Dynamical system response under Gaussian and Poisson white noises solved by deep neural network method with adaptive task decomposition and progressive learning strategy

Wantao Jia [1,2] Xiaotong Feng [1,2] Yifan Zhao,[1,2] and Wanrong Zan [3,*]

[1]*School of Mathematics and Statistics, Northwestern Polytechnical University, Xi'an 710072, China*

[2]*MOE Key Laboratory for Complexity Science in Aerospace, Northwestern Polytechnical University, Xi'an 710072, China*

[3]*School of Mathematics, Northwest University, Xi'an 710127, China*

(Received 6 January 2025; accepted 31 March 2025; published 28 April 2025)

The forward Kolmogorov equation corresponding to a system under the combined excitation of Gaussian and Poisson white noises is an integrodifferential equation (IDE). In our recent study, we introduced GL-PINNs, which integrates the Gauss-Legendre (GL) quadrature with the physics-informed neural networks (PINNs) framework for solving time-dependent IDEs. However, we observed that in scenarios such as insufficient learning of initial conditions or dynamical systems with strong temporal dependencies, GL-PINNs produced inaccurate solutions despite achieving low training loss values. This issue primarily stems from the GL-PINNs framework's failure to account for temporal causality. To address this limitation, we develop a deep neural network method called ATD-GLPINNs, which integrates adaptive task decomposition and progressive learning strategy. This approach decomposes the complex task along the time axis into an initial subtask and several extra tasks, which enable progressive learning through adaptive adjustment of task parameters. As an extension of the GL-PINNs, this algorithm adheres to temporal causality by prioritizing the training of early subtasks and dynamically allocating additional computational resources to subsequent ones. Numerical experiments demonstrate that our suggested method converges with significantly fewer training epochs compared to GL-PINNs, making it not only more efficient and robust, but also capable of reducing computational costs while improving prediction accuracy.

DOI: 10.1103/PhysRevE.111.045309

I. INTRODUCTION

The forward Kolmogorov equation describes the time evolution of the probability density function (PDF) for the solution of a stochastic differential equation (SDE), which has gained broad applications in the field of mathematics [1–4], engineering [5], physics [6,7], chemistry [8], optics [9], and economy [10]. However, finding an exact stationary solution of such an equation is generally challenging. Currently, traditional approaches [11] like the stochastic averaging method [12–14], finite difference method [15], finite element method [16], generalized cell mapping method [17], path integral method (PI) [18,19], and Galerkin method [20,21] are primarily used to solve it. However, these methods often encounter challenges such as grid discretization, high computational complexity, and the curse of dimensionality. Consequently, new approaches are urgently needed.

In recent years, deep neural network (DNN) methods have played a significant role in solving the forward Kolmogorov equation, with particular attention given to physics-informed neural networks (PINNs) [22]. In the classical PINNs framework, a fully connected feed-forward neural network is employed as the core of the surrogate model to approximate the solution of partial differential equations (PDEs) [23–25]. As a meshless method, PINNs leverages automatic differentiation [26] within neural networks to compute partial derivatives and directly solve PDEs by embedding physical constraints into the loss function. This approach provides an effective solution for analyzing dynamic system responses and identifying parameters under stochastic excitations. For solving the forward Kolmogorov equation, Xu [27] utilized the DL-FP algorithm to analyze the response of time-independent dynamical systems under the excitation of Gaussian white noise. Jiang proposed the trapz-PINNs [28] to solve the time-dependent forward Kolmogorov equation driven by α-stable Lévy noise. For the parameter identification problem, Zhang [29] introduced improved PINNs to solve the critical parameter identification problem in aircraft systems under the excitation of Gaussian white noise. Chen [30] developed a general framework based on PINNs for solving inverse problem involving discrete particle observations, encompassing Gaussian noise, Lévy noise, or a combination of both.

The presence of Poisson white noise makes the forward Kolmogorov equation corresponding to the dynamical system into an integrodifferential equation (IDE), which contributes to the difficulty of solving such an equation. Currently, the study of system response under Poisson white noise excitation [31] using DNNs method has also garnered considerable attention. In our latest research [32], to address the transient response of systems excited by combined Gaussian and Poisson white noises, we integrated PINNs with the Gaussian-Legendre (GL) quadrature formula, referring to this approach as GL-PINNs. Nevertheless, during the computational process, we observed that GL-PINNs face certain challenges in simulating complex dynamical system behaviors, such as

*Contact author: wanrongzan@nwu.edu.cn

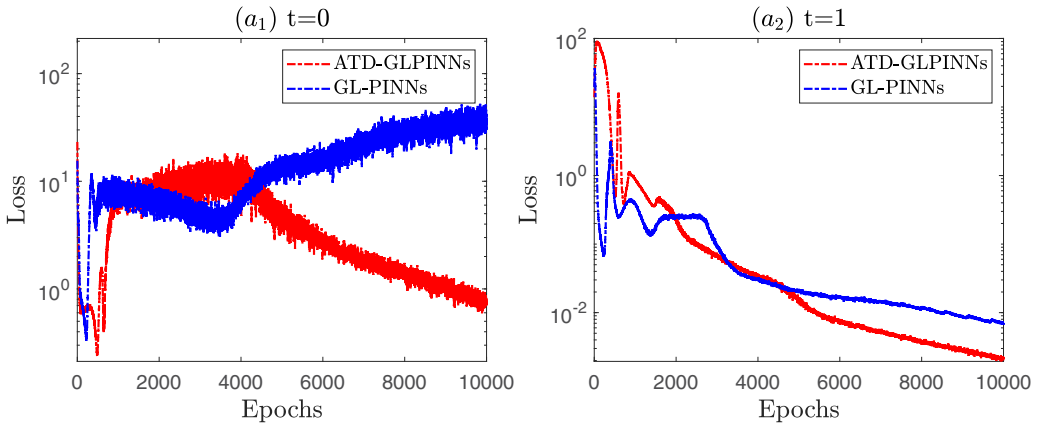

FIG. 1. The loss history of the residual term for ATD-GLPINNs (red curve) and GL-PINNs (blue curve) at $t = 0$ and $t = 1$.

insufficient learning of initial conditions and difficulties in handling systems with strong temporal dependencies. These issues suggest that GL-PINNs still have room for improvement when dealing with complex dynamical systems that exhibit significant time-dependent characteristics. In Fig. 1, the residual loss history for Example I is presented. In the left panel, the residual loss of GL-PINNs (blue curve) gradually increases in the later stages of training and eventually fluctuates within the range of 10^1 to 10^2. This highlights the difficulty of the GL-PINNs in effectively learning the initial conditions. In the right panel, the loss of GL-PINNs drops rapidly in the beginning but remains at a higher value in later stages. This suggests that GL-PINNs prioritizes learning at a later time point ($t = 1$) before properly capturing the solution at an earlier time ($t = 0$). This behavior clearly demonstrates that GL-PINNs do not adhere to temporal causality.

To address time-dependent problems, researchers have drawn inspiration from domain decomposition techniques and successfully applied them for the improvement of the PINNs algorithm (such as parareal PINN (PPINN) [33], extended PINNs(XPINNs) [34], and backward compatible PINN (bc-PINN) [35]). Building upon the concepts of adaptive task decomposition and progressive learning strategy [36], we extend the GL-PINNs algorithm and introduce our enhanced approach, termed ATD-GLPINNs, which is designed to handle time-dependent forward Kolmogorov equations. The ATD-GLPINNs divides the full-time domain into nonoverlapping sub-intervals. By adaptively adjusting the task parameters, we control the training process transitions gradually from a small time domain near the initial conditions to the full-time domain. As illustrated in Fig. 1, the left panel illustrates that ATD-GLPINNs initially exhibits a higher loss value, which gradually decreases and eventually fluctuates within the range of 10^0 to 10^1. In the right panel, ATD-GLPINNs converge faster than GL-PINNs and achieve a lower final loss history. Evidently, ATD-GLPINNs prioritizes the optimization of the solution at $t = 0$ and progressively optimizes the solutions at subsequent time instants along the time axis. The experimental results demonstrate that our progressive learning strategy not only preserves the temporal causality of the problem but also significantly enhances computational efficiency through synchronized progression and dynamic resource allocation.

The rest of this paper is organized as follows. Specifically, the system with combined Gaussian and Poisson white noises is introduced in Sec. II. In Sec. III, we review the GL-PINNs framework for solving IDE and display the ATD-GLPINNs scheme with task decomposition and adaptive progressive training strategy. In Sec. IV, several numerical results are presented to demonstrate the effectiveness of our method. Conclusions and perspectives are summarized in Sec. V.

II. SYSTEM WITH COMBINED GAUSSIAN AND POISSON WHITE NOISES

Let $\mathbf{X}(t) = [X_i(t)]_{n_x\times 1}$ denotes the vector state process that satisfies the following multidimensional Itô SDE [37]

$$\begin{aligned} \mathrm{d}\mathbf{X}(t) &= \mathbf{f}(\mathbf{X}(t), t)\mathrm{d}t + \mathbf{g}(\mathbf{X}(t), t)\mathrm{d}\mathbf{W}(t) + \mathbf{h}(\mathbf{X}(t), t, \mathbf{R})\mathrm{d}\mathbf{P}(t), \\ \mathbf{X}(0) &= \mathbf{X}_0, \end{aligned} \tag{1}$$

in which the coefficient functions are specified in terms of their dimensionality as follows

$$\begin{aligned} \mathbf{f}(\mathbf{X}(t), t) &= [f_i(\mathbf{X}(t), t)]_{n_x\times 1}, \\ \mathbf{g}(\mathbf{X}(t), t) &= [g_{i,j}(\mathbf{X}(t), t)]_{n_x\times n_w}, \\ \mathbf{h}(\mathbf{X}(t), t, \mathbf{R}) &= [h_{i,j}(\mathbf{X}(t), t, R_j)]_{n_x\times n_p}. \end{aligned} \tag{2}$$

$\mathbf{W}(t) = [W_i(t)]_{n_w\times 1}$ represents an n_w-dimensional standard Wiener process. $\mathrm{d}\mathbf{W}(\mathrm{t}) = [\mathrm{dW_i(t)}]_{\mathrm{n_w}\times 1}$ denotes the increment of Wiener process. The covariance between $\mathrm{dW_i(t)}$ and $\mathrm{dW_j(t)}$ is given by

$$\mathrm{Cov}[\mathrm{d}W_i(t), \mathrm{d}W_j(t)] = \rho_{i,j}\mathrm{d}t, \tag{3}$$

where $\rho_{i,j}$ is the correlation coefficient between components i and j. $\mathbf{P}(t) = [P_i(t)]_{n_p\times 1}$ is an n_p-dimensional independent Poisson process. $\mathbf{R} = [R_i]_{n_p\times 1}$ and R_i represents the i-th jump amplitude with probability density $\phi_{R_i}(r_i)$. The coefficient $\mathbf{h}(\mathbf{X}(t), t, \mathbf{R})$ represents the jump amplitude operator that depends on the mark $\mathbf{R}$.

The forward Kolmogorov equation [37] satisfied by the multidimensional transient probability density $\phi_X(\mathbf{x}, t)$ can be derived as

$$\frac{\partial}{\partial t}\phi_X(\mathbf{x}, t) = \mathscr{L}(\phi_X(\mathbf{x}, t)), \tag{4}$$

in which

$$\mathscr{L}(\phi_X(\mathbf{x},t)) = \frac{1}{2}\nabla_{\mathbf{x}}\left[\nabla_{\mathbf{x}}^T : [gV'g^T\phi_X]\right](\mathbf{x},t) - \nabla_{\mathbf{x}}^T[\mathbf{f}\phi_X](\mathbf{x},t) - \sum_{j=1}^{n_p}\lambda_j\phi_X(\mathbf{x},t) + \sum_{j=1}^{n_p}\lambda_j\int_{\mathcal{R}}\phi_X(\mathbf{x}-\eta_j(\mathbf{x},t,r_j),t)\left|1-\frac{\partial(\eta_j(\mathbf{x},t,r_j))}{\partial(\mathbf{x})}\right| \times \phi_{R_j}(r_j)\mathrm{d}r_j, \tag{5}$$

the symbol $\nabla_{\mathbf{x}}$ represents the gradient operator with respect to the state variable $\mathbf{x}$, commonly used for calculating partial derivatives. $\nabla_{\mathbf{x}}^T$ denotes the transpose of the state space gradient. V' is a correlation matrix defined as $V' \equiv [\rho_{i,j}]_{n_w\times n_w}$. $A : B$ is the double-dot product of two $n \times n$ matrices defined as follows:

$$\mathbf{A} : \mathbf{B} \equiv \sum_{i=1}^{n}\sum_{j=1}^{n}\mathbf{A}_{i,j}\mathbf{B}_{i,j} = \mathrm{Trace}[\mathbf{A}\mathbf{B}^T], \tag{6}$$

in which the trace operation Trace $[\mathbf{A}\mathbf{B}^T]$ refers to the sum of the diagonal elements of the matrix product of $\mathbf{A}$ and the transpose of $\mathbf{B}$.

In Eq. (5), the function $\eta_j(\mathbf{x},t,r_j)$ is the inverse transformation function. The transformation from backward to forward for the j'-th jump process can be described as follows:

$$\mathbf{x} - \mathbf{x}_0 = \boldsymbol{\eta}_{j'}(\mathbf{x},t,r_{j'}) = \widehat{\boldsymbol{h}}_{j'}(\mathbf{x}_0,t,r_{j'}), \tag{7}$$

where $\mathbf{x_0}$ represents the system's state before jump, and $\hat{h}_j(\mathbf{x},t,r_j) \equiv [h_{i,j}(\mathbf{x},t,r_j)]_{n_x\times 1}$ is the j-th jump amplitude vector corresponding to the j-th Poisson process for $j = 1 : n_p$ given in Eq. (1). It's associated Jacobian matrix is computed by

$$\frac{\partial(\eta_{j'}(\mathbf{x},t,r_{j'}))}{\partial(\mathbf{x})} = \mathrm{Det}\left[\left[\frac{\partial\eta_{j',i}(\mathbf{x},t,r_{j'})}{\partial x_j}\right]_{n_x\times n_x}\right] = \mathrm{Det}\left[\left(\nabla_{\mathbf{x}}\left[\eta_{j'}^T\right]\right)^T\right]. \tag{8}$$

III. METHODOLOGY

A. GL-PINNs algorithm for solving IDE

The GL-PINNs method, as introduced by [32], provides a more effective approach for solving the forward Kolmogorov equation corresponding to dynamical systems under the combined excitation of Gaussian and Poisson white noises. To facilitate the numerical computation, the infinite integral in Eq. (5) is approximated by a finite integral over a limited interval $[r_{lb}, r_{ub}]$, which can be discretized as follows:

$$\sum_{j=1}^{n_p}\lambda_j\int_{r_{lb}}^{r_{ub}}\phi_X(\mathbf{x}-\eta_j(\mathbf{x},t,r_j),t)\left|1-\frac{\partial(\eta_j(\mathbf{x},t,r_j))}{\partial(\mathbf{x})}\right| \times \phi_{R_j}(r_j)\mathrm{d}r_j \approx \sum_{j=1}^{n_p}\lambda_j\left[\boldsymbol{\Phi}_j \odot \left|1-\frac{\partial(\eta_j(\mathbf{x},t,r_j))}{\partial(\mathbf{x})}\right| \odot \boldsymbol{\Phi}_{R_j}\right]^T \times \boldsymbol{\omega}_j, \tag{9}$$

where $\odot$ denotes the Hadamard product. The specific expression of matrix in Eq. (9) is provided in Appendix.

In GL-PINNs framework, the training points include the following four parts: the initial training points $T_u = \{x_{IC}^j, 0\}_{j=1}^{N_u}$, the boundary training points $T_b = \{x_{BC}^j, t_{BC}^j\}_{j=1}^{N_b}$, the residual points $T_f = \{x_f^j, t_f^j\}_{j=1}^{N_f}$, and the normalized training points $T_n = \{x_n^j, t_n^j\}_{j=1}^{N_n}$. We assume that the predicted solutions are the output of the neural networks, represented as $\tilde{\phi}_X(x,t) = \phi_{NN}(x,t;\theta)$, which approximates the reference solutions $\phi_X(x,t)$. The parameter θ in this neural networks denotes the weights and bias. Training GL-PINNs algorithm aims to minimize the following loss function (10) and find the optimal parameters θ,

$$\mathrm{MSE}(\theta) = \omega_u\mathrm{MSE}_u + \omega_b\mathrm{MSE}_b + \omega_f\mathrm{MSE}_f + \omega_n\mathrm{MSE}_n, \tag{10}$$

where

$$\mathrm{MSE}_u = \frac{1}{N_u}\sum_{j=1}^{N_u}\left|\tilde{\phi}_X\left(\mathbf{x}_{IC}^j,0\right) - \phi_X\left(\mathbf{x}_{IC}^j,0\right)\right|^2, \left\{\mathbf{x}_{IC}^j,0\right\} \in T_{\mathrm{u}}, \tag{11}$$

$$\mathrm{MSE}_b = \frac{1}{N_b}\sum_{j=1}^{N_b}\left|\tilde{\phi}_X\left(\mathbf{x}_{BC}^j,t_{BC}^j,\right) - \phi_X\left(\mathbf{x}_{BC}^j,t_{BC}^j\right)\right|^2, \left\{\mathbf{x}_{BC}^j,t_{BC}^j\right\} \in T_{\mathrm{b}}, \tag{12}$$

$$\mathrm{MSE}_f = \frac{1}{N_f}\sum_{j=1}^{N_f}\left|\frac{\partial\tilde{\phi}_X\left(\mathbf{x}_f^j,t_f^j\right)}{\partial t} - \mathcal{L}\left(\tilde{\phi}_X\left(\mathbf{x}_f^j,t_f^j\right)\right)\right|^2, \left\{\mathbf{x}_f^j,t_f^j\right\} \in T_{\mathrm{f}}, \tag{13}$$

$$\mathrm{MSE}_n = \left|\int_{\Omega}\tilde{\phi}_X(\mathbf{x},t)\mathrm{d}\mathbf{x} - 1.0\right|^2, \tag{14}$$

where MSE_u, MSE_b, MSE_f, and MSE_n are the mean square errors(MSE) from initial conditions(IC), boundary conditions(BC), governing equations, and normalizing conditions, respectively. ω_u, ω_b, ω_f, and ω_n are the corresponding weights. The integral in MSE_n is computed by Monte Carlo (MC) integration method.

