in this case.

We introduce the notations  $\sigma^{(j)}$  and  $\sigma_k^{(j)}$  as the  $j$ -th eigenvalue of  $\Sigma$  and  $\Sigma_k$  respectively. Moreover, we write  $m_k^{(j)}$  as the  $j$ -th entry in  $\mu_k$ . In the following, we will skip the superscript if the result holds for all  $j$ . By simplifying Lemma 4.1, the evolution of  $m_k$  and  $\sigma_k^2$  can be simplified as follows

$$m_{k+1} = \left( 1 - \sigma^{-2}h + \frac{h\sigma^{-2}T(1 + \sigma^{-2}T)}{\sigma_k^2 - T^2\sigma_k^{-2} + 2T(1 + \sigma^{-2}T)} \right) m_k, \quad (31)$$

$$\sigma_{k+1}^2 = \left( 1 - \sigma^{-2}h + \frac{h(1 + \sigma^{-2}T)^2}{\sigma_k^2 - T^2\sigma_k^{-2} + 2T(1 + \sigma^{-2}T)} \right)^2 \sigma_k^2, \quad (32)$$

and the equation for the steady state  $\sigma_\infty$  in (30) simplifies to

$$\sigma^2 - T^2\sigma^{-2} = \sigma_\infty^2 - T^2\sigma_\infty^{-2} \quad (33)$$

with the only positive solution being  $\sigma_\infty^2 = \sigma^2$ , indicating the only steady state of  $\Sigma_k$  is exactly  $\Sigma$ .

An important observation is that  $\beta$  does not appear in the iterative relation (31) and (32), implying that the choice of  $\beta$  as a regularization parameter is mainly to ensure  $H_k$  defined in (29) is positive. Reasonable choice of  $\beta$  will not influence the evolution of the sample mean and covariance significantly. To characterize the behavior of the evolution of sample distribution  $\Pi_k$  we get at the  $k$ -th iteration, we introduce the total variation  $d_{TV}(P, Q)$  between two distributions  $P$  and  $Q$  over  $\mathbb{R}^d$  with probability density functions  $p$  and  $q$ :

$$d_{TV}(P, Q) = \int_{\mathbb{R}^d} |p(x) - q(x)| dx. \quad (34)$$

We also recall the result about the total variation between two Gaussian distributions with the same mean from [6]

$$d_{TV}(\mathcal{N}(\mu, \Sigma_1), \mathcal{N}(\mu, \Sigma_2)) = \frac{3}{2} \min \left\{ 1, \sqrt{\sum_{i=1}^d \lambda_i^2} \right\}, \quad (35)$$

where  $\lambda_i$  are eigenvalues of  $\Sigma_1^{-1}\Sigma_2 - I$ .

Next, we introduce the mixing time concerning the target distribution  $\Pi^*$  as

$$t_{\text{mix}}(\varepsilon, \Pi_0) = \min_k \{k | d_{TV}(\Pi_k, \Pi^*) \leq \varepsilon\}. \quad (36)$$

Now, we present a theorem regarding the mixing time of the proposed TT-BRWP, assuming the initial and exact means are equal. The result is comparable with Theorem 2 in [33] with slightly different assumptions and a simplified proof. The evolution of sample means  $m_k$  will be discussed in Section 4.2.

Before the main theorem, we prove the following technical lemma, which will be critical in the proof of the next theorem.

**Lemma 4.2.** *Let  $T, h \in [0, \sigma^2/4]$ , and  $\zeta \in [\sigma/2, 3\sigma/2]$ , the function*

$$f(\zeta) := 1 - h \left[ \sigma^{-2} + (1 + \sigma^{-2}T)^2 \frac{3T^2/\zeta^2 + \zeta^2 - 2T(1 + \sigma^{-2}T)}{(\zeta^2 - T^2/\zeta^2 + 2T(1 + \sigma^{-2}T))^2} \right]$$

*satisfies  $|f(\zeta)| \leq 1 - h/\sigma^2$ .*

*Proof.* Firstly, to show  $f(\zeta) \leq 1 - h/\sigma^2$ , we observe that it suffices to show

$$3T^2/\zeta^2 + \zeta^2 - 2T(1 + \sigma^{-2}T) > 0$$

for  $\zeta$  such that  $|\zeta - \sigma| \leq \sigma/2$ . This is equivalent to exploring the minimum value of a quadratic polynomial in  $\zeta^2$ , and the desired condition will boil down to

$$T^2(3 - (1 + T/\sigma^2)^2) \geq T^2$$

which is true by our choice of  $T$ .

Secondly, to show  $f(\zeta) \geq h/\sigma^2 - 1$ , it suffices to show

$$\sigma^2(1 + T\sigma^{-2})^2(3T^2\zeta^{-2} + \zeta^2 - 2T(1 + T\sigma^{-2})) \leq 6(\zeta^2 - T^2\zeta^{-2} + 2T(1 + T\sigma^{-2}))^2.$$

Taking the square root of both sides, we have

$$\begin{aligned} & \sigma(1 + T\sigma^{-2})(3T^2\zeta^{-2} + \zeta^2 - 2T(1 + T\sigma^{-2}))^{1/2} \\ & \leq \zeta^{-1}\sigma(1 + T\sigma^{-2})((3T^2 - T^2(1 + T\sigma^{-2})^2) + (\zeta^2 - T(1 + T\sigma^{-2}))^2)^{1/2} \\ & \leq \sigma(1 + T\sigma^{-2})(\zeta - T(1 + T\sigma^{-2})\zeta^{-1}). \end{aligned}$$

Now, we introduce a new function  $g(T)$  such that

$$g(T) := \sqrt{6}(\zeta^4 - T^2 + 2T(1 + T\sigma^{-2})\zeta^2) - \sigma(1 + T\sigma^{-2})(\zeta^3 - T(1 + T\sigma^{-2})\zeta).$$

Hence, it remains to verify  $g(T) > 0$  for any  $\zeta \in [\sigma/2, 3\sigma/2]$  and  $T \leq \sigma^2/4$ . As  $g(0) = \zeta^3(\sqrt{6}\zeta - \sigma)$ , we have  $g(0) > 0$ . Hence, we only need to show  $g'(T) > 0$  for  $T < \sigma^2/4$  which is

$$g'(T) = 3\sigma^{-3}\zeta T^2 + [(4\sqrt{6}\sigma^{-2}\zeta^2 + 4\sigma^{-1}\zeta - 2\sqrt{6})T + \sigma\zeta] + (2\sqrt{6}\zeta^2 - \zeta^3\sigma^{-1}).$$

Since each component of the above expression is positive, we have  $f(\zeta) \geq h/\sigma^2 - 1$  for all  $T$ , completing our proof.  $\square$

**Theorem 4.3.** *Let  $V(x) = x^\top \Sigma^{-1}x/2$  and  $\sigma_m$  be the smallest eigenvalue of  $\Sigma$ . For distribution  $\Pi_k$  evolved following Algorithm 1 with  $T, h \in [0, \sigma_m^2/4]$ , assuming  $\Sigma_0$  and  $\Sigma$  commute and  $|\sigma_0^{(j)} - \sigma^{(j)}| \leq \sigma^{(j)}/2$  for all  $j$ , we have*

$$\|\Sigma_k - \Sigma\|_F \leq (1 - \delta)^k \|\Sigma_0 - \Sigma\|_F,$$

where  $\delta = h/\sigma^2$ . Moreover, if we assume  $\mu_0 = 0$ , the total variation is bounded by

$$d_{TV}(\Pi_k, \Pi^*) \leq \frac{3}{2}C\sqrt{d}(1 - \delta)^k,$$

and the mixing time is given as

$$t_{mix}(\varepsilon, \Pi_0) = \mathcal{O}\left(\log(\sqrt{d}(1 - \delta))\right),$$

where the constant  $C$  depends on  $\Sigma$ ,  $\Sigma_0$ ,  $T$ , and  $h$ .

