SC15-009: Recent Advances in Physics-Informed Deep Learning

Physics-Informed Neural Networks (PINNs)
Part II: Discrete-time models

Paris Perdikaris
Department of Mechanical Engineering
University of Pennsylvania
email: pgp@seas.upenn.edu

Maziar Raissi
NVIDIA
email: maziar.raissi@gmail.com

USNCCM15
Austin, TX
July 28, 2019

Data-driven solution of PDEs

$$u_t + \mathcal{N}[u] = 0,\ x \in \Omega,\ t \in [0, T]$$

Subject to boundary and initial conditions.

Discrete Time Models

Let us employ the general form of Runge-Kutta methods with q stages and obtain

$$u^{n+c_i} = u^n - \Delta t \sum_{j=1}^{q} a_{ij} \mathcal{N}[u^{n+c_j}], \quad i = 1, \dots, q,$$
$$u^{n+1} = u^n - \Delta t \sum_{j=1}^{q} b_j \mathcal{N}[u^{n+c_j}].$$

Here, $u^{n+c_j}(x) = u(t^n + c_j \Delta t, x)$ for $j = 1, \dots, q$. This general form encapsulates both implicit and explicit time-stepping schemes, depending on the choice of the parameters $\{a_{ij}, b_j, c_j\}$. The above equations can be equivalently expressed as

$$u^n = u_i^n, \quad i = 1, \dots, q,$$
$$u^n = u_{q+1}^n,$$

where

$$u_i^n := u^{n+c_i} + \Delta t \sum_{j=1}^{q} a_{ij} \mathcal{N}[u^{n+c_j}], \quad i = 1, \dots, q,$$
$$u_{q+1}^n := u^{n+1} + \Delta t \sum_{j=1}^{q} b_j \mathcal{N}[u^{n+c_j}].$$

We proceed by placing a multi-output neural network prior on

$$\left[u^{n+c_1}(x), \dots, u^{n+c_q}(x), u^{n+1}(x) \right].$$

This prior assumption along with the above equations result in a physics informed neural network that takes x as an input and outputs

$$\left[u_1^n(x), \dots, u_q^n(x), u_{q+1}^n(x) \right].$$

Example (Allen-Cahn Equation)

This example aims to highlight the ability of the proposed discrete time models to handle different types of nonlinearity in the governing partial differential equation. To this end, let us consider the Allen-Cahn equation along with periodic boundary conditions

$$\begin{aligned} &u_t - 0.0001u_{xx} + 5u^3 - 5u = 0, \quad x \in [-1, 1], \quad t \in [0, 1], \\ &u(0, x) = x^2 \cos(\pi x), \\ &u(t, -1) = u(t, 1), \\ &u_x(t, -1) = u_x(t, 1). \end{aligned}$$

The Allen-Cahn equation is a well-known equation from the area of reaction-diffusion systems. It describes the process of phase separation in multi-component alloy systems, including order-disorder transitions. For the Allen-Cahn equation, the nonlinear operator is given by

$$\mathcal{N}[u^{n+c_j}] = -0.0001u_{xx}^{n+c_j} + 5\left(u^{n+c_j}\right)^3 - 5u^{n+c_j},$$

and the shared parameters of the neural networks can be learned by minimizing the sum of squared errors

