

SOLVING HIGH-DIMENSIONAL KOLMOGOROV BACKWARD EQUATIONS WITH FUNCTIONAL HIERARCHICAL TENSOR OPERATORS*

XUN TANG[†], LEAH COLLIS[†], AND LEXING YING[‡]

Abstract. Solving high-dimensional partial differential equations necessitates methods free of exponential scaling in the dimension of the problem. This work introduces a tensor network approach for the Kolmogorov backward equation via approximating directly the Markov operator. We show that the high-dimensional Markov operator can be obtained under a functional hierarchical tensor (FHT) ansatz with a hierarchical sketching algorithm. When the terminal condition admits an FHT ansatz, the proposed operator outputs an FHT ansatz for the PDE solution through an efficient functional tensor network contraction procedure. In addition, the proposed operator-based approach also provides an efficient way to solve the Kolmogorov forward equation when the initial distribution is in an FHT ansatz. We apply the proposed approach successfully to two challenging time-dependent Ginzburg–Landau models with hundreds of variables.

Key words. high-dimensional Kolmogorov backward equation, semigroup method, functional tensor network, hierarchical tensor network, curse of dimensionality

MSC codes. 65M99, 15A69, 65F99

DOI. 10.1137/24M1654075

1. Introduction. This paper focuses on solving the Kolmogorov backward equation (KBE):

$$(1.1) \quad \begin{cases} \partial_t u - \nabla V \cdot \nabla u + \frac{1}{\beta} \Delta u = 0, & x \in \Omega \subset \mathbb{R}^d, t \in [0, T), \\ u(y, T) = f(y), & y \in \Omega \subset \mathbb{R}^d, \end{cases}$$

where V is a potential function and β is the inverse temperature. The solution u can also be represented as $u(x, t) = \mathbb{E}[f(X_T) \mid X_t = x]$, where X_t follows the Langevin dynamics

$$(1.2) \quad dX_t = -\nabla V(X_t)dt + \sqrt{2\beta^{-1}}dB_t.$$

In this work, we focus on the high-dimensional setting, where the domain has hundreds of dimensions. One prevalent type of high-dimensional Kolmogorov backward equation comes from the discretization of an infinite-dimensional Markov chain into finite but very large dimensions. Similar to the prior work in [32], we focus on the motivating example of a time-dependent Ginzburg–Landau (G–L) model used for studying the phenomenological theory of superconductivity [12, 16, 17, 10]. The 2D G–L model considers a particle taking the form of a field $x(a) : [0, 1]^2 \rightarrow \mathbb{R}$. The potential function is a functional $V(x)$ defined as follows:

*Submitted to the journal’s Numerical Algorithms for Scientific Computing section April 15, 2024; accepted for publication (in revised form) November 20, 2024; published electronically March 3, 2025.
<https://doi.org/10.1137/24M1654075>

Funding: This work is partially supported by NSF grants DMS-2011699 and DMS-2208163.

[†]Institute for Computational and Mathematical Engineering, Stanford University, Stanford, CA 94305 USA (xuntang@stanford.edu, lcollis@stanford.edu).

[‡]Department of Mathematics and Institute for Computational and Mathematical Engineering, Stanford University, Stanford, CA 94305 USA (lexing@stanford.edu).

$$(1.3) \quad V(x) = \frac{\lambda}{2} \int_{[0,1]^2} |\nabla_a x(a)|^2 da + \frac{1}{4\lambda} \int_{[0,1]^2} |1 - x(a)|^2 da,$$

where λ is a parameter balancing the relative importance of the first and second terms in (1.3). Under discretization, one decomposes the unit square $[0, 1]^2$ into a grid of $d = m^2$ points $\{(ih, jh)\}$ for $h = \frac{1}{m+1}$ and $1 \leq i, j \leq m$. The discretized field is the d -dimensional vector $x = (x_{(i,j)})_{1 \leq i, j \leq m}$, where $x_{(i,j)} = x(ih, jh)$. The associated discretized potential energy is defined as

$$(1.4) \quad V(x) = V(x_{(1,1)}, \dots, x_{(m,m)}) := \frac{\lambda}{2} \sum_{v \sim w} \left(\frac{x_v - x_w}{h} \right)^2 + \frac{1}{4\lambda} \sum_v (1 - x_v^2)^2,$$

where v and w are Cartesian grid points and $v \sim w$ if and only if they are adjacent. For any $f: \mathbb{R}^d \rightarrow \mathbb{R}$, the evolution of the conditional expectation with f as a terminal condition constitutes the Kolmogorov backward equation (1.1) with the discretized potential function $V(x)$. Even a moderate discretization scheme leads to a high-dimensional problem.

1.1. Background and contribution.

Existing methods for solving the Kolmogorov backward equation. While the traditional finite difference [6, 35] and finite element schemes [20] are candidate numerical solvers for the Kolmogorov backward equation, they cannot be used for the high-dimensional case due to the curse of dimensionality. More recent work [4, 3] shows that one can overcome the curse of dimensionality by combining Monte Carlo methods with a downstream regression task. Essentially, due to the conditional expectation formula, one can use Monte Carlo to approximately query function evaluation of u at a sufficiently large number of points in $\Omega \times [0, T]$. After obtaining the training sample, one can use the approximate value of u at these points as training data for a neural network regression task.

While the regression-based perspective overcomes the curse of dimensionality, its implementation faces two major challenges. First, the method incurs a significant cost to prepare the training target, as a large number of SDE simulations is required for each training sample. Second, retraining of the neural network is unavoidable when one considers a different terminal condition for $u(\cdot, T)$.

Markov operator for solving the Kolmogorov backward equation. In this work, we propose to solve the Kolmogorov backward equation by approximating the solution operator [22, 11, 19], i.e., obtaining a mapping from the terminal condition f to the solution u . For any $t \in [0, T]$ and a stochastic process $(X_s)_{s \in [0, T]}$ satisfying (1.2), one has the following equation:

$$(1.5) \quad u(x, T - t) = \int_{\Omega} \mathbb{P}[X_T = y \mid X_{T-t} = x] f(y) dy = \int_{\Omega} \mathbb{P}[X_t = y \mid X_0 = x] f(y) dy,$$

where the second equality of (1.5) is due to the time invariance of the dynamics for (1.2). Thus, the solution operator $G_t: \Omega \times \Omega \rightarrow \mathbb{R}$ is simply the conditional density defined by

$$(1.6) \quad G_t(x, y) := \mathbb{P}[X_t = y \mid X_0 = x],$$

which is typically referred to as the *Markov operator*.

Our proposed approach to approximate the Markov operator is motivated by Bayes's rule. For any chosen initial distribution $X_0 \sim p_0$, one has

$$(1.7) \quad G_t(x, y) = \frac{1}{p_0(x)} \mathbb{P}[X_0 = x, X_t = y],$$

which shows that the conditional density G_t can be characterized through the probability density of the joint variable (X_0, X_t) , which is an important observation underlying our subsequent tensor network approach to represent the Markov operator.

Functional hierarchical tensor operator. Functional hierarchical tensor (FHT) [32, 13, 14] is a class of parametric functions suitable for reduced-order modeling. While a detailed introduction can be found in [32], we mention a few of its appealing properties. The FHT ansatz scales linearly in dimension for memory requirement, function evaluation, moment calculation, and sampling (when used as an ansatz for density). Most importantly, the FHT ansatz scales linearly in dimension to calculate marginal distribution density and conditional distribution density, which sets it apart from common neural network architectures for density estimation.

Due to the aforementioned property, one can use an FHT ansatz to represent the probability density of the joint variable (X_0, X_t) , which leads to an FHT-based approximation to the operator G_t . The learned operator allows an efficient way to obtain the approximate value of $u(x, T - t)$ for arbitrary point $x \in \Omega$ and terminal condition f . When f has an FHT structure, (1.5) allows for more efficient computation of $u(x, T - t)$ through a tensor diagram contraction.

Density estimation through hierarchical sketching. Obtaining the joint density distribution is done by using a modification of an FHT-based approach in solving the Fokker–Planck equations [32]. After running particle simulations over the Langevin dynamics in (1.2), one collects the value of the particles at the initial time 0 and the intermediate time t . Then, one performs a hierarchical sketch-based density estimation subroutine [25, 32] on samples of the joint variable (X_0, X_t) . The outputted joint density is used to represent the Markov operator according to (1.7).

Compared to the approach in [32], the hierarchical sketching subroutine in this instance has new challenges, such as the design of the hierarchical bipartition structure and the functional basis. Using sophisticated numerical techniques, we are able to solve the Ginzburg–Landau model with d in the hundreds.

1.2. Related work.

Neural network methods for density estimation. As the Markov operator is obtained through density estimation, our work is related to generative learning algorithms for which density estimation is possible. Representing G_t with the joint distribution function for (X_0, X_t) is related to density estimation. Examples include normalizing flows [31, 26], energy-based models [15, 21], and the more recent diffusion models [30, 29]. The main challenge in obtaining the Markov operator through the aforementioned works lies in the calculation of the normalization constant and high-dimensional integration through (1.5). On the other hand, representing G_t directly as a conditional distribution is related to probabilistic imputation methods [34, 7, 8] and conditional generative modeling [24, 28, 33, 5, 23]. For neural network-based density estimation tasks, the training associated with a neural network might be slow, whereas our sketch-based approach relies on simple linear algebraic subroutines and does not involve an iterative training component.

