Symbolic Gradients: Retrieving Physical Concepts Learned by Artificial Neural Networks

ECT*, Trento, Sept 11, 2026

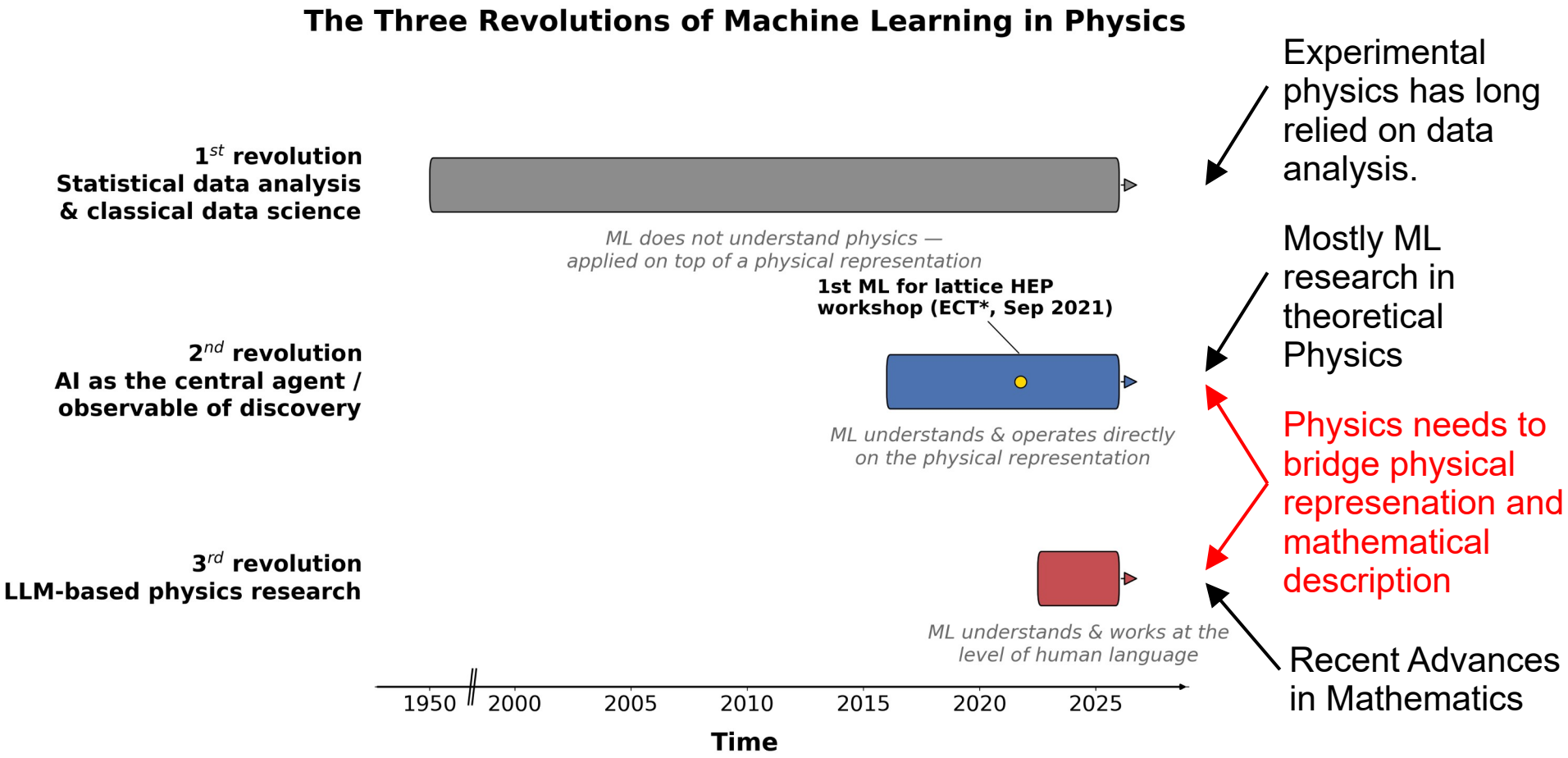
Sebastian Johann Wetzel

Overview



Happy 10 year anniversary of the 2nd Machine Learning Revolution in our field.