$$SSE = SSE_n + SSE_b,$$

where

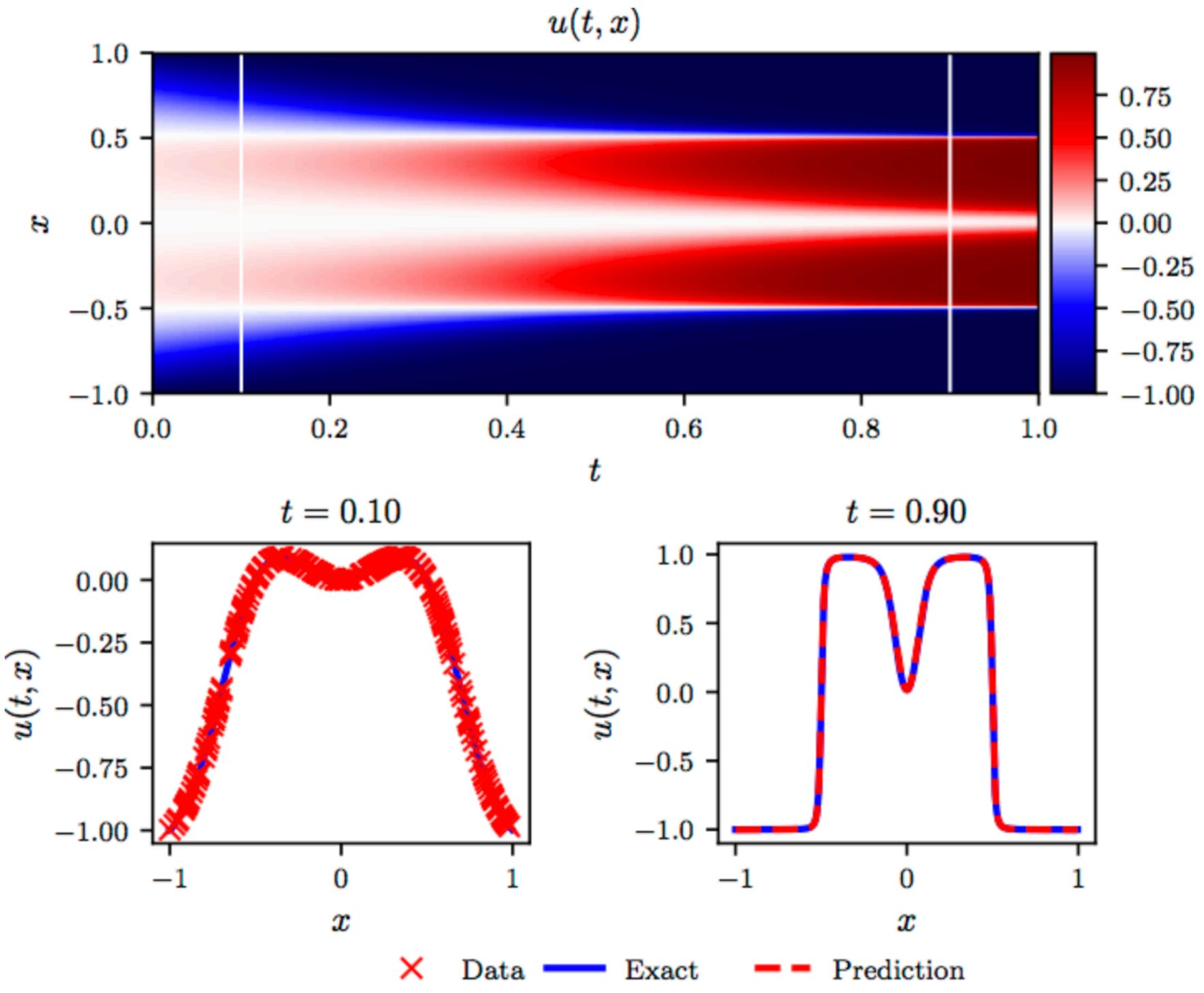
$$SSE_n = \sum_{j=1}^{q+1} \sum_{i=1}^{N_n} |u_j^n(x^{n,i}) - u^{n,i}|^2,$$

and

$$\begin{aligned} SSE_b = & \sum_{i=1}^{q} |u^{n+c_i}(-1) - u^{n+c_i}(1)|^2 + |u^{n+1}(-1) - u^{n+1}(1)|^2 \\ + & \sum_{i=1}^{q} |u_x^{n+c_i}(-1) - u_x^{n+c_i}(1)|^2 + |u_x^{n+1}(-1) - u_x^{n+1}(1)|^2. \end{aligned}$$

Here, $\{x^{n,i}, u^{n,i}\}_{i=1}^{N_n}$ corresponds to the data at time t^n.