B. ATD-GLPINNs

Owing to the convergence difficulties of GL-PINNs when solving the response of dynamical systems with strong temporal dependencies or insufficient learning of initial conditions, we extended GL-PINNs to ATD-GLPINNs, which is inspired by the adaptive task decomposition strategy in [36]. For simply describing this strategy, the task decomposition is conducted evenly along the temporal axis. We decompose the full computational domain into M subtasks labeled as

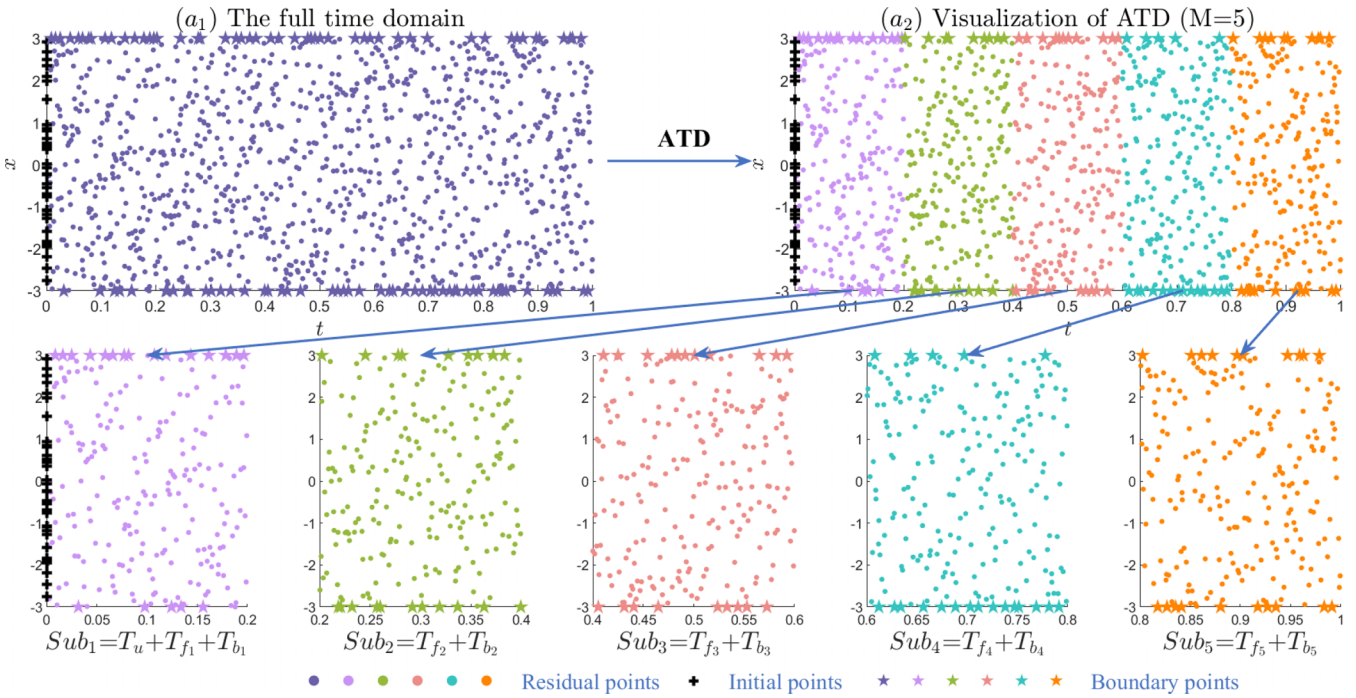

FIG. 2. The schematic diagram of the adaptive task decomposition (ATD) mechanism.

Sub_1, Sub_2,..., Sub_M, where Sub_1 denotes the initial subtask, and Sub_i $(i = 2, \cdots, M)$ denotes the extra tasks. For convenience, we use Sub_i to denote each subtask along with its corresponding set of training points. Likewise, we evenly partition the collection of residual points, boundary points, and normalizing points into M sub-sets, where the training point sets corresponding to the i-th subtask are denoted as T_{f_i}, T_{b_i} and T_{n_i}, respectively. The schematic diagram of adaptive task decomposition mechanism for $M = 5$ is shown in Fig. 2.

Figure 3 illustrates the framework of ATD-GLPINNs, in which the corresponding loss function is divided into an initial subtask and multiple extra tasks, as illustrated below

$$\mathrm{MSE}(\theta)_{\mathrm{ATD}} = \omega_u \mathrm{MSE}_u + \frac{1}{\sum_{i=1}^{M}\omega_i}\sum_{i=1}^{M}\omega_i(\mathrm{MSE}_{f_i} + \omega_b \mathrm{MSE}_{b_i} + \omega_n \mathrm{MSE}_{n_i}), \quad i = 1, 2, \cdots, M, \tag{15}$$

in which MSE_{b_i}, MSE_{f_i} and MSE_{n_i} correspond to the loss function for the boundary conditions, governing equations, and normalization conditions of the i-th subtask, respectively. During each training epoch, the update gradient is calculated as a weighted sum of the gradients from each individual partitioned loss term. The task parameter corresponding to the i-th subtask, denoted by ω_i, which governs the dynamic allocation of computational resources for the extra tasks throughout the training process

$$\omega_i = \exp\left(-\mu\left(\omega_u \mathrm{MSE}_u + \sum_{j=1}^{i-1}(\mathrm{MSE}_{f_j} + \omega_b \mathrm{MSE}_{b_j} + \omega_n \mathrm{MSE}_{n_j})\right)\right), \quad i = 2, 3, \cdots, M, \tag{16}$$

where ω_1 stands for the weight assigned to the mean square error of the initial subtask, with its default value set to 1. The

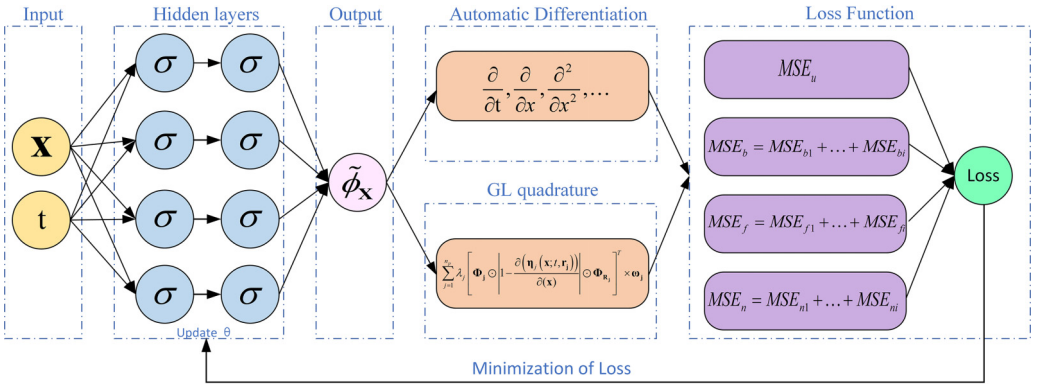

FIG. 3. Basic structure of ATD-GLPINNs to solve time-dependent problems.

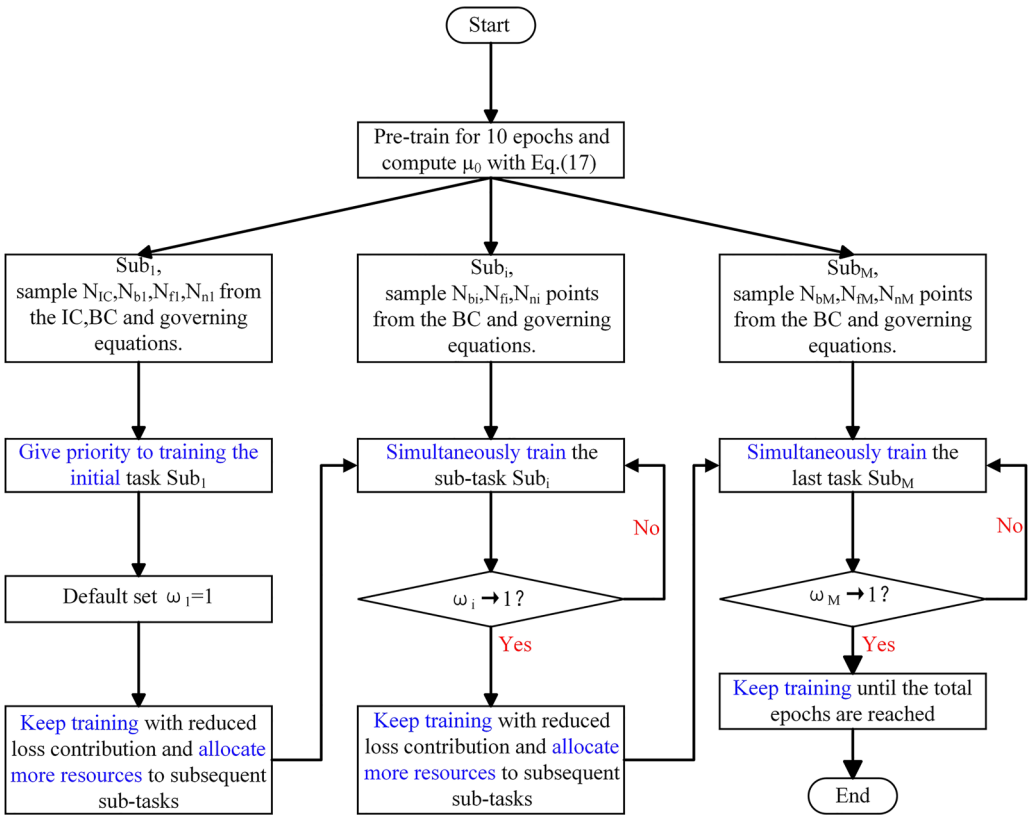

FIG. 4. Flowchart of the ATD-GLPINNs.

parameter μ determines the speed of task marching, where larger values of μ imply higher computing resource requirements. To select an appropriate value for μ, we pretrain for 10 epochs to calculate the average total loss $LOSS_{\text{total}}$, which is then used to determine the initial value of μ, denoted as μ_0

$$\mu_0 = \frac{1}{10^{\left(\text{ceil}\left(\log_{10}^{LOSS_{\text{total}}}\right)+1\right)}}, \tag{17}$$

in which the ceil denotes ceiling rounding function. The values of the μ can range from 1 to 10,000 times the initial value μ_0. It is worth noting that a larger μ significantly accelerates subtask progression, speeds up algorithm convergence, and optimizes simpler tasks more quickly. However, this setting may lead to insufficient learning for more challenging tasks. Conversely, a smaller μ ensures a more balanced training process, which allows all subtasks to be sufficiently optimized, but at the cost of a slower algorithm convergence rate. In order to achieve proper tuning between training efficiency and task equilibrium, we recommend incrementally increasing this parameter to progressively enhance the task demands, enabling ATD-GLPINNs to accurately predict the system's response.

In our work, the training process of ATD-GLPINNs algorithm involves multitask progressive learning strategy, where the task weights ω_i, $(i = 2, \cdots, M)$ are dynamically adjusted based on the total loss of the preceding subtask. Once the preceding subtask is sufficiently trained, the decline in its loss history will cause the task parameter of the current subtask to gradually approach one. It is worth mentioning that the preceding subtask continues to be trained, even though its optimization intensity has decreased and its contribution to the total loss has diminished. This approach ensures that the training of each subtask is influenced by the training state of the preceding subtask, prioritizing earlier stages before progressively allocating more resources to subsequent subtasks. Consequently, this method adheres to temporal causality. When all the task parameters ω_i take values close to 1, it indicates that the entire training process sufficiently covers the full task domain, resulting in a more comprehensive and balanced learning of the model. At this stage, the training process continues until the completion of all predefined epochs. The flowchart of ATD-GLPINNs approach is given in Fig. 4.

IV. RESULTS

Throughout this part, we systematically compare the performance of ATD-GLPINNs with the GL-PINNs algorithm. The accuracy of the predicted solutions $\tilde{\phi}_X(x,t)$ generated by the DNN methods is evaluated using the following error

$$\|\tilde{\phi}_X(x,t) - \phi_X(x,t)\|_2^2 = \int_{-\infty}^{+\infty} \left(\tilde{\phi}_X(x,t) - \phi_X(x,t)\right)^2 \mathrm{d}x, \tag{18}$$

where $\phi_X(x,t)$ is defined as the reference solutions. In this study, as the exact solutions are unavailable, we use the MC simulations as the reference solutions.

In all numerical experiments, we consider a fully connected feed-forward neural network, which includes an input layer, an output layer, and 4 hidden layers with 20 neurons per layer. The numerical implementation of our algorithms

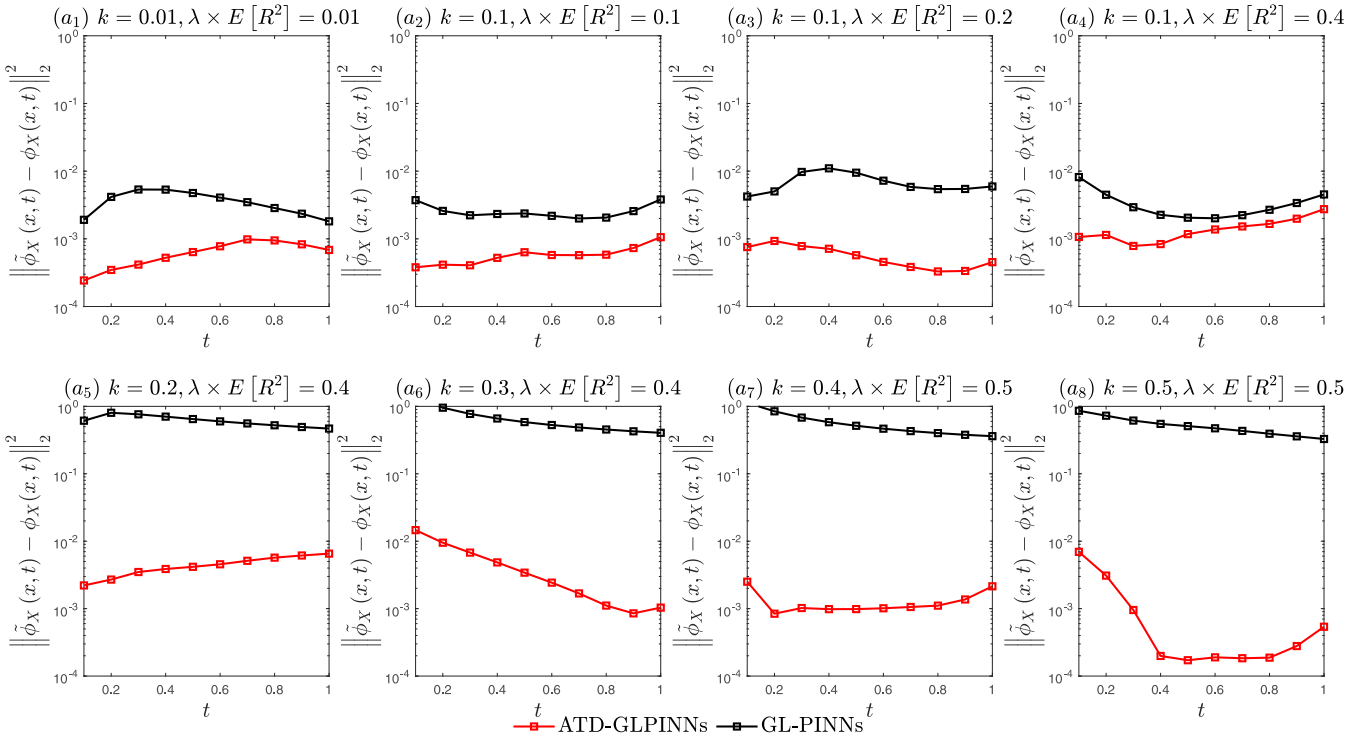

FIG. 5. The variations in errors across the entire time domain under different intensities of Gaussian and Poisson white noises. The k represents the intensity of Gaussian white noise, while $\lambda \times E[R^2]$ represents the intensity of Poisson white noise.

is based on TensorFlow. The activation function and optimizer for updating parameters adopted within this paper are the hyperbolic tangent function and the Adam optimizer. In this research, we present the results obtained at specific time instants or across the entire time domain. To evaluate the accuracy of our methods, we use the MC method with 100,000 trajectories as reference solutions. Our initial condition follows a normal distribution with the mean vector $\boldsymbol{\zeta}$ and the covariance matrix $\boldsymbol{\Sigma}$. The Gaussian and Poisson white noise considered in this paper are mutually independent. Unless otherwise stated, we consider the amplitude R of Poisson white noise follows a normal distribution. And we set $N_q = 40$ as the number of GL quadrature nodes to approximate the integral part, where the GL quadrature interval $N_{\text{sub}} = 10$ and specify four quadrature nodes in each sub-interval. The default value of $\mu = 100\mu_0 = 0.1$. The weights of the loss function are chosen: $\omega_u = 100$, $\omega_b = 1$, and $\omega_n = 1$. The notations of all examples and algorithm parameters are summarized in Table I for quick reference.

A. Example 1

As the first example, we study a nonlinear, one-dimensional case driven by the combination of Gaussian and Poisson white noises

$$\dot{X}(t) - (aX(t) + bX(t)^3) = W_G(t) + W_P(t), \tag{19}$$

the system Eq. (19) can be expressed as the following SDE

$$\mathrm{d}X(t) = (aX(t) + bX(t)^3)\mathrm{d}t + \sqrt{k}\mathrm{d}W(t) + R\mathrm{d}P(t), \tag{20}$$

the associated forward Kolmogorov equation can be rebuilt as

$$\begin{aligned}\frac{\partial}{\partial t}\phi_X(x,t) = &-(a + 3bx^2 + \lambda)\phi_X(x,t) \\ &- (ax + bx^3)\frac{\partial}{\partial x}\phi_X(x,t) + \frac{k}{2}\frac{\partial^2\phi_X(x,t)}{\partial x^2} \\ &+ \frac{\lambda}{\sqrt{2\pi\sigma_R^2}}\int_{-\infty}^{\infty}\phi_X(x-r,t)\exp\left(-\frac{r^2}{2\sigma_R^2}\right)\mathrm{d}r.\end{aligned} \tag{21}$$

We set $N_u = 100$, $N_b = 100$, $N_f = 1000$, $N_n = 2000$, and $M = 5$. Thus, with each subtask, the number of training points $N_{b_i} = 20$, $N_{f_i} = 200$, and $N_{n_i} = 400$ is assigned. The learning rate is fixed as 0.001. The computational domain is set as

TABLE I. The list of key symbols.

