SimPINNs: Simulation-Driven Physics-Informed Neural Networks for Enhanced Performance in Nonlinear Inverse Problems

Sidney Besnard, Frédéric Jurie, Jalal M. Fadili

SIMPINNS: SIMULATION-DRIVEN PHYSICS-INFORMED NEURAL NETWORKS FOR ENHANCED PERFORMANCE IN NONLINEAR INVERSE PROBLEMS

Sidney Besnard[1,2] , Frederic Jurie[1], Jalal Fadili[1]

[1]Univ. Caen Normandie, ENSICAEN, CNRS [2]Safran Data Systems

ABSTRACT

This paper introduces a novel approach to solve inverse problems by leveraging deep learning techniques. The objective is to infer unknown parameters that govern a physical system based on observed data. We focus on scenarios where the underlying forward model demonstrates pronounced nonlinear behaviour, and where the dimensionality of the unknown parameter space is substantially smaller than that of the observations. Our proposed method builds upon physics-informed neural networks (PINNs) trained with a hybrid loss function that combines observed data with simulated data generated by a known (approximate) physical model. Experimental results on an orbit restitution problem demonstrate that our approach surpasses the performance of standard PINNs, providing improved accuracy and robustness.

***Index Terms*—** Inverse problems, Neural Networks, Physics-Informed, Simulation

1. INTRODUCTION

Inverse problems play a crucial role in science by allowing to unravel the hidden properties and processes behind observed data. They allow scientists to infer and understand phenomena that are otherwise difficult or impossible to observe or measure directly. These problems involve determining the parameters of a system from some available measurements. Inverse problems have far-reaching applications spanning a wide spectrum ranging from medical imaging to non-destructive control or space imaging, as we will see.

Data-driven machine learning methods, and in particular deep neural networks, have recently emerged as powerful alternatives to variational model-based approaches for solving inverse problems. These methods include supervised and unsupervised methods, such as the Deep Inverse Prior (DIP) [1], Unrolling [2, 3], Plug-and-play (PnP) [4, 5], and generative models [6], to name a few. Unrolling and PnP rely on neural networks to learn the regularization from the data. See for instance [**?**] for a comprehensive review.

However, these approaches only make sense if the output parameter space can be equipped with a suitable notion of regularity. This is certainly the case if the input parameters are in the form of a structured signal, but is not always the case as in our setting (think of inferring a few parameters that are not structured on a grid). A naive technique would then be to train a neural network using a dataset consisting of input-output pairs, where the input is the observed data and the output is the sought-after vector of parameters [7, 8, 9]. Clearly, the neural network learns to invert the forward model (i.e. the mapping between the observed data and parameters), with the hope that it would predict the unknown parameters for new observations. This approach leverages the ability of neural networks to capture intricate patterns and non-linear relationships in the data. Unfortunately, this type of technique is only applicable when a large set of training pairs is available, which is barely the case in most practical situations. Moreover, such approaches are completely agnostic to the forward model which would produce unrealistic solutions and may not generalize well.

Physics-informed neural networks (PINNs) were primarily proposed to solve partial differential equations (PDE) [10, 11, 12, 13]. Their core idea is to supplement the neural network training with information stemming from the measurement formation model, e.g. the PDE model. In turn, this allows to restrict the space of solutions by enforcing the output of the trained neural network to comply with the physical model as described by the PDE. In turn, these methods are expected to be trained with a smaller dataset.

An aspect to keep in mind regarding PINNs is that theyr are trained using only the reconstruction error, which reflects constraints, including initial or boundary conditions, as imposed by the PDE. However, in numerous cases, achieving a low reconstruction error does not guarantee an accurate prediction of the parameters (i.e., a low parameter error). Thus, it is essential to emphasize the network's requirement for some sort of induced regularization of the solution space during training.

