Electronic Research Archive

https://www.aimspress.com/journal/era

ERA, 34(6): 4158–4171.
DOI: 10.3934/era.2026186
Received: 20 March 2026
Revised: 08 May 2026
Accepted: 11 May 2026
Published: 18 May 2026

Research article

Physics-informed stochastic models for theme park ride waiting times

William Wang[1], **Lingju Kong**[2] **and Min Wang**[3,*]

[1] International School of Qingdao, Qingdao 266000, China

[2] Department of Mathematics, University of Tennessee at Chattanooga, Chattanooga, TN 37403, USA

[3] Department of Mathematics, Kennesaw State University, Marietta, GA 30060, USA

* **Correspondence:** Email: min.wang@kennesaw.edu.

Abstract: A stochastic model for theme park ride waiting times is developed by modeling the waiting time as a continuous-time, discrete-state Markov process with state-dependent, time-varying transition rates. These transition rates are interpreted as a control term acting on the waiting-time process, allowing the model calibration task to be formulated as a data-driven optimal control problem. To solve this problem efficiently, we construct a physics-informed neural network (PINN) that embeds the Kolmogorov forward equation solver into its architecture. Under mild assumptions, we prove the existence of an optimal control, providing theoretical support for the learning procedure. Numerical simulations demonstrate the effectiveness of the PINN-based solution. The proposed framework provides an interpretable, physically consistent, and data-driven approach for modeling and forecasting ride waiting times.

Keywords: Markov chain; Kolmogorov equations; stochastic process; PINN; optimal control

1. Introduction

Theme parks are highly dynamic, nonstationary systems in which visitor flows, ride capacities, operational schedules, and environmental factors interact in stochastic and time-dependent ways [1, 2]. Visitors spend a substantial amount of their day waiting in line. The long waiting times significantly impact both visitor satisfaction and overall park operations. Accurate modeling of ride waiting times therefore plays an important role in operational planning, real-time decision making, and system optimization.

Two traditional approaches have been widely used to analyze waiting-time and queueing phenomena: classical queueing theory and discrete-event simulation. Although both frameworks provide valuable insight, each faces important limitations when applied to theme-park operations. Queueing theory

(e.g., birth–death processes and $M/M/1$ systems) offers analytical tractability and asymptotic formulas [3–5] but typically relies on assumptions such as stationary arrival rates, exponential service times, and independent flows. These assumptions rarely hold in theme parks, where visitor arrivals vary sharply throughout the day, and waiting times are restricted by physical and operational limits, see for example, Figure 1. On the other hand, discrete-event simulation can represent fine-grained operational details-such as downtime and fluctuating capacity. However, simulation models are computationally expensive, difficult to calibrate, and often lack analytical interpretability [6, 7].

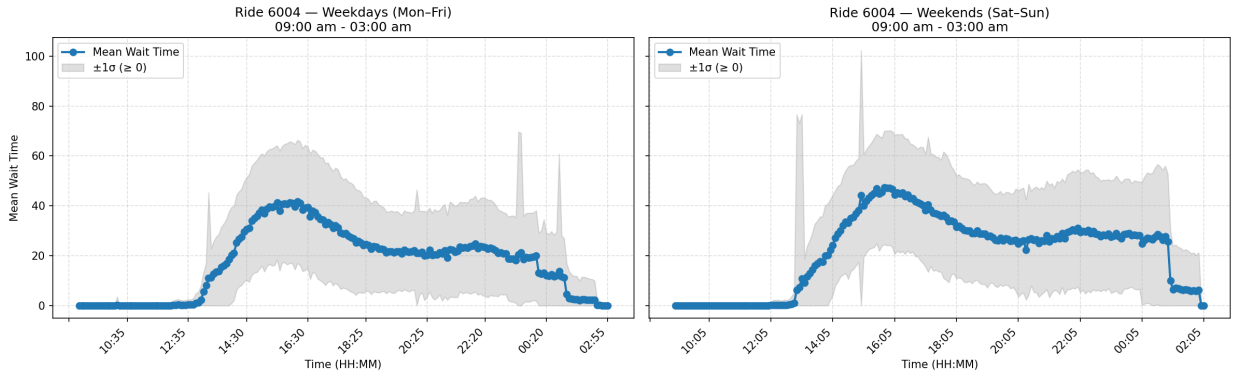

Figure 1. Statistical information for The Incredible Hulk Coaster at Universal Islands of Adventure.

The limitations of these traditional approaches motivate the development of stochastic models that incorporate time dependence, physical constraints, and data-driven flexibility. However, fitting stochastic models to real waiting-time data remains challenging. Existing calibration strategies typically rely on exhaustive parameter searches, simplifying the form of the model, or local optimization around deterministic approximations [3, 6, 8–11], and these methods become increasingly difficult to apply as the dimensionality of the system grows. Recent advances in machine learning (ML), particularly in physics-informed neural networks (PINNs), provide an opportunity to integrate mechanistic modeling with data-driven function approximation. PINNs embed known physical or stochastic laws directly in the learning process. This improves interpretability and reduces reliance on black-box models. The reader is referred to [12–16] and the references therein for recent advances and background on PINNs and ML in general.

In this work, we develop a data-driven stochastic model for ride waiting times within a single operating day. The waiting time is modeled as a continuous-time, discrete-state, nonhomogeneous Markov process [17]. The transition rates are interpreted as a control term acting on the waiting-time process, which allows us to formulate the estimation problem as an optimal control problem. We then use a PINN framework to learn these time-varying transition rates directly from the data. The existence of an optimal control is also proved under mild assumptions, thereby providing theoretical support for the learning procedure.

