Machine Learning Gravity

General Relativity Informed Neural Networks with deepxde

Alexander Dreichner[1], **Akash Kav**[2], **Zena Khadour**[3], **Tanner Nelson**[2]

Mentors: **Dr. Saeed Rastgoo**[2], **Dr. Jim Mertens**[4]

Junior Mentor: **Erik Weiss**[3]

[1]Goethe University, Frankfurt

[2]University of Alberta, Edmonton

[3]York University, Toronto

[4]Case Western Reserve University, Cleveland

Agenda

- Introduction
  - Project Overview
  - Neural Networks (NN)
  - Physics-Informed Neural Networks (PINNs)
  - Effective One Body Problem (EOB)
- Implementation & Results
  - Coding physics: classical solver solve_ivp and NN
- Future Directions
  - Discussion and further goals

Introduction

Project Overview

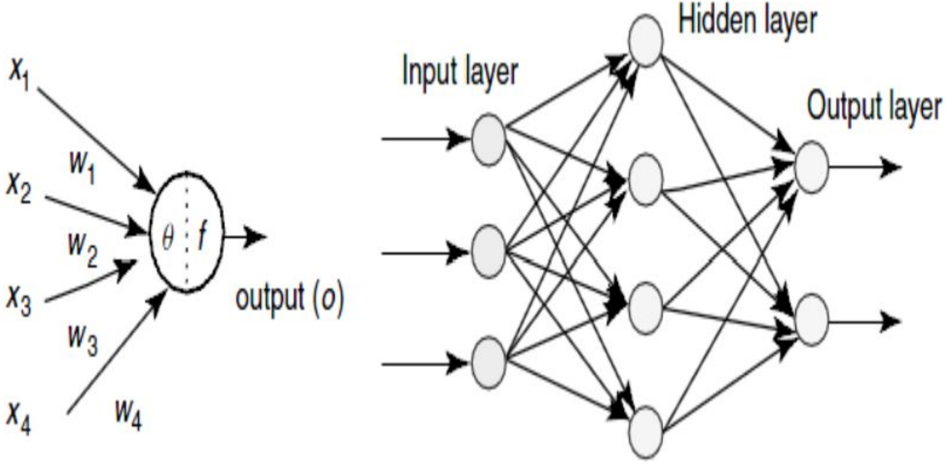
- Goals
  - Machine learning algorithms (NN) to solve EOB
  - Expand solution with higher order corrections from GR
  - Experience with deepxde Python library
  - Compare performance between classical method and neural network

Project Overview

- What methods did we use?
  - PINN with Python library deepxde.
  - Implement EOB system with two Post-Newtonian (PN) terms.
  - Use numerical (Runge-Kutta) solution from solve_ivp as reference.
  - Crucial question: how do the accuracy and time compare?

What are Neural Networks?

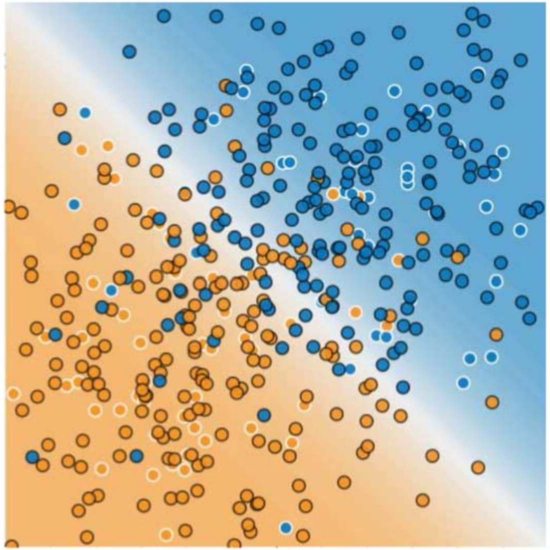
- Algorithm: network of neurons
- Neurons take in vectors and output scalars, which can be fed into other neurons
- Outputs “propagate” until final output is given
- Usually trained to optimize performance



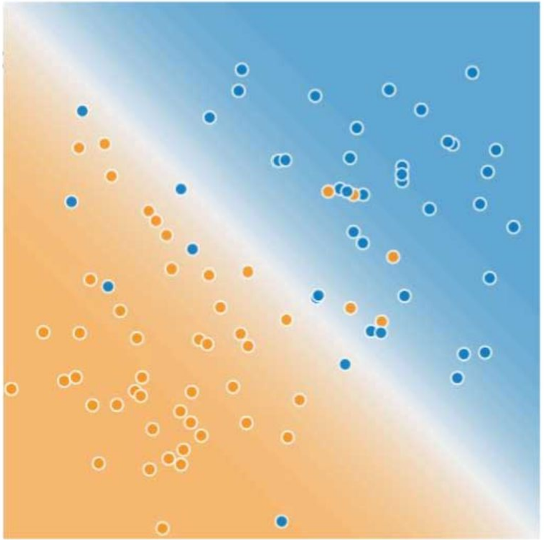
(a) Artificial neuron (b)Multilayered artificial neural network

Image Credit: Kumar et al, 2012

Why do we use Neural Networks?



Training Data



Test Data

→ Applications include: pattern recognition, data analysis and control & clustering

→ Predicting features of data

→ Capturing non-linear relationships

→ Solving complex PDEs straightforwardly

Image Credit: Google

Introduction to Physics-Informed Neural Networks

PINN:

- Python library deepxde
- Physics informed neural network
- Incorporate physical model into NN
- E.g. Burgers equation:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = v\frac{\partial^2 u}{\partial x^2}$$

- Solution has to fulfill PDE and match initial/boundary conditions $\tilde{u}$

Introduction to Physics-Informed Neural Networks

How do we introduce physics?

- Define Loss function $L = w_{data} L_{data} + w_{PDE} L_{PDE}$
- $L_{data} = \frac{1}{N} \sum_{i=1}^{N} (u(x_i, t_i) - \tilde{u}_i)^2$ ensures u(x,t) matches the initial/boundary conditions
- $L_{PDE} = \frac{1}{N_{PDE}} \sum_{j=1}^{N_{PDE}} \left(\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} - v \frac{\partial^2 u}{\partial x^2} \right)^2 |_{(x_j, t_j)}$ ensures u(x,t) obeys PDE
- w_{PDE} and w_{data} can be used to adjust the interplay between both Loss functions
- L_{data} - supervised loss (data to match exists)
- L_{PDE} - unsupervised loss

