How PINNs cheat: Predicting chaotic motion of a double pendulum

Anonymous Author(s)
Affiliation
Address
email

Abstract

Despite extensive research, physics-informed neural networks (PINNs) are still difficult to train, especially when the optimization relies heavily on the physics loss term. Convergence problems frequently occur when simulating dynamical systems with high-frequency components, chaotic or turbulent behavior. In this work, we discuss whether the traditional PINN framework is able to predict chaotic motion by conducting experiments on the undamped double pendulum. Our results demonstrate that PINNs do not exhibit any sensitivity to perturbations in the initial condition. Instead, the PINN optimization consistently converges to physically correct solutions that violate the initial condition only marginally, but diverge significantly from the desired solution due to the chaotic nature of the system. In fact, the PINN predictions primarily exhibit low-frequency components with a smaller magnitude of higher-order derivatives, which favors lower physics loss values compared to the desired solution. We thus hypothesize that the PINNs "cheat" by shifting the initial conditions to values that correspond to physically correct solutions that are easier to learn. Initial experiments suggest that domain decomposition combined with an appropriate loss weighting scheme mitigates this effect and allows convergence to the desired solution.

1 Introduction

Physics-informed neural networks (PINNs) [1, 2] are an emerging class of physics-enhanced machine learning techniques that provide a mesh-free and time-continuous approach to solving forward and inverse problems governed by differential equations. PINNs seamlessly integrate knowledge about a physical system into a neural network model by incorporating the residuals of the governing differential equations via an additional physics-based loss function. The physics loss acts as a regularization term, penalizing solutions that violate the physical laws of the underlying system. This approach has been proven successful in numerous applications [3, 4, 5, 6, 7, 8, 9, 10, 9, 11], including the simulation of dynamical systems.

However, the prediction accuracy of PINNs relies on the optimization of the multi-objective cost function consisting of the data and physics-based loss term which encode the initial conditions (ICs) and governing differential equations, respectively. Several studies have identified these competing loss terms as one cause of severe convergence problems [12, 13, 14], proposing various manual or adaptive loss weighting schemes. Still, PINNs frequently struggle to solve problems for large computational domains [15, 16, 14, 17]. Additionally, the spectral bias [18, 19, 20, 21] was shown to cause neural networks to learn low-frequency functions faster and to impede the convergence of PINNs for large computational domains [22]. As the input variable of the network is typically normalized to a fixed range in a preprocessing step, low frequencies on large computational domains become high frequency components. At the same time, a large computational domain results in a more

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

complicated landscape for PINN optimization, where nonphysical solutions may even correspond to better optima of the multi-objective problem than the desired physical solutions [23].

Overall, the properties and limitations of PINNs are still poorly understood. Thus, the aim of this work is to gain a deeper understanding of PINNs through experiments on a nonlinear dynamical system, the undamped double pendulum. Our main contributions comprise the following:

- We empirically demonstrate the difficulty of predicting chaotic motions using the traditional PINN framework on a double pendulum. We observe that the predicted trajectories violate the ICs marginally, but deviate significantly from the reference.
- We show that the PINN exhibits no sensitivity to perturbations of the given IC and prefers convergence to the same undesired solution characterized by a significantly lower physics loss compared to the reference trajectory.
- We claim that the convergence to such incorrect attractive solutions is related to the spectral power distribution and magnitude of higher-order derivatives. In other words, the PINN "cheats" by choosing an IC that corresponds to a simpler solution.
- We note that a reduced computational domain decreases the sensitivity to the IC with respect to the aforementioned measures. In combination with a suitable loss weighting scheme, the desired solution becomes the dominant one.

