Response to Reviewer itQX

Comment — Lack of Model Details and Hyperparameter Settings

The paper does not provide sufficient detail on the architecture and hyperparameter configurations of cd-PINN, such as the number of layers, model parameters, or training configurations. Given that cd-PINN is proposed as a robust solution for a wide variety of differential equations, these details are essential for reproducibility and for evaluating the computational cost of deeper or larger models. Including these specifications would enable researchers to replicate results and better understand how model depth and complexity impact generalization and accuracy.

Reply: Thank you for the comment. To address this issue, we make a comprehensive summary on the architecture and hyperparameter configurations of our cd-PINN, such as the number of layers, model parameters as well as training and test configurations, in Table C4 in the appendix.

Changes: See Appendix C and Table C4 for details.

Table 1: Setup of cd-PINN for models tested in this study.

Problem	Depth	Width	Optimizer	Learning rate	# Iterations	Collocation Points
Logistic equation	6	128	Adam + L-BFGS	1×10^{-4}	50,000	32,768
LV scenario 1	6	128	Adam + L-BFGS	1×10^{-3}	500	32,768
LV scenario 2	6	128	Adam + L-BFGS	1×10^{-4}	100,000	32,768
LV scenario 3	6	128	Adam + L-BFGS	1×10^{-3}	500	32,768
Underdamped oscillator	6	128	Adam	1×10^{-3}	50,000	32,768
Critically damped oscillator	6	128	Adam	1×10^{-3}	20,000	32,768
Overdamped oscillator	6	128	Adam	1×10^{-3}	50,000	32,768
P53 activation	6	128	Adam + L-BFGS	1×10^{-4}	10,000	16,384

Comment — Limited Scope in Differential Equation Types: Absence of Fundamental PDEs and Comparison with Neural ODEs

The current focus on ordinary differential equations (ODEs) in the paper feels restrictive, especially given the model's "PINN" designation, which implies potential applicability to a broader class of partial differential equations (PDEs) involving other derivatives, such as convection-diffusion-reaction (CDR) equations. Extending the study to include such fundamental PDEs would demonstrate cd-PINN's flexibility and relevance in more complex, real-world applications. Furthermore, comparing cd-PINN's performance with Neural ODE models would clarify its advantages and limitations, as Neural ODEs are another prominent approach for learning solutions to ordinary differential equations. Including these comparisons would provide a more comprehensive picture of where cd-PINN stands in relation to existing methods.

Reply: (1) The basic idea of continuous dependence is still applicable to PDEs, which is widely known as the well-posedness of PDEs. We found that by including the requirements of continuous dependence into the loss function, the generality of cd-PINN could be dramatically improved when compared to classical PINNs. Here we present an example of cd-PINN applied to the diffusion equations.

Consider the following Cauchy problem for the $1+1$ dimensional diffusion equation:

$$\begin{aligned}&\frac{\partial u}{\partial t}-D\frac{\partial^2 u}{\partial x^2}=0,\ x\in[-10.0,10.0],\ t\in[0.1,1.1],\\&u(x,0)=\frac{1}{\sqrt{2\pi}\sigma}\exp\Big(-\frac{(x-\mu)^2}{2\sigma^2}\Big)+\frac{1}{\sqrt{2\pi}\sigma_1}\exp\Big(-\frac{(x-\mu_1)^2}{2\sigma_1^2}\Big),\ x\in[-10.0,10.0].\end{aligned}\tag{1}$$

In practice, we set $\sigma_1=1.0$ and $\mu_1=5.0$, and use 20 real data points with $\sigma=1.0,\mu=0.0$ as labeled training data, along with 2^{14} unlabeled residual data points. To assess cd-PINN's generalization ability, we selected 10100 configurations of values for $\sigma\in[0.1,10.0]$ and $\mu\in[-5.0,5.0]$ to generate corresponding solutions as the test data set, with a total of 8080000 test data points.

