# Response to Reviewer tHxD

**Comment** — This formulation seems to only work for inputs in ($\mathbb{R}$) or low-dimensional cases, which limits its applicability to PDE problems, where the input is high dimensional.

**Reply**: We can not agree with the referee on this point. The basic idea of continuous dependence is not limited to ODEs. For PDEs, the continuous dependence on the initial value and function $f$ on the right-hand side of PDEs is known as the well-posedness. Here we present an example of cd-PINN applied to the diffusion equations. It is 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.

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

Therefore, the formulation of cd-PINN is not "only work for low-dimensional cases". Here due to scope of our current study (focusing on ODEs) and page limits, we restricted ourselves to the ODE cases. The studies on PDEs will be given in another paper.

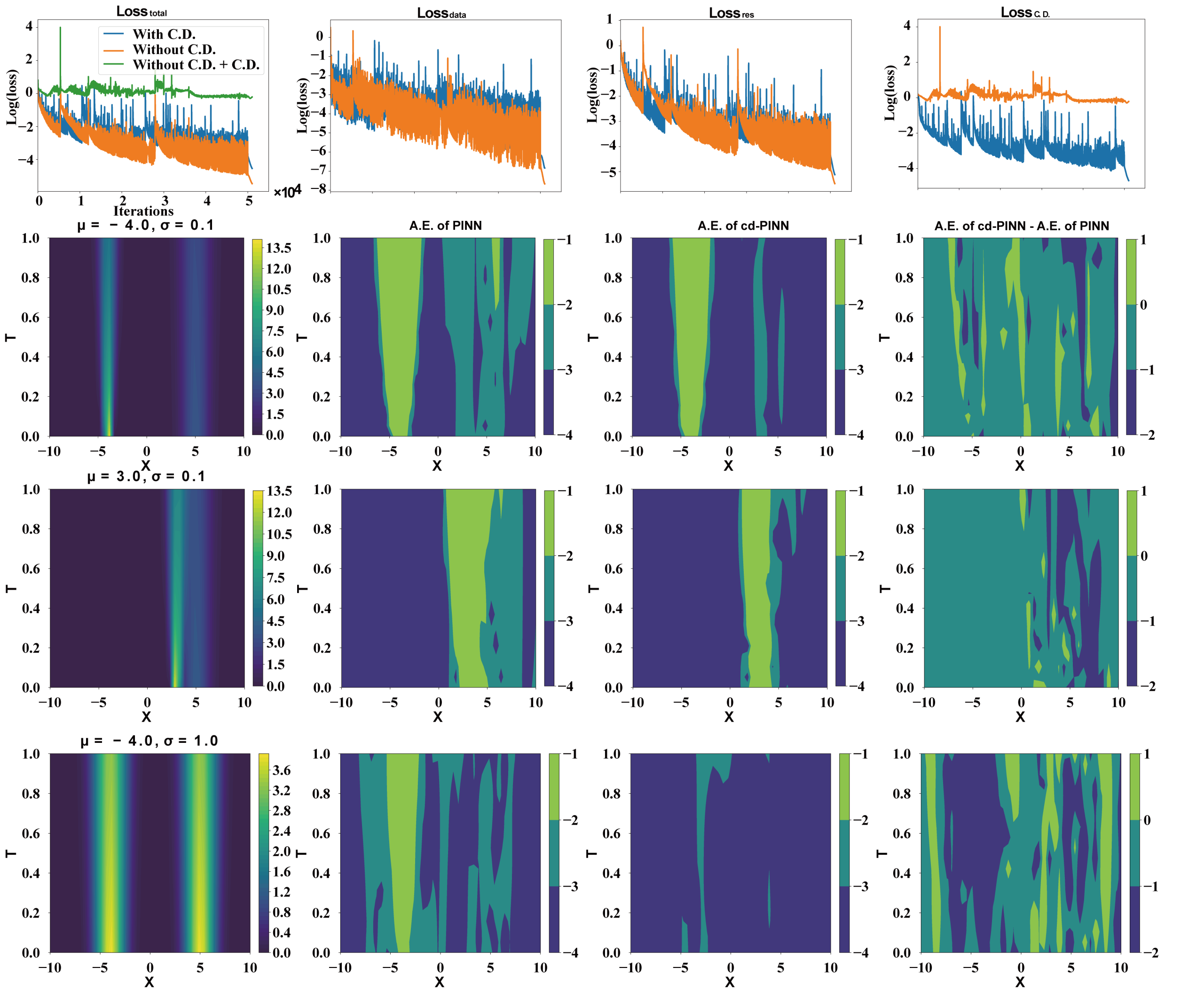


Figure 1: Result of two-dimensional diffusion equation. $(a)$ shows the loss of each part during the model training process. Each column of $(b)$, $(c)$, and $(d)$ shows the true solution, the absolute error of PINN predicted solution, the absolute error of the cd-PINN predicted solution, and the difference 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 $\mu$ and $\sigma$ respectively.

