NEURAL NETWORK DIFFERENTIAL EQUATION SOLVERS ALLOW UNSUPERVISED ERROR ESTIMATION AND CORRECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural Network Differential Equation (NN DE) solvers have surged in popularity due to a combination of factors: computational advances making their optimization more tractable, their capacity to handle high dimensional problems, easy interpretability, etc. However, most NN DE solvers suffer from a fundamental limitation: their loss functions are not explicitly dependent on the errors associated with the solution estimates. As such, validation and error estimation usually requires knowledge of the true solution. Indeed, when the true solution is unknown, we are often reduced to simply hoping that a "*low enough*" loss implies "*small enough*" errors, since explicit relationships between the two are not available. In this work, we describe a general strategy for efficiently constructing error estimates and corrections for Neural Network Differential Equation solvers. Our methods do not require *a priori* knowledge of the true solutions and obtain explicit relationships between loss functions and the errors. In turn, these explicit relationships allow for the unsupervised estimation and correction of the model errors.

1 INTRODUCTION

Deep learning has heralded new methods for many scientific disciplines – the field of numerical methods for differential equations has been no exception (1; 2; 3; 4). Deep neural network based differential equations (NN DE) solvers have been proposed under a variety of different names (PINNs (2), DGM (3), etc) – each catering to various classes of problems. However, they all share certain common features: the use of a known differential equation (DE) as the central component of an appropriate loss function, the use of existing knowledge (boundary conditions, experimental/synthetic data, etc) to constrain the search for solutions, randomized optimization methods that sample from the domain of interest at a requisite resolution, etc.

We investigate another common facet of many NN DE solvers: the lack of unsupervised error quantification/correction methods to estimate model errors without prior knowledge of the solution. Most solvers use the equation based loss functions as a surrogate measure for the error. However, while such a measure is intuitively related to the error in the solution model, an explicit description of that connection is mandatory, if error quantification is to be done without knowing the solution.

We achieve these goals by explicitly relating the loss terms and the model error. We showcase how these connections allow results on the model error that don't rely on prior knowledge of solution. We propose techniques by which these results can be used to build significantly more efficient NN DE solvers, with only marginal increases in computational complexity. We formalize our ideas into three theorems, two inequalities, and two algorithms. We validate our claims with a collection of numerical experiments on several non-trivial DEs.

For the sake of readability and simplicity, all proofs have been rigorously presented in the appendices, while the main text simply reports the results and discusses their significance.

An associated codebase is also provided, with inbuilt options for the DEs already studied as part of this work. However, the codebase has been designed so that the users may easily add their own DEs of interest (the assumptions under which this work is valid should be general enough for a wide variety of DEs from many different scientific disciplines).

2 NEURAL NETWORK DIFFERENTIAL EQUATION SOLVERS

For ease of discussion, we let $\mathcal{D} \subseteq \mathbb{R}^d$ denote some closed, bounded, path connected domain of interest for some differential equation (DE). Let $\partial\mathcal{D} \subseteq \mathcal{D}$ be the portion of the domain over which some constraint conditions on the solution exist (usually obtained as boundary conditions, empirical data, etc). Assume that our chosen DE, when given unique constraints over $\partial\mathcal{D}$, admits a unique solution $\Phi : \mathcal{D} \to \mathbb{R}^D$.

We wish to consider a (possibly non-linear) differential equation operator $\mathbf{F} : G \to H$, where G and H are some suitable spaces of functions over $\mathcal{D}$ ($\Phi \in G$). We decompose $\mathbf{F}$ as

$$\mathbf{F}[\bullet] = \mathbf{L}[\bullet] + \mathbf{N}[\bullet] + \mathbf{C} \tag{1}$$

where $\mathbf{L}$ represents the term(s) which depend linearly on Φ, $\mathbf{N}$ the represents the term(s) which depend non-linearly on Φ, and $\mathbf{C}$ are the terms independent of Φ. This additive decomposition into linear, nonlinear, and constant terms is always possible: for linear DEs, $\mathbf{N} \equiv 0 \equiv \mathbf{C}$. We have

$$\mathbf{F}[\Phi] = \mathbf{L}[\Phi] + \mathbf{N}[\Phi] + \mathbf{C} = 0 \ .$$

Let us assume we wish to construct an NN based approximation $\mathcal{N} : \mathcal{D} \to \mathbb{R}^D$. Let us also assume the NN uses analytic activation functions so that $\mathcal{N} \in \mathcal{C}^\infty$. Let W be its width (neurons per hidden layer) and $D_\mathcal{N}$ be its depth (number of hidden layers). Finally, let $\mathbf{w} \equiv \{b^1_1, w^1_{11}, w^1_{12}, \dots, b^2_1, w^2_{11}, w^2_{12}, \dots\} \in \mathcal{M} \subseteq \mathbb{R}^M$ be the M weights and biases of the NN.

2.1 EXISTENCE AND COMPLEXITY

Feedforward NNs with ReLU activation and $W \geq d + 4$ can arbitrarily well approximate any Lebesgue integrable function $\Phi : \mathcal{D} \to \mathbb{R}^D$ w.r.t. L^1 norm, provided that $\mathcal{D}$ is some compact subset of $\mathbb{R}^d$ and enough depth $D_\mathcal{N}$ is provided to the NN (5). The same holds for NNs with ReLU activation and $D_\mathcal{N} \geq \log_2(d + 1)$ provided that they are wide enough (6). Hence, there exist $\mathbf{w}$ such that $\mathcal{N}$ can arbitrarily approximate any $\Phi \in \mathcal{C}^k$ over $\mathcal{D}$, given large enough $(W, D_\mathcal{N})$.

These theoretical guarantees may be practically realized by leveraging the minimum regularity expected from Φ. Recall $\mathcal{W}^{k,\infty}(\mathcal{D})$ is the Sobolev space of order k, on compact $\mathcal{D} \subseteq \mathbb{R}^d$, w.r.t. the L^∞ norm. Let us assume that $\Phi \in \mathcal{W}^{k,\infty}(\mathcal{D})$. There exists some $\mathcal{N}$ with $(W, D_\mathcal{N})$ that can ε - approximate $\Phi : \mathcal{D} \to \mathbb{R}$, if $W = d + 1,\ D_\mathcal{N} = \mathcal{O}(\text{diam}(\mathcal{D})/\omega_f^{-1}(\varepsilon))^d$ (7), where

$$\omega_f^{-1}(\varepsilon) = \sup\{\delta : \omega_f(\delta) \leq \varepsilon\}, \quad \delta = |\mathbf{x}_1 - \mathbf{x}_2|, \quad \mathbf{x}_1, \mathbf{x}_2 \in \mathcal{D}.$$

There also exist $\mathcal{N}$ that ε - approximate $\Phi : \mathcal{D} \to \mathbb{R}$, if $M = \mathcal{O}(ln(\frac{1}{\varepsilon})\varepsilon^{-d/k}),\ D_\mathcal{N} = \mathcal{O}(ln(\frac{1}{\varepsilon}))$ (8).

2.2 OPTIMIZATION

The loss function $\mathcal{L}$ to train $\mathcal{N}$ usually takes the following form (2; 3; 4):

$$\begin{aligned}\mathcal{L} &= \mathbb{E}_{x_1 \in \mathcal{D}}\left[\|\mathbf{F}[\mathcal{N}(x_1)]\|_p^p\right] + \mathbb{E}_{x_2 \in \partial\mathcal{D}}\left[\|\Phi(x_2) - \mathcal{N}(x_2)\|_p^p\right] \\ &= \mathbb{E}_{x_1 \in \mathcal{D}}\left[\|\mathbf{L}[\mathcal{N}(x_1)] + \mathbf{N}[\mathcal{N}(x_1)] + \mathbf{C}\|_p^p\right] + \mathbb{E}_{x_2 \in \partial\mathcal{D}}\left[\|\Phi(x_2) - \mathcal{N}(x_2)\|_p^p\right]\end{aligned} \tag{2}$$

where $\|\cdot\|_p$ is the usual p-norm on $\mathbb{R}^D$. Variants of Stochastic Gradient Descent (SGD) are *almost always* capable of *eventually* reaching adequately small loss values for large enough $(W, D_\mathcal{N})$ under such loss (9). Thus as $\mathcal{L} \to 0$ during optimization, the uniqueness of Φ implies $\mathcal{N} \to \Phi$ over $\mathcal{D}$ (formally shown in Theorem 1).

ALTERNATIVE PARAMETERIZATION OF $\partial\mathcal{D}$

Sometimes, the constraint conditions may be enforced by using the following parametrization as the model for Φ at some arbitrary $x_1 \in \mathcal{D}$:

$$\mathcal{N}(x_1) = \Phi(x_2) + \text{dist}(x_1, x_2)\mathcal{N}^o \tag{3}$$

where $x_2 \in \partial\mathcal{D}$ is some corresponding *nearest* constraint point to x_1. Here, $\mathcal{N}^o$ represents the NN and $\text{dist}(x_1, x_2)$ is some metric that enforces that the model $\mathcal{N}$ is always exact over $\partial\mathcal{D}$, while allowing the NN flexibility to *learn* Φ elsewhere (ex: $1 - e^{-||x_1 - x_2||}$, as used in (10)). This parametrization can eliminate the need for the $\mathbb{E}_{x_2 \in \partial\mathcal{D}}$ term in Eq. 2. Our work is applicable in such scenarios too.