Overview

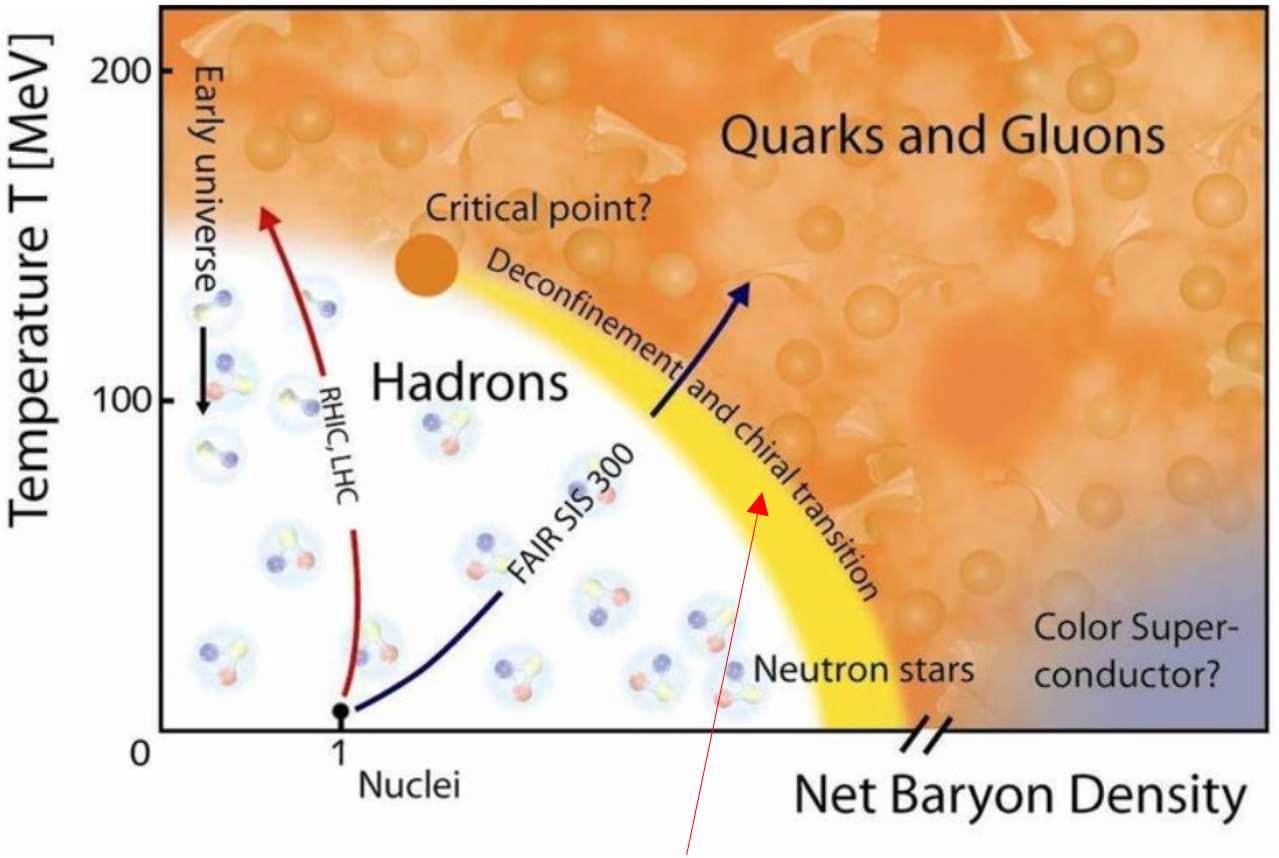

The Three Revolutions of Machine Learning in Physics
1st revolution
Statistical data analysis
& classical data science
ML does not understand physics —
applied on top of a physical representation
1st ML for lattice HEP
workshop (ECT*, Sep 2021)
2nd revolution
AI as the central agent /
observable of discovery
ML understands & operates directly
on the physical representation
3rd revolution
LLM-based physics research
ML understands & works at the
level of human language
1950
2000
2005
2010
2015
2020
2025
Time
Experimental physics has long relied on data analysis.
Mostly ML research in theoretical Physics
Physics needs to bridge physical represenation and mathematical description
Recent Advances in Mathematics

Phase Diagrams
Confinement

Machine Learning of Explicit Order Parameters: From the Ising Model to SU(2) Lattice Gauge Theory

Sebastian J. Wetzel[1] and Manuel Scherzer[1]