Monte Carlo methods. As one has access to trajectories of the particle dynamics due to Monte Carlo sampling, it is possible to perform sampling to obtain the value of u at arbitrary points. Our proposed sketching approach approximates the Markov operator through data assimilation of one single batch of particle simulations, while the Monte Carlo approach requires a new set of particle simulations for every new point queried for the PDE solution u . Thus, the Monte Carlo approach is less suitable when one is interested in the solution at more than a few points or when one is interested in the global properties of the solution.

Tensor network methods. We refer the reader to [32] for a detailed discussion on tensor network-based approaches for solving high-dimensional PDEs. It is worth mentioning that a sketch-based density estimation subroutine for the functional tensor train (FTT) has been developed in [18]. The proposed approach of this work can be similarly adapted to approximate the Markov operator with an FTT as the tensor network architecture. However, the FTT ansatz is less capable of capturing the global correlation structure, which is why this work instead focuses on the FHT ansatz. It is also possible to use the regression-based perspective mentioned in section 1.1 with the functional tensor network as the parameterized function class for performing the regression task, and a similar approach has been taken in [27, 9].

1.3. Contents and notation. We outline the structure of the remainder of the paper. Section 2 details the functional hierarchical tensor structure for approximating the Markov operator. Section 3 goes through the algorithm for solving the Kolmogorov backward equation with the functional hierarchical tensor. Section 4 details the numerical implementations and results from solving Ginzburg–Landau models in 1D and 2D.

For notational compactness, we introduce several shorthand notations for simple derivation. For $n \in \mathbb{N}$, let $[n] := \{1, \dots, n\}$. For an index set $S \subset [d]$, we let x_S stand for the subvector with entries from index set S . We use \bar{S} to denote the set-theoretic complement of S , i.e., $\bar{S} = [d] - S$.

2. Tensor network structure for the Markov operator. To address the Ginzburg–Landau and general models coming from discretizing infinite-dimensional Markov chains, we propose to represent the Markov operator via a functional hierarchical tensor ansatz based on a binary-tree-based low-rank structure. In this section, we go over the functional hierarchical tensor and the tensor network structure for the Markov operator. Importantly, to simplify the discussion, we discuss obtaining the Markov operator for a fixed $t \in [0, T]$. The symbol d is reserved for the dimension of the state-space variable, and the symbol N is reserved for the number of samples.

2.1. Functional hierarchical tensor. For completeness, we give a short introduction to the functional hierarchical tensor ansatz, which largely follows the more detailed exposition in [32]. We first describe the *functional tensor network* (FTN) representation of a $2d$ -dimensional function $g : \mathbb{R}^{2d} \rightarrow \mathbb{R}$. Let $\{\psi_{i;j}\}_{i=1}^n$ denote a collection of orthonormal function basis over a single variable for the j th coordinate, and let $C \in \mathbb{R}^{n^{2d}}$ be the tensor represented by a tensor network. The functional tensor network is the $2d$ -dimensional function defined by the following equation:

$$(2.1) \quad g(x) \equiv g(z_1, \dots, z_{2d}) = \sum_{i_1, \dots, i_{2d}=0}^{n-1} C_{i_1, \dots, i_{2d}} \psi_{i_1;1}(z_1) \cdots \psi_{i_{2d};2d}(z_{2d}) = \left\langle C, \bigotimes_{j=1}^{2d} \vec{\Psi}_j(z_j) \right\rangle,$$

where $\vec{\Psi}_j(z_j) = [\psi_{1;j}(z_j), \dots, \psi_{n;j}(z_j)]$ is an n -vector encoding the evaluation for the z_j variable over the entire j th functional basis. In particular, g is a *functional hierarchical tensor* when C is represented by a hierarchical tensor network ansatz.

The hierarchical tensor network is characterized by a hierarchical bipartition of the variable set. Without loss of generality, let $d = 2^L$ so that the variable set admits exactly $L + 1$ levels of variable bipartition. As illustrated in Figure 1(a), at the l th level, the variable index set is partitioned according to

$$(2.2) \quad [2d] = \bigcup_{k=1}^{2^l} I_k^{(l)}, \quad I_k^{(l)} := \{2^{L-l+1}(k-1) + 1, \dots, 2^{L-l+1}k\},$$

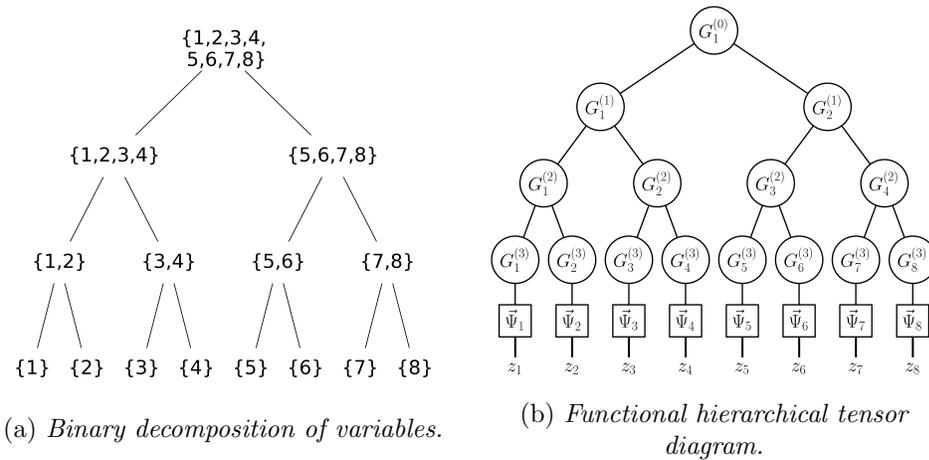


FIG. 1. Illustrations of functional hierarchical tensor with $d = 4$.

which in particular implies the recursive relation that $I_k^{(l)} = I_{2k-1}^{(l+1)} \cup I_{2k}^{(l+1)}$.

Given the hierarchical bipartition, the tensor network structure of a hierarchical tensor network has the same tree structure as in the variable bipartition, which is illustrated by Figure 1. Moreover, as Figure 1 shows, each internal bond of the tensor network is associated with an edge in the binary decomposition of the variables. Each physical bond of the tensor network connects a leaf node with a $\tilde{\Psi}_j(z_j)$ node. In summary, the graphical structure of the hierarchical tensor network is formed by a binary tree, and the physical index is at the leaf node. In terms of runtime complexity and memory complexity, the functional hierarchical tensor enjoys the same $O(d)$ scaling as that of a hierarchical tensor [25].

2.2. Main idea for the FHT representation of the Markov operator. We now describe the functional hierarchical tensor representation of the Markov operator. Let the initial distribution for $(X_s)_{s \in [0, T]}$ satisfy $X_0 \sim p_0$. By Bayes's rule and time invariance of (1.2), the Markov operator $G_t = \mathbb{P}[X_t = y \mid X_0 = x]$ satisfies the equation

$$G_t(x, y) := \frac{1}{p_0(x)} \mathbb{P}_{X_0 \sim p_0} [X_0 = x, X_t = y],$$

where $X_0 \sim p_0$ means the stochastic process $(X_s)_{s \in [0, T]}$ follows from the Langevin dynamics in (1.2) with initial condition $X_0 \sim p_0$.

In practice, one solves the KBE solution u inside a bounded domain $\mathcal{X} \subset \Omega$, and correspondingly one considers the value of $G_t(x, y)$ inside $x \in \mathcal{X}$. Thus, one can set p_0 as the uniform distribution on \mathcal{X} . Under this construction, one has

$$(2.3) \quad G_t(x, y) = \text{Vol}(\mathcal{X}) \mathbb{P}[X_0 = x, X_t = y], \quad x \in \mathcal{X}.$$

In particular, the construction of G_t in (2.3) implies solving the Markov operator can be reduced to density estimation over the joint variable (X_0, X_t) . Thus, one can employ the sketch-based density estimation subroutine in [32] to estimate the probability distribution of (X_0, X_t) as a functional hierarchical tensor ansatz, which gives the Markov operator an FHT representation. The rest of this section gives implementation details of this approach.

2.3. Network structure of the Markov operator. Representation of the Markov operator in the FHT ansatz leads to the question of the hierarchical tensor network structure for the $2d$ -dimensional probability density function

$$g(x, y) = \mathbb{P}_{X_0 \sim p_0} [X_0 = x, X_t = y].$$

We propose to use the interlacing scheme to order the joint variable. For $x = (x_1, \dots, x_d)$ and $y = (y_1, \dots, y_d)$, the interlacing scheme orders the joint variable $z = (z_1, \dots, z_{2d})$ by $z = (x_1, y_1, \dots, x_d, y_d)$. Typically, the variables x, y are ordered so that the correlation for $(x_j, x_{j'})$ and $(y_j, y_{j'})$ are strong only when the index j is close to j' . Thus, capturing the model's spatial correlation structure means the ordering of the joint variable needs to preserve the proximity. Moreover, x_j and y_j represent the value of a high-dimensional stochastic process at the same site. Therefore, capturing the temporal correlation structure means the ordering of the joint variable needs to be such that x_j is placed close to y_j . The proposed interlacing scheme preserves the spatial and temporal correlation structure, as is desired.

An ordering of the variables leads to a corresponding hierarchical bipartition through (2.2). In Figure 2, we illustrate the resultant variable hierarchical bipartition in the 2D Ginzburg–Landau (G–L) model under the proposed interlacing scheme. The 2D G–L model has the natural geometry of a 2D grid, and we apply the natural construction of alternatively partitioning the x, y variables along the two axes of the 2D grid. One can see that the proposed bipartition structure for the joint variable indeed captures important correlation structures of the problem.