*Proof.* Under the assumption that  $\Sigma_0$  and  $\Sigma$  commute and share the same eigenspaces, it is sufficient to focus on the evolution of the  $j$ -th eigenvalue of  $\Sigma_k$  for each iteration which is given by (32) as

$$\sigma_{k+1} = \left(1 - h\sigma^{-2} + h\frac{(1 + \sigma^{-2}T)^2}{\sigma_k^2 - T^2\sigma_k^{-2} + 2T(1 + \sigma^{-2}T)}\right)\sigma_k, \quad (37)$$

where we skip the superscript  $j$  to simplify notations. We regard the evolution of  $\sigma_k$  as a fixed-point iteration with the iterative function

$$\phi(\zeta) = \left(1 - h\sigma^{-2} + h\frac{(1 + \sigma^{-2}T)^2}{\zeta^2 - T^2/\zeta^2 + 2T(1 + \sigma^{-2}T)}\right)\zeta. \quad (38)$$

The derivative of  $\phi$  with respect to  $\zeta$  is

$$\phi'(\zeta) = 1 - h\left[\sigma^{-2} + (1 + \sigma^{-2}T)^2\frac{3T^2/\zeta^2 + \zeta^2 - 2T(1 + \sigma^{-2}T)}{(\zeta^2 - T^2/\zeta^2 + 2T(1 + \sigma^{-2}T))^2}\right]. \quad (39)$$

By Lemma 4.2,  $|\phi'(\zeta)| \leq 1 - \delta$ , showing that  $\sigma_k$  converges linearly with rate  $1 - \delta$ , i.e.,  $|\sigma_k - \sigma| \leq (1 - \delta)^k |\sigma_0 - \sigma|$ . Hence, we show the convergence of  $\Sigma_k$ .

Leveraging the TV norm between two Gaussians in (35), we derive

$$d_{\text{TV}}(\Pi_k, \Pi^*) \leq \frac{3}{2} C \sqrt{d} (1 - \delta)^k, \quad C = \min\{1, \max_j \{(\sigma^{(j)})^{-1} (\sigma^{(j)} - \sigma_0^{(j)})\}\}. \quad (40)$$

Furthermore, the mixing time can be derived by combining the above estimate on the TV norm and its definition.  $\square$

Before we proceed to the more general case, we would like to compare the convergence of variance with the initial density defined in (21) depicted in the above theorem with the case of using the empirical distribution as  $\rho_0$  proposed in [33]. The result in the next figure comes from the iterative function we formulated in (38).

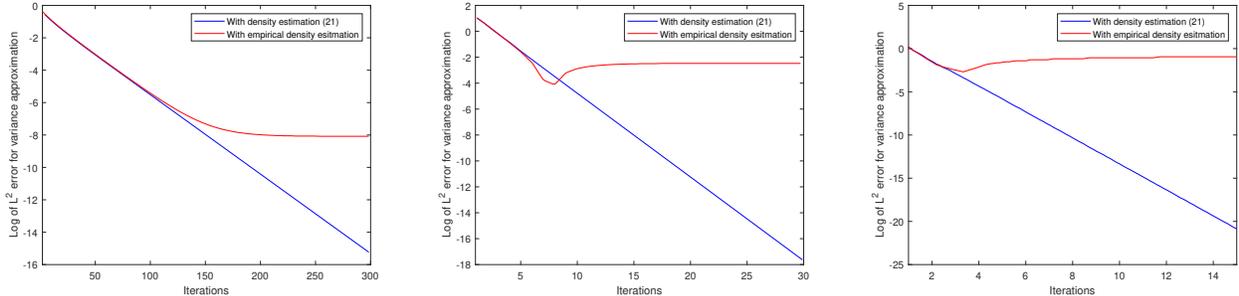


FIGURE 1. Logarithm of the approximation error for variance versus iteration using empirical distribution (blue) and density estimation defined in (21) (red) with  $\sigma = 2, 0.5, 0.25$  (from left to right),  $T = 0.1$ ,  $h = 0.1$ .

From Fig. 1, we can observe that the unbiased nature of Algorithm 1 by employing a delicate choice of the covariance matrix for  $\rho_0$  which improves the accuracy of variance approximation, especially for the case where  $\sigma$  is relatively small. Moreover, we also hope to numerically explore the influence of parameter  $T$  on the rate of convergence which is shown in the following figure with the help of iterative function (38).

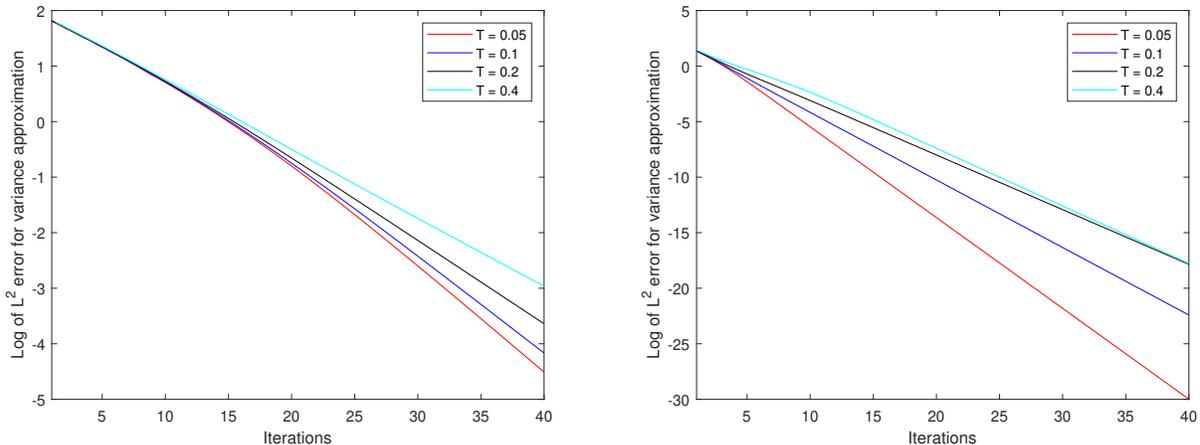


FIGURE 2. Logarithm of the approximation error of Algorithm 1 for variance versus iterations using different terminal time  $T$  with  $T = 0.05, 0.1, 0.2, 0.4$ ,  $h = 0.1$ ,  $\sigma = 0.5$  (Left) and 1 (Right).

From Fig. 2, we observe that as  $T$  becomes smaller, a slightly faster convergence can be achieved. However, as  $T$  becomes smaller, the integration in (17) requires more nodal points to obtain a satisfactory numerical approximation. Hence, we will choose a fixed  $T$  empirically in our following numerical experiments.

Next, we void the assumption that  $\Sigma_0$  and  $\Sigma$  commute with each other and derive a continuous analog of the evolution of  $\Sigma_k$  to further demonstrate its convergence property. First, from the results in Theorem 4.3 and numerical experiments in Fig. 2, the algorithm is interesting when  $T$  and  $h$  are both small. Hence, it is reasonable to drop high-order terms in  $h$  and  $T$ .