**Comment** — The assumption of continuity may not hold in chaotic systems, such as the Lorenz system.

**Reply**: Mathematically, the continuous dependence of ODE solutions on its initial value and the sensitive dependence of ODE solutions on its initial value are two closely related but different concepts. For chaotic systems, such as the Lorenz system, its long-term solutions exhibit a sensitive dependence on its initial values, which means two solutions initially contacted will diverge exponentially in time. However, it does not necessarily violate the continuous dependence.

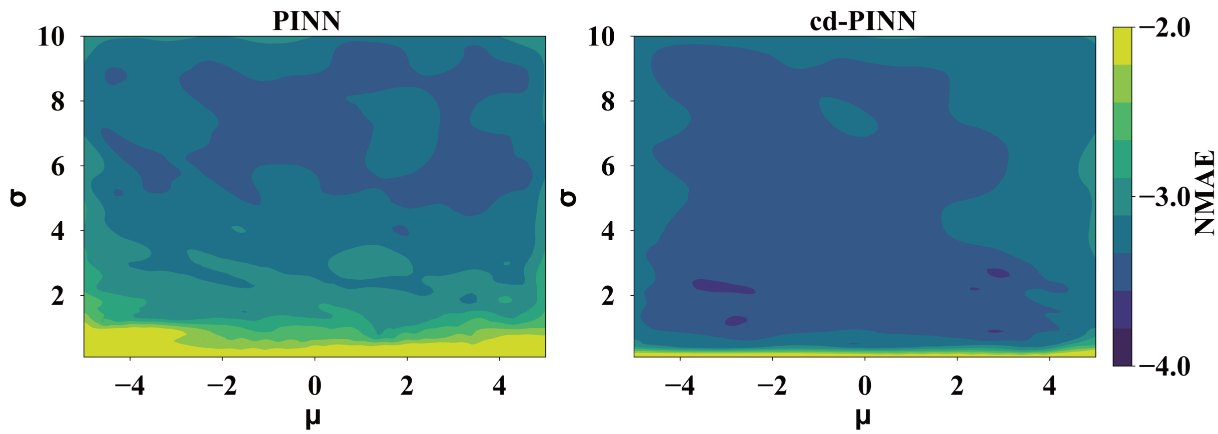


Figure 2: The mean absolute error of PINN and cd-PINN under 10100 configurations of values for $\sigma \in [0.1, 10.0]$ and $\mu \in [-5.0, 5.0]$.

Here we look into the Lorenz system, which is given by the following set of equations,

$$\begin{aligned} \frac{dx}{dt} &= \sigma(y - x), \\ \frac{dy}{dt} &= x(\rho - z) - y, \\ \frac{dz}{dt} &= xy - \beta z, \end{aligned} \tag{2}$$

where $\sigma, \rho, \beta$ are the parameters of the system. In the simulating the Lorenz system, we adopt the parameter configuration ($\sigma = 10, \rho = 28, \beta = \frac{8}{3}$) and initialize the system with conditions $x(0) = 0, y(0) = 2$ and $z(0) = 9$.