Contributions. In this paper, our aim is to demonstrate the effectiveness of neural networks in dealing with non-linear inverse problems where the space parameter is unstructured. We propose a novel hybrid approach that leverages both physics-informed and data-driven methods and which uses simulated data to induce regularization of the solution. Indeed, given the difficulty in obtaining real training pairs (observations-parameters) for many real-world problems,

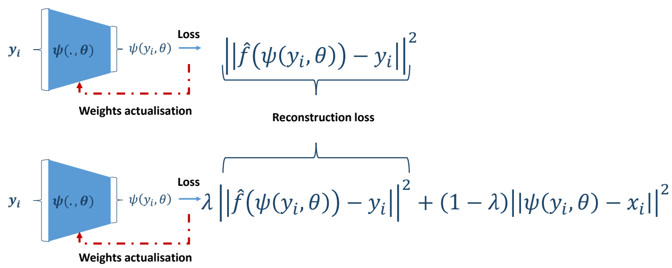

Fig. 1. Illustrations of the architectures used in PINNs and SimPINNs for inverse problems. The top part depicts the PINNs approach, where a neural network is trained to learn the inverse function of f, enabling the reconstruction of accurate x values from observed y values solely based on the observations and the underlying physics (unsupervised learning). In the bottom part, the SimPINNs (supervised) approach is shown, which utilizes the physics-based simulations to complement the training set with 'annotated' simulated data and regularize the solution.

simulations offer a convenient means to complement the training data set, though the simulated data might not be exact but accurate enough.

2. PROPOSED METHOD

Let $\mathcal{X} \subset \mathbb{R}^n$ be the space of parameters of the (physical) model, and $\mathcal{Y} \subset \mathbb{R}^m$ be the space of observations. An inverse problem consists in reliably recovering the parameters $x \in \mathcal{X}$ from noisy indirect observations

$$y = f(x) + \epsilon, \quad (1)$$

where $f : \mathcal{X} \to \mathcal{Y}$ is the forward operator, and ϵ stands for some additive noise that captures the measurement and possibly the modeling error. Throughout, we assume that f is smooth enough (at least continuously differentiable).

In the sequel, for a neural network with parameters (weights and biases) $\theta \in \Theta$, $\psi : (y, \theta) \in \mathcal{Y} \times \Theta \mapsto \psi(y, \theta)$ denotes its output.

Finally, we also assume that we have an approximate explicit model of f, sometimes referred to as a digital twin. This model, $\hat{f}$, is obtained by modelling the physical phenomena involved in the observations. It will be used later to generate simulated data.

2.1. PINNs for non-linear inverse problems

The key idea of PINNs is to incorporate the physical model into the cost function during the training process. For a neural network ψ and training samples $\{y_i : i = 1, \ldots, n\}$ with n samples, this amounts to solving the following minimization problem with the empirical loss:

$$\min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} \|y_i - \hat{f}(\psi(y_i, \theta))\|^2 \quad (2)$$

This loss function leverages the information provided by the physical forward model directly into the training loss. It is also a non-supervised method that relies solely on observations, without any knowledge of the parameter vector x_i corresponding to each y_i. Unfortunately, as was observed previously in the literature (e.g. in [14]), when f is not injective, there are infinitely many solutions $\psi(\cdot, \theta)$ which attain zero training error. This is because the forward model f may map multiple input vectors to the same output vector. For example, in the linear case, the action of f is invariant along its null space. This suggests that training a reconstruction network as (2) only from the observed data, without any additional assumptions or constraints, is not viable. Possible workarounds include explicitly constraining the output of the reconstruction network through regularization (and we are back to the variational world), or introducing invariances such as in [14]. In the forthcoming section, we will describe an alternative based on exploiting the forward model to simulate input-output pairs. This approach will help to regularize the training process and make it more robust to the non-injectivity of f.

2.2. SimPINNs: Simulation aided PINNs

In many areas of science, obtaining pairs of input (parameters)-output (observations) training data, can be a significant challenge. This can be due to various reasons, including that data are difficult or expensive to acquire. This is for instance the case in large instruments in physics. Furthermore, even if such pairs of data can be acquired, they are available only in limited quantity, which often impedes the use of data-intensive machine learning approaches.

There are however situations where even if such data are unavailable, it is possible to artificially generate the input-output pairs by leveraging knowledge of the forward model in (1), even if the latter is only approximately known. This involves generating a parameter/input vector x sampled from the range of possible input values in the model or based on the known distribution of input data. We propose to compute the corresponding simulated observation $\hat{f}(x)$, i.e. without noise. It is important to note that the forward model serves anyway as an approximation of the underlying physical phenomenon f it represents, and the simulated observation can only be considered as a perturbed version of the unknown observation due to model imperfections.