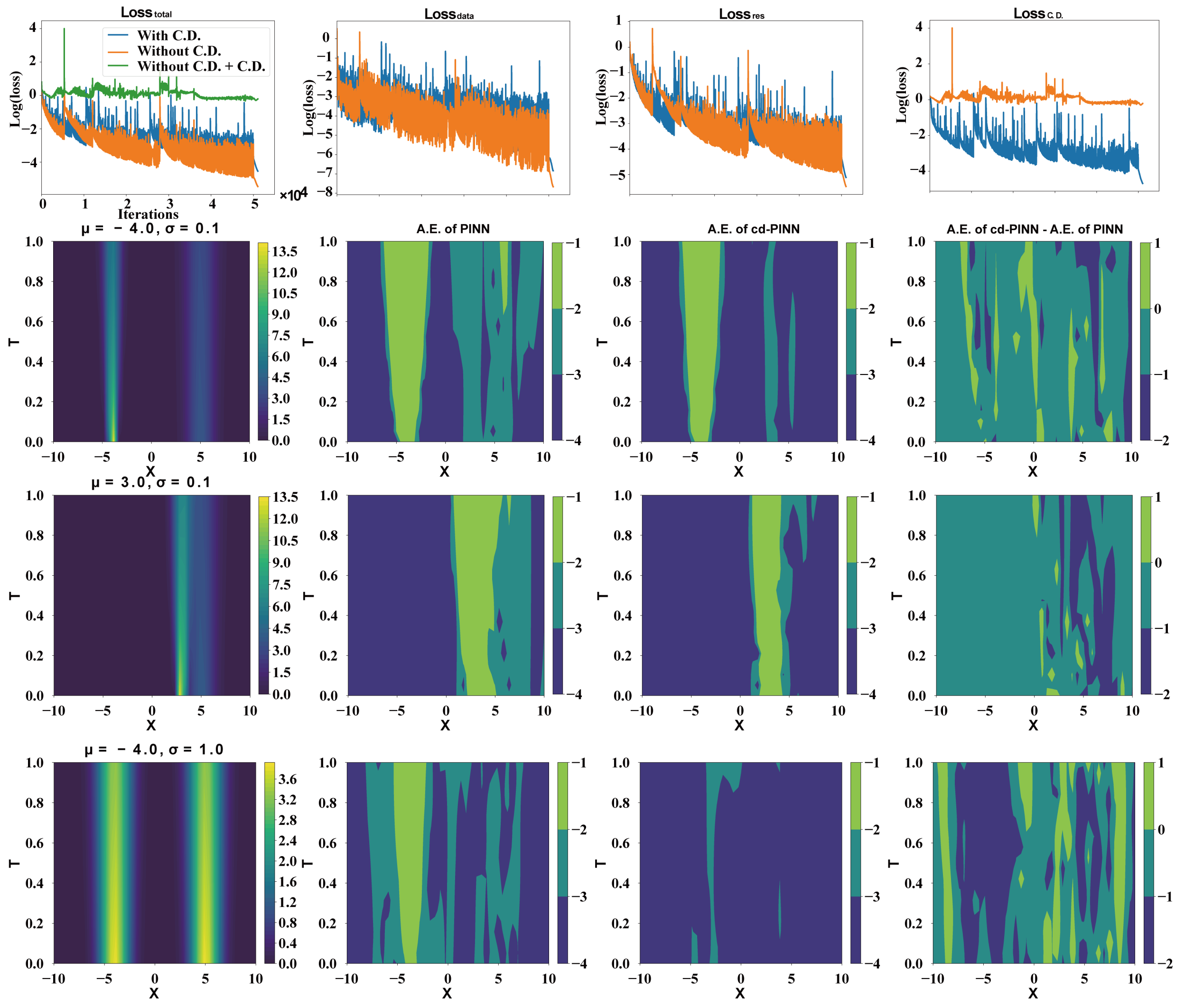

Figure 1: Result of two-dimensional diffusion equation. (a) shows the loss of each part during the model training process. Each column of (b), (c), and (d) shows the true solution, the absolute error of PINN predicted solution, the absolute error of the cd-PINN predicted solution, and the difference in the logarithmic value of the absolute error between the cd-PINN and PINN predicted solution. (a), (b) and (c) are the situations under three different combinations of μ and σ respectively.

Figure 1 shows the predicted results of cd-PINN under three different combinations of μ and σ. It can be seen that cd-PINN has good prediction results under the new μ and σ. Figure 2 shows the mean absolute errors of PINN and cd-PINN under all μ and σ. It can be seen that cd-PINN has obvious advantages over PINN. Moreover, the NRMSE of PINN is 2.07×10^{-2}, while the NRMSE of cd-PINN is 5.24×10^{-3}, an order lower than the former.

