A New Algorithm for Machine Learning, AI and Predictions

夏志宏

大湾区大学(筹)(大湾区高等研究院)
Northwestern University

Based on Joint work with Xin Li, Hongkun Zhang, Xiaotao Zheng

HKUST, Feb 24, 2025

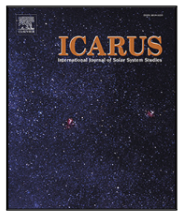
Some Interesting Astronomy
(Joint work with Jian Li (黎健), et el)

Icarus 425 (2025) 116340

Contents lists available at ScienceDirect

Icarus

journal homepage: www.elsevier.com/locate/icarus



Research Paper

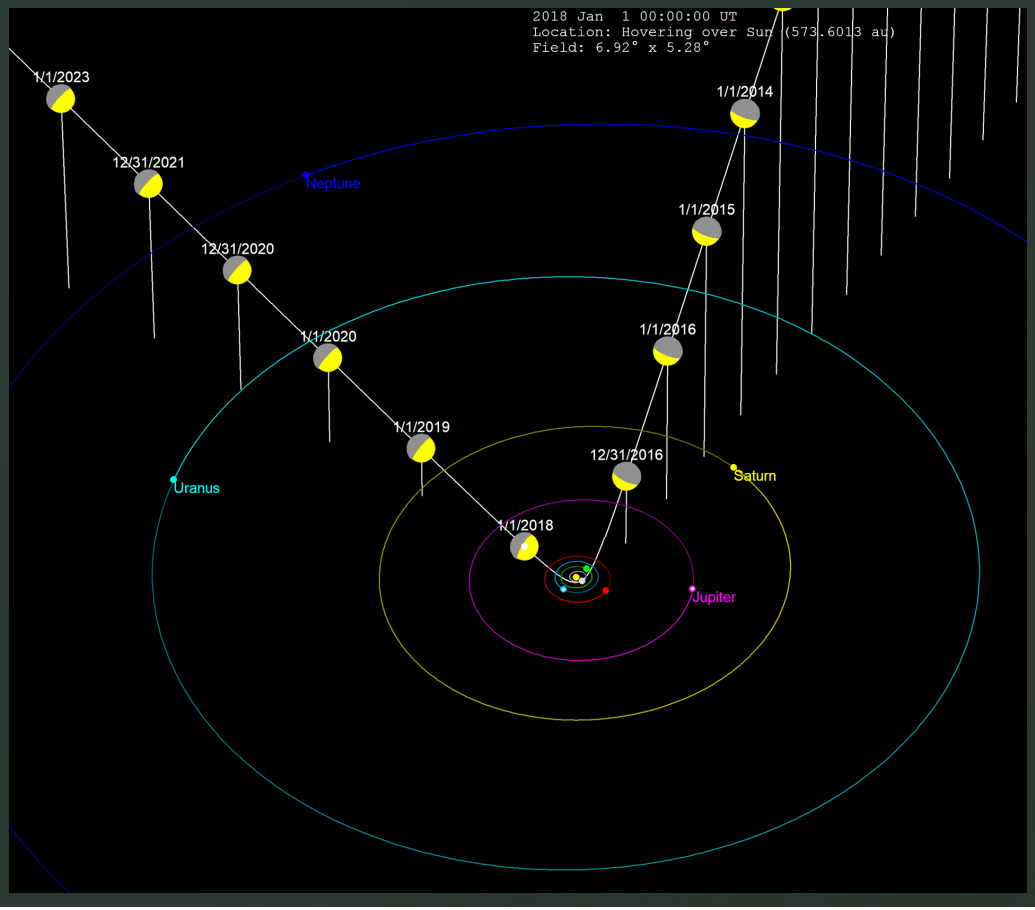
Resonant amplitude distribution of the Hilda asteroids and the free-floating planet flyby scenario☆

Jian Li [a,b,*], Zhihong Jeff Xia [c,d,**], Hanlun Lei [a,b], Nikolaos Georgakarakos [e,f], Fumi Yoshida [g,h], Xin Li [i]

我们发现很强的证据，几亿年以前曾经有一颗流浪大行星飞越太阳系!

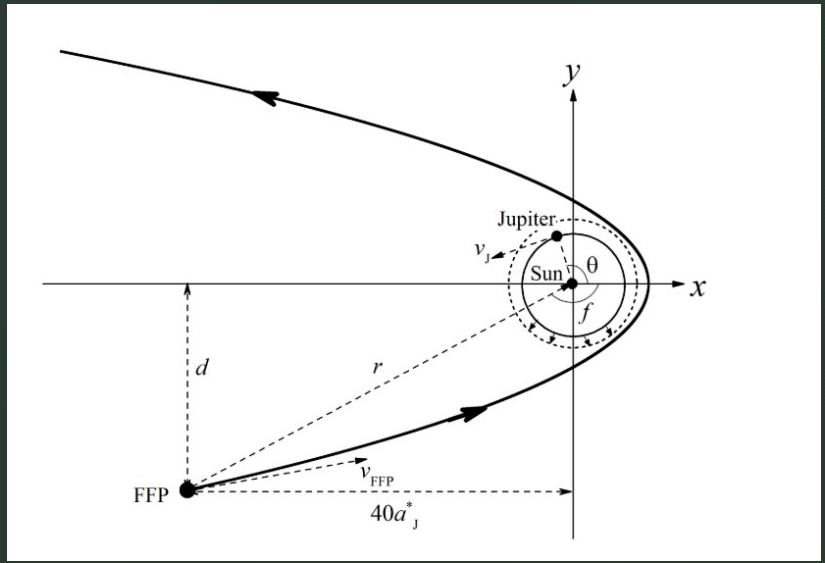
We found compelling evidence suggesting that our solar system was once invaded by a free-floating planet several hundred million years ago!

Oumuamua（奥陌陌）



- Came from outside of solar system，discovered in 2017
- At the time, it was 33 mkm away
- About 230 meters long
- Another alien object discovered in 2019

Free-Floating Planet Hypothesis



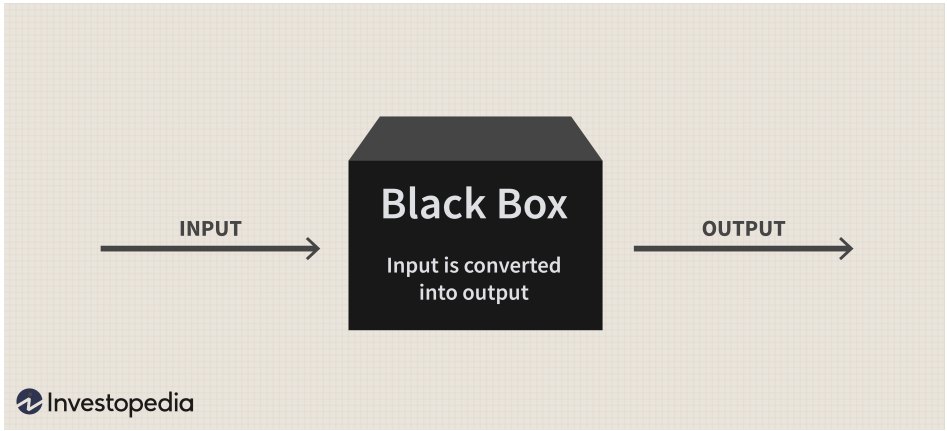
A&A 674, A138 (2023)
https://doi.org/10.1051/0004-6361/202346223