3 ERROR BOUNDS AND CONVERGENCE FOR NN DE SOLVERS

Define $\Phi_{\epsilon_1} := \Phi - \mathcal{N}$ as the error in our solution model. Some simple observations about $\mathcal{N}, \Phi_{\epsilon_1}$, the convergence of $\mathcal{L}$, and their inter-plays underpin the entirety of this work:

1. $\mathbf{F}[\mathcal{N}]$ in Eq. 2 is not explicit in Φ_{ϵ_1}: there is no explicit relation between $\mathcal{L}$ and Φ_{ϵ_1} over $\mathcal{D} - \partial\mathcal{D}$, where Φ is unknown. We don't know **if/how** $\mathbb{E}\Big[||\Phi_{\epsilon_1}||\Big] \to 0$ as $\mathcal{L} \to 0$.
2. Thus, Φ_{ϵ_1} associated with $\mathcal{N}$ is not estimable over $\mathcal{D} - \partial\mathcal{D}$ in standard NN DE solvers.
3. Optimization performance saturates when $\mathcal{N}$ settles around a local minima(s) of $\mathcal{L}$.

The first problem can be immensely intractable in nature and is often handled separately for different kinds of DEs. Formal discussion on this aspect of NN DE solvers in some kinds of problems can be found in (3; 10; 11; 12; 13). Fortunately, the scope of this work concerns NN DE solvers where our assumptions allow us to strongly combat this aspect of the problem. In particular, since $\mathcal{L} \to 0 \implies \mathbf{F}[\mathcal{N}] \to 0$, we present the following results, with formal proofs in the appendix.

Theorem 1. *Let G and H be two Banach spaces. Suppose that $\mathbf{F} : G \supset U \to H$ is Fréchet differentiable in U, that the derivative of $\mathbf{F}$ at $\Phi \in U$ is invertible, and $\mathbf{F}[\Phi] = 0$. Then, there exists a neighbourhood $V \subset H$ of 0 small enough such that*

$$\mathbf{F}[\mathcal{N}] \longrightarrow 0 \implies \mathcal{N} \longrightarrow \Phi.$$

Theorem 2. *Under the same assumptions as Theorem 1, we have*

$$\|\Phi_{\epsilon_1}\| = \mathcal{O}\left(\|\mathbf{F}[\mathcal{N}]\|\right).$$

A common observation made for NN DE solvers is that $\mathbb{E}\Big[||\mathbf{F}[\mathcal{N}]||\Big] \propto \mathbb{E}\Big[||\Phi_{\epsilon_1}||\Big]$. We find that this particular result stands over a significantly relaxed class of assumptions on $\mathbf{F}$: we have already obtained it as Theorem 2. It also validates the intuitive wisdom of using the loss function as a surrogate for error, especially when the associated constant of proportionality for the Lipschitz continuous $\mathbf{F}$ is of order unity.

However, while Theorems 1 and 2 partially mitigate the concerns raised in the first two observations, they still only give weak global estimates about the behavior of Φ_{ϵ_1} over $\mathcal{D}$. We can obtain stronger estimates on $||\Phi_{\epsilon_1}||$ by assuming structural information on the Φ dependent terms $\mathbf{L}, \mathbf{N}$ and on the interactions they might have with each other.

In particular, let us assume that $\mathrm{D}\mathbf{N}$ (Fréchet derivative of the nonlinear component $\mathbf{N}$ of $\mathbf{F}$) is a positive (or negative) definite operator (such an assumption is often taken to avoid degenerate systems). Define $\mathcal{N}_s = \Phi - s\Phi_{\epsilon_1}$. Further, let:

$$H_{\min} := \inf_{s\in[0,1]} \inf \left\{|\lambda| \,\middle|\, \lambda \in \mathrm{Spec}\left(\mathrm{D}\mathbf{N}[\mathcal{N}_s]\right)\right\}$$

Let $F_{\max}$ be the maximum of $||\mathbf{F}[\mathcal{N}]||$ over $\mathcal{D}$ for some model $\mathcal{N}$ that has finished optimizing. We then have the following inequality on $||\Phi_{\epsilon_1}||$ (see Appendix C for proof and discussion on assumptions):

$$||\Phi_{\epsilon_1}(\mathbf{x})|| < \frac{F_{\max}}{H_{\min}} \qquad \mathbf{x} \in \mathcal{D} \tag{4}$$

Inequality 4 formalizes the wisdom in the heuristics used for designing NN DE solvers and how error is intuitively estimated for such solvers, without the use of explicit connections between $\mathbf{F}[\mathcal{N}], \Phi_{\epsilon_1}$. A variation of Inequality 4 we derive in Appendix C also showcases how NN DE solver design can encode knowledge/guesses on the system's behavior. Such information can be used to fine-tune the assumptions that go into obtaining the presented inequality and lead us to modifications needed on the bound. We provide more details in Appendix C (which shall be especially relevant to users that wish to use our codebase for systems other than the ones we provide).

However, even with the stronger inequalities on $||\Phi_{\epsilon_1}||$ (which itself might still give severe overestimates, if $H_{\min}$ is not properly controlled for), the fact remains that we have no information regarding pointwise behavior of Φ_{ϵ_1} over $\mathcal{D} - \partial\mathcal{D}$. There is a scarcity of work on algorithms that can explicitly

estimate Φ_{ϵ_1} over $\mathcal{D} - \partial\mathcal{D}$ using only the information available to the model $\mathcal{N}$. More often than not, precise error estimations rely on actual knowledge of Φ over entirety of $\mathcal{D}$. In the next section, we shall provide an unsupervised algorithm to estimate Φ_{ϵ_1} over $\mathcal{D} - \partial\mathcal{D}$, without knowledge of Φ.

This endeavor, combined with the third observation will directly lead to another useful application. For any nontrivial DE, it is improbable that training $\mathcal{N}$ over $\mathcal{D}$ will yield the exact solution once a stable minima(s) of optimization is reached (stability is meant here in the sense of a converged model that is unlikely to find a significantly better minima in a *reasonable* amount of iterations). Indeed, it is demonstrably impossible if the DE does not admit closed form solutions, since $\mathcal{N}$ is always a closed form expression. Any further training would not provide any meaningful gains in accuracy. However, in that regime, even a moderately good estimate for Φ_{ϵ_1} would be immediately useful: we could simply add that estimate to the original base model and produce a better model (albeit, at additional optimization and model memory costs).

From the discussion above, it is clear that an explicit relationship between $\mathbf{F}[\mathcal{N}]$ and Φ_{ϵ_1} would imply additional benefits not covered by our Theorems and Inequality 4. We show that is a very achievable goal under the established assumptions.

4 Error Estimation and Correction

4.1 Explicit relationships between Φ_{ϵ_1} and $\mathbf{F}[\mathcal{N}]$

We begin by noticing the following:

$$\begin{aligned}\mathbf{F}[\mathcal{N}] = \quad & \mathbf{L}[\mathcal{N}] + \mathbf{N}[\mathcal{N}] + \mathbf{C} \quad = \quad \mathbf{L}[\Phi - \Phi_{\epsilon_1}] + \mathbf{N}[\mathcal{N}] + (\mathbf{N}[\Phi] - \mathbf{N}[\Phi]) + \mathbf{C} \\ &= -\mathbf{L}[\Phi_{\epsilon_1}] + \mathbf{N}[\mathcal{N}] - \mathbf{N}[\mathcal{N} + \Phi_{\epsilon_1}] \\ &= -\mathbf{L}[\Phi_{\epsilon_1}] - \mathbf{N}'[\mathcal{N}, \Phi_{\epsilon_1}] \end{aligned} \tag{5}$$

where $\mathbf{N}'[\mathcal{N}, \Phi_{\epsilon_1}] := \mathbf{N}[\mathcal{N} + \Phi_{\epsilon_1}] - \mathbf{N}[\mathcal{N}]$. We have a new equation operator $\mathbf{F}_1$

$$\mathbf{F}_1[\Phi_{\epsilon_1}] = \mathbf{F}[\mathcal{N}] + \mathbf{L}[\Phi_{\epsilon_1}] + \mathbf{N}'[\mathcal{N}, \Phi_{\epsilon_1}] = 0 \tag{6}$$

where Φ_{ϵ_1} is the only unknown quantity. The uniqueness of Φ implies that if $\mathcal{N}$ is fixed, then Φ_{ϵ_1} is a unique solution to the DE given by Eq. 6, which also implies [1]

$$\mathbf{F}_1[\Phi_{\epsilon_1}] = \mathbf{F}[\mathcal{N} + \Phi_{\epsilon_1}] = 0$$

Note that $\mathbf{N}'$ has to be obtained such that terms explicitly dependent on Φ disappear from Eq. 5: that is what allows tight bounds on Φ_{ϵ_1} in (10), without depending upon explicit knowledge of Φ (Inequality 4 is a strong generalization of that result). As a motivating example, consider the cases when the nonlinearities in $\mathbf{F}$ are degree 2 polynomials: $\mathbf{N}[\,] = [\,] \cdot [\,]$. In that case, the simple transformations above give us:

$$\mathbf{F}[\mathcal{N}] + \mathbf{L}[\Phi_{\epsilon_1}] + \mathbf{N}'[\mathcal{N}, \Phi_{\epsilon_1}] = 0 : \quad \mathbf{N}'[\mathcal{N}, \Phi_{\epsilon_1}] = (\Phi_{\epsilon_1} \cdot \Phi_{\epsilon_1} + \mathcal{N} \cdot \Phi_{\epsilon_1} + \Phi_{\epsilon_1} \cdot \mathcal{N})$$

The transformation technique itself is not important, as long as it is accurate. Indeed, we are always able to directly leverage the $\mathbf{N}[\mathcal{N}] - \mathbf{N}[\mathcal{N} + \Phi_{\epsilon_1}]$ form (or even $\mathbf{F}[\mathcal{N} + \Phi_{\epsilon_1}]$) [2] for optimization purposes in the proposed Algorithm 1. The general idea is that we are trying to obtain an equation where both $\mathbf{F}[\mathcal{N}], \Phi_{\epsilon_1}$ appear, but Φ does not. Thus, with Eq. 6 above, we may rectify one major issue with Eq. 2: the residual $\mathbf{F}[\mathcal{N}]$ can now be explicitly related to Φ_{ϵ_1} over the entirety of $\mathcal{D}$.