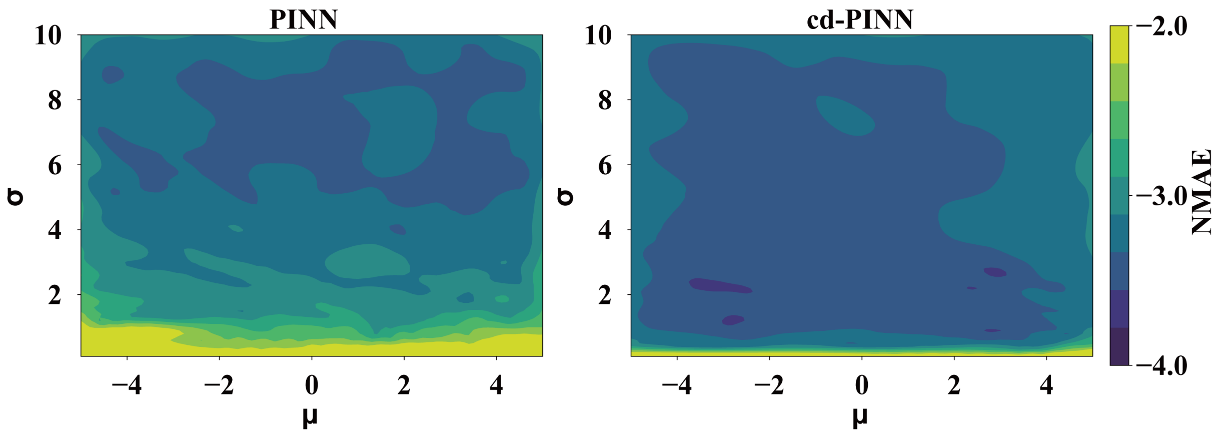

Figure 2: The mean absolute error of PINN and cd-PINN under 10100 configurations of values for $\sigma \in [0.1, 10.0]$ and $\mu \in [-5.0, 5.0]$.

We fully understood the significance of applying cd-PINN to PDE cases. However due to the scope of this paper, which focuses on the applications to ODEs, as well as page limits, we decided not to include the PDEs in the current study.

(2) Neural ODE is an alternative widely used framework for studying time series data modeled by ODEs. According to the reviewer's suggestion, we made a detailed comparison between our cd-PINN and neural ODE models on the LV model. And we found that although the Neural ODE model could effectively capture the system dynamics at the training data points, it shows a much poorer generalization ability than cd-PINN under the testing scenarios. In addition, the Neural ODE model takes a much longer training time due to the explicit implementation of Runge-Kutta integration steps, whereas PINN and cd-PINN are more computationally efficient.

Changes: See Appendix B.6. for changes.

"Neural ODEs [1] are another prominent approach for learning solutions to ordinary differential equations. As a comparison, here we implement the Neural ODE model for the Lotka-Volterra system under Scenery 1 with respect to the same data set and evaluation metrics. The Neural ODE model is numerically integrated by using the 4-order Runge-Kutta method. The model architecture takes the current state $(X(t), Y(t))$ and initial conditions (X_0, Y_0) as inputs, and predicts the state derivatives $(dX/dt, dY/dt)$ as outputs.

From Fig.3, we can see that the Neural ODE model could effectively capture the system dynamics at the training data points, but shows a much poorer generalization ability than cd-PINN under the testing scenarios ($MSE = 1.37 \times 10^{-2}$ v.s. 4.48×10^{-5} for cd-PINN). In addition, the Neural ODE model takes a much longer training time ($25,535s$ for 50,000 epochs) due to the explicit implementation of RK4 integration steps, whereas PINN and cd-PINN are more computationally efficient."