Notation	Stands for
i	The number of subtasks
N_f	The total number of residual points
N_u	The total number of initial training points
N_b	The total number of boundary training points
N_n	The total number of normalization points
N_q	The total number of GL quadrature nodes
N_{sub}	The number of GL quadrature intervals
λ	The mean arrival rate of Poisson white noise
k	The noise intensity of Gaussian white noise
W_G	The Gaussian white noise
W_P	The Poisson white noise

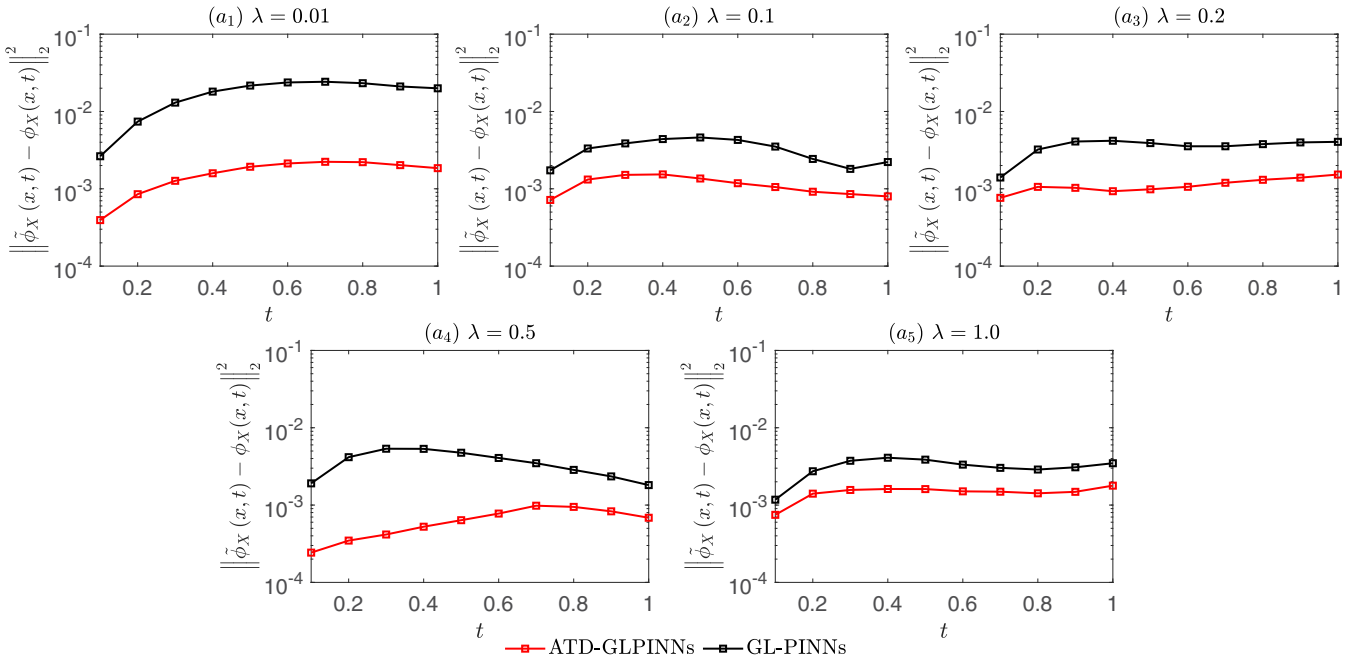

FIG. 6. The variations in errors across the entire time domain for different mean arrival rates λ.

$\Omega = [0, T] \times [x_{lb}, x_{ub}] = [0, 1] \times [-3, 3]$. Unless otherwise noted, we consider $a = -0.1$, $b = 0.5$. The initial condition values follow a normal distribution with the mean $\zeta = 0$ and the variance $\sigma^2 = 0.01$. The training process is conducted 5,000 epochs. More details about the hyperparameters and noise settings are provided in the relevant parts for each result.

Figure 5 meticulously illustrates the impact of different noise intensities on the effectiveness of our algorithm. The mean arrival rate λ of Poisson white noise is set as 0.5. The red solid line in the figure represents the errors of ATD-GLPINNs, while the black solid line corresponds to the error of GL-PINNs. The noise intensities used in the experiment are randomly selected from the range of 0.01 to 0.5. It can be seen that, even under high noise intensities, the error of the ATD-GLPINNs algorithm remains at least at the order of 10^{-3}. In contrast, the error of GL-PINNs shows considerable fluctuations, whereas ATD-GLPINNs remains stable. These observations imply that our suggested strategy significantly enhances the robustness of the ATD-GLPINNs.

The errors associated with different values of equation parameters a and b are presented in Table II across the entire time domain. We set the noise intensity to $k = 0.1$, $\lambda \times E[R^2] = 0.4$ and the mean arrival rate $\lambda = 0.01$. It is clear that the GL-PINNs still fails to converge or achieves sufficient accuracy after multiple attempts, whereas ATD-GLPINNs demonstrates superior performance with lower error and higher accuracy across various parameter settings. Notably, the error of ATD-GLPINNs can reach at least the order of 10^{-3}.

Figure 6 illustrates the effect of varying λ with fixed Poisson white noise intensity. The intensities of both Gaussian and Poisson white noises are set as 0.01. As λ increases from 0.01 to 1.0, the error of GL-PINNs (black solid line) remains generally higher, while ATD-GLPINNs (red solid line) maintains a relatively lower error. The results show that the ATD-GLPINNs method consistently exhibits lower errors across different mean arrival rates λ, significantly outperforming the GL-PINNs method.

Figure 7 examines the impact of different numbers of training epochs on the errors with $\lambda = 0.1$. The noise parameter settings are the same as in Fig. 6. As the number of epochs decreases, the errors of both algorithms increase significantly. However, ATD-GLPINNs still achieves lower error with fewer epochs. It is evident that combining adaptive task decomposition with the GL-PINNs algorithm can

TABLE II. The accuracy of the ATD-GLPINNs and GL-PINNs in different values of equation parameters a and b.

Parameter	Methods	$a = -0.5$	$a = -0.3$	$a = -0.1$	$a = 0.3$	$a = 0.5$
$b = -0.3$	ATD-GLPINNs	1.61×10^{-3}	6.10×10^{-4}	1.72×10^{-3}	8.96×10^{-4}	7.76×10^{-4}
	GL-PINNs	1.13×10^{-0}	1.78×10^{-2}	1.21×10^{-0}	2.46×10^{-1}	7.71×10^{-1}
$b = 0.3$	ATD-GLPINNs	1.32×10^{-3}	1.50×10^{-4}	5.82×10^{-3}	1.22×10^{-4}	2.09×10^{-3}
	GL-PINNs	1.49×10^{-1}	2.64×10^{-2}	1.07×10^{-0}	2.29×10^{-2}	4.91×10^{-1}
$b = 0.5$	ATD-GLPINNs	1.43×10^{-3}	5.73×10^{-4}	9.93×10^{-4}	5.41×10^{-4}	5.70×10^{-4}
	GL-PINNs	2.89×10^{-2}	2.36×10^{-2}	2.90×10^{-2}	1.68×10^{-2}	1.88×10^{-2}

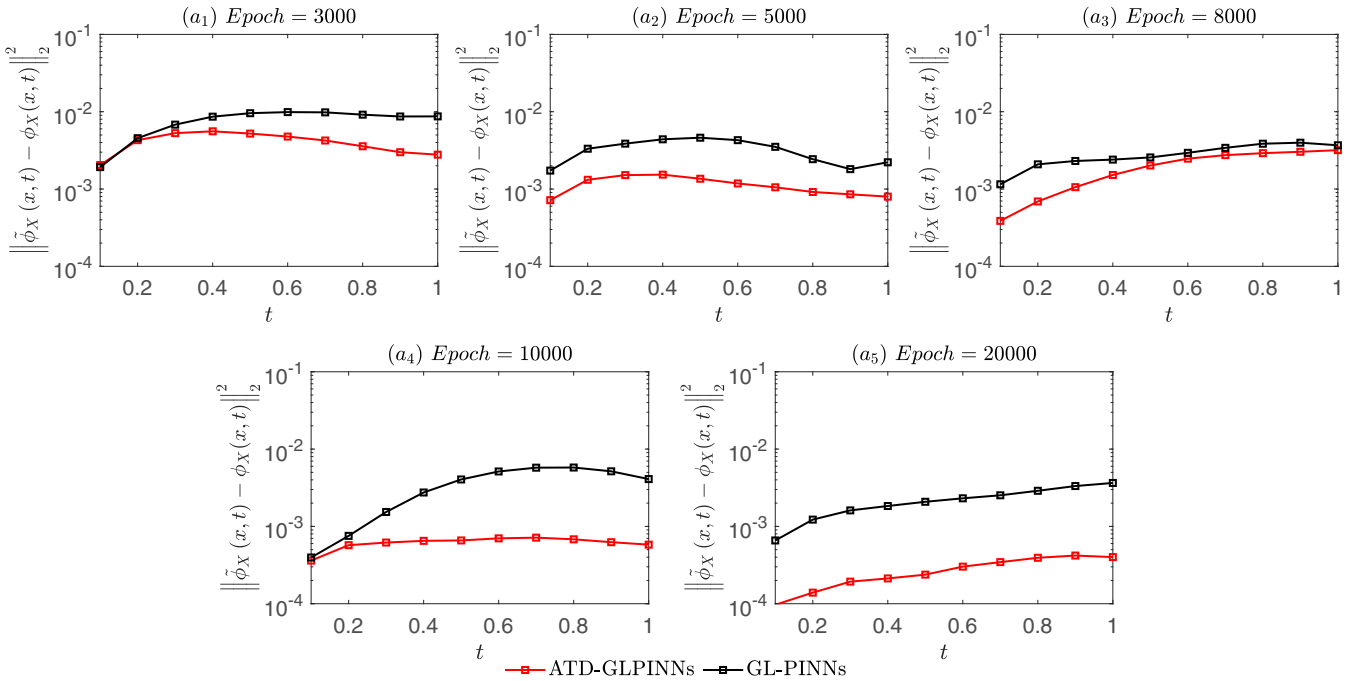

FIG. 7. The variations in errors across the entire time domain for different numbers of training epochs.

significantly reduce computational costs and enhance calculation accuracy.

Next, we demonstrate the algorithm's accuracy for initial conditions corresponding to four different values of variance σ^2 in Fig. 8, where the variance ranges from 0.005 to 0.1. The first row of the figure compares the predicted solutions of ATD-GLPINNs with the MC reference solutions, while the second row presents the comparative results of GL-PINNs. Here, the mean arrival rate λ of Poisson white noise is fixed to be 0.5. The intensity of Gaussian and Poisson

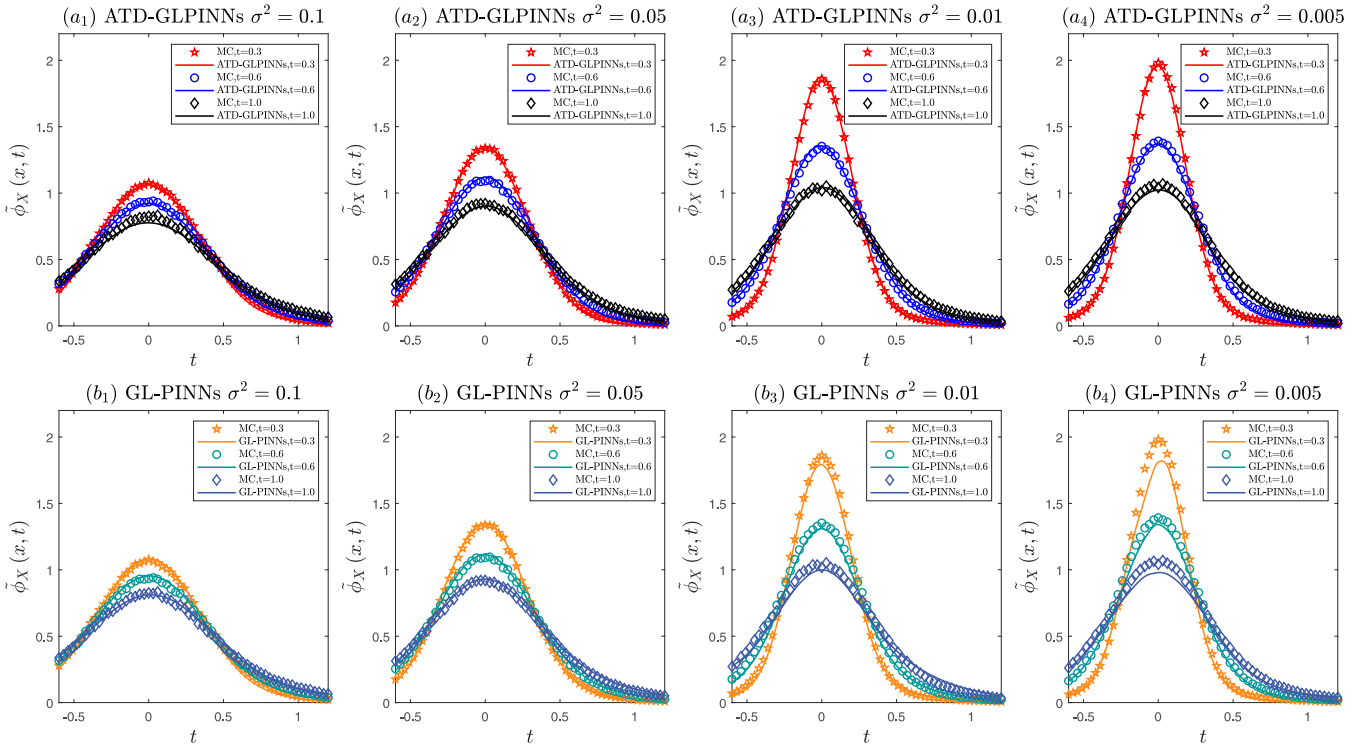

FIG. 8. The PDFs predicted by the ATD-GLPINNs (the first row) and GL-PINNs (the second row) algorithms correspond to the different initial conditions at $t = 0.3$s, 0.6s, and 1.0s. The colored lines are the predicted solutions from ATD-GLPINNs and GL-PINNs, while the colored symbols denote the MC reference solutions. The symbol σ^2 represents the variance value of the initial conditions.

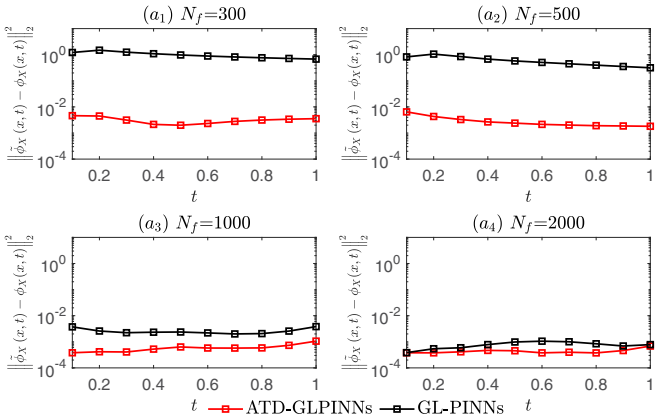

FIG. 9. The variations in errors across the entire time domain for different numbers of residual points N_f.

white noises are both set as 0.1. The results indicate that ATD-GLPINNs demonstrates superior stability and higher predictive accuracy across a range of initial settings, particularly in cases with sharp initial conditions. In contrast, GL-PINNs exhibits greater sensitivity to changes in the initial conditions.

Figure 9 displays the errors corresponding to the different values of N_f in the whole time domain, where N_f ranges from 300 to 2000. The noise parameter settings are the same as in Fig. 8. In all subplots, the prediction errors of the red solid line (ATD-GLPINNs) consistently remain lower than those of the black solid line (GL-PINNs). Moreover, as the number of residual points N_f increases, the error of GL-PINNs decreases but still remains higher than that of ATD-GLPINNs. The results demonstrate that ATD-GLPINNs achieves superior accuracy using fewer residual points, demonstrating its capacity to markedly reduce computational costs.

As described in Fig. 10, the impact of N_{sub} on algorithm accuracy is analyzed. We set the noise intensity to $k = 0.1$, $\lambda \times E[R^2] = 0.2$, and the mean arrival rate $\lambda = 0.2$. The colored lines represent the predicted solutions from ATD-GLPINNs in the first row and GL-PINNs in the second row, while the colored symbols indicate the MC reference solutions. In order to approximate the integral term, we partition the GL quadrature interval into 1, 5, 10, and 20 sub-intervals, with four quadrature nodes specified for each sub-interval. It is evident that the predicted profile of the ATD-GLPINNs algorithm closely matches the MC reference solutions. The training results remain satisfactory even with fewer GL quadrature nodes.

The sensitivity of the ATD-GLPINNs to varying μ values is further explored in Fig. 11, where the value of μ ranges from 0.001 to 10. The noise parameter settings are the same as in Fig. 10. Each subplot shows the error on the vertical axis and time t on the horizontal axis. The outcomes highlight that ATD-GLPINNs are sensitive to the parameter μ, and selecting an appropriate μ value significantly reduces the errors and enhances model performance. Unless otherwise specified, we choose $\mu = 100\mu_0$ by default.