Astronomy & Astrophysics

The invasion of a free-floating planet and the number asymmetry of Jupiter Trojans

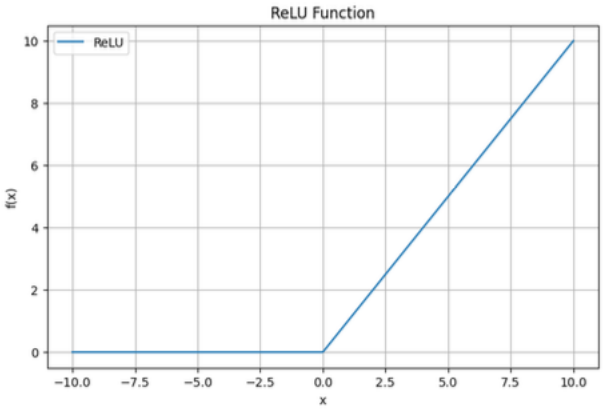
Jian Li[1,2], Zhihong Jeff Xia[3], Nikolaos Georgakarakos[4,5], and Fumi Yoshida[6,7]

Machine learning, what's inside the blackbox?



- Given a dataset $(x_i, g(x_i))$, $i = 1, 2, \ldots, n$ and a new point x, find or approximate $g(x)$!
- Is it possible? Yes, if g is nice, otherwise $g(x)$ can be anything.
- How nice? Real analytic!
- Complex analysis: data to data!

- $g(x)$ must be expressed in a functional basis;
- $g(x)$ must be trainable (self-learning);
- Popular activation functions: ReLU and Sigmoid



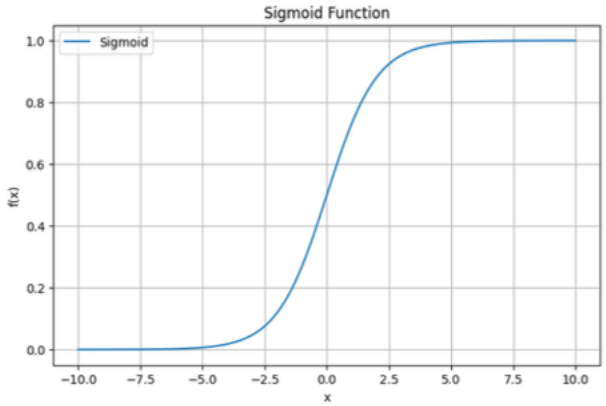
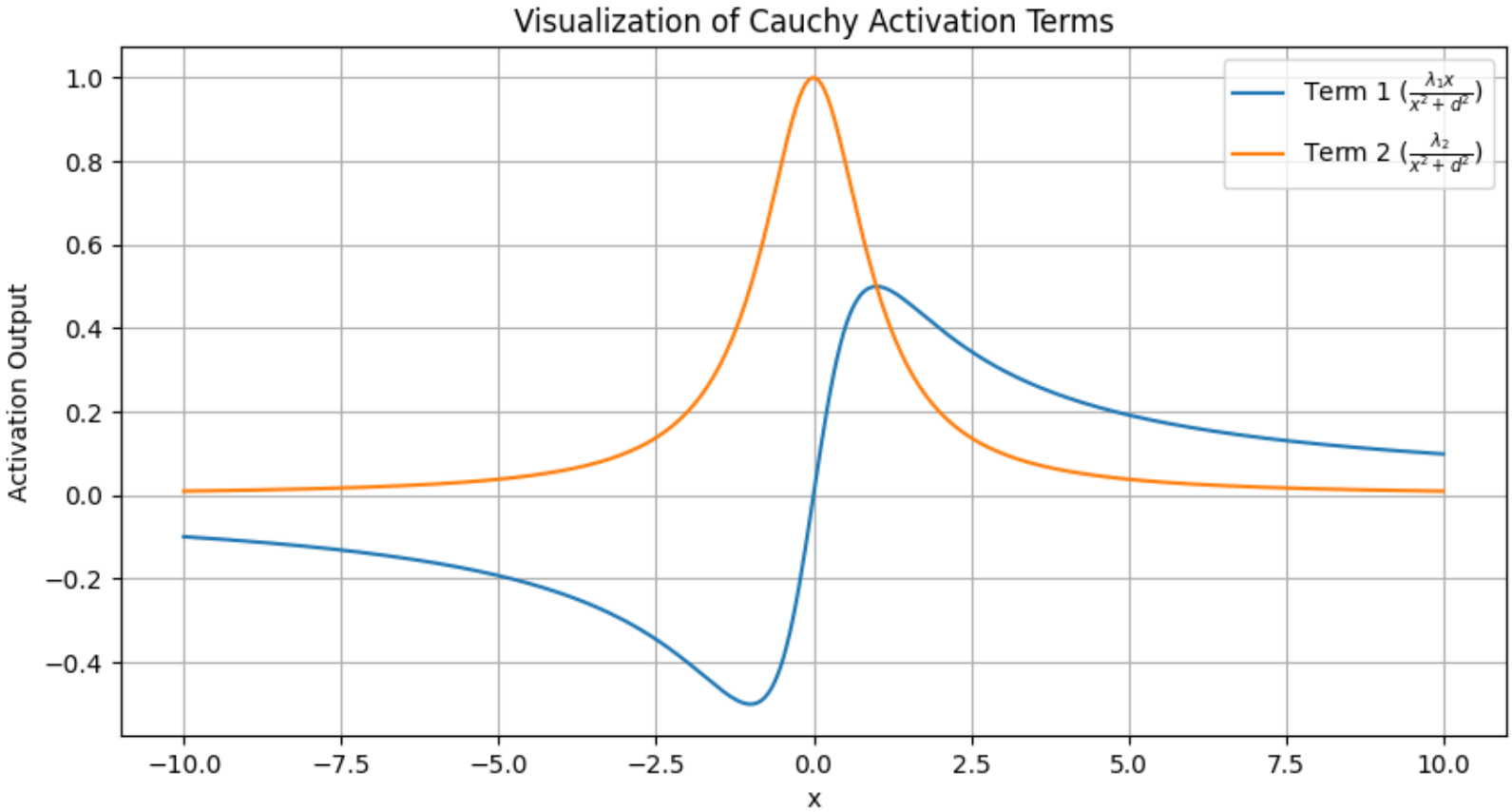


Figure: ReLU and Sigmoid

Cauchy Activation



Theorem (Cauchy Integral Theorem)

If $f(z)$ is a complex function that is analytic (holomorphic) on a compact domain U, with boundary $\partial U = C$ is a simple, closed curve, for all $z \in int(U)$,

$$f(z) = \frac{1}{2\pi i} \oint_C \frac{f(\xi)}{\xi - z} \, d\xi. \tag{1}$$

Here, $\oint_C$ denotes a line integral taken counter-clockwise around the closed contour C.

