Weak Collocation Regression method: fast reveal hidden stochastic dynamics from high-dimensional aggregate data

Liwei Lu\textsuperscript{1}, Zhijun Zeng\textsuperscript{2}, Yan Jiang\textsuperscript{3}, Yi Zhu\textsuperscript{4}*, and Pipi Hu\textsuperscript{5}\textsuperscript{†}

\textsuperscript{1,2,3,4}Yau Mathematical Sciences Center, Tsinghua University, Beijing, 100084, China.
\textsuperscript{4}Yanqi Lake Beijing Institute of Mathematical Sciences and Applications, Beijing, 101408, China.
\textsuperscript{5}Microsoft Research AI4Science, Beijing, 100080, China.

February 2, 2024

Abstract

Revealing hidden dynamics from the stochastic data is a challenging problem as the randomness takes part in the evolution of the data. The problem becomes exceedingly hard if the trajectories of the stochastic data are absent in many scenarios. In this work, we propose the Weak Collocation Regression (WCR) method to learn the dynamics from the stochastic data without the labels of trajectories. This method utilizes the governing equation of the probability distribution function—the Fokker-Planck (FP) equation. Using its weak form and integration by parts, we move all the spatial derivatives of the distribution function to the test functions which can be computed explicitly. Since the data is a sampling of the corresponding distribution function, we can compute the integrations in the weak form, which has no spatial derivatives on the distribution functions, by simply adding the values of the integrands at the data points. We further assume the unknown drift and diffusion terms can be expanded by the base functions in a dictionary with the coefficients to be determined. Cooperating the collocation treatment and linear multi-step methods, we transfer the revealing process to a linear algebraic system. Using the sparse regression, we eventually obtain the unknown coefficients and hence the hidden stochastic dynamics. The numerical experiments show that our method is flexible and fast, which reveals the dynamics within seconds in the multi-dimensional problems and can be extended to high dimensional data. The complex tasks with variable-dependent diffusion and coupled drift can be correctly identified by WCR and the performance is robust, achieving high accuracy in the cases of noisy data. The rigorous error estimate is also included to support our various numerical experiments.

Keywords— weak form, collocation of kernels, Fokker-Planck equation, aggregate data

1 Introduction

Nowadays, a large amount of data has been collected in different realms, and revealing the hidden dynamics buried in the data is an essential topic in the scientific discovery and engineering
applications. On one hand, the studies in the past several centuries have proven the success of the differential equations derived from the so-called first principle in the descriptions of the phenomena of nature, such as the Navier-Stokes equation in hydrodynamics for fluid dynamics [1], Schrödinger equation in quantum mechanics for probability current [2], Black-Scholes in computational finance for option pricing [3]. On the other hand, machine learning, especially deep learning, in recent years has attained tremendous success in computer vision, natural language processing, and many other topics in computer science [4]. Leveraging the structure of differential equations and machine learning in the data analysis shows its prospective performance in many studies [5, 6, 7, 8]. One of the most important focus is pouring new structures onto modeling the hidden dynamics from data. The physical informed neural network (PINN) adds physical constraints to the data by adding the residual of the differential equations to the loss and making learning coefficients of the unknown terms of the governing equations reliable [9]. Brunton et al. proposed a framework named “SINDy” by combining regression and sparse identification to reveal nonlinear dynamical systems [10]. The time-series data always contain a lot of missing points and even flaws with high noisy level, making analysis hard and tricky. Hu et al. proposed using symbolic ODE (ordinary differential equations) to reveal hidden dynamics from time series data [8] with the integral form making the learning process of the data with large time step more stable. More related works about inferring differential equations, see [11, 12, 13, 14, 15, 16, 17]. Most of these works leverage the structure of differential equations and machine learning schemes to model the data in the deterministic system. However, stochastic dynamics plays a vital role in the applications such as the Brownian motion in the molecular modeling and the financial predictions [18]. In contrast to the “noise” of data generated from the measurement, the randomness in the stochastic dynamics contributes the evolution of the process, making it more complicated for modeling. Fortunately, with subtly designed structure, it is still possible to reveal the hidden dynamics from the stochastic data.

In the realm of stochastic dynamics, aggregate data refers to a data format in which the full trajectory of each individual modeled by the evolution of state is not available, but rather a sample from the distribution of state at a certain time point is available [19]. For example, the data collected for single-cell DNA sequence analysis, bird migration, and social gathering are aggregate data as the individual trajectories for a long time are not possible to follow with only the collection of the states of different individuals obtained. In contrast, trajectory data includes all the information of the individual data along the time. Thus, in some literature, trajectory data and aggregate data are also called “paired” and “unpaired” data respectively [20].

For trajectory data, there exist many methods developed such as Hamiltonian neural networks [21], Hidden Markov Model (HMM) [22], Kalman Filter (KF) [23], Particle Filter (PF) [24] and related works [19, 25]. Revealing determined hidden dynamics from data above can also be regarded as one special case of trajectory data when stochastic term vanishes. However, for aggregate data, few works are investigated because of the absence of individual trajectories. By leveraging the Fokker-Planck equation, the governing function of the probability density function of the Winner process variable, researchers investigate the revealing hidden stochastic dynamics from aggregate data. Zhou et al. propose a novel method using the weak form of the Fokker-Planck Equation (FPE), a partial differential equation, to describe the density evolution of data in a sampled form, which is then combined with the Wasserstein generative adversarial network (WGAN) in the training process [19]. Chen et al. proposed a method that leverages the physical-informed structure of Fokker-Planck equations and the approximation of the probability density function and reveals the hidden dynamics from sampling data at several time points [26]. Yang et al. [20] integrated the distance measure such as Wasserstein distance used in WGAN [27] and the forward numerical solution of the parameterized stochastic differential equations as the Physics-Informed Deep Generative Models to reveal the hidden dynamics of the data.
In this work, by leveraging the weak form of the Fokker-Planck equation, we proposed a Weak Collocation Regression (WCR) method to reveal the hidden stochastic dynamics from nonequally-spaced temporal aggregate data. The Fokker-Planck equation describes the time evolution of the probability density function of the random variable in the Brownian motion. By the weak form of the Fokker-Planck equation, one can reduce the aggregate data at different time points to one dimensional temporal sequence where we have used the Monte-Carlo summation for the approximation of the weak form using data. Temporal derivatives are reduced by Linear Multistep Methods (LMMs), and then the linear system constructed by collocation of the kernels gives a precise approximation of the stochastic dynamics. The benefits are three folds.

1. Remarkable performance. Our method has a low computational cost, comparable accuracy, controlled error, and the dimensional curse can be lessened. Numerical experiments show that the 1-dimensional problem can be easily revealed within 0.02 second on the MacBook, while the computational time of the 3 or 4-dimensional problem can be limited to seconds. We can significantly reduce the error by changing the time interval and sample number. Directly numerically solving the inverse problem of stochastic differential equations may encounter the curse of dimensionality [28], where the tendency of numerical techniques requires a high computational cost growing exponentially with respect to the dimension of the variables. Benefits from the Monte-Carlo summation and the random collocations methods, the curse of dimensionality is lessened. Further, our method has natural potential for parallel computation.

2. Robustness in the data with different qualities. Our method can handle a small amount of non-equally noise data without trajectory information and even partially data missing. No trajectories are needed; the only thing needed is the summation over all the points at each given time. Our method hence has natural permutation invariance with respect to the observations of different individuals at the same time snapshot. Our method can be applied to different number of individuals at different time snapshots. The obstacle of the measurement error can be reduced greatly since the summation of the aggregate data is resistant to white noise. Numerical experiments illustrate the stability of our methods.

3. Complex tasks. Our method is suited for the coupled drift term derived from Sombrero potential, variable-dependent diffusion term, and high-contrast problem and can be extended to a wider range. Numerical experiments show that it obtained consistent results if we expand each drift and diffusion term in a high-order polynomial. It shows that in dealing with the complex task above, our method has a wider representation and shows superior performance in revealing the hidden dynamics with a mild amount of data in a high accuracy.

2 Methodology

In this paper, we consider the scenarios of the so-called aggregate data (unpaired data), where the trajectory information of each individual modeled by the evolution of state is unavailable. Only the collection of the \( N_{t_j} \) samples \( X_{t_j} = \{ x_{t_j}^i \}_{i=1}^{N_{t_j}} \) from the distribution of state \( X_t \) at certain time point \( t_j, j = 1, 2, \cdots, L \) is available. In this work, we call \( X_{t_j} \) one snapshot and hence there are \( L \) snapshots in the data set with respect to \( L \) time points.

For simplicity, we denote the set of the aggregate data as

\[
X \triangleq \{ X_{t_j} \}_{j=1}^{L} = \{ \{ x_{t_j}^i \}_{i=1}^{N_{t_j}} \}_{j=1}^{L},
\] (1)
where \( t_j \) is the time index of the \( j \)-th snapshot in the total \( L \) snapshots, \( x_{ij} \) is the \( i \)-th sample of the variable \( X_t \) at time \( t_j \). To model the stochastic dynamics of the data, the general stochastic differential equations (SDEs) for Brownian motion are considered

\[
dX_t = \mu(X_t, t)dt + \sigma(X_t, t)dW_t,
\]
with drift term \( \mu(X_t, t) \in \mathbb{R}^d \) and diffusion term \( \sigma(X_t, t) \in \mathbb{R}^{d \times w} \). Here, \( X_t \) is the \( d \)-dimensional random variable of the data, and \( W_t \) is the \( w \)-dimensional standard Brownian motion.

The density function \( p(x, t) \) of the above random variable \( X_t \) can be described by the Fokker-Planck equation (FP), see [29], and we restate the result in Lemma 1.

**Lemma 1.** Suppose \( \{X_t\} \) solves the SDEs (2), then the probability density function \( p(x, t) \) of the random variable \( X_t \) satisfies the following \( d \)-dimensional Fokker-Planck equation by the Itô integral

\[
\partial_t p = -\nabla \cdot (\mu p) + \sum_{i,j} \partial_{ij}(D_{ij}p),
\]

where \( x \in \mathbb{R}^d, t \in [0, T] \subset \mathbb{R}, p = p(x, t) \in \mathbb{R} \) is the probability density function with \( \int_{\mathbb{R}^d} p(x, t)dx = 1 \), \( \mu = [\mu_1(x, t), \mu_2(x, t), \cdots, \mu_d(x, t)]^T \), and the diffusion matrix \( [D_{ij}] = [D_{ij}(x, t)] \) is given by

\[
D = \frac{1}{2} \sigma \sigma^T.
\]

The Fokker-Planck equation bridges the gap between stochastic dynamics and the distributions of the data samples by sharing drift term and diffusion relation in SDE (2) and FP equation (3). To reveal the hidden stochastic dynamics (2), equivalently one can reveal the unknown terms in FP equation (3) by the samples which follow the corresponding probability distribution.

However, there is a huge gap between the density function and the data samples. Directly modeling the data by the Fokker-Planck equation usually needs the temporal and spatial derivatives of the density function \( p(t, x) \) with respect to \( t \) and \( x \), requiring a large amount of samples for constructing a smooth density function. It becomes even impossible to construct the smooth density function when the dimension increases. To overcome this difficulty, in this work, we propose a framework using the weak form with collocation integral kernels instead of reconstructing density function to fast reveal hidden stochastic dynamics with a mild amount of data. We briefly introduce the methods here. We first write the weak form of the FP equation and then integrated by parts, the partial derivatives in the weak form are moved to the explicit kernel function, making the computation of the spatial derivatives much easier. Followed by the Monte-Carlo summation, the integration over space can be given by the summation over the samples. And the terms with temporal derivatives can be approximated by the linear multi-step method (LMMs). Finally, with a basis expansion of the unknown drift and diffusion, the linear system is built, and the sparse regression gives a good approximation of the unknown terms. Figure 1 depicts the picture of the methodology. The weak form, LMMs, and the regression model would be detailedly discussed in the next several subsections.

### 2.1 Leverage the weak form of the Fokker-Planck equation

In the realm of PDE theories, weak solutions from the weak form of PDEs are vital in analysis and applications. The weak form is introduced to solving PDEs with neural networks by the so-called WAN method [30] and then extended to revealing the unknown parameters in PDEs [31]. Borrowing the ideas from the Galerkin methods, we alter the weak form of the FP equations by using collocations
of the kernel function as test functions. The space of the test functions thus are approximated by
colloctions of the kernels unlike the maximum optimization for the tunable parameters in WAN.
Without maximum steps for the test function, the optimization only for the unknown terms in the
revealing stochastic dynamics achieves stable performance with high robustness.

The test function is given as the kernel function \( \psi(x) : \mathbb{R}^d \rightarrow \mathbb{R} \). For example, a typical choice
be the Gaussian function with the form

\[
\phi(x; \rho, \gamma) = \prod_{i=1}^d \frac{1}{\gamma_i \sqrt{2\pi}} e^{-\frac{1}{2} \left( \frac{x_i - \rho_i}{\gamma_i} \right)^2}, \tag{5}
\]

where \( x = (x_1, \ldots, x_d)^T \in \mathbb{R}^d, \rho = (\rho_1, \ldots, \rho_d)^T \) and \( \gamma = (\gamma_1, \ldots, \gamma_d)^T \) are the expectation and
standard deviation of the Gaussian function.