Let us revisit the iterative relationship about  $\Sigma_k$  in (28) and notice that the evolution of the covariance matrix can be written as

$$\Sigma_{k+1} = \Sigma_k - h(\Sigma^{-1}\Sigma_k + \Sigma_k\Sigma^{-1} - \beta^{-1}\Sigma_T^{-1}\Sigma_k - \beta^{-1}\Sigma_k\Sigma_T^{-1}) + \mathcal{O}(h^2). \quad (41)$$

Then by the Neumann series, we notice that

$$\beta^{-1}\Sigma_T^{-1} = (2T + \Sigma_k - T\Sigma^{-1}\Sigma_k - T\Sigma_k\Sigma^{-1} + \mathcal{O}(T^2))^{-1} = \Sigma_k^{-1}(I + T(\Sigma^{-1} + \Sigma_k\Sigma^{-1}\Sigma_k^{-1} - 2\Sigma_k^{-1})) + \mathcal{O}(T^2).$$

Substituting this into (41), we arrive at

$$\Sigma_{k+1} = \Sigma_k - h[\Sigma_k\Sigma^{-1} + \Sigma^{-1}\Sigma_k - 2 - T(2\Sigma^{-1} + \Sigma_k^{-1}\Sigma^{-1}\Sigma_k + \Sigma_k\Sigma^{-1}\Sigma_k^{-1} - 4\Sigma_k^{-1})] + \mathcal{O}(h^2) + \mathcal{O}(T^2). \quad (42)$$

Now, let  $h \rightarrow 0$  and consider the continuous analog of the covariance matrix as  $\Sigma_t$  depends on time  $t$ . This leads to

$$\frac{d\Sigma_t}{dt} = -\Sigma_t\Sigma^{-1} - \Sigma^{-1}\Sigma_t + 2 + T(2\Sigma^{-1} + \Sigma_t^{-1}\Sigma^{-1}\Sigma_t + \Sigma_t\Sigma^{-1}\Sigma_t^{-1} - 4\Sigma_t^{-1}) + \mathcal{O}(T^2). \quad (43)$$

To further explore the convergence of  $\Sigma_t$ , we consider the Frobenius norm of  $(\Sigma_t - \Sigma)^\top(\Sigma_t - \Sigma)$ , which is equivalent to the trace of the matrix, to derive

$$\begin{aligned} \frac{d\|(\Sigma_t - \Sigma)^2\|_F}{dt} &= \text{Tr}\{-4(\Sigma_t - \Sigma)\Sigma^{-1}(\Sigma_t - \Sigma) + 4T(\Sigma_t - \Sigma)(\Sigma^{-1}\Sigma_t^{-1})(\Sigma_t - \Sigma)\} + \mathcal{O}(T^2) \\ &= -4\text{Tr}\{(\Sigma_t - \Sigma)(\Sigma^{-1} - T\Sigma^{-1}\Sigma_t^{-1})(\Sigma_t - \Sigma)\} + \mathcal{O}(T^2) \\ &\leq -4\text{Tr}\{(\Sigma_t - \Sigma)\Sigma^{-1}(\Sigma_t - \Sigma)\}(I - T\rho\{\Sigma^{-1}\}) + \mathcal{O}(T^2), \end{aligned} \quad (44)$$

where we have used the facts that  $\text{Tr}\{AB\} = \text{Tr}\{BA\}$  and  $|\text{Tr}\{AB\}| \leq |\text{Tr}\{A\}|\rho\{B\}$  where  $\rho$  is the spectral radius of  $B$ . The above implies that when  $(I - T\rho\{\Sigma^{-1}\})$  is positive, which is guaranteed for small  $T$ ,  $\|(\Sigma_t - \Sigma)^2\|_F$  decays to 0. The conclusion of the above discussion is summarized in the following theorem.

**Theorem 4.4.** *When  $h$  and  $T$  are sufficiently small, let  $\Sigma_t$  be the continuous analog of  $\Sigma_k$  when  $h \rightarrow 0$ . Then the Frobenius norm of  $\Sigma_t - \Sigma$  will decay to 0 for  $T$  small enough, and the decay rate is bounded by*

$$\frac{d\|(\Sigma_t - \Sigma)^2\|_F}{dt} \leq -4\text{Tr}\{(\Sigma_t - \Sigma)\Sigma^{-1}(\Sigma_t - \Sigma)\}(I - T\rho\{\Sigma^{-1}\}) + \mathcal{O}(T^2) \leq 0 \quad (45)$$

which depends on eigenvalues of  $\Sigma$ .

The above theorem shows, from the continuous perspective, that the covariance matrix of  $X_k$  will converge to  $\Sigma$  for any  $T, h$  that are sufficiently small.

To conclude this section, we provide a brief comparison between our theoretical convergence result and BRWP as well as MALA.

- In comparison to BRWP, the proposed sampling algorithm is unbiased, leading to a steady state that exactly matches the target distribution while BRWP introduces a variance shift by a factor of  $aT^2/\beta$ , as illustrated in Fig., 1.
- Contrasting with MCMC-type methods like MALA, which has a theoretical upper bound for mixing time of  $\mathcal{O}(d^2)$  [11], our method exhibits a faster convergence with a theoretical bound of  $\mathcal{O}(\log(\sqrt{d}))$ . This suggests that our approach achieves faster convergence in higher dimensions, a characteristic that will be evident in numerical experiments.

**4.2. Analysis of TT-BRWP Algorithm for a Simplified Bayesian Inverse Problem.** In this section, we shall focus on the accuracy of the proposed TT-BRWP in an interesting scenario that arises in Bayesian inverse problems and data fitting. The potential function will be

$$V(x) = \frac{\|Ax - \mu\|_2^2}{2\lambda^2} + \|\Gamma x\|_2^2 \quad (46)$$

where  $A$  and  $\Gamma$  are known linear forward operator and linear regularization operator respectively that could be  $m \times d$  for  $m \neq d$ ,  $\mu$  is the noisy observation, and  $\lambda$  is the noise level. This model corresponds to Tikhonov regularization for inverse problems and  $L^2$  regularization in data fitting. For  $V(x)$ , we can rewrite it as

$$V(x) = (x - (A^\top A + \Gamma^\top \Gamma)^{-1} A^\top \mu)^\top (A^\top A + \Gamma^\top \Gamma) (x - (A^\top A + \Gamma^\top \Gamma)^{-1} A^\top \mu) + C \quad (47)$$

where  $C$  is independent of  $x$ . Let  $\tilde{\mu} = (A^\top A + \Gamma^\top \Gamma)^{-1} A^\top \mu$  and  $\Sigma = (A^\top A + \Gamma^\top \Gamma)^{-1} A^\top \mu$ , then our goal is to draw samples from the distribution  $\mathcal{N}(\tilde{\mu}, \Sigma)$  with the proposed TT-BRWP.

We note that since  $A$  and  $\Gamma$  are available,  $\tilde{\Sigma}$  and  $\Sigma_k$  will commute as the only unknown is the scalar-valued noise level. Now, as in the previous section, we still assume  $\tilde{\mu} = 0$  since the general case can be obtained by a change of coordinates. For the evolution of sample mean  $\mu_k$ , we focus on one entry  $m_k = \mu_k(j)$  and  $\tilde{m} = \tilde{\mu}(j)$  for some  $j$  which by (31) satisfies

$$(m_{k+1} - \tilde{m}) = \left( 1 - h \frac{T + (\sigma_k^{(j)})^2}{(\sigma_k^{(j)})^4 + 2T(\sigma_k^{(j)})^2 - T^2} \right) (m_k - \tilde{m}). \quad (48)$$

To compute the total variation between two Gaussian with different mean, we recall the following result which is a simplified version of Theorem 1.8 in [1].

**Lemma 4.5.** *Suppose  $\Sigma_1, \Sigma_2$  commute and  $d_{TV}(\mathcal{N}(\mu_1, \Sigma_1), \mathcal{N}(\mu_2, \Sigma_2)) \leq 1/600$ , we have*

$$d_{TV}(\mathcal{N}(\mu_1, \Sigma_1), \mathcal{N}(\mu_2, \Sigma_2)) \leq \frac{1}{\sqrt{2}} \max \left\{ \|\Sigma_1^{-1}\|_F \|\Sigma_1 - \Sigma_2\|_F, \|\Sigma^{-1/2}(\mu_1 - \mu_2)\|_2 \right\}.$$

Then we can derive the following result regarding the total variation difference and mixing time.