A practical contribution of this work concerns the data collection itself. Many theme parks publish real-time waiting-time estimates through public-facing websites or mobile applications. However, these values are continuously updated and are therefore not archived or directly reusable. To overcome this limitation, we systematically recorded these public data streams from *Queue Times* [18] over a period of twenty-three months for multiple parks across the USA, creating a large dataset of historical

waiting-time information. This dataset enables detailed empirical analysis of daily patterns, such as differences between weekdays and weekends (Figure 1), and forms the foundation for calibrating the proposed data-driven stochastic model.

The remainder of the paper is organized as follows. In Section 2, we introduce the stochastic model and derive the Kolmogorov forward and backward equations. Section 3 presents the optimal control formulation and the PINN architecture used to implement the stochastic model. The training procedure and two applications of the trained PINN model are also presented there. A summary is given in Section 4.

2. Mathematical models

To simplify the modeling framework and reflect the structure of the collected data, we assume that rides operate independently and model the waiting time of a single ride as a one-dimensional, continuous-time, discrete-state Markov process X_t, $t \geq 0$. For practical purposes, we consider a finite time horizon $[0, T]$, where T is the daily park operation time. For any $t \in [0, T]$, the process takes values in the discrete state space $S = \{0, \lambda, 2\lambda, \ldots, \bar{X}\}$, where $\bar{X}$ is the maximum allowable waiting time during the day and is assumed to be a multiple of λ. Because the waiting time cannot exceed the total operating time, it follows that $\bar{X} \leq T$. The state $\bar{X} \in S$ represents waiting times at least $\bar{X}$, corresponding to the interval $[\bar{X}, T]$. Each state $k\lambda \in S$ with $0 \leq k\lambda < \bar{X}$ corresponds to $[k\lambda, (k+1)\lambda)$. This discretization matches the structure of the collected data.

The stochastic model is developed following a standard modeling procedure [8, 9, 17, 19]. For any $t \in [0, T]$ and $\Delta t > 0$, let $\Delta X_{(t,x)}$ denote the change of X_t over $[t, t + \Delta t]$ given $X_t = x \in S$. We assume Δt is sufficiently small so that for any $(t, x) \in [0, T] \times S$, the possible values of $\Delta X_{(t,x)}$ are $\{-\lambda, 0, \lambda\}$ with probabilities given in Table 1, where $q_1(\cdot, \bar{X}) \equiv 0$ and $q_2(\cdot, 0) \equiv 0$.

Table 1. Possible values of ΔX with the corresponding probabilities.

Change	Probability
$\Delta X_{(t,x)} = \lambda$	$q_1(t, x)\Delta t$
$\Delta X_{(t,x)} = -\lambda$	$q_2(t, x)\Delta t$
$\Delta X_{(t,x)} = 0$	$1 - [q_1(t, x) + q_2(t, x)]\Delta t$

Let $n = |S|$ be the number of elements in S, and let $\mathbf{P}(s, t) = [p_{ij}(s, t)]_{n\times n}$ be the transition probability matrix, with $p_{ij}(s, t) = \mathbb{P}(X_t = (j-1)\lambda \mid X_s = (i-1)\lambda)$, $i, j \in [1, n]_{\mathbb{Z}}$. Then, for any $s < t_1 < t$,

$$p_{ij}(s, t) = \sum_{k=1}^{n} p_{ik}(s, t_1) p_{kj}(t_1, t),$$

or equivalently, $\mathbf{P}(s, t) = \mathbf{P}(s, t_1)\mathbf{P}(t_1, t)$. Hence, for any $t > s$ and $\Delta t > 0$,

$$\mathbf{P}(s, t + \Delta t) - \mathbf{P}(s, t) = \mathbf{P}(s, t)[\mathbf{P}(t, t + \Delta t) - I].$$

By Table 1, for any $i, j \in [1, n]_{\mathbb{Z}}$,

$$p_{ij}(t, t+\Delta t) = \begin{cases} q_1(t, (i-1)\lambda)\,\Delta t, & j = i+1, \\ q_2(t, (i-1)\lambda)\,\Delta t, & j = i-1, \\ 1 - [q_1(t, (i-1)\lambda) + q_2(t, (i-1)\lambda)]\Delta t, & j = i, \\ 0, & \text{otherwise.} \end{cases}$$

Thus,

$$\mathbf{P}(s, t+\Delta t) - \mathbf{P}(s, t) = \mathbf{P}(s, t)Q(t)\,\Delta t, \tag{2.1}$$

where $Q(t) = [q_{ij}(t)]_{n\times n}$ is an $n \times n$ tridiagonal matrix with

$$q_{ij}(t) = \begin{cases} q_1(t, (i-1)\lambda), & j = i+1, \\ q_2(t, (i-1)\lambda), & j = i-1, \\ -q_1(t, (i-1)\lambda) - q_2(t, (i-1)\lambda), & j = i, \\ 0, & \text{otherwise.} \end{cases} \tag{2.2}$$

Dividing (2.1) by Δt and letting $\Delta t \to 0$ yields the Kolmogorov forward equation

$$\frac{\partial}{\partial t}\mathbf{P}(s, t) = \mathbf{P}(s, t)Q(t), \qquad \mathbf{P}(s, s) = I,$$

and similarly the backward equation

$$\frac{\partial}{\partial s}\mathbf{P}(s, t) = -Q(s)\mathbf{P}(s, t), \qquad \mathbf{P}(t, t) = I.$$

Hence, X_t is a nonhomogeneous Q-process with generator $Q(t)$.