Multiply \( \psi(x) \) on both sides of equation (3), integrate by parts, and one obtains

\[
\frac{d}{dt} \int_{\mathbb{R}^d} p(x, t) \psi(x) dx = \int_{\mathbb{R}^d} p(x, t) \mu(x, t) \cdot \nabla \psi(x) dx + \int_{\mathbb{R}^d} p(x, t) \sum_{rs} D_{rs} \partial_{rs} \psi(x) dx, \tag{6}
\]

where \( \partial_{rs} = \frac{\partial^2}{\partial x_r \partial x_s} \) and \( r, s \) are indices of the dimension.

As \( p(x, t) \) is the density of the variable \( X_t \) at time \( t \) with \( \int p(x, t) dx = 1 \), we can rewrite equation (6) as the expectation form

\[
\frac{d}{dt} \mathbb{E}_{x \sim p(x, t)}[\psi(x)] = \mathbb{E}_{x \sim p(x, t)}[\mu(x, t) \cdot \nabla \psi(x)] + \mathbb{E}_{x \sim p(x, t)}[\sum_{rs} D_{rs} \partial_{rs} \psi(x)], \tag{7}
\]

where \( \mathbb{E}_{x \sim p(x, t)} \) is the expectation over the probability density function \( p(x, t) \).

### 2.2 Approximate the weak form using data

In the scenarios of real applications, the probability distribution \( p(x, t) \) is hardly to obtain but the
data of many samplings from the distribution is always feasible thanks to the modern technology and
instruments. By the law of large numbers, the expectation over the distribution can be approximated
by the summation over the samplings of the variable related to the distribution, i.e.,

\[
\frac{1}{N_t} \sum_{i=1}^{N_t} \psi(x^i_t) \sim \mathbb{E}_{x \sim p(x, t)}[\psi(x)], \tag{8}
\]

where \( N_t \) is the number of the samplings of the variable \( X_t \) at time \( t \), and \( x^i_t \) is the \( i \)-th sampling.

For simplicity, we denote the data set over time as \( X_t \triangleq \{x^i_t\}_{i=1}^{N_t} \). Thus, with the data set of the
samplings \( X_t \), the weak form of the FP equation (5) yields

\[
\frac{d}{dt} \left( \frac{1}{N_t} \sum_{i=1}^{N_t} \psi(x^i_t) \right) = \frac{1}{N_t} \sum_{i=1}^{N_t} \mu(x^i_t, t) \cdot \nabla \psi(x^i_t) + \frac{1}{N_t} \sum_{i=1}^{N_t} \sum_{rs} D_{rs} \partial_{rs} \psi(x^i_t) + \epsilon, \tag{9}
\]

where we have used the approximations of the expectations by the summations over the samplings,
and the discussion of the error term \( \epsilon \) can be found in Appendix A. The information of the drift and
diffusion terms are hence related with the data set \( X_t \) in equation (9). The dimension of the data
has been reduced to one by the summation over the samplings with only the dependence on the time \( t \) left. This form reveals that our method is naturally suited for the high dimensional problem.
For simplicity, we denote
\[ \frac{d}{dt}y(X_t) = f(X_t, \mu, D), \quad (10) \]
where
\[ y(X_t) = \frac{1}{N_t} \sum_{i=1}^{N_t} \psi(x_i) \]
and
\[ f(X_t, \mu, D) = \frac{1}{N_t} \sum_{i=1}^{N_t} \mu(x_i, t) \cdot \nabla \psi(x_i) + \frac{1}{N_t} \sum_{i=1}^{N_t} \left( \sum_{rs} D_{sr} \partial_{rs} \psi(x_i) \right) \]
are scalar functions varying over time \( t \).

2.3 Approximate the temporal derivatives

The equation (10) is indeed a 1-dimensional parameterized ordinary differential equation where the solutions on the discrete times were given with \( y(X) \). The problem of finding unknown terms in FP equation now reduces to the inverse problem of the 1-dimensional ordinary differential equation (10). Many methods recently have been developed to reveal the dynamics such as SINDy [32], PINN [9] and ODENet [8]. Considering the simplicity, in this work, we directly use the implicit form of Linear Multistep Methods (LMMs) to construct an explicit discrete form of equation (10).

Linear Multistep Methods (LMMs) have been developed as popular numerical schemes for the integration of the ordinary differential equations for the known dynamic systems [33] with well-established mathematical theories [34]. Raissi et al. constructed multistep neural networks for data-driven discovery of nonlinear dynamical systems leveraging LMMs [35]. Recently Du et al. applied LMMs in learning hidden dynamics from the data of given state with theoretical analysis of the stabilities and the convergence for the inverse problem [36, 37]. In their work, equally-spaced version of LMMs is used. However, real data in the record usually contain a lot of missing points and even flaws, making equally-spaced version of LMMs hard to apply. In this work, with the help of variable step size Adams methods [38], we can deal with the non-equally spaced temporal data for the discovery of the hidden dynamics.

For equally-spaced temporal data, Implicit Adams methods of trapezoidal rule, Milne method, 2-step backward differentiation method (BDF2) and Adams-Moulton methods are used in this work and some of them are listed below

\[ \text{Trapezoidal rule: } y_{n+1} - y_n = \frac{h}{2} \left( f_{n+1} + f_n \right), \quad (11) \]
\[ \text{Milne method: } y_{n+1} - y_{n-1} = \frac{h}{3} \left( f_{n+1} + 4f_n + f_{n-1} \right), \quad (12) \]
where \( y_n = y(t_n) \), \( f_n = f(t_n, \theta) \) and \( h \equiv h_n = t_{n+1} - t_n \) for \( n = 1, 2, \ldots, L - 1 \).

For more general temporal data without equally spaced time, the 2-step BDF-formula of variable step size methods is as follows
\[ y_{n+1} = \frac{(1 + \omega_n)^2}{1 + 2\omega_n} y_n + \frac{\omega_n^2}{1 + 2\omega_n} y_{n-1} = h_n \frac{1 + \omega_n}{1 + 2\omega_n} f_{n+1}, \quad (13) \]
where \( \omega_n = h_n/h_{n-1} \).
Recall that the Implicit Adams methods of trapezoidal rule requires only two adjacent time points, thus the variable step-size version reads

$$y_{n+1} - y_n = \frac{h_n}{2} (f_{n+1} + f_n).$$  \hfill (14)

Our numerical experiments show that the trapezoidal rule performs well when only three time snapshots are available. In the contrast, the LMMs methods requiring 2 steps fails in such scenario even if they have higher order of accuracy. Other types of the numerical schemes have been also tested but with a worse performance compared with the LMMs.

By LMMs, we derive the discrete form of equation (10) as the following form

$$\hat{y}(X) = \hat{f}(X, \mu, D),$$ \hfill (15)

where $$\hat{y}(X)$$ is a vector constructed by $$y(X) = \{y(X_{t_j})\}_{j=1}^L$$, and $$\hat{f}(X, \mu, D)$$ is a vector with the same size given by $$f(X_{t_j}, \mu, D) = \{f(X_{t_j}, \mu, D)\}_{j=1}^L$$. Now the equation (15) gives us an algebraic equation, from which, by an apt ansatz form for the drift and diffusion terms, we can reveal $$\mu$$ and $$D$$ with the data set $$X$$.

2.4 Build the regression model

In this subsection, we investigate the sparse regression and collocations of the kernels to solve $$\mu$$ and $$D$$ from (15) with the data set $$X$$. The dictionary representations are adopted for the unknown terms. One typical choice can be the polynomial basis for the approximation of the drift and diffusion terms with the coefficients of the basis to be determined. Let

$$\Lambda = \{1, x_1, x_2, \ldots, x_d, x_1x_1, x_1x_2, \ldots, x_d^p\}^T$$ \hfill (16)

denote the $$p$$-th order complete polynomials with respect to the variable $$x = (x_1, \ldots, x_d)^T$$. The number of the terms of $$\Lambda$$ is $$b \triangleq |\Lambda| = \binom{p + d}{p}$$. In real applications, if we have some knowledge about the stochastic process, more flexible basis set $$\Lambda$$ can be chosen with diverse and fewer candidate basis.

Here, for simplicity, we suppose that $$\mu$$ and $$D$$ are independent of $$t$$. And the components $$\mu$$ of $$\mu = [\mu_i]$$ and $$D_{ij}$$ of $$D = [D_{ij}]$$ expand as

$$\mu_i = \sum_{j=1}^b c^\mu_{ij} \Lambda_j,$$

$$D_{ij} = \sum_{k=1}^b c^D_{ijk} \Lambda_k,$$ \hfill (17)

where $$\Lambda_j$$ is the $$j$$-th component in the basis set $$\Lambda$$.

Now, equation (10) has the following matrix form

$$\frac{d}{dt} y(X) = B(X) \zeta$$ \hfill (18)

where

$$\zeta = \{c^\mu_{11}, \ldots, c^\mu_{1b}, \ldots, c^\mu_{b1}, \ldots, c^\mu_{bb}, c^D_{111}, \ldots, c^D_{11b}, \ldots, c^D_{b11}, \ldots, c^D_{bbb}\}^T,$$ \hfill (19)

flatten of $$\zeta^\mu$$

flatten of $$\zeta^D$$. 

7
and the flatten scheme of $\zeta^u$ and $\zeta^D$ is shown as

$$
\zeta^u : \begin{pmatrix}
\zeta_{11}^u & \cdots & \zeta_{12}^u & \cdots & \zeta_{16}^u \\
\zeta_{21}^u & \cdots & \zeta_{22}^u & \cdots & \zeta_{26}^u \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
\zeta_{d1}^u & \cdots & \zeta_{d2}^u & \cdots & \zeta_{db}^u
\end{pmatrix},
\zeta^D : \begin{pmatrix}
\zeta_{11}^D & \cdots & \zeta_{12}^D & \cdots & \zeta_{16}^D \\
\zeta_{d1}^D & \cdots & \zeta_{dd}^D & \cdots & \zeta_{dd}^D
\end{pmatrix}
$$

and $B(\mathcal{X}) := \begin{pmatrix} b_1^T(\mathcal{X}) \\ \vdots \\ b_L^T(\mathcal{X}) \end{pmatrix}$ is a known coefficient matrix with size $L \times (db + d^2b)$, $y(\mathcal{X}) := \begin{pmatrix} y_1(\mathcal{X}) \\ \vdots \\ y_L(\mathcal{X}) \end{pmatrix}$ is a column vector with size $L$.

By applying linear multistep methods (LMMs) on the temporal derivative in (18), one can obtain a linear system about the coefficient vector $\zeta$ as

$$
A(\mathcal{X}) \zeta = \dot{y}(\mathcal{X}),
$$

where $A(\mathcal{X})$ and $\dot{y}(\mathcal{X})$ are constructed by $B(\mathcal{X})$ and $y(\mathcal{X})$ in (18) because of the linearity of the system, and the coefficient vector $\zeta$ are collected and vectorized from all the coefficients $\{ \zeta_{ijk}^b \}_{i=1}^{d}$ and $\{ \zeta_{ijk}^D \}_{i,j=1}^{d}$ in the expansions of $\mu$ and $D$ shown in (19). Now, equation (20) gives the relation between the data set $\mathcal{X}$ and the unknown parameters in the polynomial expansions of the hidden dynamics.

**Example 1.** To make it clear, we give an example to illustrate how the linear system is obtained by applying LMMs such as the trapezoidal rule (11) on the equation (18) with temporal derivatives.