2 Methods

Double pendulum The planar double pendulum is a nonlinear dynamical system that exhibits harmonic and chaotic behavior, based on its initial displacement and velocity. It consists of two point mass pendulums with masses m_1, m_2 and rod length L_1, L_2 attached to each other. The angle between the two pendulum rods and the vertical axis (i.e., θ_1, θ_2) is governed by two nonlinear second-order coupled ordinary differential equations (ODEs) of the form

$$\boldsymbol{y}'' = [\, f_1(\boldsymbol{y}, \boldsymbol{y}'),\ f_2(\boldsymbol{y}, \boldsymbol{y}') \,]^T \quad \text{with} \quad \boldsymbol{y} = [\, \theta_1,\ \theta_2 \,]^T . \tag{1}$$

For a detailed description, see Appendix A.1.

Physics-informed neural networks The unknown solution $\boldsymbol{y}$ is approximated by a fully-connected deep neural network $\boldsymbol{y_w}$ with $\boldsymbol{w}$ describing all trainable weights and biases of the network. The heart of PINNs [1] consists of combining data- and physics-based constraints into a single loss function $L(\boldsymbol{w}) = \alpha L_{IC}(\boldsymbol{w}) + (1-\alpha) L_F(\boldsymbol{w})$ via multi-objective optimization. The data loss term L_{IC} reflects the deviation from the given initial condition, while the physical term L_F incorporates the ODE residuals of the underlying physical system. Thus, the performance of PINNs depends strongly on the ODE system itself. For a detailed description of each loss term, see Appendix A.2.

Analysis of solution trajectories We analyze two measures that describe properties of the predicted solution trajectories compared to the desired one. The aim is to improve our understanding of the difficulties in training PINNs for the simulation of chaotic motion.

First, we compare the distribution of the solutions' frequency components, namely, the amount of low-frequency components. We compute this measure using the relative fraction of total signal power above a certain frequency f_0, given by

$$P_{f_0} = \frac{\sum_{k > k_{f_0}} |X_k|^2}{\sum_k |X_k|^2} \tag{2}$$

where X_k denotes the discrete Fourier transform (DFT) of a signal $x(t_n)$ (in our experiments, $\theta_1,\ \theta_2$) and k_{f_0} denotes the DFT index corresponding to the boundary frequency f_0.

As a second measure, we consider the average magnitude of the second order derivatives, which are approximated using the central finite difference method with a step size of $h = 0.005$.

$$\overline{|\theta_1''|} + \overline{|\theta_2''|} = \sum_{i=1}^{2} \frac{1}{N} \sum_{n=1}^{N} \left| \frac{\delta_h^2[\theta_i](n_h)}{h^2} \right| \tag{3}$$

The derivatives θ_1'', θ_2'' are used in the computation of the physics loss, summarized in Appendix A.2. As we consider the absolute error for the physics loss, smaller values of the derivatives also contribute to smaller values of the resulting physics loss.

3 Experiments

For all experiments, the system parameters of the double pendulum (m_1, m_2, L_1, L_2) are fixed to one. The initial angles of the pendulums are set to $\theta_1(t_0) = \theta_2(t_0) = \theta_0$ with zero initial velocities $\theta_1'(t_0) = \theta_2'(t_0) = 0$. We use a fourth-order Runge-Kutta (RK45) scheme with step size $h = 0.005$ to obtain a reference solution.

All experiments are conducted with the following network specifications and hyperparameters: network type (6x30 fully-connected neural network), network activation function (swish), output activation function (linear), weight initialization (Glorot uniform [24]), optimizer (Adam [25], default settings of Tensorflow [26]), learning rate (0.01), number of epochs (25000). Additionally, the input of the network (time t) is normalized to a fixed domain of $[-5, 5]$.

