Physics-Informed Neural Networks for fault slip monitoring: simulation, frictional parameter estimation, and prediction on slow slip events in a spring-slider system

Rikuto Fukushima[1], Masayuki Kano[2], and Kazuro Hirahara[3,4]

[1] Faculty of Science, Kyoto University, Kyoto, Japan.

[2] Graduate School of Science, Tohoku University, Sendai, Japan.

[3] RIKEN Center for Advanced Intelligence Project, Seika, Japan.

[4] Kagawa University, Takamatsu, Japan.

Corresponding author: Rikuto Fukushima (fukushima.rikuto.27n@st.kyoto-u.ac.jp) and Masayuki Kano (masayuki.kano.a3@tohoku.ac.jp)

Key Points:

- We propose Physics-Informed Neural Networks (PINNs) for fault slip simulation, frictional parameter estimation, and slip prediction.
- PINNs can reproduce slow slip events in a spring-slider system and estimate the frictional parameters from synthetic observation data.
- We investigated the potential of the predictability of subsequent fault slips from limited observation data including uncertainties.

Abstract

The episodic transient fault slips called slow slip events (SSEs) have been observed in many subduction zones. These slips often occur in regions adjacent to the seismogenic zone during the interseismic period, making monitoring SSEs significant for understanding large earthquakes. Various fault slip behaviors, including SSEs and earthquakes, can be explained by the spatial heterogeneity of frictional properties on the fault. Therefore, estimating frictional properties from geodetic observations and physics-based models is crucial for fault slip monitoring. In this study, we propose a Physics-Informed Neural Network (PINN)-based new approach to simulate fault slip evolutions, estimate frictional parameters from observation data, and predict subsequent fault slips. PINNs, which integrate physical laws and observation data, represent the solution of physics-based differential equations. As a first step, we validate the effectiveness of the PINN-based approach using a simple single-degree-of-freedom spring-slider system to model SSEs. As a forward problem, we successfully reproduced the temporal evolution of SSEs using PINNs and indicated how we should choose the appropriate collocation points depending on the residuals of physics-based differential equations. As an inverse problem, we estimated the frictional parameters from synthetic observation data and demonstrated the ability to obtain accurate values regardless of the choice of first-guess values. Furthermore, we discussed the potential of the predictability of the subsequent fault slips using limited observation data, taking into account uncertainties. Our results indicate the significant potential of PINNs for fault slip monitoring.

Plain Language Summary

Slow slip events (SSEs), which are fault slips characterized by slower velocity and longer duration compared to earthquakes, have been observed in many subduction zones. Monitoring SSEs is important for understanding large earthquakes because they occur adjacent to areas where significant earthquakes could potentially occur. Different types of fault slips, including SSEs and earthquakes, can be explained by distinct frictional properties on the fault. These frictional properties can be estimated from physical laws of fault slip and observed crustal deformation. In this study, we propose a new machine-learning based approach for fault slip monitoring. We employed Physics-Informed Neural Networks (PINNs), which simultaneously learn the physical laws and data, to simulate fault slip, estimate the frictional parameters, and predict subsequent fault slip. As a first step, we utilized a single-degree-of-freedom spring-slider system, which is the simplest physical model to simulate SSEs. We successfully simulated SSEs, estimated frictional properties from synthetic observation data, and discussed the potential for fault slip prediction. Our results suggest the significant potential of PINNs for fault slip monitoring.

1 Introduction

Recent geophysical observations have revealed that faults episodically slip slowly during the interseismic period (e.g., Hirose et al., 1999). These episodic slow fault slips, known as slow slip events (SSEs), have been observed in many subduction zones. SSEs repeatedly occur in regions adjacent to possible source areas of large earthquakes (Obara & Kato, 2016). Moreover, SSEs have been considered to share common physical mechanisms with large earthquakes. Therefore, it is crucial to monitor these slow fault slip phenomena and understand their generation mechanisms.

Various fault slip behaviors, including SSEs and earthquakes, can be explained by distinct frictional properties on the fault (e.g., Yoshida & Kato, 2003). For fault slip monitoring, it is crucial to estimate the frictional properties from current geodetic observations and predict fault slip evolutions based on physics-based models. Incorporating fault friction in the model enables us to simulate the spatio-temporal evolution of fault slip on the megathrust. In these simulations, the quasi-dynamic equation of motion (Rice, 1993), and a rate and state dependent friction (RSF) law (Dieterich, 1979), derived empirically from laboratory experiments, are frequently employed. Various fault slips can be reproduced by appropriately setting three frictional parameters (a, a–b, and d_c) that control the frictional properties on the fault in RSF. In such simulations, the frictional parameters are determined by trial and error to qualitatively reproduce the observed fault slips due to the difficulty of directly measuring these frictional parameters.

Therefore, for fault slip monitoring, it is vital to determine the appropriate frictional parameters by combining observations and physics-based models. To achieve this, data assimilations have been employed to investigate frictional parameters from observed slip velocities of afterslip (Kano et al., 2015; 2020) and long-term SSEs (Hirahara & Nishikiori, 2019). Kano et al. (2020) estimated frictional parameters from observed crustal deformation following the 2003 Tokachi-oki earthquake and predicted subsequent fault slips and crustal deformation. These studies confirmed that data assimilations enable the optimization of unknown frictional parameters on faults based on observed crustal deformation and physics-based models.

In this paper, we propose a new machine learning-based approach to simulate fault slip evolutions, estimate frictional parameters from observation data, and predict subsequent fault slips using the estimated frictional parameters. With recent advancements in machine learning, Physics-Informed Neural Networks (PINNs) have been proposed as a new deep learning method for data-driven solutions of partial differential equations as forward problems, as well as for the data-driven discovery of partial differential equations as inverse problems to investigate parameters that best describe the observed data (Raissi et al., 2019). This method involves constructing neural networks capable of solving physics-based equations by minimizing a loss function that incorporates differential equations and initial/ boundary conditions. This approach has been recently employed in numerous research fields as it provides a mesh-free framework for forward problems and provides effective solutions for inverse problems. In seismology, PINNs have been employed in various problems, including travel time calculation (Smith et al., 2021a), hypocenter inversion (Smith et al., 2021b), full-waveform inversion (Rasht-Behesht et al., 2022), seismic tomography (Agata et al., 2023), and modeling crustal deformation (Okazaki et al., 2022).

Focusing on the ability of PINNs to estimate parameters by integrating physical law and observation data, this study first applies PINNs to the simulation of slip evolution on faults. As a first step, we utilize a simple single-degree-of-freedom spring-slider system (Yoshida & Kato, 2003) to model SSE. The objectives of this study are as follows: (i) simulating the temporal evolutions of SSE as a forward problem, (ii) estimating the frictional parameters from observed slip velocity data as an inverse problem, and (iii) predicting the future evolution of SSE, including quantifying the uncertainty associated with each result. Through these calculations, we aim to verify that our new PINN-based approach is a powerful tool for simulating slip evolutions, estimating frictional parameters, and predicting fault slip evolutions.

The paper is organized as follows. Section 2 explains the fault slip model based on the RSF law and presents the results of numerical calculations. Section 3 demonstrates the forward calculations of the temporal evolution of SSE using the PINN-based approach. In Section 4, we estimate the frictional parameters from synthetic observation data, considering their uncertainties.

Finally, in Section 5, we attempt to predict subsequent evolution of SSE from limited observation data and discuss the relationship between the uncertainties of the estimated parameters and the length of the observation data.

2 Numerical Simulation

We initiate the study by conducting a numerical simulation to obtain the temporal evolution of fault slips. This simulation serves as a reference for comparing results obtained using the PINN-based approach. We adopt a single degree-of-freedom spring-slider model (Yoshida & Kato, 2003), which comprises a block and a spring (Figure 1a). In this system, the block is loaded with a constant velocity. By assuming the frictional properties between the block and the surface, we can represent various fault slips ranging from slow to fast slips on the block. The quasi-dynamic equation of motion in this model is expressed as:

$$\tau = k\left(v_{pl}\, t - x\right) - \eta v, \tag{1}$$

where τ is the shear stress, k is the stiffness of the spring, v_{pl} is the loading velocity, t is the time, and x is the accumulated slip of the block. The second term on the right-hand side represents a radiation damping approximation (Rice, 1993), which was introduced to express the stress-release by the radiation of seismic waves instead of the inertia term. The coefficient η is expressed as $\eta = \mu / 2v_s = 5 \times 10^6$ [Pa s/m], where the shear modulus μ is 3.0×10^{10} [Pa] and shear wave velocity v_s is 3×10^3 [m/s].

The RSF law (Dieterich, 1979) is often employed to express fault friction. The frictional stress τ is expressed as

$$\tau = \sigma\left(f_0 + a \log\left(\frac{v}{v_{pl}}\right) + b \log\left(\frac{\theta\, v_{pl}}{d_c}\right)\right), \tag{2}$$

where θ is the state variable, σ is the normal stress, f_0 is a frictional coefficient, and a, b, and d_c are the frictional parameters. These frictional parameters express the frictional properties of faults. If $a - b > 0$, the friction becomes rate-strengthening and if $a – b < 0$, the friction becomes rate-weakening, which makes the system unstable. The instability of the model is determined by the frictional parameters, spring stiffness k, and critical stiffness k_{crit} (Ruina, 1983) defined as:

$$k_{crit} = \frac{\sigma\,(b-a)}{d_c}. \tag{3}$$

When k is larger than k_{crit}, the system exhibits strong instability and behaves like fast earthquakes. When $k < k_{crit}$ and $k \approx k_{crit}$, the system shows a slow transient motion like SSEs. The state variable characterizes the state of the fault surface and several laws were proposed to describe the temporal evolution of the state variable. Here, we used the aging law (Ruina, 1983) described as:

$$\frac{\mathrm{d}\theta}{\mathrm{d}t} = 1 - \frac{\theta v}{d_c}. \tag{4}$$

By combining these equations Eqs. (1), (2), and (4), we can calculate the temporal evolution of slip velocity v and state variable θ. We non-dimensionalized these equations by defining:

$$p = \log\left(\frac{v}{v_{pl}}\right), \tag{5}$$

$$q = \log\left(\frac{\theta v_{pl}}{d_c}\right). \tag{6}$$

Then the target equations are written as:

$$\frac{\mathrm{d}p}{\mathrm{d}t} = \left(a\sigma + \eta\, v_{pl}\, \mathrm{e}^{p}\right)^{-1}\left(kv_{pl}(1 - \mathrm{e}^{p}) - \frac{b\sigma v_{pl}}{d_c}(\mathrm{e}^{-q} - \mathrm{e}^{p})\right), \tag{7}$$

$$\frac{\mathrm{d}q}{\mathrm{d}t} = \frac{v_{pl}}{d_c}(\mathrm{e}^{-q} - \mathrm{e}^{p}). \tag{8}$$

We set the frictional parameters a, b, and d_c, and normal stress σ to reproduce the SSE as $a = 1 \times 10^{-4}$, $a–b = –1 \times 10^{-5}$, $d_c = 5 \times 10^{-3}$ [m], and $\sigma = 10^{7}$[Pa]. The spring stiffness k is set to satisfy $k / k_{crit} = 0.9999$, which is required to cause the transient motion. We set the loading rate v_{pl} = 5 [cm/yr] = 1.58×10^{-9} [m/s]. The temporal evolution of slip velocity under these parameters was calculated by the 5th-order time-adaptive Runge–Kutta (RK) method with a tolerance of 10^{-8}. Figures 1b and 1c display the temporal evolutions of slip velocity v and state variable θ. The simulation results showed a maximum slip velocity of ~10^{-8} [m/s], a cumulative slip of ~10 cm, and recurrence intervals of SSEs of 2.5 years. These characteristics are similar to the Tokai slow slip events in the first-order approximation (Miyazaki et al., 2006). Hereafter we will use this result as a reference, aiming to calculate the temporal evolution of SSE in one cycle.

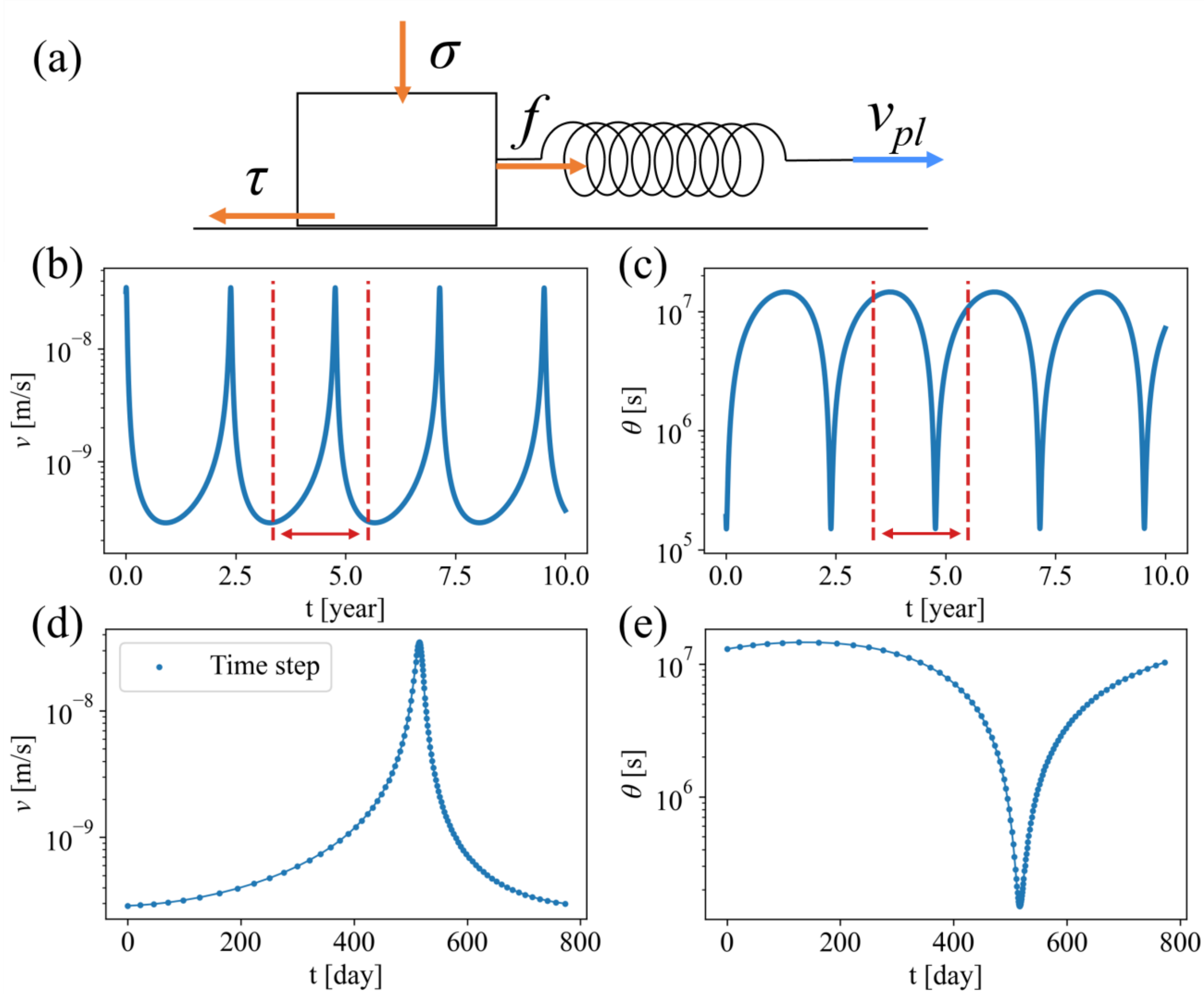


Figure 1. (a) Schematic illustration of a spring-slider model. (b-e) Results of numerical calculation. (b)(c) Temporal evolutions of (b) slip velocity v and (c) state variable θ for several cycles. (d)(e) Enlarged view of (b) and (c) focusing on one cycle indicated by red lines in (b) and (c). Blue points show the time steps used in the time-adaptive RK method.

3 Forward Problem

In this section, we describe how to model the fault slips in a spring-slider model using the PINN-based approach and discuss the results.