[1] Note that $\mathbf{F}_1[(\quad)] = \mathbf{F}[\mathcal{N} + (\quad)]$ for any appropriate function $(\quad)$

[2] However, it is sometimes profitable to obtain an explicit, separable expression for $\mathbf{N}'[\mathcal{N}, \Phi_{\epsilon_1}]$. For example, whenever Taylor expansions can be used,

$$\mathbf{N}'[\mathcal{N}, \Phi_{\epsilon_1}] \equiv \mathcal{T}_1(\mathcal{N})\Phi_{\epsilon_1} + \frac{\Phi_{\epsilon_1}^{\dagger}\mathcal{T}_2(\mathcal{N})\Phi_{\epsilon_1}}{2!} + ...$$

where $\mathcal{T}_i(\mathcal{N})$ are the ith order Taylor terms. These can be useful for analysis, since $\mathcal{N}, \Phi_{\epsilon_1}$ appear in multiplicatively separable terms in the new expressions ((10) used these forms to bound Φ_{ϵ_1}). However, we obtained Inequality 4 in a way that renders such transformations superfluous: our bounds stand under weaker assumptions.

Algorithm 1 Unsupervised Estimation and Correction of Φ_{ϵ_1}

procedure ERRORCORRECT
  initialize NN DE solver $\mathcal{N} : \mathcal{D} \to \mathbb{R}$.
  while $\mathcal{N}$ not converged **do** ▷ Initial model $\mathcal{N}$ over $\mathcal{D}$
    Optimize $\mathcal{N}$ for a step using $\mathcal{L}$ (Eq. 2)
  end while
  Freeze the parameters of $\mathcal{N}$ ▷ Fix $\mathcal{N}$ to start error analysis
  initialize Error estimation model $\mathcal{N}_{\epsilon_1}$.
  while $\mathcal{N}_{\epsilon_1}$ not converged **do** ▷ Error estimation over $\mathcal{D}$
    Optimize $\mathcal{N}_{\epsilon_1}$ for a step using $\mathcal{L}_\epsilon$ (Eq. 8)
  end while
  return $\mathcal{N} + \mathcal{N}_{\epsilon_1}$ ▷ Error correction over $\mathcal{D}$
end procedure

Eq. 6 immediately hints at a method to estimate Φ_{ϵ_1}: we use Eq. 6 with another network model, $\mathcal{N}_{\epsilon_1}$, in the manner we used $\mathcal{N}$ with Eq. 2, to obtain an approximation for Φ_{ϵ_1}. More precisely, once the optimization of $\mathcal{N}$ has finished and the network has saturated its capacity to model Φ, we use $\mathbf{F}_1$:

$$\mathbf{F}_1[\mathcal{N}_{\epsilon_1}] = \mathbf{F}[\mathcal{N}] + \mathbf{L}[\mathcal{N}_{\epsilon_1}] + \mathbf{N}'[\mathcal{N}, \mathcal{N}_{\epsilon_1}] \tag{7}$$

to define a new loss function for the error model $\mathcal{N}_{\epsilon_1}$:

$$\mathcal{L}_{\epsilon_1} = \mathbb{E}_{x_1 \in \mathcal{D}}\Big[||\mathbf{F}_1[\mathcal{N}_{\epsilon_1}(x_1)]||^p\Big] + \mathbb{E}_{x_2 \in \partial\mathcal{D}}\Big[||\Phi_{\epsilon_1}(x_2) - \mathcal{N}_{\epsilon_1}(x_2)||^p\Big] \tag{8}$$

Alternatively, since the parameters of $\mathcal{N}$ are kept unchanged, we can use the equivalent form:

$$\mathcal{L}_{\epsilon_1} = \mathbb{E}_{x_1 \in \mathcal{D}}\Big[||\mathbf{F}[\mathcal{N}(x_1) + \mathcal{N}_{\epsilon_1}(x_1)]||^p\Big] + \mathbb{E}_{x_2 \in \partial\mathcal{D}}\Big[||\Phi(x_2) - [\mathcal{N}(x_2) + \mathcal{N}_{\epsilon_1}(x_2)]||^p\Big] \tag{9}$$

In problems with homogeneous constraint conditions, $\Phi(\partial\mathcal{D}) = 0$, we may replace the second term in Eq. 8 by $\mathbb{E}\Big[||\mathcal{N}_{\epsilon_1}(x_2) + \mathcal{N}(x_2)||^p\Big]$, since $\Phi_{\epsilon_1}(\partial\mathcal{D}) = 0 - \mathcal{N}(\partial\mathcal{D}) = -\mathcal{N}(\partial\mathcal{D})$.

Therefore, by the same reasoning underlying $\mathcal{L} = 0 \implies \mathcal{N} = \Phi$, we have that $\mathcal{L}_{\epsilon_1} = 0 \implies \mathcal{N}_{\epsilon_1} = \Phi_{\epsilon_1}$. Indeed, analogs of Theorems 1, 2, and Inequality 4 are obtained directly, as stated in a generalized form in the next subsection and proved in appendix A. The ability to internally estimate the error of an NN based solution model is not only useful for numerical analysis purposes, but also for correcting the original estimate. Algorithm 1 describes the proposed procedure.

Note that the difficulty of estimating Φ and Φ_{ϵ_1} should be roughly equivalent: since our activation functions are $\mathcal{C}^\infty$, Φ, Φ_{ϵ_1} have the same regularity. As such, the optimization costs of $\mathcal{N}_{\epsilon_1}$ per iteration will not need to increase substantially (see discussion and numerical experiments in the next sections).

Finally, note that Alg. 1 provides relative corrections to some model $\mathcal{N}$. We do not claim that only error correction can provide improvements in performance to the extent shown in Fig. 2. We only claim that error correction is a useful tool to analyse and refine *any* model $\mathcal{N}$ that might have been obtained via some technique based on NN DE solvers. It should **always** be possible to error correct $\mathcal{N}$ using the strategy we are describing, as long as the associated assumptions are satisfied.

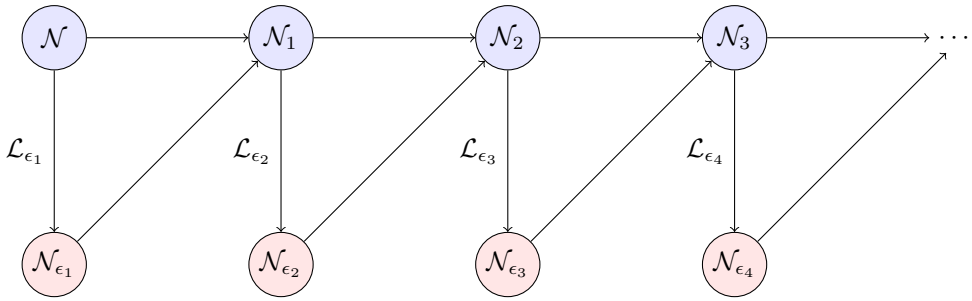


Figure 1: Graphical Model of the ith order error correction scheme. Each vertical pair of blue and red nodes denotes a specific *order* in the estimation and correction process.

4.2 HIGHER ORDER ERROR ESTIMATIONS AND CORRECTIONS

Consider some sequence of networks $\mathcal{N}_{\epsilon_1}, \mathcal{N}_{\epsilon_2}, ..., \mathcal{N}_{\epsilon_{i-1}}$, each modeling the errors $\Phi_{\epsilon_1}, \Phi_{\epsilon_2}, ..., \Phi_{\epsilon_i}$, associated with error corrected solution models $\mathcal{N}, \mathcal{N}_1, ..., \mathcal{N}_{i-1}$. Here, $\mathcal{N}_j = \mathcal{N} + \mathcal{N}_{\epsilon_1} + ... + \mathcal{N}_{\epsilon_j}$ is the jth order corrected model.

Assume the $i-1$ order error correction model $\mathcal{N}_{\epsilon_{i-1}}$ has saturated its capacity to model $\Phi_{\epsilon_{i-1}}$. We can obtain a higher order corrected model $\mathcal{N}_i$ by setting up a new network $\mathcal{N}_{\epsilon_i}$ to estimate Φ_{ϵ_i}. The new network $\mathcal{N}_{\epsilon_i}$ is optimized through the following loss function:

$$\begin{aligned} &\mathbf{F}_i[\mathcal{N}_{\epsilon_i}] := \mathbf{F}[\mathcal{N}_{\epsilon_i} + \mathcal{N}_{i-1}] = \mathbf{F}_{i-1}[\mathcal{N}_{i-1}] + \mathbf{L}[\mathcal{N}_{\epsilon_i}] + \mathbf{N}[\mathcal{N}_{i-1} + \mathcal{N}_{\epsilon_i}] - \mathbf{N}[\mathcal{N}_{i-1}] \\ &\mathcal{L}_{\epsilon_i} = \mathbb{E}_{x_1 \in \mathcal{D}}\left[||\mathbf{F}_i[\mathcal{N}_{\epsilon_i}(x_1)]||^p\right] + \mathbb{E}_{x_2 \in \partial\mathcal{D}}\left[||\Phi_{\epsilon_i}(x_2) - \mathcal{N}_{\epsilon_i}(x_2)||^p\right] \end{aligned} \tag{10}$$

Figure 1 provides a visual representation of the recursive process involved in building i^{th} order corrected models. Appendix A demonstrates how to arrive at these results.