Let $P(t) = [p_0(t), p_\lambda(t), \ldots, p_{\bar{X}}(t)]$ be $1 \times n$ with $p_x(t) = \mathbb{P}(X_t = x)$. Then

$$P(t) = P(t_0)\mathbf{P}(t_0, t),$$

and $P(t)$ satisfies the forward equation

$$\frac{d}{dt}P(t) = P(t)Q(t), \qquad P(t_0) = P_0, \quad t > t_0. \tag{2.3}$$

Model (2.3) will serve as the main dynamical system in the next section. Before proceeding to the next section, we present two expectation identities that will be useful in the model applications.

Proposition 2.1. *Let* $P(t) = [p_0(t), \ldots, p_{\bar{X}}(t)]$, $Q(t)$ *be given by* (2.2), *and* $F(\boldsymbol{X}) = [f(0), f(\lambda), \ldots, f(\bar{X})]$. *Then,*

$$\frac{d}{dt}\mathbb{E}[f(X_t)] = \mathbb{E}[(Q(t)F^T)(X_t)].$$

Proof. By (2.3), $\dfrac{d}{dt}\mathbb{E}[f(X_t)] = \dfrac{d}{dt}(P(t)F^T) = P(t)Q(t)F^T = \mathbb{E}[(Q(t)F^T)(X_t)]$.

Proposition 2.2. *Let the first and second moments be denoted by*

$$m(t) = \mathbb{E}[X_t] = \sum_{i \in S} i\, p_i(t), \qquad s(t) = \mathbb{E}[X_t^2] = \sum_{i \in S} i^2\, p_i(t).$$

With q_1, q_2 as above, the moments satisfy

$$\frac{d}{dt} m(t) = \lambda\, \mathbb{E}\big[q_1(t, X_t) - q_2(t, X_t)\big], \tag{2.4}$$

$$\frac{d}{dt} s(t) = \lambda\, \mathbb{E}\big[(2X_t + \lambda) q_1(t, X_t) - (2X_t - \lambda) q_2(t, X_t)\big]. \tag{2.5}$$

Proof. This result follows immediately from Proposition 2.1 by taking $f(x) = x$ and $f(x) = x^2$, respectively.

These identities explicitly reveal how the transition rates govern the evolution of the expectations, particularly the mean and variance, of the waiting-time process. In Section 3, we address the inverse problem of estimating q_1 and q_2 from real data, formulate the task within a data-driven optimal control framework, and present the PINN architecture used for training.

3. PINN and optimal control formulation

In this section, we develop a PINN implementation of Model (2.3), interpreting the transition rates (q_1, q_2) as controls and formulating the PINN training procedure as a data-driven optimal control problem that minimizes the loss function.

3.1. Optimal control formulation

We seek to estimate the transition rates $q_1(t, x)$ and $q_2(t, x)$ from collected waiting-time data. These transition rates are interpreted as controls acting on the waiting-time process, and they influence the dynamics solely through the generator matrix $Q(t)$ in (2.2) and the Kolmogorov forward equation (2.3).

We define the admissible control set

$$\mathcal{U} = \left\{ u = (q_1, q_2) \;\middle|\; \begin{array}{l} q_i \in L^\infty([0, T] \times S),\ 0 \le q_i(t, x) \le \bar{q},\ i = 1, 2, \\ q_1(t, \bar{X}) = 0,\ \ q_2(t, 0) = 0 \end{array} \right\}$$

for a fixed constant $\bar{q} > 0$. For an admissible control $u = (q_1, q_2) \in \mathcal{U}$, let $P^u(t)$ be the solution of

$$\frac{d}{dt} P^u(t) = P^u(t) Q^u(t), \qquad P^u(t_0) = P_0, \quad t > t_0, \tag{3.1}$$

where $Q^u(t)$ is obtained from (2.2) using q_1, q_2 from u.

Let $\{t_i\}_{i=1}^m \subset (0, T]$ denote the observation times. We consider a general loss functional of the form

$$J(u) = \Phi(P^u(t_1), \ldots, P^u(t_m)), \tag{3.2}$$

where

$$\Phi : \mathbb{R}^{m \times n} \longrightarrow \mathbb{R}$$

is assumed to be continuous. The data-driven optimal control problem is therefore to find $u^* \in \mathcal{U}$ that minimizes $J(u)$ defined by (3.2) subject to (3.1).

We now establish an existence result for the above optimal control problem.

Theorem 3.1. *Assume that the loss functional $J(u)$ is of the form* (3.2) *with Φ continuous. Then there exists at least one control $u^* \in \mathcal{U}$ such that*

$$J(u^*) = \min_{u \in \mathcal{U}} J(u),$$

subject to the state equation (3.1).

Proof. The proof follows a standard approach for proving the existence of minimizers in optimal control problems with controls in L^∞ spaces. We outline the steps below.

Step 1 (Compactness). The admissible set $\mathcal{U}$ is bounded in $L^\infty([0,T] \times S)^2$ and closed in the weak* sense, and is hence weak*-compact by the Banach–Alaoglu theorem.

Step 2 (Well-posedness). For each $u \in \mathcal{U}$, the generator $Q^u(t)$ is measurable and uniformly bounded. Thus, Model (3.1) has a unique solution $P^u \in C([0,T]; \mathbb{R}^n)$ by standard Carathéodory ordinary differential equation (ODE) theory.