In equation (18), we set $n = db + d^2b$ and

$$
B(\mathcal{X}) = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{L1} & b_{L2} & \cdots & b_{Ln} \end{pmatrix},
\quad
y(\mathcal{X}) = \begin{pmatrix} y_1(\mathcal{X}) \\ y_2(\mathcal{X}) \\ \vdots \\ y_L(\mathcal{X}) \end{pmatrix}
$$

By the trapezoidal rule, the matrix $A(\mathcal{X}) \in \mathbb{R}^{(L-1) \times (db + d^2b)}$ and vector $\dot{y}(\mathcal{X}) \in \mathbb{R}^{L-1}$ in the linear system can be assembled by

$$
A(\mathcal{X}) = h \frac{2}{T} \begin{pmatrix} b_{11} + b_{21} & b_{12} + b_{22} & \cdots & b_{1n} + b_{2n} \\ b_{21} + b_{31} & b_{22} + b_{32} & \cdots & b_{2n} + b_{3n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{L-1,1} + b_{L,1} & b_{L-1,2} + b_{L,2} & \cdots & b_{L-1,n} + b_{Ln} \end{pmatrix},
\quad
\dot{y}(\mathcal{X}) = \begin{pmatrix} y_1(\mathcal{X}) - y_1(\mathcal{X}) \\ y_2(\mathcal{X}) - y_2(\mathcal{X}) \\ \vdots \\ y_L(\mathcal{X}) - y_{L-1}(\mathcal{X}) \end{pmatrix}.
$$

Here, the rows of the matrix $A(\mathcal{X})$ or the elements of the vector $\dot{y}(\mathcal{X})$ are constructed by the rows of $B(\mathcal{X})$ or the elements of $y(\mathcal{X})$ where the number of rows is reduced by one because of the trapezoidal rule (11).
A sparse regression applied to the linear system then reveals the hidden dynamics from the data. But before that, we would discuss the collocation strategies to improve the robustness and accuracy. Recall that in this work, the test function \( \psi \) is considered as the Gaussian function

\[
\psi(x, \rho, \gamma) = \prod_{i=1}^{d} \frac{1}{\gamma_i \sqrt{2\pi}} e^{-\frac{1}{2} \left( \frac{x_i - \rho_i}{\gamma_i} \right)^2} \]

given by \([5]\), one can easily obtain the the specific form of the linear system \([20]\) as

\[
A(X, \rho, \gamma) \zeta = \hat{y}(X, \rho, \gamma) \tag{23}
\]

by the replacement of \( \psi(x) = \phi(x, \rho, \gamma) \).

The collections of the test functions is taken as \( \mathcal{C}_{\rho, \gamma}^d = \{ \phi(\cdot, \rho_m, \gamma_m) \}_{m=1}^M \) where \( \rho_m \in \mathbb{R}^d \) and \( \gamma_m \in \mathbb{R}^d \). Here we have used the notations \( \rho_m \) and \( \gamma_m \) to denote the expectation and standard deviation of the \( m \)-th Gaussian function. Taking the \( m \)-th Gaussian function \( \phi(\cdot, \rho_m, \gamma_m) \) as the test function, the linear system with respect to \( \zeta \) yields

\[
A_m \zeta = \hat{y}_m, \tag{24}
\]

where \( A_m = A(X, \rho_m, \gamma_m) \) and \( \hat{y}_m = \hat{y}(X, \rho_m, \gamma_m) \). The values of the unknown coefficients \( \zeta \) can be revealed solving this linear system. However, integration with one test function \( \phi(x, \rho, \gamma) \) only reveals part of the information from the given data. Hence, we build the linear systems over the whole test function collection \( \mathcal{C}_{\rho, \gamma}^d \). Namely, we assemble and solve the stacked linear equation

\[
\tilde{A} \zeta = \tilde{y}, \tag{25}
\]

where

\[
\tilde{A} = \begin{pmatrix}
A_1 \\
A_2 \\
\vdots \\
A_M
\end{pmatrix}, \quad \text{and} \quad \tilde{y} = \begin{pmatrix}
\hat{y}_1 \\
\hat{y}_2 \\
\vdots \\
\hat{y}_M
\end{pmatrix} \tag{26}
\]

are constructed by the matrix \( \{ A_m \}_{m=1}^M \) and the vector \( \{ \hat{y}_m \}_{m=1}^M \) over all of the test functions in the collection \( \mathcal{C}_{\rho, \gamma}^d \).

To better approximate the complete test functional space, the collection \( \mathcal{C}_{\rho, \gamma}^d \) needs to be large enough, which means a huge amount of computational cost. To overcome this difficulty, we borrow the ideas of the functional basis from the Galerkin methods and the collocation points from the collocation methods. It is key to generate the collection of the test functions \( \mathcal{C}_{\rho, \gamma}^d \) taking into account the efficiency and the accuracy. For simplicity, we randomly generate \( M \) parameters of \( \rho_m \) in the range of the values of the data \( X \). Thus the test function collection \( \mathcal{C}_{\rho, \gamma}^d \) is obtained and a large linear system \([25]\) is then built.

Followed by the linear regression and sparse identification algorithm (STRidge) \([10]\), the explicit form of the solved dynamics fits the data, see the STRidge Algorithm \([2]\) in appendix for details. Specifically, to give the sparse results, the linear regression is applied but with the hard threshold \( \eta \) taken, i.e., smaller components of \( \zeta \) less than \( \eta \) would be set zero and then without those rows and columns, linear regression continues until converges. Distributing the components of \( \zeta \) to the drift and diffusion terms gives the explicit form of the hidden stochastic dynamics.

The result of the WCR method error analysis is summarized in Theorem \([1]\).

**Theorem 1.** Let \( \hat{\zeta} = A^t b \) be the learned model coefficients and \( \zeta^* \) the true model coefficients, \( \alpha \) is the order of linear multi-step method. For \( C \) independent of sample number \( N \) and time interval \( \Delta t \), the following holds

\[
\mathbb{E} \left[ \left\| \hat{\zeta} - \zeta^* \right\|_\infty \right] \leq C \left( \frac{1}{\sqrt{N} \Delta t} + \Delta t^\alpha \right).
\]

9
The process of the error analysis is based mainly on the approach in [39]. Following this method, the error of WCR is divided into two parts, each of which is separately estimated. And the complete proof is included in Appendix A.

Furthermore, for fixed number of basis \( b \), the computational complexity of WCR is \( O(LNMd) \), with the number of snapshots \( L \), samples \( N \), test functions \( M \) and dimension \( d \). See Appendix B for the detail.

The pseudo code of the algorithm is exhibited in Algorithm 1. Figure 1 shows the whole procedure of the methodology.

**Remark 1.** Another choice of the ansatz for the drift and diffusion terms can be the neural network leveraging its so-called universal approximation properties. But it would require nonlinear optimization, which enhance the complexity of the algorithm, and left for further investigations beyond the scope of this work. We focus on introducing the framework of the weak form to reveal the hidden dynamics in this work.

**Remark 2.** We drew inspiration from the Kernel Density Estimation method and convolutional neural networks when selecting the Gaussian function as our test function. The Gaussian function offers a simple form and can be easily adjusted using only two parameters: the mean and standard deviation. This flexibility allows it to adapt to different data distributions effectively. Moreover, the Gaussian function demonstrates excellent smoothness and performs well in probability density estimation tasks. Its concentrated values and rapid decay make it suitable for the proofs presented in the appendix of our work. However, we would like to emphasize that our method is not limited to Gaussian functions alone and can be applied with other types of test functions as well.
Figure 1: The diagram of the weak collocation regression method. The aggregate data set $X$ on panel (b) is the collection of $L$ snapshots of samples at time $t_1, t_2, \ldots, t_L$ from one unknown stochastic process. We model this process by the stochastic differential equations in panel (a) with unknown drift $\mu(X_t, t)$ and diffusion $\sigma(X_t, t)$ terms. By sampling Gaussian kernels in panel (c), for each kernel, the weak form in panel (d) gives the algebraic relation of the unknown terms and the data set. By the LMMs and the basis expansion of the unknown terms, a linear system is built and further combined together to form a large system over all of the collocation kernels. Finally, the sparse linear regression gives the sparse regression of the drift and diffusion terms and hence the hidden dynamics is revealed.
Algorithm 1: Weak Collocation Regression method (WCR).

**Result:** The explicit form of the governing stochastic equation of the data.

**Input:** Aggregate data set $X$.

1. Sample $M$ expectations $\{\rho_m\}_{m=1}^M$ in the region which contains all of the samples in the data set $X$;
2. Set the expectations $\{\gamma_m\}_{m=1}^M$ as hyper-parameters with default 1 in each dimension;
3. Generate the Gaussian function collections $C_{\rho,\gamma}^d = \{\phi(\cdot, \rho_m, \gamma_m)\}_{m=1}^M$ using above expectations and deviations;
4. for $m <= M$ do
   5. Compute and assemble the vector $\mathbf{y}(X) = \begin{pmatrix} \cdots \frac{1}{N_t} \sum_{i=1}^{N_t} \phi(x_{l,i}, \rho_m, \gamma_m) \cdots \end{pmatrix}$ by the left term of the weak form (9) where the test function is $\psi = \phi(x_{l,i}, \rho_m, \gamma_m)$ and $l$ is the index of the $l$-th time snapshots;
6. Compute and assemble the matrix $B(X)$ in (18) by the right term of the weak form (9) over all of the time snapshots with the help of the basis expansions of each entry of drift vector $\mu = [\mu_i]$ and diffusion matrix $D = [D_{ij}]$,
   $$\mu_i = \sum_{j=1}^{b} \zeta_{ij}^\mu A_j; \quad D_{ij} = \sum_{k=1}^{b} \zeta_{ijk}^D A_k;$$
7. Assemble the vector $\zeta$ by $\zeta^\mu = [\zeta_{ij}^\mu]$ and $\zeta^D = [\zeta_{ijk}^D]$ using the flatten scheme in (19);
8. Compute the matrix $A_m$ and $\hat{\mathbf{y}}_m$ by the matrix $B(X)$ and $\mathbf{y}(X)$ using the LMMs, e.g., the trapezoidal rule (11);
9. end
10. Stack the linear system $\tilde{A}\zeta = \tilde{\mathbf{y}}$ with $\tilde{A} = (A_1^T, \ldots, A_M^T)^T$ and $\tilde{\mathbf{y}} = (\mathbf{y}_1^T, \ldots, \mathbf{y}_M^T)^T$;
11. Compute $\zeta$ by a sparse linear regression of $\tilde{A}\zeta = \tilde{\mathbf{y}}$;
12. Reconstruct the drift $\mu$ and diffusion $D$ by distributing $\zeta$.

3 Numerical experiments

Data acquisition. All the raw data used in this work for the experiments of revealing the hidden dynamics are obtained by integrating the given SDEs
$$dX_t = \mu_t dt + \sigma_t dW_t, \quad t \geq 0$$
from $t = 0$ to $t = T$ with Euler-Maruyama scheme
$$\tilde{X}_{(i+1)\delta t} = \tilde{X}_{i\delta t} + \mu_i \delta t + \sigma_i \sqrt{\delta t} \tilde{N}_i.$$ (27)
Here $\delta t$ is the time step of the numerical scheme, $\tilde{N}_i$ are i.i.d standard Gaussian random variables and the intial values are sampled from a given distribution such as a Gaussian function. The same random seed is adopted among different experiments. The experimental data is then sampled from these trajectories, i.e., $N_t$ points are sampled at each time snapshot to remove the trajectory information, and only $L$ time snapshots are chosen as the experimental data, denoted as $X =$
\{X_t\}_{i=1}^L = \{\{x_i^j\}_{j=1}^{N_i}\}_{i=1}^L$. The time snapshots can be non-equally spaced where the time interval \(\Delta t_i = t_{i+1} - t_i\) varies and if the time interval is equal we denote the interval as \(\Delta t\). The random noise is added to the raw data with the noise level \(\delta\) as \(\hat{x}_i^j = x_i^j + \delta U_i^j x_i^j\) where \(U_i^j\) is a random variable.

**Experimental setups.** In this work, we use the collocations of the Gaussian functions as the test functions for the weak form. We sample these Gaussian kernels by randomly sampling the expectations \(\rho_m\), \(m = 1, 2, \ldots, M\) using Latin Hypercub Sampling (LHS) method \cite{40} in the region of the data, i.e., the hypercube containing all of the data. And the standard deviation \(\gamma = \gamma I_d\), where \(I_d\) is the identity matrix, is chosen as a hyper parameter with default \(\gamma = 1\) in each sampled Gaussian function. In all cases, our experiments show that all of the non-zero coefficients are correctly identified and all of the zero terms are eliminated by the sparse regression. Hence in this work, we define the Maximum Relative Error (MRE) of non-zero terms as the criterion for the evaluation of the experimental results, where \(\theta_i\) represents the \(i\)-th parameter of the drift and diffusion terms and \(\hat{\theta}_i\) is the learned parameter from data. All the experiment were done on the MacBook Pro 2021 with an M1 chip. We summarize the notations used throughout the experiments in Table 1.

| Variable   | definition | Variable   | definition                       |
|------------|------------|------------|----------------------------------|
| \(L\)     | Number of time snapshots | \(\Delta t\) | Time interval of snapshots       |
| \(\gamma\) | Gaussian variance       | \(\rho_m\)  | Expectation of the \(m\)-th Gaussian kernel |
| \(M\)     | Gaussian sample number  | \(N_i\)    | Sample number at \(i\)-th time snapshot |
| MRE        | Maximum Relative Error  | \(\delta\) | Noise level*                     |

Table 1: Notations used throughout the experiments. *We add multiplicative noise by \(x = x + \delta U x\), where \(U\) is an uniform random variable in [-1,1]. See section 3.5 for details.