3.1 Method

A neural network was constructed to model the temporal evolutions of $p(t)$ and $q(t)$ (Figure 2). The network uses an input layer with one node corresponding to time t, and an output layer with two nodes corresponding to $p(t)$ and $q(t)$. It has a nine-layer fully connected neural network and uses the hyperbolic tangent as the activation function. The number of intermediate layers is eight with twenty nodes each. In total, the neural network has 162 biases and 2860 weights, and we can solve the differential equations by optimizing these neural network parameters. We follow the original framework of PINNs (Raissi et al., 2019) to construct this neural network configuration. In this study, the network biases are initialized to zero, and the network weights were initialized by normal Xavier initialization (Glorot & Bengio, 2010), which is widely used in PINNs. In this initialization method, the weights are selected from the Gaussian distribution.

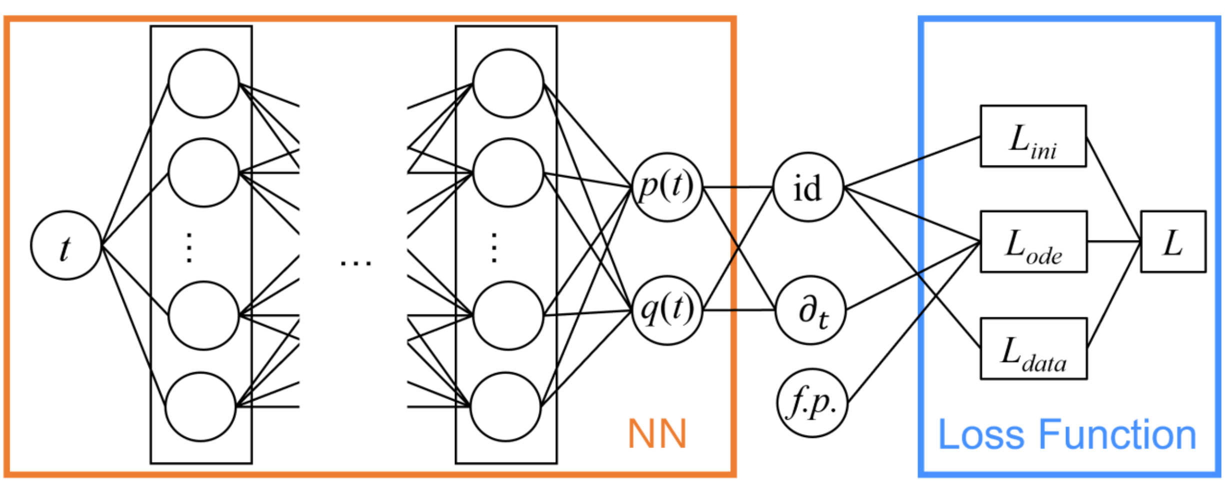


Figure 2. Structure of PINNs. The neural networks have an input, time t, and the corresponding outputs $p(t)$ and $q(t)$. The loss function L is calculated by using the frictional parameters ($f.p.$) and

operating the identity function (id) and the time derivation ($\boldsymbol{\partial_t}$) to p and q. In Section 3, we solve the forward problem using L_{ini} (Eq. (12)) and L_{ode} (Eq. (13)). We introduce L_{data} (Eq. (15)) in the inversion in Sections 4 and 5.

In the PINNs, the neural networks learn the behavior of the equation by defining the loss function considering the misfit between target equations and derivatives of the network output calculated by automatic differentiation. In this problem, we define the residuals of the differential equations as:

$$r_p(t) = \frac{\mathrm{d}p_{NN}}{\mathrm{d}t} - \left(a\sigma + \eta v_{pl} \mathrm{e}^{p_{NN}}\right)^{-1} \left(k v_{pl}(1 - \mathrm{e}^{p_{NN}}) - \frac{b\sigma v_{pl}}{d_c}(\mathrm{e}^{-q_{NN}} - \mathrm{e}^{p_{NN}})\right), \tag{9}$$

$$r_q(t) = \frac{\mathrm{d}q_{NN}}{\mathrm{d}t} - \frac{v_{pl}}{d_c}(\mathrm{e}^{-q_{NN}} - \mathrm{e}^{p_{NN}}), \tag{10}$$

where p_{NN} and q_{NN} are the PINNs outputs. Then the loss function L is defined as:

$$L = L_{ini} + L_{ode}, \tag{11}$$

$$L_{ini} = (p_{NN}(0) - p_{ini})^2 + (q_{NN}(0) - q_{ini})^2, \tag{12}$$

$$L_{ode} = t^* \sum_{i=1}^{N} r(t_i)^2 \, \Delta t_i. \tag{13}$$

where $r(t_i)^2 = r_p(t_i)^2 + r_q(t_i)^2$, p_{ini} and q_{ini} are the initial conditions of p and q. As shown in Figures 1b, and 1c, our simple frictional model produces the same repeating SSE cycles after several unstable cycles when the numerical effect of the initial conditions disappears. Focusing on one cycle (Figures 1d and 1e), we set the initial time $t = 0$ at the time when the slip velocity is lowest during one cycle. The initial conditions of p_{ini} and q_{ini} were evaluated at this time $t = 0$. L_{ini} and

L_{ode} represent the residuals of the initial conditions and those of the governing equations, respectively. L_{ode} can be calculated at the arbitrary points t_i, which are called collocation points. Δt_i indicates time intervals of collocation points and N is the number of collocation points. L_{ode} is defined as the discretization of the L2 norm: $\int r(t)^2 \mathrm{d}t$, representing the residuals of governing equations. We non-dimensionalized L_{ode} by multiplying t^*, which represents the characteristic time in a spring-slider system defined as $t^* = d_c / v_{pl}$ (Segall, 2010). We used the L-BFGS method (Liu & Nocedal, 1989) to optimize the network weights and biases by minimizing the loss function. Training is finished when the decrease in the loss function per one optimization step becomes less than the predetermined threshold value of 10^{-12}.

To calculate L_{ode}, the selection of collocation points is required. In this study, we employed two types of collocation points: equidistant and non-equidistant. Equidistant collocation points have constant time intervals, while non-equidistant collocation points have adaptive time intervals. The time steps of the time-adaptive RK method, as shown in Figures 1d and 1e, were chosen as non-equidistant collocation points. This is based on the idea that a higher density of collocation points should be selected where the slip behavior in the system equation changes rapidly. However, this method can only be applied when the slip behavior is known prior to the calculation. In practical situations where there is no prior knowledge of the slip behavior, equidistant collocation points are used. In the RK calculation, the total number of collocation points is 103, with maximum and minimum intervals of ~870 and ~ 10 h, respectively. For equidistant collocation points, intervals are set to be 100, 200, and 400 h with the corresponding number of points being 187, 94, and 47, respectively. By comparing these results, we investigate the impact of collocation point sampling on the learning of neural network parameters.

We validate the ability of PINNs to reproduce the SSE by comparing the PINNs outputs with the results derived from a numerical calculation using the RK method (Figures 1d and 1e). To qualitatively evaluate the misfit between the PINNs outputs and the reference values, we defined the relative errors (*RE*) as $RE = |v_{NN} - v_{RK}| / v_{RK}$ or $RE = |\theta_{NN} - \theta_{RK}| / \theta_{RK}$. Here v_{RK} and θ_{RK} represent the reference values at adaptive time steps used in the RK method, whereas v_{NN} and θ_{NN} represent the PINNs outputs corresponding to those times. Please note that the *RE* are calculated at RK time steps.

3.2 Results and discussion

We solved the spring-slider problem using PINNs, applying both equidistant and non-equidistant collocation points (Figures 3 and 4). Here, time intervals for equidistant collocation points were set as 100 h. Note that $v_{NN}(t)$ and $\theta_{NN}(t)$ in these figures are calculated at time steps with constant time intervals of 10 h, which are denser than equidistant collocation points of 100 h. This is used to check the interpolation ability of PINNs. The PINNs successfully reproduced the temporal evolution of SSE in both cases (Figures 3a, 3b, 4a, and 4b). The neural network parameters were optimized until the number of iterations reached 11,875 and the loss function yielded $L_{ini} = 3.56 \times 10^{-8}$ and $L_{ode} = 3.19 \times 10^{-6}$ for the case of equidistant collocation points (Figure 3c). When using non-equidistant collocation points, the number of iterations increased to 17,674, with loss functions $L_{ini} = 5.21 \times 10^{-10}$ and $L_{ode} = 1.94 \times 10^{-6}$ (Figure 4c).

The maximum values of *RE* were $\sim 10^{-1}$ and $\sim 10^{-2}$ for the equidistant and non-equidistant collocation points case, respectively (Figures 3d and 4d). This suggests that training with non-equidistant collocation points yields more precise results than using equidistant collocation points.

The residuals of the governing equations, represented by $r(t)^2$ (Figures 5a and 5b), help in elucidating such results. It is important to note that $r(t)^2$ differs from *RE*: $r(t)^2$ represents the discrepancy between the time derivatives of p_{NN} and q_{NN} and the governing equations, while *RE* represents the misfit between the PINNs outputs (v_{NN} and θ_{NN}) and the reference values (v_{RK} and θ_{RK}). The scarcity of collocation points around peak velocity in the case of equidistant collocation points results in larger $r(t)^2$ values compared to non-equidistant collocation points, thereby increasing the difference between the output of PINNs and the reference values. These results suggest that for accurate calculations, a larger number of collocation points are required at timings when the slip velocity changes dramatically, which aligns with the concept of the time-adaptive RK method.

While the use of non-equidistant collocation points yields a more accurate *RE*, the successful calculation of PINNs with equidistant collocation points is important to apply PINNs for practical use. Without prior knowledge of the solution of the equations, it is impossible to collect the collocation points at peak times of slip velocity. This approach ensures a feasible methodology when the temporal pattern of changes in the system is unknown.

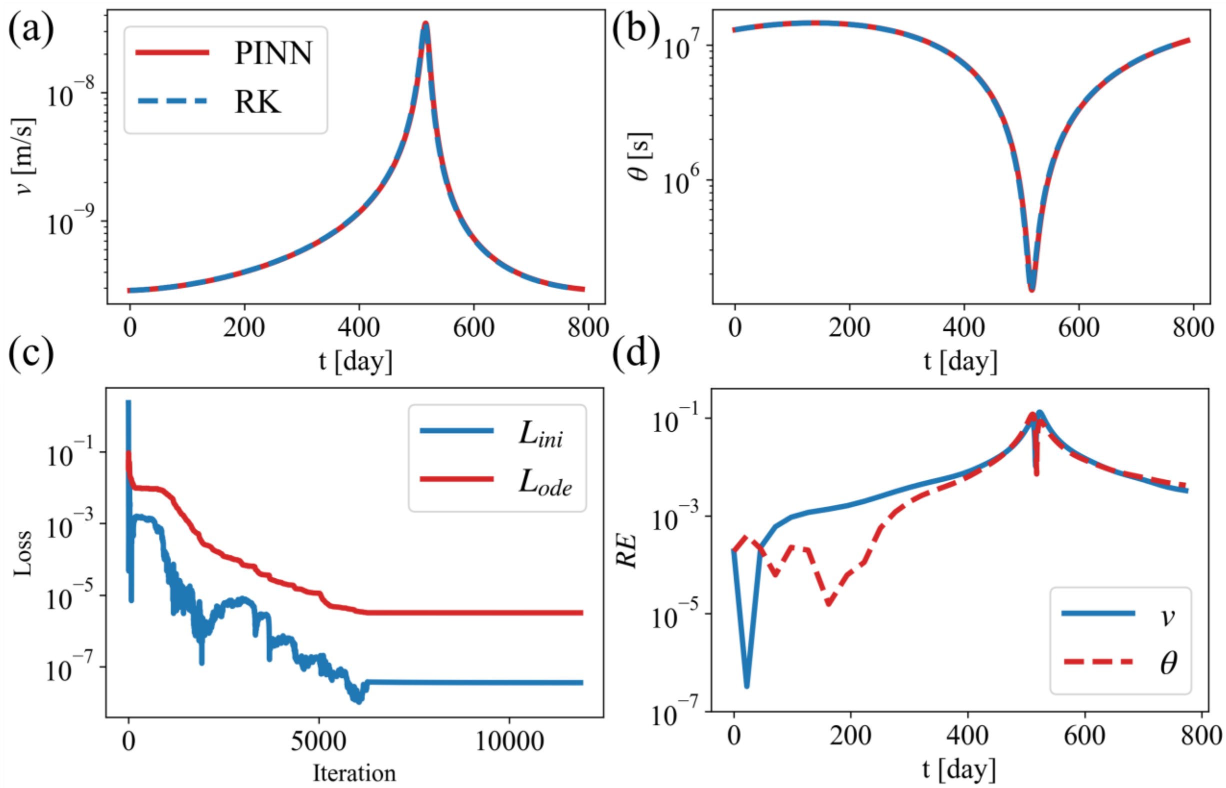


Figure 3. Calculation results using equidistant collocation points (with time intervals of 100 h). (a and b) The temporal evolution of (a) v and (b) θ calculated by PINNs (red line) and the RK method (blue line). (c) Learning curve for L_{ini} and L_{ode}. The neural network parameters were converged after 11,875 iterations with $L_{ini} = 3.56 \times 10^{-8}$ and $L_{ode} = 3.19 \times 10^{-6}$. (d) RE of v and θ.

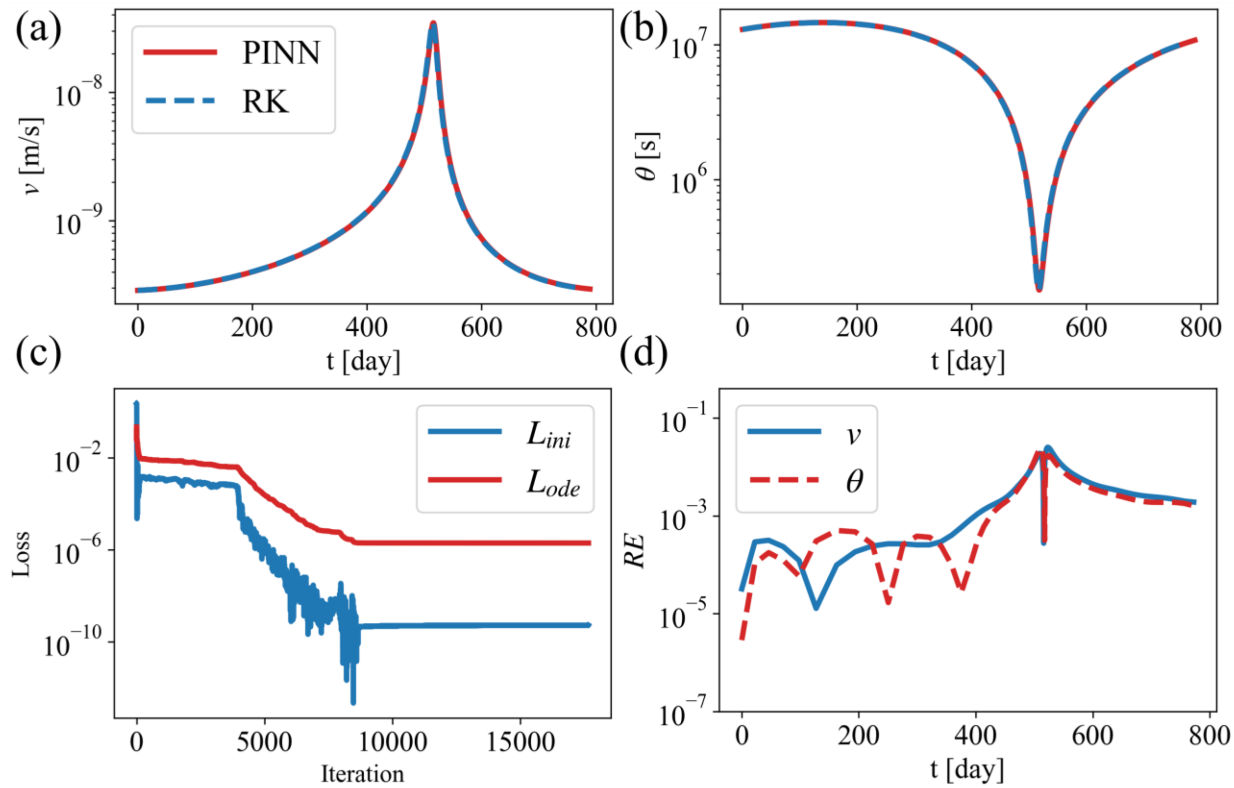
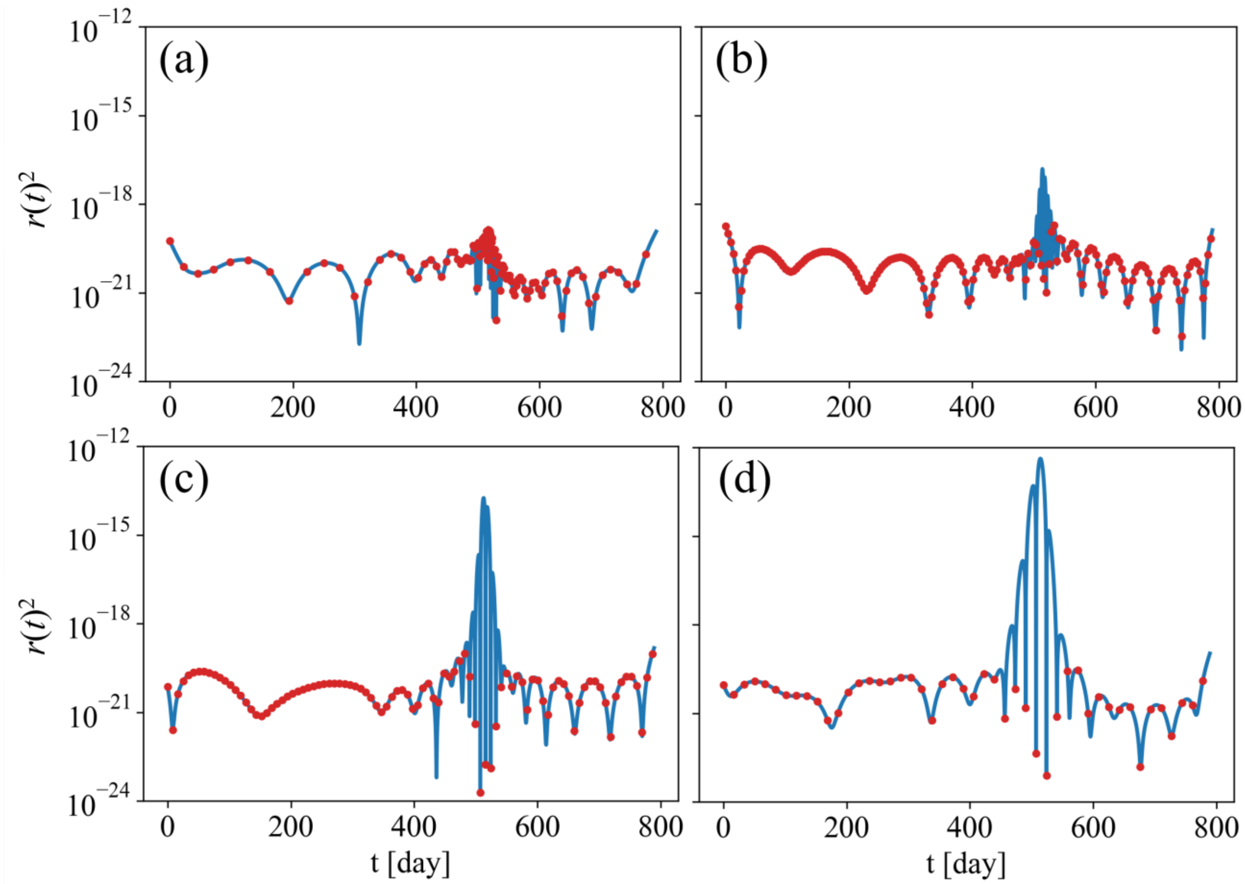