Summarizing our discussion above, we propose to train a neural network ψ by replacing (2) with

$$\min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}_\lambda(x_i, y_i, \theta), \text{where} \quad (3)$$

$$\mathcal{L}_\lambda(x, y, \theta) = \lambda \|y - \hat{f}(\psi(y, \theta))\|^2 + (1 - \lambda)\|\psi(y, \theta) - x\|^2.$$

Here $\lambda \in]0, 1[$ balances between the two terms: fidelity to the observation and reconstruction error. The determination of an optimal value for λ can be a challenging task. In our study, we employed an empirical approach to estimate this value by performing cross-validation.

In the case of real observations, the value of x is unknown, making it impossible to calculate the second term in the loss function in (3). Therefore, only the first term, which focuses on reconstruction fidelity, is used for such data.

The ratio between the number of real data, denoted as N_o (where only the observation is known), and the number of simulated data, denoted as N_s, plays a significant role in the analysis. Consequently, the influence of this ratio have to be thoroughly examined through experimental studies.

3. EXPERIMENTAL RESULTS

We validate the proposed method by applying it to an orbit restitution problem, where the dimension of $\mathcal{X}$ ($n = 6$) is relatively smaller than that of $\mathcal{Y}$ ($m = 64^2$). This problem encompasses several intriguing aspects that make it particularly compelling. Firstly, the underlying physics and the involved forward operator exhibit non-linearities, which is a primary focus in real-world research problems. Secondly, an orbit is defined by six orbital elements, while the received data exists in a significantly larger space, such as the image space in our case. Consequently, the forward operator maps from a smaller parameter space to a substantially larger image space. The third aspect of this problem pertains to the challenging nature of obtaining input-output pairs, as it requires integrating raw acquisition with non-trivial evaluations and determination of orbit parameters.

It is important to note that despite initially appearing simple due to the presence of more equations than unknown variables, this problem poses additional difficulties. The involved physics operator is non-linear, making the problem extremely ill-posed with non-trivial equivalence classes. This complexity makes it more challenging than it may look at first glance. In the subsequent sections, we present the details of the problem, including the experimental settings and the obtained results.

3.1. Problem and dataset

The objective is to invert an orbit propagator using images obtained from a simulated sensor.

The forward operator performs the projection of orbits expressed in the Terrestrial Reference Frame (TRF) onto an image, similar to a ground track orbit projection. Consequently, the received data correspond to the projection of an orbit onto a sensor positioned on the Earth (see Figures 2 and 3).

For this problem, we opt to represent each orbit using 3 Keplerian parameters: inclination, eccentricity, and periapsis.

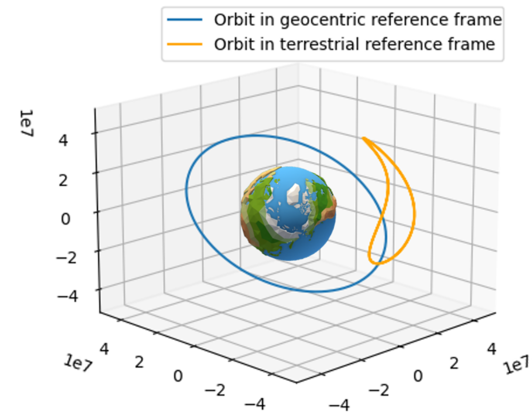

Fig. 2. Orbit Restitution: Visualization of an orbit in multiple reference frames. In this example, the depicted orbit has an inclination of 45 degrees, an eccentricity of 0.4, and an apogee of 42164×10^3 meters.

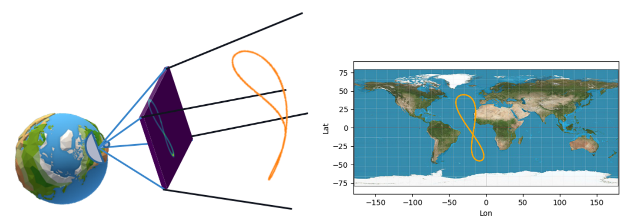

Fig. 3. Orbit restitution: Simulation of the orbit projection, as depicted in Figure 2. The left image exhibits the simulated sensor exhibiting the projected orbit, while the right image depicts the orbit projected onto the ground plane.

This choice leads to a wide range of images and results, while avoiding trivial equivalence classes in the parameter set.