Cauchy's integral formula states that the value of a function at any point z can be determined using known values of the function on a closed curve C that encloses the point. This concept is remarkably similar to machine learning principles, where a function's values at new points are inferred from known values.

Theorem (Cauchy Approximation Theorem)

Let $f(z^1, z^2, \ldots z^N)$ be an analytic function in an open domain $U \subset \mathbb{C}^N$ and let $M \subset U$ be a compact subset in U. Given any $\epsilon > 0$, there is a list of points $(\xi_k^1, \ldots, \xi_k^N)$, for $k = 1, 2, \ldots, m$, in U and corrresponding parameters $\lambda_1, \lambda_2, \ldots, \lambda_m$, such that

$$\left| f(z^1, z^2, \ldots z^N) - \sum_{k=1}^{m} \frac{\lambda_k}{(\xi_k^1 - z^1)(\xi_k^2 - z^2) \cdots (\xi_k^N - z^N)} \right| < \epsilon, \qquad (2)$$

for all points $(z^1, z^2, \ldots z^N) \in M$.

The order of approximation is

$$o((\frac{1}{m})^k)$$

for any integer k!

Theorem ("Universal" Universal Approximation Theorem)

Let Φ be a family of functions in $C(\mathbb{R}, \mathbb{R})$ with the universal approximation property. Let

$$\Phi^N = \{\phi_a(a_1x_1 + a_2x_2 + \ldots + a_Nx_N) \mid (a_1, \ldots, a_N) \in \mathbb{R}^N, \phi_a \in \Phi\}$$

Then, the family of functions Φ^N has the universal approximation property in $C(\mathbb{R}^N, \mathbb{R})$, i.e, every continuous function in $\mathbb{R}^N$ can be approximated by a linear combination of functions in Φ^N, uniformly over a compact subset in $\mathbb{R}^N$.

If it works for 1-D, then it work for any dimension!

Applications: two self-learning networks

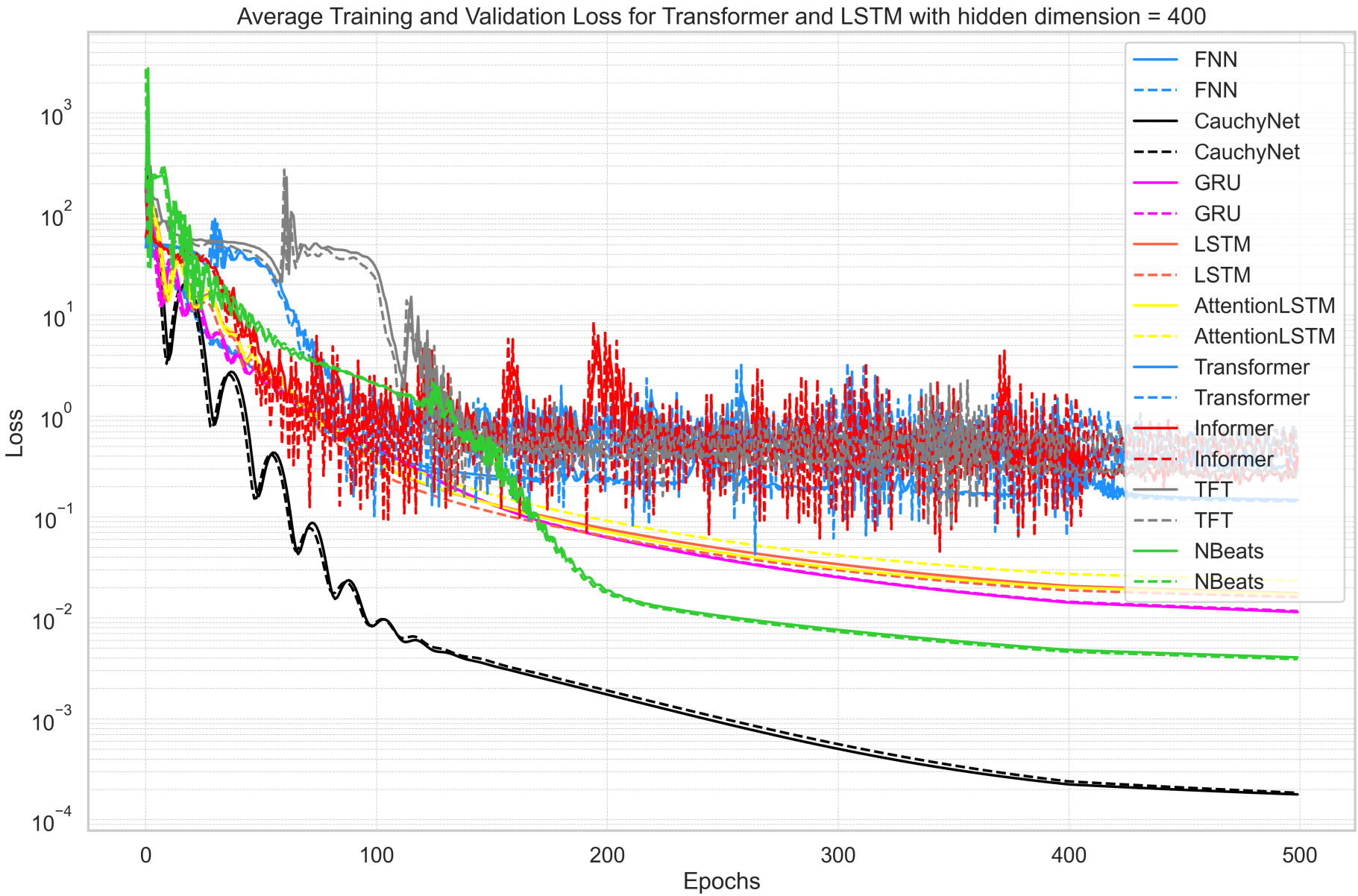
- For scientifical computation and engineering: solving differential equations, data fitting, inverse problems

  The CauchyNet: Using Cauchy Approximation Theorem

- For artificial intelligence (AI), for example, image recognition, transformers (ChatGPT)

  The XNet: Using Fully Connected Linear Layer + Cauchy Activation

Comparison


Average Training and Validation Loss for Transformer and LSTM with hidden dimension = 400
Loss
Epochs
FNN
FNN
CauchyNet
CauchyNet
GRU
GRU
LSTM
LSTM
AttentionLSTM
AttentionLSTM
Transformer
Transformer
Informer
Informer
TFT
TFT
NBeats
NBeats