### 3.1 Typical 1-dimensional problem

One-dimensional stochastic problem widely exists in the scientific and engineering fields, such as the population growth, asset price and investments \cite{41}. To better illustrate the abilities of our WCR method for dealing with complex tasks, in this section, we focus on revealing the hidden dynamics from 1d aggregate data avoiding the difficulties brought by the dimension of the data. We consider the following 1d model

\[
\frac{dX_t}{dt} = \mu(X_t)dt + \sigma(X_t)dW_t,
\]

where drift \(\mu(X_t)\) and diffusion \(\sigma(X_t)\) terms reduce to scalar functions. Here we mainly focus on the five cases: (I) Cubic polynomial problem with only three snapshots; (II) Variable-dependent diffusion problem; (III) Quintic polynomial problem with high contrast; (IV) General form of the basis dictionary; (V) General drift term out of basis. And these cases give a direct illustration of the good performance of WCR on the complex tasks.

(I) **Cubic polynomial problem with three snapshots.** In this case, the raw data of the experiment are generated by the 1d cubic polynomial form

\[
\frac{dX_t}{dt} = (X_t - X_t^3)dt + dW_t
\]

(29)
in from $t = 0$ to $t = 1$ with initial values at $t = 0$ sampled from a Gaussian distribution $\mathcal{N}(0, 0.1)$ using Euler-Maruyama scheme \cite{27}. We obtained 10,000 samples at each snapshot, and only three snapshots at (a) $t = 0.1, 0.3, 0.5$ and (b) $t = 0.2, 0.5, 1$ are adopted as the experimental data. In this case, we expand the drift term as the third-order basis expansion form

$$\mu(x) = \lambda_0 + \lambda_1 x + \lambda_2 x^2 + \lambda_3 x^3,$$

and the diffusion term is treated as a tunable parameter $D(x) = \frac{1}{2} \sigma^2 = D_0$. We have used the same experimental setup for the same problem of Chen’s work \cite{26}.

Only three snapshots are available in the data set (a) and (b). Further the time intervals of the snapshots in (b) are not equal. Thus the approximation of the temporal derivatives requires a reliable scheme to overcome the difficulties. Here we use the variable step-size version of Implicit Adams methods of trapezoidal rule \cite{14}, which can be applied on small amount of time snapshots and non-equally spaced time series data. For the collocation kernels, 20 Gaussian kernels are sampled by sampling the expectations of the Gaussian function using LHS method with the standard variance set as $\gamma = 0.85$.

The results are shown in Figure \ref{fig:2} and Table \ref{tab:2}. We have compared the results with Chen’s results \cite{26} as the state of the art (sota) to show that for the same problem, WCR method can achieve a comparable accuracy but with a much less computational cost. All of the non-zero terms are correctly identified and all of the zero terms are eliminated by the sparse regression. The MRE of (a) and (b) are less than 4.6% and 0.42% respectively. Most notably, the experiments of applying WCR on the data set (a) and (b) are all completed within 0.02s on a MacBook Pro 2021 with an M1 chip.

For a mild amount of the data, WCR method still achieves remarkable performances. We sampled 1000 points each snapshot in the case (b) above without changing other setups to test our framework. In this case, WCR method still achieves a good result with the MRE less than 4% within only 0.006s on the MacBook pro.

Figure 2: The results of 1d cubic polynomial problem compared with Chen’s work (sota). Reveal the unknown drift and diffusion terms with 10000 samples of $X_t$ at different time snapshots: (a) Observations at $t = 0.1, 0.3, 0.5$; (b) Observations at $t = 0.2, 0.5, 1$ where the samples are generated by the given SDE with drift term $\mu_{\text{true}} = x - x^3$ and diffusion term $\sigma = 1$. The inference results are denoted by $\mu_{\text{true}}, \mu_{\text{chen}}$ (sota) and $\mu_{\text{wcr}}$ (ours).
Table 2: The results of 1d cubic polynomial problem compared with Chen’s work (sota). Reveal the unknown drift and diffusion terms with 10000 samples of $X_t$ at different time snapshots: (a) Observations at $t = 0, 0.1, 0.3, 0.5$; (b) Observations at $t = 0.2, 0.5, 1$ where the samples are generated by the given SDE with drift term $\mu_{\text{true}} = x - x^3$ and diffusion term $\sigma = 1$. The inference results are denoted by Chen (sota) and WCR (ours).

| Parameter | $\lambda_0$ | $\lambda_1$ | $\lambda_2$ | $\lambda_3$ | $\sigma$ |
|-----------|-------------|-------------|-------------|-------------|---------|
| True      | 0           | 1           | 0           | -1          | 1       |
| (a) Chen  | 0.0051      | 0.8422      | -0.0071     | -0.8994     | 1.0347  |
| (a) WCR   | 0           | **1.0160**  | 0           | **-1.0457** | **1.0127** |
| (b) Chen  | 0.0225      | 0.9638      | -0.0010     | -1.0035     | 1.0138  |
| (b) WCR   | 0           | **0.9982**  | 0           | **-1.0019** | **1.0042** |

Table 3: Experiment in section 3.1(I)(b) with drift term $\lambda_0 + \lambda_1 x + \lambda_2 x^2 + \lambda_3 x^3$ and diffusion term $\sigma$. The experiment setup is same with section 3.1(I)(b).

| Parameter | $\lambda_0$ | $\lambda_1$ | $\lambda_2$ | $\lambda_3$ | $\sigma$ |
|-----------|-------------|-------------|-------------|-------------|---------|
| True      | 0           | 1           | 0           | -1          | 1       |
| With sparse identification | 0 | 0.9982 | 0 | -1.0019 | 1.0042 |
| Without sparse identification | 0.0217 | 0.9584 | -0.0118 | -0.9751 | 1.0082 |

(II) Variable-dependent diffusion term problem. Many attentions in the existing literature are paid to the problem with constant diffusion in revealing the hidden dynamics from data. But a more general setting is the non-constant diffusion, i.e., the variable-dependent diffusion. In this part, we would show that WCR method also works for the variable-dependent case. The variable-dependent stochastic equation

$$dX_t = (\lambda_0 + \lambda_1 X_t + \lambda_2 X_t^2 + \lambda_3 X_t^3)dt + (\sigma_0 + \sigma_1 X_t)dW_t,$$

is considered as the true model to generate the raw data of the snapshots at time $t = 0, 0.2, 0.5, 1$. Our task is to reveal the true model from the data by the parameterized SDE form

$$dX_t = (\lambda_0 + \lambda_1 X_t + \lambda_2 X_t^2 + \lambda_3 X_t^3)dt + (\sigma_0 + \sigma_1 X_t)dW_t,$$

where $\lambda_i, i = 0, 1, 2, 3$ and $\sigma_j, j = 0, 1$ are the tunable parameters. Note that, the weak form of the Fokker-Planck is leveraged in our method, where the drift term $\mu(x)$ and the $1 \times 1$ diffusion matrix $D$ is revealed. By setting $D(x) = b_0 + b_1 x + b_2 x^2$, WCR method gives the approximation of the
values of $\lambda_i, i = 0, 1, 2, 3$ and $b_j, j = 0, 1, 2$. And by the relation $D = \frac{1}{2}\sigma^2$, a nonlinear regression is applied on the diffusion matrix $D$ and the parameters $\sigma_i, i = 0, 1$ are approximated, giving the explicit form of the governing equation (30). The results of the parameters are listed in Table 4 and WCR method still works well in the variable-dependent problem.

| Parameter | $\lambda_0$ | $\lambda_1$ | $\lambda_2$ | $\lambda_3$ | $\sigma_0$ | $\sigma_1$ |
|-----------|-------------|-------------|-------------|-------------|-------------|-------------|
| True      | 0           | 1           | 0           | -1          | 1           | 1           |
| WCR       | 0.9688      | 0           | -1.0264     | 0.9955      | 1.0326      |

Table 4: The results of 1d cubic polynomial problem when diffusion term is not a constant. Reveal the unknown drift and diffusion terms with 10000 samples of $X_t$ at $t = 0, 0.2, 0.5, 1$ where the samples are generated by the given SDE with drift term $\mu_{\text{true}} = x - x^3$ and diffusion term $\sigma = 1 + x$.

**Quintic polynomial drift problem.** Quintic polynomial drift has fifth-order polynomial terms, raising the difficulties of the revealing the true form of the hidden stochastic dynamics. The problem becomes more subtle when the coefficients of the polynomial terms are in different scale. In this experiment, we would show that for the high order polynomial drift term with high contrast coefficient, WCR method can still reach a good accuracy of modeling the data.

The raw data are generated by integrating the following SDE from $t = 0$ to $t = 10$

$$dX_t = \mu dt + dW_t,$$

where the drift term is the high order polynomial form with high contrast coefficient as

$$\mu \Delta = -x(x - 1)(x - 2)(x - 3)(x - 4) = -24x + 50x^2 - 35x^3 + 10x^4 - x^5.$$ (32)

In this experiment, WCR is applied on the aggregate data $X$ to reveal the hidden stochastic dynamics. Two kinds of the data with time interval (a) $\Delta t = 0.1$ and (b) $\Delta t = 0.5$ with $N = 5000$ points in each time snapshot are considered. To model the data, we take the fifth order of the polynomial expansion for the drift term with

$$\mu(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4 + \theta_5 x^5,$$

where $\theta_i, i = 0, 1, \ldots 5$ are the tunable parameters to be determined by the data. The diffusion term is chosen as one tunable parameter $D_0$.

In the collocation of the kernels, 200 Gaussian functions are sampled by randomly generating their expectations $\{\rho_i\}_{i=1}^{200}$ in the region of the data with the default standard variance $\gamma = 1$. For the approximation of the temporal derivatives, the Milne method is applied. The computation costs within 2 seconds and the results are summarized in Table 5. From the result, we can see that WCR method achieves a good performance in the higher problem with less than 3% Max Relative Error (MRE) within seconds on the Macbook Pro.

**General form of the basis dictionary.** As mentioned in Section 2.4, the dictionary represents are adopted for the unknown terms. In the above experiments, we simply choose the polynomial basis as the dictionary for their simplicity and interpretability. And polynomials were primarily employed as the basis in the experimental investigations conducted in this paper. Nevertheless, it should be noted that the proposed method is not restricted to polynomial bases and can be extended to more general functions. In this part, we depict a more general function dictionary which consists of both polynomial and trigonometric functions to demonstrate the abilities of our methods for the general form of the approximations.
Table 5: The results of one-dimensional high contrast problem with different time snapshot. Reveal the unknown drift and diffusion terms with (a) $\Delta t = 0.1$; (b) $\Delta t = 0.5$, where the samples are generated by the true SDE with drift term $\mu(x) = -24x + 50x^2 - 35x^3 + 10x^4 - x^5$ and diffusion term $\sigma = 1$, gaussian functions $M = 200$, samples number $N = 5000$.

The experimental data in this subsection was obtained by discretizing the following stochastic differential equation (33) using the Euler-Maruyama method with a step size of 0.1 from $t = 0$ to $t = 1$.

\[ dX_t = (X_t + \cos(3X_t))dt + dB_t \quad (33) \]

Then we expand the drift term using the following composite basis $\Lambda$.

\[
\Lambda = \left\{ \begin{array}{c}
1 \\
x \\
x^2 \\
x^3
\end{array} \right\} \otimes \left\{ \begin{array}{c}
1 \\
\cos x \\
\cos 2x \\
\cos 3x
\end{array} \right\}
\]

\[
\text{flatten} = \{1, \cos x, \cos 2x, \cos 3x, x, x \cos x, x \cos 2x, x \cos 3x, \cdots, x^3 \cos 2x, x^3 \cos 3x\} \quad (34)
\]

The basis is composed of a product of polynomials with degree no more than 3 and cosine functions with frequencies of 0, 1, 2, and 3. Therefore, there are a total of 16 terms, and with the addition of diffusion represented by a constant, there are 17 parameters to be solved. For the selection of test functions, 20 Gaussian functions with a variance of 1 were chosen, and their means were sampled from the data region using the Latin Hypercube Sampling method. There are 10,000 samples at each time point. The results are shown in Table 6. It can be seen that all the redundant base terms have been successfully eliminated by sparse regression. In the face of the problem of trigonometric basis, the WCR method can still achieve good performance with less than 4% maximum relative error.

Table 6: The results of one-dimensional trigonometric basis drift problem. Reveal the unknown dynamics with samples $N = 10000$ and gaussian functions $m = 20$. The drift term is $\mu(x) = x + \cos 3x$ and diffusion term $\sigma = 1$.

| coefficient | $x$ | $\cos 3x$ | other drift terms | diffusion | MRE | Time (s) |
|-------------|-----|-----------|-------------------|-----------|-----|---------|
| True        | 1.029 | 1.038    | 0                 | 1         | -   | -       |
| Learned     | 1    | 1        | 0                 | 1         | 1.0043 | 3.8%    |

(V) General drift term out of basis. To illustrate the performance of the basis expansion, we investigate the scenario of the complex drift term without complete expansion of the basis. Namely, in this subsection, we try to reveal the hidden dynamics from data of the given form

\[ dX_t = -2X_te^{-X_t^2}dt + dW_t \quad (35) \]
where the drift term $-2xe^{-x^2}$ cannot be completely expanded in finite polynomial basis. However, we would show that with the order of the polynomial basis increases, the revealing results still meet the needs.