Step 3 (Continuity of the control-to-state map). If $u_k \overset{*}{\rightharpoonup} u$ in L^∞, then $Q^{u_k}(t) \overset{*}{\rightharpoonup} Q^u(t)$ in $L^\infty([0,T] \times S)^{n \times n}$. From linearity,

$$\frac{d}{dt}(P^{u_k} - P^u) = P^{u_k} Q^{u_k} - P^u Q^u.$$

Uniform boundedness of Q^{u_k} and Grönwall's inequality imply $\|P^{u_k} - P^u\|_{C([0,T])} \to 0$.

Step 4 (Lower semicontinuity of the loss). By continuity of Φ and the uniform convergence $P^{u_k}(t_i) \to P^u(t_i)$, we have $J(u_k) \to J(u)$.

Step 5 (Existence). By combining Steps 1–4, we may now complete the proof. Let $\{u_k\}$ be a minimizing sequence. By weak* compactness, a subsequence converges to some $u^* \in \mathcal{U}$ in the weak* sense. Lower semicontinuity yields

$$J(u^*) \leq \liminf_{k \to \infty} J(u_k),$$

so u^* is optimal.

Remark 3.1.

(1) The conclusion of Theorem 3.1 holds for any continuous data-driven loss functional and is therefore independent of both the specific neural network (NN) implementation of u and the optimization algorithm used during training.

(2) Typical examples of the loss functional include weighted least squares, smoothed absolute deviations, log-type losses, and other standard data-fitting objectives used in PINN training.

3.2. PINN model

The optimal control problem formulated in Section 3.1 is solved by developing a PINN model. This idea has been used in [19, 20]. The PINN model uses the initial waiting time x_0 together with the model parameters, $(T, \bar{X}, \lambda)$ as the input, and it outputs the predicted probability distribution $P(t \mid x_0) = \mathbb{P}(X_t \mid X_0 = x_0)$, $0 < t \leq T$. Two sub-NNs, namely q_1-net and q_2-net, are developed to approximate the control functions q_1 and q_2 in Model (3.1), respectively. Then, an ODE-Solver layer is used to solve Model (3.1) and output $P(t \mid x_0)$. The PINN architecture is shown in Figure 2.

The roles of the blocks shown in Figure 2 are summarized below:

- The t block generates a discrete time interval $[0, T]_{\mathbb{Z}} := \{0, \Delta t, 2\Delta t, \cdots, T\}$ needed for other blocks based on the step size Δt to discretize the time interval $[0, T]$.
- The S block generates the state set $S = \{0, \lambda, 2\lambda, \ldots, \bar{X}\}$ needed for other blocks based on $\bar{X}$ and λ.
- The P_0 block converts the input x_0 to the initial distribution $P(t_0)$.
- The q_1-net and q_2-net approximate the transition rates $q_1(t, x)$ and $q_2(t, x)$, respectively. Both use fully connected neural network architectures. This structure allows the rates to vary with time and state. It enables the model to capture complex patterns observed in real data. As a result, the proposed approach substantially extends classical stochastic models that assume constant or overly simple transition rates.
- The ODE-Solver block numerically solves Model (3.1) on the grid $[0, T]_{\mathbb{Z}} \times S$ using the implicit backward Euler method due to its stability for approximating ODE solutions. The reader is referred to [21] for the details.

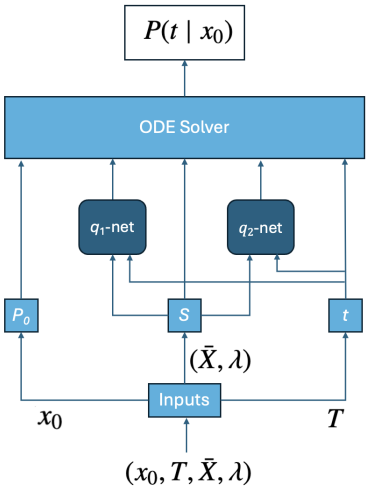

Figure 2. The PINN architecture.

The forward propagation process of the PINN is to solve Model (3.1) based on the input. The algorithm is summarized as Algorithm 1 below.

Algorithm 1 Forward propagation for (3.1)

1: Initialize T, Δt, grid $[0, T]_{\mathbb{Z}}$, initial distribution P_0, state set S, and control networks for $u = (q_1, q_2)$.

2: For $t = \Delta t, 2\Delta t, \ldots, T$:

- Evaluate (q_1, q_2) on S and form $Q^u(t)$ via (2.2).
- Advance $P^u(t - \Delta t) \to P^u(t)$ using an ODE solver for (3.1).

3: Return $\{P^u(t_i)\}_{i=1}^m$.

3.3. Training process

The PINN model is trained using real data. We have collected 23 months of waiting-time information for rides in several theme parks across the United States. For each ride, the data are organized as multiple daily time series, which allows us to compute statistical summaries and empirical distributions of waiting times throughout a typical operating day. Exploratory data analysis (EDA) further reveals that weekday and weekend patterns differ noticeably; therefore, they should be modeled separately. See Figure 1 for an illustrative example.

The PINN is trained using the loss function

$$\mathcal{L}(u) = w_m \left\| m^u(t) - m^{\text{emp}}(t) \right\|_2^2 + w_v \left\| v^u(t) - v^{\text{emp}}(t) \right\|_2^2 + w_{\text{dist}} \left\| P^u(t) - P^{\text{emp}}(t) \right\|_2^2, \tag{3.3}$$

which combines three components: (i) a *mean* term aligning the predicted mean waiting time $m^u(t)$ with the empirical mean $m^{\text{emp}}(t)$, (ii) a *variance* term aligning the predicted variance $v^u(t)$ with the empirical variance $v^{\text{emp}}(t)$, and (iii) a *distribution* term measuring the mean squared error between the predicted distribution $P^u(t)$ and the empirical distribution $P^{\text{emp}}(t)$. The non-negative weights w_m, w_v, w_{dist} determine the relative importance of the three terms.