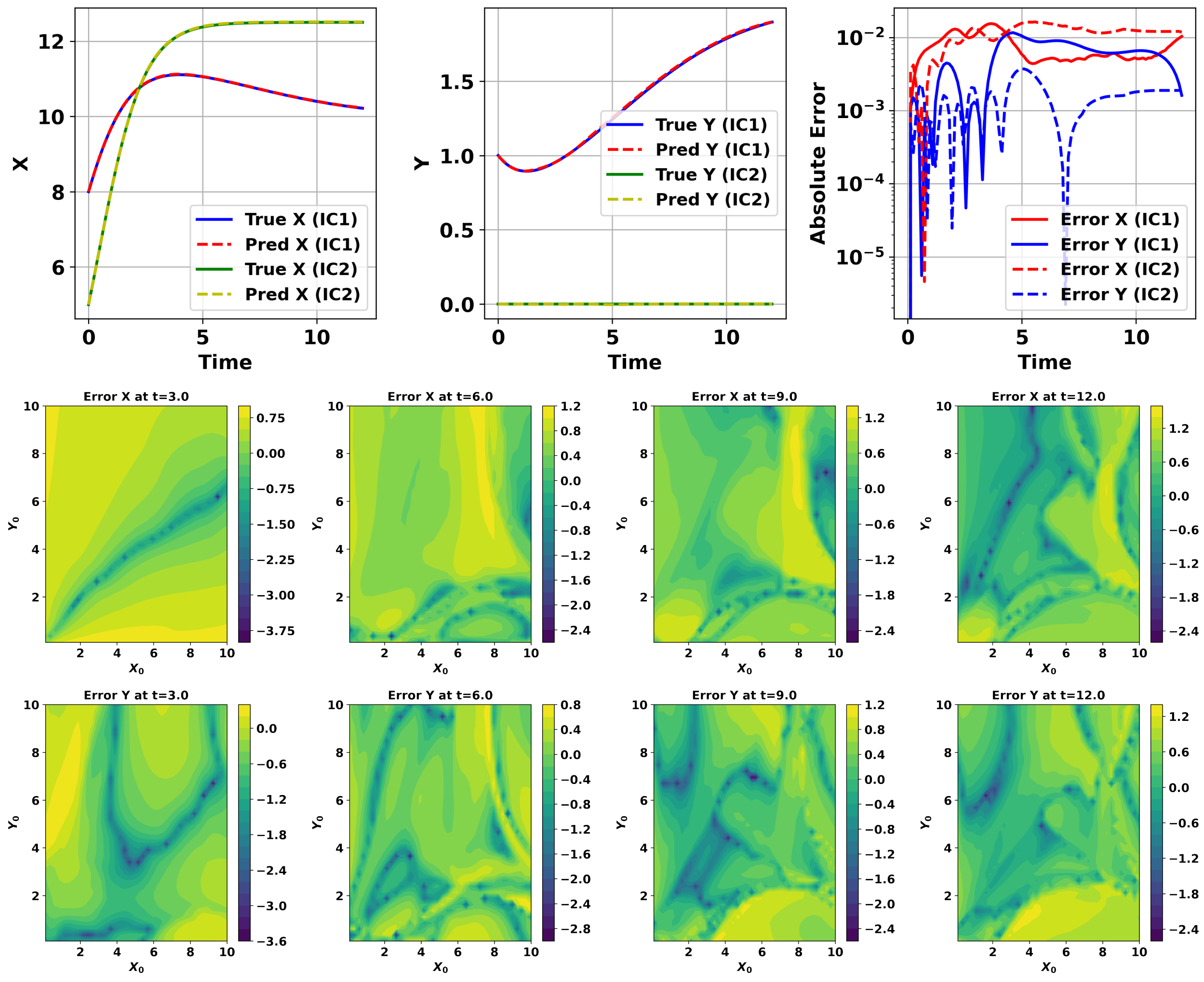

Figure 3: Results of Neural ODEs for the LV model on the training (upper) and test (bottom) data sets separately.

Comment — Extra-interpolation Capabilities

While cd-PINN claims to learn continuous flow of parameters of ODE, it is unclear whether the model performs well on unseen initial values and parameters. If cd-PINN can indeed generalize to genuinely novel scenarios (e.g., significantly out-of-distribution conditions), this would represent a major strength. Several ablation studies about these cases would be highlight cd-PINN's performance and applicability. Furthermore, a discussion on whether cd-PINN can reliably extrapolate beyond trained distributions would add clarity regarding its practical usability in diverse differential equation settings.

Reply: To investigate how cd-PINN performs on unseen initial values and parameters, we double and quadruple the test region of the LV system – the second example in the main text, while still keeping training data region unchanged (within $0 - 10$). From Figure 4, it can be seen that by doubling the test region, the absolute errors of the predicted solutions by cd-PINN is still lower than 10^{-1}, which can meet the requirements of most real-world applications.

Figure 5 further shows the predicted results by quadrupling the test region. In this case, cd-PINN still show better generalization ability than PINN. Actually, if we want to make sure that cd-PINN has sufficiently accurate prediction capabilities in a wider range of initial values or parameters, we can expand the sampling range of residual points and increase the number of residual points, which is very easy to implement.

To sum up, we can conclude that our cd-PINN performs well on unseen initial values and parameters, and it can be indeed generalized to significantly out-of-distribution conditions. Moreover, Vanilla PINN provides a natural ablation study of cd-PINN. The corresponding results are highlighted in Figures 1 and 3 in the main text.