We compare the $L_2$ relative error between the approximate solution under polynomial basis and the true drift terms. The data is observed at $t = 0, 0.1, 0.2, \cdots, 1$ with 10,000 samples at each snapshot. The diffusion terms and $L_2$ relative errors of the drift terms under different orders of polynomial basis are presented in Table 7. Figure 3 illustrates the functional graph of the drift terms. The calculation interval for relative error is $[-1, 1]$, because our method is a supervised learning approach, and the values of drift terms in regions without data are inherently unlearnable. As the order of polynomial basis increases, the results obtained by WCR gradually approach the true values of the drift terms. The best learning performance is achieved when the order is 9, with a relative error of 0.05% and a diffusion term of 0.9994.

| basis order | 3   | 4   | 5   | 6   | 7   | 8   | 9   |
|-------------|-----|-----|-----|-----|-----|-----|-----|
| $L_2$ relative error | 0.0734 | 0.0728 | 0.0242 | 0.0205 | 0.0042 | 0.0008 | 0.0005 |
| diffusion    | 0.9331 | 0.9332 | 0.9653 | 0.9699 | 0.9863 | 0.9984 | 0.9994 |

Table 7: The results of the drift terms $-2xe^{-x^2}$ under various orders of polynomial basis. The $L_2$ relative error of the drift terms is calculated over the interval $[-1, 1]$. The data was obtained from 10,000 observations at $t = 0, 0.1, 0.2, \cdots, 1$. And 20 Gaussian functions were used for revealing the dynamics.

3.2 2-dimensional problem with coupled drift terms

Sombrero potential $V$ is a well-known potential form that the symmetry breaking is triggered in the quantum mechanics. The gradient of the potential acts as the force and gives the drift term of the
stochastic process as $\mu = \nabla V$. The drift term induced by the Sombrero potential has the coupled terms in each dimension, i.e., $\mu_1$ and $\mu_2$ are not independent. To illustrate the abilities of revealing hidden stochastic dynamics with coupled drift terms, we apply WCR method on the aggregate data generated by the 2-dimensional Brownian motion with the coupled drift term

$$\mu = -\nabla V = \begin{pmatrix} 10x_1 - 4x_1^3 - 4x_1x_2^2 \\ 10x_2 - 4x_2x_1^2 - 4x_2^3 \end{pmatrix}, \quad (36)$$

which is induced by the gradients of the Sombrero potential

$$V = -5\|x\|^2 + \|x\|^4 = -5(x_1^2 + x_2^2) + (x_1^2 + x_2^2)^2. \quad (37)$$

The true model decorated by the coupled drift (36) has the SDE form

$$dx = \begin{pmatrix} c_{11}x_1^3 + c_{12}x_1^2 + c_{13}x_1x_2^2 + c_{14}x_1^2x_2^2 + c_{15}x_1x_2^3 + c_{16}x_1^2x_2^3 \\ c_{21}x_2^3 + c_{22}x_2^2 + c_{23}x_1x_2^2 + c_{24}x_1^2x_2^2 + c_{25}x_1x_2^3 + c_{26}x_1^2x_2^3 \end{pmatrix} dt + \begin{pmatrix} \sigma_1 \\ \sigma_2 \end{pmatrix} dW_t, \quad (38)$$

where $c_{11} = c_{21} = 10$, $c_{12} = c_{13} = c_{22} = c_{23} = -4$ corresponding to the coupled drift term (36) and $\sigma_1 = \sigma_2 = 1$ for simplicity.

The raw data are generated by integrating the SDE from $t = 0$ to $t = 15$ using Euler-Maruyama method. Take $N = 20,000$ samples out of the total 40,000 points in each snapshot and collect totally $L = 151$ snapshots at $t = 0, 0.1, \ldots, 15$, we have the aggregate data set $X = \{X\}_{L=1}^{12} = \{\{x_i\}_{j=1}^{N}\}_{L=1}^{42}$ for the experiment. In this subsection, we apply WCR method on the data to reveal the true model and give a direct illustration of the performance of our method.

To reveal the hidden dynamics of the 2-dimensional stochastic process, 200 gaussian kernels are randomly sampled in the regime of the data and the Milne method (12) is used to approximate the temporal derivatives. The elements of the unknown drift and coefficient term are approximated by

$$\Lambda = \{1, x_1, x_2, x_1^2, x_1x_2, x_2^2, x_1^3, x_1^2x_2, x_1x_2^2, x_2^3, x_1^3x_2, x_1^2x_2^2, x_1x_2^3, x_2^4\}, \quad (39)$$

with $|\Lambda| = 15$. For simplicity we denote the basis vector $\Lambda = (1, x_1, x_2, \ldots, x_2^4)^T$ which is the vectorization of $\Lambda$. Note that, the highest order of the drift term is three in the true model, but we have used a higher order term to approximate the unknown terms of both drift and diffusion terms. It follows that the unknown drift and diffusion terms are approximated by the following form

$$\mu = \begin{pmatrix} \rho_1 \\ \rho_2 \end{pmatrix} = \begin{pmatrix} \xi_1 \cdot \Lambda \\ \xi_2 \cdot \Lambda \end{pmatrix}, \quad D = \begin{pmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{pmatrix} = \begin{pmatrix} \xi_1^D \cdot \Lambda & \xi_2^D \cdot \Lambda \\ \xi_1^D \cdot \Lambda & \xi_2^D \cdot \Lambda \end{pmatrix}, \quad (40)$$

where $\xi_i \in \mathbb{R}^d$, $\xi_i^D \in \mathbb{R}^d$ are coefficient vectors with dimension $d = |\Lambda| = 15$. Thus for each drift term and each element of the diffusion matrix, we have 15 coefficients, making a total of 90 parameters to be revealed. By a sparse linear regression, instead of the 90 coefficients for a forth-order complete polynomial basis, only 8 coefficients corresponding to those in (38) are listed for simplicity, as the others are identified correctly as zero. All non-zero terms of the true model are not missed and no redundant coefficients which should be zeros are not superfluous by WCR method. The results are shown in Table 8. The Max Relative Error (MRE) is about 2.42% of the coefficients. The experiment is done on the Macbook pro within 21 seconds. The results in Table 8 illustrate the abilities of WCR method for revealing hidden dynamics with coupled terms under high-order expansion.
Table 8: The results of the 2-dimensional problem with Sombrebro potential. Reveal the unknown drift and diffusion terms with the expansion of forth-order complete polynomial basis. No redundant coefficients have been learned with only 8 out of the 90 coefficients nonzero corresponding to the true model. The Max Relative Error of the non-zero coefficients is 2.42%.

### 3.3 Multi-dimensional problem

Many methods depending on the integration over space always get stuck when the dimension of the data increases because of the exponential increase of the computational cost. Even under three or four dimension, the integration is computationally expensive, despite of the incapacity for higher dimensions such as 10d or 20d. In the contrast, WCR method takes a more subtle strategy to avoid the direct computation of the integral. Thanks to the weak form, the spatial derivatives of the probability function have been transferred to the test function, making the integral simply computed by the summation over the samples avoiding the curse of the dimensionality. Hence WCR method can be naturally extended to the higher dimension other than one or two dimension.

In this subsection, we take the same setup consistent with the 3d and 4d problems in [26] and to show that WCR method can achieve a high accuracy within seconds. The true model is d-dimensional extension of the SDE (29) with the form

\[
dX_t^i = (X_t^i - (X_t^i)^3) dt + dW_t^i, \quad i = 1, 2, \ldots, d, \tag{41}
\]

where the drift term is \(\mu_i = x_i - x_i^3\) in the \(i\)-th dimension. The raw data are generated by the true model using Euler-Maaruma method. To get the aggregate data set, 100,000 samples each snapshot are taken and the snapshots at (i) \(t = 0.1, 0.3, 0.5, 0.7, 1\) (ii) \(t = 0.1, 0.2, 0.3, 0.5, 0.7, 0.9, 1\) are collected as the experimental data.

In the experiment, the unknown drift term in each dimension is approximated by

\[
\hat{\mu}_i = \theta_0 + \theta_1 x_i + \theta_2 x_i^2 + \theta_3 x_i^3, \tag{42}
\]

and the diffusion term approximated by a tunable parameter \(D_i, i = 1, 2, \ldots, d\). 100 Gaussian kernels are sampled by the expectations using LHS method with the default standard deviation where \(\gamma = 1\). We list the Max Relative errors and the computational time in Table 9 and the results show our WCR method achieves a high accuracy in the 3d and 4d cases within seconds on the MacBook Pro. Figure ??fig.PK3d4d) depicts the learned drift terms compared with the true ones.

### 3.4 High-dimensional problem

For three or four dimensional data, WCR method shows a high accuracy of revealing the hidden stochastic dynamics. Higher dimensional problem especially over ten dimensional brings more challenges to the modeling task because of the so-called curse of the dimensionality. In this subsection, we would show that WCR method can be applied on the high dimensional problems thanks to the Monte-Carlo approximation of the weak form. We consider the 10 and 20 dimensional aggregate data generated by the true model

\[
dX_t^i = (X_t^i - (X_t^i)^3) dt + dW_t^i, \quad i = 1, 2, \ldots, d, \tag{43}
\]
Table 9: The Max Relative Error of the learned coefficients and the computational time on the MacBook Pro of revealing the unknown 3d, 4d cubic polynomial problems. The aggregate data \( X \) is composed of the time snapshots at (i) \( t = 0.1, 0.3, 0.5, 0.7, 0.9 \) (ii) \( t = 0.1, 0.2, 0.3, 0.5, 0.7, 0.9, 1 \) with 100,000 points in each snapshot. The unknown dynamics is approximated by the drift \( \hat{\mu}_i = \theta_0 + \theta_1 x_i + \theta_2 x_i^2 + \theta_3 x_i^3 \) and diffusion \( D_i \) in each dimension with tunable parameters \( \theta_i \) and \( D_i \), \( i = 1, 2, \ldots, d \). And 100 gaussian kernels are sampled to give the composed linear system.

|       | 3D-(i) | 3D-(ii) | 4D-(i) | 4D-(ii) |
|-------|--------|---------|--------|---------|
| MRE   | 3.39%  | 1.86%   | 7.04%  | 3.18%   |
| Time(s) | 3.6    | 4.7     | 6.1    | 7.9     |

Figure 4: The learned drift terms of the 3d and 4d problems. Reveal the unkown drift and diffusion terms with 100,000 samples each snapshot at time \( t = 0.1, 0.3, 0.5, 0.7, 1 \) of (a) 3-dimensional problem; (b) 4-dimensional problem, where the samples are generated by the given SDE with drift term \( \mu_{\text{true}} = x - x^3 \) and diffusion term \( \sigma = 1 \) in per dimension. The inference results are denoted by \( \mu_{\text{true}} \) and \( \mu_{i, \text{wcr}} \) for the \( i \)-th dimension.
with $d = 10, 20$ using Euler-Maruyama method. Several time snapshots of the data at time $t = 0, 0.1, 0.2, 0.3, \cdots, 1$ are used as the experiment data.

In the experiment, the fourth order basis is used to approximate the drift term in each dimension as $\mu_i = \theta_{00} + \theta_{01} x_1 + \theta_{02} x_2^2 + \theta_{03} x_3^3$ and the diffusion term is approximated by a tunable parameter $\sigma$, making totally $5d$ tunable parameters to be learned. Milne method is used for the temporal derivatives. Different number of the data samples in each snapshot and the Gaussian kernels are considered in the high dimensional problem.

In the 10-dimensional problem, we consider the four cases for samples and kernels (a) 10,000 samples of $X_t$ using 1000 Gaussian kernels; (b) 100,000 samples of $X_t$ using 1000 Gaussian kernels; (c) 10,000 samples of $X_t$ using 10,000 Gaussian kernels; (d) 100,000 samples of $X_t$ using 10,000 Gaussian kernels. The four cases are the combinations of the 10,000 and 100,000 samples with 1000 and 10,000 Gaussian kernels. Figure 5 gives the display of the learned drift terms compared with the true model in each case. From the Figure, it is easy to check that the learned results get better when the number of the sample points and Gaussian kernels increases. And the Max Relative Error for the best result in case (d) achieves less than 7.8% within 20 minutes. We also employ a coupled system in ten dimensions to showcase the capabilities of the WCR method. For detailed information, please refer to Appendix C.

In the 20-dimensional case, we take 10000 gaussian kernels and 100000 samples of $X_t$. The experiment is completed within 47 minutes on the MacBook Pro and the result is shown in Figure 6.

### 3.5 Performance on data of different qualities

In real applications, the data obtained by various means usually contain many missing points and even flaws. The data may be also noisy with useful signals deeply buried. These facts make analyzing and extracting useful models from the data hard and tricky. We have shown that WCR performs well with only several snapshots with non-equally spaced data in the one dimensional case. In this subsection, we further investigate the performance of WCR on the data with different qualities, such as short and long time intervals, different number of the samples in the snapshots, and noises.