2.4. Functional basis for the FTN ansatz. The choice of the functional basis is intricately linked to our choice of the bounded domain $x \in \mathcal{X}$. In this work,

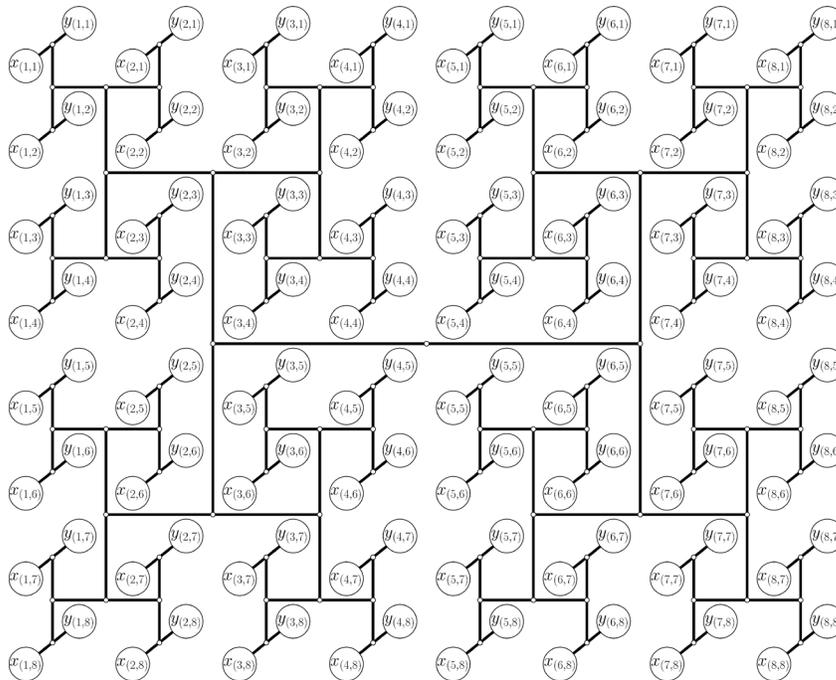


FIG. 2. Illustration of the hierarchical bipartition of a 2D Ginzburg–Landau model discretized to an 8×8 grid. The variable $x = (x_{(i,j)})_{1 \leq i,j \leq m}$ denotes the random variable at the initial time 0, and $y = (y_{(i,j)})_{1 \leq i,j \leq m}$ denotes the random variable at the intermediate time $T - t$.

we choose a simple hypercube structure for the domain, i.e., $\mathcal{X} = \prod_{j \in [d]} [a_j, b_j]$. This construction is beneficial in three important aspects. First, any bounded region one is interested in can always be covered by a hypercube, and so any domain \mathcal{X} can always be reduced to the hypercube case. Second, the simplicity in \mathcal{X} ensures the initial distribution p_0 has a simple sampler during particle simulation. Most importantly, the separable structure of \mathcal{X} avoids the occurrence of the failure mode where spurious correlation from the irregularity of the domain prevents the joint density for (X_0, X_t) from having a low-rank structure.

The choice of functional basis is a direct result of the hypercube structure $\mathcal{X} = \prod_{j \in [d]} [a_j, b_j]$. As a minimal requirement, the chosen functional basis for the x variable needs to allow the functional hierarchical tensor to faithfully represent the uniform distribution on \mathcal{X} . Thus, for each x_j variable, we pick the functional basis to be a collection of the orthonormal basis of $L^2([a_j, b_j])$ with the mild requirement that its span contains the constant function, i.e., the indicator function $\chi_{[a_j, b_j]}$.

In practice, it is cumbersome to maintain a separate function basis for each variable. For our implementation, we use a fixed orthonormal basis $\{\psi_i\}_{i=1}^n \subset L^2([-1, 1])$ with $\psi_1 = \frac{1}{\sqrt{2}}\chi_{[-1, 1]}$. The function basis for x_j can be constructed by applying an appropriate linear change of coordinate to $\{\psi_i\}_{i=1}^n$ to obtain a basis in $L^2([a_j, b_j])$. In the Ginzburg–Landau model, the distribution of X_t is effectively compactly supported inside the hypercube $[-2, 2]^d$, which is why the functional basis for the y variable is also obtained by change of coordinate to $\{\psi_i\}_{i=1}^n$.

2.5. Functional hierarchical sketching for joint density. As the construction for the variable hierarchical bipartition structure and the basis are fixed, one can apply the functional hierarchical tensor sketching algorithm in [32] to obtain an FHT-based characterization of the joint density function for (X_0, X_t) . The details of the sketching algorithm can be found in [32]. The complexity of the sketching algorithm has a runtime complexity of $O(Nd)$, where N is the sample size, and d is the state-variable dimension.

We remark that the generalization to a variable-dependent function basis requires only a straightforward modification to the subroutine. Let $\{w^{(i)} := (w_1^{(i)}, \dots, w_{2d}^{(i)})\}_{i=1}^N$ be the samples of the joint variable (X_0, X_t) ordered according to the proposed interlacing scheme. In our case, we choose to obtain the variable-dependent functional basis $\{\psi_{i;j}\}_{i=1}^n$ through the aforementioned linear change of coordinate on a fixed basis function $\{\psi_i\}_{i=1}^n \subset L^2([-1, 1])$. Thus, it is mathematically equivalent to apply a change of coordinate to the received sample $\{w^{(i)}\}_{i=1}^N$ and then perform the sketch-based density estimation algorithm with the variable-invariant functional basis $\{\psi_i\}_{i=1}^n$, which is the setting in [32]. In general, to accommodate the variable-dependent function basis, the modified sketching procedure would use $\{\psi_{i;j}\}_{i=1}^n$ for the functional basis evaluation on the j th coordinate of each sample.

To give the reader some intuition on the approach, we give a condensed description of the main procedure underlying the sketch-based algorithm for the $2d$ -dimensional joint density function for (X_0, X_t) . For the reader’s convenience, relevant equations are included in Figure 3 in terms of the tensor diagram.

Let the tensor cores for the joint density g be denoted by $\{P_k^{(l)}\}_{k \in [2^l], l=0, \dots, L+1}$, where $P_k^{(l)}$ is the k th tensor core at level l . For simplicity, we outline the procedure for solving for the tensor core at intermediate levels with $0 < l < L + 1$.

Using the notation for hierarchical bipartition as in (2.2), write $a := I_{2k}^{(l+1)}$, $b := I_{2k}^{(l+1)}$ and $f = [2d] - a \cup b$. In this case, $P_k^{(l)}$ is a 3-tensor $P_k^{(l)} : [r_a] \times [r_b] \times [r_f] \rightarrow \mathbb{R}$ for some $r_a, r_b, r_f \in \mathbb{N}$. The structural low-rankness property of g in (2.1) implies that

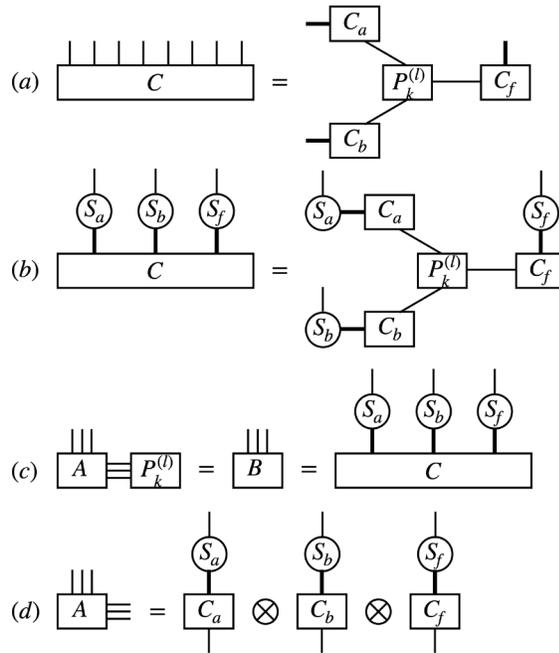


FIG. 3. Tensor diagram representation of main equations used in the hierarchical sketching algorithm. Equation (2.4) is summarized in (a). Equation (2.5) is shown in (b) and (c). The coefficient term in (2.5) is shown in (d).

there exist $C_a : [n^{|a|}] \times [r_a] \rightarrow \mathbb{R}$, $C_b : [n^{|b|}] \times [r_b] \rightarrow \mathbb{R}$, $C_f : [r_f] \times [n^{|f|}] \rightarrow \mathbb{R}$ such that the following equation holds (illustrated in Figure 3(a)):

$$(2.4) \quad C(i_1, \dots, i_{2d}) = \sum_{\alpha, \beta, \theta} C_a(i_a, \alpha) C_b(i_b, \beta) P_k^{(l)}(\alpha, \beta, \theta) C_f(\theta, i_f).$$

Equation (2.4) is exponential-sized, and thus, one needs to use sketching to solve a system of smaller size. The hierarchical sketching algorithm essentially solves the overdetermined linear system (2.4) for $P_k^{(l)}$ with the use of sketch functions. Let $\tilde{r}_a, \tilde{r}_b, \tilde{r}_f$ be integers so that $\tilde{r}_a > r_a, \tilde{r}_b > r_b, \tilde{r}_f > r_f$. Through the sketch functions $S_a : [n^{|a|}] \times [\tilde{r}_a] \rightarrow \mathbb{R}, S_b : [n^{|b|}] \times [\tilde{r}_b] \rightarrow \mathbb{R}, S_f : [n^{|f|}] \times [\tilde{r}_f] \rightarrow \mathbb{R}$, one can contract the linear system in (2.4) with S_a, S_b, S_f at the variables i_a, i_b, i_f , respectively, which leads to the following sketched linear system (illustrated in Figure 3(b)–(d)):

$$(2.5) \quad B(\mu, \nu, \zeta) = \sum_{\alpha, \beta, \theta} A_a(\mu, \alpha) A_b(\nu, \beta) A_f(\zeta, \theta) P_k^{(l)}(\alpha, \beta, \theta),$$

where A_a, A_b, A_f are respectively the contraction of C_a, C_b, C_f by S_a, S_b, S_f and B is the contraction of C by $S_a \otimes S_b \otimes S_f$. The equation (2.5) can be seen as the following linear system of $\tilde{r}_a \tilde{r}_b \tilde{r}_f$ equations for the unknown $P_k^{(l)}$:

$$(A_a \otimes A_b \otimes A_f) P_k^{(l)} = B.$$

In practice, one has access to g through the received sample $\{w^{(i)}\}_{i=1}^N$. Through somewhat involved derivations, which are outlined in [32], one shows that B, A_a, A_b, A_f can be obtained by calculating statistical moments of the distribution g , which can

be done efficiently through sample-based moment estimation. Thus, the hierarchical sketching algorithm systematically collects statistical moments and subsequently solves tensor cores by solving independent linear systems.