3.1 Results

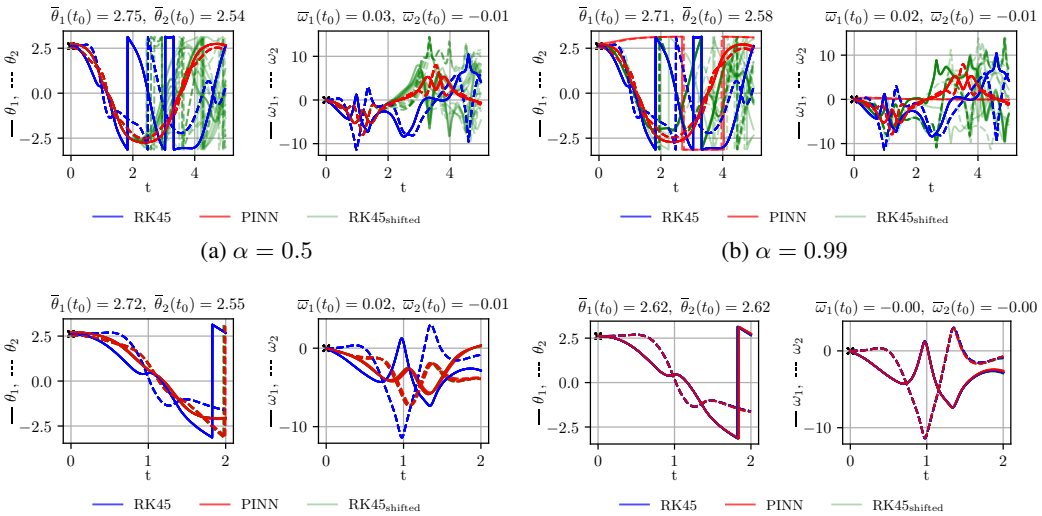
First, the initial angles of the double pendulum are selected as $\theta_0 = 150° \approx 2.62\,\text{rad}$, resulting in chaotic motion. We train ten PINN instances with the same IC and network specifications, but different seeds for the weight initialization. As the traditional PINN framework incorporates the optimization of the IC as a soft constraint, deviations from the true value are allowed.

The predictions obtained by minimizing the unweighted loss function with $\alpha = 0.5$ differ heavily from the reference solution, as illustrated in Figure 1 (a). Despite this deviation, the predicted trajectories can be shown to correspond to physically correct solutions. Indeed, repeating the RK simulation with the shifted ICs that where chosen by the PINN instances from the previous experiment, we observe that in the initial phase ($t < 2$), the RK solution follows the PINN predictions exactly (green lines in Figure 1). Due to the chaotic nature of the pendulum, the RK trajectories begin to diverge for $t > 2$ due to minor differences in the imposed ICs. Continuing the RK simulation at a later time step, just before the trajectories begin to diverge, again yields an exact match between PINN and RK (not shown here). Interestingly, looking at the physics loss averaged over the ten PINN instances we obtain a value of $\overline{L}_{F,\alpha=0.5} = 1.79 \cdot 10^{-3}$, with a standard deviation of $6.44 \cdot 10^{-4}$, that is significantly lower compared to the reference solution with $L_{F,RK} = 5.74 \cdot 10^{-1}$. From these results, we find that the PINNs achieve a solution with an overall lower physics loss by, first, shifting the IC accordingly and, second, violating the physical constraints slightly in the initial phase.

A stronger emphasis on the IC ($\alpha = 0.99$) does not improve the prediction, as can be seen in Figure 1 (b). Although the IC is fulfilled more precisely, the shape of the trajectories still coincides well with the results of the unweighted PINN ($\alpha = 0.5$). This causes the RK solution for the shifted ICs to deviate more from the PINN predictions. Instead of converging to the solution that matches the shifted ICs, the PINNs choose to neglect the physical constraints in the initial phase and thus fail to exhibit the sensitivity to ICs characteristic for chaotic systems. Compared to the unweighted case, the results for $\alpha = 0.99$ agree more accurately with the IC, but consequently need to violate the physics more severely in order to reach this solution trajectory. Despite the violation of the physical constraints, the average physics loss over the ten PINN instances is still in a similar range to the reference with $\overline{L}_{F,\alpha=0.99} = 2.70 \cdot 10^{-1}$ and a standard deviation of $4.7 \cdot 10^{-1}$.

In a next step, we confirm that the convergence in the chaotic regime is strongly affected by the length of the computational domain and an appropriate choice of α. Only after decreasing the computational domain and increasing α, the PINN converges to the correct solution, shown in Figure 1 (c) and (d).