The true model in this experiment has the following form

$$
\begin{align*}
    dx_1 &= -0.5x_1 dt + dW_t \\
    dx_2 &= -0.7x_2 dt + dW_t \\
    dx_3 &= -x_3 dt + dW_t,
\end{align*}
$$

(44)

where the three dimensions are not coupled for the simplicity and the mainly focus on the data qualities. The raw data are generated by integrating the true model from $t = 0$ to $t = 10$, and different kinds of snapshots are chosen as the aggregate data. Noise is added for the study of the noise affect.

To reveal the hidden dynamics, the unknown drift term is expanded by the first order complete polynomials as

$$
\text{drift} = \begin{pmatrix}
    \theta_{00} + \theta_{01} x_1 + \theta_{02} x_2 + \theta_{03} x_3 \\
    \theta_{10} + \theta_{11} x_1 + \theta_{12} x_2 + \theta_{13} x_3 \\
    \theta_{20} + \theta_{11} x_1 + \theta_{22} x_2 + \theta_{23} x_3
\end{pmatrix},
$$

(45)

and the diffusion matrix is approximated by diagonal matrix $D = \begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{pmatrix}$, where $[\theta_{ij}]$ and $[\sigma_i]$ are tunable parameters to be revealed.
Figure 5: The results of 10d problems with different number of samples per time snapshot and gaussian kernels. Reveal the unknown drift and diffusion terms with (a) 10000 samples of $X_t$ using 1000 gaussian kernels; (b) 100000 samples of $X_t$ using 1000 gaussian kernels; (c) 10000 samples of $X_t$ using 10000 gaussian kernels; (d) 100000 samples of $X_t$ using 10000 gaussian kernels at time snapshots $t = 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1$ of 10-dimensional problem, where the samples are generated by the given SDE with drift term $\mu_{\text{true}} = x - x^3$ and diffusion term $\sigma = 1$ in per dimension. The inference results are denoted by $\mu_i$ for the learned drift in the $i$-th dimension.
Figure 6: The results of 20-dimensional problems with 100000 samples of $X_t$ using 10000 gaussian kernels at time $t = 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1$, where the samples are generated by the given SDE with drift term $\mu_{\text{true}} = x_i - x_i^3$ and diffusion term $\sigma = 1$ in per dimension. The inference results are denoted as $\mu_i$ for the $i$-th dimensional drift.
**Time interval and sample number.** We investigate in this part how the time interval and the sample number of the data affect the accuracy of the results in our WCR method. We consider four cases of the time interval $\Delta t_1 = 0.1$, $\Delta t_1 = 0.2$, $\Delta t_1 = 0.5$ and $\Delta t_1 = 0.9$ of the snapshots in the total interval $[0, 10]$. To be concise, the total number of the snapshots $L = N_t + 1$, where $T = 10$ and $\Delta t$ is one of the time interval. For example we have $L = 101$ snapshots in the case of $\Delta t_1 = 0.1$ with each snapshot are equally spaced with $\Delta t_1 = 0.1$. For the number of the samples in each snapshot, we also consider four cases with $N_1 = 1000$, $N_2 = 2000$, $N_3 = 5000$, and $N_4 = 10000$. Thus we have totally tested $4 \times 4 = 16$ cases to investigate the performance of WCR on the time interval and sample number of the data.

In all experiments, other setups are kept the same where 100 Gaussian functions with standard variance $\gamma = 1$ are sampled as the test functions. And the Milne method is used for the temporal derivatives. Three different random seeds are used in each experiment for the average result. All experiments share the same three seeds for the fair comparison.

We summarize all the results in Figure 7. With only $N = 1000$ samples and mild time interval $\Delta t = 0.9$, the maximum relative error can reach a good accuracy around 5%. Further, with the increasing number of the samples in each snapshot and the decreasing time interval from $\Delta t_4 = 0.9$ to $\Delta t_1 = 0.1$, from $N_1 = 1000$ to $N_2 = 10000$, the results generally get better. And all of the cases result in a low Max Relative Error giving the proof of the robustness of our method in relative poor data qualities.

![Figure 7](image)

Figure 7: The heat map of the errors of the drift coefficients and diffusion for the 3-dimensional stochastic process in $T = [0, 10]$ with different trajectory samples $N_1 = 1000$, $N_2 = 2000$, $N_3 = 5000$, $N_4 = 10000$ and different time interval $\Delta t_1 = 0.1$, $\Delta t_2 = 0.2$, $\Delta t_3 = 0.5$, and $\Delta t_4 = 0.9$. The true governing stochastic equation reads $dx_1 = -0.5x_1 dt + dW_1$, $dx_2 = -0.7x_2 dt + dW_2$, and $dx_3 = -x_3 dt + dW_3$, where the coefficients of $\mu_1$, $\mu_2$, and $\mu_3$ are $-0.5$, $-0.7$ and $-1$, and the diffusion $\sigma_1 = \sigma_2 = \sigma_3 = 1$. (a) shows the max relative error of the coefficients for $N_i$ and $\Delta t_j$; (b) shows the relative error of per coefficient for $N_i$ and $\Delta t_j$ with $i, j = 1, 2, 3, 4$.

**Noise.** In this part, we consider the performance of our WCR method on the noisy data. The raw data are obtained by the same procedure as the above and the random noise is added with the noise level $\delta$ $\tilde{x}_i = x_i + \delta U_i x_i$, where $U_i$ is an uniform random variable in $[-1, 1]$. Here the data with 10000 samples in each snapshot are used and time intervals $\Delta t = 0.1$ are chosen to show the robustness of our methods on the noise.
The results are listed in Table 10 and the Max Relative Error reaches around 0.6% with the white noise level $\delta = 10\%$ for both two cases. Further, when we dive into the detail of the accuracy for drift and diffusion terms respectively, adding noise to the data didn’t change the accuracy of the diffusion term much unlike the drift term.

| Noise $\delta$ | $\delta = 0\%$ | $\delta = 10\%$ | $\delta = 20\%$ | $\delta = 30\%$ | $\delta = 40\%$ |
|----------------|----------------|----------------|----------------|----------------|----------------|
| MRE            | 0.76%          | 0.63%          | 1.92%          | 5.62%          | 11.69%         |
| MRE in drift   | 0.49%          | 0.54%          | 1.92%          | 5.62%          | 11.69%         |
| MRE in diffusion| 0.76%          | 0.63%          | 0.28%          | 0.94%          | 2.36%          |

Table 10: The results of Three-dimensional linear problem with different time snapshot and noise. Reveal the unknown drift and diffusion terms with $\Delta t = 0.1$ and $\delta = 0, 10, 20, 30, 40\%$, where 10000 samples each snapshot are generated by the given SDE with drift term $\mu(x, y, z) = [-0.5x, -0.7y, -z]^T$ and diffusion term $\sigma = I_{3 \times 3}$.

4 Conclusion

In this work, leveraging the weak form of the Fokker-Planck equation and the collocations of the Gaussian kernels, we proposed a framework called the Weak Collocation Regression method (WCR) to fast reveal the hidden stochastic dynamics from high-dimensional aggregate data. The lack of trajectory information makes aggregate data (unpaired data) more difficult to reveal the hidden dynamics. However, the data distribution follows the Fokker-Planck equation under some assumptions, such as the Brownian motion. By transferring the spatial derivatives to the test function in the weak form of the Fokker-Planck equation, we have an integral form of the density function with derivatives of the test functions. Thus the integral form can be easily approximated by summing the derivatives of the test functions over the samples. Using collocations of the Gaussian functions as the test functions and the dictionary representation of the unknown terms, we build an extensive linear system. Linear regression and sparse identification lead to unknown terms, thus revealing the hidden dynamics.

Our experiments show that WCR is numerically efficient, taking only 0.02s on the Macbook Pro for the 1-dimensional problem with remarkable accuracy and only seconds for 3-dimensional and 4-dimensional problems. With the LMMs of variable step size, WCR achieves good accuracy in the non-equally spaced time data. Approximating the integral form using the summations over samples makes the WCR method naturally relieve the curse of dimensionality. And the extra computational cost is required on the collocations of the kernels but does not exponentially increase with dimension. The variable-dependent diffusion problem and coupled drift terms usually bring more difficulties, but the WCR method exhibits high accuracy in these complex problems. And for the data with noise, the WCR method also obtains a stable performance when adding the white noise. With different time intervals and samples, WCR shows its robustness without losing too much accuracy when the data quality worsens.

When comparing with the methods presented in the literature, [19] employed SDE to generate data at encrypted time points and optimized it using the Wasserstein distance. Similarly, [20] used SDE to artificially augment data and compared its distance with existing data using GANs. On the other hand, [26] incorporated the residual of the FP equation across the entire space and time into the objective function, computing the derivatives through auto-differentiation. In contrast, our methods directly solve the algebraic equations of unknown terms using the available data. The
accuracy of the results obtained greatly relies on the precision of the derivatives approximated by the data. However, by utilizing Gaussian functions as test functions in the weak form, we significantly alleviate this issue as it eliminates the need for spatial derivatives. Consequently, the only remaining task is to accurately approximate the temporal derivatives of the one-dimensional sequence, as shown in equation \[9\]. It is important to note that the accuracy still depends on the precision of the temporal derivatives, necessitating relatively dense snapshots over time.

Despite the advantages mentioned above, the Weak Collocation Regression method still has some limitations for investigation in future work. Our approach slows down for the ultimately high-dimensional problems such as 100 dimensions since more Gaussian kernel collocations are needed, requiring high computational costs. More effective sampling schemes, such as active learning for the collocation of the Gaussian kernels, are necessary to reduce the computational cost. Neural networks might be another choice for the test function with a min-max optimization for reducing the cost. If prior information about the problem is available, it can be effectively incorporated into the dictionary, thereby mitigating the curse of dimensionality. However, if we lack prior information regarding the high-dimensional dynamics, expanding the unknown terms in a general basis form would lead to a significant increase in the number of terms. Consequently, traditional linear regression may prove less efficient. To address this issue, it may be worth considering a neural network approximation for the unknown terms. By adopting this approach, our framework remains intact, albeit with the substitution of linear regression for machine learning optimization schemes like gradient descent. However, this particular modification is left as an avenue for future research and exploration. Further discussion about the general form besides the Brownian motion is needed but also left for future work.

Acknowledgements

This work was supported by the National Key R&D Program of China (Grant No. 2021YFA0719200) and the National Natural Science Foundation of China (Grant No. 92370125). The authors would like to thank the helpful discussions from Dr. Liu Hong.

References

[1] Roger Temam. *Navier-Stokes equations: theory and numerical analysis*, volume 343. American Mathematical Soc., 2001.

[2] Leslie E Ballentine. *Quantum mechanics: a modern development*. World Scientific Publishing Company, 2014.

[3] Peter Klein. Pricing black-scholes options with correlated credit risk. *Journal of Banking & Finance*, 20(7):1211–1229, 1996.

[4] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[5] George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.

[6] Maziar Raissi, Alireza Yazdani, and George Em Karniadakis. Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. *Science*, 367(6481):1026–1030, 2020.
[7] Steven L Brunton and J Nathan Kutz. *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press, 2022.

[8] Pipi Hu, Wuyue Yang, Yi Zhu, and Liu Hong. Revealing hidden dynamics from time-series data by odenet. *Journal of Computational Physics*, 461:111203, 2022.

[9] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

[10] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937

[11] Zichao Long, Yiping Lu, Xianzhong Ma, and Bin Dong. Pde-net: Learning pdes from data. In *International Conference on Machine Learning*, pages 3208–3216, 2018.

[12] Zichao Long, Yiping Lu, and Bin Dong. Pde-net 2.0: Learning pdes from data with a numeric-symbolic hybrid deep network. *Journal of Computational Physics*, 399:108925, 2019.

[13] Gert-Jan Both, Subham Choudhury, Pierre Sens, and Remy Kusters. Deepmod: Deep learning for model discovery in noisy data. *Journal of Computational Physics*, 428:109985, 2021.

[14] Zhao Chen, Yang Liu, and Hao Sun. Physics-informed learning of governing equations from scarce data. *Nature communications*, 12(1):1–13, 2021.

[15] Guochang Lin, Pipi Hu, Fukai Chen, Xiang Chen, Junqing Chen, Jun Wang, and Zuoqiang Shi. Binet: Learning to solve partial differential equations with boundary integral networks. *arXiv preprint arXiv:2110.00352*, 2021.

[16] Qin Lou, Xuhui Meng, and George Em Karniadakis. Physics-informed neural networks for solving forward and inverse flow problems via the boltzmann-bgk formulation. *Journal of Computational Physics*, 447:110676, 2021.

[17] Lu Lu, Pengzhan Jin, and George Em Karniadakis. Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators. *arXiv preprint arXiv:1910.03193*, 2019.

[18] Daniel Dufresne. The integral of geometric brownian motion. *Advances in Applied Probability*, 33(1):223–241, 2001.