We have direct analogs of Theorems 1 and 2, for the ith order errors and error models as:

Theorem 3. *Under the assumptions above on* $\mathbf{F}$ *the analogous result holds for* $\mathbf{F}_i$ *for all* $i \in \mathbb{N}$*, i.e. we can choose a neighbourhood* $U_i \subset G$ *of* 0 *small enough such that if* $\mathcal{N}_{\epsilon_i} \in U_i$ *then*

$$\mathbf{F}_i[\mathcal{N}_{\epsilon_i}] \longrightarrow 0 \implies \mathcal{N}_{\epsilon_i} \longrightarrow \Phi_{\epsilon_i}.$$

Furthermore,

$$\|\Phi_{\epsilon_i}\| = \mathcal{O}\left(\|\mathbf{F}_i[\mathcal{N}_i]\|\right).$$

Inequality 4 becomes (under the same additional structural assumptions on the system):

$$\|\Phi_{\epsilon_i}\| < \frac{F_{i_{\max}}}{H_{\min}} \tag{11}$$

Algorithm 2 in the appendix provides the pseudo-code to setup higher order corrections.

5 WHY ERROR CORRECTION WORKS AND HOW WE CAN MAKE IT BETTER

NN DE solvers transform the problem of estimating Φ into one of finding an appropriate $\mathcal{N}$ in some suitably constrained functional space. The use of gradient descent type methods turns this into a stochastic dynamical process in that space.

Given our assumption regarding uniqueness of Φ, the probability that $\mathcal{N}$ will end up exactly modeling Φ within a finite number of iterations is 0 (assuming Φ lies in an even moderately non-trivial space). Note that this point is true without even considering the fact that $\mathcal{N}$ is constrained to be a closed form model, whereas most non-trivial DE solutions are not. As such, $\mathcal{N}$ is doomed to languish in a local minima(s) of optimization, no matter how many resources we make available to it. Its capacity to approximate Φ is necessarily limited and once that capacity is saturated, further training of $\mathcal{N}$ is a waste of computational resources.

Error correction expands that capacity for approximating Φ, by introducing new trainable parameters (as components of $\mathcal{N}_{\epsilon_1}$) that exist exclusively for modeling $\Phi_{\epsilon_1} = \Phi - \mathcal{N}$. Since the parameters of the base network $\mathcal{N}$ are frozen, we have locked in the amount of performance already achieved on the base model. Further, since we assume error correction is being turned on when the base model's approximation capacity is effectively saturated (which is certain to happen), the gains we make with $\mathcal{N}_{\epsilon_1}$ are truly unachievable without it.

We have put forth a claim that error correction is no more involved or resource hungry than the standard NN DE solver method. Since we assume $\mathcal{C}^\infty$ functions are being used by the network as activations, Φ and $\Phi - \mathcal{N}$ are in the same functional space, have the same regularity, etc. Essentially, we meant that Φ_{ϵ_1} is no more *complicated* than Φ. However, it bears noting that modeling Φ_{ϵ_1} using $\mathcal{N}_{\epsilon_1}$ can be a very different kind of problem than modeling Φ using $\mathcal{N}$. For example, Φ_{ϵ_1} presumably lies in a different scale of magnitude than Φ and might present itself as a different kind of structure over $\mathcal{D}$ (for example, in one sense, it might be significantly more or less "oscillatory" than Φ).

This is where the knowledge of our DE (and the error correction DEs) comes into play. Theorem 1 allows us to estimate both the scale and complexity of Φ_{ϵ_1}, relative to $\mathcal{N}$. The correlation expected

between Φ_{ϵ_1} and $\mathbf{F}[\mathcal{N}]$ means even crude analysis of $\mathbf{F}[\mathcal{N}]$ is enough to point us in the correct direction, vis a vis the scale and complexity we expect $\mathcal{N}_{\epsilon_1}$ to model.

For example, we can bound the expected scale of Φ_{ϵ_1} by simply analysing the max of $||\mathbf{F}[\mathcal{N}]||$ over $\mathcal{D}$: the linear correlation result already gives us one way to weakly bound Φ_{ϵ_1}. The strong inequality 4 may allow for more precise control, when its assumptions are satisfied.

Similarly, over a domain $\mathcal{D}$, we could perform a FFT analysis of $\mathbf{F}[\mathcal{N}]$ and gauge how oscillatory $\mathcal{N}_{\epsilon_1}$ should be relative to $\mathcal{N}$ (since we have control over both $\mathcal{N}$ and $\mathbf{F}[\mathcal{N}]$ and we expect Φ_{ϵ_1} to be linearly correlated to $\mathbf{F}[\mathcal{N}]$). Other methods of estimating relative complexity of features should also be possible, since we have complete access to the behavior of $\mathbf{F}[\mathcal{N}]$.

The code we provide allows the user to supply their own estimates for relative scale and complexity of $\mathcal{N}_{\epsilon_1}$ wrt $\mathcal{N}$, while keeping our preferred defaults. We note again, that in no way is prior knowledge of Φ assumed: only the information already (known before) or (computed) using the NN DE solver is being used. Domain knowledge of an expert building an NN DE solver for a system of their interest is thus rewarded during error correction.

Finally, note that error correction inherits one major issue prevalent within machine learning: the ambiguity in determining if $\mathcal{N}$ has reached (or nearly reached) its capacity for effectively modelling Φ. The issue is caused because of lack of work on *a priori* or *in situ* convergence guarantees: it is extremely tough to gauge whether a loss trajectory has flattened due to saturation in modeling capacity or it is simply in a non-trivial region of dynamics from which it could escape at any time. If error correction is attempted while the base model itself is fully capable of providing significant increases in accuracy, the additional resources will simply be wasted (and may even prove to be counter-productive). Indeed, in our numerical experiments, we found that convergence of the base model $\mathcal{N}$ was a practical pre-requisite for significant gains via error correction.

However, in our proposed approach, the issue is still somewhat mitigated. Error correction can not cost significantly more than the standard methods (it can at worst be twice as expensive). Further, it invariably leads to very significant improvements in accuracy when used at the appropriate time. As such, it can always be used as an unsupervised, post-processing/refinement routine without significant risks.

6 Numerical Experiments

Let us exemplify the above discussion with an example DE that contains terms with linear, non-linear, and no dependence on the solution. In particular, we investigate a variant of the nonlinear Poisson-Boltzmann equation (nPBE), which serves as an important tool for the study of electrostatic interactions with widespread applications in biology and medicine (14). We will choose $\mathbf{L}[\,] = -\Delta[\,], \mathbf{N}[\,] \equiv \sinh[\,], \mathbf{C} \equiv g$, where g is some independent function/term. Thus, we define:

$$\mathbf{F}[\,] \equiv -\Delta[\,] + \sinh[\,] + g \tag{12}$$

For some solution attempt $\mathcal{N} : \mathbb{R}^d \to \mathbb{R}$ and associated error $\Phi_{\epsilon_1} = \Phi - \mathcal{N}$, we get

$$\mathbf{F}[\mathcal{N}] = \Delta[\Phi_{\epsilon_1}] + \sinh[\mathcal{N}] - \sinh[\mathcal{N} + \Phi_{\epsilon_1}] \tag{13}$$

Thus, replacing Φ_{ϵ_1} with an error model $\mathcal{N}_{\epsilon_1}$, we get the residual equation for the new loss as:

$$\mathbf{F}_1[\mathcal{N}_{\epsilon_1}] := \mathbf{F}[\mathcal{N} + \mathcal{N}_{\epsilon_1}] = \mathbf{F}[\mathcal{N}] - \Delta[\mathcal{N}_{\epsilon_1}] + \sinh[\mathcal{N}](\cosh[\mathcal{N}_{\epsilon_1}] - 1) + \cosh[\mathcal{N}]\sinh[\mathcal{N}_{\epsilon_1}] \tag{14}$$

Now, to setup $\mathcal{N}_{\epsilon_1}$ as the error NN DE solver (after halting the optimization of $\mathcal{N}$), we can use Eq. 14 to obtain an approximation of Φ_{ϵ_1}. We not only get an estimate for Φ_{ϵ_1}, we also get to use $\mathcal{N}_{\epsilon_1}$ as a correction term, while retaining the advantages that make $\mathcal{N}$ a lucrative option.

6.1 Error correction results

We choose $\mathcal{D} = [-\pi, \pi]^d \subset \mathbb{R}^d$, and randomly sample points from this domain as training data. The exact form of our chosen DE, considered over $\mathcal{D}$ with solution $\Phi : \mathcal{D} \to \mathbb{R}$ and homogeneous boundary conditions $\Phi(\partial\mathcal{D}) = 0$, is:

$$\mathbf{F}[\Phi] = -\Delta[\Phi] + \sinh[\Phi] + g = 0 \tag{15}$$

For our experiments, g has the following form:

$$g(\mathbf{x}) = -\omega^2 d\left[\sin(\omega x_1)...\sin(\omega x_d)\right] - \sinh(\sin(\omega x_1)...\sin(\omega x_d)) \qquad \mathbf{x} \equiv \{x_1, ..., x_d\} \in \mathcal{D}$$

The solution to this DE, which showcases significant scale and feature variation over the domain of interest (see Fig. 3), is given by:

$$\Phi(\mathbf{x}) = \sin(\omega x_1)...\sin(\omega x_d) \qquad \mathbf{x} \in \mathcal{D} \tag{16}$$

Note that the solvers never see this solution, and only have access to their own parameters and the operator $\mathbf{F}$ during optimization. We only use the exact solution to verify that our claims hold true.