Remark 3.2.

(1) The training process can be interpreted as approximating the optimal control u^* that minimizes the loss (3.3).

(2) The PINN framework enables the use of state-of-the-art optimization techniques and modern machine-learning hardware.

(3) The existence of an optimal control at the function level (Theorem 3.1) provides theoretical justification for the training process.

3.4. Simulations

We conclude this section by presenting some representative simulation results. In our experiment, the q_1-net and q_2-net are implemented as fully connected feedforward networks with 10 hidden layers and 5 neurons per layer. The activation function in each hidden layer is $\arctan(x)$, and the output layer uses the activation $f(x) = \sqrt{|x|}$ to ensure the smoothness and non-negativity of the learned transition rates. The model parameters are chosen as $\lambda = 5$ minutes and $\bar{X} = 50$ minutes to match the collected data.

The loss weights in (3.3) are taken to be $w_m = 20$ and $w_v = w_{\text{dist}} = 1$, placing greater emphasis on matching the empirical mean while still enforcing agreement in both variance and distributional shape. The optimization problem is solved using particle swarm optimization (PSO) with a swarm size of 64, executed on a desktop PC running Ubuntu 22.04 LTS and equipped with an Intel Core i5-11400F CPU and 16 GB of RAM.

The model is trained using weekday data from the Incredible Hulk Coaster at Universal Islands of Adventure, collected between October 2023 and August 2025. Figure 3 displays the statistical summaries and empirical distributions produced by the trained PINN model. The close agreement between the model-generated statistics and the empirical data demonstrates the feasibility of the proposed PINN-based approach and its ability to learn realistic, time-varying transition rates that accurately capture both the daily dynamics and variability of ride waiting times.

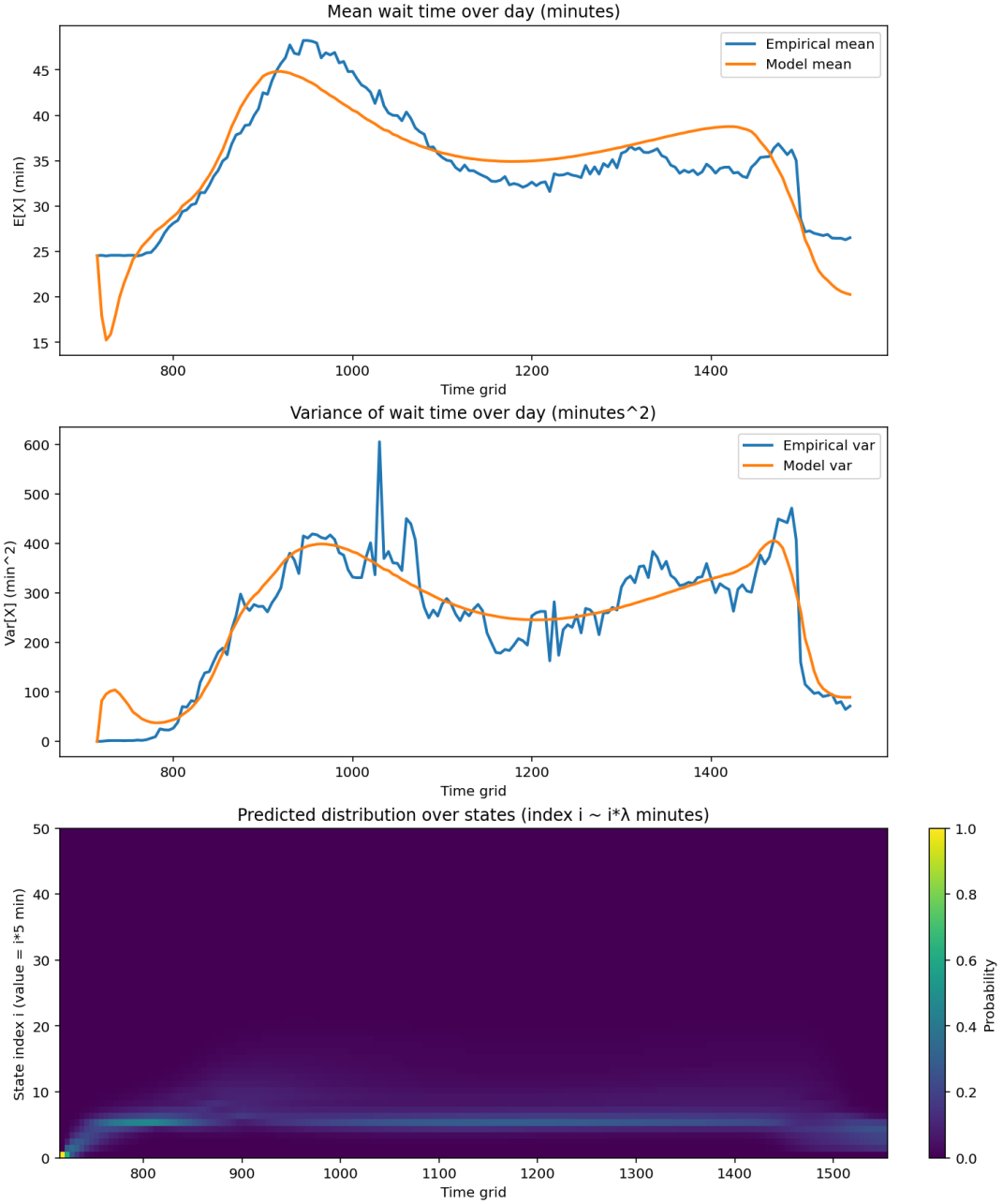

Figure 3. Statistical summaries and empirical distributions generated by the trained PINN model.