3. Solving for the Kolmogorov backward equations. In this section, we show how the Markov operator gives rise to an approximated solution to the Kolmogorov backward equation. Section 3.1 summarizes the main workflow used to solve the Kolmogorov backward equation. Section 3.2 discusses solving for the Kolmogorov backward equation when the terminal condition admits an FHT ansatz and comments on the case for the general terminal condition. Section 3.3 discusses solving for the Kolmogorov forward equation and solving Kolmogorov equations under general SDE dynamics.

3.1. Main workflow. We summarize our approach to solving the Kolmogorov backward equation. The given information regarding the PDE includes the potential function $V : \mathbb{R}^d \rightarrow \mathbb{R}$, a hypercube region \mathcal{X} , and terminal time T . The main workflow is summarized in the following simple steps:

1. Select K time steps $0 = t_0 \leq t_1 \leq \dots \leq t_K = T$. Then, with a chosen numerical scheme (e.g., Euler–Maruyama), run N independent instances of SDE simulations on the stochastic dynamic equation (1.2) to time T with the initial condition $X_0 \sim \text{Unif}(\mathcal{X})$. Obtain N trajectories and record the trajectory data $\{X_{t_j}^{(i)}\}_{j \in [K], i \in [N]}$.
2. For $j = 1, \dots, K$, obtain an approximated joint density $g_j(x, y) \approx \mathbb{P}[X_0 = x, X_{t_j} = y]$ by calling the sketch-based density estimation algorithm on the sample $\{(X_0^{(i)}, X_{t_j}^{(i)})\}_{i \in [N]}$. Obtain the Markov operator G_t at $t = t_j$ through $G_{t_j}(x, y) = \text{Vol}(\mathcal{X})g_j(x, y)$.
3. For any (x, t) , output the continuous-in-time solution for $u(x, t)$, defined as follows: Suppose that $t \in [0, T]$ satisfies $T - t \in (t_j, t_{j+1})$; then $u(x, t)$ can be approximated with $u_j(x)$ and $u_{j+1}(x)$ via an appropriate interpolation. The values of $u_j(x)$ and $u_{j+1}(x)$ are obtained with the corresponding Markov operator. The details are in section 3.2.

3.2. FHT ansatz for the solution. After approximating the Markov operator $G_t(x, y)$ on $x \in \mathcal{X}$ via the sketch-based density estimation, one obtains the value of $u(x, T - t)$ for $x \in \mathcal{X}$ through the following equation. Let g_{FHT} denote the functional hierarchical tensor ansatz for the joint density obtained by the sketch-based density estimation algorithm. One then has

$$(3.1) \quad u(x, T - t) = \int_{\Omega} G_t(x, y)f(y) dy \approx \text{Vol}(\mathcal{X}) \int_{\Omega} g_{\text{FHT}}(x, y)f(y) dy, \quad x \in \mathcal{X}.$$

In this subsection, we discuss the special case in which the terminal condition f admits an FHT ansatz. In this case, we can indeed show that the solution $u(x, T - t)$ admits an FHT ansatz through an efficient tensor contraction.

We give two motivating examples in which f admits an FHT ansatz. First, when f is separable, i.e., when $f(y) = \prod_{j \in [d]} h_j(y_j)$, one can show that f is of rank one across any hierarchical bipartition. Second, when f is a sum of univariate functions, i.e., when $f(y) = \sum_{j \in [d]} h_j(y_j)$, one can show that f is of rank two across any hierarchical bipartition. For general terminal condition f , one can likewise apply an FHT-based approximation to the terminal condition. Discussion on the approximation of general functions by functional hierarchical tensors can be found in [1, 2] and references therein. At the end of this subsection, we give the procedure for obtaining the solution for general f without FHT approximations.

We now construct the FHT ansatz for the PDE solution $u \approx \text{Vol}(\mathcal{X}) \int_{\Omega} g_{\text{FHT}}(x, y) f(y) dy$. Suppose $d = 2^L$, where $L \geq 2$. We use the notational convention that $G_k^{(l)}$ denotes the tensor core of an FHT ansatz corresponding to the k th node at the l th level. When the node is not a root or leaf node, the tensor core $G_k^{(l)}$ is a 3-tensor and is otherwise a 2-tensor. The tensor indexing convention strictly follows that of [32].

Let the tensor core of the solved joint density g_{FHT} be denoted by $\{P_k^{(l)}\}_{k \in [2^l], l=0, \dots, L+1}$, and let the tensor core of the terminal condition f be denoted by $\{F_k^{(l)}\}_{k \in [2^l], l=0, \dots, L}$. We shall give an explicit FHT representation of u by a collection of tensor cores $\{U_k^{(l)}\}_{k \in [2^l], l=0, \dots, L}$. Indeed, as f and g_{FHT} are of compatible network structure, obtaining the FHT representation for $\int_{\Omega} g_{\text{FHT}}(x, y) f(y) dy$ is done through a simple tensor contraction. See Figure 4 for a tensor diagram illustration of the procedure.

As the illustrated tensor diagram indicates, to form the FHT ansatz for u , the construction is through combining the tensor cores from g_{FHT} and f . For the root level $l = 0$, we construct the tensor core $U_1^{(0)}$ as the Kronecker product between two 2-tensors $P_1^{(0)}$ and $F_1^{(0)}$, where in addition a coefficient term is applied to absorb the normalization constant from the volume of \mathcal{X} :

$$U_1^{(0)}((\alpha, \mu), (\beta, \nu)) = \text{Vol}(\mathcal{X}) P_1^{(0)}(\alpha, \beta) F_1^{(0)}(\mu, \nu).$$

For an intermediate level $l \notin \{0, L\}$ and an arbitrary block index $k \in [2^l]$, the tensor core $U_k^{(l)}$ is the 3-tensor formed by a tensor-equivalent of Kronecker product between the two 3-tensors $P_k^{(l)}$ and $F_k^{(l)}$:

$$U_k^{(l)}((\alpha, \mu), (\beta, \nu), (\theta, \xi)) = P_k^{(l)}(\alpha, \beta, \theta) F_k^{(l)}(\mu, \nu, \xi).$$

The construction for the leaf level is more involved, as one needs to perform tensor contraction over four tensor cores. In this case, for $k \in [d]$, one constructs the 2-tensor $U_k^{(L)}$ by the following formula (see illustration in Figure 4):

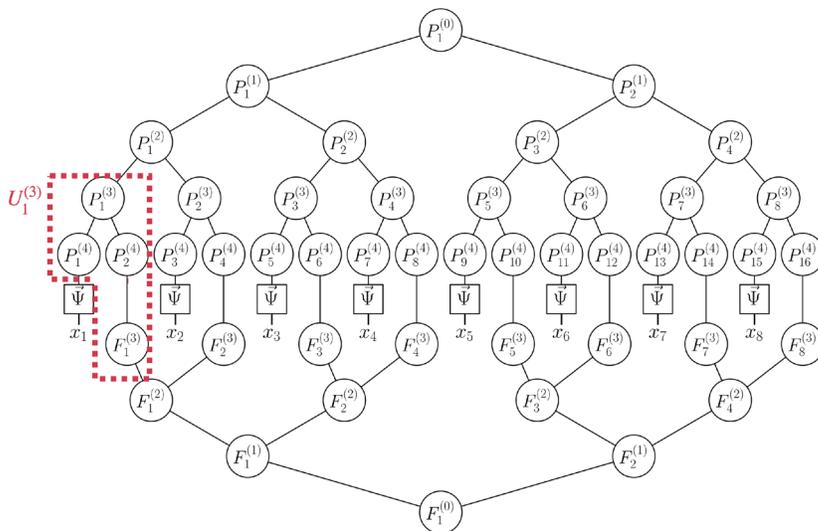


FIG. 4. Illustration of the tensor contraction diagram for the approximated solution to the Kolmogorov backward equation for $d = 8$.

$$U_k^{(L)}(i, (\theta, \xi)) = \sum_{i', \alpha, \beta} P_{2k-1}^{(L+1)}(i, \alpha) P_{2k}^{(L+1)}(i', \beta) P_k^{(L)}(\alpha, \beta, \theta) F_k^{(L)}(i', \xi).$$

We remark that the approach likewise generalizes to the situation when terminal condition f is a sum of functions admitting FHT ansatz. In that case, one can contract each FHT with the FHT approximation of the Markov operator, which outputs the approximated PDE solution as a sum of functions with the FHT ansatz.