The following figure summarizes our predictions after the network has been trained using the above loss function.



> *Allen-Cahn equation:* **Top:** Solution along with the location of the initial training snapshot at t=0.1 and the final prediction snapshot at t=0.9. **Bottom:** Initial training data and final prediction at the snapshots depicted by the white vertical lines in the top panel.

Data-driven discovery of PDEs

$$u_t + \mathcal{N}[u;\lambda] = 0,\ x \in \Omega,\ t \in [0,T],$$

Subject to boundary and initial conditions.

Discrete Time Models

We begin by employing the general form of Runge-Kutta methods with q stages and obtain

$$u^{n+c_i} = u^n - \Delta t \sum_{j=1}^{q} a_{ij} \mathcal{N}[u^{n+c_j}; \lambda], \quad i = 1, \dots, q,$$
$$u^{n+1} = u^n - \Delta t \sum_{j=1}^{q} b_j \mathcal{N}[u^{n+c_j}; \lambda].$$

Here, $u^{n+c_j}(x) = u(t^n + c_j \Delta t, x)$ for $j = 1, \dots, q$. This general form encapsulates both implicit and explicit time-stepping schemes, depending on the choice of the parameters $\{a_{ij}, b_j, c_j\}$. The above equations can be equivalently expressed as

$$u^n = u_i^n, \quad i = 1, \dots, q,$$
$$u^{n+1} = u_i^{n+1}, \quad i = 1, \dots, q.$$

where

$$u_i^n := u^{n+c_i} + \Delta t \sum_{j=1}^{q} a_{ij} \mathcal{N}[u^{n+c_j}; \lambda], \quad i = 1, \dots, q,$$
$$u_i^{n+1} := u^{n+c_i} + \Delta t \sum_{j=1}^{q} (a_{ij} - b_j) \mathcal{N}[u^{n+c_j}; \lambda], \quad i = 1, \dots, q.$$

We proceed by placing a multi-output neural network prior on

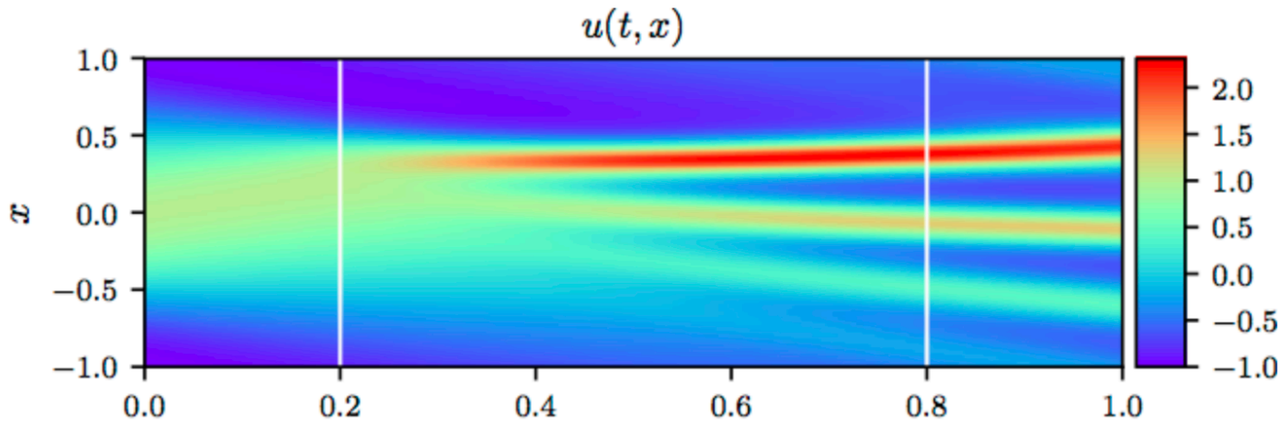
$$\left[u^{n+c_1}(x), \dots, u^{n+c_q}(x) \right].$$