In Fig. 12, subfigures $(a_1) - (a_3)$ present the loss history for both ATD-GLPINNs and GL-PINNs. Ssubfigure (a_4) illustrates the evolution of the task parameter ω_i, $(i = 2, \cdots, M)$ over epochs, where ω_1 is fixed at 1 by default, and ω_2 to ω_5 corresponds to the task parameter values of Sub_2 to Sub_5. The noise parameter settings are the same

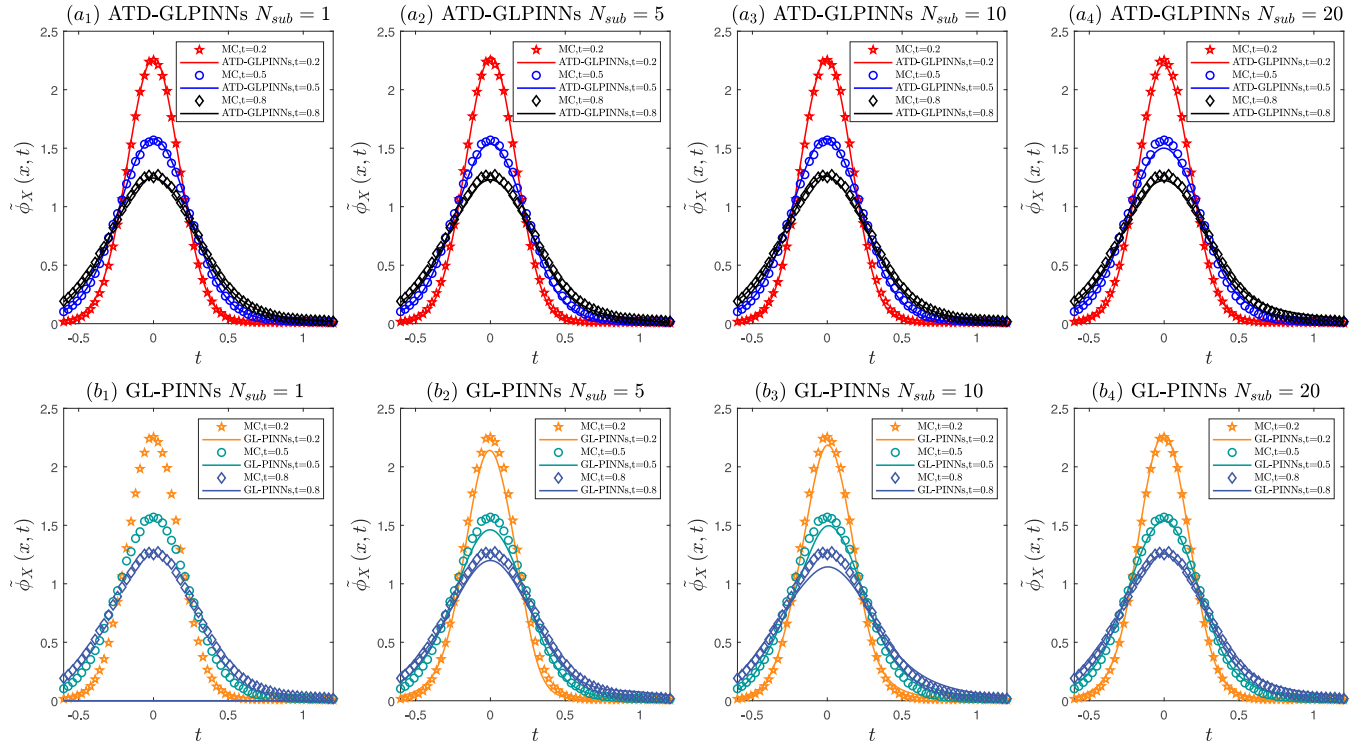

FIG. 10. The PDFs predicted by the ATD-GLPINNs (the first row) and GL-PINNs (the second row) algorithms correspond to the different numbers of GL quadrature sub-intervals N_{sub} at $t = 0.2, 0.5, 0.8$.

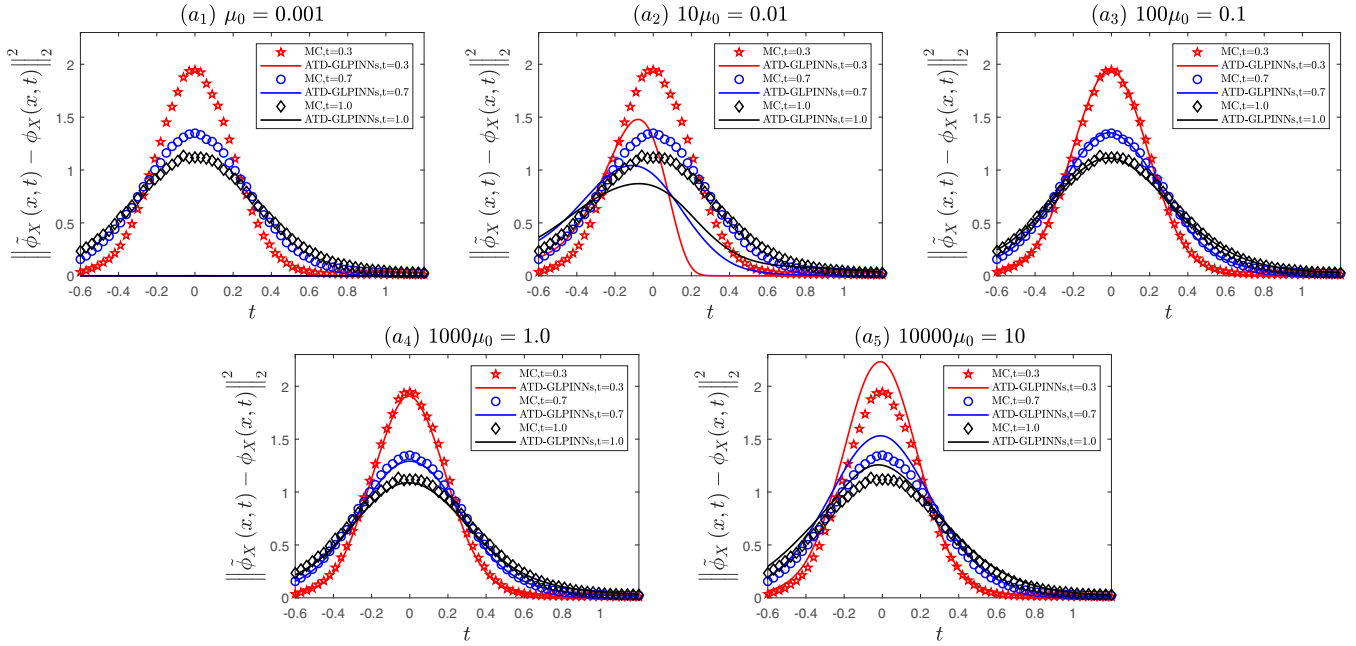

FIG. 11. The sensitivity test of the value of parameter μ with ATD-GLPINNs.

as in Fig. 10. As shown in the figure, the loss history of ATD-GLPINNs (red dashed line) decreases rapidly in the initial stages and reaches a low stable value with fewer epochs, whereas the loss history of GL-PINNs (blue dashed line) decreases more slowly and eventually stabilizes at a higher loss value. It is evident from the results that ATD-GLPINNs not only converges faster but also maintains lower loss values across all loss terms, demonstrating superior performance and convergence efficiency. Additionally, subfigure (a_4) shows that the ATD-GLPINNs algorithm trains multiple subtasks simultaneously while prioritizing the optimization of those closer to the initial condition. This demonstrates that the ATD-GLPINNs adheres to the principle of temporal causality.

B. Example 2

Consider the following nonlinear van der Pol oscillator with combined Gaussian and Poisson white noises

$$\ddot{X}(t) + \gamma(-1 + X(t)^2 + \dot{X}(t)^2)\dot{X}(t) + X(t) = W_G(t) + W_P(t), \tag{22}$$

let $X_1(t) = X(t)$, $X_2(t) = \dot{X}_1(t)$, the system Eq. (22) can be expressed as the following SDE

$$\mathrm{d}\begin{bmatrix} X_1(t) \\ X_2(t) \end{bmatrix} = \begin{bmatrix} X_2(t) \\ -\gamma(-1 + X_1(t)^2 + X_2(t)^2)X_2(t) - X_1(t) \end{bmatrix}\mathrm{d}t + \begin{bmatrix} 0 \\ \sqrt{k} \end{bmatrix}\mathrm{d}W(t) + \begin{bmatrix} 0 \\ R \end{bmatrix}\mathrm{d}P(t). \tag{23}$$

Then, the forward Kolmogorov equation excited by Gaussian and Poisson white noises is as follows:

$$\begin{aligned} \frac{\partial}{\partial t}\phi_X(x_1, x_2, t) = &\left[\gamma\left(-1 + x_1^2 + 3x_2^2\right) - \lambda\right]\phi_X(x_1, x_2, t) - x_2\frac{\partial \phi_X(x_1, x_2, t)}{\partial x_1} \\ &+ \left[\gamma\left(-1 + x_1^2 + x_2^2\right)x_2 + x_1\right]\frac{\partial \phi_X(x_1, x_2, t)}{\partial x_2} + \frac{k}{2}\frac{\partial^2 \phi_X(x_1, x_2, t)}{\partial x_2^2} \\ &+ \frac{\lambda}{\sqrt{2\pi\sigma_R^2}}\int_{-\infty}^{\infty}\phi_X(x_1, x_2 - r, t)\exp\left(-\frac{r^2}{2\sigma_R^2}\right)\mathrm{d}r. \end{aligned} \tag{24}$$

We set $N_u = 1000$, $N_b = 1000$, $N_f = 3000$, $N_n = 6000$, and $\mathrm{M} = 5$. Therefore, with each subtask allocated the number of training points $N_{b_i} = 200$, $N_{f_i} = 600$, and $N_{n_i} = 1200$. The learning rate is fixed as 0.005. Ω is given as $[0, T] \times [x_{1_{lb}}, x_{1_{ub}}] \times [x_{2_{lb}}, x_{2_{ub}}] = [0, 1] \times [-2.5, 2.5] \times [-2.5, 2.5]$. Unless noted, we consider $\gamma = 0.2$. The initial condition values follow a multivariate normal distribution, where $\boldsymbol{\zeta} = [\zeta_1, \zeta_2]^T$ and $\Sigma = diag[\sigma_1^2, \sigma_2^2]$. We set $\zeta_1 = 0.0$, $\zeta_2 = 0.0$, $\sigma_1^2 = 0.1$, and $\sigma_2^2 = 0.1$. The training process is conducted 20,000 epochs. More details on the hyperparameters and noise settings are available in the associated parts of each results.

To further evaluate the performance of the ATD-GLPINNs, we obtain the errors for various μ and the equation parameter

TABLE III. The accuracy of the ATD-GLPINNs and GL-PINNs in different values of hyperparameter μ and equation parameter γ.

Parameter	State Variables	$\mu_0 = 0.001$	$10\mu_0 = 0.01$	$100\mu_0 = 0.1$	$1000\mu_0 = 1.0$	$10000\mu_0 = 10.0$	GL-PINNs
$\gamma = -0.2$	X_1	5.88×10^{-3}	2.03×10^{-3}	2.76×10^{-4}	2.00×10^{-3}	1.01×10^{-2}	4.96×10^{-3}
	X_2	3.22×10^{-3}	2.28×10^{-3}	6.14×10^{-5}	8.19×10^{-4}	5.39×10^{-3}	2.12×10^{-3}
$\gamma = -0.1$	X_1	7.11×10^{-3}	2.20×10^{-3}	1.12×10^{-3}	1.08×10^{-2}	1.86×10^{-3}	1.73×10^{-3}
	X_2	5.21×10^{-3}	1.89×10^{-3}	4.34×10^{-4}	5.18×10^{-3}	8.16×10^{-4}	9.50×10^{-4}
$\gamma = 0.1$	X_1	3.83×10^{-3}	4.53×10^{-3}	2.54×10^{-4}	5.46×10^{-4}	8.83×10^{-4}	2.04×10^{-3}
	X_2	3.39×10^{-3}	2.49×10^{-3}	1.11×10^{-4}	3.74×10^{-4}	8.81×10^{-4}	9.32×10^{-4}
$\gamma = 0.2$	X_1	1.30×10^{-2}	2.24×10^{-3}	2.23×10^{-4}	6.61×10^{-4}	1.96×10^{-3}	1.43×10^{-3}
	X_2	9.19×10^{-3}	3.31×10^{-3}	1.33×10^{-4}	6.88×10^{-4}	2.13×10^{-3}	6.66×10^{-4}

γ, as shown in Table III. Columns 3 to 7 correspond to various μ values, ranging from 0.001 to 10, while the last column represents the error of the GL-PINNs. We set the noise intensity to $k = 0.2$, $\lambda \times E[R^2] = 0.4$, and the mean arrival rate $\lambda = 0.5$. As indicated in Table III, the performance of ATD-GLPINNs is sensitive to the hyperparameter μ, and the model with $100\mu_0$ achieves the best results among the tested μ values. Moreover, ATD-GLPINNs maintains high computational accuracy under different values of γ.

The impact of the mean arrival rate λ on the ATD-GLPINNs algorithm is highlighted in Fig. 13, where the value of λ ranges from 0.1 to 2.0. The noise intensities for both Gaussian and Poisson noises are set to 0.02. As seen in this figure, the error of the ATD-GLPINNs algorithm (red and orange solid lines) are relatively lower and exhibit better stability across different mean arrival rates.

The marginal PDF corresponding to different numbers of residual points N_f is shown in Fig. 14, where the number of N_f ranges from 2000 to 10000. The two noise intensities are set as $k = 0.1$, $\lambda \times E[R^2] = 0.4$, and the mean arrival rate $\lambda = 0.5$. The PDF predicted by ATD-GLPINNs demonstrates a higher degree of agreement with the MC reference solutions, particularly when using fewer residual points. In comparison, the PDF predicted by GL-PINNs exhibits some deviation under the same conditions. These results indicate that ATD-GLPINNs achieves greater accuracy and consistency even with fewer residual points, whereas GL-PINNs needs significantly more residual points to attain similar levels of accuracy.

Figure 15 displays the results for different training epochs with $\lambda = 0.5$, where the number of epochs ranges from 5000 to 30000. Both Gaussian and Poisson noise have the same intensity, set to 0.02. The first row shows the error of variable X_1, while the second row shows the error of variable X_2. As the number of training epochs increases, the error of ATD-GLPINNs gradually decreases. In contrast, GL-PINNs shows poorer performance under the same training epoch setting. This demonstrates the effectiveness of our strategy for time-dependent problems, exhibiting lower computational cost.

Figure 16 illustrates the loss history of ATD-GLPINNs and GL-PINNs in subfigures $(a_1) - (a_3)$, while subfigure (a_4) depicts the evolution of task parameter ω_i, $(i = 2, \cdots, M)$ over epochs. The noise intensity is set to $k = 0.1$, $\lambda \times E[R^2] = 0.2$, and the mean arrival rate $\lambda = 0.5$. The loss history of ATD-GLPINNs decreases more rapidly after training begins and converges to smaller values in the later stages of training, which demonstrates that ATD-GLPINNs can achieve stronger stability and better optimization performance during the training process. Additionally, the subfigure (a_4) shows that the ATD-GLPINNs terminates training when all task parameters approach one and the total epochs are completed.

The comparison of joint PDFs corresponding to different variances of the initial conditions is shown in Fig. 17, where the variance ranges from 0.005 to 1.0. The intensities of Gaussian and Poisson white noises are both set as 0.05, and the mean arrival rate $\lambda = 0.1$. The first to third columns correspond to the PDF of the MC reference solutions,

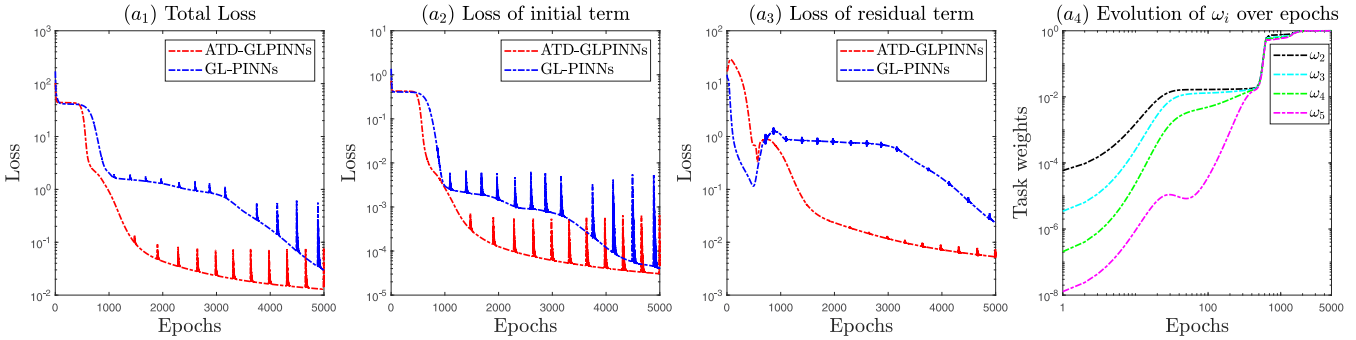

FIG. 12. The loss history of ATD-GLPINNs (red dashed line) and GL-PINNs (blue dashed line), in which subfigures $(a_1) - (a_3)$ correspond to the total loss, the loss of initial term, and the loss of residual term, respectively. The subfigure (a_4) shows a log-log plot for the evolution of task parameter ω_i, $(i = 2, \cdots, M)$ over epochs, where ω_1 is fixed at 1 by default.

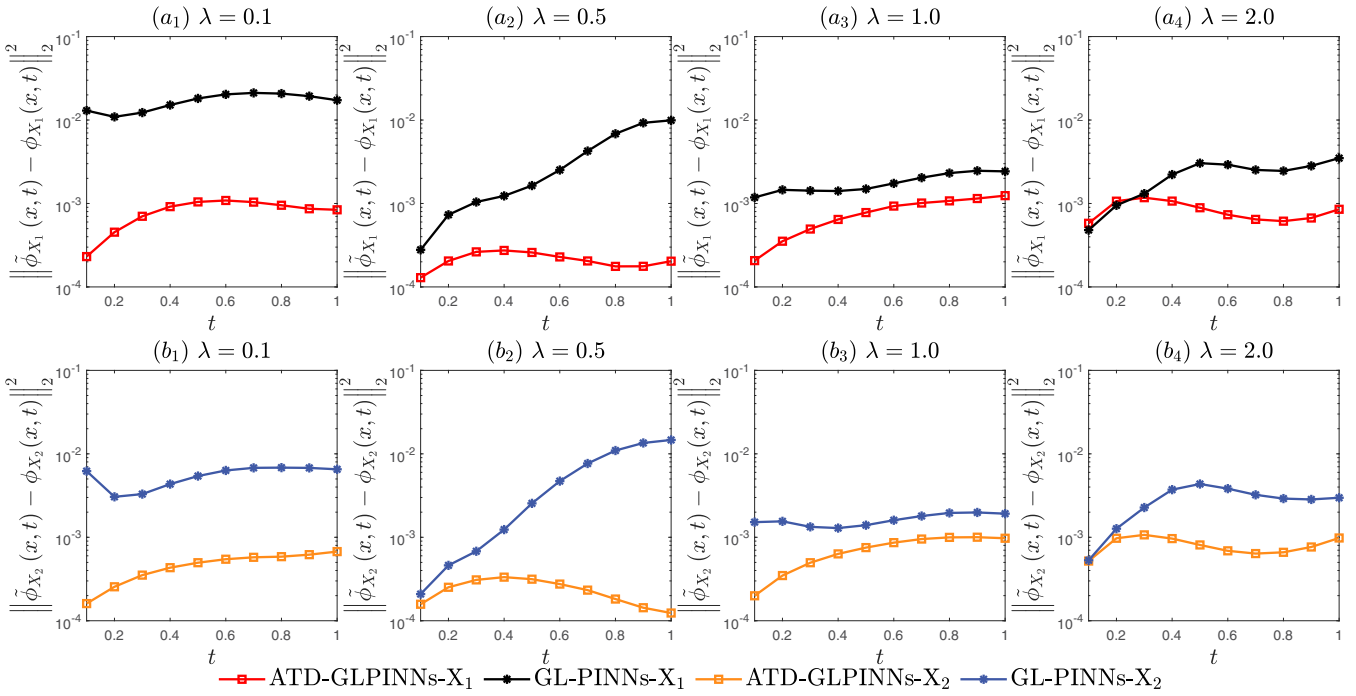

FIG. 13. The variations in errors across the entire time domain for different mean arrival rates λ. The first row shows the error of variable X_1 and the second row of X_2.