For the case of a general terminal condition, one can also obtain an approximated PDE solution. In this case, one can use the fact that conditional distribution with a functional hierarchical tensor is efficient with only an $O(d)$ scaling. Thus, for $x \in \mathcal{X}$, one uses g_{FHT} to perform an approximate conditional sampling of the X_t condition on $X_0 = x$. After M samples $(y^{(i)})_{i=1}^M$ are collected, the solution to the PDE at $u(x, T-t)$ can be approximated by

$$u(x, T-t) \approx \frac{1}{M} \sum_{i=1}^M f(y^{(i)}).$$

In this case, the FHT-based approximated solution is more efficient than simulating the conditional distribution of X_t based on SDE simulation over the dynamics in (1.2). The reason is that the cost of calculating the drift term in the SDE simulation is proportional to the cost of evaluating the gradient of V , as well as the total number of times in which the gradient is called. In practical cases, even in the case where the cost of evaluating ∇V is $O(d)$ and $t = O(1)$, one would still keep a small step size for stability, which leads to a cost often much higher than the conditional sampling subroutine for FHT. Thus, the proposed approach can be quite beneficial when the cost of SDE simulation is much higher than the cost of conditional sampling.

3.3. Extensions.

FHT ansatz for solution to the Kolmogorov forward equation. After approximating the Markov operator $G_t(x, y)$ on $x \in \mathcal{X}$ via the sketch-based density estimation, one likewise obtains the solution to the Kolmogorov forward equation through tensor contraction. Assume that \mathcal{X} is sufficiently large such that the support of the initial distribution $q(x)$ is concentrated in \mathcal{X} except for a negligible volume. Then, the functional hierarchical tensor ansatz for the joint density outputs

$$p(y, t) = \int_{\Omega} G_t(x, y) q(x) dx \approx \text{Vol}(\mathcal{X}) \int_{\Omega} g_{\text{FHT}}(x, y) q(x) dx, \quad x \in \mathcal{X},$$

which is an approximate solution to the Kolmogorov forward equation with q as the initial distribution. We discuss the case in which q is in the form of an FHT ansatz or when q is characterized in terms of samples following the distribution governed by q . The latter case can be converted to the former case by using the sketch-based algorithm to obtain an FHT approximation to q via the samples. Let the tensor core of g_{FHT} be denoted by $\{P_k^{(l)}\}_{k \in [2^l], l=0, \dots, L+1}$, and let the tensor core of q be denoted by $\{Q_k^{(l)}\}_{k \in [2^l], l=0, \dots, L}$. The procedure of deriving the tensor core for the FHT approximation of p follows exactly as the tensor contraction in section 3.2, and thus we omit the details. The procedure is illustrated via a tensor diagram in Figure 5.

Kolmogorov equations beyond Langevin dynamics. Though we have been focusing on Langevin dynamics for simplicity, our method can be generalized to a wider range of SDE dynamics. In particular, one can consider a general particle system with the following form:

$$(3.2) \quad dX_t = f(X_t, t)dt + \sigma(X_t, t)dB_t,$$

Downloaded 04/30/25 to 171.66.13.217 . Redistribution subject to SIAM license or copyright; see https://pubs.siam.org/terms-privacy

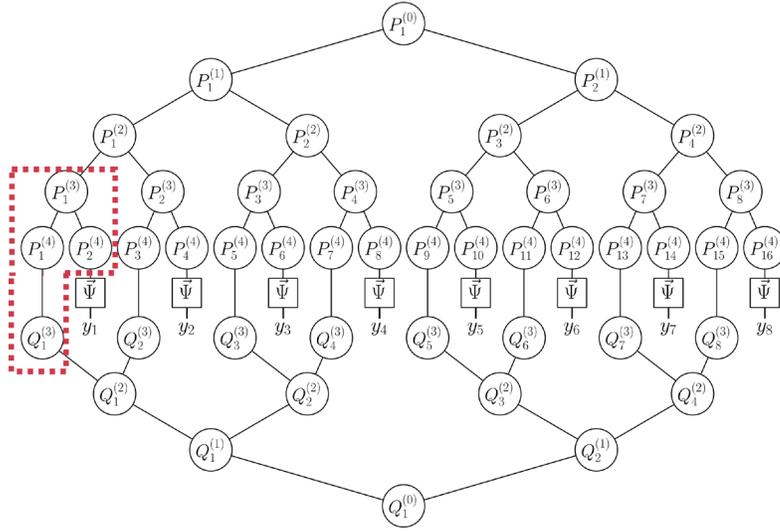


FIG. 5. Illustration of the tensor contraction diagram for the approximated solution to the Kolmogorov forward equation for $d = 8$.

where $f : \mathbb{R}^d \times [0, T] \rightarrow \mathbb{R}^d$ and $\sigma : \mathbb{R}^d \times [0, T] \rightarrow \mathbb{R}^{d \times d}$ are general advection and diffusion terms. The formula for the Kolmogorov forward equation is

$$\partial_t p = \nabla \cdot (pf) + \frac{1}{2} \sum_{i,j,k=1}^d \partial_{x_i x_j} (\sigma_{ik} \sigma_{jk} p), \quad x \in \Omega \subset \mathbb{R}^d, t \in [0, T].$$

In this case, solving for the forward equation requires only simple modifications to the proposed routine. In this case, for a hypercube region \mathcal{X} , one can sample $X_0 \sim \text{Unif}(\mathcal{X})$ and then perform the SDE simulation with (3.2). The Markov operator G_t can likewise be obtained through the probability density function via

$$G_t(x, y) = \text{Vol}(\mathcal{X}) \mathbb{P}_{X_0 \sim \text{Unif}(\mathcal{X})} [X_0 = x, X_t = y], \quad x \in \mathcal{X},$$

and the solution $p(y, t)$ can be obtained by contracting the Markov operator with the prescribed initial condition q .

The situation for the Kolmogorov backward equation is more involved. In this case, the PDE is

$$\partial_t u + \nabla u \cdot f + \frac{1}{2} \sum_{i,j,k=1}^d \sigma_{ik} \sigma_{jk} \partial_{x_i x_j} u = 0, \quad x \in \Omega \subset \mathbb{R}^d, t \in [0, T].$$

Importantly, the dynamics (3.2) is no longer time-invariant, and the relevant solution operator admits the representation

$$G_{t,T}(x, y) := \mathbb{P}[X_T = y | X_t = x] = \frac{1}{p_0(x)} \mathbb{P}[X_t = x, X_T = y].$$

In this case, obtaining the value of $u(x, T - t)$ at each $t \in [0, T]$ requires a separate set of SDE simulations. For a hypercube region \mathcal{X} , one can sample $X_t \sim \text{Unif}(\mathcal{X})$ and

then perform the SDE simulation with (3.2). The solution operator $G_{t,T}$ can likewise be obtained through the probability density function via

$$G_{t,T}(x, y) = \text{Vol}(\mathcal{X}) \mathbb{P}_{X_t \sim \text{Unif}(\mathcal{X})} [X_t = x, X_T = y], \quad x \in \mathcal{X}.$$

Once the solution operator is obtained, the solution can be similarly obtained either through conditional sampling or tensor contraction, depending on whether f admits an FHT-based representation.

Remark 3.1. The setting of this work admits a straightforward extension to the functional tensor train (FTT) ansatz. Similar to the case of FHT, we take the joint variable ordering $z = (x_1, y_1, \dots, x_d, y_d)$. Under this ordering, one can use the FTT ansatz to model the density function $g(x, y) = \mathbb{P}_{X_0 \sim p_0} [X_0 = x, X_t = y]$. Density estimation of FTT is a special case of density estimation of FHT, and implementation details can be found in [18]. Likewise, g can be used to give rise to an FTT representation of the Markov operator G_t .

4. Application to the Ginzburg–Landau model. We apply the proposed Markov operator method to a discretized Ginzburg–Landau model in 1D and 2D. For the sake of simplicity, we solve for the backward equation at $t = 0$. In general, solving for the backward equation solution at $t \in [0, T]$ can be done by running the algorithm at multiple time slots as prescribed in section 3.

4.1. Implementation details.

The choice of basis function. For the Ginzburg–Landau model, the $\int_a |1-x(a)|^2 da$ term ensures that the distribution of each entry of X_T is essentially bounded within the interval $[-2, 2]$. For simplicity, we consider the situation whereby one is interested in the value of the KBE solution u for $\mathcal{X} = [-2, 2]^d$. In the proposed method for the Markov operator estimation procedure, the sample is obtained by generating $X_0 \sim \text{Unif}([-2, 2]^d)$ and subsequently generating X_t by simulating the SDE in (1.2). Thus, the joint density function for the variable (X_0, X_t) can be modeled to have compact support in $[-2, 2]^{2d}$. Therefore, we use the polynomial representation, choosing a maximal degree parameter q and picking the first $n = q + 1$ Legendre basis polynomial in $[-2, 2]$, denoted $\{\psi_i\}_{i=0}^q$.