[19] Shaojun Ma, Shu Liu, Hongyuan Zha, and Haomin Zhou. Learning stochastic behaviour from aggregate data. In *International Conference on Machine Learning*, pages 7258–7267. PMLR, 2021.

[20] Liu Yang, Constantinos Daskalakis, and George E Karniadakis. Generative ensemble regression: Learning particle dynamics from observations of ensembles with physics-informed deep generative models. *SIAM Journal on Scientific Computing*, 44(1):B80–B99, 2022.

[21] Samuel Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.
[22] Daniel Alshamaa, Aly Chkeir, Farah Mourad-Chehade, and Paul Honeine. A hidden markov model for indoor trajectory tracking of elderly people. In 2019 IEEE Sensors Applications Symposium (SAS), pages 1–6. IEEE, 2019.

[23] Maryam Baradaran Khalkhali, Abedin Vahedian, and Hadi Sadoghi Yazdi. Vehicle tracking with kalman filter using online situation assessment. Robotics and Autonomous Systems, 131:103596, 2020.

[24] Nuno Pessanha Santos, Victor Lobo, and Alexandre Bernardino. Unmanned aerial vehicle tracking using a particle filter based approach. In 2019 IEEE Underwater Technology (UT), pages 1–10. IEEE, 2019.

[25] Yongkun Fang, Chao Wang, Wen Yao, Xijun Zhao, Huijing Zhao, and Hongbin Zha. On-road vehicle tracking using part-based particle filter. IEEE transactions on intelligent transportation systems, 20(12):4538–4552, 2019.

[26] Xiaoli Chen, Liu Yang, Jinqiao Duan, and George Em Karniadakis. Solving inverse stochastic problems from discrete particle observations using the fokker–planck equation and physics-informed neural networks. SIAM Journal on Scientific Computing, 43(3):B811–B830, 2021.

[27] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In International conference on machine learning, pages 214–223. PMLR, 2017.

[28] Richard E Bellman. Adaptive control processes. In Adaptive Control Processes. Princeton university press, 2015.

[29] Hannes Risken. Fokker-planck equation. In The Fokker-Planck Equation, pages 63–95. Springer, 1996.

[30] Yaohua Zang, Gang Bao, Xiaojing Ye, and Haomin Zhou. Weak adversarial networks for high-dimensional partial differential equations. Journal of Computational Physics, 411:109409, 2020.

[31] Gang Bao, Xiaojing Ye, Yaohua Zang, and Haomin Zhou. Numerical solution of inverse problems by weak adversarial networks. Inverse Problems, 36(11):115003, 2020.

[32] Samuel H Rudy, Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Data-driven discovery of partial differential equations. Science Advances, 3(4):e1602614, 2017.

[33] Herman Heine Goldstine. A History of Numerical Analysis from the 16th through the 19th Century, volume 2. Springer Science & Business Media, 2012.

[34] Kendall Atkinson, Weimin Han, and David E Stewart. Numerical solution of ordinary differential equations, volume 108. John Wiley & Sons, 2011.

[35] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Multistep neural networks for data-driven discovery of nonlinear dynamical systems. arXiv preprint arXiv:1801.01236, 2018.

[36] Rachael T Keller and Qiang Du. Discovery of dynamics using linear multistep methods. SIAM Journal on Numerical Analysis, 59(1):429–455, 2021.

[37] Qiang Du, Yiqi Gu, Haizhao Yang, and Chao Zhou. The discovery of dynamics via linear multistep methods and deep learning: Error estimation. SIAM Journal on Numerical Analysis, 60(4):2014–2045, 2022.
Appendix

A Error Analysis

In this section, we are going to analyze the error of the Weak Collocation Regression (WCR) method. The key ingredients in our error analysis are inspired by [39, 44]. The error in our method mainly consists of the following aspects: (a) approximations of the integrals; (b) approximations of the temporal derivative; (c) the linear regression. Briefly speaking, the error in computing the integrals by Monte Carlo method is $O\left(\frac{1}{\sqrt{N}}\right)$, and the error in approximating the temporal derivative by the linear multistep method is $O(\Delta t^\alpha)$. In this section, our primary focus lies in examining the influence of sample size $N$ and the time interval $\Delta t$ on the error. The order of the linear multistep method is denoted as $\alpha$, and the ordinary least squares method is used.

In a general context, we represent the linear multistep method for the differential equation $\frac{d}{dt}y(t) = f(t)$ as $y_{n+k} = \sum_{i=0}^{k-1} \beta_i y_{n+i} + \sum_{i=0}^{k-1} \alpha_i f_{n+i}$. Consequently, we define the temporal discrete operator for the left-hand side as $D_L f(t, y) := \frac{1}{\Delta t}(y(t + k\Delta t) - \sum_{i=0}^{k-1} \alpha_i y(t + i\Delta t))$, while the right-hand side is represented as $D_R f(t, y) := \sum_{i=0}^{k} \beta_i f(t + i\Delta t, y(t + i\Delta t))$. Once the library $\Lambda = \{\Lambda_1(x), \Lambda_2(x), \cdots, \Lambda_b(x)\}$ and the test functions $\{\psi_m\}_{m=1}^M$ are fixed, the true value $A \in \mathbb{R}^{M(L-k) \times (db+d^2b)}$ of the coefficient matrix $A \in \mathbb{R}^{M(L-k) \times (db+d^2b)}$ is determined as follows:

\[
X_{im}^{ij} = \begin{cases} 
\sum_{l=0}^{k} \beta_l \int_{\mathbb{R}^d} p(t_l, x) \Lambda_j(x) \cdot \nabla \psi_m(x) dx, & 1 \leq j \leq db \\
\sum_{l=0}^{k} \beta_l \int_{\mathbb{R}^d} \sum_{r,s=1}^{d} p(t_l, x) \Lambda_j(x) \frac{\partial}{\partial x_r x_s} \psi_m(x) dx, & db + 1 \leq j \leq db + d^2b 
\end{cases},
\]

\[
A = \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_M \end{pmatrix}, \quad (46)
\]
where $\mathbf{A}_m := (\mathbf{A}_m^i)^T \in \mathbb{R}^{(L-k) \times (db+d^2b)}$ is the block within $\mathbf{A}$ derived from test function $\psi_m$, and the operation $\sum_{l=0}^k \beta_l \times (\cdot)$ represents the linear multistep method. Importantly, it’s noteworthy that the true value $\mathbf{A}$ remains unaffected by both sample number $N$ and time interval $\Delta t$, thus establishing its independence from the chosen computational format. Subsequently, we list the assumptions for the problem.

**Assumption 1.**

1. The aggregate data $X = \{x_{i,n}^j\}_{i=1}^N$ is observed from a strong solution to \[ \partial_t \psi(x) = -\nabla \cdot (A \nabla \psi(x)) \] for $t_j \in [0, T]$ with constant sample number.

2. The probability density function $p(x, t)$ is smooth enough in the time direction. \[ \left( \frac{\partial}{\partial t} p(x, t) \right)_{t=0}^{N} \text{ is uniformly bounded with respect to } x. \]

3. The test functions $\{\psi_k\}_{k=1}^{M}$ belong to the Schwartz space \[ S(\mathbb{R}^d) := \{ f \in C^\infty(\mathbb{R}^d) \mid \sup_{x \in \mathbb{R}^d} |x^\alpha D^\beta f(x)| < \infty, \forall \alpha, \beta \} \]

The functions in library $\Lambda$ have algebraic growth. The test functions together with the library $\Lambda$ are such that the true value matrix $\mathbf{A}$ has full column rank.\[ \text{Subsequently, relying on assumption 1(3), it follows that } \mathbf{A}_m = 1 \forall m \in \mathbb{R}^{(L-k) \times (db+d^2b)} \text{ has full column rank.} \]

4. Moreover, $\|\mathbf{A}\|_\infty \leq C_2$ almost surely, where $\|\mathbf{A}\|_\infty$ is the induced matrix $\infty$-norm of $\mathbf{A}$.

5. The true functions $\mu^r$ and $D^r$ are in the span of $\Lambda$.

The coefficient matrix $\mathbf{A} \in \mathbb{R}^{M(L-k) \times (db+d^2b)}$ used in linear regression is provided by the following expression:

\[
\mathbf{A}^{ij} = \sum_{l=0}^k \beta_l \left( \frac{1}{N} \sum_{n=1}^N A_j(x_{i,n}^r) \cdot \nabla \psi_m(x_{i,n}^r) \right), \quad 1 \leq j \leq db
\]

\[
\sum_{l=0}^k \beta_l \left( \frac{1}{N} \sum_{n=1}^N \sum_{r,s=1}^d A_j(x_{i,n}^r) \frac{\partial}{\partial x_r x_s} \psi_m(x_{i,n}^s) \right), \quad db + 1 \leq j \leq db + d^2b
\]

where $\mathbf{A}_m := (\mathbf{A}_m^i)^T \in \mathbb{R}^{(L-k) \times (db+d^2b)}$ is the block within $\mathbf{A}$ derived from test function $\psi_m$, and $x_{i,n}^r$ represents the nth sample at time $t_i$. Here the coefficient matrix $\mathbf{A}$ exhibits a relationship with sample number $N$, but it remains independent of the time interval $\Delta t$. It becomes evident that the size of matrix $\mathbf{A}$ and $\mathbf{X}$ are congruent, with the difference in corresponding elements quantified as $\mathbf{X} - \mathbf{A} = \mathcal{O}(\sqrt{N})$. Subsequently, relying on assumption 1(3), it follows that $\mathbf{A}$ also attains full column rank for sufficiently large values of $N$. Likewise, in the ordinary least squares method, $\mathbf{A}^T := (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T$, which also implies $\|\mathbf{A}\|_\infty = \|\mathbf{X}\|_\infty + \mathcal{O} \left( \frac{1}{\sqrt{N}} \right)$. Therefore, in accordance with assumption 1(4), we can deduce that $\|\mathbf{A}\|_{\infty} \leq 2C_2$ for sufficiently large values of $N$.

Given a test function $\psi(x)$ and any probability density function $\rho(x, t)$, define the continuous-time weak-form residual

\[
\mathcal{R}(\rho(x, t), \psi(x)) = \int_{\mathbb{R}^d} \frac{\partial}{\partial t} \rho(x, t) \psi(x) dx - \int_{\mathbb{R}^d} \rho(x, t) \mu(x) \cdot \nabla \psi(x) dx - \int_{\mathbb{R}^d} \rho(x, t) \sum_{r,s} D_{rs} \partial_{rs} \psi(x) dx,
\]

\[\text{Remark 2.4 of [44].}\]
If \( \rho(x, t) = p(x, t) \) is a weak solution to \([\text{3}]\) and then the residual \( \mathcal{R}(\rho(x, t), \psi(x)) = 0 \) for any \( t \) and test function \( \psi \). To incorporate discrete effects, we introduce the discrete-time weak-form residual \( \mathcal{R}_{\Delta t}(\rho(x, t), \psi(x)) \) by replacing the differential operator with the LMM discret

\[
\mathcal{R}_{\Delta t}(\rho(x, t), \psi(x)) = D_t^L \int_{\mathbb{R}^d} \rho(x, t)\psi(x)dx - D_t^R \int_{\mathbb{R}^d} \rho(x, t)\mu(x) \cdot \nabla \psi(x)dx + \rho(x, t) \sum_{rs} D_{sr} \partial_{rs} \psi(x)dx,
\]

(49)

Next, we estimate the errors induced by the discrete-time derivative and Monte Carlo computation of expectations respectively. And the final error analysis is stated in Theorem \([\text{1}]\) by incorporating the linear regression.

**Step 1: Estimate the error of linear multistep method**

We first estimate the approximation error of discrete time derivative, which we denoted as \( \mathcal{E}_1(\rho(x, t), \psi(x)) \) for convenience

\[
\mathcal{E}_1(\rho(x, t), \psi(x)) = \mathcal{R}(\rho(x, t), \psi(x)) - \mathcal{R}_{\Delta t}(\rho(x, t), \psi(x)).
\]

(50)

Using assumption \([\text{1}]\) we can deduce the truncation error

\[
|\partial_t p(x, t) - D_t^L p(x, t) + (D_t^R p(x, t) - p(x, t))\mu(x) \cdot \nabla \psi(x) + (D_t^R p(x, t) - p(x, t)) \sum_{rs} D_{sr} \partial_{rs} \psi(x)| \leq C_3 \Delta t^\alpha
\]

(51)

where \( C_3 \) is independent of \( \Delta t \), and directly implies the order of approximation error

\[
|\mathcal{E}_1(p(x, t), \psi(x))| \leq \left| \int_{\mathbb{R}^d} |\psi(x)| \cdot |\partial_t p(x, t) - D_t^L p(x, t) + (D_t^R p(x, t) - p(x, t))\mu(x) \cdot \nabla \psi(x) + (D_t^R p(x, t) - p(x, t)) \sum_{rs} D_{sr} \partial_{rs} \psi(x)| dx \right|
\]

\[
\leq \int_{\mathbb{R}^d} |\psi(x)| \cdot C_3 \Delta t^\alpha dx \leq C_4 \Delta t^\alpha
\]

Since \( \psi \) belongs to the Schwartz space, the spatial integration is bounded. Here \( C_4 \) is dependent of the test function \( \psi \), but independent of \( \Delta t \).