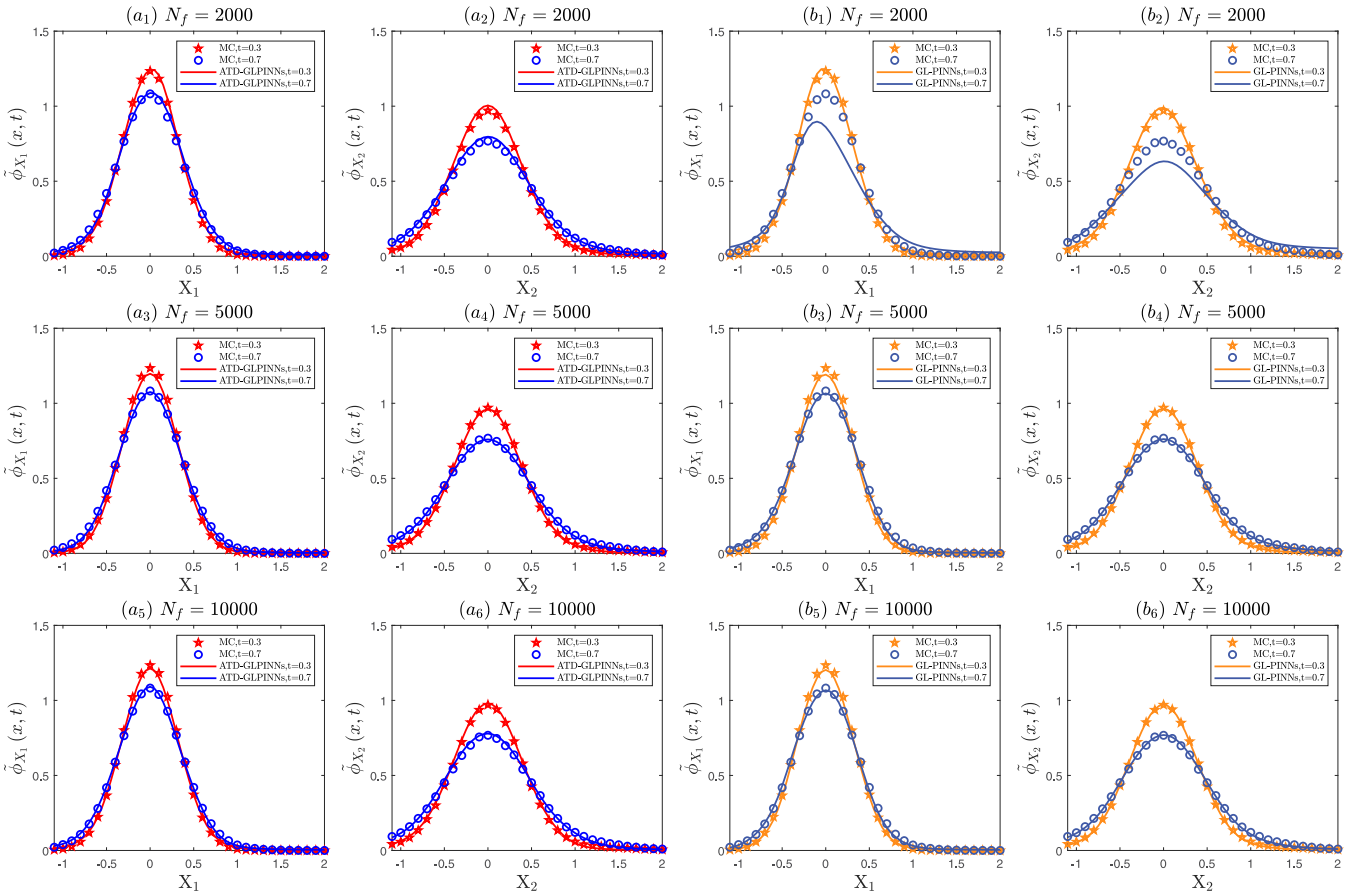

FIG. 14. The marginal PDFs predicted by the ATD-GLPINNs (Columns 1–2) and GL-PINNs (Columns 3–4) correspond to the different numbers of residual points N_f at $t = 0.3$ and $t = 0.7$. The colored lines are the predicted solutions from ATD-GLPINNs and GL-PINNs, and the colored symbols denote the MC reference solutions.

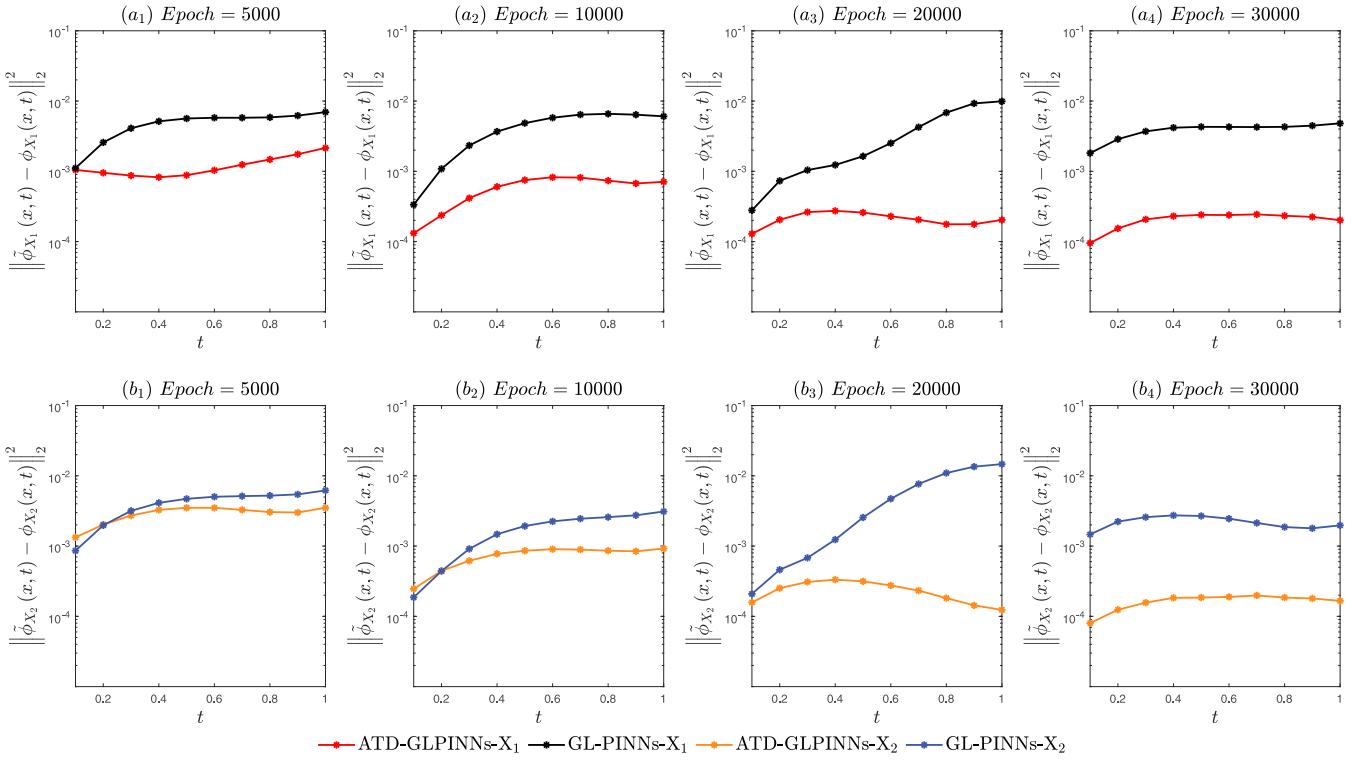

FIG. 15. The variations in errors across the entire time domain for different numbers of training epochs. The first row shows the error of variable X_1, and the second row shows the error of X_2.

the ATD-GLPINNs predicted solutions, and the GL-PINNs predicted solutions, respectively. As the variance decreases, the corresponding joint PDF peak in the figure becomes sharper. The outcome demonstrates that even under sharp initial conditions, ATD-GLPINNs can accurately predict the solution of the forward Kolmogorov equation (24), which further demonstrates the effectiveness of our approach.

Figure 18 presents a comparison of the capability between ATD-GLPINNs and GL-PINNs under different noise intensities with $\lambda = 0.5$. The intensity of Gaussian white noise is randomly selected within the range of 0.02 to 0.3, while the intensity of Poisson white noise is randomly chosen within the range of 0.02 to 0.5. As the noise intensities change, ATD-GLPINNs consistently maintains a lower error around 10^{-3}, which demonstrates superior robustness and stability. This highlights the exceptional capability of our method to precisely capture the dynamic behaviors of the system and maintain high accuracy.

Figure 19 evaluates the prediction accuracy of our method under different values of N_{sub}, where the value of N_{sub} ranges from 1 to 20. The noise intensity is set to $k = 0.1$, $\lambda \times E[R^2] = 0.2$, and the mean arrival rate $\lambda = 0.5$. As indicated by the color bar, it can be seen that dividing the sub-intervals into too few sections significantly reduces the accuracy of GL-PINNs, whereas ATD-GLPINNs maintains high accuracy even with fewer integration points.

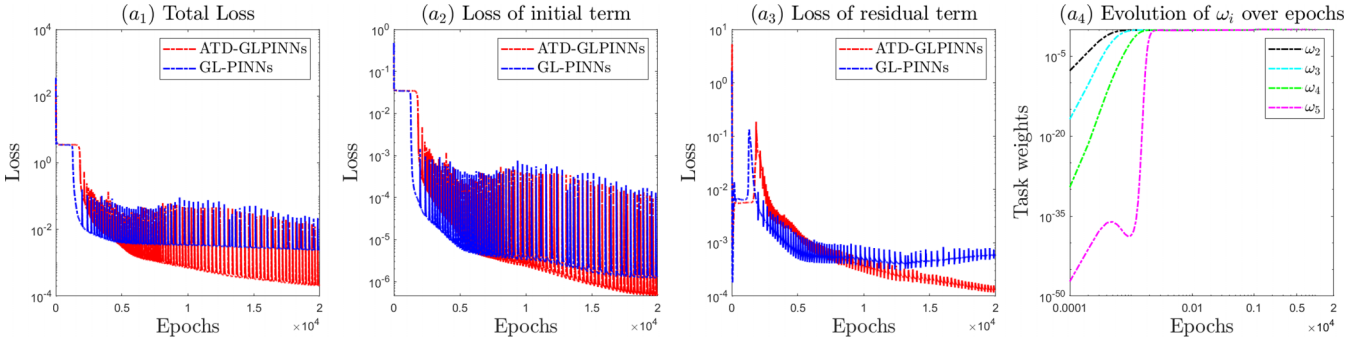

FIG. 16. The loss history of ATD-GLPINNs (red dashed line) and GL-PINNs (blue dashed line), in which subfigures $(a_1) - (a_3)$ correspond to the total loss, the loss of initial term, and the loss of residual term, respectively. The subfigure (a_4) shows a log-log plot for the evolution of task parameter ω_i, $(i = 2, \cdots, M)$ over epochs, where ω_1 is fixed at 1 by default.

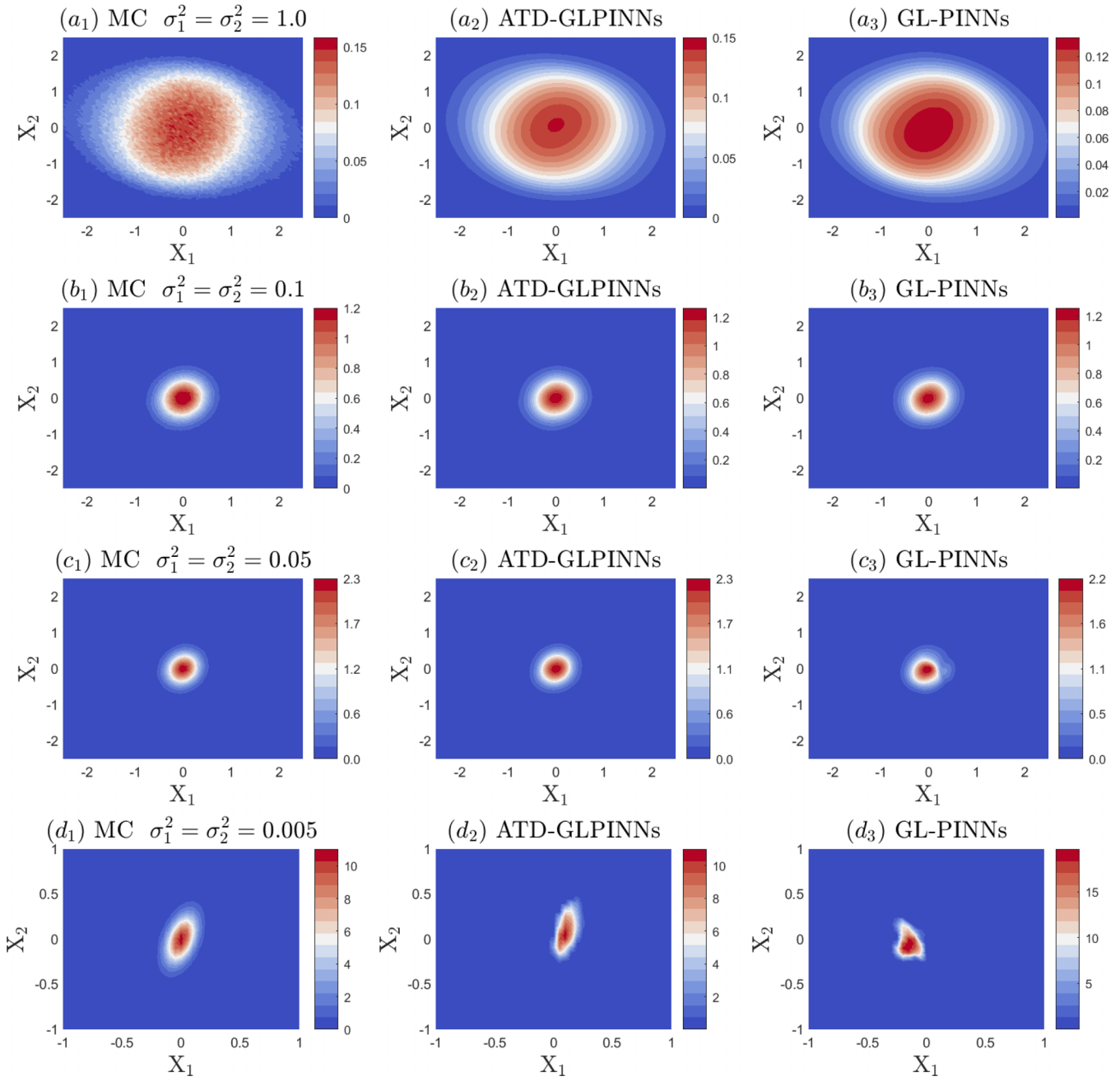

FIG. 17. The joint PDFs predicted by the ATD-GLPINNs (the second column) and GL-PINNs (the third column) algorithms correspond to the different initial conditions at $t = 0.5$, where the symbols σ_1^2 and σ_2^2 represent the variance value of the initial conditions.