Further Comparison

	Training Loss	Val Loss	Test Loss	Test MAE	Time (s)	Num Parameters
CauchyNet	**0.00018**	**0.00019**	**0.00019**	**0.01033**	5.96	**128**
MLP	0.14840	0.14316	0.14870	0.29960	5.09	4353
GRU	0.01144	0.01171	0.01259	0.06577	9.50	12929
LSTM	0.01758	0.01608	0.02566	0.09145	10.02	17217
AttentionLSTM	0.01722	0.02338	0.02257	0.08535	10.88	17282
Transformer	0.28937	0.37719	0.35716	0.50329	15.53	50625
Informer	0.37888	0.48884	0.46792	0.52238	15.50	50625
TemFusionTransformer	0.29353	0.39218	0.40044	0.52959	22.44	104321
NBeats	0.00407	0.00391	0.00403	0.04403	**4.93**	579

Figure: Comparison with popular models

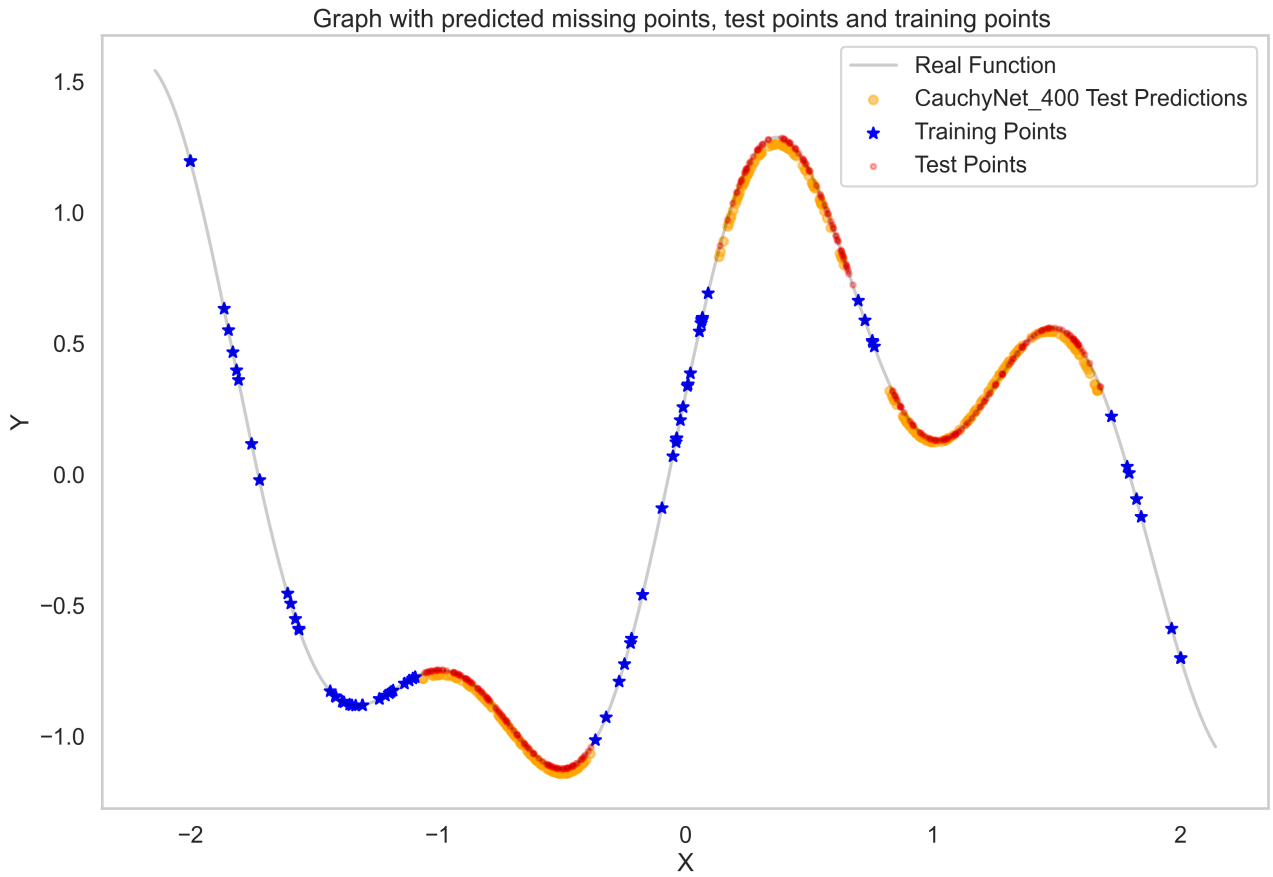


Figure: The figure illustrates the learned and actual missing data in the time series. The neura network model we use was the CauchyNet(400).