Remark 3.3. The numerical experiments were conducted on a standard desktop computer. Particle swarm optimization (PSO) was selected as the training algorithm due to the limited computational resources available; however, gradient-based optimization methods may be employed when high-performance computing (HPC) and/or GPU resources are accessible. In addition, the backward Euler solver utilizes a relatively large time-step $\Delta t = 1.25$ minutes in the present experiments to reduce computational cost. Smaller time-steps may further improve accuracy when additional computational resources permit.

The trained PINN model can be applied to investigate various properties of the stochastic process beyond one-day simulations. The following example illustrates how the trained model can be used to forecast the distribution of waiting times when the system is initialized at an arbitrary time and state.

Example 3.1. Consider the solution $P^u(t)$ of Model (3.1) with initial conditions $t_0 = 1000$ minutes (4:40 pm) and $X_{t_0} = 35$ minutes. Using the trained PINN, we compute the future distribution $P^u(t)$ between 4:40 p.m. and 6:40 p.m. The predicted mean $m(t)$ together with the one- and two-standard-deviation intervals $[m(t) - \sigma, m(t) + \sigma]$ and $[m(t) - 2\sigma, m(t) + 2\sigma]$, and the predicted probability distribution are shown in Figure 4. This type of information enables short-term forecasting of waiting times and can serve as a decision-support tool for operational planning.

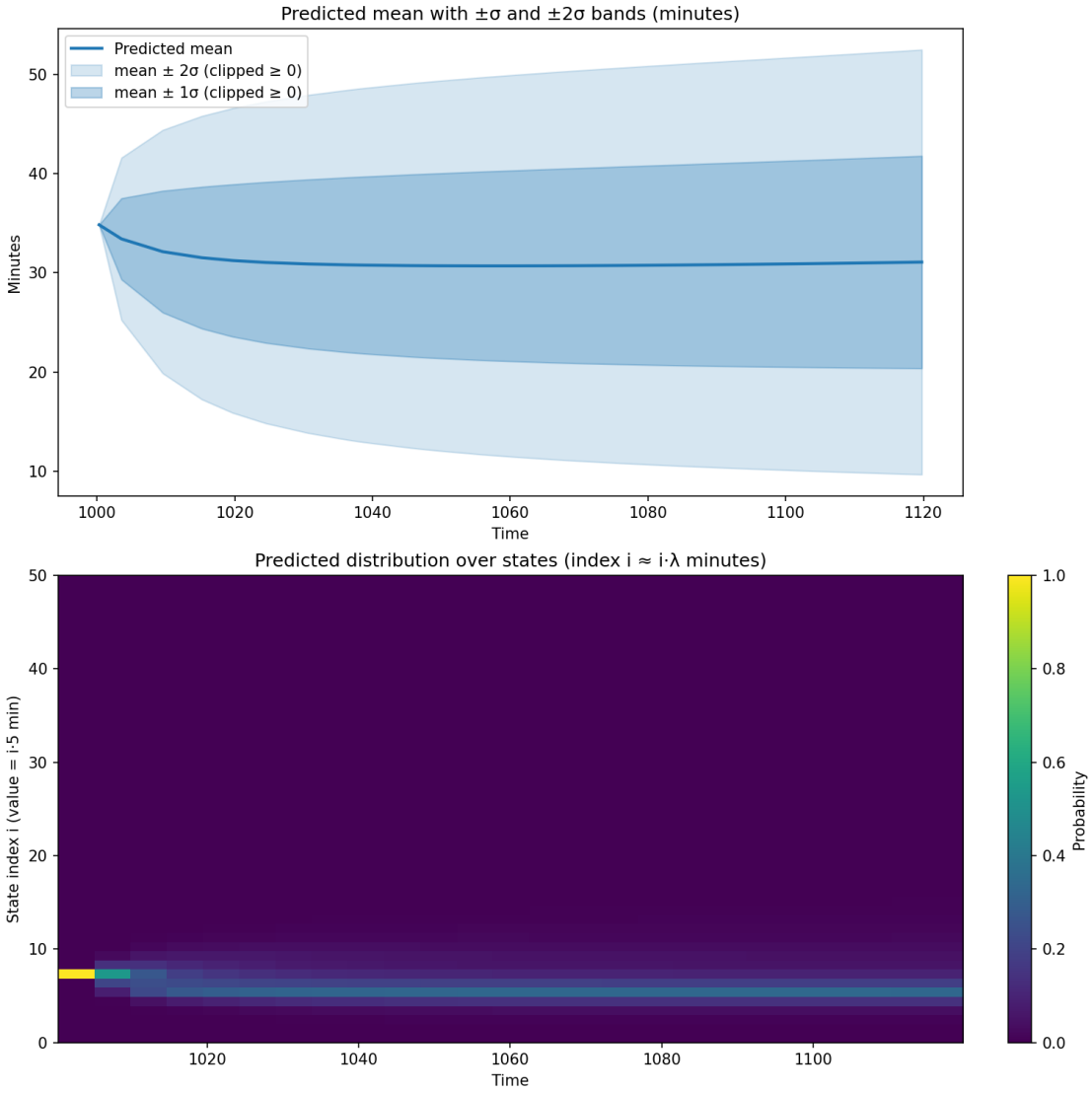

Figure 4. Forecasted mean waiting time with confidence intervals (above) and probability distribution (below) for the scenario $t_0 = 4{:}40$ p.m. and $X_{t_0} = 35$ minutes.