In the above system, we consider $x_0$ as an additional input into the model and add continuous dependency constraints on it. We trained three models, namely an MLP model for data regression, PINN with a fixed initial value, and the cd-PINN. For the data-regression model, the training data consists of $10000$ label data points, with the aim of testing whether the MLP has the ability to fit the solutions to the Lorenz system. For PINN and cd-PINN, the training data consists of $1000$ label data points and $2^{14}$ residual data points.

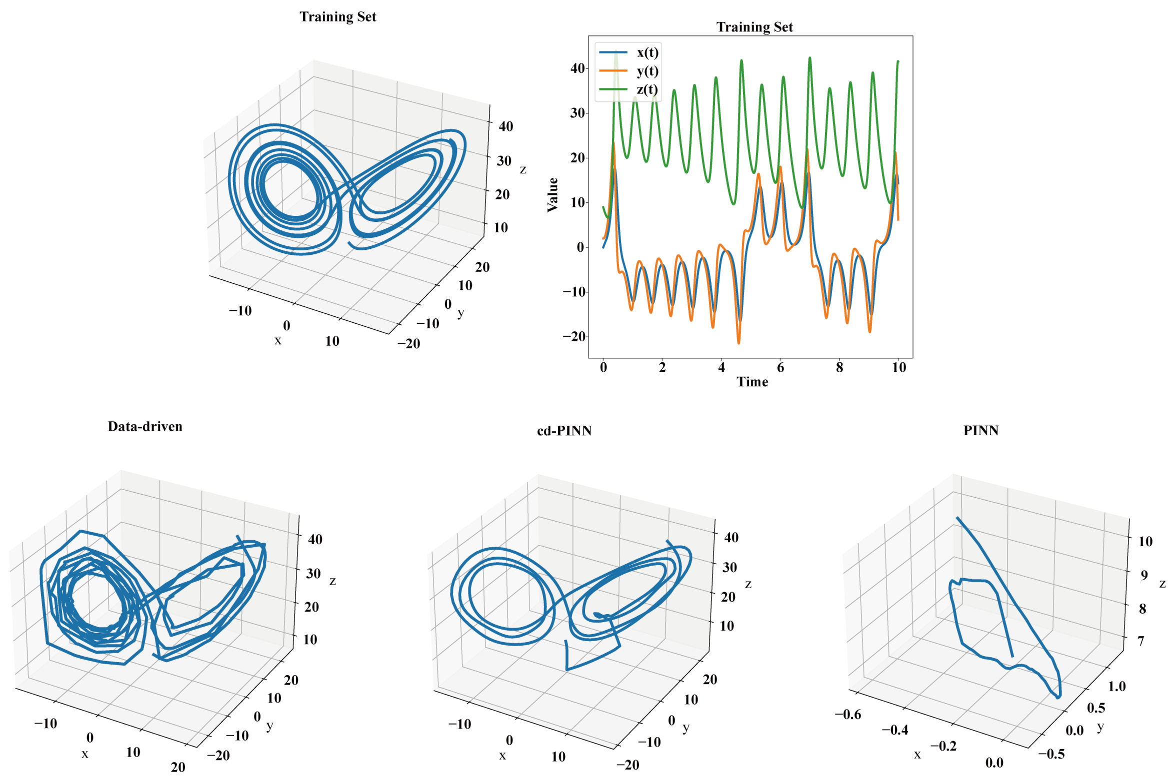


Figure 3: Result of Lorenz system

Figure 3 shows the training data and the results of the three models respectively. It can be seen that even the data-regression model is unable to fit the solution in a satisfatory way. Under the premise that DNN cannot completely fit the solution of the Lorenz system based on the data, we further tested the cd-PINN and PINN. It can be seen that adding continuity constraints about the initial value is helpful, but is still not good enough. Such a result is not surprising. It was proposed in paper [2, 3] that DNN can considered as a low-pass filter and is difficult to encode high-frequency components. Therefore, we believe that the failure of our model in the Lorenz system is mainly due to the fact that DNN cannot fit high-frequency functions well. A preliminary attempt by replacing the MLP with LSTM fails for the Lorenz case too, alternative more complicated and powerful network structures, like transformer, is still under test.