Figure 4. Calculation results using non-equidistant collocation points. (a and b) The temporal evolution of (a) v and (b) θ calculated by PINNs (red line) and the RK method (blue line). (c) Learning curve for L_{ini} and L_{ode}. The neural network parameters were converged after 17,674 iterations with $L_{ini} = 5.21 \times 10^{-10}$ and $L_{ode} = 1.94 \times 10^{-6}$. (d) *RE* of v and θ.

Next, we solved the spring-slider problem using PINNs, varying the number of equidistant collocation points, and discussed the relationship between the number of collocation points and the *RE*. In principle, increasing the number of collocation points enhances accuracy but slows down the computation speed. Thus, we explored the level of accuracy we could achieve in training the neural networks with fewer collocation points. We employed 94 and 47 equidistant collocation points corresponding to intervals of 200 and 400 h, respectively, while we used 187 equidistant collocation points with intervals of 100 h in the previous section.

The results indicate that we can accurately model the temporal evolution of fault slip even with fewer collocation points (Figures 6a, 6b, 6d, and 6e). The neural network parameters were optimized after 6,492 iterations with the loss function of $L_{ini} = 1.51 \times 10^{-7}$ and $L_{ode} = 2.04 \times 10^{-6}$ in the case of 200 h equidistant collocation points. For the case of 400 h equidistant collocation points, the iteration increased to 9,942, and the loss function is $L_{ini} = 1.45 \times 10^{-6}$ and $L_{ode} = 9.41 \times 10^{-8}$. A decreasing number of collocation points results in larger *RE* values after the velocity peak time (Figures 6c and 6f). The residuals of the governing equations at each time t (Figures 5c and 5d) indicate that if there are fewer collocation points, the entire $r(t)^2$ is not optimized adequately, and $r(t)^2$ locally increases, leading to worse *RE* values. However, even with the number of collocation points reduced to 47 (Figure 6f), the maximum *RE* is 10^{-1}, which is noteworthy considering that the number of collocation points is less than half of that in the non-equidistant collocation point case. This result demonstrates the high interpolation ability of neural networks and suggests the potential of PINNs for rapid computation with fewer collocation points in large-scale problems. It is important to note that the computation speed of PINNs depends not only on the number of collocation points but also on the number of iterations required for optimization convergence.



298

299 **Figure 5.** Time series of residuals of the governing equations $r(t)^2$ on (a) non-equidistant and (b–
300 d) equidistant collocation points with time intervals of (b) 100, (c) 200, and (d) 400 h. The red
301 points represent the collocation points.

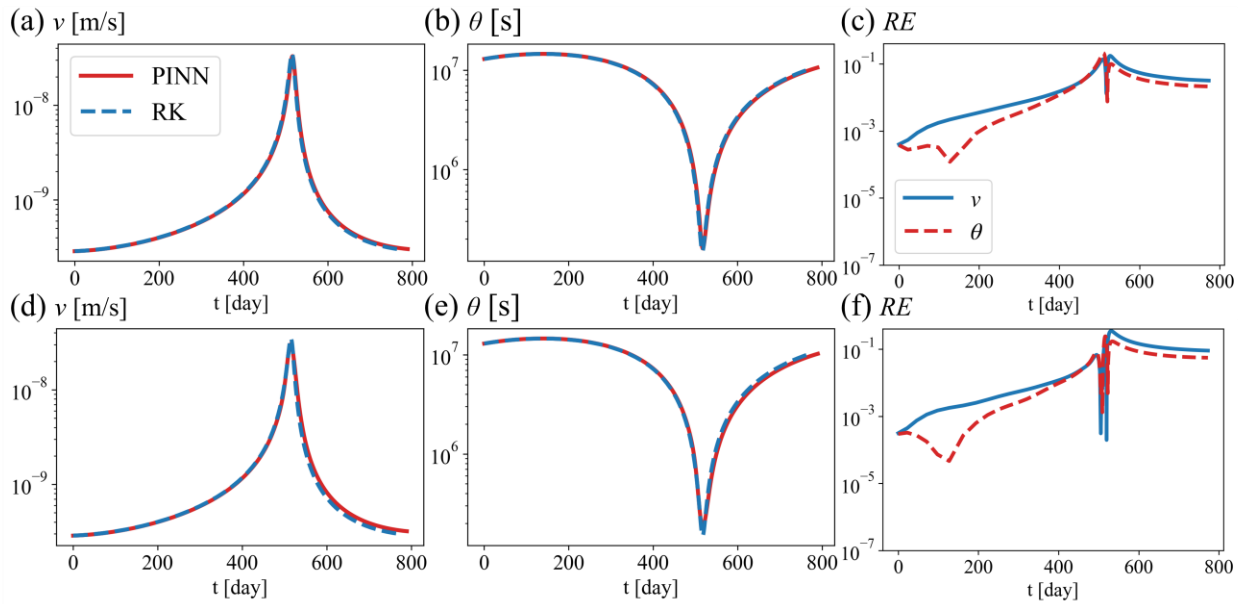


Figure 6. Calculation results using equidistant collocation points with time intervals of (a–c) 200 h and (d–f) 400 h. (a and b) The temporal evolution of (a) v and (b) θ calculated by PINNs (red line) and the RK method (blue line) for time intervals of 200 h. (c) *RE* of v and θ. (d–f) Same as (a–c) but for time intervals of 400 h.

In summary, the relationship between the number of collocation points and calculation accuracy can be understood as follows. Firstly, fewer collocation points require long range interpolation of the residuals of the differential equation, $r(t)^2$, leading to insufficient optimization of $r(t)^2$. Secondly, large residuals of differential equations result in larger *RE* values. Understanding this relationship aids in determining the best collocation points for precise calculations. When considering problems involving modeling faster slip, more complex interpolation is required, as the temporal change in slip velocity is more drastic, thus demanding smaller time intervals for precise calculation. Solving stiff equations also requires smaller intervals because the residuals of differential equations significantly impact the solution of such equations.

Due to these reasons, applying PINNs to earthquake models, which involve rapid slips and require solving very stiff equations, proves challenging. Therefore, in this study, we first calculated the temporal evolution of SSE at the initial step of applying PINNs to fault slip modeling. However, some recent studies have tried to develop techniques for solving stiff equations within the framework of PINNs (e.g., Guo et al., 2022). Leveraging these methods will overcome the challenges associated with using PINNs to model fast slips or earthquakes in the future.

3.3 Uncertainty quantification in forward problems

In this subsection, we discuss the uncertainties of PINNs arising from the initial network parameters. Recent studies have highlighted the importance of uncertainty quantification in PINNs and have proposed a method for its calculation (e.g., Yang & Perdikaris, 2019). In this study, we used normal Xavier initialization, and the initial values varied based on the random seed value. Consequently, the optimized network parameters we eventually obtained are influenced by the differences in the initial network parameters. We quantified the uncertainties of the converged network parameters by repeating the optimization process with different values of initial network parameters. Figure 7 represents the results optimized from 10 different initial network parameters using equidistant collocation points with the intervals of 100 h. The maximum relative error ranges from 10^{-2} to 10^{-1}, indicating uncertainties due to the selection of initial network parameters.

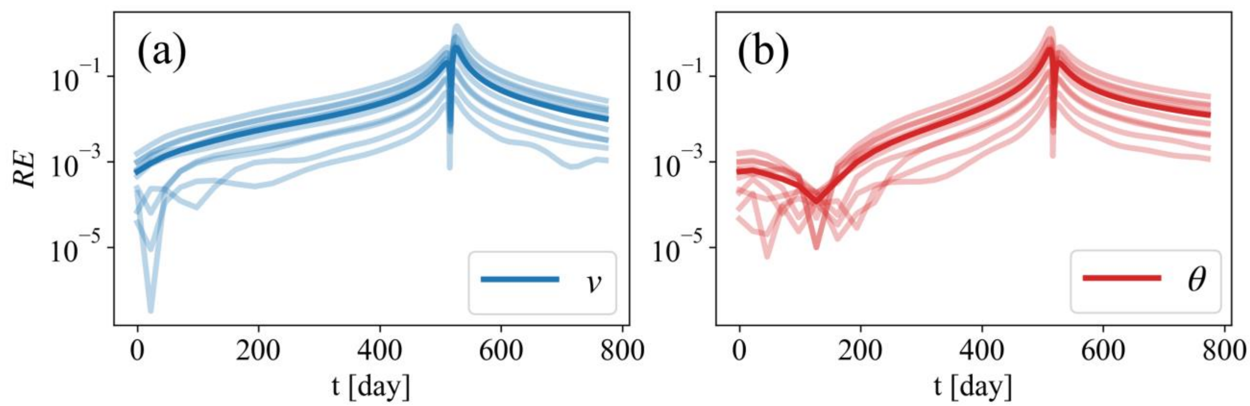


Figure 7. *RE* of (a) v and (b) θ optimized from 10 different initial network parameters (thin lines) and their mean values (dark lines).

4 Inverse Problem

One of the significant advantages of PINNs is their inherent flexibility to extend to the inverse problems. In this section, we extend a forward problem for simulating fault slips described in Section 3 to an inverse problem for estimating unknown frictional parameters from the observation data.

4.1 Method and synthetic data

To extend to the inverse problems, we add a misfit term related to observed data to the loss function used in the forward problem, allowing us to simultaneously learn from the observation data and physical laws. We estimate three frictional parameters a, a–b, and d_c by giving synthetic data of the slip velocity including some errors as the observation data into the neural network. Although the frictional parameters control slip behavior on a fault, it is difficult to measure these parameters directly on the plate interface. Therefore, if PINNs can effectively solve this inverse problem, they would become a powerful tool for improving our understanding of fault properties.

We modified the loss function of Eq. (11) as follows:

$$L = L_{ini} + L_{ode} + L_{data}, \tag{14}$$

where

$$L_{data} = \frac{1}{N_{data}} \sum_{i=1}^{N_{data}} \left(p_{NN}(t_i) - p_{obs}(t_i)\right)^2 . \tag{15}$$

The third term in Eq. (14) represents a misfit term for observation data defined as the squared residuals between the observation data for slip velocity p_{obs} and the PINNs output p_{NN}, where N_{data} is the number of data points. Since the orders of magnitude of the frictional parameters vary, we defined the logarithm of the frictional parameters as $\alpha = \log a$, $\beta = \log(-(a-b))$ and $\gamma = \log d_c$. Note that this transformation implicitly assumes that $a-b$ is negative. We simultaneously optimized the frictional parameters α, β, and γ along with the neural network parameters by minimizing the loss function L (Figure 2). The true frictional parameters are set to be $a = 1.0 \times 10^{-4}$, $a-b = -1.0 \times 10^{-5}$, and $d_c = 5.0 \times 10^{-3}$ [m], as used in the previous forward problem. The first-guess values of the frictional parameters are set to be $a = 1.0 \times 10^{-3}$, $a-b = -1.0 \times 10^{-6}$, and $d_c = 5.0 \times 10^{-2}$ [m], assuming a prior knowledge of the frictional parameters ranging from × 0.1 to × 10 relative to true values. We discussed these first-guess values in Section 4.3.

To verify whether we can estimate the frictional parameters using the PINN-based approach, we utilized synthetic slip velocity data. We generated this synthetic data with the constant time intervals by adding the observation error to the true values v_{true}. Hence, the synthetic observation data v_{obs} is

$$v_{obs} = (1 + Er)\, v_{true}, \tag{16}$$

where Er is the observation error. In order to generate true values with different constant time intervals, we utilized PINNs, allowing us to express the continuous function. As numerical calculation results are discrete, it was necessary to perform calculations again, based on the specific times of interest whenever we wanted to obtain velocities at new time steps. To avoid this, we trained the neural network by providing the initial conditions, the governing equations, and the results calculated by the RK method. Notably, we used the results of numerical calculations to achieve more precise training, although this is not strictly necessary to train the PINNs. As a result, the obtained neural network is a continuous function that represents the solution to this problem, enabling us to obtain the velocity at any arbitrary time without recalculating the solution. In other words, we can interpolate the discrete outputs of numerical calculations using the PINN-based approach. We assume the observation error Er follows a Gaussian distribution with a mean of zero and standard deviations of σ_{er} = 0.1 or 0.25. Time intervals of the observation data are set to be 100, 200, and 400 h.

In this section, we employ equidistant collocation points with intervals of 100 h. Training is finished when the change of the frictional parameters per one optimization step becomes smaller than the threshold value of 10^{-5}.

4.2 Results

Figure 8 and Table 1 summarize the results of the PINNs outputs and the optimized frictional parameters when we use the observation data with intervals of 100 h. The neural network can obtain an output that fits the data well by solving the differential equations (Figures 8a and

8d), and it retrieved the true frictional parameters (Figures 8b, 8e, and Table 1). We successfully optimized the frictional parameters with residuals smaller than 3.7 % compared to the true values. These results indicate that the PINN-based approach is useful for estimating the frictional parameters in inverse problems.

When using the synthetic slip velocity data with σ_{er} = 0.1, the network parameters and the frictional parameters were simultaneously optimized after 2,113 iterations (Figure 8c). All terms in the loss functions were reduced by learning the physical laws and observation data, ultimately reaching $L_{ini} = 1.31 \times 10^{-6}$, $L_{ode} = 8.49 \times 10^{-5}$, and $L_{data} = 1.05 \times 10^{-2}$. In the case of σ_{er} = 0.25, the optimization was converged after 3,075 iterations and the loss functions of $L_{ini} = 8.31 \times 10^{-5}$, $L_{ode} = 5.01 \times 10^{-4}$, and $L_{data} = 9.54 \times 10^{-2}$ (Figure 8f).