C. Example 3

Consider the following Duffing oscillator driven by both Gaussian and Poisson white noises

$$\ddot{X}(t) + \beta\dot{X}(t) - \alpha X(t) + \delta X(t)^3 = cX(t)W_G(t) + eX(t)W_P(t), \tag{25}$$

let $X_1(t) = X(t)$, $X_2(t) = \dot{X}_1(t)$, the system Eq. (25) can be rewritten in the following SDE

$$\mathrm{d}\begin{bmatrix} X_1(t) \\ X_2(t) \end{bmatrix} = \begin{bmatrix} X_2(t) \\ -\beta X_2(t) + \alpha X_1(t) - \delta X_1(t)^3 \end{bmatrix}\mathrm{d}t + \begin{bmatrix} 0 \\ c\sqrt{k}X_1(t) \end{bmatrix}\mathrm{d}W(t) + \begin{bmatrix} 0 \\ eRX_1(t) \end{bmatrix}\mathrm{d}P(t), \tag{26}$$

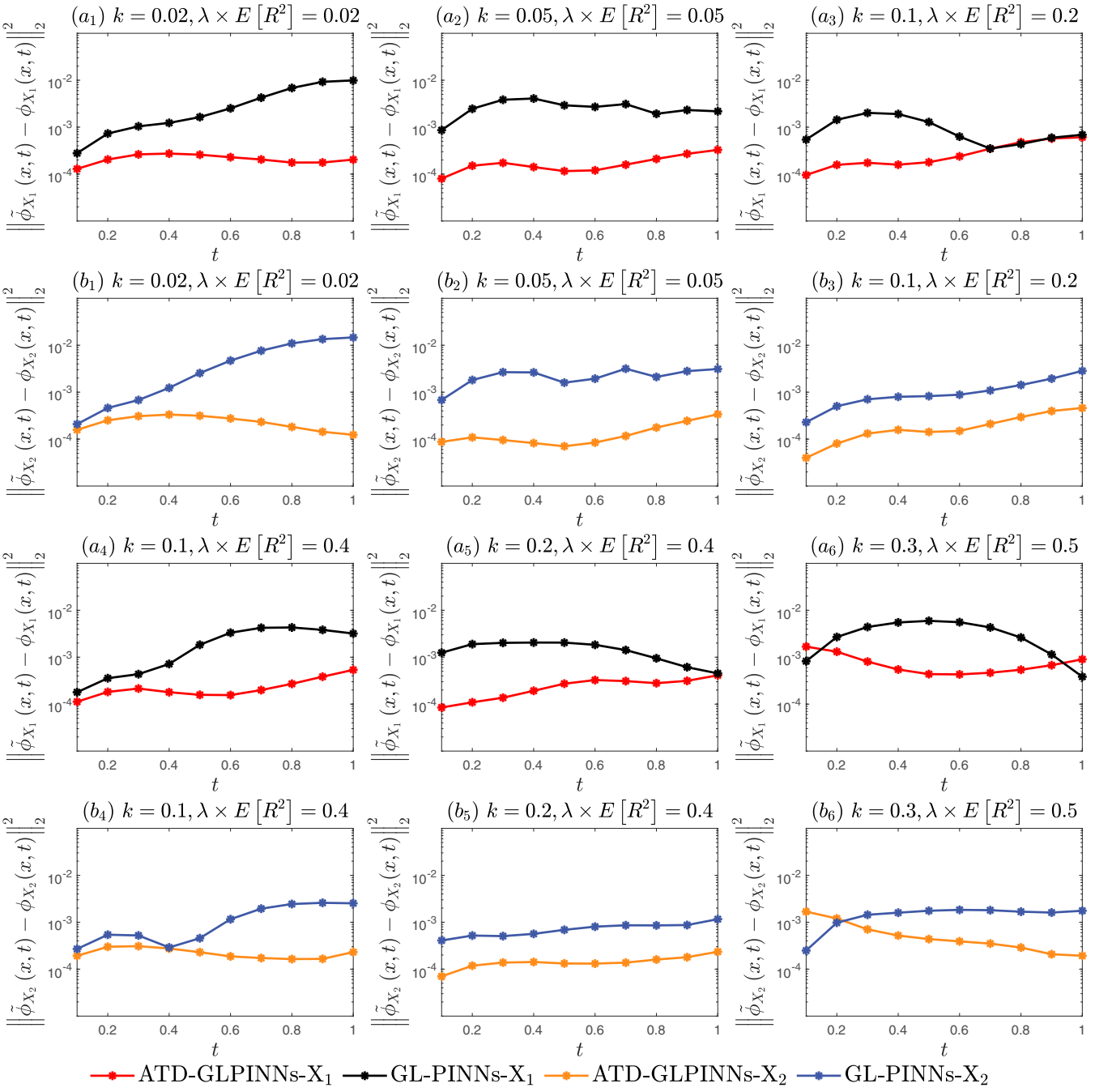

FIG. 18. The variations in errors across the entire time domain under different intensities of Gaussian and Poisson white noise. The k represents the intensity of Gaussian white noise, while $\lambda \times E[R^2]$ represents the intensity of Poisson white noise. The first and third rows show the error of variable X_1 and the second and fourth row of X_2.

the forward Kolmogorov equation corresponding to Eq. (26) as follows

$$\begin{aligned}\frac{\partial}{\partial t}\phi_X(x_1,x_2,t) &= (\beta-\lambda)\phi_X(x_1,x_2,t) - x_2\frac{\partial}{\partial x_1}\phi_X(x_1,x_2,t)\\ &+ \left(\beta x_2 - \alpha x_1 + \delta {x_1}^3\right)\frac{\partial}{\partial x_2}\phi_X(x_1,x_2,t)\\ &+ \frac{k}{2}c^2 {x_1}^2\frac{\partial^2}{\partial x_2^2}\phi_X(x_1,x_2,t) + \frac{\lambda}{\sqrt{2\pi\sigma_R^2}}\int_{-\infty}^{+\infty}\phi_X\left(x_1,\frac{x_2}{1+er},t\right)\exp\left(-\frac{r^2}{2\sigma_R^2}\right)\mathrm{d}r.\end{aligned} \tag{27}$$

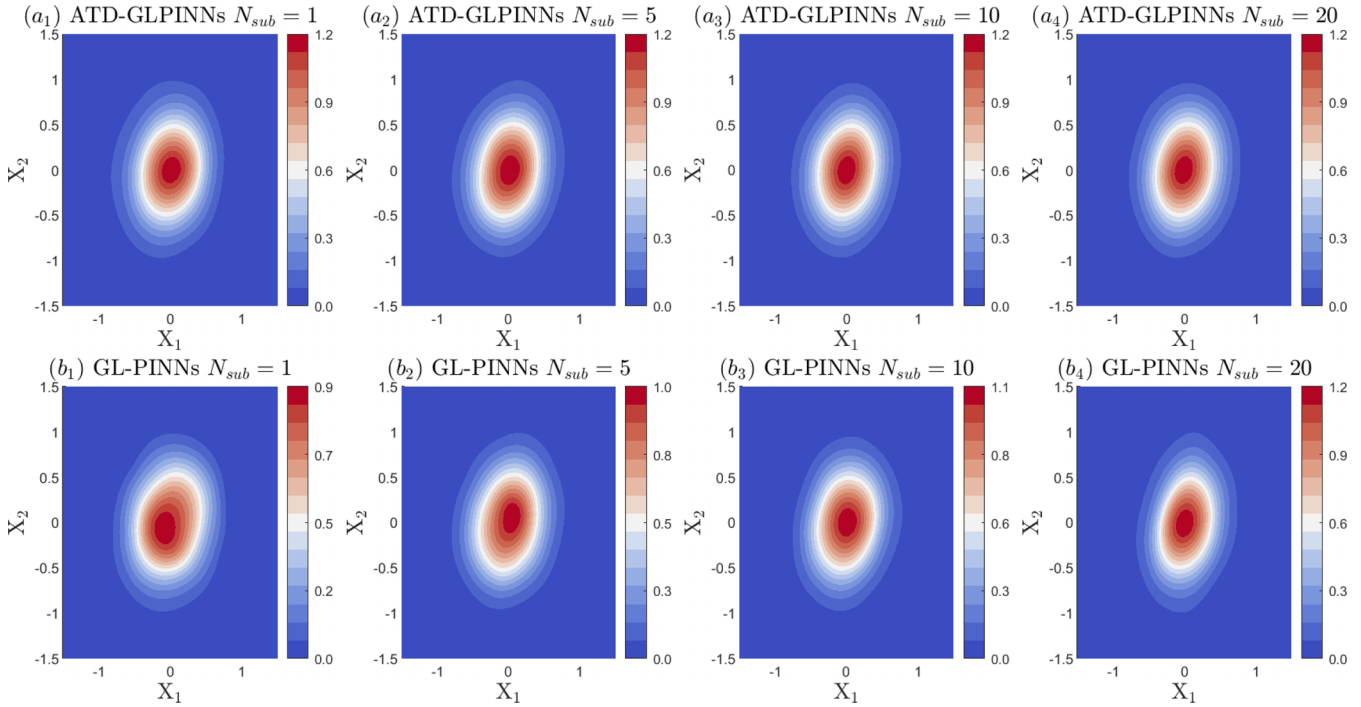

FIG. 19. The joint PDFs predicted by the ATD-GLPINNs (the first row) and GL-PINNs (the second row) algorithms correspond to the different numbers of GL sub-intervals N_{sub} at $t = 0.5$.

We set $N_u = 1000$, $N_b = 1000$, $N_f = 5000$, $N_n = 10000$, and $M = 5$. Thus, with each subtask allocated the number of training points $N_{b_i} = 200$, $N_{f_i} = 1000$, $N_{n_i} = 2000$. The learning rate is fixed as 0.005. The Ω is given as $[0, T] \times [x_{1_{lb}}, x_{1_{ub}}] \times [x_{2_{lb}}, x_{2_{ub}}] = [0, 1] \times [-2, 2] \times [-2, 2]$. Unless stated, we consider $\beta = 0.1$, $\alpha = 1.0$, $\delta = 2.0$, $a = 1.0$ and $b = 1.0$. The initial condition values follow a multivariate normal distribution, where $\boldsymbol{\zeta} = [\zeta_1, \zeta_2]^T$ and $\Sigma = diag[\sigma_1^2, \sigma_2^2]$. We set $\zeta_1 = 0.0$, $\zeta_2 = 0.0$, $\sigma_1^2 = 0.05$, $\sigma_2^2 = 0.05$. The training process is conducted 20,000 epochs. More details on the hyperparameters and noise settings are available in the associated parts of each results.

In Table IV, we investigate the effect of the nonlinear stiffness coefficient δ on the accuracy of the ATD-GLPINNs and GL-PINNs when δ takes values of 0.5, 1.0, 2.0, 3.0, and 4.0, with results provided at $t = 0.3$ and $t = 0.7$. Both the two noise intensities and the mean arrival rate of the Poisson white noise are set to 0.1. The results demonstrate that regardless of the variations in the nonlinear stiffness coefficient δ, the ATD-GLPINNs maintains a high level of predictive accuracy, with the errors at least on the order of 10^{-4}, indicating the stability and high accuracy of our approach.

The joint PDFs predicted by ATD-GLPINNs and GL-PINNs under the influence of different numbers of GL sub-intervals N_{sub} are presented in Fig. 20, with $\lambda = 0.2$. The intensity of Gaussian white noise is set to $k = 0.2$ and the intensity of Poisson white noise is set to 0.4. As can be seen from the color bar on the right, the change in the number of N_{sub} has a noticeable effect on the peak of the joint PDF predicted by GL-PINNs. The results show that ATD-GLPINNs provides more accurate joint PDF predictions with a smaller number of N_{sub}, while GL-PINNs requires a greater number of N_{sub} to achieve similar predictive accuracy.

In Fig. 21, we illustrate the effect of varying the mean arrival rate λ on the predictive accuracy of ATD-GLPINNs and GL-PINNs, with λ ranging from 0.01 to 2.0. The intensity of two noises are set as 0.2. The red and black solid

TABLE IV. The accuracy of the ATD-GLPINNs and GL-PINNs in different values of equation parameter δ.

Method	State Variables	$\delta = 0.5$	$\delta = 1.0$	$\delta = 2.0$	$\delta = 3.0$	$\delta = 4.0$
ATD-GLPINNs-$t = 0.3$	X_1	5.90×10^{-4}	1.50×10^{-4}	3.40×10^{-4}	5.71×10^{-4}	4.13×10^{-4}
	X_2	5.30×10^{-4}	1.60×10^{-4}	1.89×10^{-4}	6.46×10^{-4}	4.15×10^{-4}
GL-PINNs-$t = 0.3$	X_1	2.92×10^{-3}	4.18×10^{-3}	1.39×10^{-3}	1.43×10^{-2}	3.11×10^{-3}
	X_2	2.44×10^{-3}	3.62×10^{-3}	2.46×10^{-3}	1.52×10^{-2}	2.74×10^{-3}
ATD-GLPINNs-$t = 0.7$	X_1	4.33×10^{-4}	5.83×10^{-4}	5.04×10^{-4}	9.49×10^{-4}	9.66×10^{-4}
	X_2	4.61×10^{-4}	5.61×10^{-4}	4.43×10^{-4}	1.53×10^{-3}	9.37×10^{-4}
GL-PINNs-$t = 0.7$	X_1	1.09×10^{-3}	5.44×10^{-3}	1.40×10^{-3}	1.80×10^{-2}	1.53×10^{-3}
	X_2	1.21×10^{-3}	5.67×10^{-3}	1.93×10^{-3}	2.12×10^{-2}	2.70×10^{-3}

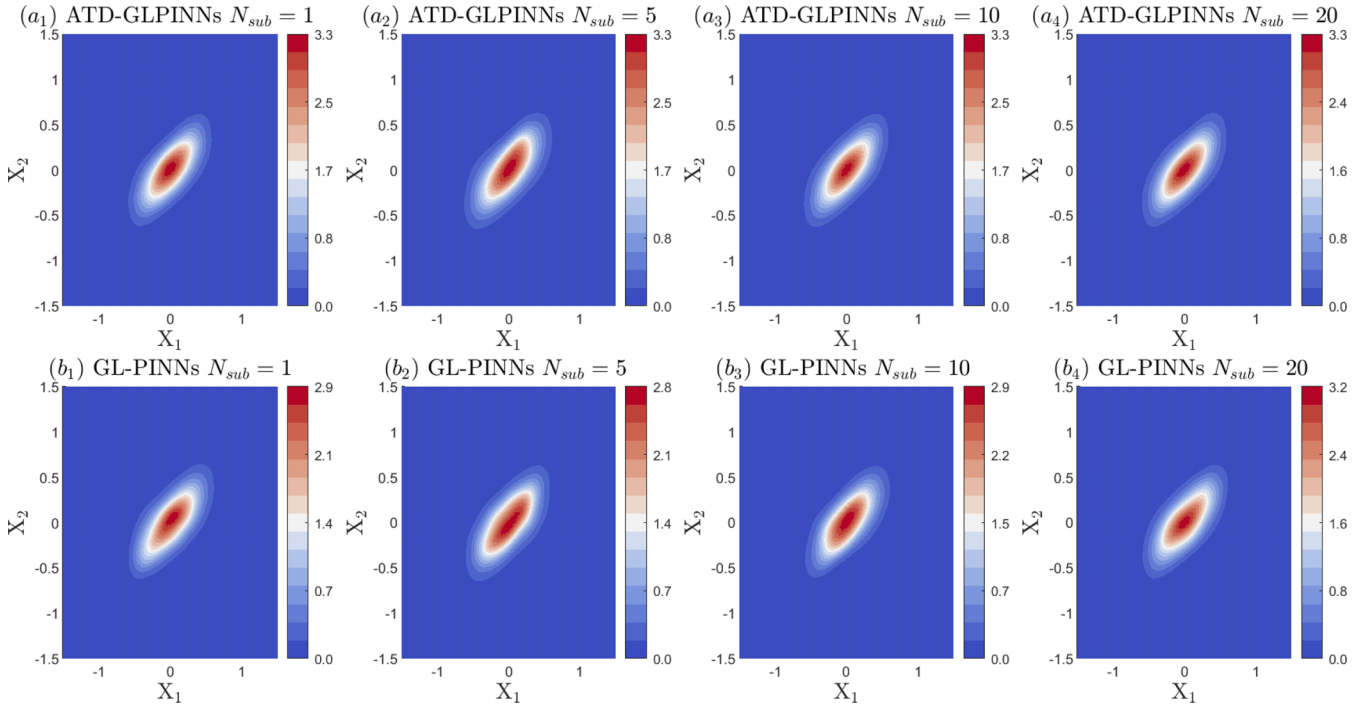

FIG. 20. The joint PDFs predicted by the ATD-GLPINNs (the first row) and GL-PINNs (the second row) correspond to the different numbers of GL sub-intervals N_{sub} at $t = 0.5$.

lines represent the errors between the ATD-GLPINNs and GL-PINNs predicted solutions and the MC reference solutions for variable X_1, respectively, while the orange and blue solid lines represent the errors for variable X_2. It can be seen that ATD-GLPINNs maintains high prediction accuracy regardless of changes in the mean arrival rate, when the intensity of Poisson white noise is fixed.

In Fig. 22, we present the variations in errors under different training epoch settings, with the number of epochs ranging from 5,000 to 30,000. The noise parameter settings are the same as in Fig. 21. The results show that the suggested adaptive decomposition strategy significantly improves the computational efficiency and maintains high accuracy with fewer epochs. To ensure the best performance of the two algorithms, we set the total epochs $E = 20000$ for training.

Figure 23 shows the trends in total loss, initial term loss, and residual term loss for ATD-GLPINNs (red dashed line) and GL-PINNs (blue dashed line) in subfigures $(a_1) - (a_3)$. subfigure (a_4) illustrates the progression of the task parameter ω_i, $(i = 2, \cdots, M)$ throughout the training epochs. The noise parameters are setting as $k = 0.01$, $\lambda \times E[R^2] = 0.1$ and $\lambda = 0.2$. It can be observed that, as training epochs increase, each loss component in ATD-GLPINNs converges to lower values more rapidly than in GL-PINNs. The rapid decrease in the loss history suggests that ATD-GLPINNs can quickly fit the target function and capture key features when handling complex problems, allowing the model to reach the optimal solution sooner and improving overall computational efficiency. Additionally, the last subfigure shows that the ATD-GLPINNs follows temporal causality and stops training when all task parameters approach one and the total epochs have been completed.

As indicated in Fig. 24, we demonstrate the error between the ATD-GLPINNs and GL-PINNs compared to the MC reference solutions across the whole time domain under different intensities of two noises. Here, the mean arrival rate $\lambda = 0.2$. The noise intensities in this example are randomly selected from the range of 0.01 to 1.0. As noise intensity increases, the prediction errors of ATD-GLPINNs for X_1 and X_2 (shown in red and orange solid line, respectively) remain consistently low. In contrast, the errors of GL-PINNs (shown in black and blue solid line) show a clear upward trend as noise intensity rises, with significantly larger increases at higher noise levels. These results confirm the superior capability and robustness of ATD-GLPINNs in capturing complex dynamic behaviors under challenging conditions.

In Fig. 25, we show the joint PDF of ATD-GLPINNs and GL-PINNs with the number of residual points $N_f = 3000$, 5000, and 8000 and the hyperparameters $\mu = \mu_0$, $100\mu_0$, and $10000\mu_0$. The details of error using different values of μ and N_f at $t = 0.5$ are listed in Table V. The two noise intensities are set as $k = 1.0$, $\lambda \times E[R^2] = 1.0$, and the mean arrival rate $\lambda = 0.2$. The results reveal that even with relatively fewer residual points, the ATD-GLPINNs with $\mu = 100\mu_0$ can still maintain prediction accuracy very close to the MC reference solutions. In contrast, GL-PINNs shows relatively lower predictive accuracy when the number of residual points is limited.