**Theorem 4.6.** *Let  $V(x)$  be defined in (46) and  $\sigma_m$  be the smallest eigenvalue of  $\tilde{\Sigma}$ . For distribution  $\Pi_k$  evolved following Algorithm 1 with  $T \leq \sigma_m^2/(2 + 2\sqrt{2})$ ,  $h \leq \sigma_m^2/4$ , assuming the initial distribution satisfies  $d_{TV}(\Pi_0, \Pi^*) \leq 1/600$  and  $|\sigma_0^{(j)} - \sigma^{(j)}| \leq \sigma^{(j)}/2$  for all  $j$ , we have*

$$\|\mu_k - \tilde{\mu}\|_\infty \leq (1 - \delta)^k \|\mu_0 - \tilde{\mu}\|_\infty, \quad \|\Sigma_k - \tilde{\Sigma}\|_F \leq (1 - \delta)^k \|\Sigma_0 - \tilde{\Sigma}\|_F,$$

where  $\delta = h/(2\sigma^2)$ . Moreover, the total variation is bounded by

$$d_{TV}(\Pi_k, \Pi^*) \leq \frac{1}{\sqrt{2}} C \sqrt{d} (1 - \delta)^k,$$

and the mixing time is given as

$$t_{mix}(\varepsilon, \Pi_0) = \mathcal{O} \left( \log(\sqrt{d}(1 - \delta)) \right),$$

where the constant  $C$  depends on  $\mu, \mu_0, T$ , and  $h$ .

*Proof.* The convergence of covariance is given by Theorem 4.3. Next, to show  $\|\mu_k - \tilde{\mu}\|_\infty \leq (1 - \delta)^k \|\mu_0 - \tilde{\mu}\|_\infty$ , by the iterative relation in (48), we observe it suffices to verify

$$h \leq (T + \sigma^2) - \frac{2T^2}{(T + \sigma^2)} \leq \frac{h}{\delta}, \quad (49)$$

where we skip the superscript  $j$  for simplification. Then we may take  $f(t) := (t + \sigma^2) - 2t^2/(t + \sigma^2)$  and it suffices to show  $f(0), f(\sigma^2/4)$  satisfy the above inequalities and  $f(t)$  is monotonic.

Next, all the above conditions will be equivalent to

$$h \leq \sigma^2 \leq h/\delta, \quad h \leq 23/20\sigma^2 \leq h/\delta, \quad \sigma^4 - 2T\sigma^2 - T^2 \geq 0;$$

which are satisfied by our choice of parameters.

Finally, the statement on total variation norm and mixing time follows from Lemma 4.5 and their definitions directly.  $\square$

In conclusion, Theorem 4.6 establishes the linear convergence of the proposed method for the potential function (46), which is relevant in the context of Bayesian inverse problems and data fitting applications.

Moreover, from (48), we can also derive the continuous analogue  $m_t$  for any component of  $\mu$  which is

$$|m_t - \tilde{m}| = \exp\left(-\frac{T + \sigma^2}{(T + \sigma^2)^2 - 2T^2}t\right)|m_0 - \tilde{m}|,$$

and if  $(T + \sigma^2)^2 - 2T^2 > 0$ , i.e.,  $T \leq \sigma^2/(\sqrt{2} - 1)$ ,  $m_t$  converges monotonically to  $\tilde{m}$ . This provides a clear continuous analog of results in Theorem 4.6

**4.3. Numerical Consideration and Computational Complexity.** In this subsection, we shall briefly discuss some numerical details in the implementation of TT-BRWP in Algorithm 1 and the efficiency of computation which provides important practical guidance on its application.

For the computation of  $\hat{\eta}_T$  and  $\nabla\hat{\eta}_T$  in line 7, we utilize a similar kernel formulation introduced in (18) which provides a solution for the forward heat equation in (11). Hence, we have

$$\hat{\eta}_T(x) = \left(\frac{\beta}{2\pi T}\right)^{d/2} \int_{\mathbb{R}^d} \exp\left(-\frac{\beta\|x-y\|_2^2}{4T}\right)\hat{\eta}_0(y)dy \tag{50}$$

and its gradient will be

$$\nabla\hat{\eta}_T(x) = -\left(\frac{\beta}{2\pi T}\right)^{d/2} \int_{\mathbb{R}^d} \frac{\beta(x-y)}{2T} \exp\left(-\frac{\beta\|x-y\|_2^2}{4T}\right)\hat{\eta}_0(y)dy. \tag{51}$$

We observe that all the above computations as well as  $\hat{\eta}_T \circ \eta_T$  can be implemented in a tensor-train format efficiently by using  $K_d$  defined in (19).

Moreover, in algorithm 1, when  $x$  is near the boundary of the discretization grid, the value of  $\eta_0$

$$\eta_0(x) = \int_{\mathbb{R}^d} \exp\left(-\frac{\beta}{2}\left(V(x) - \frac{\|x-y\|_2^2}{2T}\right)\right)dy, \tag{52}$$

will be extremely small which creates underflow and induces difficulties in approximating  $1/\eta_0$ . In this scenario, one may compute  $-\log(\eta_0)$  on line 2 and compute the exponential of a tensor by Taylor’s polynomial of exponential function which enhances the numerical stability with the price of increased computational complexity.

Finally, for the interpolation on line 10, linear interpolation on a gridded mesh is employed which has proven to be able to provide satisfactory numerical accuracy and efficiency.

## 5. NUMERICAL EXPERIMENTS

In this section, we test the proposed TT-BRWP in algorithm 1 and compare it with the classical Unadjusted Langevin Algorithm (ULA), the unbiased Metropolis-adjusted Langevin algorithm (MALA), and the BRWP with MC integration in [33]. For all our following experiments, the initialization  $X_0$  is always sampled from a Gaussian distribution with mean 0 and variance 1. Moreover, for all the following TT-BRWP and BRWP algorithms, we choose  $\beta = 4$  and 1 respectively.

**5.1. Gaussian Distribution. Example 1:** In this example, we explore the case  $V(x) = -(x-a)^\top \Sigma^{-1}(x-a)/2$  in  $\mathbb{R}^6$  where  $\Sigma = \sigma^2 I_6$  for different choices of  $\sigma$  to validate our theoretical results on the convergence of variance presented in Theorem 4.3.

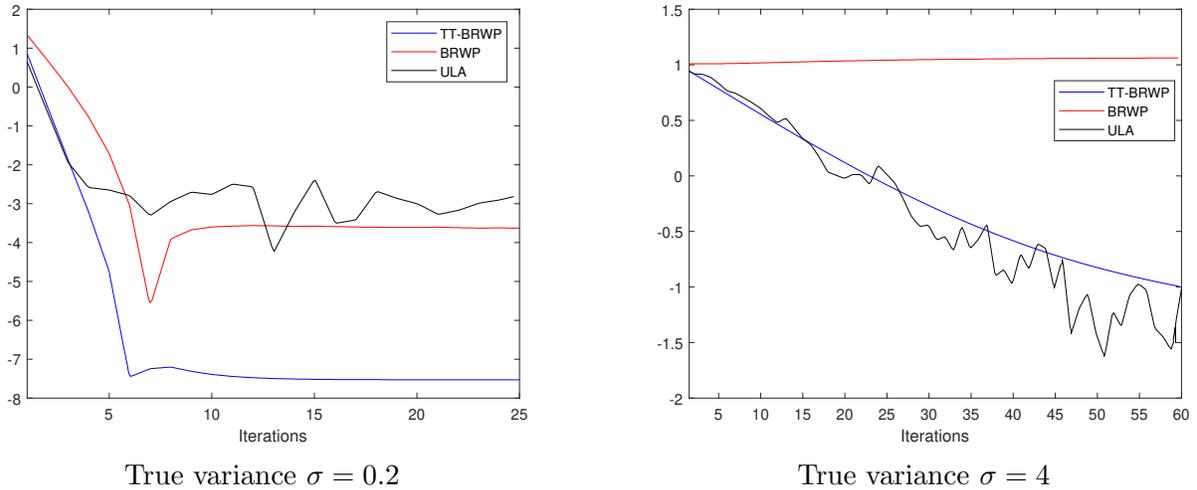


FIGURE 3. Example 1. The logarithm of the  $L^2$  error for the variance with samples generated from TT-BRWP (blue), BRWP (red), and ULA (black).

From Fig. 3, we observe clear linear convergence of the variance for Gaussian distributions with both  $\sigma = 0.2$  and  $\sigma = 4$ . From left, TT-BRWP exhibits faster convergence for small  $\sigma$  and improves the final estimation significantly due to the unbiased nature; from right, TT-BRWP is more robust compared to BRWP for large  $\sigma$  as BRWP degenerates in this case.