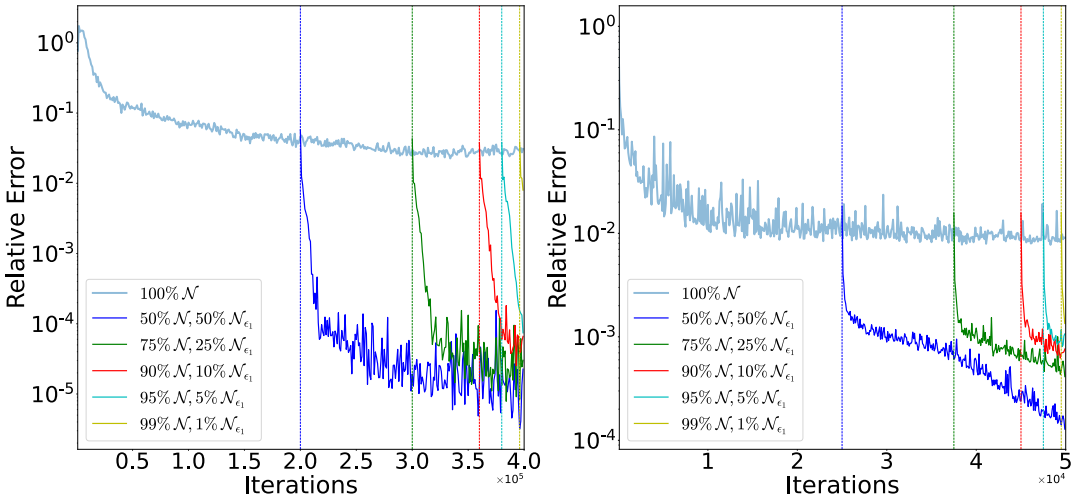
Given a network $\mathcal{N}$ to model the true solution Φ, we optimize $\mathcal{N}$ using the mean squared loss $\mathcal{L}$:

$$\mathcal{L}(\mathcal{N}) = \mathbb{E}_{x_1 \sim \mathcal{D}}\left[||-\Delta(\mathcal{N}(x_1)) + \sinh[\mathcal{N}(x_1)] + g(x_1)||^2\right] + \mathbb{E}_{x_2 \sim \partial\mathcal{D}}\left[||\mathcal{N}(x_2)||^2\right] \tag{17}$$

The error correction model $\mathcal{N}_{\epsilon_1} : \mathcal{D} \to \mathbb{R}$ is effectively optimized using the following loss:

$$\mathcal{L}_{\epsilon_1} = \mathbb{E}_{x_1 \sim \mathcal{D}}\left[||\mathbf{F}_1[\mathcal{N}_{\epsilon_1}(x_1)]||^2\right] + \mathbb{E}_{x_2 \sim \partial\mathcal{D}}\left[||\mathcal{N}(x_2) + \mathcal{N}_{\epsilon_1}(x_2)||^2\right] \tag{18}$$

As discussed before, $\mathcal{L} \to 0 \implies \mathbf{F}[\mathcal{N}(\mathbf{x})] \to 0$, over $\mathbf{x} \in \mathcal{D}$, which implies $\mathcal{N} \to \Phi$. Similarly, $\mathcal{L}_{\epsilon_1} \to 0 \implies \mathbf{F}_1[\mathcal{N}_{\epsilon_1}(\mathbf{x})] \to 0$, which implies $\mathcal{N}_{\epsilon_1} \to \Phi_{\epsilon_1}$. We use Eq. 16 to benchmark efficacy of our proposed algorithm for (ω, d) choices of $(5, 2)$ and $(1, 4)$. The goal of the experiments was to figure out what impact error correction would have, if used in lieu of the standard approach, while running for an equivalent number of training iterations. As such, choices on width, depth, activation functions, etc were kept the same between $\mathcal{N}$ and $\mathcal{N}_{\epsilon_1}$. The experiments also demonstrate the effects of invoking error correction at different stages of training (Fig. 2).



(a) 2 dimensions, $\omega = 5$ (b) 4 dimensions, $\omega = 1$

Figure 2: Relative errors for a single order correction on nPBE. Legend labels indicate duration of training $\mathcal{N}$, and duration of training $\mathcal{N}_{\epsilon_1}$ for correction respectively.

We define our Relative Error as:

$$\text{Relative Error} := \left(\sum_{k=1}^{K}[\Phi(\mathbf{x}_k) - \mathcal{N}_i(\mathbf{x}_k)]^2 \Big/ \sum_{k=1}^{K}[\Phi(\mathbf{x}_k)]^2\right)^{\frac{1}{2}}$$

where $\mathcal{N}_i : \mathcal{D} \to \mathbb{R}$ is the ith order corrected, NN DE based approximation for Φ over $\mathcal{D}$. K represents the number of sampled points in $\mathcal{D}$ for the calculation. We also define a computational cost metric τ, which represents the time taken per iteration by the standard and proposed method algorithm.

Table 1: **Performance comparison**

System	Method	Median τ (10^{-2} s)	Median Rel. Error (10^{-4})
nPBE (5, 2)	Standard	8.60	436
nPBE (5, 2)	Alg. #1	11.8	0.37
nPBE (1, 4)	Standard	72.1	87.3
nPBE (1, 4)	Alg. #1	89.8	1.87
NL Osc	Standard	1.39	1.73
NL Osc	Alg. #1	1.68	0.02
HH	Standard	1.37	3.08
HH	Alg. #1	1.63	0.18

In Fig. 2, we plot the relative error dynamics of the various choices made. Notice the very significant and immediate gains produced in accuracy by error correction.

All numerical experiments were done using a 2019 MacBook Pro with a 2.6 GHz 6-Core Intel Core i7 processor and 16 GBs of 2667 MHz DDR4 RAM.

The associated codebase with this work provides numerical experiments on some other nonlinear/chaotic DEs (it is designed to be compatible with any DE operator with permissible Frechet derivatives and allow any finite order of correction). The results on those systems are also tabulated in Table 1 (first order error correction was injected at 50% of the run-time, for all other examples). All results are in line with our expectations. The additional systems are described in appendix D.

7 Conclusions

The surging popularity of NN DE solvers presents exciting possibilities in many scientific fields: their capacity to sidestep the curse of dimensionality (15), general advances in computing/GPU power, and their easy interpret-ability make for a powerful combo. As such, it is important that the solution models they provide be capable of validation over domains where true solutions are not available. Our work proposes one class of methods by which this current deficiency in many types of NN DE solvers may be handled.

Notice that our proposed method does not fully solve the problem: we have somewhat transferred the ambiguity associated with the estimate $\mathcal{N}$ into the ambiguity associated with the estimate $\mathcal{N}_{\epsilon_i}$. However, this is still substantially better than the situation that existed before: even an ambiguous estimate of Φ_{ϵ_1} has clear practical benefits. More importantly, Theorem 1 and Inequality 4 allow us to obtain weak/strong bounds on $||\Phi_{\epsilon_1}||$ without the need to construct $\mathcal{N}_{\epsilon_1}$ (and Theorem 3 and Inequality 11 extend it to any order desired). Further, we can add useful corrections to any order under the assumed conditions as well. Whenever their assumptions are satisfied, our theorems and inequalities do fully resolve the problems we made the central questions of this work. Future work will focus on proving similar results under weaker assumptions, so that even larger classes of DEs may come within the remit of error correction.

Lastly, we note that the general ideas we are suggesting are not radically different than those that already exist for classical numerical methods: higher order corrections are about as old as the field itself. By pairing the many significant advantages of modern NNs with those old ideas, we simply hope to have presented a blueprint that will be useful for a wide class of scientific problems that are being tackled using NN DE solvers.

References

[1] I. E. Lagaris, A. Likas, and D. I. Fotiadis. Artificial neural networks for solving ordinary and partial differential equations. *IEEE Transactions on Neural Networks and Learning Systems*, 9:987–1000, 1998.

[2] 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*, 378:686–707, 2019.

[3] J. Sirignano and K. Spiliopoulos. Dgm: A deep learning algorithm for solving partial differential equations. *Journal of Computational Physics*, 375:1339–1364, 2018.

[4] J. Han, A. Jentzen, and Weinan E. Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences*, 2018.

[5] Z. Lu, H. Pu, F. Wang, Z. Hu, and L. Wang. The expressive power of neural networks: A view from the width. *Advances in Neural Information Processing Systems (NIPS)*, 30, 2017.

[6] R. Arora, A. Basu, P. Mianjy, and A. Mukherjee. Understanding deep neural networks with rectified linear units. *arXiv:1611.01491*, 2018.

[7] B. Hanin and M. Sellke. Approximating continuous functions by relu nets of minimal width. *arXiv:1710.11278*, 2018.

[8] D. Yarotsky. Error bounds for approximations with deep relu networks. *Neural Networks*, 94:103–114, 2017.

[9] Simon S. Du, Jason D. Lee, Haochuan Li, Liwei Wang 0001, and Xiyu Zhai. Gradient Descent Finds Global Minima of Deep Neural Networks. In *Proceedings of the 36th International Conference on Machine Learning*, pages 1675–1685. PMLR, 2019.

[10] M. Mattheakis, D. Sondak, A. S. Dogra, and P. Protopapas. Hamiltonian neural networks for solving differential equations. *Physical Review E 105, 065305*, 2022.

[11] Yeonjong Shin, Jérôme Darbon, and George Em Karniadakis. On the convergence of physics informed neural networks for linear second-order elliptic and parabolic type pdes. *Communications in Computational Physics*, 28(5):2042–2074, 2020.

[12] Philipp Grohs, Fabian Hornung, Arnulf Jentzen, and Philipp Zimmermann. Space-time error estimates for deep neural network approximations for differential equations. *arXiv:1908.03833*, 2019.