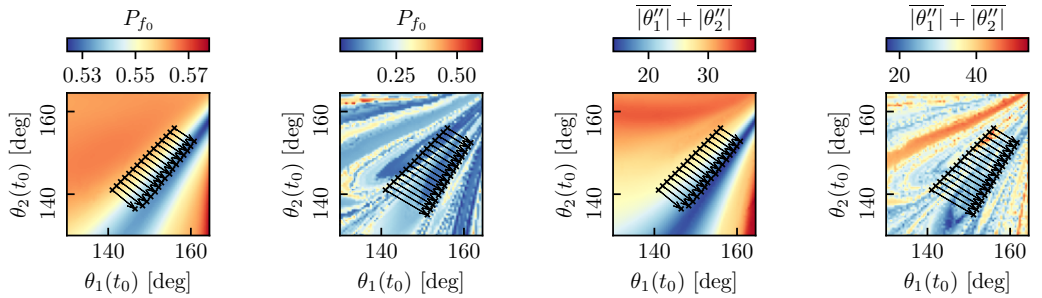
Lastly, the experiment is extended to initial angles of $\theta_0 = \{141°, \ldots, 156°\}$, focusing on the unweighted case ($\alpha = 0.5$) and one PINN instance with a fixed seed for the initial weights. In the previous experiment we observed that all PINNs equally shifted the IC to reach a solution with a significantly lower physics loss. Now, we aim to assess why the shifted solution is more attractive compared to the reference solution. To do so, we compare the spectral power ratio P_{f_0} above the frequencies $f_0 = 0.5\,\text{Hz}$, and the average magnitude of the derivatives $\overline{|\theta_1''|} + \overline{|\theta_2''|}$, shown in Figure 2. Both measures are computed using RK. Indicated arrows point from the true IC θ_0 to the shifted IC of the PINN ($\alpha = 0.5$) prediction. From Figure 2 (a) we can observe that the PINN shifts the IC to solutions with primarily low-frequency components. Figure 2 (b) indicates that the shifted solutions also exhibit smaller magnitudes of the second-order derivatives, which lead to smaller physics loss values. In particular, this effect is prominent for a computational domain of $T = 2$, where the PINN predictions lie in a valley of small values for both measures. This valley is also



(a) $\alpha = 0.5$ (b) $\alpha = 0.99$

(c) $\alpha = 0.5$ (d) $\alpha = 0.99$

Figure 1: Results for the prediction of the solution trajectory given the IC of $\theta_0 = 150° \approx 2.62\,\text{rad}$ and different lengths of the computational domain: $T = 5$ *(top)*, $T = 2$ *(bottom)*.



(a) $T = 2$ (b) $T = 5$ (c) $T = 2$ (d) $T = 5$

Figure 2: Results for the spectral power ratio P_{f_0} above $f_0 = 0.5\,\text{Hz}$ *(left)* and magnitude of the second order derivatives $\overline{|\theta_1''|} + \overline{|\theta_2''|}$ *(right)*, computed with RK. Arrows point from the true IC to the predicted IC of the unweighted PINN ($\alpha = 0.5$).

visible for the computational domain of $T = 5$, although being more narrow. Moreover, for $T = 5$ the chaotic property of the double pendulum becomes evident by the strong influence of the IC on the values of the two measures we considered.

4 Discussion and limitations

Our experiments revealed that the traditional PINN framework is not suitable for simulating chaotic motion. As the goal of the PINN optimization is solely to minimize the combined loss function of data and physics loss term, it is able to "cheat" by either shifting the initial condition to reach a simpler function or by violating the physics in a small domain to again reach the simpler function. In both cases, the trajectories resulted in lower physics loss values compared to the reference solution.