Bipartition structure. Similar to the hierarchical bipartition proposed in [32], we recursively bipartition the Cartesian grid along its axes in an alternating fashion. The structure for the two-dimensional case is shown in Figure 2 and can be done likewise to other dimensions. Explicitly, assume for simplicity that a function u in Δ -dimensions is discretized to the $d = m^\Delta$ points denoted $\{u_{i_1, \dots, i_\Delta}\}_{i_1, \dots, i_\Delta \in [m]}$ with $m = 2^\mu$. In the Markov operator case, one considers the discretization of u at initial and terminal time. Thus, one defines variables $\{x_{i_1, \dots, i_\Delta}, y_{i_1, \dots, i_\Delta}\}_{i_1, \dots, i_\Delta \in [m]}$, where $x_{i_1, \dots, i_\Delta}, y_{i_1, \dots, i_\Delta}$ respectively corresponds to the discretization of u at the initial and terminal time for the location (i_1, \dots, i_Δ) . For each index i_δ , one performs a length μ binary expansion $i_\delta = a_{\delta 1} \dots a_{\delta \mu}$. The variable $x_{i_1, \dots, i_\Delta}, y_{i_1, \dots, i_\Delta}$ is respectively given index $2k - 1, 2k$, where k has the length $\mu\Delta$ binary expansion $k = a_{11}a_{21} \dots a_{\Delta-1, \mu}a_{\Delta, \mu}$. The associated binary decomposition structure follows from the given index according to (2.2). One can check that the construction exactly corresponds to performing binary partition by sweeping along the Δ dimensions. For intuition on this construction, one can check that the bipartition in Figure 2 corresponds to the indexing $k = a_{11}a_{21}a_{12}a_{22}a_{13}a_{23}$.

The choice of sketch function. For the density estimation procedure under hierarchical sketching, the quality of the sketch function is critical to the fidelity of the Markov operator approximation. In our work, the sketch functions we choose are of two kinds. The first kind of sketch function consists of monomials of the Legendre

Downloaded 04/30/25 to 171.66.13.217 . Redistribution subject to SIAM license or copyright; see https://pubs.siam.org/terms-privacy

basis of low degree. For $a = I_k^{(l)}$ or $a = [2d] - I_k^{(l)}$ (see definition in (2.2)), the chosen sketch functions are of the following form:

$$f(z_a) = \prod_{j \in a} \psi_{i_j}(z_j)$$

for some choice of $0 \leq i_j \leq q$ for each j . One typically chooses f with small total degree defined by $\deg(f) = \sum_{j \in a} i_j$.

The second kind of sketch function is motivated by the renormalization group, in particular by the coarse-graining of the variables. In particular, one chooses a cluster $h \subset a$, and the resultant function g_h is the average over variables in h :

$$g_{h,i}(z_a) \equiv \frac{1}{|h|} \sum_{j \in h} z_j.$$

One can augment the sketch function set by including terms such as $g_{h,1}^2$. Empirically, including such coarse-grained functions greatly increases model performance and stability.

Sketch postprocessing. For efficient performance of the sketching operation, it is often advantageous to apply postprocessing to reduce the noise level for the sketching operation during its moment estimation tasks. In particular, an important subroutine of the tensor sketching operation boils down to performing the estimation of $\mathbb{E}[g(W)]$ for a function g and a $2d$ -dimensional random variable $W = (X_0, X_t)$. Because the variables within X_0 are pairwise independent, it turns out that one has $\mathbb{E}[g(W)] = 0$ in a large number of cases. Therefore, we apply a simple hard truncation rule whereby we estimate $\mathbb{E}[g(W)]$ to be zero when the sample mean for $g(W)$ is not larger than twice the estimated standard deviation for the sample mean, and otherwise we take the sample mean as the estimate. While this step can be skipped when one uses a large number of samples for the FHT density estimation subroutine, we observe great practical noise reduction by applying this approach.

Solution postprocessing. After the density estimation step is complete, one obtains the estimation of the joint density according to

$$\mathbb{P}[X_0 = x, X_t = y] \approx g_{\text{FHT}}(x, y),$$

and the Markov operator is estimated by $G_t(x, y) \approx \text{Vol}(\mathcal{X})g_{\text{FHT}}(x, y)$. In practice, it does not hold that $\int g_{\text{FHT}}(x, y) dy = 1/\text{Vol}(\mathcal{X})$, and one achieves higher accuracy by adopting the following approximation of G_t :

$$(4.1) \quad G_t(x, y) \approx \frac{g_{\text{FHT}}(x, y)}{\int_s g_{\text{FHT}}(x, s)}.$$

Correspondingly, the approximation of the backward solution is obtained by

$$(4.2) \quad u(x, T-t) = \int_{\Omega} G_t(x, y)f(y) dy \approx \frac{1}{\int_s g_{\text{FHT}}(x, s)} \int_{\Omega} g_{\text{FHT}}(x, y)f(y) dy, \quad x \in \mathcal{X}.$$

The formula in (4.2) is a division between two FHT-based functions, which is why (4.2) is not an FHT-based format for $u(x, T-t)$. To avoid this shortcoming, one can perform a simple FHT-based interpolation algorithm to obtain an FHT-based representation for u . When one has a function with a black-box access to its evaluation, the hierarchical sketching algorithm can be used as an interpolation algorithm, and we omit the details for the sake of simplicity.

4.2. 1D Ginzburg–Landau potential. We first consider a 1D Ginzburg–Landau model. The potential energy is defined as

$$(4.3) \quad V(x_1, \dots, x_m) := \frac{\lambda}{2} \sum_{i=1}^{m+1} \left(\frac{x_i - x_{i-1}}{h} \right)^2 + \frac{1}{4\lambda} \sum_{i=1}^m (1 - x_i^2)^2,$$

where $h = \frac{1}{m+1}$ and $x_0 = x_{m+1} = 0$. In particular, we fix $m = 128$, $\lambda = 0.005$, and $\beta = \frac{1}{20}$. The dimension is $d = 128$.

We perform $N = 60000$ SDE simulations with $T = 1$ and $\delta t = \frac{1}{1000}$ with initial distribution $p_0 = \text{Unif}([-2, 2]^d)$. The maximal Legendre degree q is taken to be $q = 20$, and the maximal internal bond dimension is $r = 10$.

As an initial inquiry into the quality of the estimated joint density function $\mathbb{P}[X_0 = x, X_T = y]$, Figure 6 compares the marginal distribution of (x_{15}, x_{20}) and (y_{15}, y_{20}) between empirical distributions and the model obtained by hierarchical tensor sketching. One can see that hierarchical tensor sketching can capture the independence between the x variables as well as the correlation between the y variables.

To assess the performance of the Markov operator in solving the backward equation, we propose to test the Markov operator with the following two terminal conditions:

$$(4.4) \quad f_+(y_1, \dots, y_d) = \frac{1}{d} \sum_{j=1}^d (y_j - 1)^2, \quad f_-(y_1, \dots, y_d) = \frac{1}{d} \sum_{j=1}^d (y_j + 1)^2,$$

where the solution to the backward equation corresponds to the expected squared Euclidean distance from the two metastable states $y_+ = (1, \dots, 1)$ and $y_- = (-1, \dots, -1)$ of the Ginzburg–Landau model.

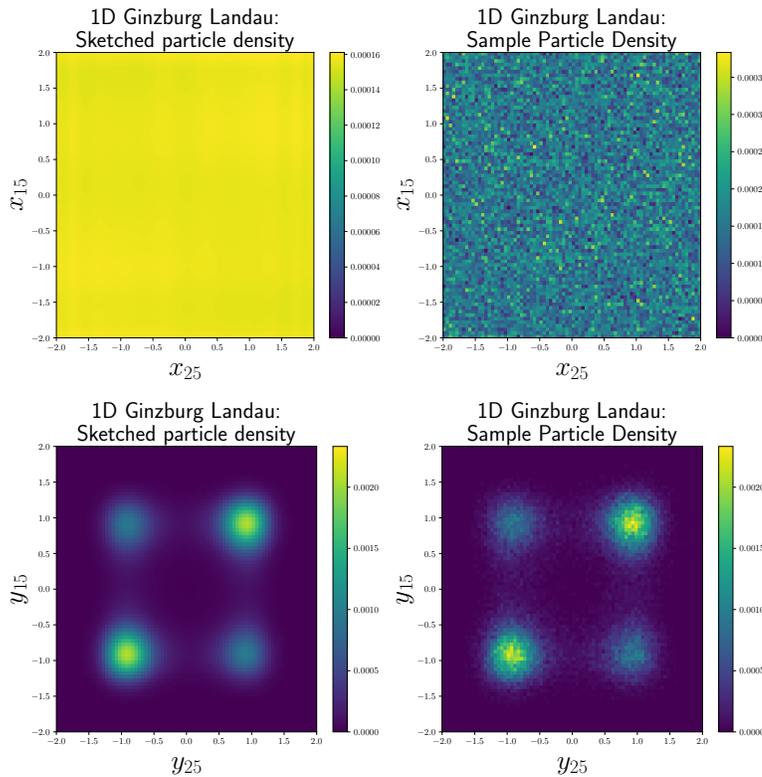


FIG. 6. 1D Ginzburg–Landau model. The plots of the marginal distribution at (x_{15}, x_{25}) and (y_{15}, y_{25}) .

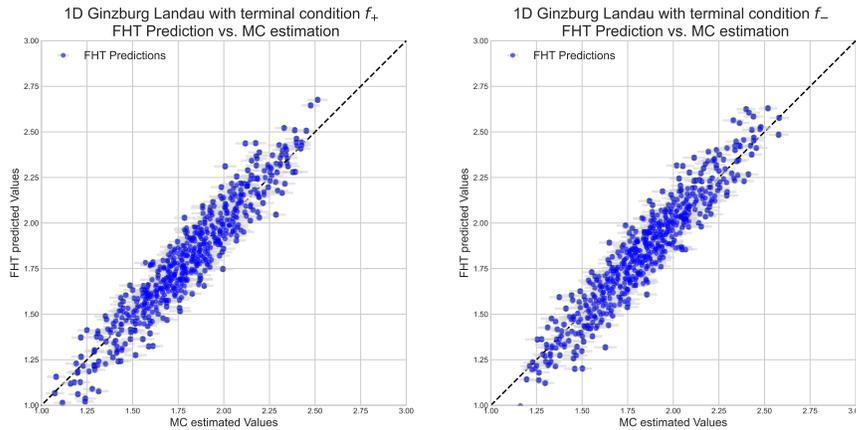


FIG. 7. 1D Ginzburg–Landau model. The plot compares the function evaluation of the FHT-based approximation with reference values obtained from Monte Carlo estimations, where the error bar indicates the standard deviation from each Monte Carlo estimation.