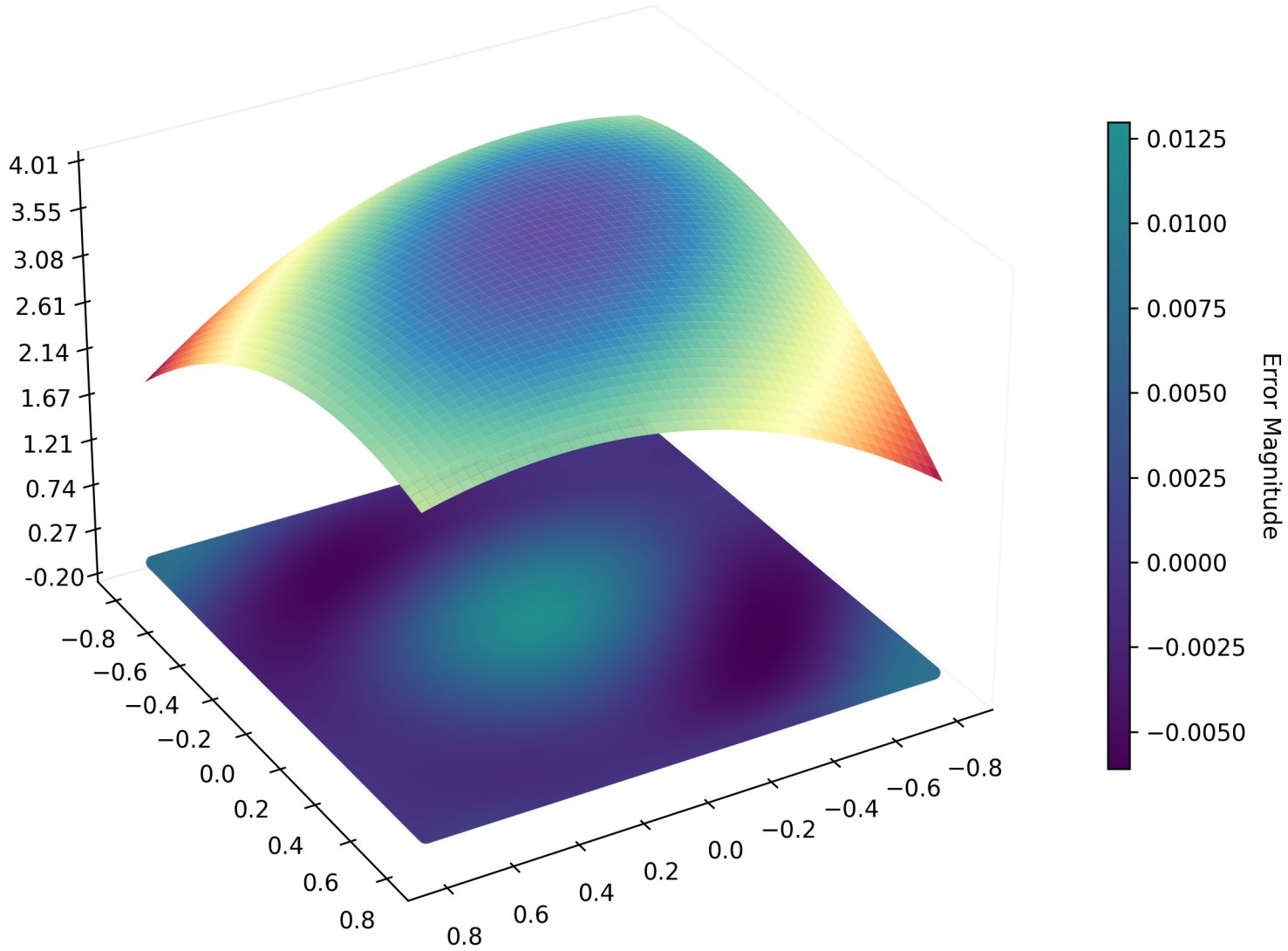


Figure: The figure illustrates the learned surface, as well as the error surface.

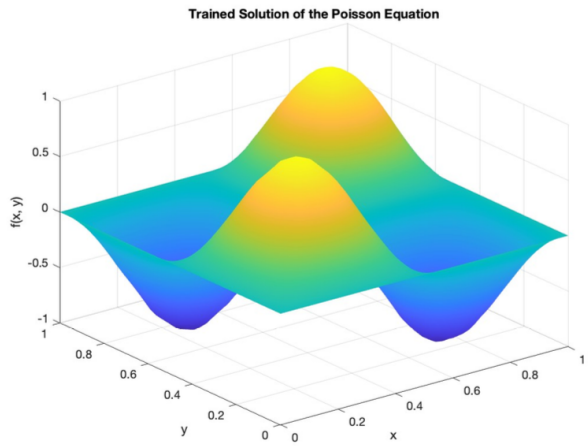


Figure 17: Cauchy activation

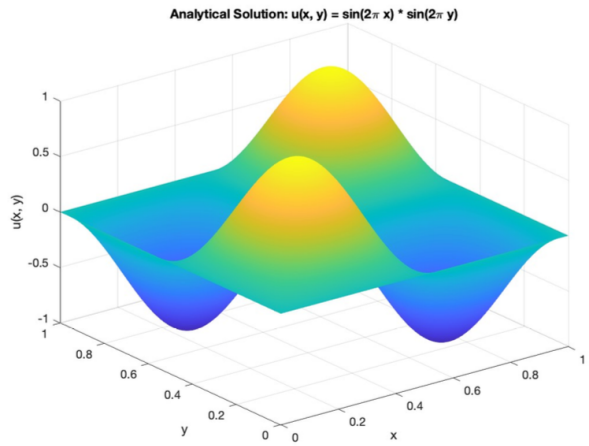


Figure 18: Analytic solution

Figure: Solving Poisson Equation.

XNet: MNIST

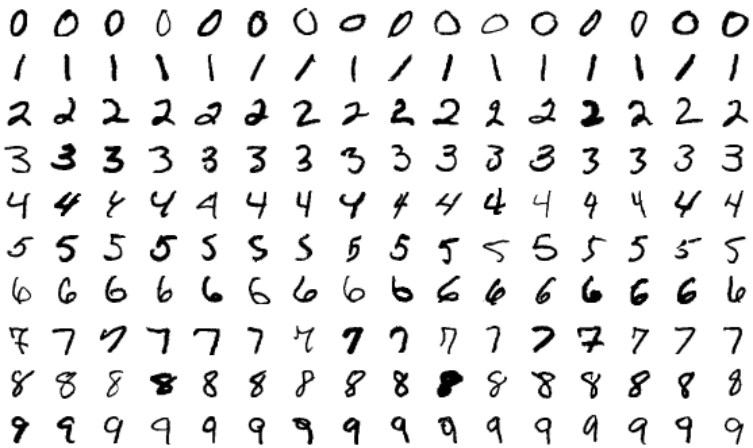
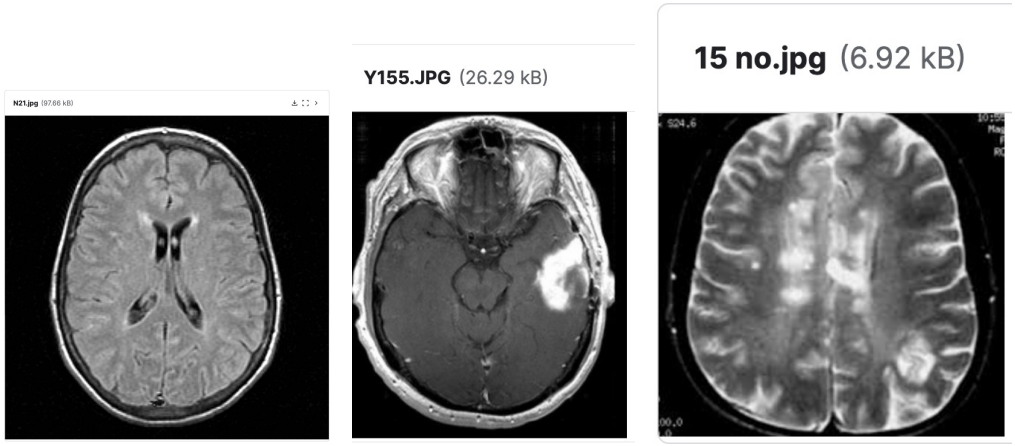


Figure: Test on MNIST

XNet with one hidden layer: accuracy 98.5%.
XNet with one convolution and one hidden layer: accuracy 98.9%.
ReLU with one convolution and one hidden layer: accuracy 97%.
AlexNet, with more than 10 convolution layers, accuracy 99.1%.
Despite the significant difference in the number of parameters, our models achieve high accuracy with fewer parameters.

Tumor



Medical data: Medical imaging sourced from Kaggle.