We argue that these convergence problems stem, on the one hand, from the *spectral bias* that causes PINNs to converge to low-frequency solutions first. On the other hand, we observed that those solutions also exhibit lower magnitudes of the derivatives that comprise the calculation of the physics loss and hence explain low physics loss values. We find that reducing the computational domain combined with an appropriate loss weighting scheme allows convergence to the correct solution. In future research, we intend to extend our experiments to additional ODE-systems in order to substantiate our claims presented in this work.

References

[1] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2 2019.

[2] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," *Nature Reviews Physics*. [Online]. Available: www.nature.com/natrevphys

[3] C. L. Wight and J. Zhao, "Solving allen-cahn and cahn-hilliard equations using the adaptive physics informed neural networks," *Communications in Computational Physics*, vol. 29, pp. 930–954, 7 2020. [Online]. Available: https://arxiv.org/abs/2007.04542v1

[4] F. S. Costabal, Y. Yang, P. Perdikaris, D. E. Hurtado, and E. Kuhl, "Physics-informed neural networks for cardiac activation mapping," *Frontiers in Physics*, vol. 8, p. 42, 2 2020.

[5] K. Shukla, P. C. D. Leoni, J. Blackshire, D. Sparkman, and G. E. Karniadakis, "Physics-informed neural network for ultrasound nondestructive quantification of surface breaking cracks," *Journal of Nondestructive Evaluation*, vol. 39, pp. 1–20, 9 2020. [Online]. Available: https://link.springer.com/article/10.1007/s10921-020-00705-1

[6] S. Wang and P. Perdikaris, "Deep learning of free boundary and stefan problems," *Journal of Computational Physics*, vol. 428, p. 109914, 3 2021.

[7] E. Kharazmi, M. Cai, X. Zheng, Z. Zhang, G. Lin, and G. E. Karniadakis, "Identifiability and predictability of integer- and fractional-order epidemiological models using physics-informed neural networks," *Nature Computational Science 2021 1:11*, vol. 1, pp. 744–753, 11 2021. [Online]. Available: https://www.nature.com/articles/s43588-021-00158-0

[8] G. Pang, L. U. Lu, and G. E. Karniadakis, "fpinns: Fractional physics-informed neural networks," *https://doi.org/10.1137/18M1229845*, vol. 41, pp. A2603–A2626, 8 2019. [Online]. Available: https://epubs.siam.org/doi/abs/10.1137/18M1229845

[9] M. Yin, X. Zheng, J. D. Humphrey, and G. E. Karniadakis, "Non-invasive inference of thrombus material properties with physics-informed neural networks," *Computer Methods in Applied Mechanics and Engineering*, vol. 375, p. 113603, 3 2021.

[10] M. Raissi, A. Yazdani, and G. E. Karniadakis, "Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations," *Science*, vol. 367, pp. 1026–1030, 2 2020.

[11] S. Cai, Z. Wang, F. Fuest, Y. J. Jeon, C. Gray, and G. E. Karniadakis, "Flow over an espresso cup: Inferring 3d velocity and pressure fields from tomographic background oriented schlieren videos via physics-informed neural networks," 2021.

[12] S. Wang, Y. Teng, and P. Perdikaris, "Understanding and mitigating gradient pathologies in physics-informed neural networks a preprint," 2020. [Online]. Available: https://github.com/PredictiveIntelligenceLab/GradientPathologiesPINNs.

[13] X. Jin, S. Cai, H. Li, and G. E. Karniadakis, "Nsfnets (navier-stokes flow nets): Physics-informed neural networks for the incompressible navier-stokes equations," *Journal of Computational Physics*, vol. 426, 3 2020. [Online]. Available: https://arxiv.org/abs/2003.06496v1

[14] F. M. Rohrhofer, S. Posch, and B. C. Geiger, "On the pareto front of physics-informed neural networks," 5 2021. [Online]. Available: http://arxiv.org/abs/2105.00862

[15] A. D. Jagtap and G. E. Karniadakis, "Extended physics-informed neural networks (xpinns) : A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations." [Online]. Available: https://github.com/AmeyaJagtap/XPINNs.