For both terminal conditions, to compare the performance with the true backward equation, we estimate the value of $u(x, 0)$ at $N_u = 600$ randomly sampled points. For each point, we sample 100 SDE trajectories to approximately query the evaluation of u . The comparison with the predicted value through the Markov operator is plotted in Figure 7, which shows that the approximated PDE solution matches the Monte Carlo estimation well. Overall, the average error relative to the true value for both terminal conditions is less than 4%.

For the final numerical test, we consider using the Markov operator to probe the transitional properties of the Ginzburg–Landau model. In the Ginzburg–Landau model, one natural question to ask is whether a particular initial state x favors entering proximity to the state y_+ or y_- under the Langevin dynamics. One way to analyze this is to consider two functions u_+, u_- , which solves the backward equation for the following two terminal conditions:

$$(4.5) \quad g_+(y_1, \dots, y_d) = \exp\left(-\frac{2}{d} \sum_{j=1}^d (y_j - 1)^2\right), \quad g_-(y_1, \dots, y_d) = \exp\left(-\frac{2}{d} \sum_{j=1}^d (y_j + 1)^2\right),$$

and the propensity for a state x to be near y_+ is characterized by the ratio

$$\iota(x) = \frac{u_+(x, 0)}{u_+(x, 0) + u_-(x, 0)}.$$

In this case, both g_+ and g_- are separable and admit a highly accurate FHT-based approximation, which is why one can directly solve for u_+, u_- through FHT contraction with the Markov operator. To consider an interesting case, we plot the predicted value of $\iota(x)$ along the direction of $x(t) = (t, \dots, t) = t\mathbf{1}_d$. As can be seen in Figure 8, $\iota(x(t))$ has a sharp transition at the region of $t \in [-0.15, 0.15]$, which indicates that the state $x(t)$ has a high likelihood of entering a proximity to y_+ when $t > 0$ is away from zero. Quite remarkably, this sharp transition is accurately captured by the FHT solution approximated by the Markov operator.

4.3. 2D Ginzburg–Landau potential. We consider a 2D Ginzburg–Landau model. The potential energy is defined as

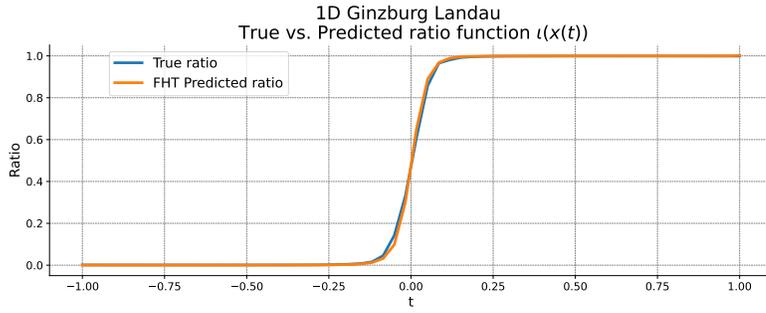


FIG. 8. 1D Ginzburg–Landau model. Plot of $\iota(x(t))$, i.e., the propensity of $x(t) = (t, \dots, t)$ to enter a proximity of $y_+ = (1, \dots, 1)$. One can see that the Markov operator approach can capture the sharp transition at $t = 0$.

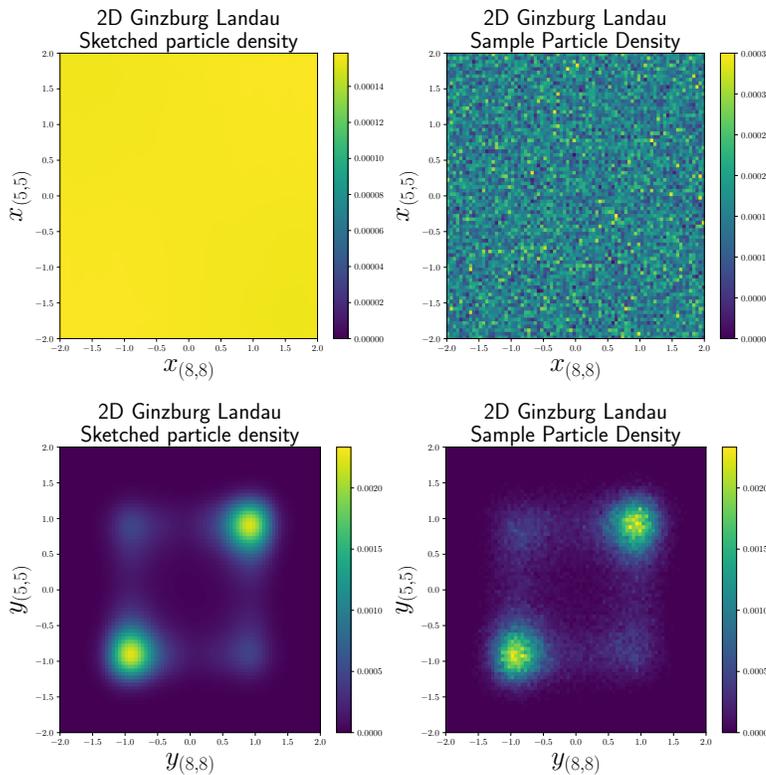


FIG. 9. 2D Ginzburg–Landau model. The plots of the marginal distribution at $(x_{(5,5)}, x_{(8,8)})$ and $(y_{(5,5)}, y_{(8,8)})$.

$$(4.6) \quad V(x_{(1,1)}, \dots, x_{(m,m)}) := \frac{\lambda}{2} \sum_{v \sim w} \left(\frac{x_v - x_w}{h} \right)^2 + \frac{1}{4\lambda} \sum_v (1 - x_v^2)^2,$$

where $h = \frac{1}{m+1}$ and $x_{(l,0)} = x_{(l,m+1)} = x_{(0,l)} = x_{(m+1,l)} = 0$ for $l = 1, \dots, m$. The parameters are $m = 16$, $\lambda = 0.03$, $\beta = \frac{1}{5}$. The dimension d is 256.

We perform $N = 60000$ SDE simulations with $T = 1$ and $\delta t = \frac{1}{1000}$, with the initial distribution $p_0 = \text{Unif}([-2, 2]^d)$. The maximal Legendre degree q is taken to be $q = 10$, and the maximal internal bond dimension is $r = 15$.

In Figure 9, we plot the marginal distribution of $(x_{(5,5)}, x_{(8,8)})$ and $(y_{(5,5)}, y_{(8,8)})$. In particular, the model can capture the correlation of the y variable with a relatively

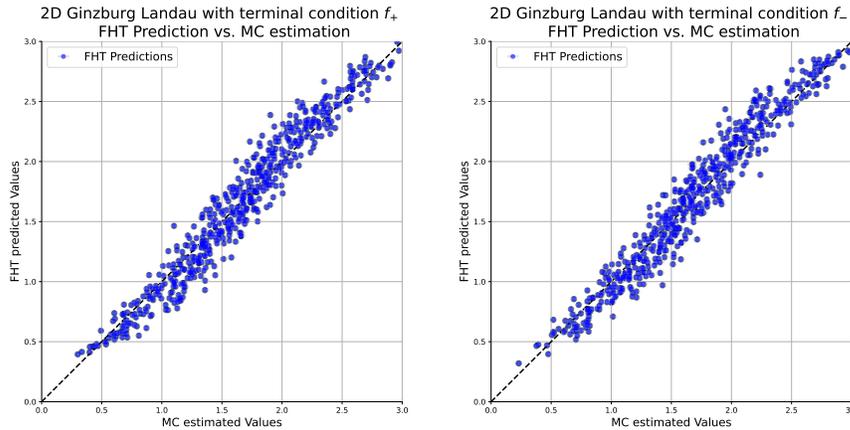


FIG. 10. 2D Ginzburg–Landau model. The plot compares the function evaluation of the FHT-based approximation with reference values obtained from Monte Carlo estimations, where the error bar indicates the standard deviation from each Monte Carlo estimation.

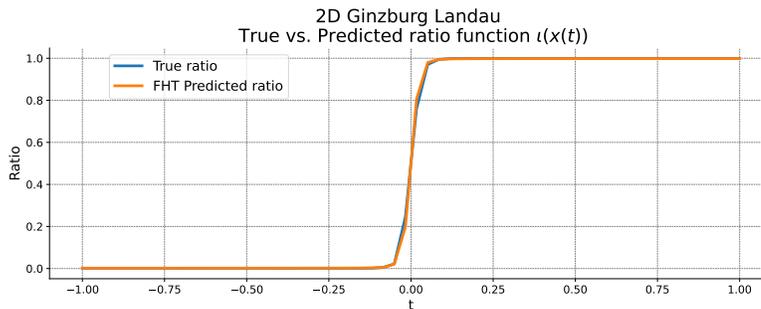


FIG. 11. 2D Ginzburg–Landau model. Plot of $\iota(x(t))$, i.e., the propensity of $x(t) = (t, \dots, t)$ to enter a proximity of $y_+ = (1, \dots, 1)$. One can see that the Markov operator approach can capture the sharp transition at $t = 0$.

small internal bond, which is numerical proof for the hierarchical tensor network ansatz to capture correlation structures when the variables have a 2D lattice structure.