It is worth noting that L_{data} did not change significantly during the latter half of optimization and ultimately converged to a relatively large value compared to L_{ini} and L_{ode} due to the observation error. Even if the PINNs outputs completely fit the result of numerical calculations, the values of L_{data} were 1.06×10^{-2} and 9.73×10^{-2} in the cases of σ_{er} = 0.1 and 0.25, respectively. Therefore, this learning curve of L_{data} indicates how PINN estimated the frictional parameters. Initially, the neural network parameters were optimized to fit the observed velocity data, but at that time, the value of L_{ode} is large because the frictional parameters deviate from the true values, and they could not reproduce that result. Afterward, PINN searched for the frictional parameters that could decrease L_{ode} while fixing the velocity output. In other words, PINN initially tried to fit the data by discarding the governing equations, and then optimized the frictional parameters to comply with the physics. This disregard for physical laws at the initial stage of optimization is characteristic of the PINN-based inversion method.

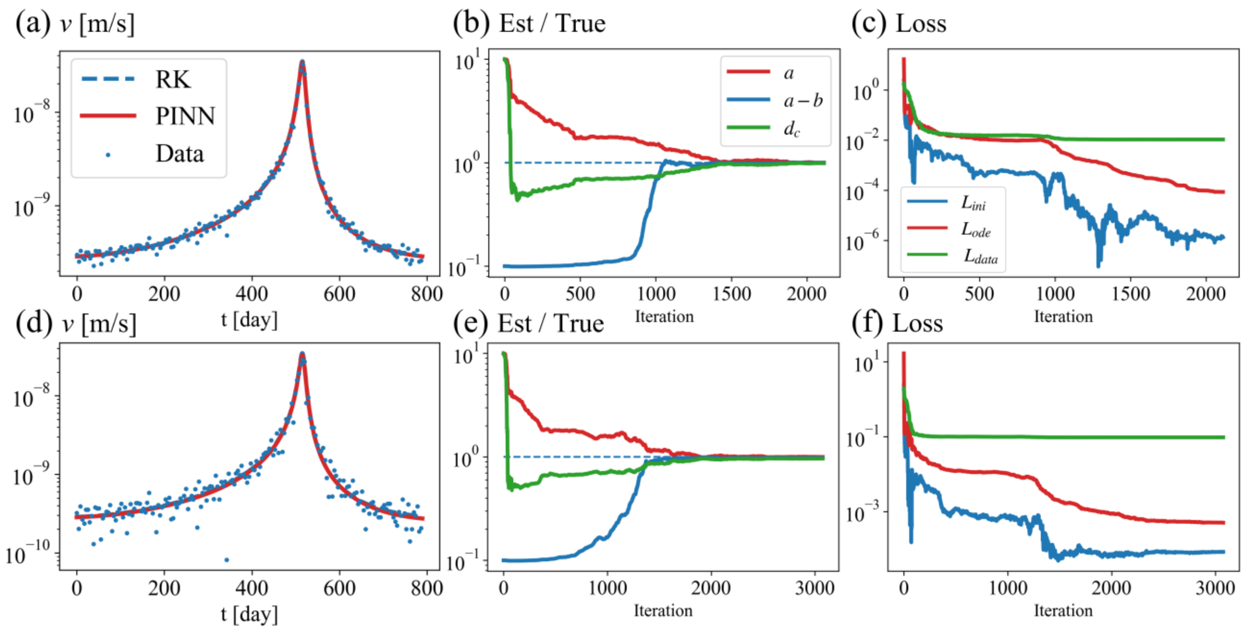


Figure 8. (a–c) Results of parameter estimation using the velocity data with the observation error of σ_{er} = 0.1. (a) Temporal evolution of *v* calculated by PINNs (red line) and the RK method (blue line). Note that the blue line is invisible because it overlaps the red line. Blue points show the synthetic data including observation error. (b) Values of estimated frictional parameters normalized by their true values on each iteration. The red, blue, and green lines represent the value of *a*, *a*–*b*, and d_c, respectively. (c) Learning curve for L_{ini}, L_{ode}, and L_{data}. The parameters converged after 2,113 iterations with $L_{ini} = 1.31 \times 10^{-6}$, $L_{ode} = 8.49 \times 10^{-5}$, and $L_{data} = 1.05 \times 10^{-2}$. (d–f) Same as (a–c) but for the case of σ_{er} = 0.25. The parameters converged after 3,075 iterations with $L_{ini} = 8.31 \times 10^{-5}$, $L_{ode} = 5.01 \times 10^{-4}$, and $L_{data} = 9.54 \times 10^{-2}$.

Table 1. True values, first-guess values, and estimated values of the frictional parameters using two synthetic data points. Error represents the relative error defined by |True value – Estimated value| / |True value| and Ratio represents the ratio of first-guess value to the true value.

	True value	First-guess value	Estimated value (Noise: σ_{er} = 0.1)	Estimated value (Noise: σ_{er} = 0.25)
a	1×10^{-4}	1×10^{-3} Ratio: × 10	1.004×10^{-4} Error: 0.4%	0.997×10^{-4} Error: 0.3%
$a–b$	$–1 \times 10^{-5}$	$–1 \times 10^{-6}$ Ratio: × 0.1	$–0.998 \times 10^{-5}$ Error: 0.1%	$–0.980 \times 10^{-5}$ Error: 2.0%
d_c [m]	5×10^{-3}	5×10^{-2} Ratio: × 10	4.945×10^{-3} Error: 1.1%	4.817×10^{-3} Error: 3.7%

Figure 9 summarizes the results of the PINNs outputs and the optimized frictional parameters obtained from the observation data with intervals of 200 and 400 h, considering a standard deviation of σ_{er} = 0.1. It is important to note that while it is possible to estimate the frictional parameters from the observation data with fewer data points, insufficient data points can lead to inaccurate parameter retrieval. As anticipated, when intervals are 200 h, the true frictional parameters can be successfully estimated (Figure 9c). However, when intervals are extended to 400 h, the estimate of the true frictional parameters fails (Figure 9f). With the estimated frictional parameters in the case of 400 h data, the system behaves more drastically, resulting in a peak velocity larger than the true slip velocity (Figure 9d). This discrepancy arises due to the limited number of data points available near the peak velocity. Consequently, when using fewer data points, we are unable to adequately constrain the frictional parameters, highlighting a limitation imposed by the model. To ensure reliable estimations of the frictional parameters, it is essential to employ a sufficient amount of data, aligning with our intuitive understanding of the problem.

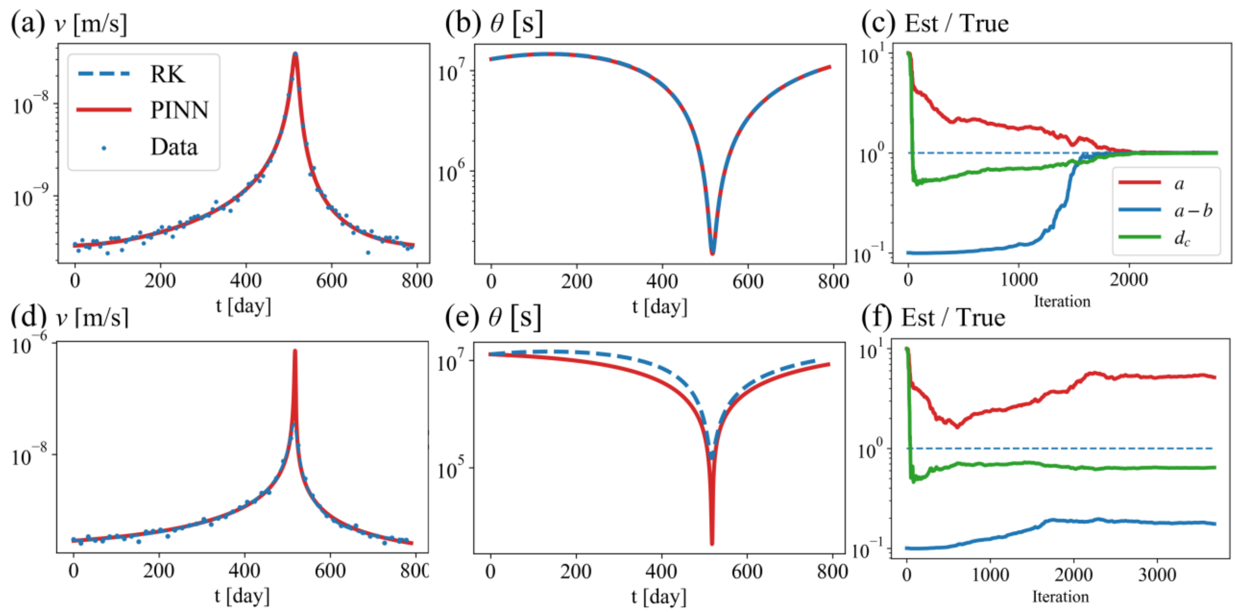


Figure 9. (a–c) Results of parameter estimation using the velocity data with time intervals of 200 h. (a and b) Temporal evolution of (a) v and (b) θ calculated by PINNs (red line) and the RK method (blue line). Note that the blue line is invisible because it overlaps the red line. Blue points represent the synthetic data points including the observation errors. (c) Values of estimated frictional parameters normalized by their true values on each iteration. The red, blue, and green lines represent the values of a, a–b, and d_c, respectively. (d–f) Same as (a–c) but for the case with time intervals of 400 h.

4. 3 Uncertainty quantification in inverse problems

Quantifying the uncertainties of estimated frictional parameters is essential for evaluating the robustness of the PINN-based inversion method. Additionally, this uncertainty analysis enables the evaluation of uncertainties associated with the resulting slip velocities, which is crucial for understanding the relationship between estimated frictional parameters and slip motion (Ito et al., 2022). To achieve this, we performed the optimization process multiple times using different first-guess values for the frictional parameters and various initial neural network parameters.

We trained the neural networks using eight different first-guess values (cases A–H) for the frictional parameters, as presented in Table 2. These values cover a range from ×0.1 to ×10 relative to the true values, assuming some prior knowledge of the frictional parameters. In each case, the optimization process was repeated using 10 different initial neural network parameters to estimate the frictional parameters. The neural networks were trained until the change in frictional parameters per optimization step became smaller than a reference value of 10^{-5} and L_{ode} was less than 10^{-3}. Synthetic data with a noise level of σ_{er} = 0.1 and intervals of 100 h were utilized.

Figure 10a illustrates the estimated values for each iteration across all eight cases. Although the optimization trajectories vary depending on the first-guess values, it was observed that in all cases, the estimated parameters eventually converged to the true values. Upon closer examination of each trajectory, it was noticed that d_c reaches ~50 % of the true values shortly after the start of optimization in all cases. This result suggests that estimating the order of magnitude of d_c from the observation data is relatively straightforward, and the gradient of the loss function with respect to d_c is larger compared to the gradients with respect to other frictional parameters. Except for cases E and F, respective trajectories of the 10 optimizations are relatively similar. However, the trajectories of cases E and F differed depending on the initial values of the neural networks. Furthermore, the trajectories of cases A–B, and cases C, D, G–H were comparable to each other. These findings indicate that the gradients of the loss function with respect to frictional parameters are small when the first-guess values of *a* are small, and they are large when the first-guess values of *a*–*b* are large. Conversely, the gradients are relatively large in other cases.

Figure 10b represents the distribution of all estimated parameters, with their means and standard deviations shown in Table 3. The residuals of the estimated parameters reach up to 1.5%, and the standard deviations are as large as 0.66% of the true values. The uncertainties of the optimized parameters themselves are similar for all three frictional parameters. Specifically focusing on a and d_c, the residuals of the estimated parameters are larger than their standard deviations. This suggests that these discrepancies are primarily attributed to observation errors rather than the effect of initial parameters in the PINNs.

Figure 10c depicts the means and standard deviations of the estimated parameters for all cases. The variations in the means of estimated parameters, influenced by different first-guess values, are smaller compared to the variations caused by variations in the initial values of neural network parameters. This indicates that the choice of first-guess values for the frictional parameters does not significantly impact the estimation results. In summary, the estimated parameters are not significantly influenced by the first-guess values of initial parameters, despite the fact that the optimization trajectories are affected by the first-guess values.

Table 2. First-guess values using in frictional parameter estimation.

	Case A	Case B	Case C	Case D	Case E	Case F	Case G	Case H
a	× 10	× 10	× 10	× 10	× 0.1	× 0.1	× 0.1	× 0.1
a–b	× 10	× 10	× 0.1	× 0.1	× 10	× 10	× 0.1	× 0.1
d_c	× 10	× 0.1	× 10	× 0.1	× 10	× 0.1	× 10	× 0.1

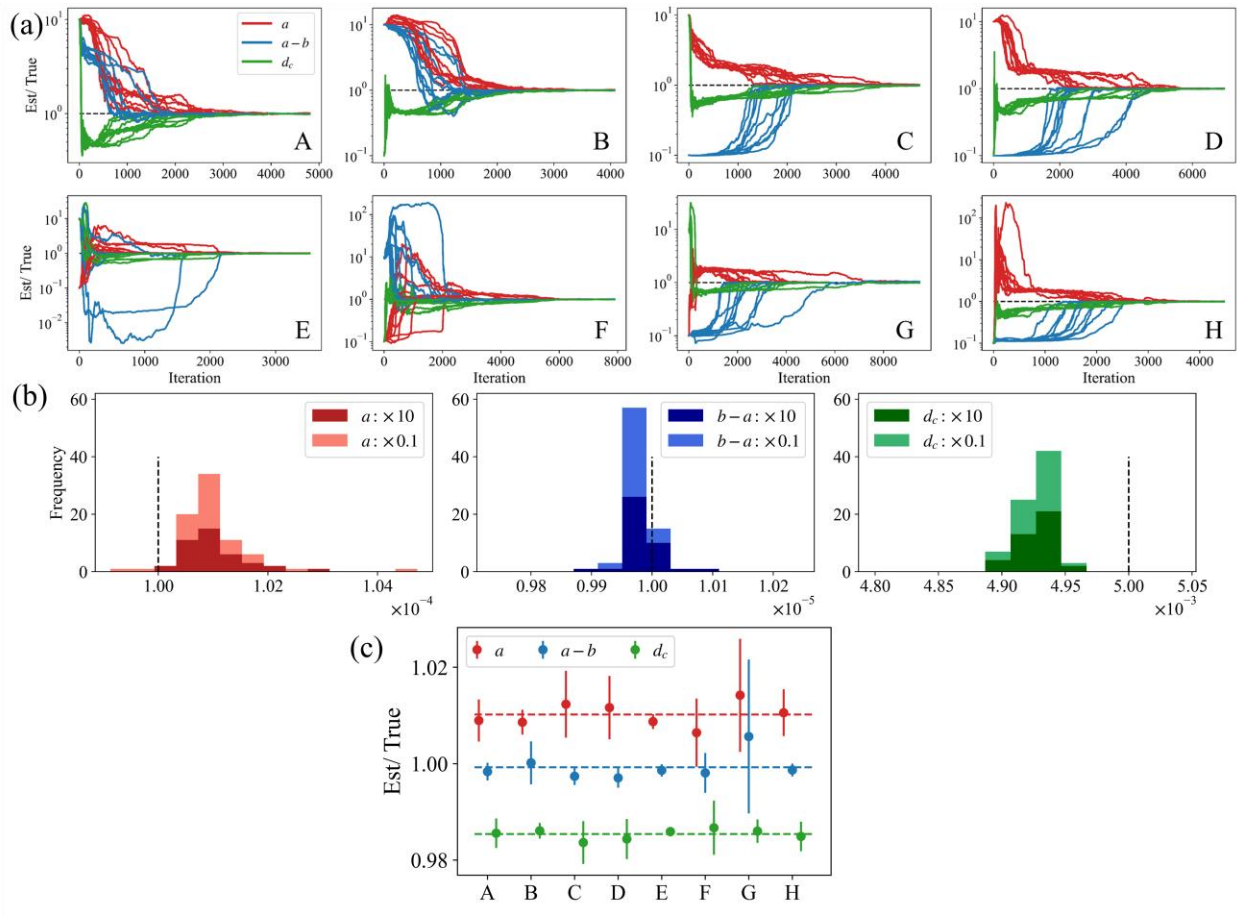


Figure 10. (a) Trajectories of estimated frictional parameters from the 10 different initial network parameters in cases A–H presented in Table 2, normalized by the true values. The red, blue, and green lines represent the values of a, a–b, and d_c, respectively. (b) The distribution of the estimated frictional parameters in all cases. Dark and light colors express the results starting the optimization process from smaller and larger first-guess values. The dashed lines indicate the true values. (c) Means of estimated frictional parameters in each case with their uncertainties. The length of the error bar indicates the standard deviation for each case and the dashed lines represent the means of estimated frictional parameters for all cases.