**Example 2:** Next, we consider the case when  $\Sigma$  is a general SPD matrix. We pick

$$\hat{\Sigma} = \begin{bmatrix} 0.4 & 0.2 & 0.3 \\ 0.2 & 3 & 0.2 \\ 0.3 & 0.2 & 6 \end{bmatrix}$$

and let  $\Sigma = \begin{bmatrix} \hat{\Sigma} & \mathbf{0} \\ \mathbf{0} & I_3 \end{bmatrix}$ . Then we examine the evolution of the mean and the total variation for the distribution generated by TT-BRWP, BRWP, and ULA, respectively.

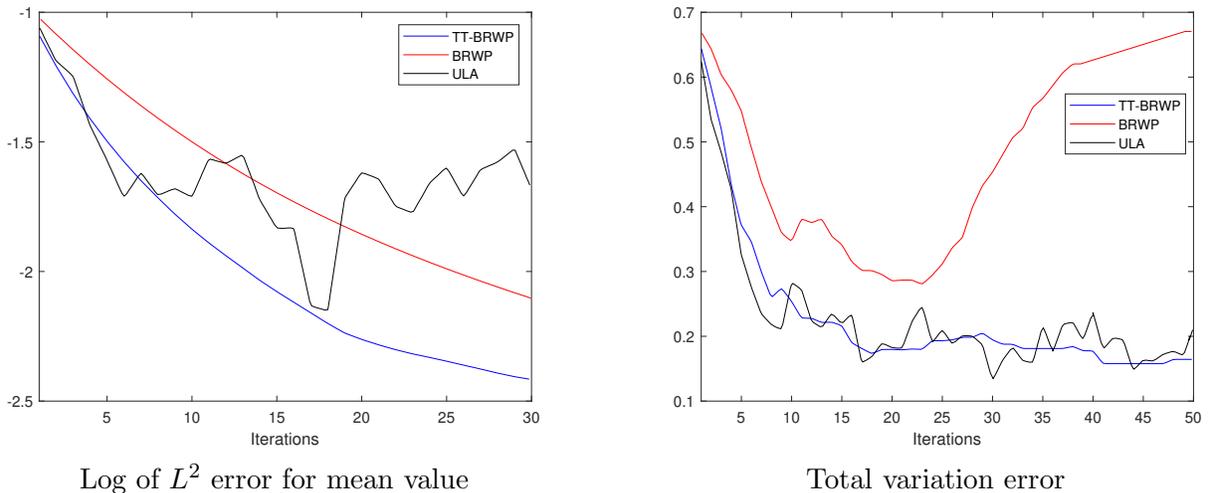


FIGURE 4. Example 2. The error of mean for samples generated for TT-BRWP (blue), BRWP (red), and ULA (black) versus iterations.

From the first plot of Fig. 4, TT-BRWP provides the best approximation of the sample mean among the three algorithms. Moreover, from the second plot, BRWP with MC integration degenerates for this case which shows the necessity of the TT approximation.

**5.2. Gaussian Mixture and Bimodal Distribution. Example 3:** In the third example, we consider sampling from a Gaussian mixture distribution in  $\mathbb{R}^6$  with

$$\rho^*(x) = \frac{1}{Z} \frac{\exp(-\|x - a\|_2^2/2) + \exp(-\|x + a\|_2^2/2)}{2},$$

for a normalization term  $Z$ . We pick  $a = (1, -0.5, 0, 0, 0, 0)^\top$  in  $\mathbb{R}^6$  and choose  $h = 0.02$ ,  $T = 0.02$  to ensure the convergence of sampling algorithms.

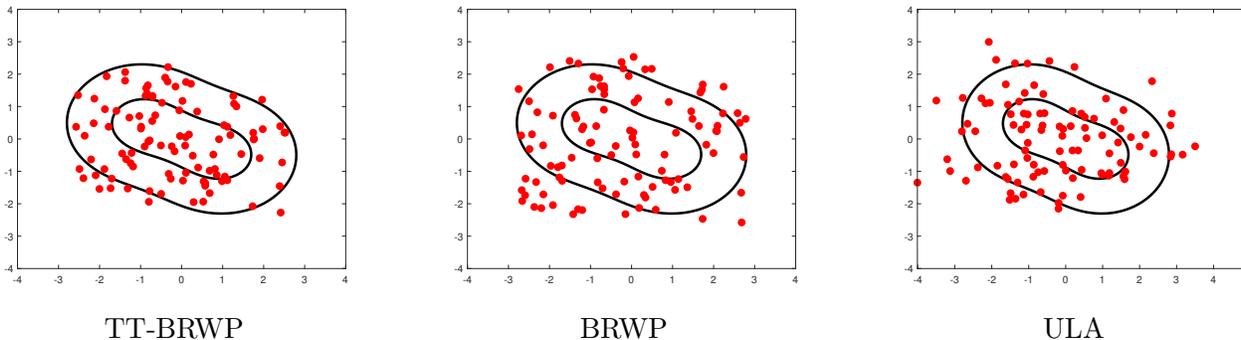


FIGURE 5. Example 3. Evolution of particles for different algorithms after 30 iterations where the contour lines are 0.6 and 0.1.

From Fig. 5, TT-BRWP in the first plot tends to give a more structured distribution of samples for the case of Gaussian mixture distribution, while ULA has many more points that fall outside the region of high probability of the original distribution.

**Example 4:** In this example, we consider sampling from a bimodal distribution (double moon) in  $\mathbb{R}^6$  with

$$\rho^*(x) = \frac{1}{Z} \exp(-2(\|x\| - a)^2) [\exp(-2(x_1 - a)^2) + \exp(-2(x_1 + a)^2)].$$

We test the case  $a = 2$  with  $T = 0.01$  and  $h = 0.01$ .

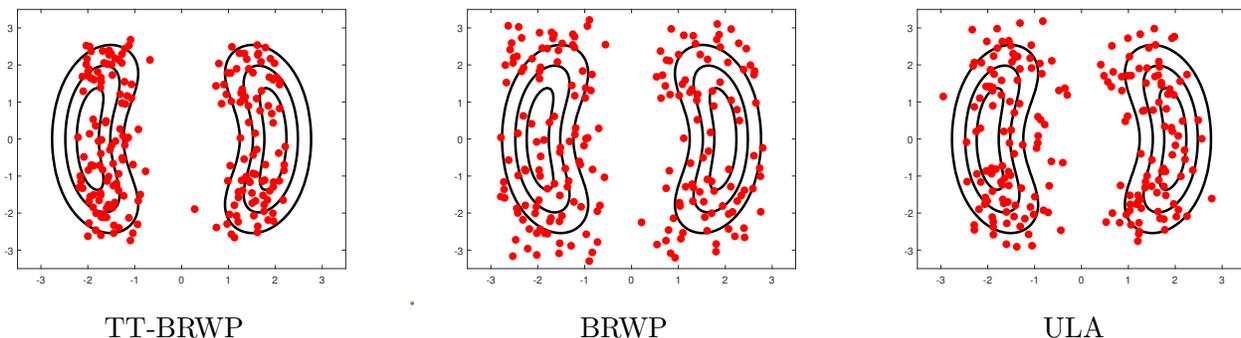


FIGURE 6. Example 4. Evolution of sample points for different algorithms after 30 iterations where the contour lines are 0.8, 0.4, and 0.1 in 20 iterations.

From Fig. 6, it is evident that TT-BRWP provides a set of particles that are concentrated more in the high probability region of the target distribution within only 20 iterations, while BRWP and ULA converge more slowly.

**5.3. Nonconvex Potential Function. Example 5:** In this example, we present a particularly interesting case where the potential function  $V(x)$  is nonconvex, which is known to be highly challenging due to the slow convergence of MC integration as shown in table 3.1. We consider

$$V(x) = \|x - a\|_{1/2}^2 = \left( \sum_{j=1}^3 |x_j - a(j)|^{1/2} \right)^2,$$

with  $a = (1, 1, 0)^\top$  in  $\mathbb{R}^3$ .