The marginal PDFs predicted by the two approaches under different initial conditions are shown in Fig. 26, with variances set to 0.05, 0.1, and 0.5, and the mean arrival rate λ set to 0.2. The two noise intensities are set as $k = 0.2$, $\lambda \times E[R^2] = 0.4$. As the initial variance decreases, the peaks of the predicted marginal PDFs become sharper. Under various initial conditions, the PDF predicted by ATD-GLPINNs are in high

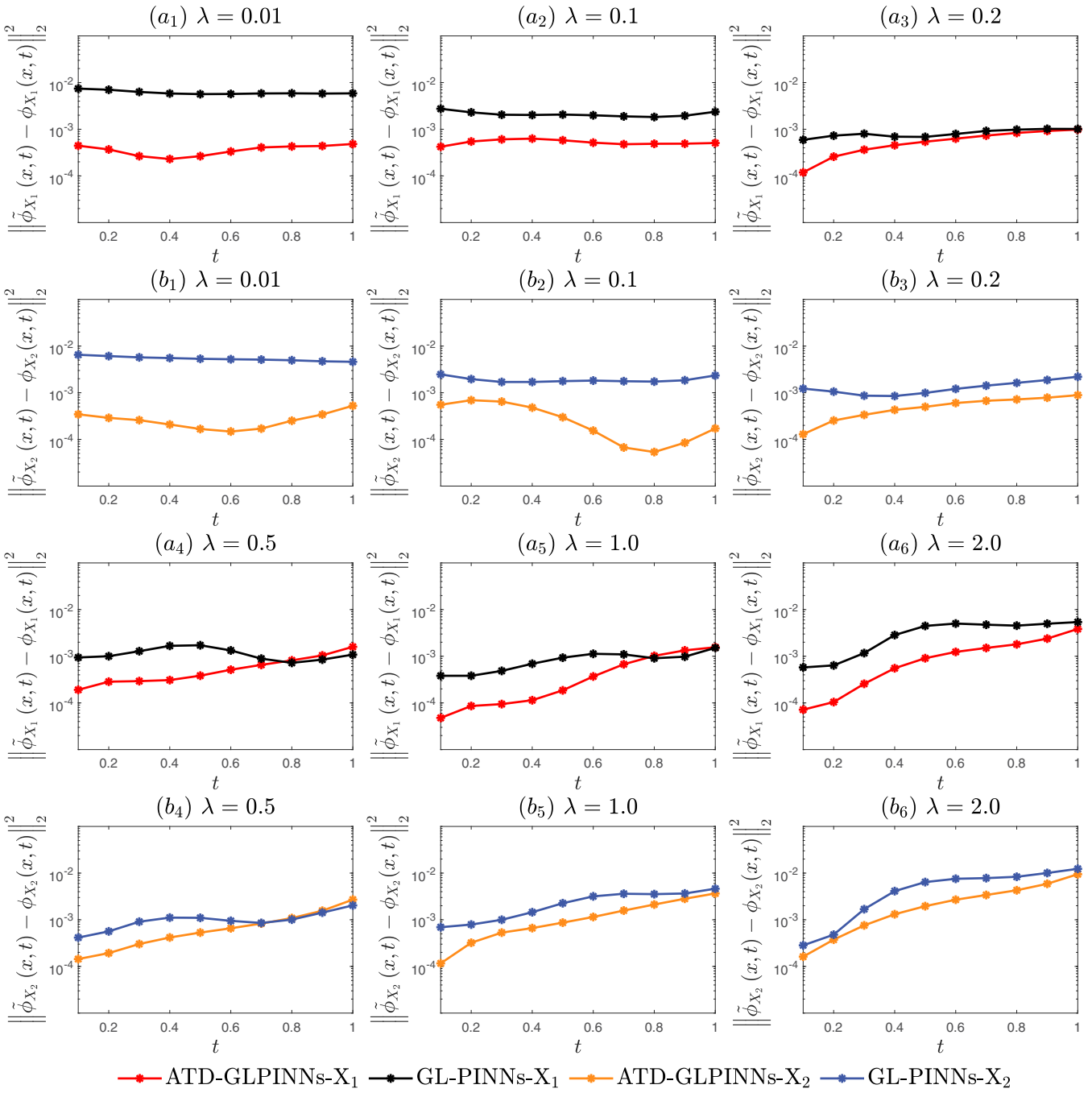

FIG. 21. The variations in errors across the entire time domain for different mean arrival rates λ. The first and third rows show the error of variable X_1, while the second and fourth rows show the error of X_2.

TABLE V. The accuracy of ATD-GLPINNs and GL-PINNs is influenced by the number of residual points N_f and the initial value of the task hyperparameter μ_0 at $t = 0.5$.

N_f	State Variables	$\mu_0 = 0.001$	$10\mu_0 = 0.01$	$100\mu_0 = 0.1$	$1000\mu_0 = 1.0$	$10000\mu_0 = 10.0$	GL-PINNs
3000	X_1	3.36×10^{-2}	3.98×10^{-2}	6.40×10^{-4}	2.24×10^{-3}	1.71×10^{-2}	5.07×10^{-3}
	X_2	1.00×10^{-2}	3.64×10^{-2}	8.23×10^{-5}	3.01×10^{-3}	1.54×10^{-2}	1.53×10^{-3}
5000	X_1	1.34×10^{-2}	2.38×10^{-3}	3.92×10^{-4}	9.52×10^{-4}	7.87×10^{-3}	1.56×10^{-2}
	X_2	1.04×10^{-2}	1.32×10^{-3}	1.15×10^{-4}	1.40×10^{-3}	6.00×10^{-3}	7.79×10^{-3}
8000	X_1	1.12×10^{-2}	1.37×10^{-2}	8.52×10^{-4}	5.10×10^{-3}	7.64×10^{-3}	3.55×10^{-3}
	X_2	8.00×10^{-3}	1.06×10^{-2}	4.99×10^{-4}	4.49×10^{-3}	5.36×10^{-3}	2.51×10^{-3}

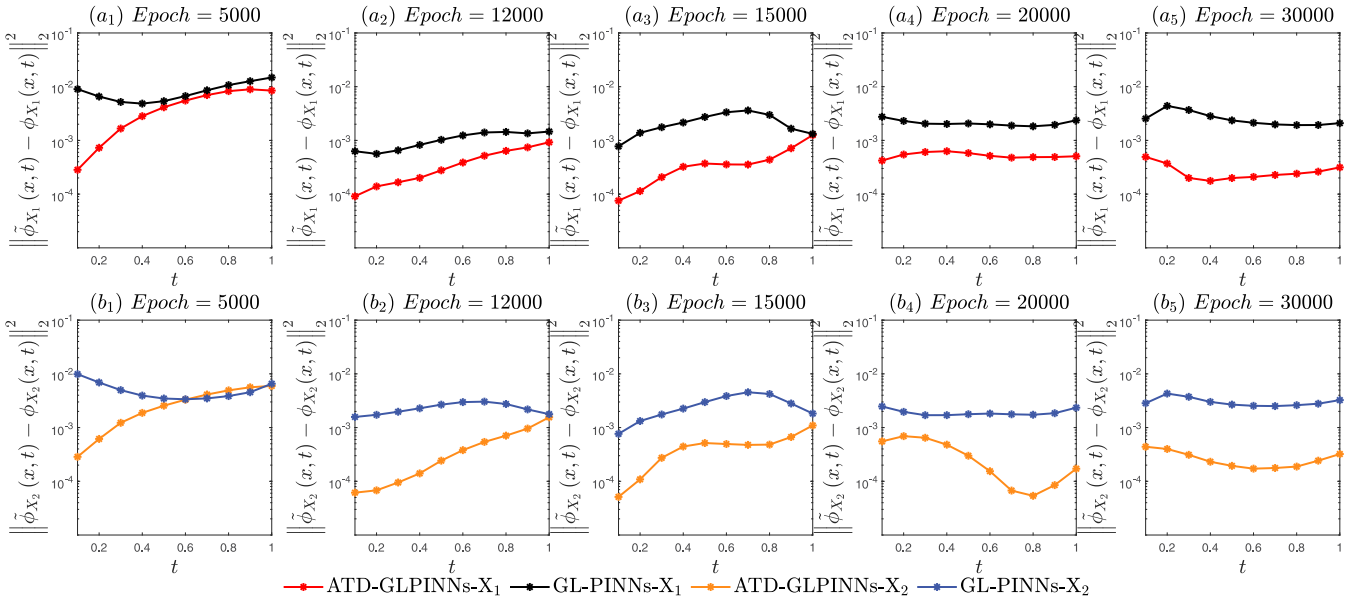

FIG. 22. The variations in errors across the entire time domain for different numbers of training epochs. The first row shows the error of variable X_1 and the second row of X_2.

agreement with the MC reference solutions, whereas those of GL-PINNs exhibits slight deviations. These results indicate that ATD-GLPINNs is more accurate and stable than GL-PINNs when handling different initial conditions.

V. CONCLUSIONS

In this study, we have developed a new approach referred to as ATD-GLPINNs, designed to address the issues of convergence difficulties and low accuracy caused by GL-PINNs not adhering to time causality. For time-dependent issues, the entire computational domain is evenly divided into an initial subtask and several extra tasks along the time axis. By adaptively adjusting task parameters ω_i, $(i = 2, \cdots, M)$, the training progress of each subtask is ensured to be based on the state of the preceding one, while keeping the weight of the initial subtask fixed. This dynamic weight adjustment mechanism ensures efficient use of training resources while maintaining temporal causality between subtasks, enabling effective multitask model training. Several numerical examples have demonstrated the effectiveness and robustness of our method, especially in scenarios with fewer residual points, fewer epochs, or sharp initial conditions. Additionally, we observed that ATD-GLPINNs exhibits strong convergence stability during the model optimization process. In contrast, the GL-PINNs method shows weaker generalization ability and unstable optimization performance when dealing with complex problems, which often requiring multiple runs to achieve a satisfactory convergence result.

In addition, compared to traditional approaches such as the finite difference method, path integral method, and others, the ATD-GLPINNs demonstrates significant advantages. It adopts a mesh-free structure, which enables efficient solutions for complex geometric domain problems. Additionally, the ATD-GLPINNs is highly adaptable and can be further extended to solve inverse problems in data-driven stochastic dynamical systems, which traditional approaches struggle to address.

It is worth noting that while the ATD-GLPINNs method demonstrates effectiveness in the numerical examples, it still

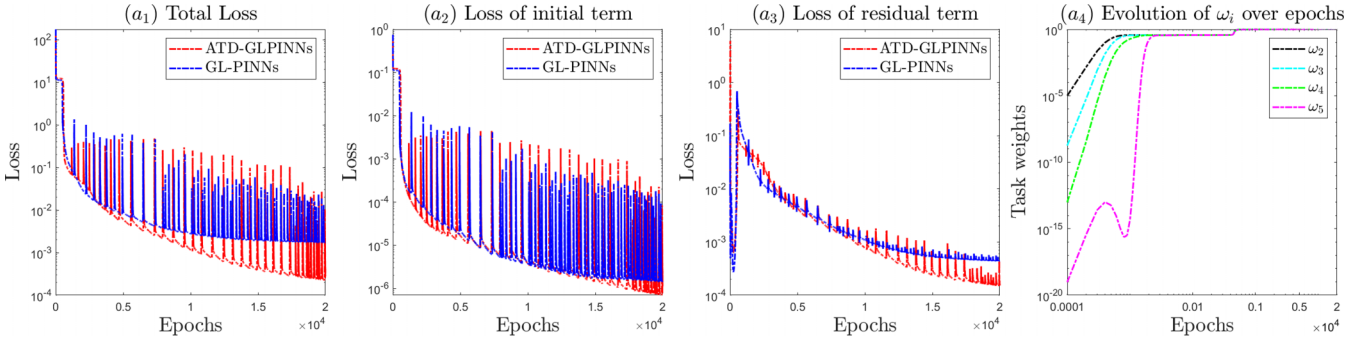

FIG. 23. The loss history of ATD-GLPINNs (red dashed line) and GL-PINNs (blue dashed line), in which subfigures $(a_1) - (a_3)$ correspond to the total loss, the loss of initial term, and the loss of residual term, respectively. The subfigure (a_4) shows a log-log plot for the evolution of task parameter ω_i, $(i = 2, \cdots, M)$ over epochs, where ω_1 is fixed at 1 by default.

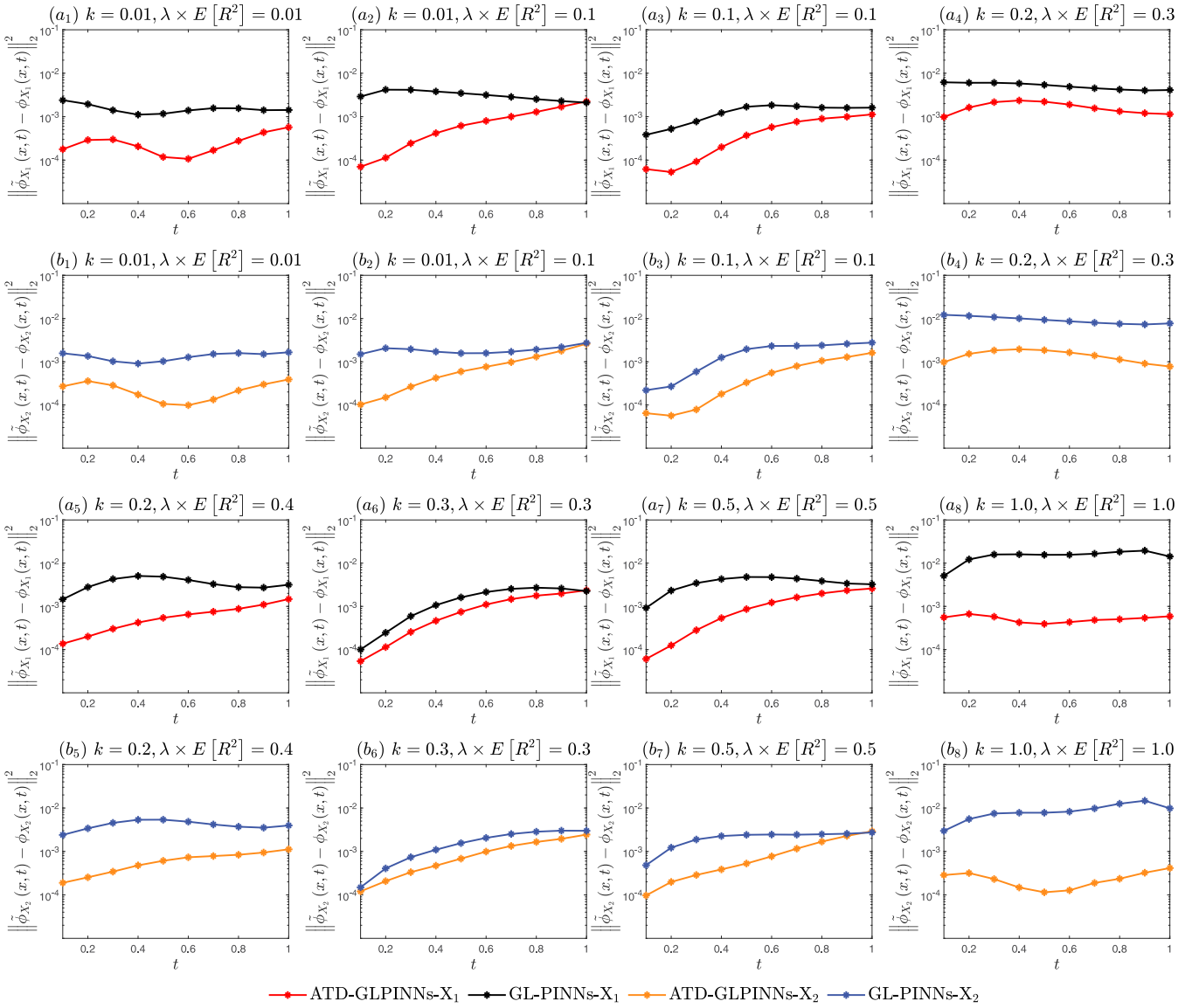

FIG. 24. The variations in errors across the entire time domain under different intensities of Gaussian and Poisson white noise. The k represents the intensity of Gaussian white noise, while $\lambda \times E[R^2]$ represents the intensity of Poisson white noise. The first and third row show the error of variable X_1 and the second and fourth row of X_2.

has certain limitations: (1) For the integrodifferential form of the forward Kolmogorov equation considered in this paper, the GL quadrature formula is employed for numerical integration, which significantly increases the computational complexity and cost in high-dimensional problems. (2) The ATD-GLPINNs is specifically designed for systems driven by Gaussian and Poisson white noises. Nevertheless, it is not well-suited for fractional-order dynamical systems with significant non-Markovian properties, dynamical systems driven by Lévy noise, and so on. (3) The performance of ATD-GLPINNs heavily depends on the selection of optimal hyperparameters. A reasonable hyperparameter configuration can significantly enhance the algorithm's convergence and optimization. However, identifying suitable hyperparameters for complex problems remains challenging. Therefore, future research can explore alternative numerical integration methods to improve the efficiency and accuracy of computations in high-dimensional problems. Moreover, the ATD-GLPINNs algorithm could be extended to stochastic dynamical systems driven by non-Gaussian noise to enhance its applicability to a broader range of stochastic excitations. In addition, we can further investigate adaptive hyperparameter optimization strategies [38–40] to reduce manual tuning efforts and enhance the model's adaptability and robustness under various conditions.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (Grants No. 11872306 and No. 12402037) and the Key International (Regional) Joint Research Program of the National Natural Science Foundation of China (12120101002).