Table 3. Means and standard deviations of the estimated frictional parameters. The Error represents the relative error defined by |True value – Estimated value| / |True value|. The Ratio represents the standard deviation normalized by the true values.

	a	a–b	d_c[m]
Mean	1.010×10^{-4} Error: 1.0%	-0.999×10^{-5} Error: 0.1%	4.927×10^{-3} Error: 1.5%
Standard deviation	6.6×10^{-7} Ratio: 0.66%	6.4×10^{-8} Ratio: 0.64%	1.7×10^{-5} Ratio: 0.34%

5. Prediction of SSE evolution

For fault slip monitoring, it is crucial to predict the future temporal evolution of fault slip from the observation data. In this section, we attempt to estimate the frictional parameters from the observation data for an observation period shorter than the whole cycle of SSE and predict subsequent slip evolution.

5.1 Method

In Section 4, we estimated the frictional parameters from observation data over a full cycle (~800 days) by optimizing the neural network parameters and the frictional parameters to minimize the loss function. In this section, we consider situations where slip velocities are partially observed, meaning that SSE is currently ongoing, and we aim to predict its future evolution. This can be achieved simply by changing the data period in the loss function on observation data L_{data} (Eq. (15)) and by optimizing the neural network parameters and the frictional parameters utilizing the same loss function L (Eq. (14)). The collocation points are set at a constant time intervals of 100 h during one cycle, which is the same setting used in Section 4.

In this section, we use the observation data for the initial 400, 500, and 600 days of one cycle. The data period of 400 days corresponds to the timing before the slip velocity reaches the value of the loading velocity, v_{pl} (~1.58×10^{-9} [m/s]). We attempt to predict when the next SSE will occur based on the observations before a large slip occurs. The 500-day data period is the duration just before the slip velocity reaches its maximum, and we attempt to predict when the observed ongoing SSE will terminate. The 600-day period indicates the duration after the slip velocity decreases to the value of v_{pl}, and prediction from this data mainly focuses on estimating the frictional parameters from observation data after SSE has occurred. We generated three noisy synthetic data with a standard deviation of σ_{er} = 0.1 and intervals of 100 h, following the method described in Section 4.1.

We optimized the neural network and frictional parameters using eight different first-guess values (cases A-H) and repeated the optimization with 10 different initial parameters. The neural networks were trained until the change in the frictional parameters per optimization step was less than the threshold value of 10^{-6} or the iteration reached 20,000. Considering the difficulty of training due to a small number of data, the threshold value was set as 10^{-6}, which is smaller than the value of 10^{-5} utilized in Section 4.

5.2 Results and discussions

Figures 11a–c demonstrate the cases where future slip evolutions were successfully predicted using the data for 400, 500, and 600 days, respectively. Even when we use observation data for a shorter period such as 400 days, we succeeded in estimating the frictional parameters and predicting the temporal evolution of SSE well. However, we sometimes failed in parameter estimation as shown in Figures 11d–f. Figure 12 shows the histograms of all the estimated parameters from each observation data set. Focusing on the distribution of estimated parameters using observation data for 400 days, we can find a peak that is far from the true values, and the ratio of successful results is ~35%. Conversely, when using longer observation data, the frequency at the true values gradually increases, and the success ratio increases to ~50% and ~98% for the case of 500 and 600 days, respectively.

These results are interpreted as follows: Observation data for shorter periods do not sufficiently constrain the appropriate frictional parameters. As a result, depending on the first-guess value of the frictional parameters and initial neural network parameters, optimized parameters are likely to converge to incorrect values, resulting in inaccurate predictions. Figure 12 indicates that the success ratio dramatically increases when we use observation data for 600 days, suggesting that the observation data after the peak time of slip velocity are important to constrain the frictional parameters. The difficulty in parameter estimation prior to the SSE peak time has also been pointed out in the data-assimilation approach (Fujita, 2019) and is not exclusive to the PINN-based approach. This is inherent in the physics of fault slip and poses a critical problem for predicting fault slip evolution. Consequently, a stochastic approach is required to predict SSE before it occurs and, in this research, we evaluated the probability distribution of estimated frictional parameters by repeating the deterministic optimization with different initial values.

In this paper, we repeatedly referred to the uncertainties of PINNs in Sections 3.3, 4.3, and 5. For accurate uncertainty quantification of PINNs, Bayesian physics-informed neural networks (B-PINNs) have been proposed (Yang et al., 2021), and in seismology, they have been applied to hypocenter inversion (Izzatullah et al., 2022) and seismic tomography (Agata et al., 2023). B-PINNs treat neural network parameters as stochastic variables, enabling us to calculate the posterior probability parameters using Hamiltonian Monte Carlo or variational inference. Therefore, the application of B-PINNs in fault slip monitoring will become a powerful tool to evaluate the uncertainties of neural networks and estimated frictional parameters, enabling a more accurate stochastic prediction of fault slip evolution.

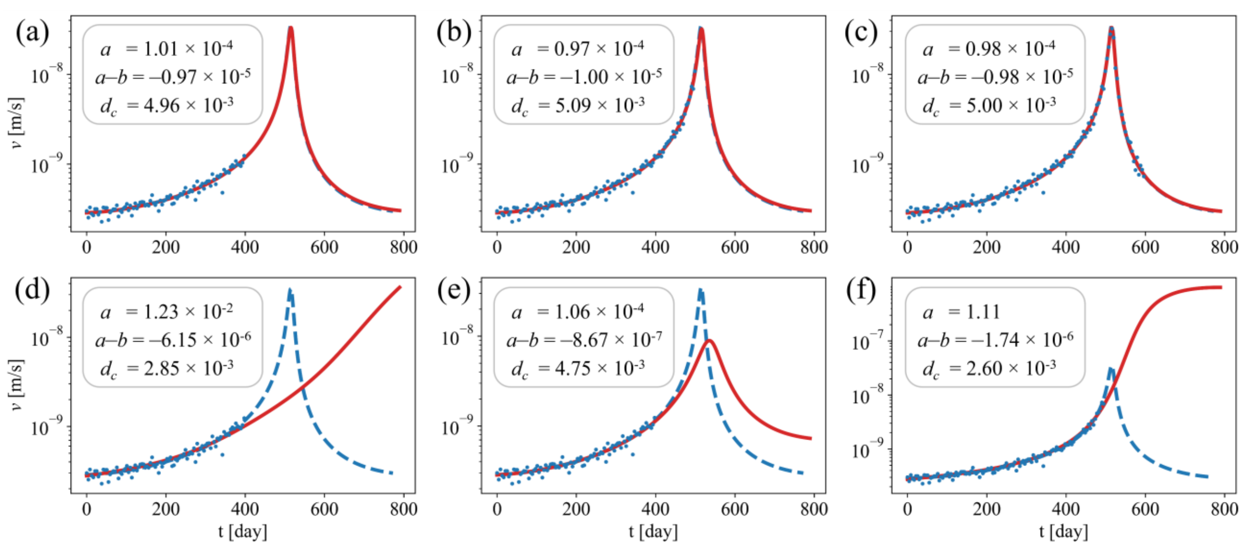


Figure 11. Examples of prediction results. The red and blue lines indicate the temporal evolution of v calculated by PINNs and the RK method. The blue points represent the synthetic data points, including observation errors. (a-c) Examples of successful prediction from observation data for (a) 400, (b) 500, and (c) 600 days. (d–f) Examples of unsuccessful prediction from observation data for (d–e) 400 and (f) 500 days.

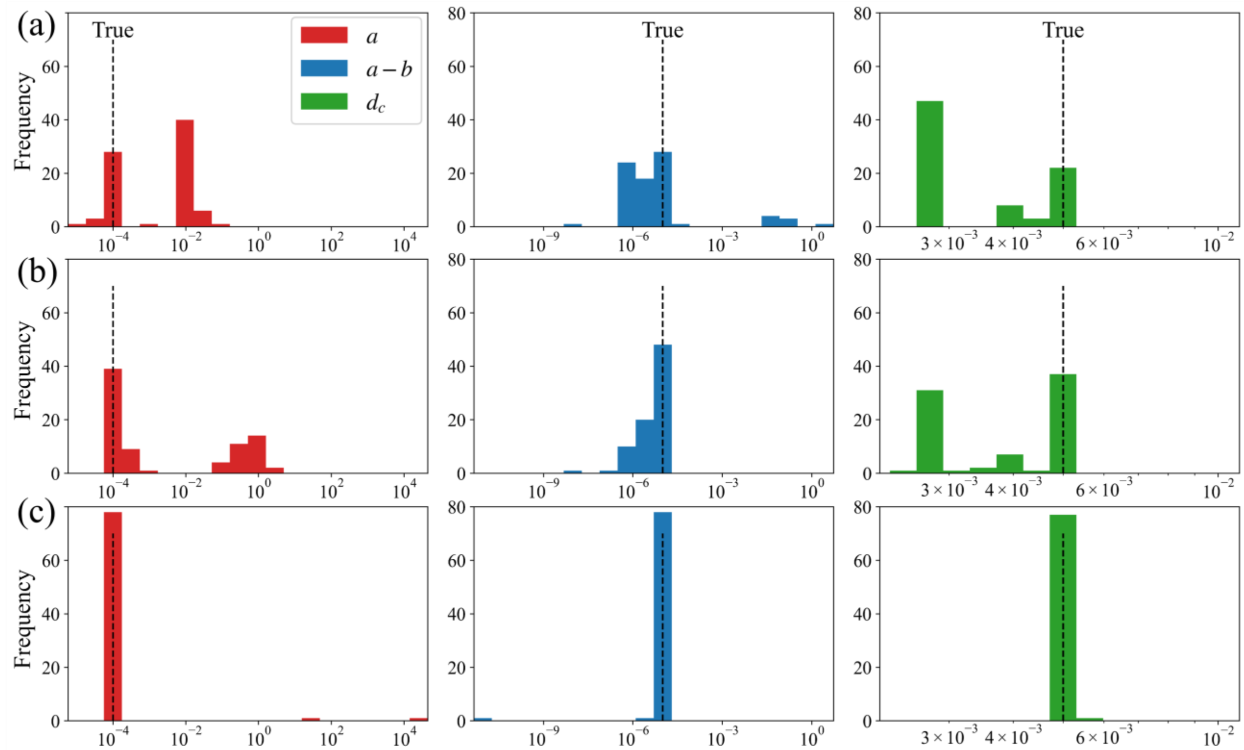


Figure 12. Histograms of the estimated frictional parameters from observation data for (a) 400, (b) 500, and (c) 600 days.

6. Conclusions

We proposed a new machine learning-based method for simulating, estimating frictional parameters, and predicting fault slips, and validated the effectiveness of this approach on slow slip events in a spring-slider system. In the forward simulation, PINNs accurately reproduced the temporal evolution of SSE, and the appropriate selection of collocation points played a crucial role in interpolating the residuals of equations. In frictional parameter estimation, the PINN-based approach successfully estimated the frictional parameters regardless of the first-guess values when using observation data for one cycle. For fault slip prediction, we achieved the evaluation of the probability of future fault slip using the PINN-based approach, and the likelihood of accurate fault slip prediction increased with longer observation periods. These results indicate that the PINN-based approach is highly effective for simulating fault slips, estimating frictional parameters, and predicting subsequent fault slips based on estimated parameters. Therefore, we strongly believe that PINNs have tremendous potential as a powerful tool for fault slip monitoring.

Acknowledgments

This work was supported by the MEXT Project for Seismology toward Research Innovation with Data of Earthquake (STAR-E) [Grant Number JPJ010217] and by the JSPS KAKENHI [Grant Numbers 18K03779, 21K03694, 23H00466, and 23K03552].

References

Agata, R., Shiraishi, K., & Fujie, G. (2023). Bayesian seismic tomography based on velocity-space Stein variation gradient descent for physics-informed neural network. *arXiv preprint arXiv: 2301.07901*. https://doi.org/10.48550/arXiv.2301.07901

Dieterich, J. H. (1979). Modeling of rock friction 1. Experimental results and constitutive equations. *Journal of Geophysical Research: Solid Earth*, 84, 2161–2168. https://doi.org/10.1029/JB084iB05p02161

Fujita, M. (2019). Estimation of the frictional parameters and slip evolution on fault with the ensemble Kalman filter: application to the Bungo long-term SSE (Master's thesis). Graduate School of Science, Kyoto University, Japan (in Japanese)

Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 9, 249–256. https://proceedings.mlr.press/v9/glorot10a.html

Guo, J., Wang, H., & Hou, C. (2022). A Novel Adaptive Causal Sampling Method for Physics-Informed Neural Networks. *arXiv preprint arXiv:2210.12914*. http://arxiv.org/abs/2210.12914

Hirahara, K., & Nishikiori, K. (2019). Estimation of frictional properties and slip evolution on a long-term slow slip event fault with the ensemble Kalman filter: numerical experiments. *Geophysical Journal International*, *219*(3), 2074–2096. https://doi.org/10.1093/gji/ggz415

Hirose, H., Hirahara, K., Kimata, F., Fujii, N., & Miyazaki, S. (1999). A slow thrust slip event following the two 1996 Hyuganada earthquakes beneath the Bungo Channel, southwest Japan. *Geophysical Research Letters*, *26*(21), 3237–3240. https://doi.org/10.1029/1999GL010999

Ito, S., Kano, M., & Nagao, H. (2022). Adjoint-based uncertainty quantification for inhomogeneous friction on a slow-slipping fault. *Geophysical Journal International*, *232*(1), 671–683. https://doi.org/10.1093/gji/ggac354

Izzatullah, M., Yildirim, I. E., Waheed, U. B., & Alkhalifah, T. (2022). Laplace HypoPINN: physics-informed neural network for hypocenter localization and its predictive uncertainty. *Machine Learning: Science and Technology*, *3*(4), 045001. https://doi.org/10.1088/2632-2153/ac94b3

Kano, M., Miyazaki, S., Ishikawa, Y., Hiyoshi, Y., Ito, K., & Hirahara, K. (2015). Real data assimilation for optimization of frictional parameters and prediction of afterslip in the 2003

Tokachi-oki earthquake inferred from slip velocity by an adjoint method. *Geophysical Journal International*, *203*(1), 646–663. https://doi.org/10.1093/gji/ggv289

Kano, M., Miyazaki, S., Ishikawa, Y., & Hirahara, K. (2020). Adjoint-based direct data assimilation of GNSS time series for optimizing frictional parameters and predicting postseismic deformation following the 2003 Tokachi-oki earthquake. *Earth, Planets and Space*, *72*(1). https://doi.org/10.1186/s40623-020-01293-0

Liu, D. C., & Nocedal, J. (1989). On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, *45*(1), 503–528. https://doi.org/10.1007/BF01589116

Miyazaki, S., Segall, P., McGuire, J. J., Kato, T., & Hatanaka, Y. (2006). Spatial and temporal evolution of stress and slip rate during the 2000 Tokai slow earthquake. *Journal of Geophysical Research: Solid Earth*, *111*(3). https://doi.org/10.1029/2004JB003426

Obara, K., & Kato, A. (2016). Connecting slow earthquakes to huge earthquakes. *Science*, *353*(6296), 253–257. https://doi.org/10.1126/science.aaf1512

Okazaki, T., Ito, T., Hirahara, K., & Ueda, N. (2022). Physics-informed deep learning approach for modeling crustal deformation. *Nature Communications*, *13*(1). https://doi.org/10.1038/s41467-022-34922-1

Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, *378*, 686–707. https://doi.org/10.1016/j.jcp.2018.10.045

Rasht-Behesht, M., Huber, C., Shukla, K., & Karniadakis, G. E. (2022). Physics-informed neural networks (PINNs) for wave propagation and full waveform inversions. *Journal of Geophysical Research: Solid Earth*, *127*(5). https://doi.org/10.1029/2021jb023120

Rice, J. R. (1993). Spatio-temporal complexity of slip on a fault. *Journal of Geophysical Research*, *98*(B6), 9885. https://doi.org/10.1029/93jb00191