Changes: See Page 8 for changes.

"Furthermore, our numerical simulations reveal that even for fixed initial values or parameters, the accuracy and convergence rate of cd-PINN are usually much better than PINN (see Appendix B.5.). Meanwhile, for unseen initial values and parameters, like data points beyond the training set, the cd-PINN also shows a satisfactory performance (data not shown), demonstrating a major strength of cd-PINN that it can indeed generalize to genuinely novel scenarios. We contribute these improvements to the inclusion of additional mathematical constraints on continuous dependence."

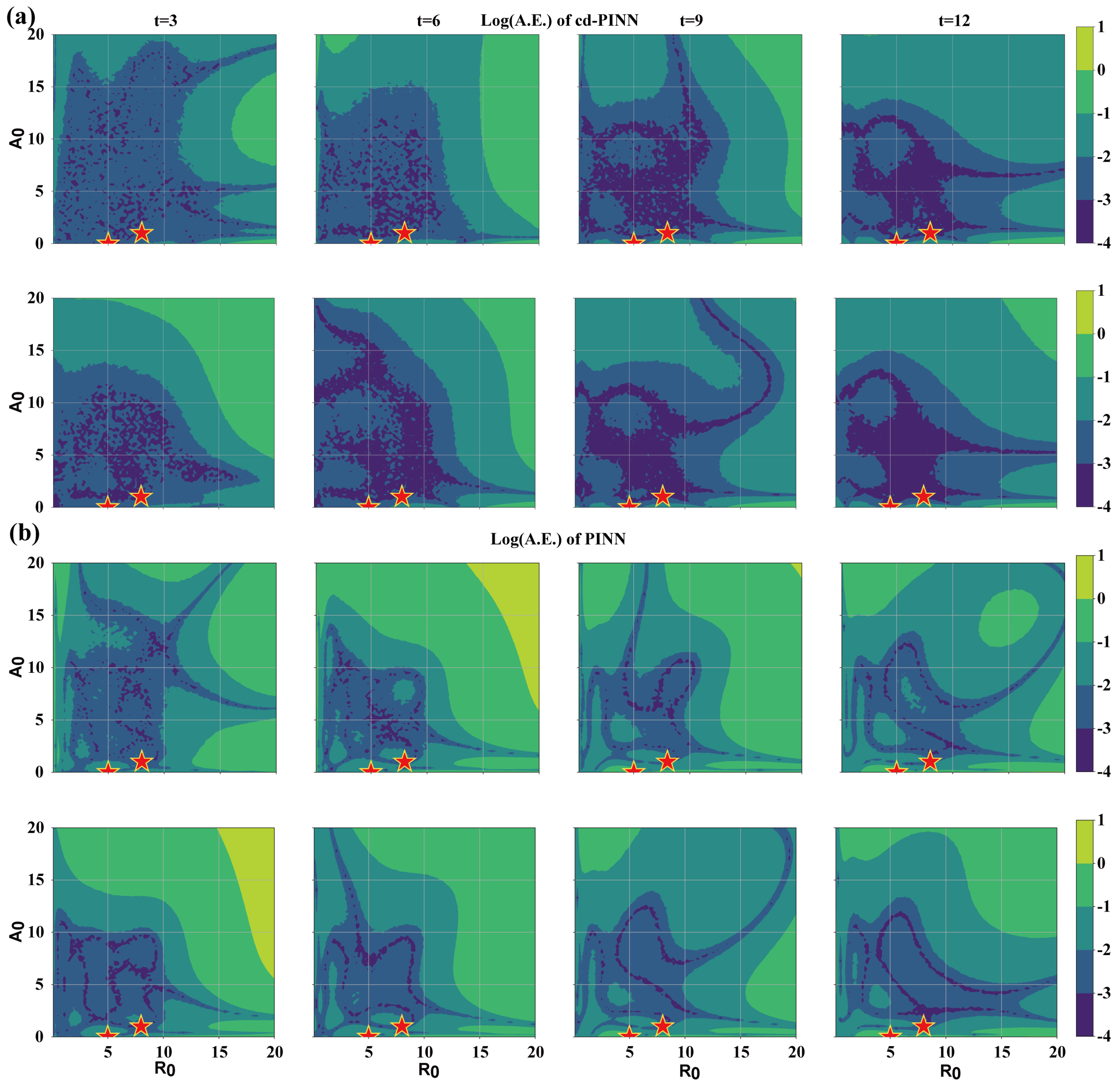

Figure 4: Comparison of the absolute errors of cd-PINN and PINN predicted results under twice the training interval. The first row of (a) is the logarithm of the absolute error in predicting R by cd-PINN, while the second row is the same result for A. (b) is the same result for PINN.