In this problem, the forward operator is the mapping:

$$\begin{array}{rcccl} f & : & [0,1[\times[0,2\pi[\times[0,2\pi[& \rightarrow & \mathcal{I}_{64\times 64}(\mathbb{R}) \\ & & e, i, \omega & \mapsto & f(e, i, \omega). \end{array} \quad (4)$$

This mapping simulates the entire system, encompassing the satellite position (achieved through an orbital propagator), the radiation pattern of the antenna, and the projection to the final image (assumed to be 64×64 pixels). In this experiment, it is assumed that the approximate and actual physical models are identical, i.e. $\hat{f} = f$. To compute the satellite position while preserving differentiability of $\hat{f}$, an analytical Keplerian propagator is employed.

Finally, the neural network used in this problem consists of 5 dense layers, with each layer containing 784 neurons and employing a ReLU activation function. The neural network is then trained with the two previously defined loss functions in (2) and (3) (see Figure 1)

ArgPer($\times 10^{-5}$)	Number of real observations (N_o)				
N_s	0	1000	10000	20000	40000
0	-	2.4541	1.2970	0.9947	0.9081
1000	2.0546	2.3412	1.1546	0.9731	1.1742
10000	1.1470	2.1942	0.8591	0.7854	1.0644
20000	0.8401	0.7721	0.6924	0.6571	0.6431
40000	0.4541	0.4412	0.4201	0.4121	0.4212

Eccentricity($\times 10^{-5}$)	Number of real observations (N_o)				
N_s	0	1000	10000	20000	40000
0	-	1.5587	1.4424	1.1232	0.8638
1000	1.3545	1.5225	0.7542	1.1036	0.8452
10000	1.6556	1.5245	0.7054	1.1023	0.7214
20000	1.4684	1.0845	1.2781	0.6306	0.6251
40000	0.3895	0.3856	0.3776	0.3435	0.2861

Inclination	Number of real observations (N_o)				
N_s	0	1000	10000	20000	40000
0	-	0.0132	0.0021	0.0009	0.0004
1000	0.0134	0.0114	0.0017	0.0012	0.0003
10000	0.0022	0.0028	0.0024	0.0008	0.0007
20000	0.0014	0.0007	0.0007	0.0007	0.0005
40000	0.0002	0.0002	0.0002	0.0001	0.0001

Reconstruction	Number of real observations (N_o)				
N_s	0	1000	10000	20000	40000
0	-	1.3243	1.1667	0.8742	0.3023
1000	1.4581	0.7064	0.1314	0.0989	0.0852
10000	0.2467	0.1510	0.1394	0.0920	0.0955
20000	0.0966	0.0831	0.0968	0.0750	0.0701
40000	0.0304	0.0305	0.0298	0.0271	0.0251

Table 1. Error evaluation across various parameters, using a test set containing only observations.

3.2. Results and analysis

As shown in Table 1, the SimPINNs method effectively leverages information from both real observations and simulated data. It systematically outperforms the PINNs method in terms of parameter reconstruction error; PINNs is the row with $N_s = 0$. The most favorable results are achieved when employing the largest number of real and generated data points ($N_o = N_s = 40k$), with an average improvement of a factor of 2 for the parameter error and a factor 10 for the reconstruction error, compared to the unsupervised PINNs method.

The impact of incorporating orbital physics in the forward operator becomes evident in the significant benefit it provides for image reconstruction in this particular use case. Due to the influence of orbital dynamics, even small changes in the parameters can have a substantial impact on the satellite's orbit and drastically alter the resulting projection on the observed image. As a result, two images with nearly identical parameters can exhibit significant differences. In this context, the reconstruction loss plays a crucial role in assisting the network in handling these high-gradient values that may not be adequately captured by the supervised loss alone. By combining both the simulated and real data and training via (3), the approach leverages the strengths of both data types and provides an improved solution with reduced parameter error.

Figure 4 presents a selection of projected orbits along with their corresponding reconstructions.

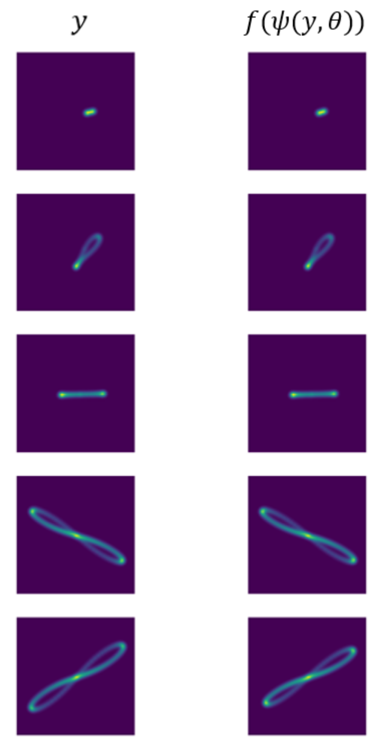

Fig. 4. Reconstruction Results using the SimPINNs Method.