The next example illustrates an application of the Kolmogorov backward equation using the trained transition rates (q_1, q_2).

Example 3.2. In this example, we demonstrate how to estimate the probability that the waiting time at a terminal time t_T falls below a prescribed threshold by solving the Kolmogorov backward equation with the trained transition rates (q_1, q_2).

Let X_t denote the waiting time at time t, and consider the terminal time $t_T = 1200$ minutes (8:00 p.m.). For a threshold $x_T = 10$ minutes, we define the backward value function

$$v(s, x) = \mathbb{E}^{s,x}\left[\mathbf{1}_{\{X_{t_T} \le x_T\}}\right] = \mathbb{P}(X_{t_T} \le x_T \mid X_s = x),$$

which represents the probability that the waiting time at the terminal time t_T will be no more than x_T minutes, conditional on the process being in state $X_s = x$ at an earlier time $s < t_T$.

Then, for the Q-process induced by the learned PINN rates, the vector-valued function $\mathbf{v}(s) = (v(s, 0), v(s, \lambda), \ldots, v(s, \bar{X})) \in \mathbb{R}^{n\times 1}$ satisfies the Kolmogorov backward equation

$$\frac{d}{ds}\mathbf{v}(s) = -Q^{u}(s)\mathbf{v}(s), \ s < t_T, \quad \text{with the terminal condition } v(t_T, x) = \mathbf{1}_{\{x\le x_T\}}, \ x \in S. \tag{3.4}$$

Solving this linear ODE backward from t_T to t_s yields the complete probability profile $v(s, x)$ over the window $[t_s, t_T] \times S$. For additional background on the Kolmogorov backward equation, see, for example, [17].

For illustration, we choose a start time of $t_s = 990$ minutes (4:30 p.m.) and compute the backward probability distribution $v(t_s, x)$ for all states $x \in S$. The backward ODE system (3.4) is solved by a separate ODE solver using the explicit Euler's method with the trained transition rates (q_1, q_2). Figure 5 shows the slice $v(t_s, x)$ as a function of x. These results quantify, for each possible state x at time t_s, the likelihood that the waiting time at the terminal time t_T will not exceed the threshold x_T. Scenario-based probabilities such as $v(t_s, x_0)$ can support real-time operational decisions, such as guest communication or late-afternoon staffing adjustments.

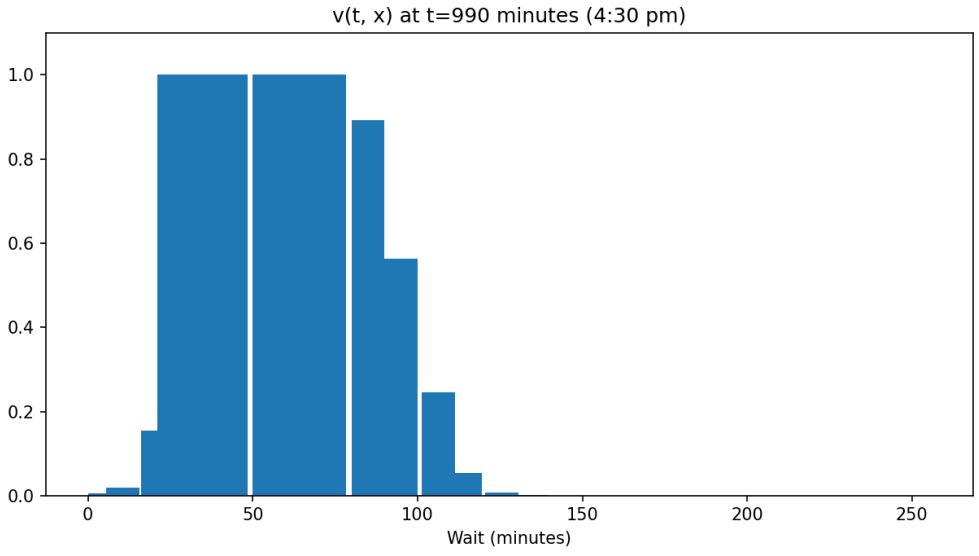

Figure 5. Slice $v(t_s, x)$ versus x (minutes) at $t_s = 4{:}30$ p.m., showing the conditional probability that the waiting time at the terminal time $t_T = 1200$ is at most $x_T = 10$ minutes given X_{t_s} for all possible initial states.

4. Conclusions

In this work, we developed a stochastic modeling and learning framework for characterizing ride waiting times in theme parks. By formulating the waiting time as a continuous-time, discrete-state nonhomogeneous Markov process with state- and time-dependent transition rates, we obtained a flexible probabilistic model capable of capturing daily variations in system dynamics. Interpreting the transition rates as a control allowed us to recast the inverse problem of estimating these rates from data as a data-driven optimal control problem. Within this framework, we established the existence of an optimal control that minimizes the prescribed loss functional, providing a rigorous mathematical foundation for the learning procedure.