Ruina, A. (1983). Slip Instability and State Variable Friction Laws. *Journal of Geophysical Research*, *88*(B12), 10359–10370. https://doi.org/10.1029/JB088iB12p10359

Segall, P. (2010). *Earthquake and Volcano Deformation*. Princeton University Press. https://doi.org/10.1515/9781400833856

Smith, J. D., Azizzadenesheli, K., & Ross, Z. E. (2021). EikoNet: Solving the Eikonal Equation With Deep Neural Networks. *IEEE Transactions on Geoscience and Remote Sensing*, *59*(12), 10685–10696. https://doi.org/10.1109/TGRS.2020.3039165

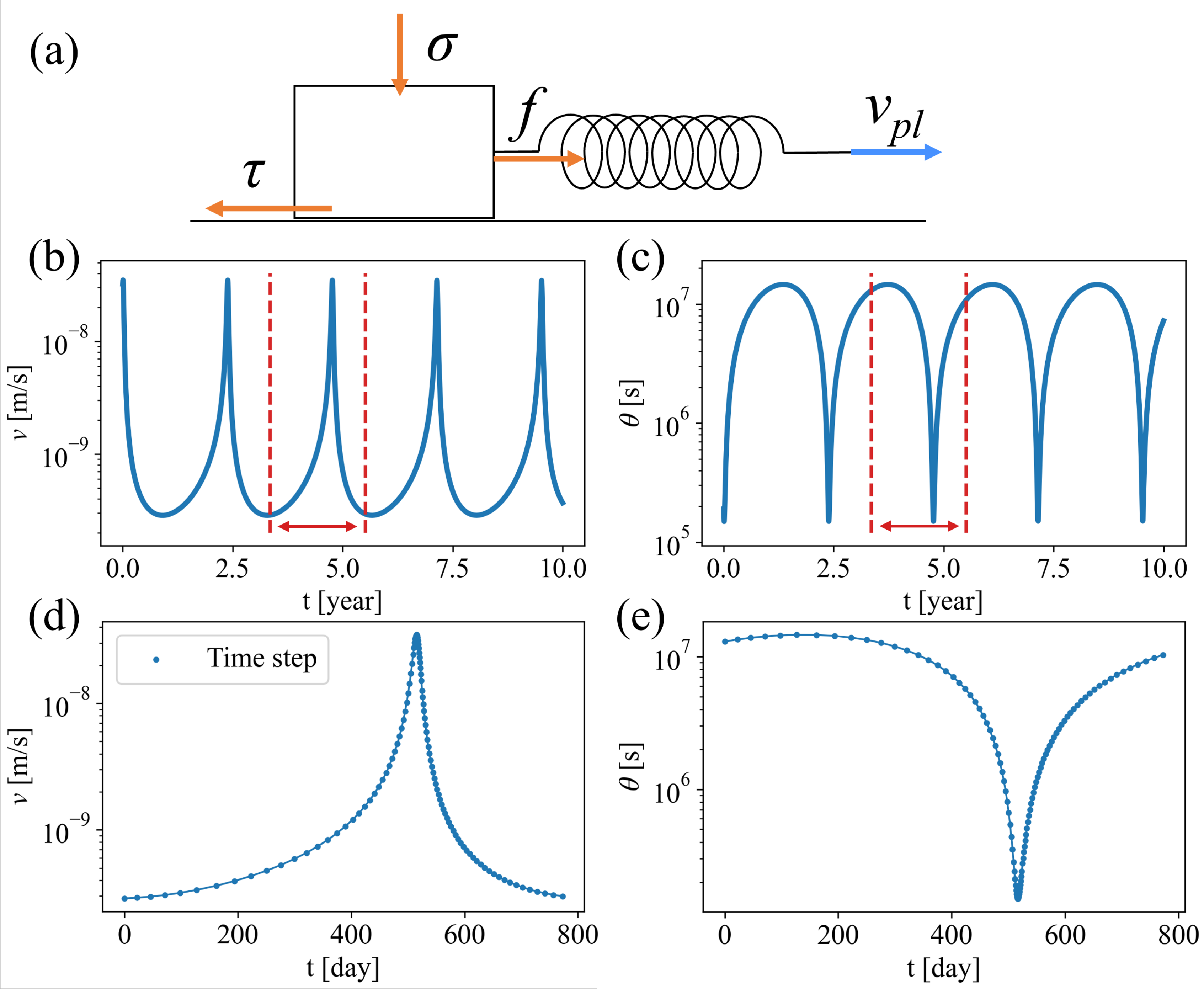
Smith, J. D., Ross, Z. E., Azizzadenesheli, K., & Muir, J. B. (2021). HypoSVI: Hypocentre inversion with Stein variational inference and physics informed neural networks. *Geophysical Journal International*, *228*(1), 698–710. https://doi.org/10.1093/gji/ggab309

Yang, L., Meng, X., & Karniadakis, G. E. (2021). B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data. *Journal of Computational Physics*, *425*, 109913. https://doi.org/10.1016/j.jcp.2020.109913

Yang, Y., & Perdikaris, P. (2019). Adversarial uncertainty quantification in physics-informed neural networks. *Journal of Computational Physics*, *394*, 136–152. https://doi.org/10.1016/j.jcp.2019.05.027

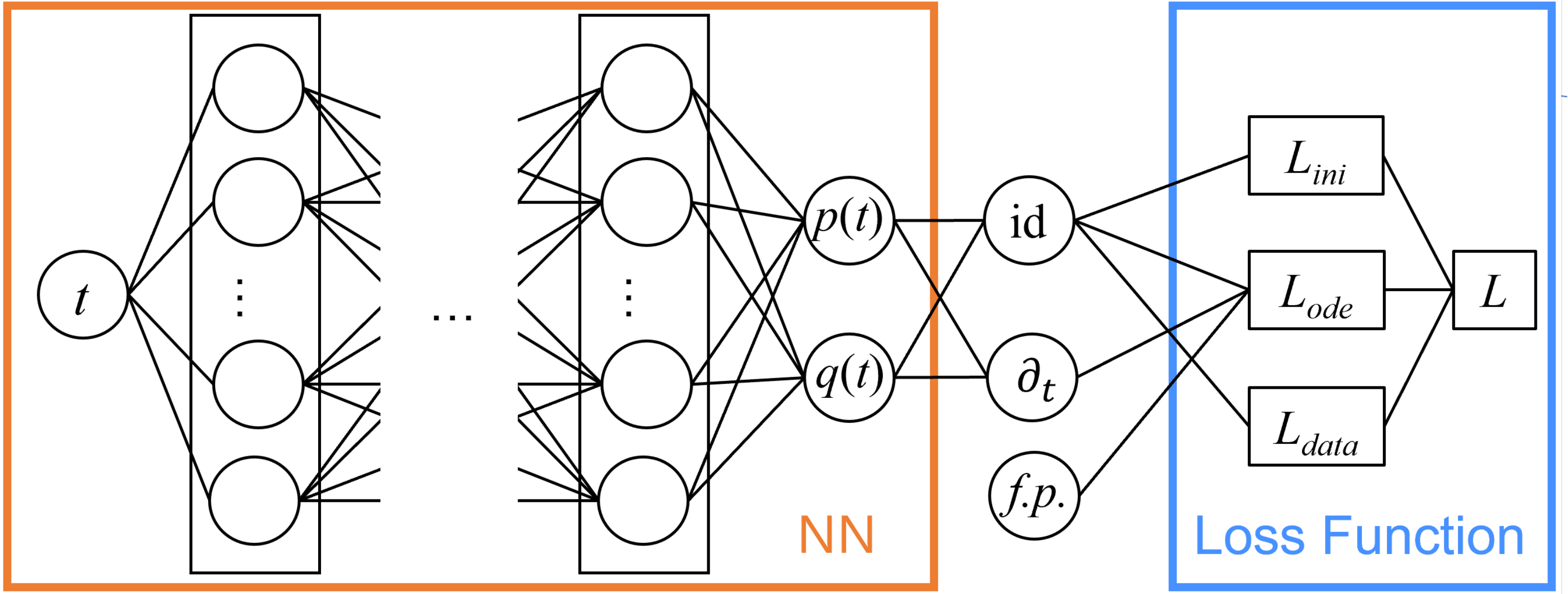
Yoshida, S., & Kato, N. (2003). Episodic aseismic slip in a two-degree-of-freedom block-spring model. *Geophysical Research Letters*, *30*(13), 1681. https://doi.org/10.1029/2003GL017439

Figure 1.



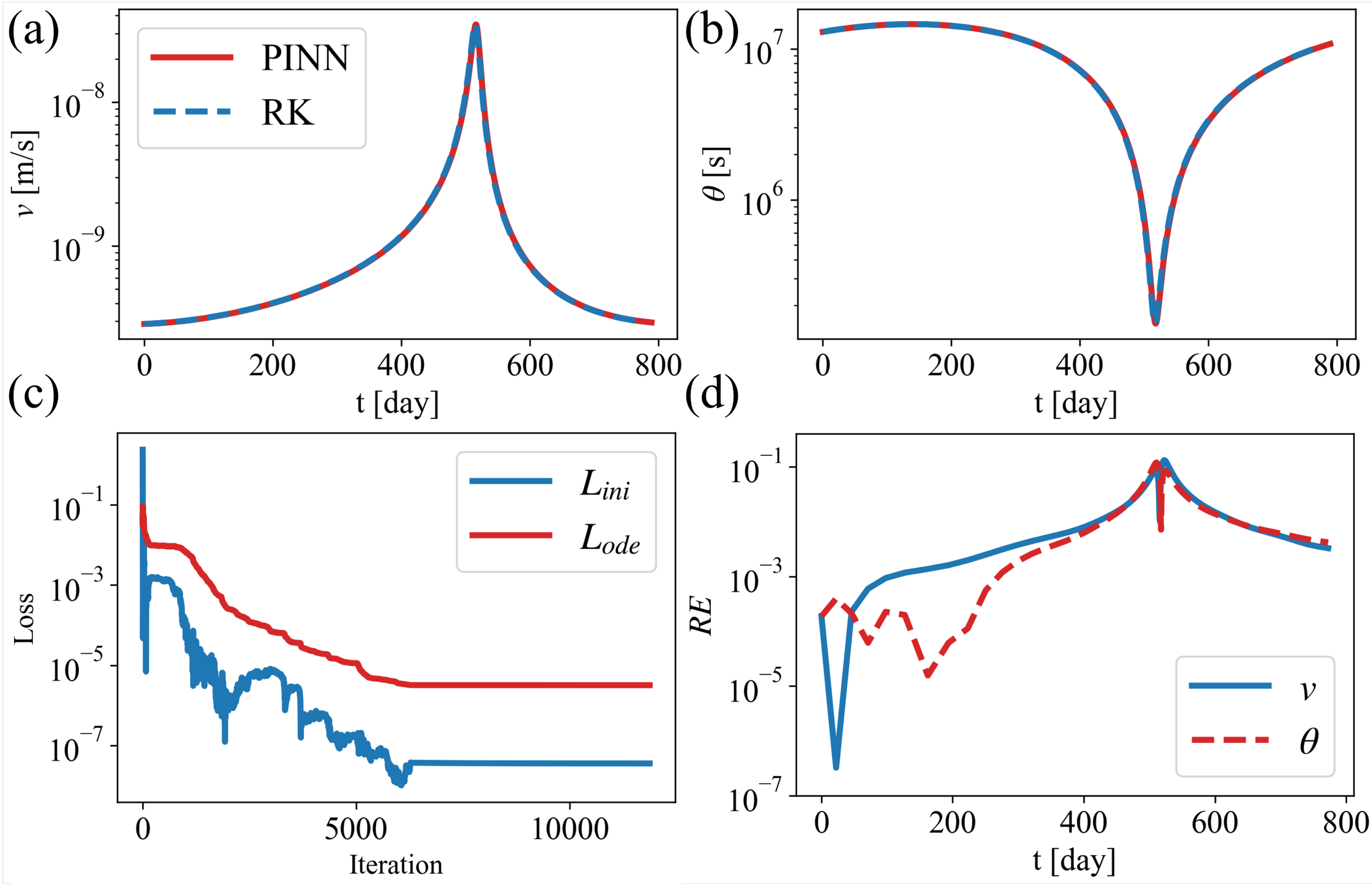
(a)
σ
f
v_{pl}
τ
(b)
v [m/s]
t [year]
(c)
θ [s]
t [year]
(d)
Time step
v [m/s]
t [day]
(e)
θ [s]
t [day]

Figure 2.



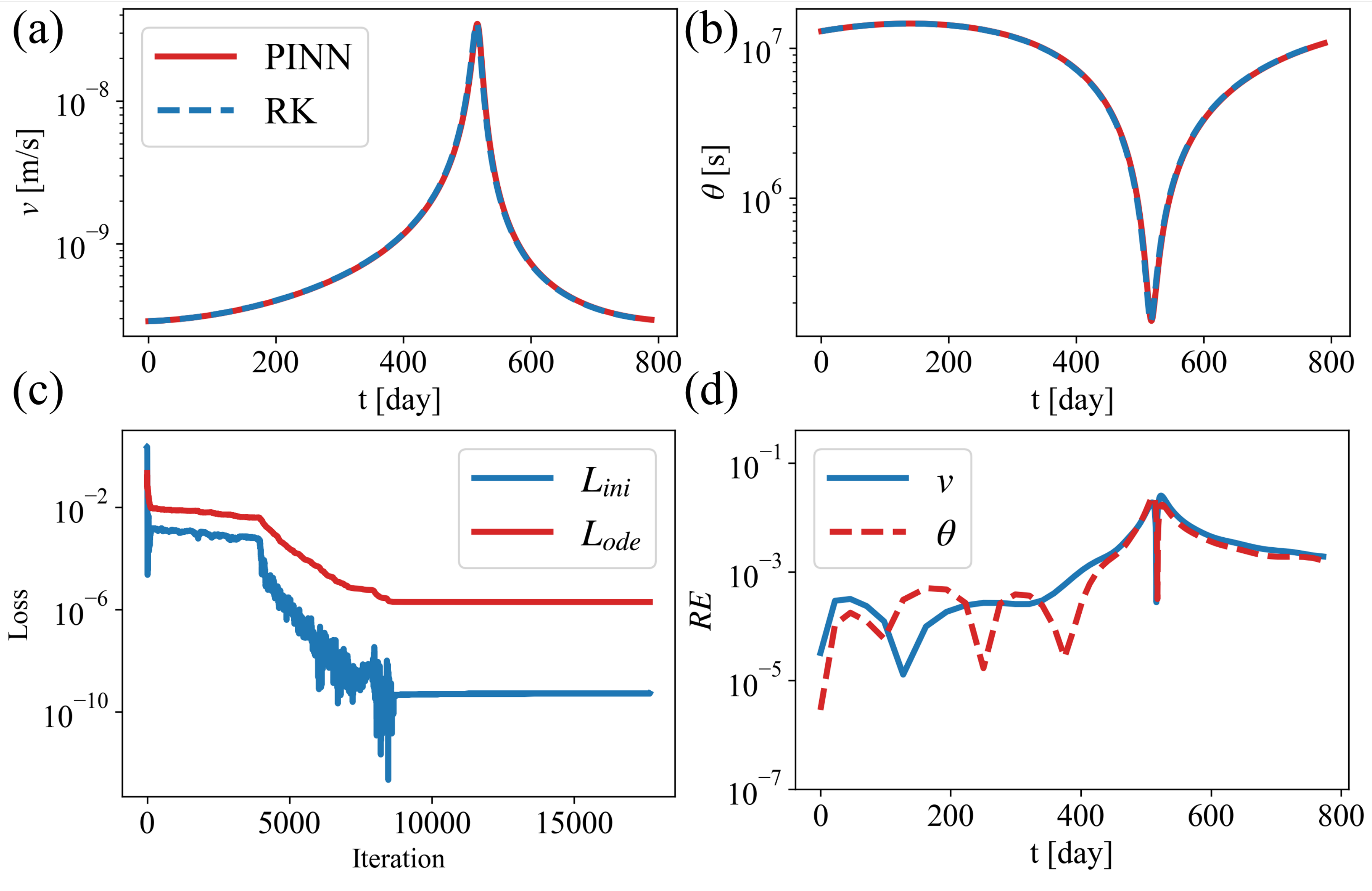
t
$\cdots$
$p(t)$
$q(t)$
id
∂_t
$f.p.$
NN
L_{ini}
L_{ode}
L_{data}
L
Loss Function

Figure 3.