[16] B. Moseley, A. Markham, and T. Nissen-Meyer, "Finite basis physics-informed neural networks (fbpinns): A scalable domain decomposition approach for solving differential equations a preprint," 2021.

[17] A. Krishnapriyan, A. Gholami, S. Zhe, R. Kirby, and M. W. Mahoney, "Characterizing possible failure modes in physics-informed neural networks," *Advances in Neural Information Processing Systems*, vol. 34, pp. 26 548–26 560, 2021.

[18] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. Hamprecht, Y. Bengio, and A. Courville, "On the spectral bias of neural networks," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 6 2019, pp. 5301–5310.

[19] Z.-Q. J. Xu, Y. Zhang, T. Luo, Y. Xiao, and Z. Ma, "Frequency principle: Fourier analysis sheds light on deep neural networks," *arXiv preprint arXiv:1901.06523*, 2019.

[20] B. Ronen, D. Jacobs, Y. Kasten, and S. Kritchman, "The convergence rate of neural networks for learned functions of different frequencies," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[21] Y. Cao, Z. Fang, Y. Wu, D.-X. Zhou, and Q. Gu, "Towards understanding the spectral bias of deep learning," *arXiv preprint arXiv:1912.01198*, 2019.

[22] S. Wang, H. Wang, and P. Perdikaris, "On the eigenvector bias of fourier feature networks: From regression to solving multi-scale pdes with physics-informed neural networks," 2020. [Online]. Available: https://github.com/

[23] F. M. Rohrhofer, S. Posch, C. Gößnitzer, and B. C. Geiger, "Understanding the difficulty of training physics-informed neural networks on dynamical systems," *arXiv preprint arXiv:2203.13648*, 2022.

[24] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 315–323.

[25] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[26] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," 2016. [Online]. Available: https://arxiv.org/abs/1603.04467

[27] M. A. Nabian, R. J. Gladstone, and H. Meidani, "Efficient training of physics-informed neural networks via importance sampling," *Computer-Aided Civil and Infrastructure Engineering*, vol. 36, no. 8, pp. 962–977, apr 2021. [Online]. Available: https://doi.org/10.1111%2Fmice.12685

[28] C. J. Arthurs and A. P. King, "Active training of physics-informed neural networks to aggregate and interpolate parametric solutions to the navier-stokes equations," *Journal of Computational Physics*, vol. 438, p. 110364, aug 2021. [Online]. Available: https://doi.org/10.1016%2Fj.jcp.2021.110364

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [No] There are none.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [No] There are no theoretical results in this work.
   (b) Did you include complete proofs of all theoretical results? [No] There are no theoretical proofs in this work.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] We provided the mean value and variance.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No] Not relevant for the results in this work.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [No] No assets used.
   (b) Did you mention the license of the assets? [No] No assets used.
   (c) Did you include any new assets either in the supplemental material or as a URL? [No] No assets used.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No] No data/information about individuals was used..
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] No data/information about individuals was used.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [No] No crowdsourcing or conducted research with human subjects was used.
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [No] No crowdsourcing or conducted research with human subjects was used.
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [No] No crowdsourcing or conducted research with human subjects was used.

A Appendix

A.1 Double pendulum

The motion of the planar double pendulum (see Figure 3) is described by the following nonlinear initial-value problem with $\boldsymbol{y} = [\theta_1, \theta_2]^T$

$$\boldsymbol{y}'' = \begin{bmatrix} f_1(\boldsymbol{y}, \boldsymbol{y}') \\ f_2(\boldsymbol{y}, \boldsymbol{y}') \end{bmatrix} = \begin{bmatrix} \frac{m_2 L_1 \omega_1^2 \sin(2\Delta\theta) + 2 m_2 L_2 \omega_2^2 \sin \Delta\theta + 2 g m_2 \cos\theta_2 \sin \Delta\theta + 2 g m_1 \sin\theta_1}{-2 L_1 (m_1 + m_2 \sin^2 \Delta\theta)} \\ \frac{m_2 L_2 \omega_2^2 \sin(2\Delta\theta) + 2(m_1+m_2) L_1 \omega_1^2 \sin \Delta\theta + 2 g (m_1+m_2) \cos\theta_1 \sin \Delta\theta}{2 L_2 (m_1 + m_2 \sin^2 \Delta\theta)} \end{bmatrix} \tag{4}$$