**Comment — Writing:**

1. The input parameter ($\mu$) could benefit from further discussion. It would be helpful to include an example alongside Equation (1).

2. It would improve clarity to separate the previous formulation of PINN from the newly proposed cd-PINN. Equations (5) and (6) should be moved to a dedicated subsection for PINN.

3. For PDEs, it would be beneficial to cite PINO, which combines neural operators with PINNs using pre-training and fine-tuning: Li, Zongyi, et al. "Physics-informed neural operator for learning partial differential equations." ACM/JMS Journal of Data Science.

**Reply**: Thank you for the suggestions. The manuscript has been revised accordingly.

**Changes**: 1. The following sentence has been added below Eq. 1.
"For example, given the growth model $du/dt = ru^k$, where $r, k \in \mathbb{R}$, we have $\boldsymbol{\mu} = (r, k)$."
2. The formulation of PINN including Eqs. 5 and 6 is removed to the appendix for clarity.
3. The paper on PINO has been cited in the revision. See below.
"PINO [1] is the first hybrid approach incorporating data and PDE constraints at different resolutions to learn the solution operator of a given family of parametric PDEs."

**Comment** — Could the author give better motivation of application of cd-PINN beyond PINN?

**Reply**: The basic motivation of cd-PINN are very clear:
(1) Improving the generality of PINN framework for solving various differential equations. So that without dramatically increasing the computlational cost, we can obtain an efficient numerical solver which are suitable for a group of equations rather than single one.
(2) Speeding up the solvation of inverse problems involving differential equations. When applying PINN to the inverse problems, the neural network has to be trained repeatly in order to find the correct solutions corresponding to the differential equations with specified parameters. Therefore, our cd-PINN is particularly suitable for this task, since it can solve equations with diverse parameters without further fine tuning.

**Comment** — Can cd-PINN improve the accuracy or convergence rate compared to PINN give a fixed parameters?

**Reply**: To compare the accuracy and convergence rate of cd-PINN with those of PINN for fixed initial values or parameters, we conduct experiments on the LV system. Here we train the PINN model with fixed initial values $R_0 = 8.0$ and $A_0 = 1.0$. The training data set includes $20$ real data points and $2^{14}$ residual data points. The same setup is adopted for the cd-PINN. During the training procedure, we first use the Adam optimizer to train for 10000 epochs and then call the LBFGS optimizer for further optimization.