(a)
PINN
RK
10^{-8}
10^{-9}
v [m/s]
0
200
400
600
800
t [day]
(b)
10^{7}
10^{6}
θ [s]
0
200
400
600
800
t [day]
(c)
L_{ini}
L_{ode}
10^{-1}
10^{-3}
10^{-5}
10^{-7}
Loss
0
5000
10000
Iteration
(d)
10^{-1}
10^{-3}
10^{-5}
10^{-7}
RE
v
θ
0
200
400
600
800
t [day]

Figure 4.



(a)
PINN
RK
10^{-8}
10^{-9}
v [m/s]
0
200
400
600
800
t [day]
(b)
10^{7}
10^{6}
θ [s]
0
200
400
600
800
t [day]
(c)
L_{ini}
L_{ode}
10^{-2}
10^{-6}
10^{-10}
Loss
0
5000
10000
15000
Iteration
(d)
v
θ
10^{-1}
10^{-3}
10^{-5}
10^{-7}
RE
0
200
400
600
800
t [day]

Figure 5.

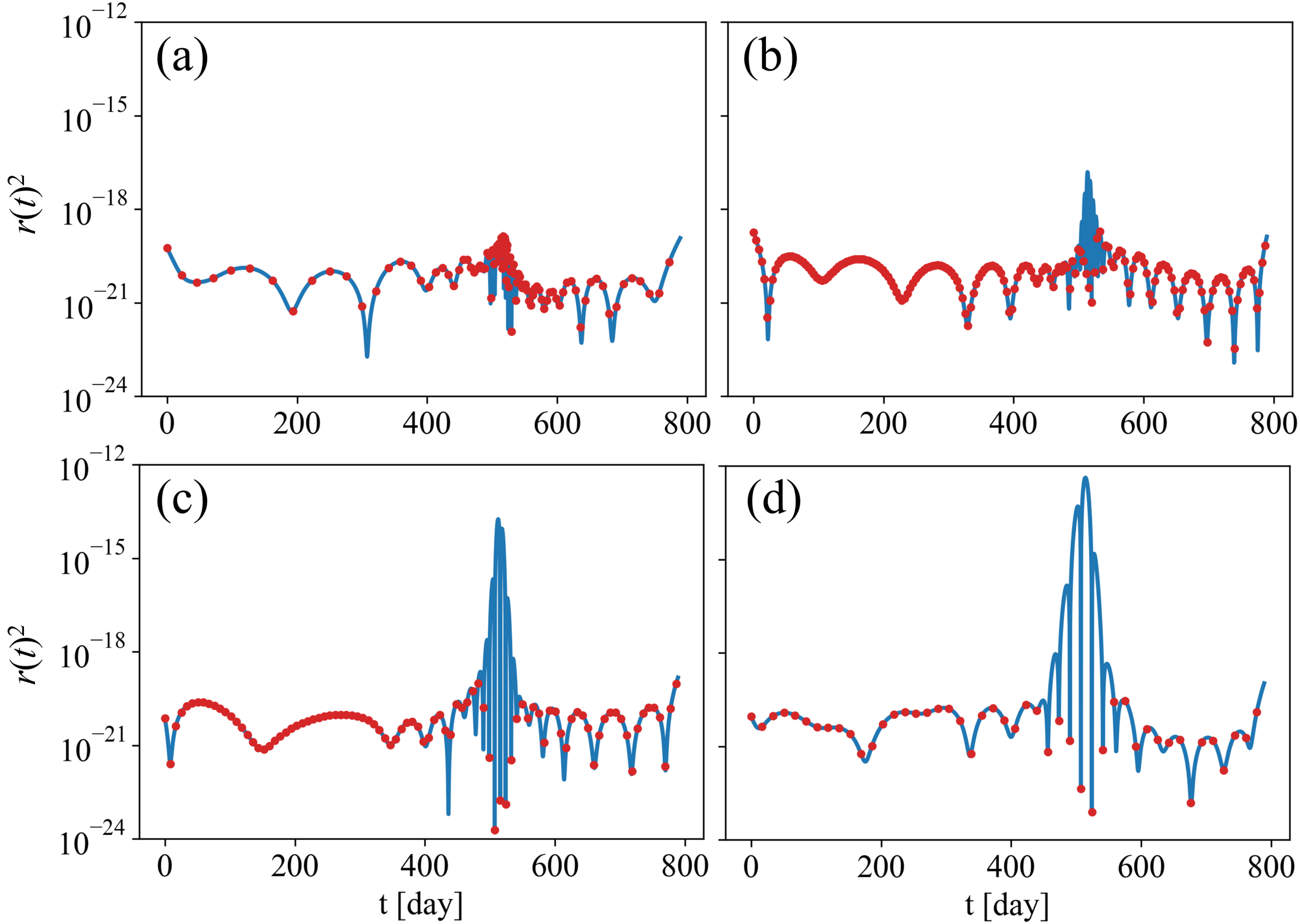
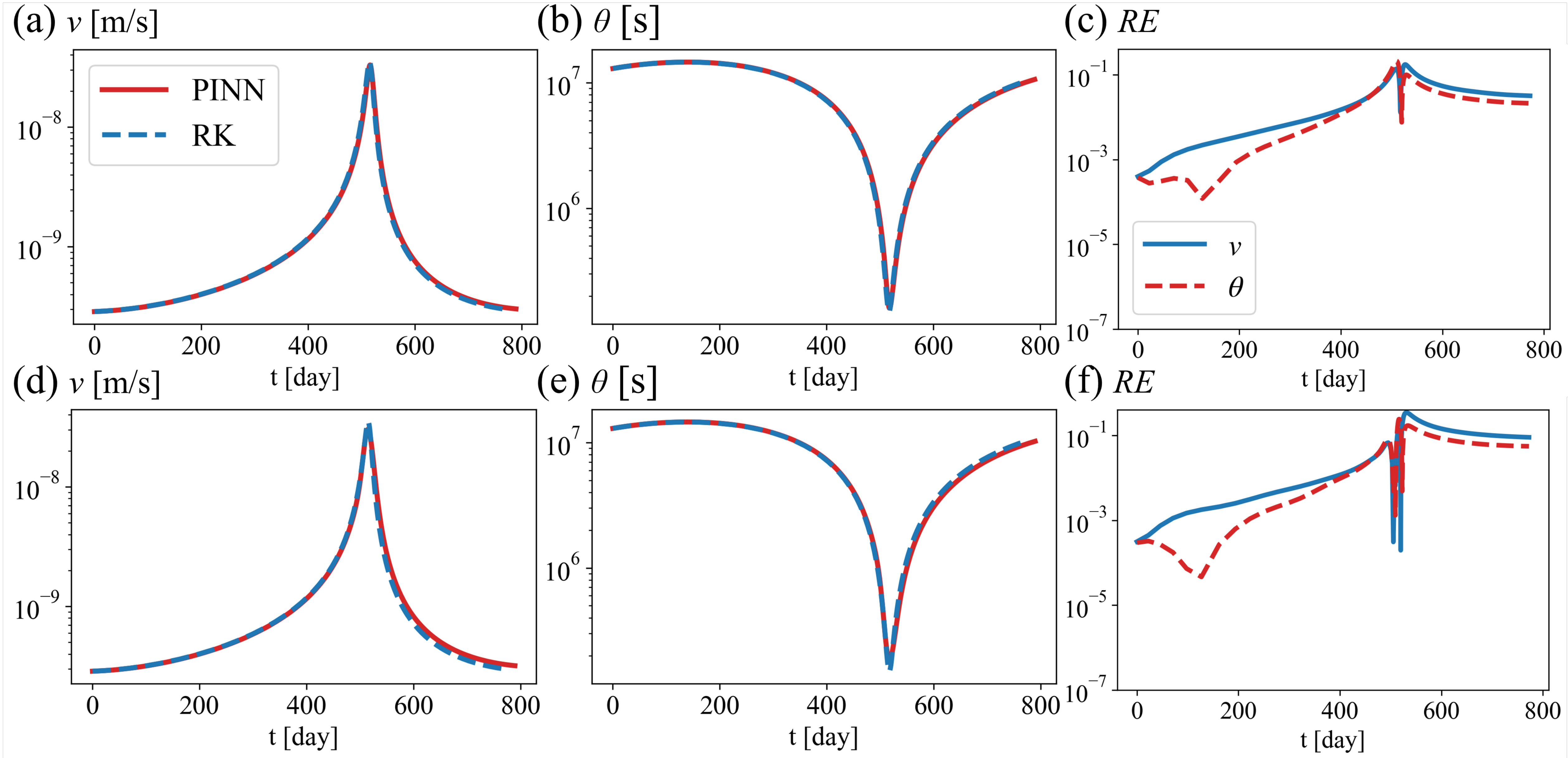
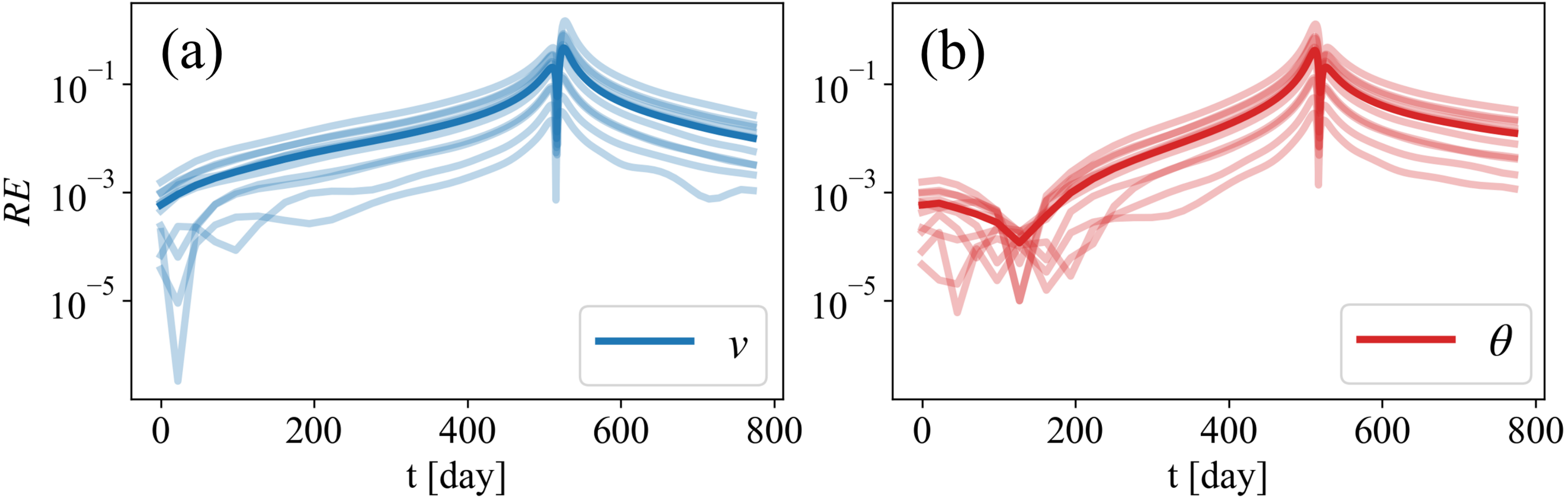

(a)
(b)
(c)
(d)
$r(t)^2$
10^{-12}
10^{-15}
10^{-18}
10^{-21}
10^{-24}
0
200
400
600
800
t [day]

Figure 6.



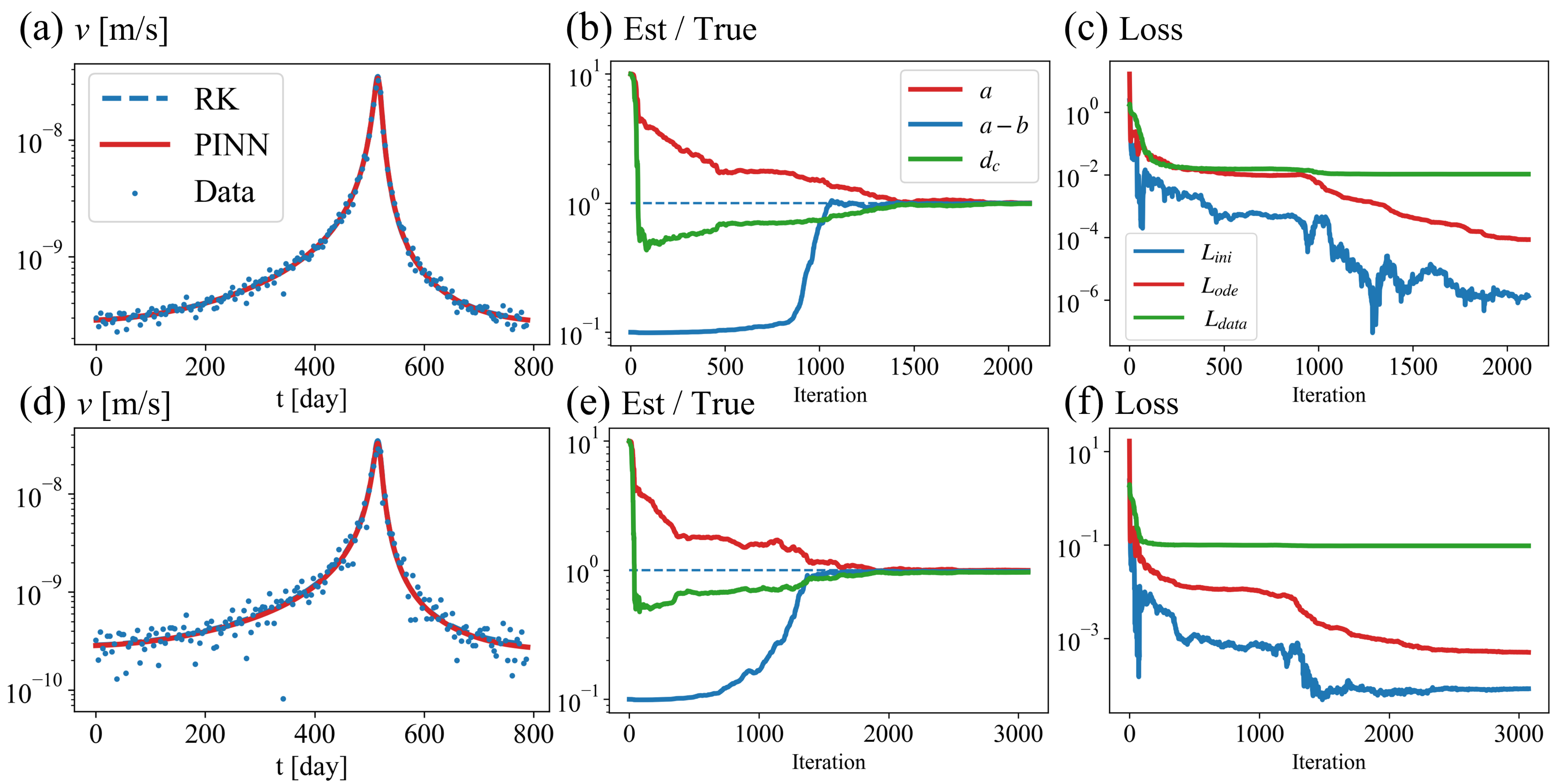
(a) v [m/s]
PINN
RK
10^−8
10^−9
0
200
400
600
800
t [day]
(b) θ [s]
10^7
10^6
0
200
400
600
800
t [day]
(c) RE
10^−1
10^−3
10^−5
10^−7
v
θ
0
200
400
600
800
t [day]
(d) v [m/s]
10^−8
10^−9
0
200
400
600
800
t [day]
(e) θ [s]
10^7
10^6
0
200
400
600
800
t [day]
(f) RE
10^−1
10^−3
10^−5
10^−7
0
200
400
600
800
t [day]

Figure 7.