Introduction to Physics-Informed Neural Networks

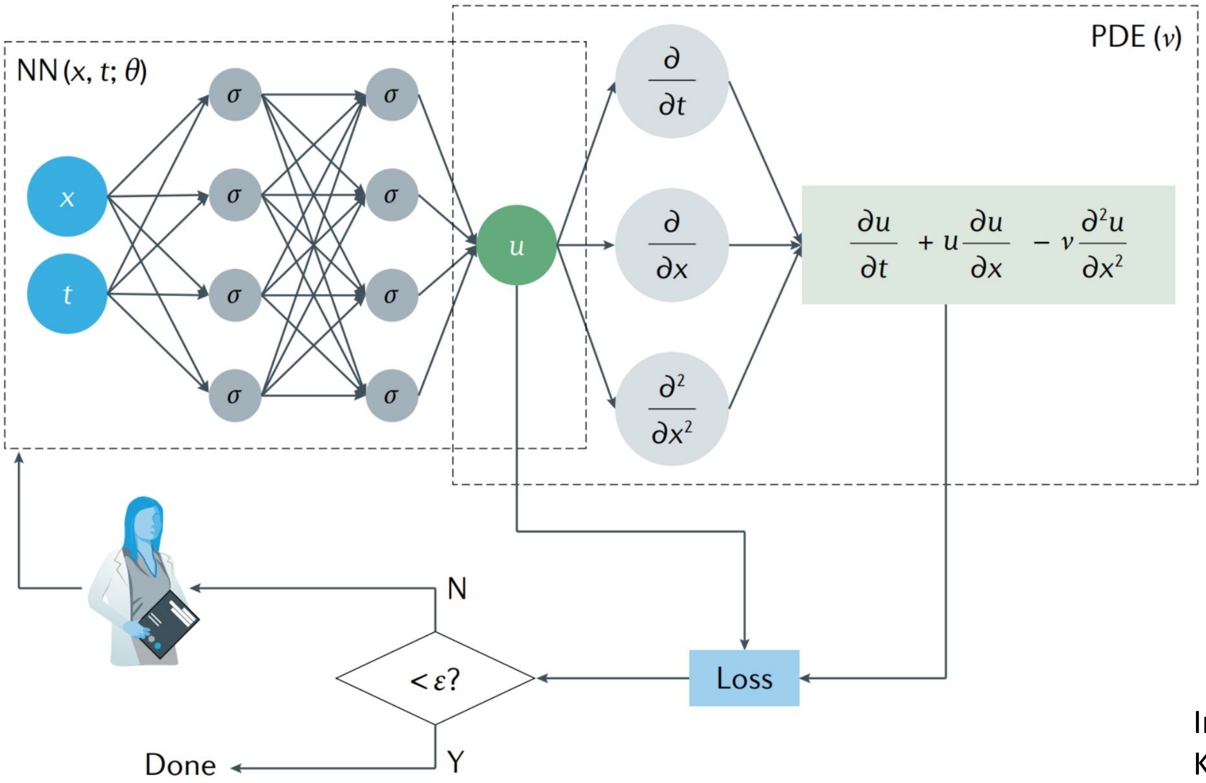


Image Credit:
Karniadakis et al, 2021

Effective One-Body Problem

- The effective one-body problem is a simplification of a central force system involving two particles into an effective single particle in an external potential

- The approach involves a change of coordinates to express the positions as a single difference vector in the center of mass frame

- It also exploits symmetries, which lead to conservation of energy, momentum, and angular momentum, to simplify the problem further

Effective One-Body Problem

- The Newtonian EOB is the special case where the force between the particles is the gravitational force which leads to this Lagrangian:

$$L = \frac{1}{2}\mu\left(\dot{r}^2 + r^2\dot{\theta}^2\right) - \frac{GM\mu}{r}$$

  where μ is the reduced mass, M is the total mass, G is the gravitational constant, r is the separation distance, and θ is the angle

- Plugging into the Euler-Lagrange equations yields:

$$\ddot{r} = r\dot{\theta}^2 + \frac{GM}{r^2} \qquad \ddot{\theta} = -\frac{2\dot{r}\dot{\theta}}{r}$$

Post Newtonian Corrections to EOB

- The post-Newtonian expansion is an approximation of solutions to the Einstein field equations
- It is most accurate in the regime of small speeds and weak fields
- It turns out that PN expansions are rather effective and even apply to situations outside of the expected regimes, e.g. black hole inspirals

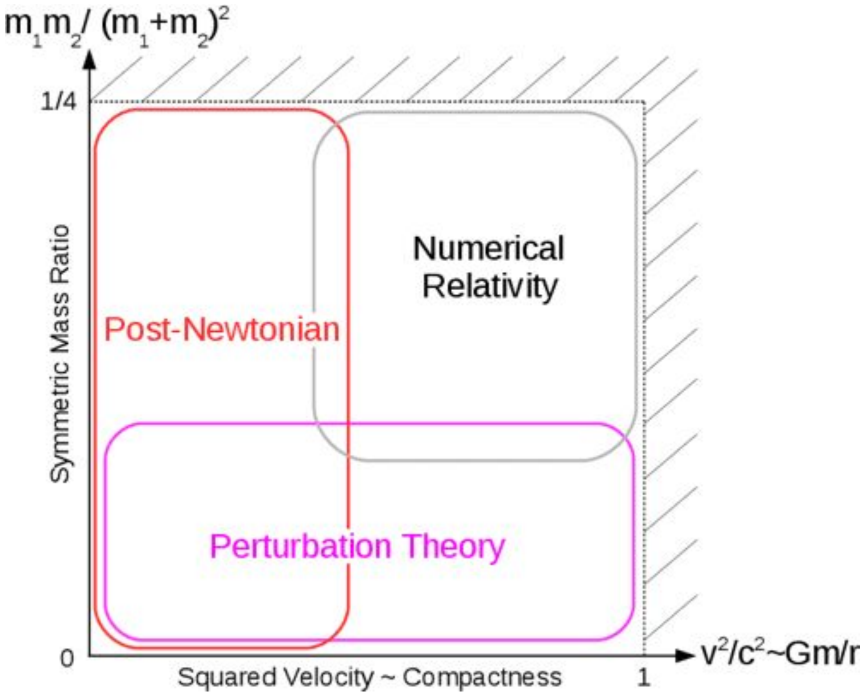


Image Credit: Blanchet, 2018

Post Newtonian Corrections to EOB

- For our purposes, the post-Newtonian corrections can be added as generalized force terms in the Euler-Lagrange equations:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q} = Q_q^{(1PN)} + Q_q^{(2.5PN)} + Q_q^{(DF)}$$

- First order PN correction
- Second PN term which corresponds to the radiation of gravitational waves
- Dynamical friction term due to dark matter (we did not end up including this in our work)