4. CONCLUSIONS

This paper explores a physics-informed neural network-based framework to solve non-linear inverse problems on unstructured data. The proposed method, SimPINNs, leverages the physics model to generate input-output training data. The study demonstrates that simulation-aided training provides more information compared to conventional PINNs or vanilla neural network training. By utilizing the approximated physics operator, the model achieves improved learning and generalization over test datasets. SimPINNs shows potential for addressing challenging inverse problems with limited training data, offering new insights into physics-informed and simulation-aided training of neural networks for inverse problems.

Acknowledgements

The research presented in this paper is, in part, funded by the French National Research Agency (ANR) through the grant ANR-19-CHIA-0017-01-DEEP-VISION.

5. REFERENCES

[1] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky, "Deep image prior," International Journal of Computer Vision, vol. 128, no. 7, pp. 1867–1888, mar 2020.

[2] Pei Peng, Shirin Jalali, and Xin Yuan, "Solving inverse problems via auto-encoders," IEEE J. Sel. Areas Inf. Theory, vol. 1, no. 1, pp. 312–323, 2020.

[3] Karol Gregor and Yann LeCun, "Learning fast approximations of sparse coding," in ICML. 2010, pp. 399–406, Omnipress.

[4] Singanallur V. Venkatakrishnan, Charles A. Bouman, and Brendt Wohlberg, "Plug-and-play priors for model based reconstruction," in 2013 IEEE Global Conference on Signal and Information Processing, 2013, pp. 945–948.

[5] Zihui Wu, Yu Sun, Alex Matlock, Jiaming Liu, Lei Tian, and Ulugbek S. Kamilov, "SIMBA: Scalable inversion in optical tomography using deep denoising priors," IEEE Journal of Selected Topics in Signal Processing, vol. 14, no. 6, pp. 1163–1175, oct 2020.

[6] Ashish Bora, Ajil Jalal, Eric Price, and Alexandros G. Dimakis, "Compressed sensing using generative models," in Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017, Doina Precup and Yee Whye Teh, Eds. 2017, vol. 70 of Proceedings of Machine Learning Research, pp. 537–546, PMLR.

[7] Christopher A. Metzler and Gordon Wetzstein, "Deep s3pr: Simultaneous source separation and phase retrieval using deep generative models," in ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021, pp. 1370–1374.

[8] Christopher Metzler, Phillip Schniter, Ashok Veeraraghavan, and Richard Baraniuk, "prDeep: Robust phase retrieval with a flexible deep network," in Proceedings of the 35th International Conference on Machine Learning, Jennifer Dy and Andreas Krause, Eds. 10–15 Jul 2018, vol. 80 of Proceedings of Machine Learning Research, pp. 3501–3510, PMLR.

[9] Yair Rivenson, Yibo Zhang, Harun Günaydın, Da Teng, and Aydogan Ozcan, "Phase recovery and holographic image reconstruction using deep learning in neural networks," Light: Science and Applications, vol. 7, no. 2, pp. 17141–17141, oct 2017.

[10] Maziar Raissi, Paris Perdikaris, and George Em 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, 2019.

[11] Maziar Raissi, "Deep hidden physics models: Deep learning of nonlinear partial differential equations," The Journal of Machine Learning Research, vol. 19, no. 1, pp. 932–955, 2018.

[12] Ziya Uddin, Sai Ganga, Rishi Asthana, and Wubshet Ibrahim, "Wavelets based physics informed neural networks to solve non-linear differential equations," Scientific Reports, vol. 13, no. 1, pp. 2882, Feb 2023.

[13] Shengze Cai, Zhicheng Wang, Sifan Wang, Paris Perdikaris, and George Em Karniadakis, "Physics-Informed Neural Networks for Heat Transfer Problems," Journal of Heat Transfer, vol. 143, no. 6, pp. 060801, 04 2021.

[14] Dongdong Chen, Julian Tachella, and Mike Davies, "Equivariant imaging: learning beyond the range space," in ICCV. 2021, pp. 4379–4388, IEEE/CVF.