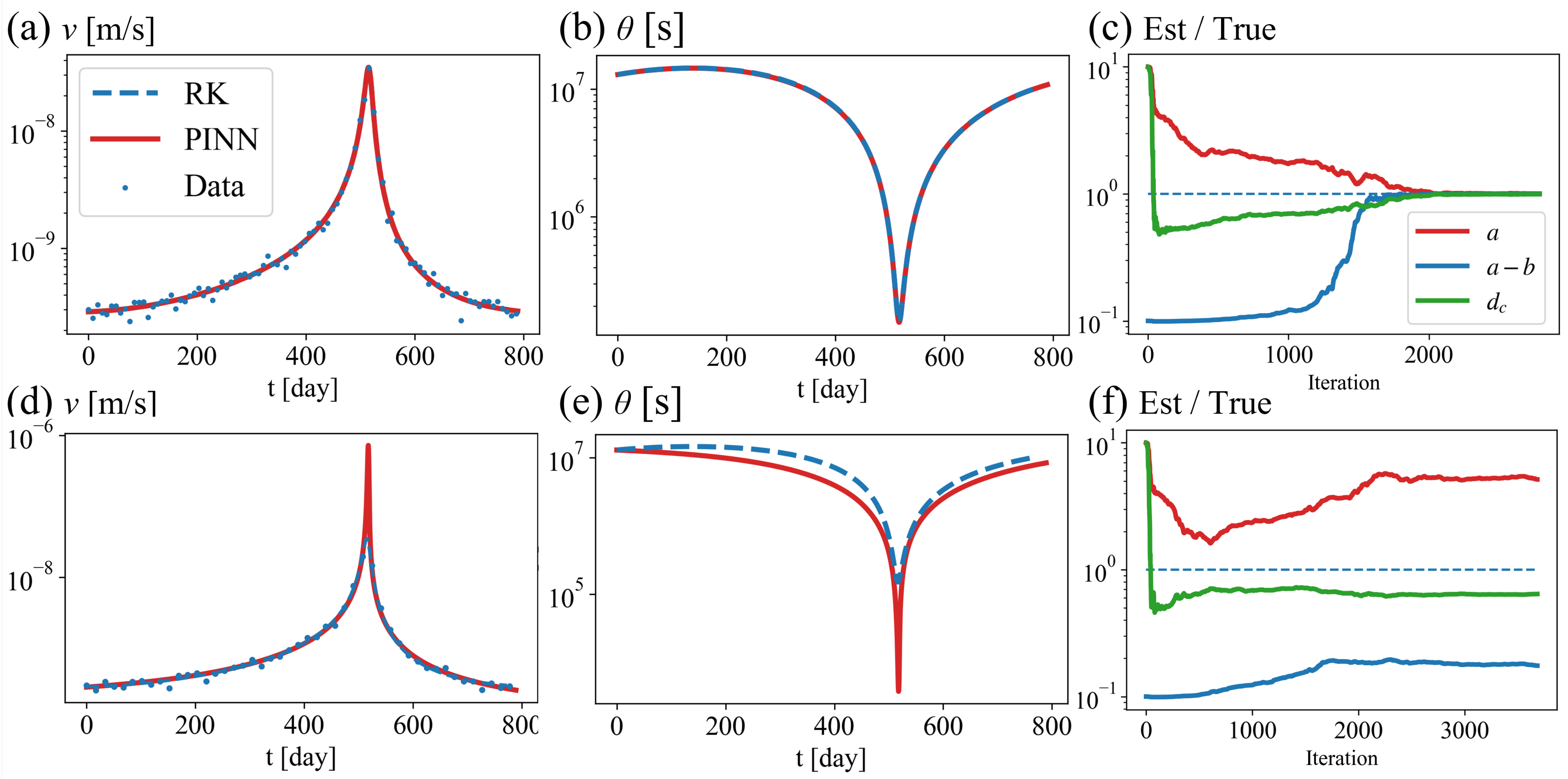
(a)
(b)
10^{-1}
10^{-3}
10^{-5}
RE
0
200
400
600
800
t [day]
v
θ

Figure 8.



(a) v [m/s]
RK
PINN
Data
t [day]
(b) Est / True
a
a − b
d_c
Iteration
(c) Loss
L_ini
L_ode
L_data
Iteration
(d) v [m/s]
t [day]
(e) Est / True
Iteration
(f) Loss
Iteration

Figure 9.



(a) v [m/s]
RK
PINN
Data
t [day]
(b) θ [s]
t [day]
(c) Est / True
a
a − b
d_c
Iteration
(d) v [m/s]
t [day]
(e) θ [s]
t [day]
(f) Est / True
Iteration

Figure 10.

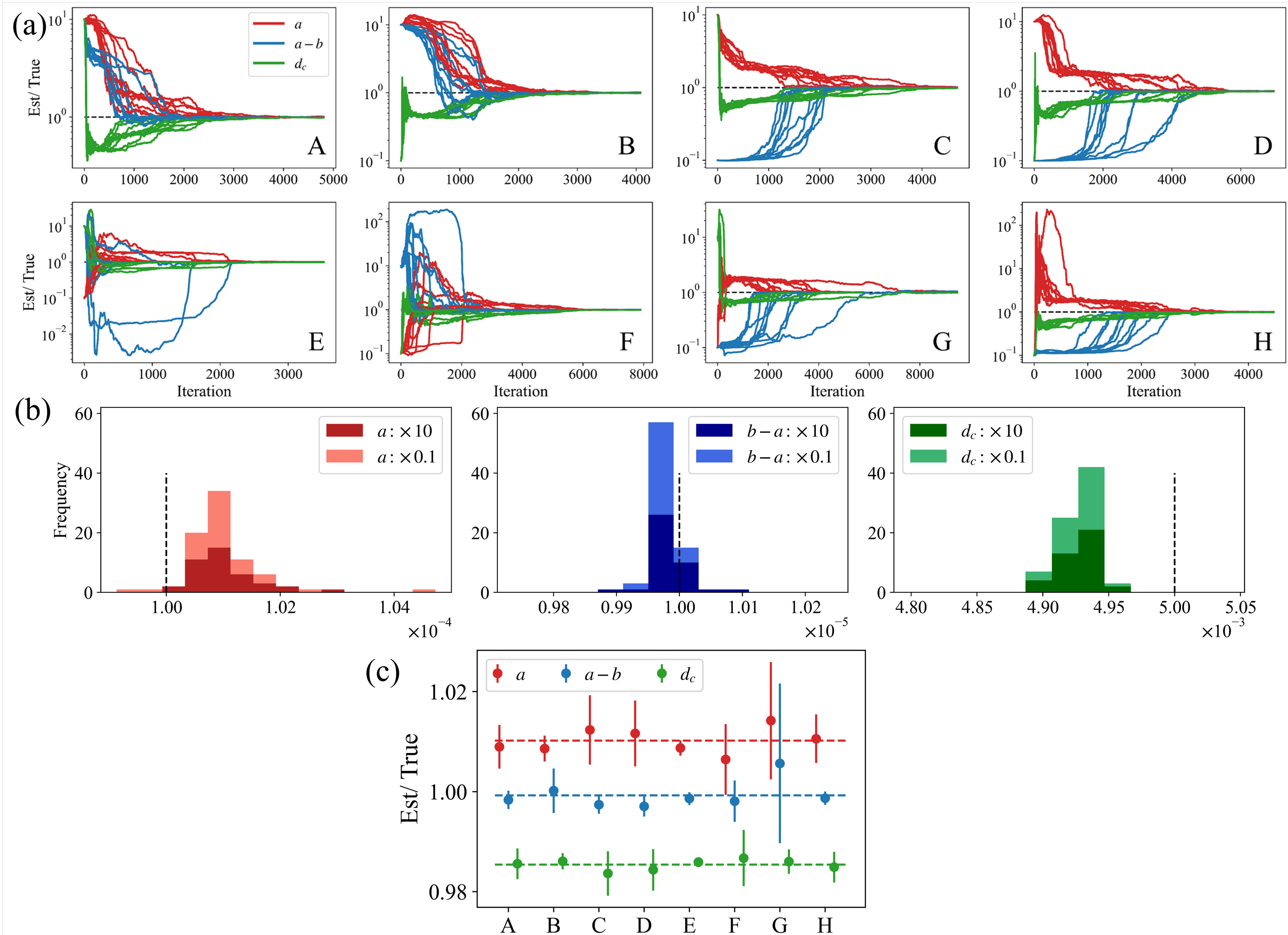
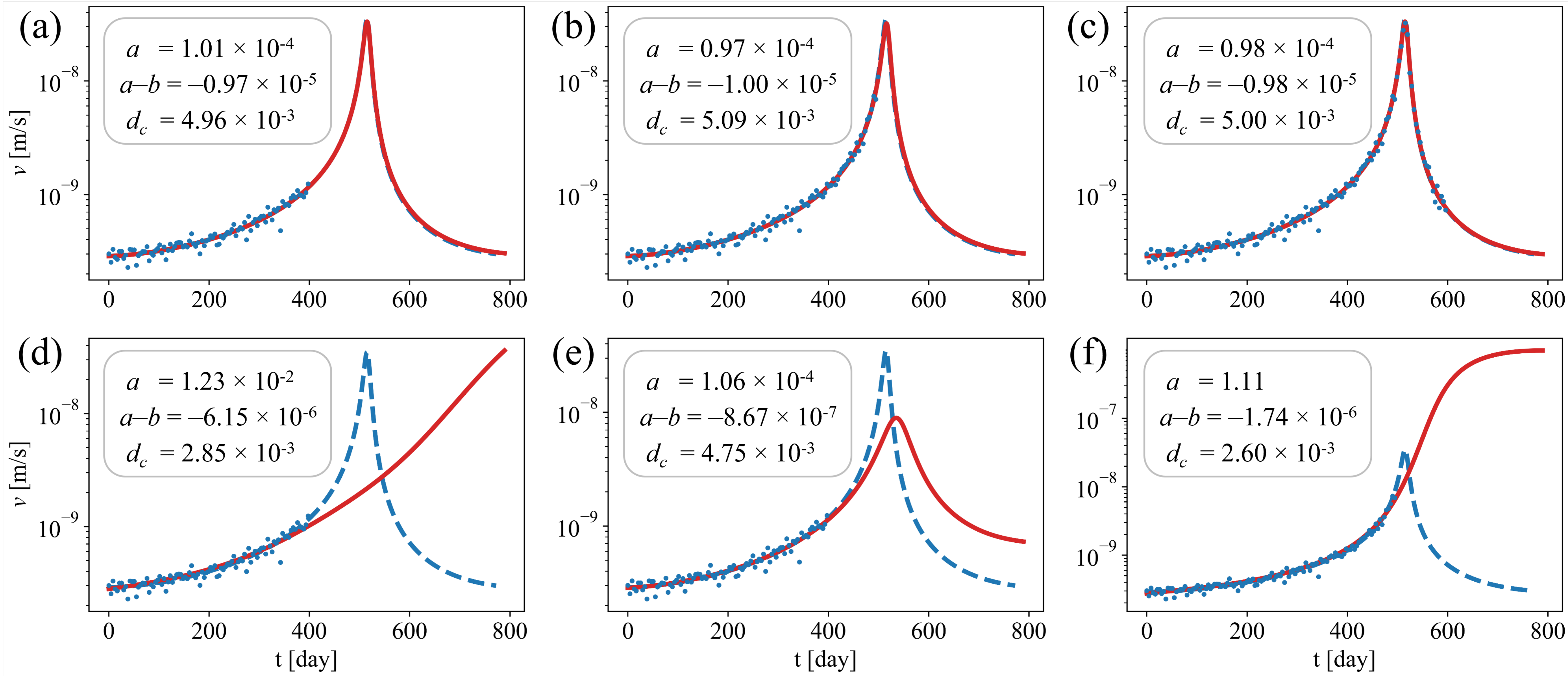
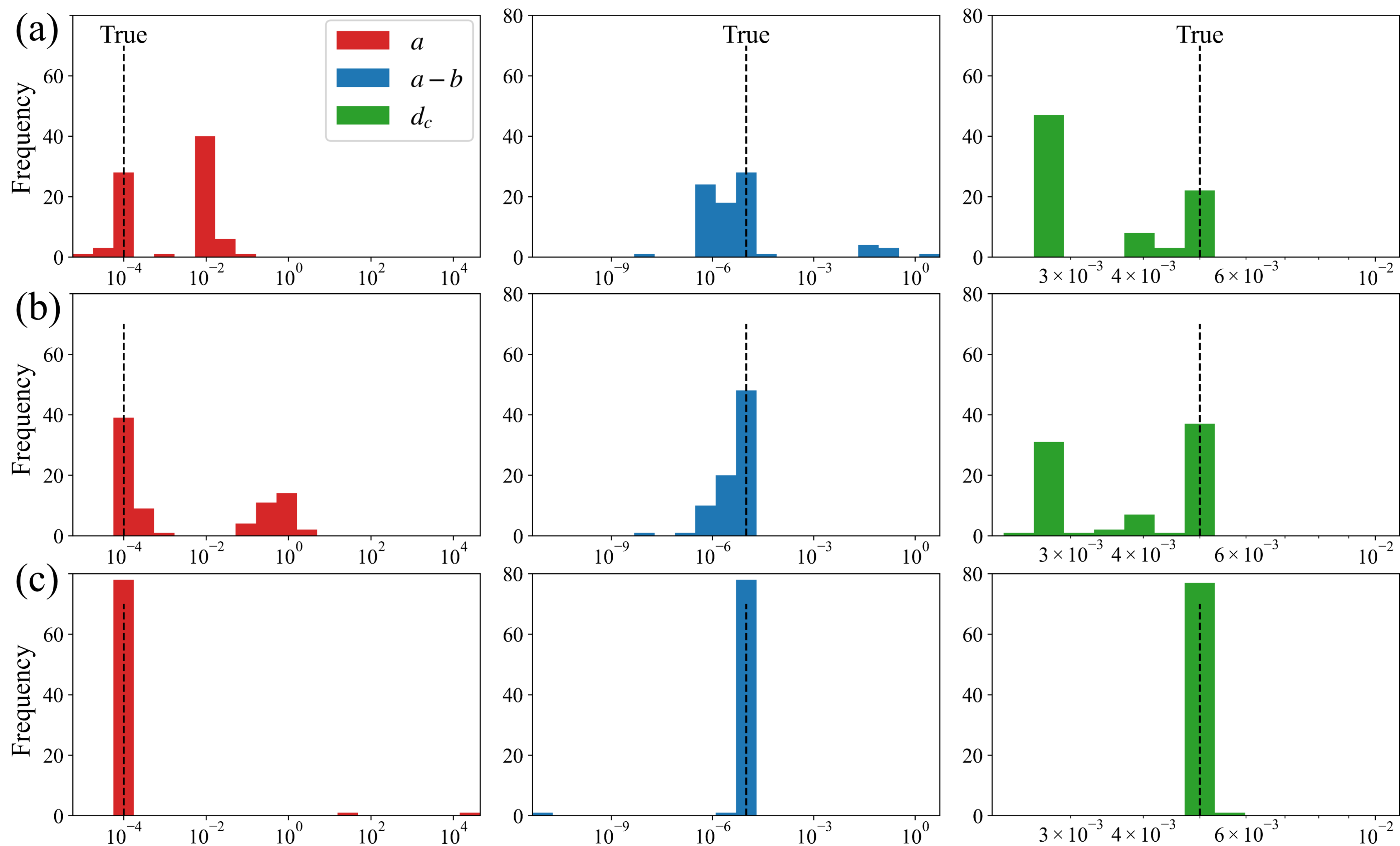

(a)
a
a − b
d_c
Est/ True
Iteration
A
B
C
D
E
F
G
H
(b)
Frequency
$a: \times 10$
$a: \times 0.1$
$b-a: \times 10$
$b-a: \times 0.1$
$d_c: \times 10$
$d_c: \times 0.1$
$\times 10^{-4}$
$\times 10^{-5}$
$\times 10^{-3}$
(c)
Est/ True
A B C D E F G H

Figure 11.



(a)
$a = 1.01 \times 10^{-4}$
$a-b = -0.97 \times 10^{-5}$
$d_c = 4.96 \times 10^{-3}$
(b)
$a = 0.97 \times 10^{-4}$
$a-b = -1.00 \times 10^{-5}$
$d_c = 5.09 \times 10^{-3}$
(c)
$a = 0.98 \times 10^{-4}$
$a-b = -0.98 \times 10^{-5}$
$d_c = 5.00 \times 10^{-3}$
(d)
$a = 1.23 \times 10^{-2}$
$a-b = -6.15 \times 10^{-6}$
$d_c = 2.85 \times 10^{-3}$
(e)
$a = 1.06 \times 10^{-4}$
$a-b = -8.67 \times 10^{-7}$
$d_c = 4.75 \times 10^{-3}$
(f)
$a = 1.11$
$a-b = -1.74 \times 10^{-6}$
$d_c = 2.60 \times 10^{-3}$
v [m/s]
t [day]
10^{-7}
10^{-8}
10^{-9}
0
200
400
600
800

Figure 12.



(a)
(b)
(c)
True
Frequency
a
$a-b$
d_c