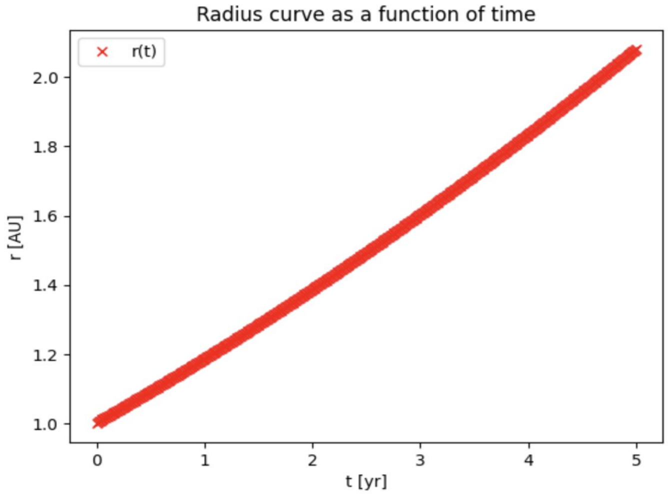
Implementation & Results

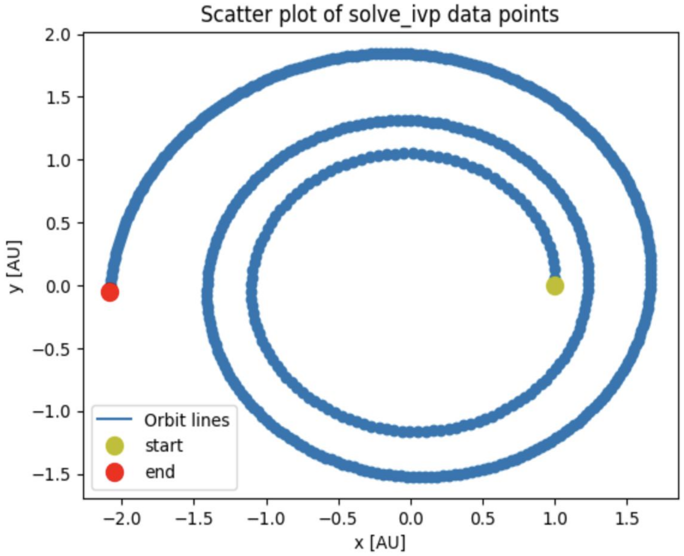
Implementing EOB Equations into Neural Networks

- I implemented **dimensional** equations into the NN and results
- Solve_ivp, then deepxde with variables r, θ
- Important parameters: μ (reduced mass), η (symmetric mass ratio)
- Broad picture: do solve_ivp solutions make sense?
- Expand on specific of learnings:
  - Non-dimensionalization of equations
  - 1PN corrections, rescaling, study of one orbit only
  - Train on steadily increasing time domain for increased accuracy

Dimensional Equations - Classical Solver

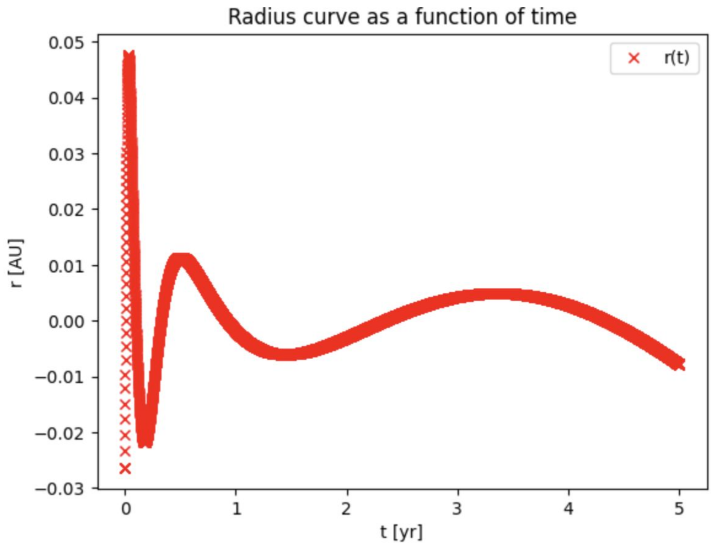
- solve_ivp used to solve differential equations
- $m_1 = 2\ M_\odot$; $m_2 = 30\ M_\odot$; $r_0 = 1$ AU
- Gravitational slingshot orbit: not bounded
- Initial conditions obeyed
- Appears physical, therefore reference solution

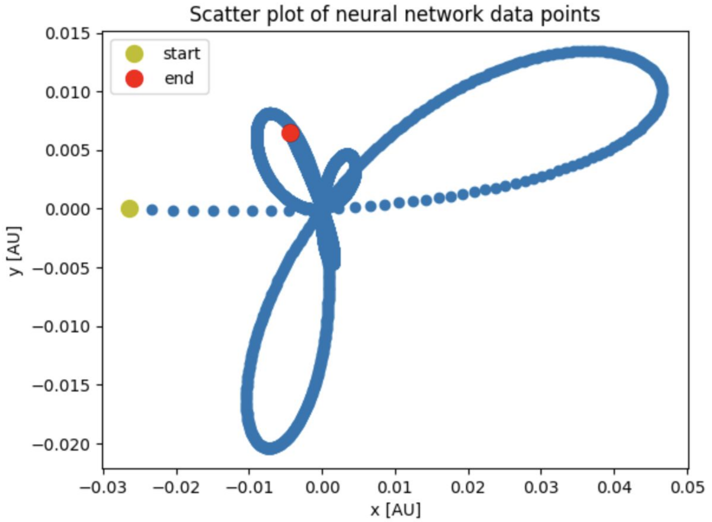




Dimensional Equations - NN

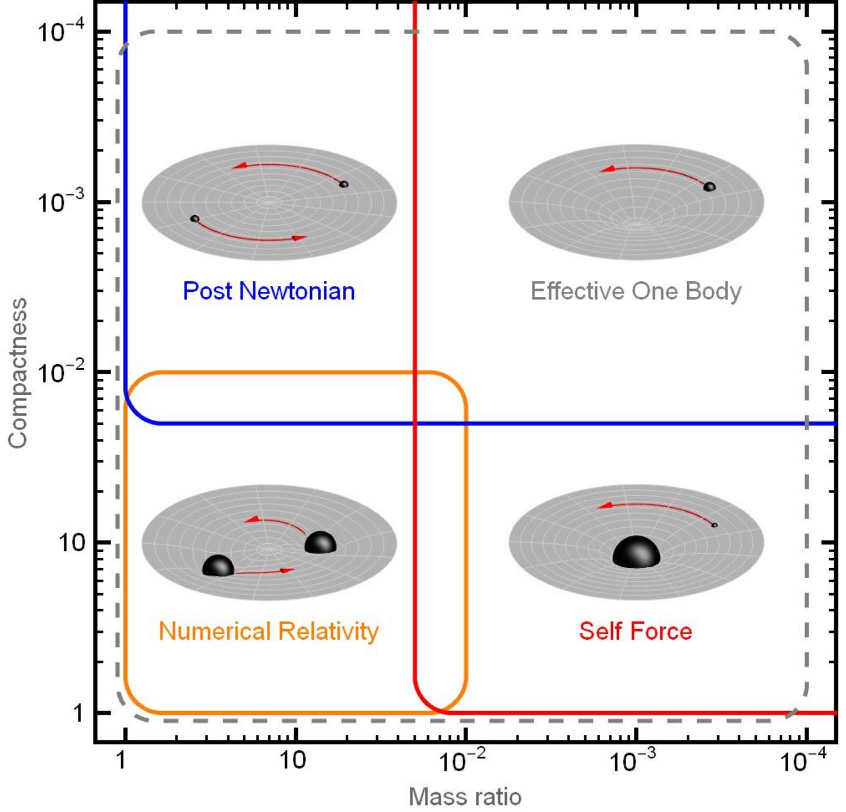
- NN solution bounded and precessing
- Unphysical radius through zero
- The initial conditions are not obeyed
- NN incorrectly learning
- How do our parameters line up with PN regime?