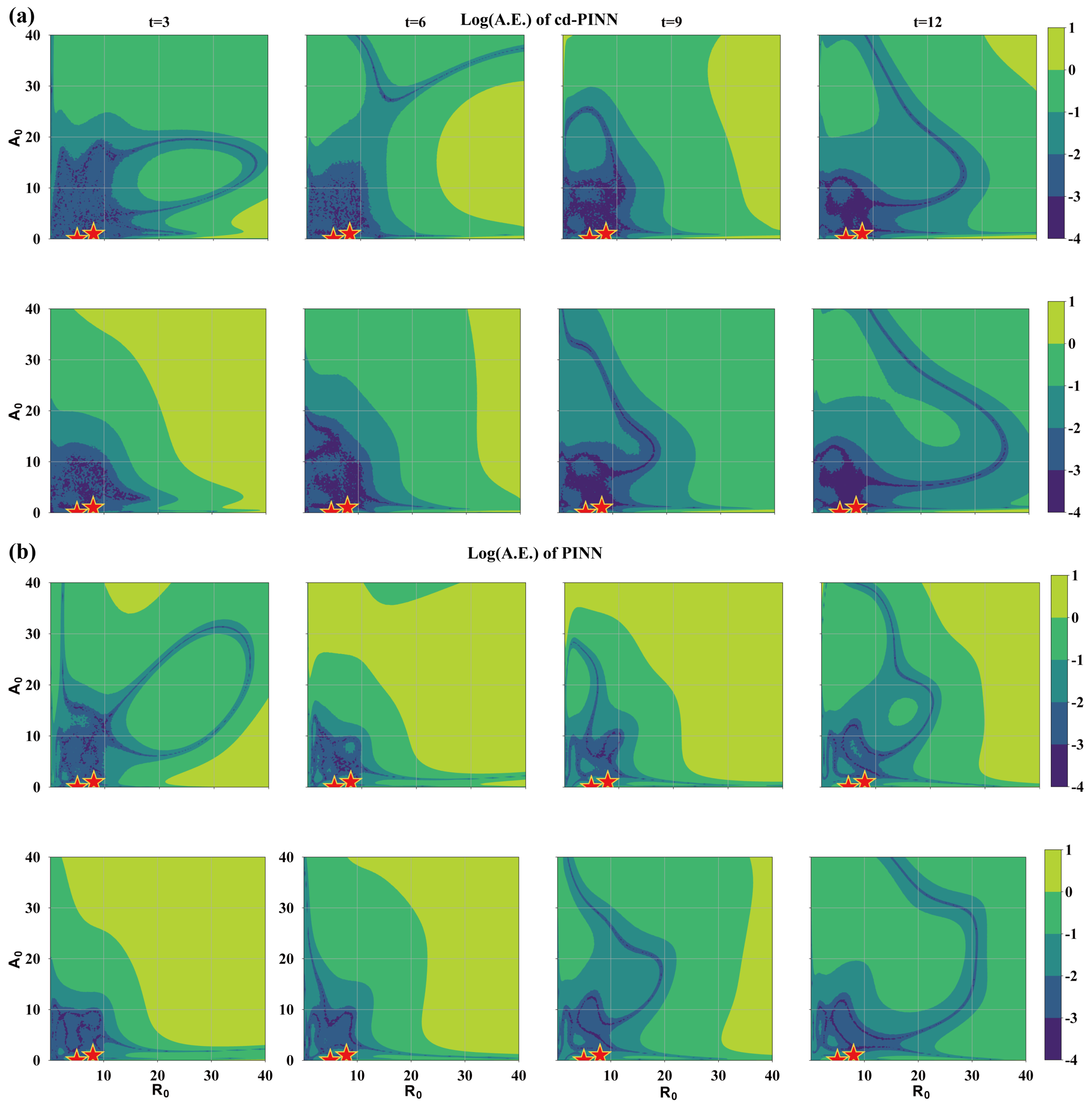

Figure 5: Comparision of the absolute errors of cd-PINN and PINN predicted results under quadruple the training interval. The first row of (a) is the logarithm of the absolute error in predicting R by cd-PINN, while the second row is the same result for A. (b) is the same result for PINN.

References

[1] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. Advances in Neural Information Processing Systems, 31, 2018.