- **Brain Scan Tumor Classification**

Method	Dataset	Test Accuracy
Deep CNN	Brain Tumor 1 (Binary)	88%
Our Method	Brain Tumor 1 (Binary)	98%
Deep CNN	Brain Tumor 2 (Multiclass)	90%
Our Method	Brain Tumor 2 (Multiclass)	92%

Dataset 1 contains fewer than 1000 images sourced from Kaggle, Dataset 2 contains fewer than 10,000 images.

1. Cifar10:
   60000 32x32 color images in 10 classes, with 6000 images per class.
   - ResNet9, accuracy 92%, using 5,000,000 parameters.
   - XNet + 1 convolution, accuracy 90%, using less than 10,000 parameters.
   - Further hyperparameter tuning is anticipated to potentially exceed the ResNet9's 92% accuracy.
2. Cifar 100:
   - ResNet56, accuracy 75%, using 850,000 parameters
   - XNet + 1 convolution, accuracy 68% <5000 parameters.
   - Our model achieves high accuracy with significantly fewer parameters. Further hyperparameter tuning is anticipated to potentially exceed the ResNet56's 75% accuracy

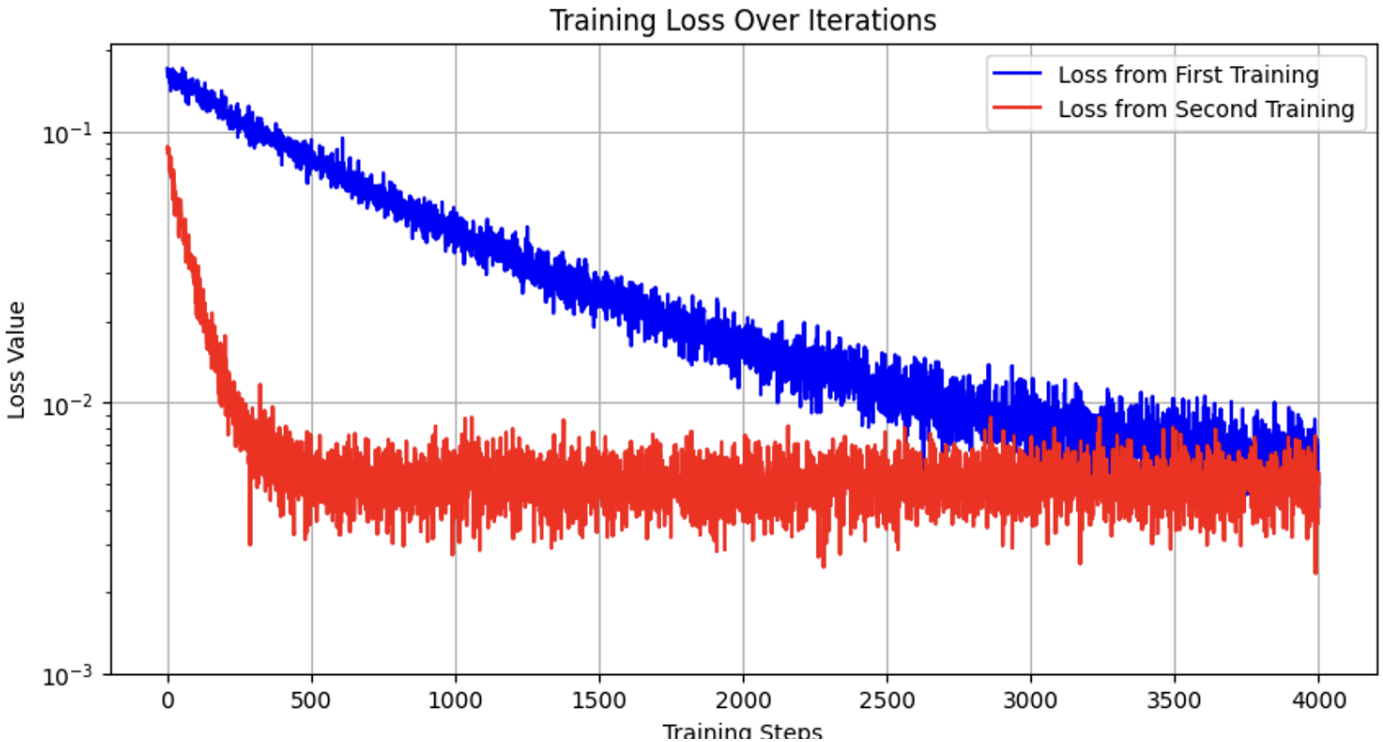
100 dimensional Allen-Cahn Equation

Allen-Cahn Equation

a reaction-diffusion equation for the modeling of phase separation and transition in physics. Here we consider a typical Allen-Cahn equation with the "double-well potential" in 100-dimensional space

$$\frac{\partial u}{\partial t}(t,x) = \Delta u(t,x) + u(t,x) - [u(t,x)]^3,$$

with initial condition $u(0,x) = g(x)$.



Compare XNet with KAN

KAN uses Kolmogorov-Arnold Representation Theorem. We compared XNet with various applications. This is a typical comparison.

Table: Comparison of XNet and KAN on the Poisson equation.

Metric	MSE	RMSE	MAE	Time (s)
PINN [2,20,20,1]	1.7998e-05	4.2424e-03	2.3300e-03	48.9
XNet (20)	1.8651e-08	1.3657e-04	1.0511e-04	57.2
KAN [2,10,1]	5.7430e-08	2.3965e-04	1.8450e-04	286.3
XNet (200)	1.0937e-09	3.3071e-05	2.1711e-05	154.8

More Comparison

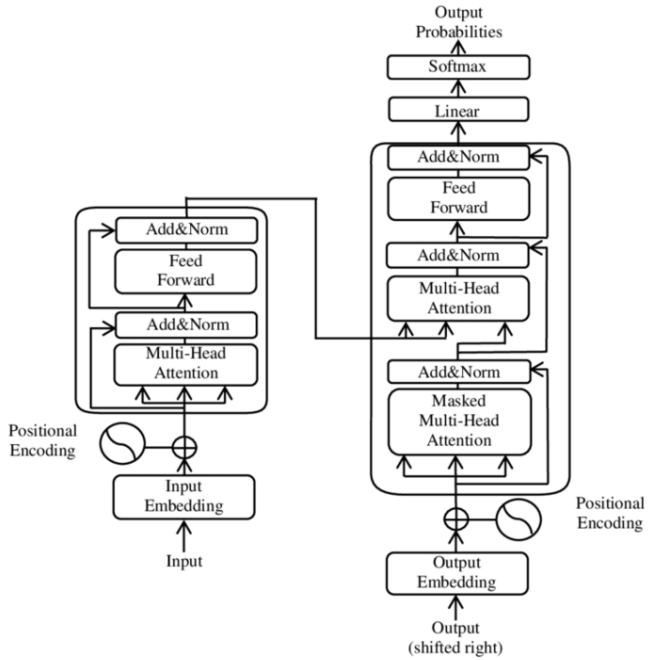
Model	L^2 error	MSE	L_∞ norm
PINN [1]	1.23×10^{-1}	4.76×10^{-3}	2.07×10^{-1}
gPINN [35]	7.50×10^{-2}	1.75×10^{-3}	9.70×10^{-2}
RBA [41]	1.90×10^{-2}	1.12×10^{-4}	3.89×10^{-2}
gPINN + RBA	6.54×10^{-2}	1.33×10^{-3}	1.16×10^{-1}
compleX-PINN	$\mathbf{3.67 \times 10^{-3}}$	$\mathbf{4.18 \times 10^{-6}}$	$\mathbf{1.01 \times 10^{-2}}$

Numerical results of the Diffusion-reaction equation for different models (under the same setup).