Parameter Regime for PN Corrections

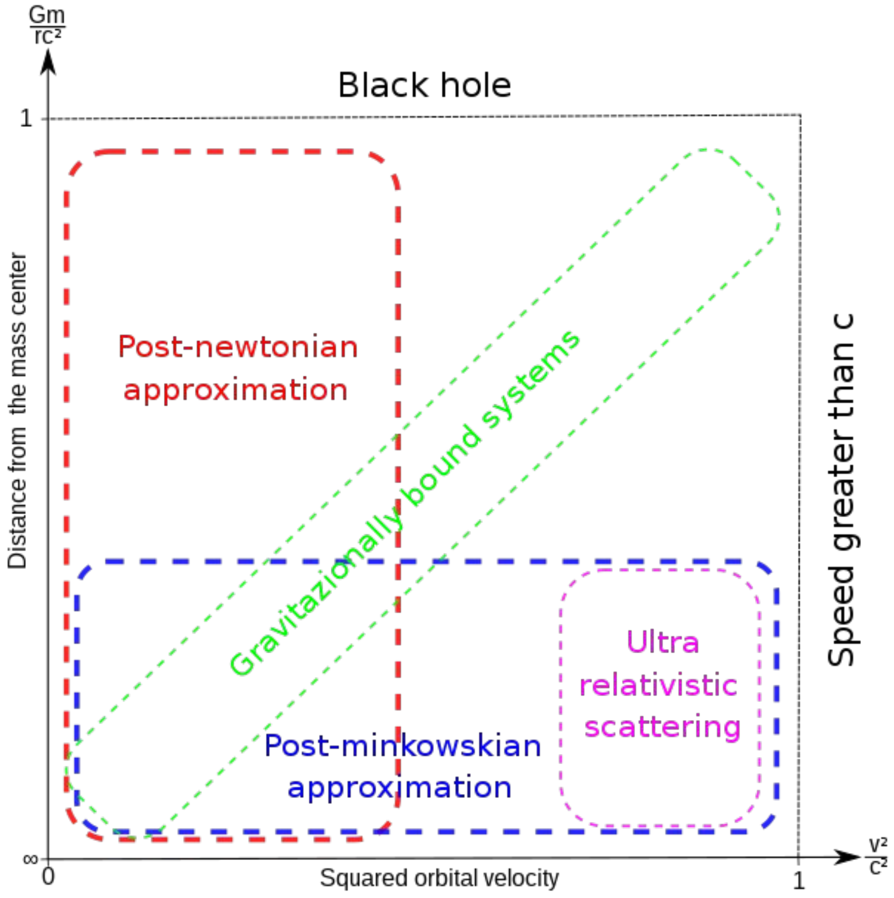
- Compactness: ratio of distance from event horizon R_s/R
- Mass ratio $X = m_2/m_1$: borderline
- Try new mass and radius values:
  - $m_1 = 1\ M_\odot$; $m_2 = 10^{-3}\ M_\odot$; $r_0 = 10$ AU.
- $X \to 0$, $\eta \to X$



Wikipedia

Parameter Regime for PN Corrections

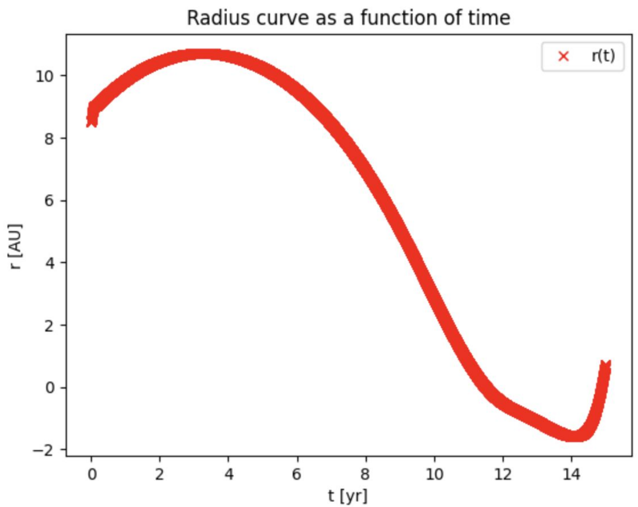
- Gravitationally bound: small region in parameter space
- Further confirmation: solve_ivp physical
- Unbound orbits are physical within PN
- Can NN learn unbound orbit?

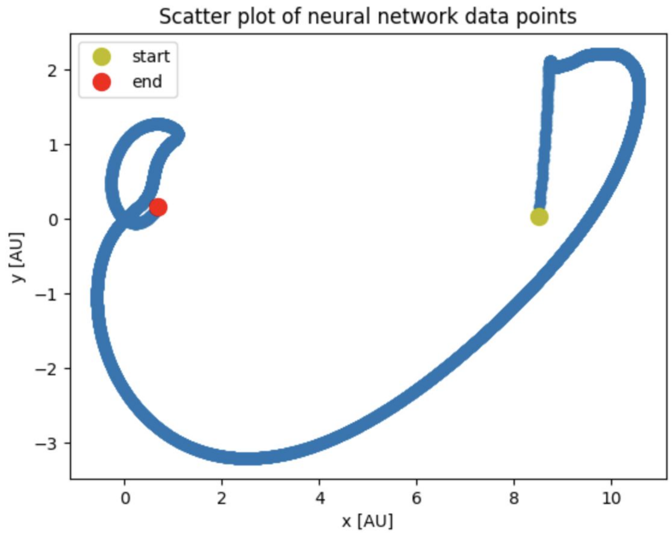


Wikipedia

Dimensional Equation Corrected - NN

- New parameters implemented. Radius passes through zero
- Initial conditions still not obeyed, but better
- Makes even less sense
- Consider dimensionless equations for simplicity





Non-dimensionalised PN Equations

$$Q_{r_*}^{(1PN)} = \frac{Gm\mu}{r_*^2 c^2}\left[(4+2\eta)\frac{Gm}{r_*} - (3\eta+1)r_*^2\dot{\theta_*}^2 + \dot{r_*}^2(-\tfrac{7}{2}\eta+3)\right]$$

$$\Downarrow$$

$$F_r^{(1PN)} \equiv \frac{2R_s}{\mu c^2} Q_{r_*}^{(1PN)} = \left[(2+\eta)r^{-3} - (3\eta+1)\dot{\theta}^2 + \frac{1}{2}(6-7\eta)r^{-2}\dot{r}^2\right]$$

→ ND variables: $r = \frac{r_*}{r_0}$, $t = \frac{t_*}{t_0}$

→ Scale parameters: $r_0 = ct_0 = R_s = \frac{2Gm}{c^2}$

→ Sol’n only depends on η and ICs!