Similar to section 4.2, we consider the KBE solution to terminal conditions f_+ , f_- as defined in (4.4). Overall, Figure 10 shows that the average error relative to the true value for both terminal conditions is less than 8%. The 2D example has a higher error than the 1D case due to the increased number of variables. Lastly, in Figure 11, we repeat the experiment setting in section 4.2 and plot $\iota(x)$ along the direction of $x(t) = (t, \dots, t) = t\mathbf{1}_d$. One can see that the Markov operator can capture the sharp transition at $t \in [-0.15, 0.15]$.

5. Conclusion. We introduce a novel Markov operator approach that combines particle methods with a functional hierarchical tensor network to solve the high-dimensional Kolmogorov backward equations. The algorithm is applied to the discretized Ginzburg–Landau model in 1D and 2D. This method demonstrates that tensor-network-based operators can be used to tackle high-dimensional equations. An open question is whether one can extend a tensor network approach to solving related nonlinear equations, such as the Hamilton–Jacobi equation.

REFERENCES

- [1] M. BACHMAYR, R. SCHNEIDER, AND A. USCHMAJEV, *Tensor networks and hierarchical tensors for the solution of high-dimensional partial differential equations*, *Found. Comput. Math.*, 16 (2016), pp. 1423–1472, <https://doi.org/10.1007/s10208-016-9317-9>.
- [2] J. BALLANI, L. GRASEDYCK, AND M. KLUGE, *Black box approximation of tensors in hierarchical Tucker format*, *Linear Algebra Appl.*, 438 (2013), pp. 639–657, <https://doi.org/10.1016/j.laa.2011.08.010>.
- [3] C. BECK, S. BECKER, P. GROHS, N. JAAFARI, AND A. JENTZEN, *Solving the Kolmogorov PDE by means of deep learning*, *J. Sci. Comput.*, 88 (2021), 73, <https://doi.org/10.1007/s10915-021-01590-0>.
- [4] J. BERNER, M. DABLANDER, AND P. GROHS, *Numerically solving parametric families of high-dimensional Kolmogorov partial differential equations via deep learning*, in *Advances in Neural Information Processing Systems* 33, 2020, pp. 16615–16627.
- [5] A. BHATTACHARYYA, M. HANSELMANN, M. FRITZ, B. SCHIELE, AND C.-N. STRAEHLE, *Conditional Flow Variational Autoencoders for Structured Sequence Prediction*, preprint, arXiv:1908.09008, 2019.
- [6] M. J. BRENNAN AND E. S. SCHWARTZ, *Finite difference methods and jump processes arising in the pricing of contingent claims: A synthesis*, *J. Financ. Quant. Anal.*, 13 (1978), pp. 461–474, <https://doi.org/10.2307/2330152>.
- [7] W. CAO, D. WANG, J. LI, H. ZHOU, L. LI, AND Y. LI, *BRITS: Bidirectional recurrent imputation for time series*, in *Advances in Neural Information Processing Systems* 31, 2018, pp. 6776–6786.
- [8] Z. CHE, S. PURUSHOTHAM, K. CHO, D. SONTAG, AND Y. LIU, *Recurrent neural networks for multivariate time series with missing values*, *Sci. Rep.*, 8 (2018), 6085, <https://doi.org/10.1038/s41598-018-24271-9>.
- [9] Y. CHEN, J. HOSKINS, Y. KHOO, AND M. LINDSEY, *Committer functions via tensor networks*, *J. Comput. Phys.*, 472 (2023), 111646, <https://doi.org/10.1016/j.jcp.2022.111646>.
- [10] W. E, W. REN, AND E. VANDEN-EIJNDEN, *Minimum action method for the study of rare events*, *Comm. Pure Appl. Math.*, 57 (2004), pp. 637–656, <https://doi.org/10.1002/cpa.20005>.
- [11] Y. FAN AND L. YING, *Solving electrical impedance tomography with deep learning*, *J. Comput. Phys.*, 404 (2020), 109119, <https://doi.org/10.1016/j.jcp.2019.109119>.
- [12] V. L. GINZBURG, V. L. GINZBURG, AND L. LANDAU, *On the Theory of Superconductivity*, Springer, 2009.
- [13] W. HACKBUSCH, *Tensor Spaces and Numerical Tensor Calculus*, Springer Ser. Comput. Math. 42, Springer, 2012.
- [14] W. HACKBUSCH AND S. KÜHN, *A new scheme for the tensor representation*, *J. Fourier Anal. Appl.*, 15 (2009), pp. 706–722, <https://doi.org/10.1007/s00041-009-9094-9>.
- [15] G. E. HINTON, *Training products of experts by minimizing contrastive divergence*, *Neural Comput.*, 14 (2002), pp. 1771–1800, <https://doi.org/10.1162/089976602760128018>.
- [16] K.-H. HOFFMANN AND Q. TANG, *Ginzburg-Landau Phase Transition Theory and Superconductivity*, Internat. Ser. Numer. Math. 134, Springer, 2001.
- [17] P. C. HOHENBERG AND A. P. KREKHOV, *An introduction to the Ginzburg–Landau theory of phase transitions and nonequilibrium patterns*, *Phys. Rep.*, 572 (2015), pp. 1–42, <https://doi.org/10.1016/j.physrep.2015.01.001>.
- [18] Y. HUR, J. G. HOSKINS, M. LINDSEY, E. M. SToudenMIRE, AND Y. KHOO, *Generative modeling via tensor train sketching*, *Appl. Comput. Harmon. Anal.*, 67 (2023), 101575, <https://doi.org/10.1016/j.acha.2023.101575>.
- [19] N. KOVACHKI, Z. LI, B. LIU, K. AZIZZADENESHELI, K. BHATTACHARYA, A. STUART, AND A. ANANDKUMAR, *Neural Operator: Learning maps between function spaces with applications to pdes*, *J. Mach. Learn. Res.*, 24 (2023), pp. 1–97.
- [20] P. KUMAR AND S. NARAYANAN, *Solution of Fokker-Planck equation by finite element and finite difference methods for nonlinear systems*, *Sadhana*, 31 (2006), pp. 445–461, <https://doi.org/10.1007/BF02716786>.
- [21] Y. LECUN, S. CHOPRA, R. HADSELL, M. RANZATO, AND F. HUANG, *A tutorial on energy-based learning*, in *Predicting Structured Data*, MIT Press, 2006.
- [22] L. LIN, J. LU, AND L. YING, *Fast construction of hierarchical matrix representation from matrix–vector multiplication*, *J. Comput. Phys.*, 230 (2011), pp. 4071–4087, <https://doi.org/10.1016/j.jcp.2011.02.033>.
- [23] Y. LU AND B. HUANG, *Structured output learning with conditional generative flows*, in *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, PKP Publishing Services, 2020, pp. 5005–5012, <https://doi.org/10.1609/aaai.v34i04.5940>.

- [24] M. MIRZA AND S. OSINDERO, *Conditional Generative Adversarial Nets*, preprint, arXiv:1411.1784, 2014.
- [25] Y. PENG, Y. CHEN, E. M. STOUDENMIRE, AND Y. KHOO, *Generative Modeling via Hierarchical Tensor Sketching*, preprint, arXiv:2304.05305, 2023.
- [26] D. REZENDE AND S. MOHAMED, *Variational inference with normalizing flows*, in International Conference on Machine Learning, PMLR, 2015, pp. 1530–1538.
- [27] L. RICHTER, L. SALLANDT, AND N. NÜSKEN, *Solving high-dimensional parabolic pdes using the tensor train format*, in International Conference on Machine Learning, PMLR, 2021, pp. 8998–9009.
- [28] K. SOHN, H. LEE, AND X. YAN, *Learning structured output representation using deep conditional generative models*, in Advances in Neural Information Processing Systems 28, 2015, pp. 3483–3491.
- [29] Y. SONG, C. DURKAN, I. MURRAY, AND S. ERMON, *Maximum likelihood training of score-based diffusion models*, in Advances in Neural Information Processing Systems 34, 2021, pp. 1415–1428.
- [30] Y. SONG AND S. ERMON, *Generative modeling by estimating gradients of the data distribution*, in Advances in Neural Information Processing Systems 32, 2019, pp. 11918–11930.
- [31] E. G. TABAK AND E. VANDEN-ELJNDEN, *Density estimation by dual ascent of the log-likelihood*, Commun. Math. Sci., 8 (2010), pp. 217–233, <https://doi.org/10.4310/CMS.2010.v8.n1.a11>.
- [32] X. TANG AND L. YING, *Solving high-dimensional Fokker-Planck equation with functional hierarchical tensor*, J. Comput. Phys., 511 (2024), 113110.
- [33] C. WINKLER, D. WORRALL, E. HOOGBOOM, AND M. WELLING, *Learning Likelihoods with Conditional Normalizing Flows*, preprint, arXiv:1912.00042, 2019.
- [34] J. YOON, W. R. ZAME, AND M. VAN DER SCHAAR, *Estimating missing data in temporal data streams using multi-directional recurrent neural networks*, IEEE Trans. Biomed. Eng., 66 (2018), pp. 1477–1490, <https://doi.org/10.1109/TBME.2018.2874712>.
- [35] J. ZHAO, M. DAVISON, AND R. M. CORLESS, *Compact finite difference method for American option pricing*, J. Comput. Appl. Math., 206 (2007), pp. 306–321, <https://doi.org/10.1016/j.cam.2006.07.006>.