Model	L^2 error	MSE
PINN [1]	4.46×10^{-1}	6.75×10^{-2}
RBA [41]	1.76×10^{-1}	9.73×10^{-3}
compleX-PINN	7.34×10^{-3}	1.68×10^{-5}
compleX-RBA	$\mathbf{4.23 \times 10^{-4}}$	$\mathbf{5.59 \times 10^{-8}}$

Numerical results of the 1D Wave equation for different models (under the same setup).

XNet: LLM (ChatGPT), Transformer



Injecting XNet into a Transformer featuring two encoder blocks for efficient sequence processing:

Achieve far better accuracy on the test dataset with a significantly lower number of parameters, and reduce the training time per epoch to 1/10 of the original time on the Wikitext-2 dataset.

- The X-Net is very accurate, sometimes too accurate!

- CauchyNet and X-Net are yet to be tested on larger platforms.

Space-Time

Given a finite dimensional dynamical system $f : M \to M$, and an observable $g : M \to \mathbb{R}$, Birkhoff Ergodic Theorem States that the space average of g is the same as the time average of g for any generic point of an ergodic invariant measure. This indicate certain space-time equivalency, which becomes particularly important when the original dynamical system is either unknown or too complex. More precisely

Theorem (Birkhoff Ergodic Theorem)

Let x be a genric point for an ergodic probability measure μ, then

$$\lim_{k\to\infty} \frac{1}{n}\left(g(x) + g(f(x)) + \ldots g(f^{n-1}(x))\right) = \int_M g d\mu$$

Data driven prediction

Given a sequence of observables

$$g(x), g(f(x)), \ldots, g(f^n(x))$$

Or, many sequences of observables

$$g(x_1), g(f(x_1)), \ldots, g(f^n(x_1))$$

$$g(x_2), g(f(x_2)), \ldots, g(f^n(x_2))$$

$$g(x_3), g(f(x_3)), \ldots, g(f^n(x_3))$$

$$\ldots$$

- Can we predict $g(f^{n+1}(x_1)), g(f^{n+2}(x_1)), \ldots$?
- For a new point y, can we predict

$$g(y), g(f(y)), g(f^2(y)), \ldots$$

  Given that a first few values are known?
- Can we recover the system $f : M \to M$ from sample data for a single observable g?

n-body problem example

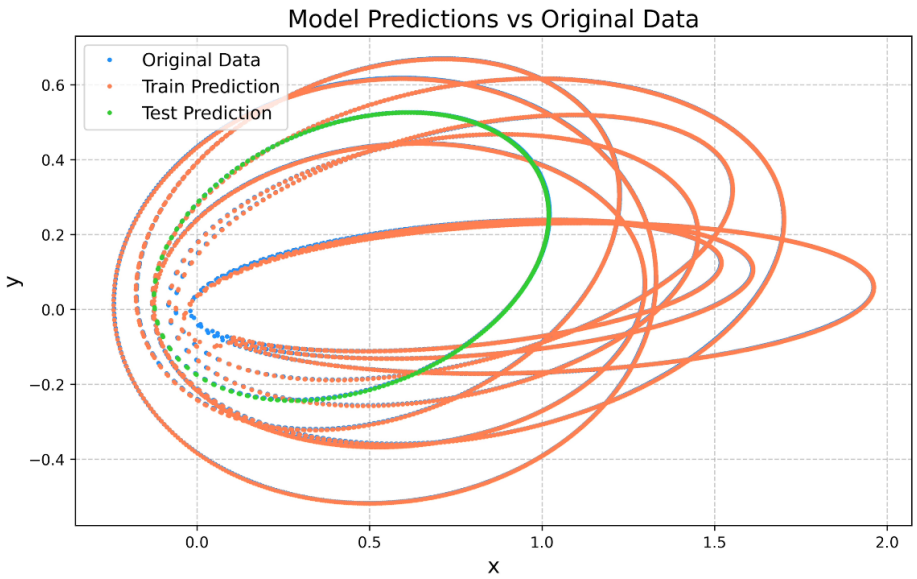
n point masses moving under Newtonian gravitation.

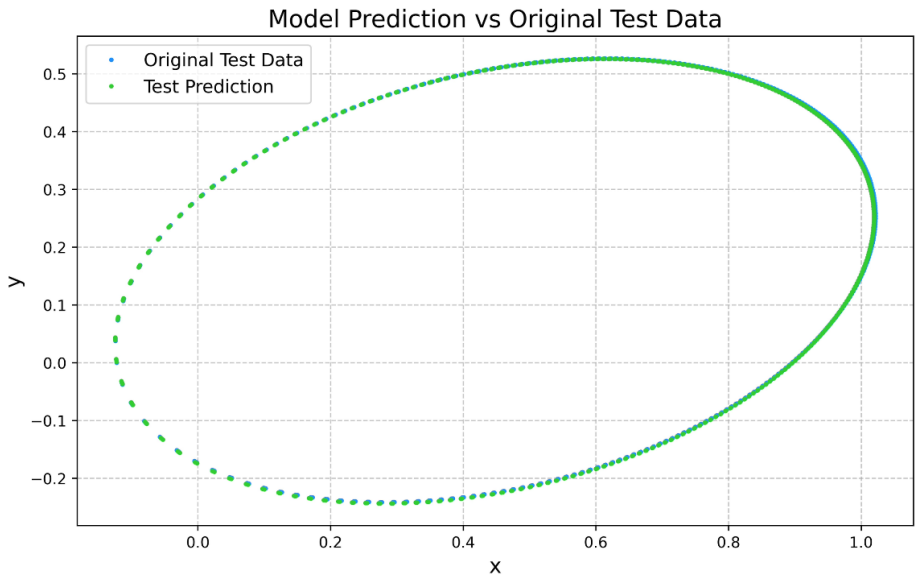
- We have observational data for one star (one body);
- The observational data contains only directions of one star from earth;
- We don't know anything about other bodies, we don't even know how many bodies;
- We don't know anything Newton knew.

Can we, using machine learning, predict future orbits of this star? Can we tell what n is?