Rastgoo et al, 2023
& courtesy of Erik Weiss

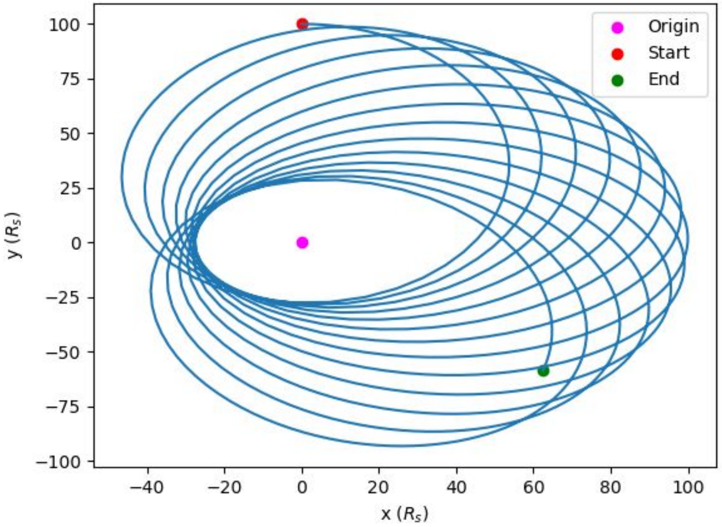
Non-dimensionalised PN Equations

$$Q_{\theta_*}^{(1PN)} = \frac{Gm\mu}{c^2}\dot{r}_*\dot{\theta}_*[(4-2\eta)]$$

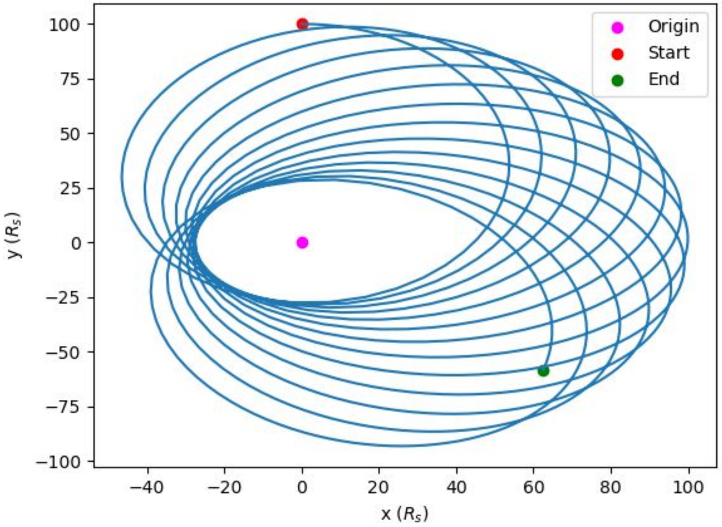
$$F_{\theta}^{(1PN)} \equiv \frac{1}{\mu c^2}Q_{\theta_*}^{(1PN)} = \left[(2-\eta)\dot{r}\dot{\theta}\right]$$

Rastgoo et al, 2023
& courtesy of Erik Weiss

Non-dimensional Solutions - RK Method



1PN Correction with RK methods



2.5PN Correction with RK methods

Non-dimensional Solutions - RK Method

$$\boxed{\frac{df_{\rm b}}{dt} = \frac{192\pi}{5} f_{\rm b}^2 \left(\frac{2\pi G \mathscr{M} f_{\rm b}}{c^3}\right)^{5/3} F(e)}$$

→ $\frac{\dot{f}}{f} \propto \eta f^{\frac{8}{3}} \approx 10^{-12}$

→ $\frac{\Delta f}{f} \approx 10^{-8}$ over one orbit

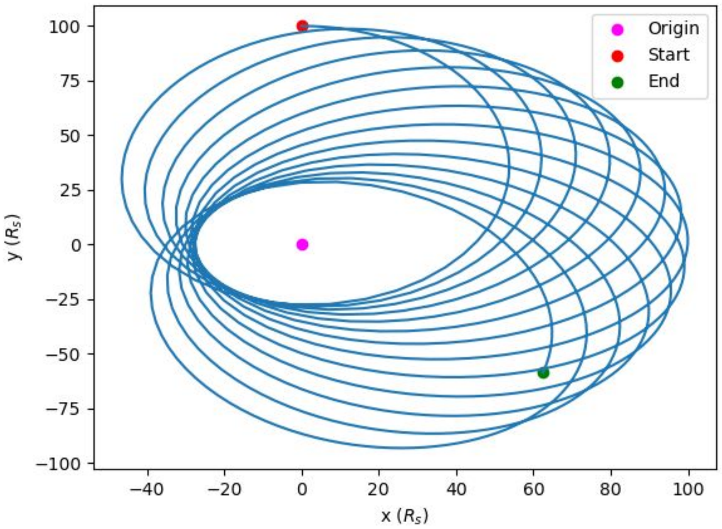
→ LOTS of orbits to see a difference

→ Numerical errors outweigh frequency

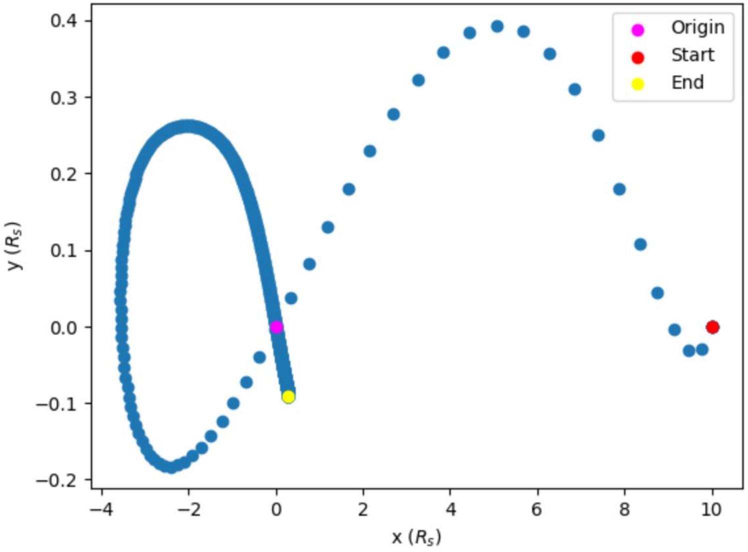
increase

Credit: Will, 2011

Non-dimensional Solutions - RK vs. PINN



1PN Correction with RK methods



1PN Correction with Neural Networks

Implementation & Findings

Key findings for PINNs:

- Reciprocals lead to infinite losses → multiply equations by r^n
- NN prefers values of order unity → rescale reference scales depending on initial radius r_i in units of $\frac{GM}{c^2}$

$$r_0 = \frac{GMr_i}{c^2} \qquad t_0 = \frac{GMr_i\sqrt{r_i}}{c^3}$$

- Equations at 1 PN order (r, t dimensionless. Different to Zena's)

$$\ddot{r} = r\dot{\phi}^2 + \frac{1}{r^2}\left[-1 + (4+2\eta)\frac{1}{rr_i} - (3\eta+1)\frac{r^2}{r_i}\dot{\phi}^2 + \frac{\dot{r}^2}{r_i}\left(-\frac{7}{2}\eta + 3\right)\right]$$

$$\ddot{\phi} = \frac{1}{r^2 r_i^2}(4 - 2\eta - 2rr_i)\dot{r}r_i\dot{\phi}$$

Implementation & Findings

Key findings for PINNs:

- Loss weights crucial to precise result
- Second equation seems less important → choose lower loss weights
    - Why? Newtonian case: angular momentum $L = mr^2\dot{\phi}$ conserved
    - $\ddot{r}$ depends on r only
    - Once we have r it is easy to find ϕ
    - PN corrections are small, so this should (approximately) still hold
- Train on newtonian case first, add PN corrections later
- Difficult to extend time beyond one orbit

Results

Improvements lead to good results

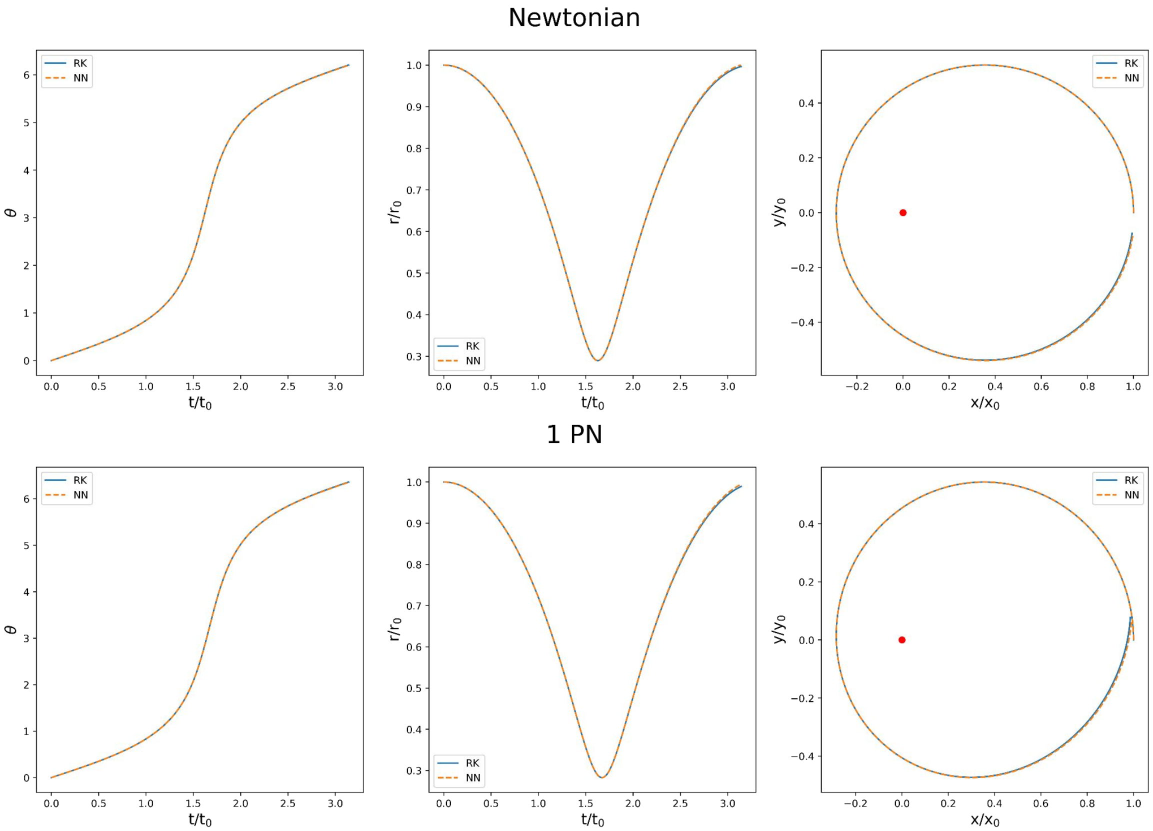
NN matches RK

Newtonian:

- Elliptical, closed orbit

1 PN:

- Precession visible



Results

Now increase timespan by 10%:

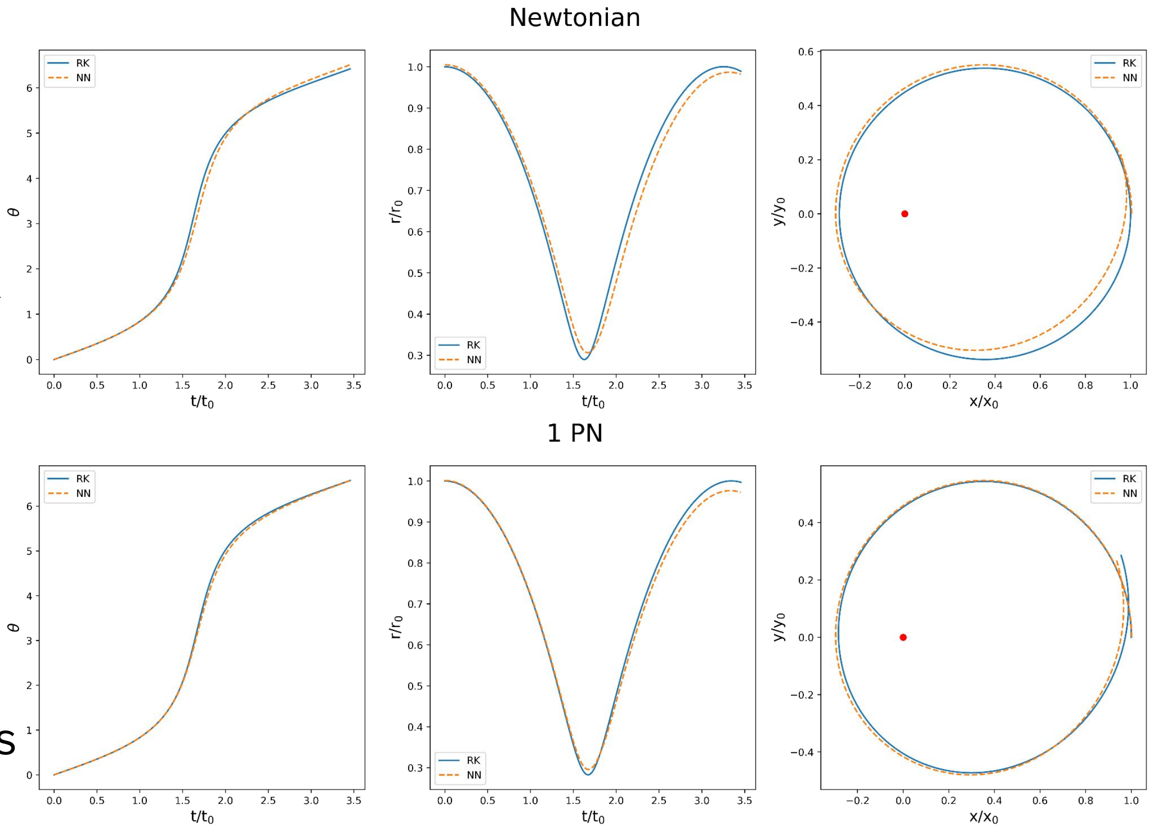
Result from NN deviates from RK

Tendency to underestimate r

Solution for θ better than for r

However, orbit still close to what is expected

Longer training with more points leads to better solution, but takes longer



Difficulties Encountered in NN

- One difficulty we encountered while training the neural network was that it often did not obey the initial conditions that were fed into it
- This issue was partially resolved by modifying the output of the network such that it satisfied these conditions, e.g. we could modify the radius in the following way:

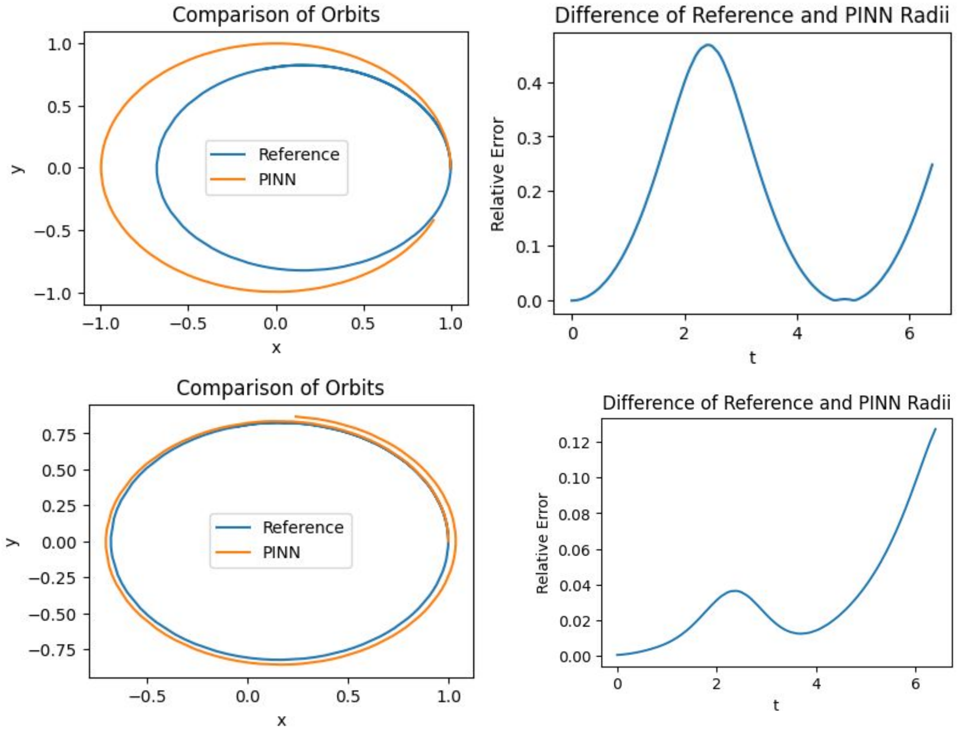
$$r_{new} = r_{old}t + r_0$$

- The neural network also usually struggled to follow an elliptical path, so we trained it on a circle first to get it moving along the correct trajectory

Results for Newtonian NN

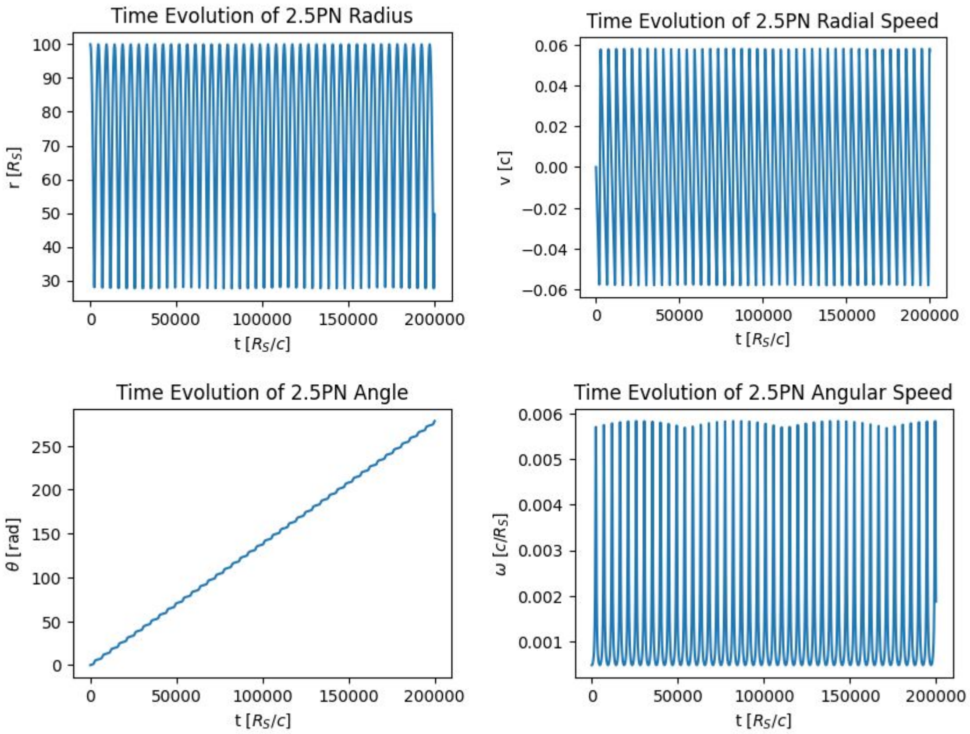
- Training the neural network on the entire time interval generally led to poor results

- To fix this issue, we instead trained the network on a small time segment and then slightly increased this segment and continued training, etc.



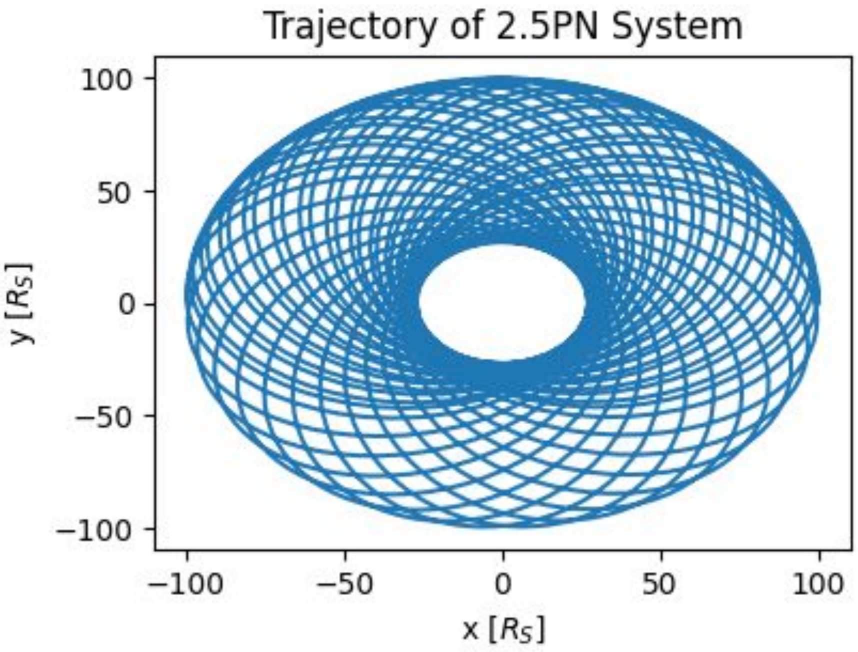
Results for PN Corrections

- Here is an example of the result we obtained for the evolution of coordinates and velocities of the reference solution for the 2.5PN correction