$$\text{subject to the initial conditions} \quad \boldsymbol{y}_0 = [\boldsymbol{y}(t_0), \boldsymbol{y}'(t_0)]^T \tag{5}$$

where $\theta_1' = \omega_1,\ \theta_2' = \omega_2,\ \Delta\theta = \theta_1 - \theta_2$, and $g = 9.81$ is the gravitational acceleration.

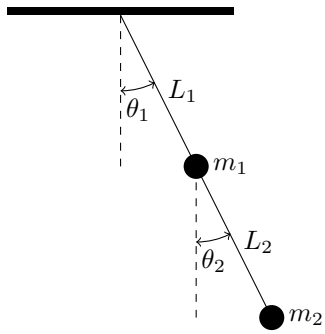


Figure 3: Schematic of the double pendulum with point masses $m_1,\ m_2$, and rod lengths $L_1,\ L_2$.

A.2 Physics-informed neural networks

The unknown solution $\boldsymbol{y}$ is approximated by a fully-connected deep neural network $\boldsymbol{y}_{\boldsymbol{w}}$ with $\boldsymbol{w}$ describing all trainable weights and biases of the network. For simulating the motion of the double pendulum, the PINN [1] is trained by minimizing the following combined loss function

$$L(\boldsymbol{w}) = \alpha L_{IC}(\boldsymbol{w}) + (1-\alpha) L_F(\boldsymbol{w}) \tag{6}$$

with the data-based loss

$$\begin{aligned} L_{IC}(\boldsymbol{w}) = \Big(\theta_{1,\boldsymbol{w}}(t_0) - \theta_1(t_0)\Big)^2 + \Big(\theta_{2,\boldsymbol{w}}(t_0) - \theta_2(t_0)\Big)^2 + \\ + \Big(\theta_{1,\boldsymbol{w}}'(t_0) - \theta_1'(t_0)\Big)^2 + \Big(\theta_{2,\boldsymbol{w}}'(t_0) - \theta_2'(t_0)\Big)^2 . \end{aligned} \tag{7}$$

and the physics-based loss consisting of the ODE-residuals of the physical system

$$L_{F_1}(\boldsymbol{w}) = \frac{1}{N_{col}} \sum_{i=1}^{N_{col}} \left| \theta_{\boldsymbol{w},1}''\left(t_{col}^{(i)}\right) - f_1(\theta_{\boldsymbol{w},1}, \theta_{\boldsymbol{w},2}, \theta_{\boldsymbol{w},1}', \theta_{\boldsymbol{w},2}') \right|^2 \tag{8}$$

$$L_{F_2}(\boldsymbol{w}) = \frac{1}{N_{col}} \sum_{i=1}^{N_{col}} \left| \theta_{\boldsymbol{w},2}''\left(t_{col}^{(i)}\right) - f_2(\theta_{\boldsymbol{w},1}, \theta_{\boldsymbol{w},2}, \theta_{\boldsymbol{w},1}', \theta_{\boldsymbol{w},2}') \right|^2 \tag{9}$$

$$L_F(\boldsymbol{w}) = L_{F_1}(\boldsymbol{w}) + L_{F_2}(\boldsymbol{w}) \tag{10}$$

where $\{t_i\}_{i=1}^{N_{col}}$ denotes the collocation points where the physics loss is evaluated. These collocation points can be chosen arbitrarily within the computational domain, either by using fixed locations or by adaptive resampling [27, 3, 28]. In this work, $N_{col} = 1024$ collocation points are sampled uniformly in each training epoch. [1]

[1] The code needed to reproduce our results is available at `https://anonymous.4open.science/r/How-PINNs-Cheat-F23F`