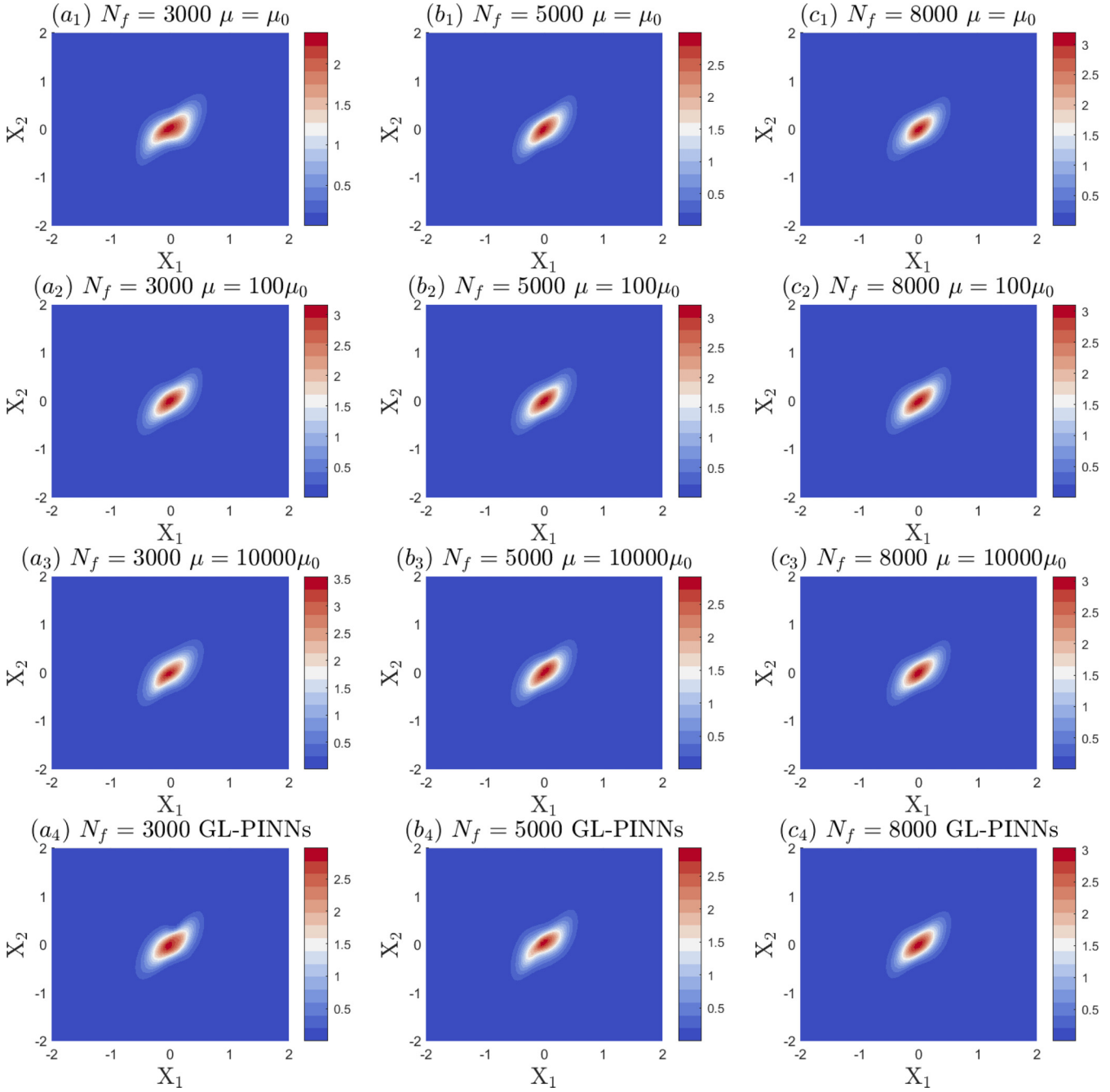

FIG. 25. The joint PDFs predicted by the ATD-GLPINNs magenta(the first three rows) and GL-PINNs (the fourth row) algorithms correspond to the different values of N_f and hyperparameters μ.

DATA AVAILABILITY

The data supporting the findings of this article are openly available [41], and embargo periods may apply.

APPENDIX: THE SPECIFIC EXPRESSION OF MATRIX IN Eq. (9)

$$\begin{aligned}\mathbf{\Phi}_j = &\left[\phi_X\left(\mathbf{x}-\eta_j\left(\mathbf{x},t,r_j^{1,1}\right),t\right),\cdots,\phi_X\left(\mathbf{x}-\eta_j\left(\mathbf{x},t,r_j^{1,N_r}\right),t\right),\cdots,\phi_X\left(\mathbf{x}-\eta_j\left(\mathbf{x},t,r_j^{N_y,1}\right),t\right),\right.\\ &\left.\cdots,\phi_X\left(\mathbf{x}-\eta_j\left(\mathbf{x},t,r_j^{N_y,N_r}\right),t\right)\right]^T,\\ \left|1-\frac{\partial(\eta_j(\mathbf{x},t,\mathbf{r_j}))}{\partial(\mathbf{x})}\right| = &\left[\left|1-\frac{\partial\left(\eta_j\left(\mathbf{x},t,r_j^{1,1}\right)\right)}{\partial(\mathbf{x})}\right|,\cdots,\left|1-\frac{\partial\left(\eta_j\left(\mathbf{x},t,r_j^{1,N_r}\right)\right)}{\partial(\mathbf{x})}\right|,\right.\end{aligned}$$

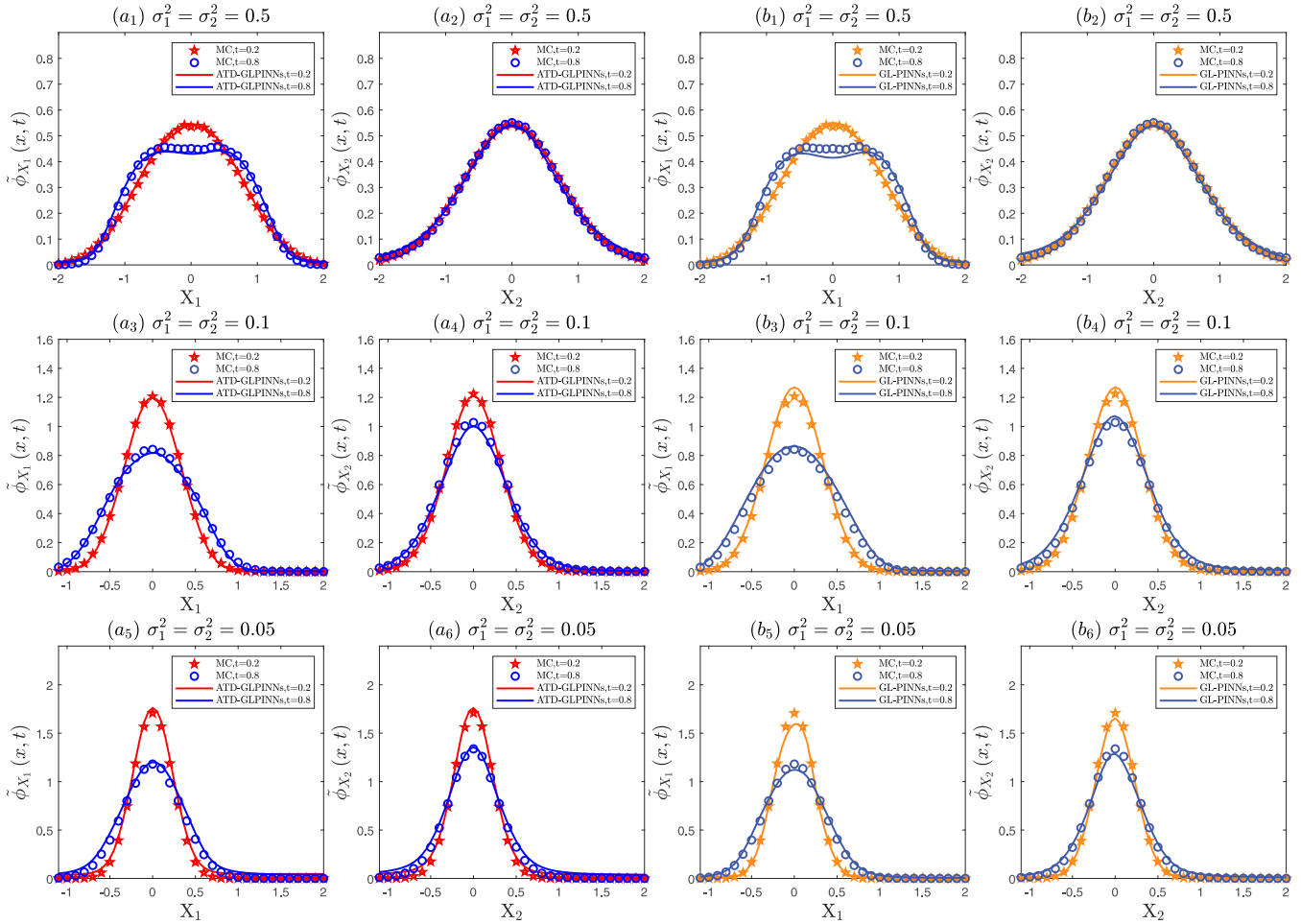

FIG. 26. The marginal PDFs predicted by the ATD-GLPINNs (Columns 1–2) and GL-PINNs (Columns 3–4) algorithms correspond to the different initial conditions at $t = 0.2$ and $t = 0.8$. The colored lines are the predicted solutions from ATD-GLPINNs and GL-PINNs, and the colored symbols denote the MC reference solutions. The symbols σ_1^2 and σ_2^2 represent the variance value of the initial conditions.

$$\times \cdots, \left|1-\frac{\partial\left(\eta_j\left(\mathbf{x},t,r_j^{N_y,1}\right)\right)}{\partial(\mathbf{x})}\right|,\cdots,\left|1-\frac{\partial\left(\eta_j\left(\mathbf{x},t,r_j^{N_y,N_r}\right)\right)}{\partial(\mathbf{x})}\right|\Bigg]^T,$$

$$\boldsymbol{\Phi}_{R_j}=\left[\phi_{R_j}\left(r_j^{1,1}\right),\cdots,\phi_{R_j}\left(r_j^{1,N_r}\right),\cdots,\phi_{R_j}\left(r_j^{N_y,1}\right),\cdots,\phi_{R_j}\left(r_j^{N_y,N_r}\right)\right]^T,$$

$$\boldsymbol{\omega}_j=\left[\omega_j^{1,1},\cdots,\omega_j^{1,N_r},\cdots,\omega_j^{N_y,1},\cdots,\omega_j^{N_y,N_r}\right]^T,$$

where $r_j^{k,l}, k=1,\cdots,N_y, l=1,\cdots,N_r$ represents the l-th GL quadrature node in the k-th GL integral interval of the j-th integral variable.

[1] X. D. Feng, L. Zeng, and T. Zhou, Solving time dependent Fokker-Planck equations via temporal normalizing flow, Commun. Comput. Phys. **32**, 401 (2022).

[2] P. Herrera and M. Sandoval, Structure of the active Fokker-Planck equation, Phys. Rev. E **109**, 014140 (2024).

[3] T. Dupont, S. Giordano, F. Cleri, and R. Blossey, Short-time expansion of one-dimensional Fokker-Planck equations with heterogeneous diffusion, Phys. Rev. E **109**, 064106 (2024).

[4] M. L. Hu, W. R. Zan, W. T. Jia, and J. J. Sun, Stochastic dynamics analysis of quasi-partially integrable Hamiltonian system based on NN-SAM, Int. J. Non Linear Mech. **170**, 104993 (2025).

[5] H. Zhang, Y. Xu, Q. Liu, X. L. Wang, and Y. G. Li, Solving Fokker-Planck equations using deep KD-tree with a small amount of data, Nonlinear Dyn. **108**, 4029 (2022).

[6] W. R. Zan, Y. Xu, J. Kurths, A. V. Chechkin, and R. Metzler, Stochastic dynamics driven by combined Lévy-Gaussian noise: Fractional Fokker-Planck-Kolmogorov equation and solution, J. Phys. A: Math. Theor. **53**, 385001 (2020).

[7] M. Lee, V. Gupta, and L. K. B. Li, Fokker-Planck dynamics of two coupled van der Pol oscillators subjected to stochastic forcing, Phys. Rev. E **110**, 044202 (2024).

[8] A. Ceccato and D. Frezzato, Remarks on the chemical Fokker-Planck and Langevin equations: Nonphysical currents at equilibrium, J. Chem. Phys. **148**, 064114 (2018).

[9] J. A. Devlin and M. R. Tarbutt, Laser cooling and magneto-optical trapping of molecules analyzed using optical Bloch equations and the Fokker-Planck-Kramers equation, Phys. Rev. A **98**, 063415 (2018).

[10] G. Furioli, A. Pulvirenti, E. Terraneo, and G. Toscani, Fokker-Planck equations in the modeling of socio-economic phenomena, Math. Models Methods Appl. Sci. **27**, 115 (2017).

[11] W. T. Jia, Z. Jiao, W. R. Zan, and W. Q. Zhu, Probability density of the solution to nonlinear systems driven by Gaussian and Poisson white noises, Probabilistic Eng. Mech. **77**, 103658 (2024).

[12] W. Q. Zhu and G. Q. Cai, Nonlinear stochastic dynamics: A survey of recent developments, Acta. Mech. Sin. **18**, 551 (2002).

[13] W. T. Jia and W. Q. Zhu, Stochastic averaging of quasi-integrable and non-resonant Hamiltonian systems under combined Gaussian and Poisson white noise excitations, Nonlinear Dyn. **76**, 1271 (2014).

[14] K. L. Song, M. Bonnin, F. L. Traversa, and F. Bonani, A stochastic averaging mathematical framework for design and optimization of nonlinear energy harvesters with several electrical DOFs, Commun. Nonlinear Sci. Numer. Simul. **139**, 108306 (2024).

[15] F. Ureña, L. Gavete, Á. G. Gómez, J. J. Benito, and A. M. Vargas, Non-linear Fokker-Planck equation solved with generalized finite differences in 2D and 3D, Appl. Math. Comput. **368**, 124801 (2020).

[16] V. Scalera, P. Ansalone, S. Perna, C. Serpico, and M. d'Aquino, Numerical solution of the Fokker-Planck equation by spectral collocation and finite-element methods for stochastic magnetization dynamics, IEEE Trans. Magn. **58**, 1 (2022).

[17] X. L. Yue, W. Xu, Y. Xu, and J. Q. Sun, Non-stationary response of MDOF dynamical systems under combined Gaussian and Poisson white noises by the generalized cell mapping method, Probabilistic Eng. Mech. **55**, 102 (2019).

[18] W. R. Zan, W. T. Jia, and Y. Xu, Reliability of dynamical systems with combined Gaussian and Poisson white noise via path integral method, Probabilistic Eng. Mech. **68**, 103252 (2022).

[19] A. Pirrotta and R. Santoro, Probabilistic response of nonlinear systems under combined normal and Poisson white noise via path integral method, Probabilistic Eng. Mech. **26**, 26 (2011).

[20] M. Zanella, Structure preserving stochastic Galerkin methods for Fokker-Planck equations with background interactions, Math Comput. Simul. **168**, 28 (2020).

[21] M. Hosseininia, M. H. Heydari, and M. Razzaghi, Meshless local Petrov-Galerkin method for 2D fractional Fokker-Planck equation involved with the ABC fractional derivative, Comput. Math Appl. **125**, 176 (2022).

[22] M. Raissi, P. Perdikaris, and G. E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. **378**, 686 (2019).

[23] J. Yu, L. Lu, X. H. Meng, and G. E. Karniadakis, Gradient-enhanced physics-informed neural networks for forward and inverse PDE, Comput. Methods Appl. Mech. Eng. **393**, 114823 (2022).

[24] S. Mishra and R. Molinaro, Estimates on the generalization error of physics-informed neural networks for approximating PDEs, IMA. J. Numer. Anal. **43**, 1 (2023).

[25] L. Yang, X. H. Meng, and G. E. Karniadakis, B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data, J. Comput. Phys. **425**, 109913 (2021).

[26] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, Automatic differentiation in machine learning: a survey, J. Mach. Learn Res. **18**, 1 (2018).

[27] Y. Xu, H. Zhang, Y. G. Li, K. Zhou, Q. Liu, and J. Kurths, Solving Fokker-Planck equation using deep learning, Chaos. **30**, 013133 (2020).

[28] S. B. Jiang and X. F. Li, Solving the non-local Fokker-Planck equations by deep learning, Chaos. **33**, 043107 (2023).

[29] Y. Zhang, Z. R. Jin, L. Wang, K. X. Zheng, and W. T. Jia, Stochastic dynamics of aircraft ground taxiing via improved physics-informed neural networks, Nonlinear Dyn. **112**, 3163 (2024).

[30] X. L. Chen, L. Yang, J. Q. Duan, and G. E. Karniadakis, Solving inverse stochastic problems from discrete particle observations using the Fokker-Planck equation and physics-informed neural networks, SIAM J. Sci. Comput. **43**, B811 (2021).

[31] W. T. Jia, M. L. Hu, W. R. Zan, and F. Ni, Data-driven methods for the inverse problem of suspension system excited by jump and diffusion stochastic track excitation, Int. J. Non Linear Mech. **166**, 104819 (2024).

[32] W. T. Jia, X. T. Feng, M. L. Hao, and S. C. Ma, Deep neural network method to predict the dynamical system response under random excitation of combined Gaussian and Poisson white noises, Chaos Soliton Fract. **185**, 115134 (2024).

[33] X. H. Meng, Z. Li, D. K. Zhang, and G. E. Karniadakis, PPINN: Parareal physics-informed neural network for time-dependent PDEs, Comput. Methods Appl. Mech. Eng. **370**, 113250 (2020).

[34] A. D. Jagtap and G. E. Karniadakis, Extended physics-informed neural networks (XPINNs): A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations, Commun. Comput. Phys. **28**, 2002 (2020).

[35] R. Mattey and S. Ghosh, A novel sequential method to train physics informed neural networks for Allen Cahn and Cahn Hilliard equations, Comput. Methods Appl. Mech. Eng. **390**, 114474 (2022).

[36] J. C. Yang, X. Q. Liu, Y. Diao, X. Chen, and H. K. Hu, Adaptive task decomposition physics-informed neural networks, Comput. Methods Appl. Mech. Eng. **418**, 116561 (2024).

[37] F. B. Hanson, *Applied Stochastic Processes and Control for Jump-Diffusions: Modeling, Analysis and Computation* (PA: Society for Industrial and Applied Mathematics, Philadelphia, 2007).

[38] J. Ollar, C. Mortished, R. Jones, J. Sienz, and V. Toropov, Gradient based hyper-parameter optimisation for well conditioned kriging metamodels, Struct Multidiscip Optim. **55**, 2029 (2017).

[39] S. Khan, A. Rizwan, A. N. Khan, M. Ali, R. Ahmed, and D. Kim, A multi-perspective revisit to the optimization methods of neural architecture search and hyper-parameter optimization for non-federated and federated learning environments, Comput. Electr. Eng. **110**, 108867 (2023).

[40] Y. F. Xia, C. Z. Liu, Y. Y. Li, and N. N. Liu, A boosted decision tree approach using Bayesian hyper-parameter optimization for credit scoring, Expert Syst. Appl. **78**, 225 (2017).

[41] W. T. Jia, X. T. Feng, Y. F. Zhao, and W. R. Zan, Data for "Dynamical system response under Gaussian and Poisson white noises solved by deep neural network method with adaptive task decomposition and progressive learning strategy", https://1drv.ms/f/c/c7299d077d1d48d5/EhzQtllixOpLrNZpSucIy4IB9E1fuxWa9sWjdSKZLDNtsg?e=mxvFU9 (accessed April 2, 2025).