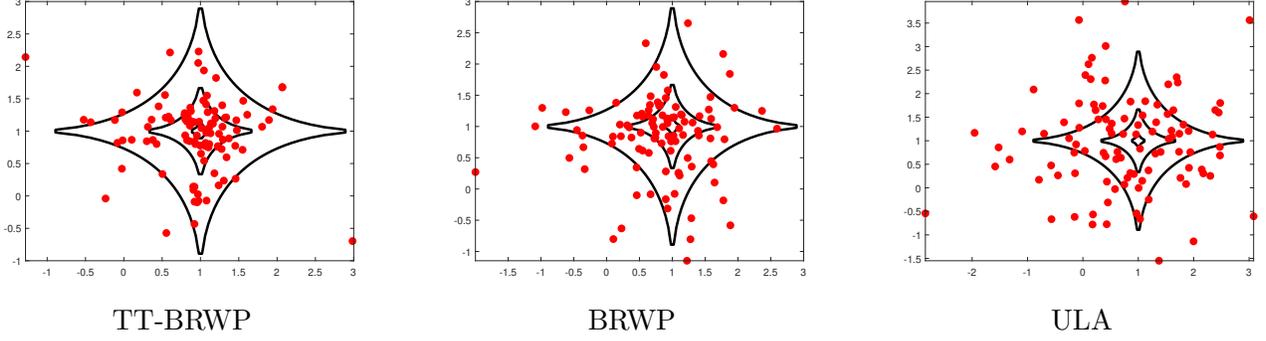


FIGURE 7. Example 5. Evolution of sample points for different algorithms where the contour lines are 0.8, 0.4, and 0.1 after 20 iterations.

From Fig. 7, we can observe that the proposed TT-BRWP algorithm provides a quite accurate set of samples that distributed follows the desired non-convex density function, while samples from BRWP with MC integration and ULA does not exhibit clear similarity with the target density function.

**5.4. Bayesian Inverse Problems.** In this subsection, we will examine the accuracy and robustness of the proposed TT-BRWP to tackle several interesting and ill-posed inverse problems using Bayesian inference.

Let us first recall the general setting for Bayesian inverse problems. Firstly, we write the measurement process as

$$y = \mathcal{G}(x^*) + \zeta,$$

where  $\zeta$  is random noise,  $x^*$  is the underlying truth we would like to recover, and  $\mathcal{G}$  is the forward operator. The objective of Bayesian inverse problems is to estimate the posterior distribution

$$\pi(x|y) = \pi(y|x)\pi(x), \quad (53)$$

where  $\pi(x)$  is the prior density, and  $\pi(y|x)$  is the likelihood function depending on the forward operator and noise level. Existing sampling algorithms will often suffer from slow convergence or even divergence, especially for high-dimensional or nonlinear scenarios.

**Example 6:** The first example we consider is a classical ill-posed inverse problem for recovering the initial distribution of the heat equation. Let  $u$  be the solution of the heat equation in the sense that

$$\frac{\partial}{\partial t} u(t, x) = \frac{\partial^2}{\partial x^2} u(t, x), \quad u(t, 0) = u(t, \pi) = 0, \quad u(0, x) = u_0(x), \quad u(T, x) = u_T(x), \quad x \in [0, \pi].$$

Then our goal is to recover  $u_0$  from noisy measurement data  $u_T$ . To simplify the computation, we assume  $u_0$  is composed of a series of trigonometric functions, and our task is to recover  $\theta_k$  in  $u_0 = \sum_{k=1}^d \theta_k \sin(kx)$  with  $d = 10$ .

In our experiment, let  $\theta$  represent vectors containing coefficients  $\theta_k$ ,  $y$  be the measurement data polluted by noise with a noise level  $\sigma^2 = 0.1$ , the likelihood function is chosen as

$$\pi(\theta|y) = \exp \left( - \frac{\|u(T, x, \theta) - y\|_2^2}{2\sigma^2} \right). \quad (54)$$

The prior distribution is chosen as the  $L_1$  norm of  $\theta$  as a sparse constraint, and hence, the potential function  $V(\theta)$  will simply be

$$V(\theta) = \frac{\|u(T, x, \theta) - y\|_2^2}{2\sigma^2} + \tau\|\theta\|_{L^1},$$

where  $\tau$  is a small regularization parameter. Then our reconstruction results are presented in Fig. 8.

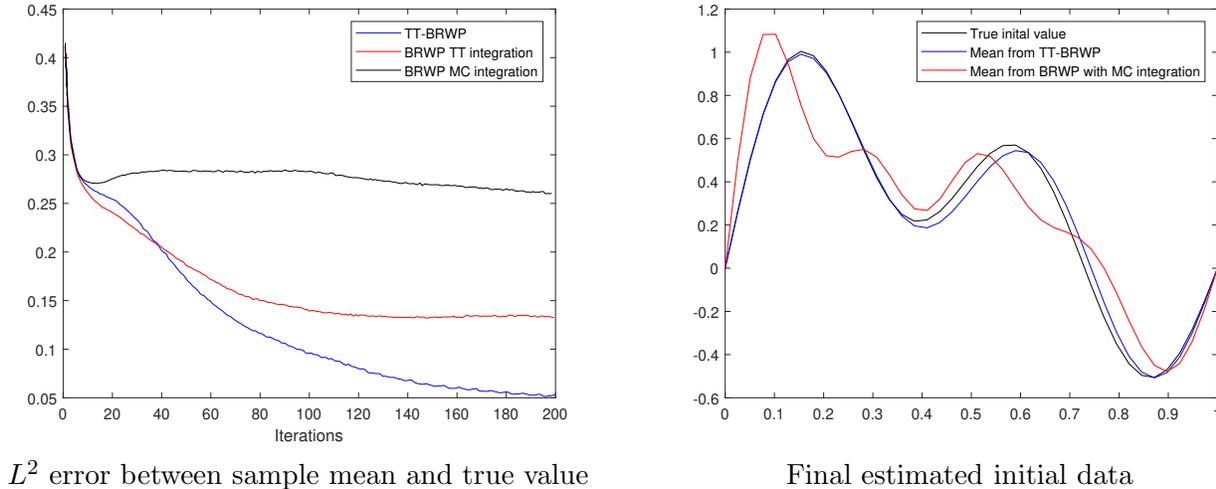


FIGURE 8. Example 6: Recovering initial data for the heat equation under  $L^1$  regularization for  $T = 0.1$  and  $h = 0.01$ .

From Fig. 8, we can observe that TT-BRWP in blue provides a much better reconstruction compared to sampling algorithms with MC integration in red line and also ULA which does not converge for this task. Moreover, even if we compare BRWP with TT integration (in black) and MC integration (in red) in the first plot, TT integration can indeed improve the accuracy significantly, while the final estimation is still clearly biased due to the employment of empirical distribution for the estimation of the density function.

**Example 7:** We consider a nonlinear inverse problem for an elliptic boundary value problem from [13]. The potential  $p$  satisfies

$$-\frac{d}{dy} \left( \exp(x_1) \frac{d}{dy} p(y) \right) = 1, \quad y \in [0, 1], \tag{55}$$

with  $p(0) = 0$  and  $p(1) = x_2$ . Then the solution to the forward problem has an explicit solution

$$p(y) = x_2 y + \exp(-x_1) \left( -\frac{y^2}{2} + \frac{y}{2} \right). \tag{56}$$

Due to the exponential term, it is clear that this inverse problem is nonlinear. Then for measurement points  $y_1, y_2 \in [0, 1]$ , the forward operator will be

$$\mathcal{G}(x) = (p(y_1), p(y_2))^T. \tag{57}$$

By employing an  $L^2$  regularization term, the potential function will simply be

$$V(x) = \frac{\|\mathcal{G}(x) - \tilde{p}\|_2^2}{2\sigma^2} + \tau\|x\|_2^2, \tag{58}$$

for noisy measurement data  $\tilde{p}$ , and noise level  $\sigma = 0.1$  which corresponds to 10% noise in the measurement data. In our experiments, we choose  $x^* = [0.4, 1]$ ,  $y_1 = 0.25$ , and  $y_2 = 0.75$ . The reconstruction results for reconstruction error with different choices of step size are presented in Fig. 9.