To approximate the optimal control, we introduced a PINN architecture that combines NN parameterizations of the transition rates with a numerical solver for the Kolmogorov forward equation. This design preserves model interpretability, leverages classical numerical structure, and allows the use of modern ML optimization tools. The model was trained using 23 months of real waiting-time data, and the loss function incorporated empirical information at the levels of mean, variance, and full probability distribution.

Simulations using the learned controls demonstrated close agreement between the model-generated statistics and the empirical data, illustrating the feasibility and effectiveness of the proposed PINN-based approach. The results show that the model captures key structural features of daily waiting-time dynamics while maintaining computational tractability and interpretability. Overall, the present work demonstrates that combining stochastic modeling, optimal control theory, and physics-informed machine learning provides a powerful and interpretable framework for analyzing large-scale, real-world waiting-time systems.

For simplicity, the present study focuses on modeling the waiting-time dynamics of a single ride and treats rides as independent. In practice, however, waiting times across multiple rides are often coupled through shared visitor flows, operational constraints, and guest decision-making. Extending the proposed framework to multiride systems with interacting stochastic dynamics would be a natural and important direction for future research. Such extensions could enable the analysis of systemwide effects, sensitivity with respect to control policies, and coordinated operational strategies, and may lead to additional analytical insights and richer numerical results.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Conflict of interest

The authors declare there is no conflicts of interest.

References

1. M. Steptoe, R. Krüger, R. Garcia, X. Liang, R. Maciejewski, A visual analytics framework for exploring theme park dynamics, *ACM Trans. Interact. Intell. Syst.*, **8** (2018), 4. https://doi.org/10.1145/3162076

2. G. Hernandez-Maskivker, G. Ryan, M. Pàmies, Waiting times at theme parks: How managers interpret waiting, *Tourismos*, **11** (2016), 158–184. https://doi.org/10.26215/tourismos.v11i4.498
3. D. Gross, J. F. Shortle, J. M. Thompson, C. M. Harris, *Fundamentals of Queueing Theory*, Wiley, 2008.
4. X. Chen, D. Worthington, Staffing of time-varying queues using a geometric discrete time modelling approach, *Ann. Oper. Res.*, **252** (2017), 63–84. https://doi.org/10.1007/s10479-015-2058-3
5. J. Li, Q. Li, Analysis of queue management in theme parks introducing the fast pass system, *Heliyon*, **9** (2023), https://doi.org/10.1016/j.heliyon.2023.e18001.
6. A. Law, *Simulation Modeling and Analysis*, 5th edition, McGraw Hill, 2015.
7. W. H. Fun, E. H. Tan, R. Khalid, S. Sararaks, K. F. Tang, I. Ab Rahim, et al., Applying discrete event simulation to reduce patient wait times and crowding: The case of a specialist outpatient clinic with dual practice system, *Healthcare (Basel)*, **10** (2022), 189. https://doi.org/10.3390/healthcare10020189
8. E. Allen, *Modeling with Itô Stochastic Differential Equations*, Springer, 2007.
9. A. Chadwick, S. S. Ho, Y. Li, M. Wang, A discrete model for bike share inventory, *Int. J. Differ. Equations*, **15** (2020), 363–375.
10. L. Kong, M. Wang, Deterministic and stochastic online social network models with varying population size, *DCDIS Ser. A Math. Anal.*, **30** (2023), 253–275.
11. L. Kong, M. Wang, Optimal control for an ordinary differential equation online social network model, *Differ. Equations Appl.*, **14** (2022), 205–214. http://dx.doi.org/10.7153/dea-2022-14-13
12. A. Farea, O. Yli-Harja, F. Emmert-Streib, Understanding physics-informed neural networks: Techniques, applications, trends, and challenges, *AI*, **5** (2024), 1534–1557. https://doi.org/10.3390/ai5030074
13. K. Luo, J. Zhao, Y. Wang, J. Li, J. Wen, J. Liang, et al., Physics-informed neural networks for PDE problems: A comprehensive review, *Artif. Intell. Rev.*, **58** (2025), 323. https://doi.org/10.1007/s10462-025-11322-7
14. W. He, J. Li, X. Kong, L. Deng, Multi-level physics informed deep learning for solving partial differential equations in computational structural mechanics, *Commun. Eng.*, **3** (2024), 151. https://doi.org/10.1038/s44172-024-00303-3
15. S. Cuomo, V. S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, F. Piccialli, Scientific machine learning through physics–informed neural networks: Where we are and what's next, *J. Sci. Comput.*, **92** (2022), 88. https://doi.org/10.1007/s10915-022-01939-z
16. D. Llorente-Vidrio, M. Ballesteros, I. Salgado, I. Chairez, Deep learning adapted to differential neural networks used as pattern classification of electrophysiological signals, *IEEE TPAMI*, **44** (2022), 4807–4818. https://doi.org/10.1109/TPAMI.2021.3066996
17. W. E, T. Li, E. Vanden-Eijnden, *Applied Stochastic Analysis*, AMS, 2019.
18. *Queue Times*. Available from: https://queue-times.com/en-US/.
19. L. Kong, R. Shi, M. Wang, A physics-informed neural network model for social media user growth, *Appl. Comput. Intell.*, **4** (2024), 195–208. https://doi.org/10.3934/aci.2024012

20. L. Kong, R. Shi, M. Wang, An age-structured model for COVID-19 hospitalization rate, *Mathematics*, **14** (2026), 58. https://doi.org/10.3390/math14010058

21. D. Kincaid, W. Cheney, *Numerical Analysis: Mathematics of Scientific Computing*, 3rd edition, American Mathematical Society, 2002.

AIMS Press