**Step 2: Estimate the error of Monte Carlo method**

Then we consider the difference between exact probability density function \( p(x, t) \) and empirical density function \( p^N(x, t) \) with discrete temporal differential operator. Similarly, the approximation error is denoted as \( \mathcal{E}_2(p(x, t), p(x)) \)

\[
\mathcal{E}_2(\rho(x, t), \psi(x)) = \mathcal{R}_{\Delta t}(\rho(x, t), \psi(x)) - \mathcal{R}_{\Delta t}(\rho^N(x, t), \psi(x)).
\]

(52)

Then

\[
|\mathcal{E}_2(p(x, t), \psi(x))| \leq \left| \int_{\mathbb{R}^d} |\psi(x)| \cdot |D_t^L p(x, t) - D_t^L p^N(x, t) + D_t^R (p^N(x, t) - p(x, t))\mu(x) \cdot \nabla \psi(x) + D_t^R (p^N(x, t) - p(x, t)) \sum_{rs} D_{sr} \partial_{rs} \psi(x)| dx \right|
\]
which can be decomposed of two type of term. The first type is the inner product of test function and probability density function with discrete temporal differential operator. Hence, by setting $\alpha_k = 1$, the first term can be estimated by

$$\left| \int_{\mathbb{R}^d} \psi(x) \mathcal{D}_t^L \rho(x, t) - \psi(x) \mathcal{D}_t^L \rho^N(x, t) \, dx \right|$$

$$\leq \sum_{i=0}^{\infty} |\alpha_i| \left| \int_{\Omega} \psi(x) \left( \frac{\rho(x, t + i \Delta t) - \rho^N(x, t + i \Delta t)}{\Delta t} \right) \, dx \right| \triangleq I_1,$$

by the result of Theorem 1.1 of [45], we have

$$\mathbb{E} \left[ \int_{\Omega} \psi(x) \left( \rho(x, t) - \rho^N(x, t) \right) \, dx \right] \leq \frac{C_6}{\sqrt{N}}$$

with $C_6$ depending on test function $\psi$, but independent with $N, t, \Delta t$. So

$$\mathbb{E}[I_1] \leq \frac{C_6}{\sqrt{N \Delta t}}.$$

Here $C_6$ is independent of $\Delta t, N$, and dependent of test function $\psi$. The other terms are of the form $\left| \int_{\mathbb{R}^d} \Phi(\mathbf{r}, x, t)(\rho(x, t) - \rho^N(x, t)) \, dx \right|$, and thus the expectation is $O(\frac{1}{\sqrt{N \Delta t}})$.

After further accounting for the error from linear regression, Theorem [4] provides the final error estimation for this method.

**Theorem 1.** Let $\hat{\mathbf{c}} = \mathbf{A}^\dagger \mathbf{b} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}$ be the learned model coefficients and $\mathbf{c}^*$ the true model coefficients. There exists $C$ independent of $N$ and $\Delta t$ such that

$$\mathbb{E} \left[ \| \hat{\mathbf{c}} - \mathbf{c}^* \|_\infty \right] \leq C(\frac{1}{\sqrt{N \Delta t}} + \Delta t^\alpha).$$

**Proof.** Using that $\mathbf{c}^*$ and $\mathbf{D}^*$ are in the span of $\mathbb{C}^d_{p, \gamma}$, we have that

$$\mathbf{b}_k = \mathbf{D}_t^L \int_{\mathbb{R}^d} \psi_k(x) \rho^N(x, t) \, dx = \mathcal{R}_\Delta t(\rho^N(x, t), \psi_k(x)) + \mathbf{A}_k^T \mathbf{c}^* := \mathbf{R}_k + \mathbf{A}_k^T \mathbf{c}^*,$$

where $\mathbf{A}_k^T$ is the $k$th row of $\mathbf{A}$. From the previous result, we have

$$\mathbb{E} [\| \mathbf{R}_k \|] \leq \mathbb{E} [\| \mathbf{E}_2(p(x, t), \psi_k(x)) \|] + \mathbb{E} [\| \mathbf{E}_1(p(x, t), \psi_k(x)) \|] + \mathbb{E} [\| \mathcal{R}(p(x, t), \psi_k(x)) \|]$$

$$\leq C' \left( \Delta t^\alpha + \frac{1}{\sqrt{N \Delta t}} \right),$$

and $C'$ is independent of $N$ and $\Delta t$. Based on the previously derived results that $\mathbf{A}$ has full column rank, it holds that $\hat{\mathbf{c}} = \mathbf{A}^\dagger \mathbf{b} = \mathbf{A}^\dagger \mathbf{R} + \mathbf{c}^*$, hence the result follows from the uniform bound on $\| \mathbf{A}^\dagger \|_\infty$:

$$\mathbb{E} \left[ \| \hat{\mathbf{c}} - \mathbf{c}^* \|_\infty \right] \leq \mathbb{E} [\| \mathbf{A}^\dagger \|_\infty] \| \mathbf{R} \|_\infty \leq C' \mathbb{E} [\| \mathbf{A}^\dagger \|_\infty] \left( \Delta t^\alpha + \frac{1}{\sqrt{N \Delta t}} \right) \leq 2C' C_2 \left( \Delta t^\alpha + \frac{1}{\sqrt{N \Delta t}} \right).$$

For fixed sample number $N$, the observation time interval $\Delta t$ is optimally chosen as $N^{-\frac{1}{2\alpha+1}}$ so that the error is $O(\frac{1}{\sqrt{N \Delta t}})$. 

33
B Computational complexity analysis

In Weak Collocation Regression method, the explicit form of Gaussian function taken as test function and its first and second derivatives can be calculated in advance.

$$\phi(x, p, \gamma) = \prod_{i=1}^{d} \frac{1}{\gamma_i \sqrt{2\pi}} e^{-\frac{1}{2} \left( \frac{x_i - p_i}{\gamma_i} \right)^2},$$ \hspace{1cm} (53)

$$\frac{\partial}{\partial x_i} \phi(x, p, \gamma) = -\phi(x, p, \gamma) \cdot \frac{x_i - p_i}{\gamma_i^2}$$ \hspace{1cm} (54)

$$\frac{\partial^2}{\partial x_i \partial x_j} \phi(x, p, \gamma) = \begin{cases} \phi(x, p, \gamma) \left( \frac{(x_i - p_i)^2}{\gamma_i^2} - \frac{1}{\gamma_i^2} \right) & i = j \\ \phi(x, p, \gamma) \cdot \frac{x_i - p_i}{\gamma_i^2} \cdot \frac{x_j - p_j}{\gamma_j^2} & i \neq j \end{cases}$$ \hspace{1cm} (55)

Therefore, the computational complexity of $\phi(x, p, \gamma)$ is $\mathcal{O}(LNd)$, where $L, N, d$ stands for number of snapshots, sample number and dimension respectively. Thanks to the object-oriented characteristics of python, it is not necessary to recompute $\phi(x, p, \gamma)$ in (54)(55).

On computing known coefficient matrix $B(X)$ and column vector $y(X)$, there are $\mathcal{O}(LNMd)$ and $\mathcal{O}(LM)$ computational cost respectively, for we expand the diagonal of the diffusion matrix with a fixed number of basis $b$. Here $M$ stands for the number of the Gaussian functions. When applying linear multistep method, we obtain the linear system $(A, \hat{y})$ via $\mathcal{O}(L)$ additional computational cost.

On computing linear regression and sparse identification, `np.linalg.lstsq` finishes the calculation with QR factorization. The computational complexity of QR factorization is $\mathcal{O}(b^2LM)$, which takes little time because $b$ is a constant, and it is already covered by $\mathcal{O}(LNMd)$ in computing coefficient matrix. A large number of experiments show that linear regression and sparse identification takes a very low proportion of the total execution time, see Table 11. Therefore, the choice of computing matrix inverse has little effect.

| Time(s) | 1d cubic | 1d quintic | 2d sombrero | 3d cubic | 5d cubic | 10d cubic |
|---------|----------|-----------|-------------|----------|----------|----------|
| STRidge | 0.0004   | 0.001     | 0.132       | 0.0006   | 0.003    | 0.016    |
| Total   | 0.0177   | 0.541     | 28.9        | 3.84     | 66.5     | 213.5    |
| STRidge/Total | 2.26% | 0.18% | 0.46% | 0.02% | 0.005% | 0.008% |

Table 11: STRidge and the total program time under different examples

In conclusion, the computational complexity of Weak Collocation Regression method is $\mathcal{O}(LNMd)$ under condition of fixed the step of linear multistep method. It can be found that the computational complexity of WCR does not increase exponentially with the problem dimension.
C 10-dimensional coupled system

To illustrate our performance of WCR method in the high-dimensional coupled system, we consider a 10 dimensional aggregate data generated by the true model

\[ dX_t = \mu(X_t)dt + \sigma dW_t, \quad X_t \in \mathbb{R}^d \]

\[ \mu(x) = \begin{pmatrix} x_1 - x_3 \\ x_2 - x_3 \\ \vdots \\ x_d - x_3 \end{pmatrix}, \quad \sigma = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix} \]

with \( d = 10 \) and 100000 samples using Euler-Maruyama method. Several time snapshots of the data at time \( t = 0, 0.1, 0.2, 0.3, \ldots, 1 \) are used as the experiment data.

In the experiment, the forth order basis is used to approximate the drift term in each dimension as \( \mu_i = \theta_0 + \theta_1 x_i + \theta_2 x_i^2 + \theta_3 x_i^3 \), and the diagonal and 1-subdiagonal in diffusion term is approximated by a tunable parameter \( \sigma_i \), making totally \( 6d - 1 \) tunable parameters to be learned. Milne method is used for the temporal derivatives. 2000 and 10000 Gaussian functions were employed respectively, with a variance of 1 and mean values obtained by sampling from the region of the data using the Latin hypercube sampling method.

It is worth noting that the diffusion term has coupling, which means that the matrix \( D = \frac{1}{2} \sigma \sigma^T \) we learn may not necessarily be a symmetric matrix. If we denote \( \hat{D} = \frac{1}{2}(D + D^T) \), then \( \hat{D} \) is a symmetrical matrix. By performing a Cholesky decomposition on \( \hat{D} \), the lower triangular matrix obtained is the output diffusion term \( \sigma \). Table 12 presents the results of the calculation, which indicate that using 10,000 Gaussian functions, our method achieves a maximum relative error accuracy of 6.36% within 30 minutes. The results here demonstrate that the WCR method is effective in addressing high-dimensional problems with coupling.

| Gaussian | MRE in drift | MRE in diffusion | MRE | Time(min) |
|----------|-------------|-----------------|-----|-----------|
| \( m = 2000 \) | 11.8% | 6.22% | 11.8% | 5 |
| \( m = 10000 \) | 6.36% | 4.35% | 6.36% | 30 |

Table 12: The results of 10-dimensional problem with coupled diffusion term. Reveal the unknown dynamics with \( m = 2000, 10000 \) Gaussian functions, where there are 100000 samples each snapshots.

D STRidge Algorithm

Here we provide the details of the sequential thresholded ridge regression (STRidge) algorithm. In the STRidge method, each linear regression step retains the variables that were not sparsified in the previous regression. And if the original linear equation has \( n \) unknowns, the sparse regression operation is performed for a maximum of \( n \) iterations. STRidge will terminate directly if either of the following two conditions is met: 1) After a regression step, no additional variables are removed compared to the previous regression; 2) All variables have been removed. For further details of the STRidge algorithm, please refer to Algorithm 2.
Algorithm 2: STRidge Algorithm for Solving Linear System $Ax = b$

**Result**: Vector $x \in \mathbb{R}^n$ s.t. $Ax = b$

**Input**: Coefficient matrix $A \in \mathbb{R}^{m \times n}$, vector $b \in \mathbb{R}^m$, regular terms $\lambda > 0$, threshold $\eta > 0$

1 Compute $x$ by ridge regression $x = \arg \min_w \|Aw - b\|^2 + \lambda \|w\|^2$, set $p = n$;

2 while True do

3 Select index set $S^+ = \{x > \eta\}, S^- = \{x \leq \eta\}$;

4 if $\text{card}\{S^+\} = p$ then

5 | break

6 else

7 | $p = \text{card}\{S^+\}$

8 end

9 if $\text{card}\{S^+\} = 0$ then

10 | break

11 end

12 $x[S^-] = 0, x[S^+] = \arg \min_w \|A[:, S^+]w - b[:, S^+]\|^2 + \lambda \|w\|^2$

13 end

14 if $S^+ \neq \emptyset$ then

15 | $x[S^+] = \arg \min_w \|A[:, S^+]w - b[:, S^+]\|^2$

16 end