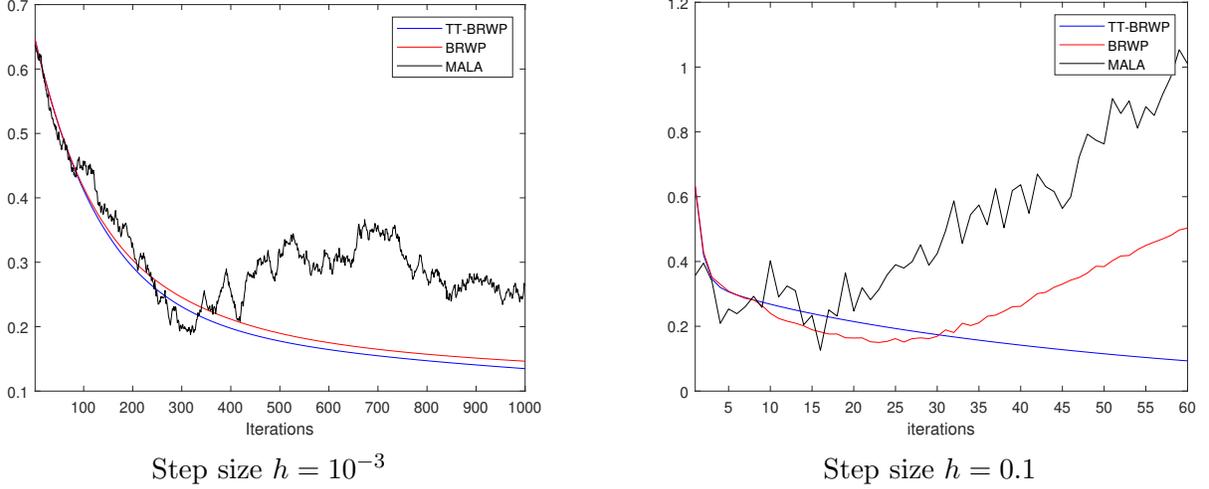


FIGURE 9. Example 7:  $L^2$  error of the estimation of parameters  $x^*$  versus iteration for an inverse elliptic boundary value problem with different step sizes.

From Fig. 9, we observe that when the step size is relatively small, all three methods under consideration exhibit convergence to the stationary distribution, while TT-BRWP in blue provides the most accurate result after a large number of iterations. Moreover, when the step size is large, i.e., 0.1, in the second plot, TT-BRWP is the only one that provides convergence in a few iterations which demonstrates its strong robustness.

**Example 8.** In this example, we examine a nonlinear non-convex inverse problem from [17] where we consider the following Cauchy problem

$$\frac{\partial^2 u}{\partial t^2}(t, x, \theta) - \frac{\partial^2 u}{\partial x^2}(t, x, \theta) = 0, \quad u(0, x, \theta) = h(x, \theta), \quad u_t(0, x, \theta) = 0, \quad (59)$$

where  $h(x, \theta)$  is a unknown source function parameterized by  $\theta \in \mathbb{R}^d$  such that

$$h(x, \theta) = \sum_{j=1}^d \frac{\sin(k(x - \theta_j))}{k|x - \theta_j|}, \quad (60)$$

where the sinc function can be considered as an approximation to a point source which also makes the problem itself non-convex. Moreover, by d'Alembert's formula, the solution to the Cauchy problem will be

$$u(t, x, \theta) = \frac{h(x - t, \theta) + h(x + t, \theta)}{2}. \quad (61)$$

The measurement data is considered as  $\tilde{u}(x_i, t_i, \theta^*)$  for  $x_i$  and  $t_j$  distributed uniformly on  $[-2, 2]$  and  $[0, 1]$  for  $i = 1, \dots, N_i$  and  $j = 1, \dots, N_j$ .

We again employ a  $L^2$  regularization term and the potential function we are interested in will be

$$V(\theta) = \frac{\sum_{i=1}^{N_i} \sum_{j=1}^{N_j} |u(t_j, x_i, \theta) - \tilde{u}(t_j, x_i, \theta^*)|^2}{2\sigma^2} + \tau \|\theta\|_2^2, \quad (62)$$

with  $\sigma = 1$  and  $N_i = N_j = 21$ . Let  $\theta^* = [-0.9, -0.3, 0.4, 1]$ . The result is presented in Fig. 5.4.

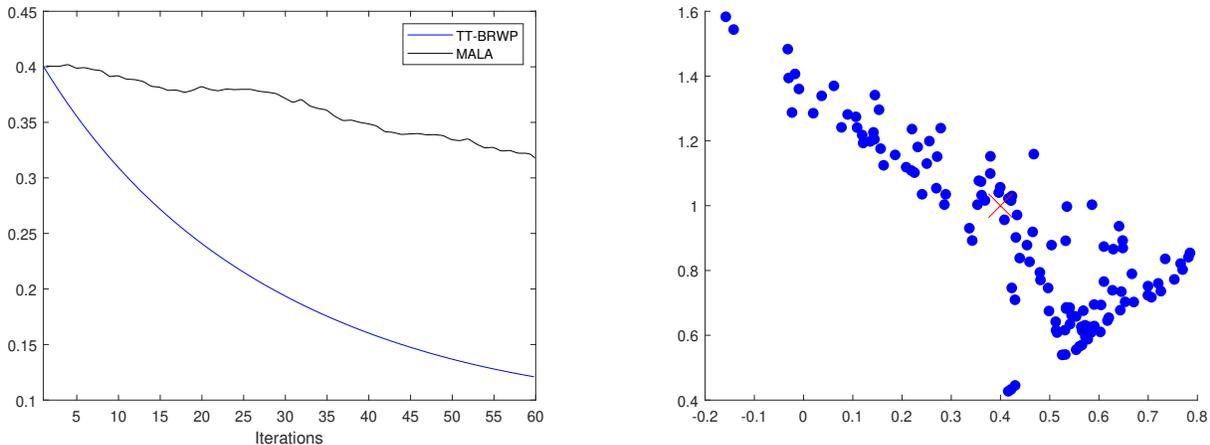


FIGURE 10. Example 8: Error of the estimation for  $\theta$  versus iteration (left) and generated particles from TT-BRWP for  $(\theta_3, \theta_4)$  (right) where exact values are  $\theta_3^* = 0.4$ ,  $\theta_4^* = 1$ .

From Fig. 5.4, we observe that TT-BRWP converges much faster than MALA to the desired distribution, and the generated particles are distributed near the exact value. We remark that BRWP diverges for this case.

## 6. CONCLUSION.

In this paper, we proposed a sampling algorithm based on the tensor train approach, aiming to draw samples from potentially high-dimensional and complex distributions. Our method is inspired by a kernel formulation for the regularized Wasserstein proximal operator and employs tensor train approximation for high-dimensional integration. Specifically, by accurately approximating the crucial score function and employing a suitable kernel density approximation, our new sampling algorithm demonstrates superior accuracy, stability, and speed compared to BRWP and Langevin dynamic types sampling algorithms.

There are several intriguing avenues for future exploration. From a theoretical standpoint, establishing the accuracy and mixing time of the proposed algorithm for general non-Gaussian distributions, as verified by our numerical experiments, would be highly attractive. A comprehensive theoretical treatment would enhance the method’s applicability to a broader range of practically important scenarios. Moreover, from an algorithmic perspective, exploring the interplay of tensor methods with Wasserstein proximal kernels, investigating the acceleration of MCMC algorithms within the current framework, and exploring related kernel methods for addressing challenging high-dimensional scientific computing problems would be interesting avenues for further research.