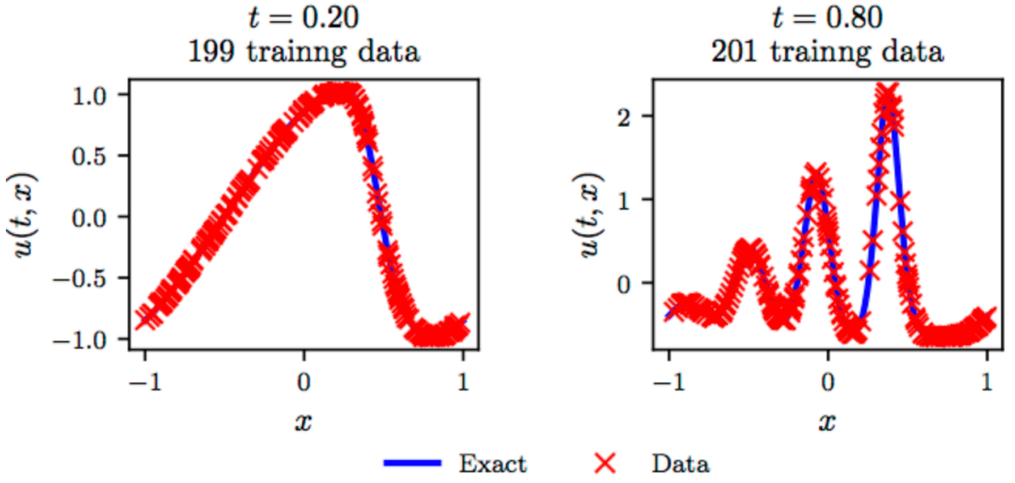
This prior assumption result in two physics informed neural networks

$$\left[u_1^n(x), \dots, u_q^n(x), u_{q+1}^n(x) \right],$$

and

$$\left[u_1^{n+1}(x), \dots, u_q^{n+1}(x), u_{q+1}^{n+1}(x) \right].$$





Given noisy measurements at two distinct temporal snapshots $\{\mathbf{x}^n, \mathbf{u}^n\}$ and $\{\mathbf{x}^{n+1}, \mathbf{u}^{n+1}\}$ of the system at times t^n and t^{n+1}, respectively, the shared parameters of the neural networks along with the parameters λ of the differential operator can be trained by minimizing the sum of squared errors