[13] Fabian Hornung, Arnulf Jentzen, and Diyora Salimova. Space-time deep neural network approximations for high-dimensional partial differential equations. *arXiv:2006.02199*, 2020.

[14] José A Iglesias and Svetoslav Nakov. Weak formulations of the nonlinear poisson-boltzmann equation in biomolecular electrostatics. *Journal of Mathematical Analysis and Applications*, page 126065, 2022.

[15] P. Grohs, Fabian Hornung, Arnulf Jentzen, and Philippe Von Wurstemberger. A proof that artificial neural networks overcome the curse of dimensionality in the numerical approximation of black-scholes partial differential equations. *Mem. Amer. Math. Soc. (to appear)*, 2018.

[16] Serge Lang. *Differential and Riemannian Manifolds*. Graduate texts in mathematics, 160. ; 160. Springer, New York, 1995.

A SETTING UP HIGHER ORDER CORRECTIONS

Using Eq. equation 7, we are able to define new loss $\mathcal{L}_{\epsilon_1}$ using $\mathbf{F}_1[\mathcal{N}_{\epsilon_1}]$ (or equivalently, $\mathbf{F}[\mathcal{N}+\mathcal{N}_{\epsilon_1}]$). Define $\Phi_{\epsilon_2} := \Phi_{\epsilon_1} - \mathcal{N}_{\epsilon_1} = \Phi - (\mathcal{N} + \mathcal{N}_{\epsilon_1})$, and use the same trick used to derive Eq. 5 to realize that,

$$\begin{aligned}
\mathbf{F}_1[\mathcal{N}+\mathcal{N}_{\epsilon_1}] &= \mathbf{F}[\mathcal{N}] + \mathbf{L}[\Phi_{\epsilon_1} - \Phi_{\epsilon_2}] + \mathbf{N}[\mathcal{N}+\mathcal{N}_{\epsilon_1}] - \mathbf{N}[\mathcal{N}] + (\mathbf{N}[\mathcal{N}+\Phi_{\epsilon_1}] - \mathbf{N}[\mathcal{N}+\Phi_{\epsilon_1}]) \\
&= -\mathbf{L}[\Phi_{\epsilon_2}] + \mathbf{N}[\mathcal{N}+\mathcal{N}_{\epsilon_1}] - \mathbf{N}[\mathcal{N}+\Phi_{\epsilon_1}] \\
&= -\mathbf{L}[\Phi_{\epsilon_2}] + \mathbf{N}[\mathcal{N}+\mathcal{N}_{\epsilon_1}] - \mathbf{N}[\mathcal{N}+\mathcal{N}_{\epsilon_1}+\Phi_{\epsilon_2}] \qquad (19)
\end{aligned}$$

Let us define the 1st order corrected model as $\mathcal{N}_1 := \mathcal{N} + \mathcal{N}_{\epsilon_1}$, and with a slight abuse of notation, $\mathbf{F}_1[\mathcal{N}_1] := \mathbf{F}[\mathcal{N} + \mathcal{N}_{\epsilon_1}]$. Thus, we may rewrite Eq. 19 as:

$$\mathbf{F}_1[\mathcal{N}_1] + \mathbf{L}[\Phi_{\epsilon_2}] + \mathbf{N}[\mathcal{N}_1 + \Phi_{\epsilon_2}] - \mathbf{N}[\mathcal{N}_1] = 0$$

Note the functional similarity to Eq. 6. Further, to setup a 2nd order correction estimate $\mathcal{N}_{\epsilon_2}$ for Φ_{ϵ_2}, we need to minimize over the following residual, similar to Eq. 7:

$$\mathbf{F}_2[\mathcal{N}_{\epsilon_2}] = \mathbf{F}_1[\mathcal{N}_1] + \mathbf{L}[\mathcal{N}_{\epsilon_2}] + \mathbf{N}[\mathcal{N}_1 + \mathcal{N}_{\epsilon_2}] - \mathbf{N}[\mathcal{N}_1]$$

In general, given an $(i-1)^{\text{th}}$ order corrected model $\mathcal{N}_{i-1}$ using $\mathcal{N}_{i-1} = \mathcal{N} + \mathcal{N}_{\epsilon_1} + \mathcal{N}_{\epsilon_2} + ... + \mathcal{N}_{\epsilon_{i-1}}$, we can define $\Phi_{\epsilon_i} := \Phi - \mathcal{N}_{i-1}$ and $\mathbf{F}_{i-1}[\mathcal{N}_{i-1}]$ as the corresponding residual for the preceding equations. Then, Φ_{ϵ_i} is the unique solution to the following equation

$$\mathbf{F}_{i-1}[\mathcal{N}_{i-1}] + \mathbf{L}[\Phi_{\epsilon_i}] + \mathbf{N}[\mathcal{N}_{i-1} + \Phi_{\epsilon_i}] - \mathbf{N}[\mathcal{N}_{i-1}] = 0 \tag{20}$$

The error can be estimated using a differential equation solver (e.g. a NN) with residual:

$$\mathbf{F}_i[\mathcal{N}_{\epsilon_i}] := \mathbf{F}[\mathcal{N}_{\epsilon_i} + \mathcal{N}_{i-1}] = \mathbf{F}_{i-1}[\mathcal{N}_{i-1}] + \mathbf{L}[\mathcal{N}_{\epsilon_i}] + \mathbf{N}[\mathcal{N}_{i-1} + \mathcal{N}_{\epsilon_i}] - \mathbf{N}[\mathcal{N}_{i-1}] \tag{21}$$

Figure 1 depicts the extension of the error estimation and correction framework as a graphical model. Notice that the initial solver estimator and the error estimators may be pre-fetched before carrying out the optimization steps.

Since $\Phi_{\epsilon_i} = \Phi - \mathcal{N}_{i-1}$ is always known exactly over $\partial\mathcal{D}$ for all i, we can optimize the ith order error model $\mathcal{N}_{\epsilon_i}$ using $\mathcal{L}_{\epsilon_i}$:

$$\mathcal{L}_{\epsilon_i} = \mathbb{E}_{x_1 \sim \mathcal{D}}\left[||\mathbf{F}_i[\mathcal{N}_{\epsilon_i}(x_1)]||^p\right] + \mathbb{E}_{x_2 \sim \partial\mathcal{D}}\left[||\Phi_{\epsilon_i}(x_2) - \mathcal{N}_{\epsilon_i}(x_2)||^p\right] \tag{22}$$

Algorithm 2 demonstrates the procedure for higher order estimation and correction. The uniqueness of Φ as a solution to $\mathbf{F}[\,]$ also implies the uniqueness of Φ_{ϵ_i} as a solution to $\mathbf{F}_i[\,]$. Unpacking the definition of $\mathbf{F}_i$ and $\mathcal{N}_{i-1}$ we find that

$$\mathbf{F}_i[\Phi_{\epsilon_i} + \mathcal{N}_{i-1}] = \mathbf{F}[\Phi - \mathcal{N}_{i-1} + \mathcal{N}_{i-1}] = \mathbf{F}[\Phi] = 0$$

and therefore the solution to this equation is still unique and convergence of $\mathcal{L}_{\epsilon_i} \to 0$ implies $\mathbf{F}_i \to 0$, and thus also $\mathcal{N}_{\epsilon_i} \to \Phi_{\epsilon_i}$. Theorem 3 follows naturally from Theorems 1 and 2. Inequality 11 follows naturally from Inequality 4.

Algorithm 2 describes the proposed implementation.

Algorithm 2 Internal Error Estimation and Correction to Order m

procedure ERRORCORRECT(m)
  initialize NN DE solver $\mathcal{N} : \mathcal{D} \to \mathbb{R}$
  initialize Error estimators $\{\mathcal{N}_{\epsilon_i}\}_{i=1}^m$
  Train $\mathcal{N}$, and define $\hat{\mathcal{N}} := \mathcal{N}$ ▷ $\hat{\mathcal{N}}$ serves as a dummy variable
  for $i = 1, \ldots, m$ **do**
    Train $\mathcal{N}_{\epsilon_i}$ until loss converges
    Save parameter states of $\mathcal{N}_{\epsilon_i}$
    $\hat{\mathcal{N}} := \hat{\mathcal{N}} + \mathcal{N}_{\epsilon_i}$
    Freeze parameters of $\hat{\mathcal{N}}$
  end for
  return $\mathcal{N} + \sum_{i=1}^m \mathcal{N}_{\epsilon_i}$ ▷ Equivalently, return $\hat{\mathcal{N}}$
end procedure

B THEOREMS ON THE RELATIONSHIP BETWEEN ERROR AND LOSS

We quickly summarize the Banach space version of the inverse function theorem that allows us to establish that $\mathbf{F}[\mathcal{N}] \to 0$ implies $\mathcal{N} \to \Phi$. Let G and H be two Banach spaces, and $U \subset G$ an open subset. A continuous map between the two Banach spaces $\mathbf{F} : G \supset U \to H$ is said to be Fréchet differentiable at a point $x \in U$ iff there exists a linear bounded operator $L_x : G \to H$ such that

$$\|\mathbf{F}(y) - \mathbf{F}(x) - L_x(y - x)\|_F = o(\|y - x\|_E).$$

If the map $x \mapsto \mathrm{D}\mathbf{F}(x) = L_x$ is continuous, then one says that $\mathbf{F} \in \mathscr{C}^1(U)$.