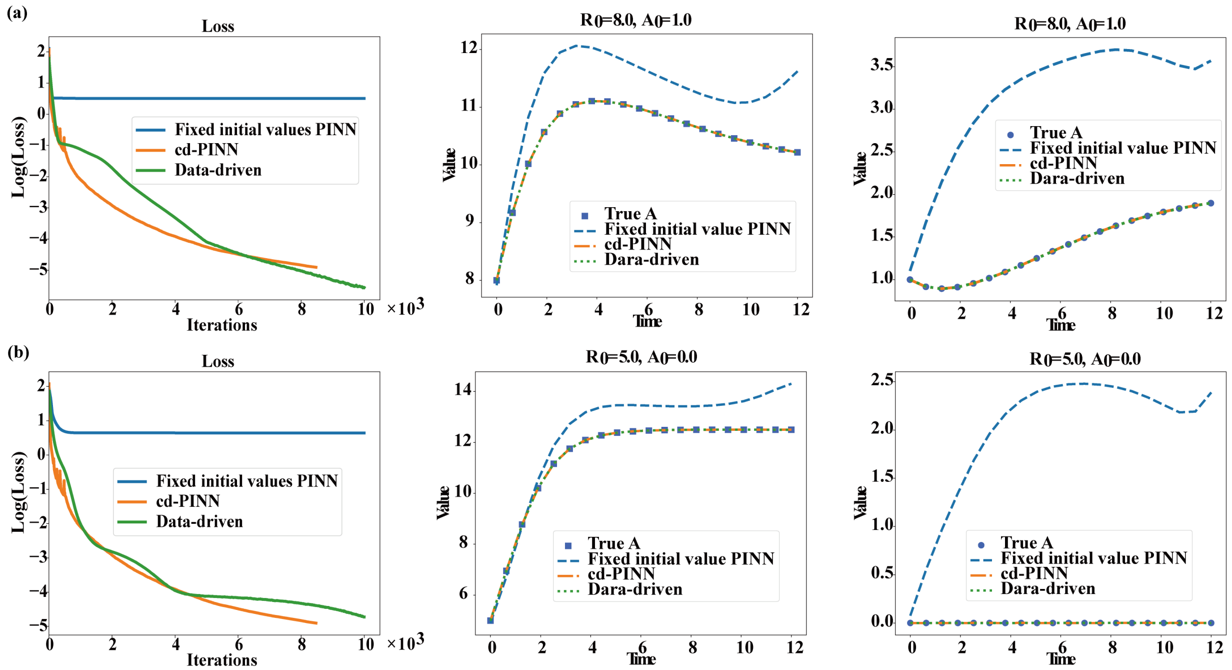


Figure 4: Comparison on the convergence rate and accuracy of cd-PINN, PINN with fixed initial value and a data-regression MLP model. $(a)$ and $(b)$ illustrate the results for $R_0 = 8.0, A_0 = 1.0$ and $R_0 = 5.0, A_0 = 0.0$ respectively.

The results presented in Figure 4 clearly demonstrate that the cd-PINN has a better convergence rate and accuracy than those of PINN. Most astonishingly, in the current case, the convergence rate of cd-PINN is even comparable to a pure data regression model by MLP. And we believe this is a general feature of cd-PINN, which from the other side highlights the significance of inclusion of continuous dependence.

**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."

**Comment** — Is it possible to apply the cd-PINN and viscosity in Burgers equation or Reynolds number in Navier-Stokes?

**Reply**: A very nice suggestion. Our preliminary results show that incorporating the condintious dependence could indeed improve generality of cd-PINN on the visocity solutions of Burgers equation. But when extending to non-viscous Burgers equation, additional approaches are required due to the discontinuity of the solutions.

**Changes**: See Page 11 for changes.

"In the current paper, we restrict our study to ordinary differential equations for clarity. Obviously, the same approach is applicable to partial differential equations too, e.g. the viscosity in Burgers equation or the Reynolds number in Navier-Stokes equations. However, it should be noted that the PDE cases are far more complicated in general. For example, in many cases the parameter dependence of PDEs may be continuous but not necessarily differentiable. This subtle distinction is crucial for certain contexts, such as the shock structures in hyperbolic conservation laws. Under these situations, we need to turn to more general conditions, like the Rankine-Hugoniot jump condition, to determine the exact locations where the shock structure arises. The related work is ongoing."

## References

[1] Zongyi Li, Hongkai Zheng, Nikola Kovachki, David Jin, Haoxuan Chen, Burigede Liu, Kamyar Azizzadenesheli, and Anima Anandkumar. Physics-informed neural operator for learning partial differential equations. ACM/JMS Journal of Data Science, 1(3):1–27, 2024.

[2] Zhi-Qin John Xu, Yaoyu Zhang, Tao Luo, Yanyang Xiao, and Zheng Ma. Frequency principle: Fourier analysis sheds light on deep neural networks. arXiv preprint arXiv:1901.06523, 2019.

[3] Zhi-Qin John Xu, Yaoyu Zhang, and Yanyang Xiao. Training behavior of deep neural network in frequency domain. In Neural Information Processing: 26th International Conference, ICONIP 2019, Sydney, NSW, Australia, December 12–15, 2019, Proceedings, Part I 26, pages 264–274. Springer, 2019.