$$SSE = SSE_n + SSE_{n+1},$$

where

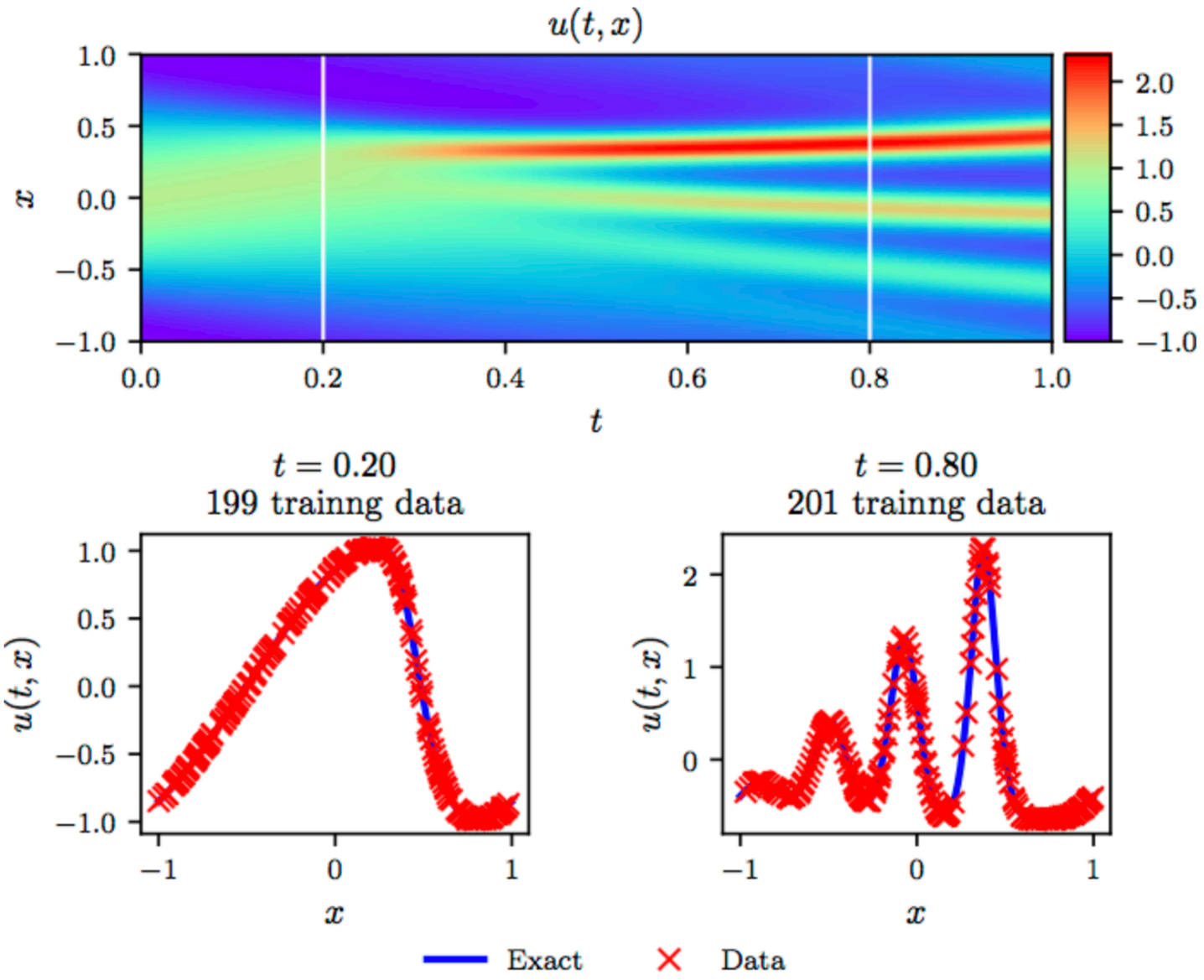
$$SSE_n := \sum_{j=1}^{q} \sum_{i=1}^{N_n} |u_j^n(x^{n,i}) - u^{n,i}|^2,$$

and

$$SSE_{n+1} := \sum_{j=1}^{q} \sum_{i=1}^{N_{n+1}} |u_j^{n+1}(x^{n+1,i}) - u^{n+1,i}|^2.$$

Here, $\mathbf{x}^n = \left\{x^{n,i}\right\}_{i=1}^{N_n}$, $\mathbf{u}^n = \left\{u^{n,i}\right\}_{i=1}^{N_n}$, $\mathbf{x}^{n+1} = \left\{x^{n+1,i}\right\}_{i=1}^{N_{n+1}}$, and $\mathbf{u}^{n+1} = \left\{u^{n+1,i}\right\}_{i=1}^{N_{n+1}}$.

The results of this experiment are summarized in the following figure.



Correct PDE	$u_t + uu_x + 0.0025u_{xxx} = 0$
Identified PDE (clean data)	$u_t + 1.000uu_x + 0.0025002u_{xxx} = 0$
Identified PDE (1% noise)	$u_t + 0.999uu_x + 0.0024996u_{xxx} = 0$

> *KdV equation:* **Top:** Solution along with the temporal locations of the two training snapshots. Middle: Training data and exact solution corresponding to the two temporal snapshots depicted by the dashed vertical lines in the top panel. **Bottom:** Correct partial differential equation along with the identified one.