Lemma 1. *Let $\mathbf{F} : U \to H$ be a $\mathscr{C}^1$-map. Suppose that there exists a point $x_0 \in U$ such that $\mathrm{D}\mathbf{F}(x_0)$ is an isomorphism (i.e. it has a continuous inverse). Then there exists a neighbourhood $V \subset H$ of $\mathbf{F}(x_0)$ and a $\mathscr{C}^1$ function $\mathbf{F}^{-1} : V \to G$ that is a local inverse of $\mathbf{F}$.*

Proof. See Theorem 5.2 of (16). □

Using this result, we derive all the theorems presented in this work.

Theorem 1. *Suppose that $\mathbf{F} : G \supset U \to H$ is Fréchet differentiable in U, that the derivative of $\mathbf{F}$ at $\Phi \in U$ is invertible, and $\mathbf{F}[\Phi] = 0$. There is a neighbourhood $V \subset H$ of 0 small enough such that*

$$\mathbf{F}[\mathcal{N}] \longrightarrow 0 \implies \mathcal{N} \longrightarrow \Phi.$$

Proof. By Lemma 1, we can choose neighbourhoods $\Phi \in U' \subset U$ and $0 \in V \subset H$ such that $\mathbf{F} : U' \to V$ is a diffeomorphism. Then, if $\mathcal{N} \in U'$ the continuity of $\mathbf{F}^{-1}$ implies

$$\mathbf{F}[\mathcal{N}] \longrightarrow 0 \implies \mathcal{N} \longrightarrow \Phi.$$

which is the assertion. □

Theorem 2. *Under the same assumptions as above we*

$$\|\mathcal{N} - \Phi\| = \mathcal{O}\left(\|\mathbf{F}[\mathcal{N}]\|\right).$$

Proof. Since by Lemma 1, $\mathbf{F}^{-1}$ is differentiable at 0, it follows that it is locally Lipschitz continuous around 0, implying that

$$\|\mathcal{N} - \Phi\| = \mathcal{O}\left(\|\mathbf{F}[\mathcal{N}]\|\right).$$

□

Theorem 3. *Under the assumptions above on $\mathbf{F}$ the analogous result holds for all $\mathbf{F}_i$, i.e. we can choose a neighbourhood $V_i \subset H$ of 0 and $U_i \subset G$ small enough such that if $\mathcal{N}_{\epsilon_i} \in U_i$ then*

$$\mathbf{F}_i[\mathcal{N}_{\epsilon_i}] \longrightarrow 0 \implies \mathcal{N}_{\epsilon_i} \longrightarrow \Phi_{\epsilon_i}.$$

Furthermore,

$$\|\Phi_{\epsilon_i}\| = \mathcal{O}\left(\|\mathbf{F}_i[\mathcal{N}_{\epsilon_i}]\|\right).$$

Proof. This follows directly from the definition of $\mathbf{F}_i, \mathcal{N}_{i-1}$, and Φ_{ϵ_i}. □

The constant of proportionality is approximately given by $\lambda_{\min}^{-1}$ where

$$\lambda_{\min} := \inf_{\lambda \in \mathrm{Spec}(\mathrm{D}\mathbf{F}[\Phi])} |\lambda|$$

the eigenvalue with minimal absolute value in the spectrum of $\mathrm{D}\mathbf{F}[\Phi]$. The example from section 6 fits nicely into this paradigm as the map

$$\begin{aligned} \mathbf{F} : W^{2,\infty}(\mathbb{R}^3) &\longrightarrow W^{0,\infty}(\mathbb{R}^3) \\ f &\longmapsto -\Delta f + \sinh(f) + g \end{aligned}$$

is continuous and continuously Fréchet differentiable with derivative

$$\mathrm{D}\mathbf{F}(f) = -\Delta + \cosh(f)$$

which is everywhere continuously invertible as a linear map. The constant of proportionality is approximately 1.

C Strict Upper Bounds on $\|\Phi_{\epsilon_1}\|$

We will begin our attempt to ascertain bounds on $||\Phi_{\epsilon_1}||$ on classical Hamiltonian systems first, to exemplify the intuition and the technique involved in the generic bound we wish to develop. Once that is achieved, we will show how a generalization follows in a straightforward manner.

C.1 HAMILTONIAN SYSTEMS

To start, let us remember $\Phi_{\epsilon_1} := \mathcal{N} - \Phi = (N_1 - \phi_1, \ldots, N_D - \phi_D)$ for a D-dimensional dynamical system to be solved on a domain of interest $[0, T]$. Note that D is necessarily even for a Hamiltonian dynamical system. Let us construct a worst case scenario for the norm of Φ_{ϵ_1}. We write the equation in terms of the Hamiltonian formulation:

$$\frac{d\Phi}{dt} = J\nabla\mathcal{H}(\Phi) \tag{23}$$

where $\mathcal{H}$ is the appropriate Hamiltonian, and J is the symplectic matrix

$$J = \begin{pmatrix} 0 & -I_{D/2} \\ I_{D/2} & 0 \end{pmatrix} \tag{24}$$

and $I_{D/2}$ is the identity matrix. The NN DE solver is trained using $\mathbf{F}[\mathcal{N}]$ given by

$$\mathbf{F}[\mathcal{N}] = J\nabla\mathcal{H}(\mathcal{N}) - \frac{d\mathcal{N}}{dt} \tag{25}$$

The equation above represents D separate differential equations in a vector form. Since $\mathbf{F}[\Phi](t) = 0$ for all t we can write $\mathbf{F}[\mathcal{N}]$ as follows, suppressing the time dependence

$$\mathbf{F}[\mathcal{N}] = -\int_0^1 \mathrm{D}\mathbf{F}[\mathcal{N}_s]\Phi_{\epsilon_1} ds = \dot{\Phi}_{\epsilon_1} - \left[\int_0^1 J\left(\mathrm{D}^2\mathcal{H}\right)(\mathcal{N}_s)ds\right] \cdot \Phi_{\epsilon_1} =: \dot{\Phi}_{\epsilon_1} - R\Phi_{\epsilon_1} \tag{26}$$

where $\mathcal{N}_s := \Phi - s\Phi_{\epsilon_1}$, and $\mathrm{D}^2\mathcal{H}(\Phi)$ is the Hessian matrix of $\mathcal{H}$ with components $\partial_i\partial_j\mathcal{H}(\mathcal{N})$. In order to be able to extract a meaningful error bound from Eq. equation 26 we need to make some structural assumption: We assume that at a time $t_{\Phi_{\epsilon_1}} \in [0, T]$ at which $||\Phi_{\epsilon_1}||$ obtains its maximum

$$\left[R(t_{\Phi_{\epsilon_1}})\Phi_{\epsilon_1}(t_{\Phi_{\epsilon_1}})\right] \cdot \dot{\Phi}_{\epsilon_1}(t_{\Phi_{\epsilon_1}}) = 0$$

Note that the assumption of focusing all error in only one component of Φ, as done in (10), already implies the above, but not vice versa. As such, we are using a much weaker assumption than (10). [3] Then, for time $t_{\Phi_{\epsilon_1}} \in [0, T]$, we have

$$||R\Phi_{\epsilon_1}||_2^2 \leq ||R\Phi_{\epsilon_1}||_2^2 + ||\dot{\Phi}_{\epsilon_1}||_2^2 = ||\mathbf{F}[\mathcal{N}]||_2^2 \tag{27}$$

We next need an estimate on the matrix norm of the inverse of R for every $t \in [0, T]$. We can ignore J as it is an orthogonal matrix and only consider

$$\int_0^1 \left(\mathrm{D}^2\mathcal{H}\right)(\mathcal{N}_s)ds \ .$$

We can do this if we know that $\mathrm{D}^2\mathcal{H}(x)$ is a strictly positive or strictly negative definite matrix for all $x \in \mathcal{D} \subset \mathbb{R}^D$ in our domain of interest, which is a standard assumption for the motion to be non-degenerate. W.l.o.g., assuming that $\mathrm{D}^2\mathcal{H}$ is positive definite, and setting

$$H_{\min} := \min_{x\in\mathcal{D}} \|\mathrm{D}^2\mathcal{H}(x)^{-1}\|^{-1} = \min_{x\in\mathcal{D}} \lambda_{\min}(\mathrm{D}^2\mathcal{H}(x))$$

where $\lambda_{\min}$ is the smallest eigenvalue of the matrix $\mathrm{D}^2\mathcal{H}(x)$, we have for all $t \in [0, T]$

$$\|R^{-1}\| \leq H_{\min}^{-1} \ .$$

Finally, letting

$$F_{\max} := \max_{t\in[0,T]} ||\mathbf{F}[\mathcal{N}](t)||_2$$

we reach the an estimate on $||\Phi_{\epsilon_1}||$ as:

$$||\Phi_{\epsilon_1}|| \leq \frac{F_{\max}}{H_{\min}} \tag{28}$$

Additionally, if we assume that at the time $t_{\dot{\Phi}_{\epsilon_1}}$ at which $|\dot{\Phi}_{\epsilon_1}|$ reaches its maxima,

$$\left[R(t_{\dot{\Phi}_{\epsilon_1}})\Phi_{\epsilon_1}(t_{\dot{\Phi}_{\epsilon_1}})\right] \cdot \dot{\Phi}_{\epsilon_1}(t_{\dot{\Phi}_{\epsilon_1}}) = 0$$

we have the following bound on $\dot{\Phi}_{\epsilon_1}$ as well

$$||\dot{\Phi}_{\epsilon_1}|| \leq F_{\max} \tag{29}$$

[3] Unfortunately, we need this assumption to separate the two terms appearing on the right hand side of Eq. equation 26.