## REFERENCES

- [1] J. ARBAS, H. ASHTIANI, AND C. LIAW, *Polynomial time and private learning of unbounded Gaussian mixture models*, arXiv:2303.04288, (2023).
- [2] C. J. BÉLISLE, H. E. ROMELJN, AND R. L. SMITH, *Hit-and-run algorithms for generating multivariate distributions*, Math. Oper. Res., 18 (1993), pp. 255–266.
- [3] S. BROOKS, A. GELMAN, G. JONES, AND X.-L. MENG, *Handbook of Markov chain Monte Carlo*, CRC press, 2011.
- [4] H. CHUNG AND J. C. YE, *Score-based diffusion models for accelerated MRI*, Med. Image Anal., 80 (2022), p. 102479.
- [5] T. CUI, K. J. LAW, AND Y. M. MARZOUK, *Dimension-independent likelihood-informed MCMC*, J. Comput. Phys., 304 (2016), pp. 109–137.
- [6] L. DEVROYE, A. MEHRABIAN, AND T. REDDAD, *The total variation distance between high-dimensional Gaussians with the same mean*, arXiv:1810.08693, (2018).
- [7] P. DHARIWAL AND A. NICHOL, *Diffusion models beat gans on image synthesis*, Adv. Neural Inf. Process. Syst., 34 (2021), pp. 8780–8794.

- [8] R. DIAN, S. LI, AND L. FANG, *Learning a low tensor-train rank representation for hyperspectral image super-resolution*, IEEE Trans. Neural Netw. Learn. Syst., 30 (2019), pp. 2672–2683.
- [9] A. DOUCET, W. GRATHWOHL, A. G. MATTHEWS, AND H. STRATHMANN, *Score-based diffusion meets annealed importance sampling*, Adv. Neural Inf. Process. Syst., 35 (2022), pp. 21482–21494.
- [10] A. DURMUS AND É. MOULINES, *High-dimensional bayesian inference via the unadjusted langevin algorithm*, Bernoulli, (2016).
- [11] R. DWIVEDI, Y. CHEN, M. J. WAINWRIGHT, AND B. YU, *Log-concave sampling: Metropolis-Hastings algorithms are fast*, J. Mach. Learn. Res., 20 (2019), pp. 1–42.
- [12] R. N. GANTNER AND C. SCHWAB, *Computational higher order quasi-Monte Carlo integration*, Springer, 2016.
- [13] A. GARBUNO-INIGO, F. HOFFMANN, W. LI, AND A. M. STUART, *Interacting Langevin diffusions: Gradient structure and ensemble Kalman sampler*, SIAM J. Appl. Dyn. Syst., 19 (2020), pp. 412–441.
- [14] M. GRIEBEL AND H. HARBRECHT, *Analysis of tensor approximation schemes for continuous functions*, Found. Comput. Math., (2021), pp. 1–22.
- [15] Y. HUR, J. G. HOSKINS, M. LINDSEY, E. SToudenMIRE, AND Y. KHOO, *Generative modeling via tensor train sketching*, Appl. Comput. Harmon. Anal., 67 (2023), p. 101575.
- [16] D. KWON, Y. FAN, AND K. LEE, *Score-based generative modeling secretly minimizes the Wasserstein distance*, Adv. Neural Inf. Process. Syst., 35 (2022), pp. 20205–20217.
- [17] J. LATZ, J. P. MADRIGAL-CIANCI, F. NOBILE, AND R. TEMPONE, *Generalized parallel tempering on Bayesian inverse problems*, Stat. Comput., 31 (2021), p. 67.
- [18] W. LI, S. LIU, AND S. OSHER, *A kernel formula for regularized Wasserstein proximal operators*, arXiv:2301.10301, (2023).
- [19] Y.-A. MA, N. S. CHATTERJI, X. CHENG, N. FLAMMARION, P. L. BARTLETT, AND M. I. JORDAN, *Is there an analog of Nesterov acceleration for gradient-based MCMC?*, Bernoulli, 27 (2021), pp. 1942 – 1992.
- [20] Y.-A. MA, Y. CHEN, C. JIN, N. FLAMMARION, AND M. I. JORDAN, *Sampling can be faster than optimization*, Proc. Natl. Acad. Sci. U.S.A., 116 (2019), pp. 20881–20885.
- [21] D. J. MACKAY, *Information theory, inference and learning algorithms*, Cambridge University Press, 2003.
- [22] D. MAOUTSA, S. REICH, AND M. OPPER, *Interacting particle solutions of Fokker–Planck equations through gradient–log–density estimation*, Entropy, 22 (2020).
- [23] K. L. MENGENSEN AND R. L. TWEEDIE, *Rates of convergence of the Hastings and Metropolis algorithms*, Ann. Stat., 24 (1996), pp. 101–121.
- [24] I. OSELEDETS AND E. TYRTYSHNIKOV, *TT-cross approximation for multidimensional arrays*, Linear Algebra Its Appl., 432 (2010), pp. 70–88.
- [25] I. V. OSELEDETS, *Tensor-train decomposition*, SIAM J. Sci. Comput., 33 (2011), pp. 2295–2317.
- [26] G. PARISI, *Correlation functions and computer simulations*, Nucl. Phys. B., 180 (1981), pp. 378–384.
- [27] Z. QIN, A. LIDIAK, Z. GONG, G. TANG, M. B. WAKIN, AND Z. ZHU, *Error analysis of tensor-train cross approximation*, Adv. Neural Inf. Process. Syst., 35 (2022), pp. 14236–14249.
- [28] L. RICHTER, L. SALLANDT, AND N. NÜSKEN, *Solving high-dimensional parabolic PDEs using the tensor train format*, in Int. Conf. Mach. Learn., PMLR, 2021, pp. 8998–9009.
- [29] P. B. ROHRBACH, S. DOLGOV, L. GRASEDYCK, AND R. SCHEICHL, *Rank bounds for approximating Gaussian densities in the tensor-train format*, SIAM-ASA J. Uncertain. Quantif., 10 (2022), pp. 1191–1224.
- [30] C. SCHILLINGS AND C. SCHWAB, *Sparse, adaptive Smolyak quadratures for Bayesian inverse problems*, Inv. Prob., 29 (2013), p. 065011.
- [31] Y. SONG AND S. ERMON, *Generative modeling by estimating gradients of the data distribution*, Adv. Neural Inf. Process. Syst., 32 (2019).
- [32] A. M. STUART, *Inverse problems: A Bayesian perspective*, Acta Numer0, 19 (2010), pp. 451–559.
- [33] H. Y. TAN, S. OSHER, AND W. LI, *Noise-free sampling algorithms via regularized Wasserstein proximals*, 2023.
- [34] A. TJANDRA, S. SAKTI, AND S. NAKAMURA, *Compressing recurrent neural network with tensor train*, in 2017 International Joint Conference on Neural Networks, IEEE, 2017, pp. 4451–4458.
- [35] G. VIDAL, *Efficient classical simulation of slightly entangled quantum computations*, Physical Review Letters, 91 (2003), p. 147902.
- [36] Y. WANG, P. CHEN, AND W. LI, *Projected Wasserstein gradient descent for high-dimensional Bayesian inference*, SIAM-ASA J. Uncertain. Quantif., 10 (2022), pp. 1513–1532.
- [37] Y. WANG AND W. LI, *Accelerated information gradient flow*, J. Sci. Comput., 90 (2022), pp. 1–47.
- [38] T. XIFARA, C. SHERLOCK, S. LIVINGSTONE, S. BYRNE, AND M. GIROLAMI, *Langevin diffusions and the Metropolis-adjusted Langevin algorithm*, Stat. Probab. Lett., 91 (2014), pp. 14–19.

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF CALIFORNIA, LOS ANGELES

*Email address:* `fghan@math.ucla.edu`

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF CALIFORNIA, LOS ANGELES, LOS ANGELES, CA, USA.

*Email address:* `sjomath@math.ucla.edu`

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF SOUTH CAROLINA, COLUMBIA, SC, USA.

*Email address:* `wuchen@mailbox.sc.edu`