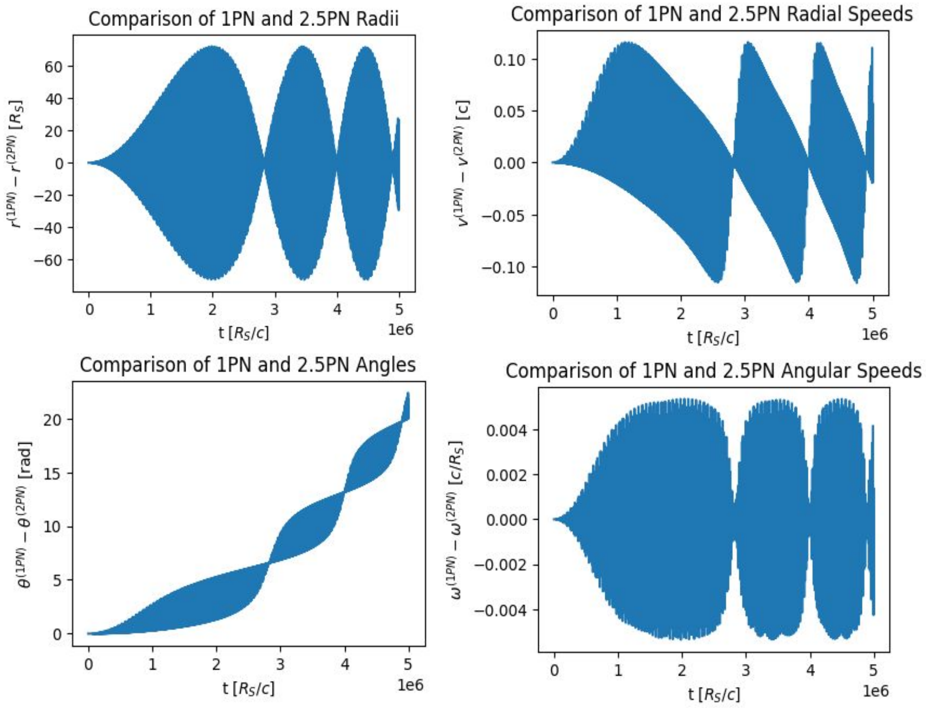
Results for PN Corrections

- This plot shows how the system (up to the 2.5PN correction) evolves in space
- As seen here, the additional terms in the PN expansion lead to precession of the orbit



Results for PN Corrections

- Here we show how the 1PN corrected solution and the 2.5PN corrected solution differ from each other

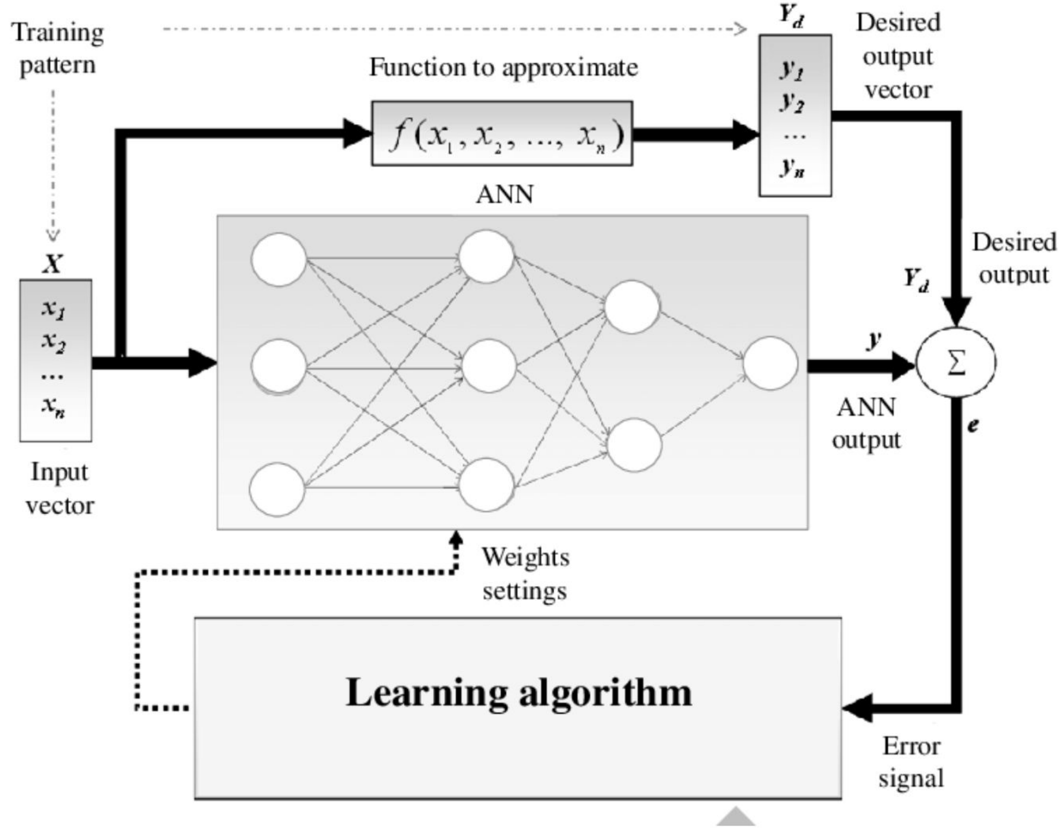


Future Directions

Future Directions

- Hard constraints
  - Want to prevent radius from passing through zero.
- Consequently, NN will realize $r \leq 0$ unphysical
  - Will hard constraints improve NN learning?
  - Slingshot orbits?
  - Bounded orbits obeying initial conditions?

Future Directions



→ Try training network on solve.ivp sol'n

→ Still a PINN in that case?

→ Solve_ivp data for one set of params/ICs may improve results!

Image Credit: Quinonez et al, 2015

Future Directions

- Repeat experiments using many different network architectures to find which setup more consistently leads to the best output
  - Vary the number of nodes per layer and the number of layers
  - Try different activation functions
  - Try different optimizers
  - Change how many iterations to train the model for

Summary

- NN can help us solve EOB
- In-/Output should be [0,1] $\rightarrow$ Non-dimensionalize and scale equations
- Network parameters need to be accurately adjusted to fit problem
- Pre-existing solutions useful to assess accuracy of results and can be used to train NN
- So far PINN results lack accuracy: problems with periodicity

Thank you for listening!