A 2-body experiment

Simple scenerio: Observe some trajectories in the configuration space of the 2-body problem, predict new trajectories





What if the only input data is for x-coordinate? It works too, but we need to know why first.

Space-Time Swap

- Given a dynamical system

$$\mathbf{x}^1 = \mathbf{f}(\mathbf{x}^0), \quad \mathbf{x} = (x_1, x_2, x_3, \ldots, x_N)$$

- The only observable quantity is x_1, and we know nothing about the dynamics, not even the dimension!
- Can we recover the system from the data:

$$x_1^0, x_1^1, x_1^2, \ldots ?$$

- Key: **Space-Time Swap!**

Dynamical Systems

$$
\begin{aligned}
x_1^1 &= f_1(x_1^0, \ldots, x_N^0) = f_1(\mathbf{x}^0) \\
x_1^2 &= f_1(x_1^1, \ldots, x_N^1) = f_1(\mathbf{f}(\mathbf{x}^0)) \\
&\ldots \\
x_1^{N-1} &= f_1(\mathbf{f}^{N-2}(\mathbf{x}^0))
\end{aligned}
$$

- If certain conditions (implicit function theorem) are satisfied, we can solve $x_2^0, \ldots, x_N^0$ from $x_1^1, x_1^2, \ldots, x_1^{N-1}$
- This implies that, there is a function h, such that

$$x_1^N = h(x_1^0, x_1^1, \ldots, x_1^{N-1})$$

- In this case, x_1^N is uniquely determined by $x_1^0, x_1^1, \ldots, x_1^{N-1}$.
- We have a new dynamical system H:

$$(x_1^0, x_1^1, \ldots, x_1^{N-1}) \mapsto (x_1^1, x_1^2, \ldots, x_1^N)$$

- We have a new dynamical system H:

$$(x_1^0, x_1^1, \ldots, x_1^{N-1}) \mapsto (x_1^1, x_1^2, \ldots, x_1^N)$$

- The dynamical system H is topologically conjugate to f, it captures all the information from f.
- What if the implicit function condition is not globally satisfied?

Taking more data!

$$
\begin{aligned}
x_1^1 &= f_1(x_1^0, \ldots, x_N^0) = f_1(\mathbf{x}^0) \\
x_1^2 &= f_1(x_1^1, \ldots, x_N^1) = f_1(\mathbf{f}(\mathbf{x}^0)) \\
&\ldots \\
x_1^K &= f_1(\mathbf{f}^{K-1}(\mathbf{x}^0))
\end{aligned}
$$

- There are K equations, $N-1$ unknowns.
- Locally, if the derivative (an $(N-1) \times K$ matrix) of the above equation has rank $N-1$, then $x_2^0, \ldots, x_N^0$ can be solved.
- Assume that there is a global solution for $x_2^0, \ldots, x_N^0$, then the original system can be embedded in the dynamical system:

$$
(x_1^0, x_1^1, \ldots, x_1^K) \mapsto (x_1^1, x_1^2, \ldots, x_1^{K+1})
$$

- The rank condition can be satisfied if x_1 is "generic" and $K \geq (2N+1)$ (Whitney's Embedding Theorem).

Takens' Embedding Theorem

Theorem

Let M be a smooth manifold of dimension N and let

$$f : M \to M$$

be a diffeomorphism. Fix any $K \geq 2N + 1$, let g be generic smooth function

$$g : M \to \mathbb{R}.$$

Then the map $\phi : M \to \mathbb{R}^K$ defined by

$$\phi(x) = \left(g(x), g(f(x)), \ldots, g(f^{K-1}(x))\right)$$

is an embedding of M.

Takens' theorem is used to reconstruct strange attractors.

Theorem (Space-Time Swap)

Let M be a smooth manifold of dimension N and let

$$f : M \to M$$

be a diffeomorphism. Assume that the dimension of the fixed points of f^k for $k < N$ is less than k. Then there is an integer $N \leq K \leq 2N + 1$, such that for any function

$$g_0 : M \to \mathbb{R}.$$

there is a function g, C^1-close to g_0 and a function h on a subset of $\mathbb{R}^K$ such that

$$g(f^K(x)) = h(g(x), g(f(x)), \ldots, g(f^{K-1}(x)))$$

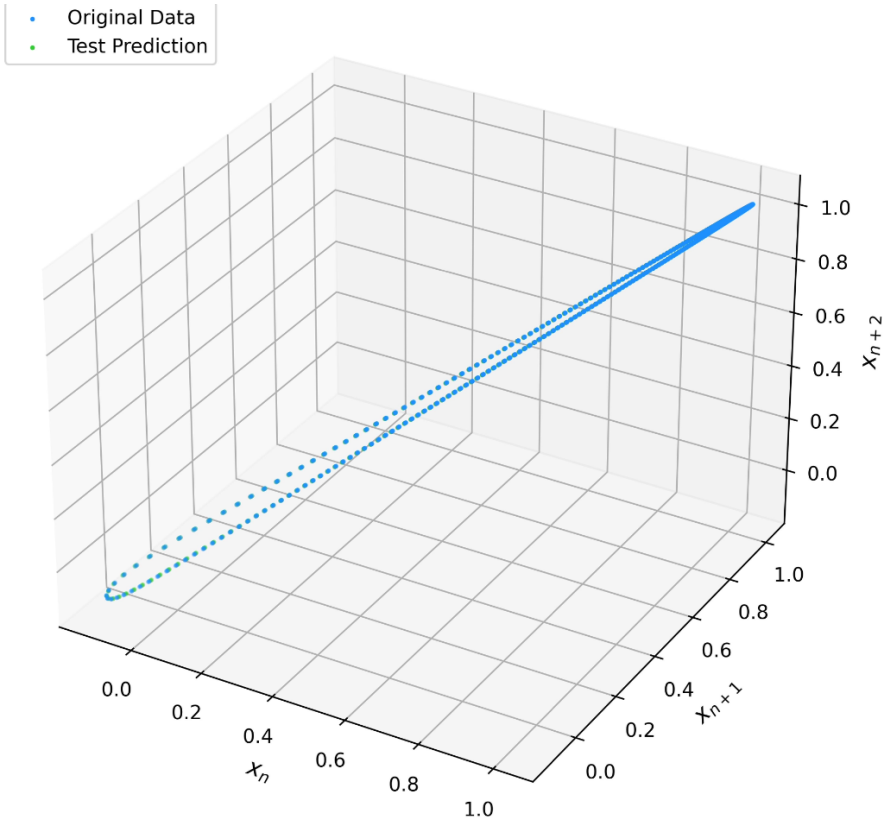
for all $x \in M$.

Objective of machine learning: find the function h.

XNet is the best algorithm to find such function h!

Another 2-body experiment

Simple scenerio: 2-body problem, observing only x-coordinate, i.e., the function g is just the first coordinate, predict new trajectories



The orbit is still a circle, but different geometry.

Thank You!!!