C.2 BOUNDS ON $||\Phi_{\epsilon_1}||$ IN MORE GENERAL SETTINGS

Eq. 26 generalizes beyond Hamiltonian systems in a straightforward manner for $\mathbf{F}[\,] \equiv \mathbf{L}[\,]+\mathbf{N}[\,]+\mathbf{C}$ as:

$$\mathbf{F}[\mathcal{N}] = -\int_0^1 \mathrm{D}\mathbf{F}[\mathcal{N}_s]\Phi_{\epsilon_1} ds = \left[\int_0^1 \mathrm{D}\mathbf{N}[\mathcal{N}_s]ds\right] \cdot \Phi_{\epsilon_1} - \mathbf{L}[\Phi_{\epsilon_1}] =: -R\Phi_{\epsilon_1} - \mathbf{L}[\Phi_{\epsilon_1}] \tag{30}$$

The generalized version of the $R\Phi_{\epsilon_1} \cdot \dot{\Phi}_\epsilon = 0$ assumption is that at the $\mathbf{x}_{\Phi_{\epsilon_1}} \in \mathcal{D}$ that $||\Phi_{\epsilon_1}||$ takes its maxima in, we have:

$$\left[R(\mathbf{x}_{\Phi_{\epsilon_1}})\Phi_{\epsilon_1}(\mathbf{x}_{\Phi_{\epsilon_1}})\right] \cdot \mathbf{L}[\Phi_{\epsilon_1}](\mathbf{x}_{\Phi_{\epsilon_1}}) = 0 \qquad \mathbf{x}_{\Phi_{\epsilon_1}} \in \mathcal{D}$$

The next assumption we generalize is the one made to have non-degeneracy in the solutions we were trying to model. As such, we assume that $\mathrm{D}\mathbf{N}$ is a positive definite (or negative definitive) operator. From then on, we define $H_{\min}$ in a similar manner. More precisely, we say:

$$H_{\min} := \inf_{s\in[0,1]} \inf\left\{|\lambda| \,\middle|\, \lambda \in \text{Spec}\left(\mathrm{D}\mathbf{N}[\mathcal{N}_s]\right)\right\}$$

With that in place, we obtain the following inequalities for $||\Phi_{\epsilon_1}||$:

$$||\Phi_{\epsilon_1}|| \leq \frac{F_{\max}}{H_{\min}} \tag{31}$$

The generalized version of the assumption on $\mathbf{x}_{\dot{\Phi}_\epsilon}$ where $|\dot{\Phi}_\epsilon|$ attains its maxima similarly gives:

$$||\dot{\Phi}_\epsilon|| \leq F_{\max} \tag{32}$$

The assumption that $\left[R(\mathbf{x}_{\Phi_{\epsilon_1}})\Phi_{\epsilon_1}(\mathbf{x}_{\Phi_{\epsilon_1}})\right] \cdot \mathbf{L}[\Phi_{\epsilon_1}](\mathbf{x}_{\Phi_{\epsilon_1}}) = 0$ is critical in allowing us strong estimates on $||\Phi_{\epsilon_1}||$. Superficially, it might seem like too strong of an assumption (and even somewhat of a non-sequitur).

However, note that $\mathbf{L}[\,]$ in a DE is often the domain derivative term (we denote it as $\nabla_{\mathbf{x}}$ in this discussion, where $\mathbf{x} \in \mathcal{D}$ represents an arbitrary element in the domain. As such, $\nabla_{\mathbf{x}}$ are the spatial derivatives for spatial DEs, time derivatives for ODEs, a spatio-temporal one for space-time DEs, etc). When $||\Phi_{\epsilon_1}||$ achieves its maxima, $\Phi_{\epsilon_1} \cdot \nabla_{\mathbf{x}}(\Phi_{\epsilon_1}) = 0$. When $\mathbf{L} \equiv \nabla_{\mathbf{x}}$, we indeed have $\Phi_{\epsilon_1} \cdot \mathbf{L}(\Phi_{\epsilon_1}) = 0$. As such, what we are really assuming is that the R term effectively has Φ_{ϵ_1} as its eigenvector on $\mathbf{x}_{\Phi_{\epsilon_1}}$ (since that means $R\Phi_{\epsilon_1} \cdot \nabla_{\mathbf{x}}(\Phi_{\epsilon_1})$ is 0). For example, such is the case for Hamiltonian systems without mixed position-momentum terms. What we are really constraining, is the behavior exhibited by the operator $\mathbf{N}$ in a system of interest.

If the problem of interest comes from a scientific domain with which the user has familiarity, they may supply a different sort of assumption based on those considerations. For example, we might be able to say:

$$||R\Phi_{\epsilon_1} \cdot \nabla_{\mathbf{x}}\Phi_{\epsilon_1}|| < \frac{\epsilon}{2}$$

at the $\mathbf{x}_{\Phi_{\epsilon_1}}$ where $||\Phi_{\epsilon_1}||$ takes its maxima. The generalized bound simply changes to:

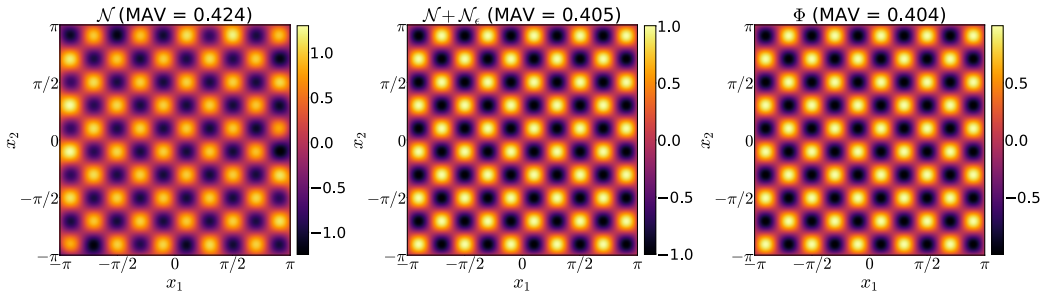
$$||\Phi_{\epsilon_1}|| \leq \frac{F_{\max} + \epsilon}{H_{\min}} \tag{33}$$

D ADDITIONAL NUMERICAL EXPERIMENTS

As part of validating our claims, we also performed numerical experiments on some other non-trivial DEs: quartic nonlinear oscillator and Henon-Heiles. We chose our experiments such that the assorted set of examples (when including nPBE) represents systems that showcase non-trivial chaotic, spatial, and temporal phenomena amongst the combined examples. Table 1 presents the median results from the collection of randomized experiments conducted for each system. We describe the chosen systems below:

D.1 NONLINEAR POISSON BOLTZMANN EQUATION

We have already given details of the equation in the main paper, alongside the associated choices of dimensionality and frequency parameter. Below, we plot what the $(\omega, d) = (5, 2)$ system should look like when modeled without (a) and with (b) error correction, alongside the true solution (c). To quantify the visual, we also quote the mean average value of $||\mathcal{N}||, ||\mathcal{N} + \mathcal{N}_{\epsilon_1}||, ||\Phi||$ respectively.



(a) $\mathcal{N}$ (b) $\mathcal{N} + \mathcal{N}_{\epsilon_1}$ (c) Φ

Figure 3: Heatmaps of non-error corrected $\mathcal{N}$ & error corrected solver with $\mathcal{N}_{\epsilon_1}$ at 50% of the total training iterations (left & middle), and Φ (right) for the $(\omega, d) = (5, 2)$ setting.

$\mathcal{N}$ and $\mathcal{N}_{\epsilon_1}$ were fully connected Neural Networks, with 4 hidden layers, each with 50 sine activation functions. The base model was trained for 400000 iterations, while error models were trained for a varying number of iterations, depending on when the error correction models were activated (see Fig. 2). We sampled 1024 points per iteration.

The $(\omega, d) = (1, 4)$ model was trained for 50000 iterations, with 8192 points sampled per iteration.

D.2 HENON HEILES

Henon Heiles is represented by the following ODE governing the dynamics of $\Phi \equiv \{x, y, p_x, p_y\}^\dagger$:

$$\dot{\Phi} = -J\nabla\mathcal{H}, \quad J = \begin{pmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{I} & \mathbf{0} \end{pmatrix}, \quad \mathcal{H} \equiv \frac{x^2 + y^2 + p_x^2 + p_y^2}{2} + \frac{y(3x^2 - y^2)}{3} \tag{34}$$

where $\mathbf{I}$ is the 2×2 identity matrix. We picked $[0, 6\pi]$ as the time domain of interest. All $\Phi(0) \equiv \{x(0), y(0), p_x(0), p_y(0)\}$ were picked s.t. $\mathcal{H}(\mathbf{z}(0)) < \frac{1}{6}$, $x(0) \in [-\frac{\sqrt{3}}{2}, \frac{\sqrt{3}}{2}]$, and $y(0) \in [-0.5, 1 - \sqrt{3}|x|]$, which corresponds to the subset of the phase space that lead to bounded orbits.

The models each had 2 hidden layers, with 50 sine activation functions per layer. The base models were trained for 50000 iterations, with an error correction made at 25000 iterations. We sampled 200 time points were iteration.

D.3 NONLINEAR QUARTIC OSCILLATOR

The nonlinear quartic oscillator is represented by the following ODE governing the dynamics of $\Phi \equiv \{x, p_x\}$:

$$\dot{\Phi} = -J\nabla\mathcal{H}, \quad J = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, \quad \mathcal{H} \equiv \frac{x^2 + p_x^2}{2} + \frac{x^4}{4} \tag{35}$$

We sample $\Phi(0)$ such that $\{x(0), p_x(0)\} \in [-1, 1] \times [-1, 1]$ each time.

The models had 2 hidden layers each, with 50 sine activation functions per layer. The base models were trained for 50000 iterations, with an error correction made at 25000 iterations. We sampled 200 time points were iteration.

The associated codebase allows the user to add other systems as per their choice.