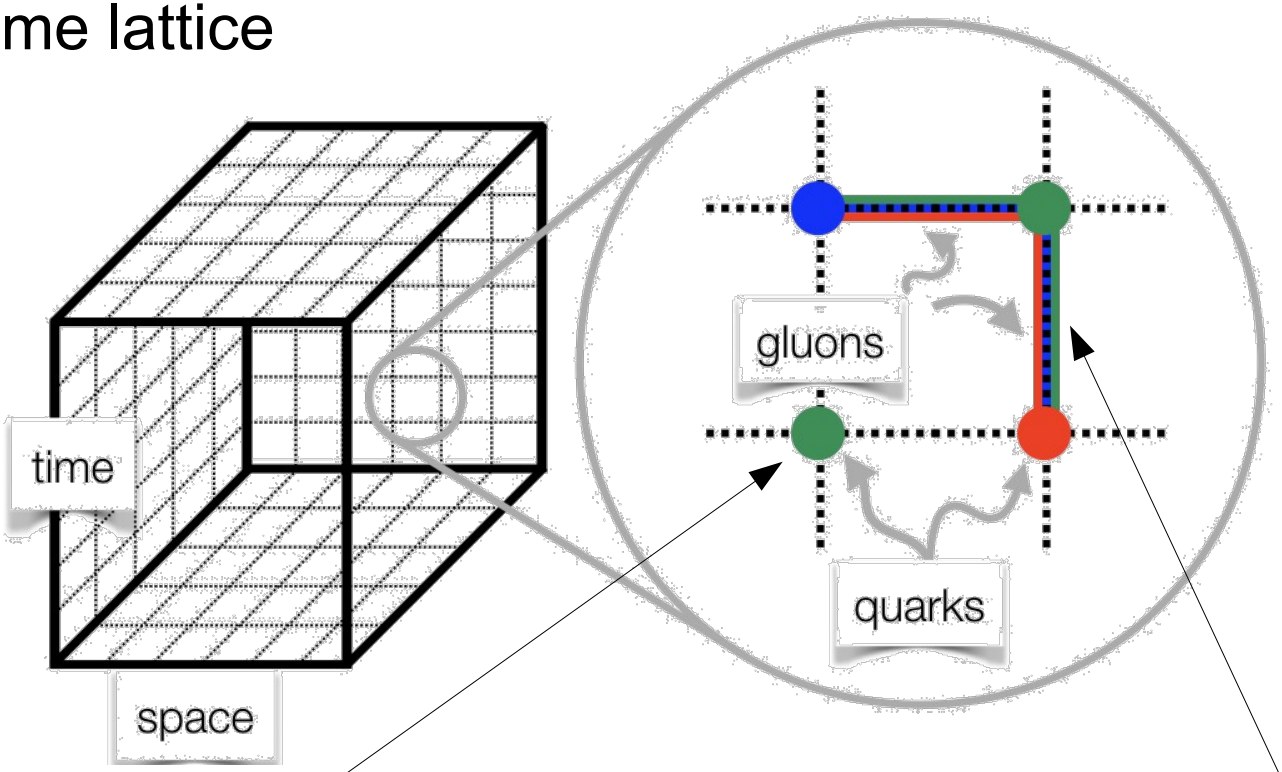
Phase Diagrams
Confinement



Confinement-Deconfinement
Phase Transition

SU(2) Lattice Gauge Theory

Space time lattice



Quarks on heavy static lattice sites.

Gluons on the connections between lattice sites are described by Matrices

~~$U_\mu^x \in SU(3)$~~

$U_\mu^x \in SU(2)$

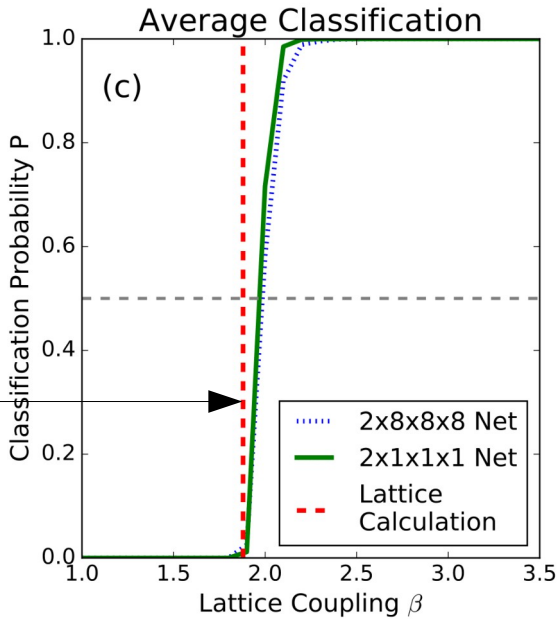
Supervised Learning
SU(2) Lattice Gauge Theory

$$U_{\mu}^{x} U_{\nu}^{x+\hat{\mu}} U_{-\mu}^{x+\hat{\mu}+\hat{\nu}} U_{-\mu}^{x+\hat{\nu}}$$

$$S_{\mathrm{Wilson}}[U] = \beta_{\mathrm{latt}} \sum_{x} \sum_{\mu<\nu} \mathrm{Re\ tr}\left(1 - U_{\mu\nu}^{x}\right)$$

- Train Neural Network at well known points in phase diagram
- Labels: Phase

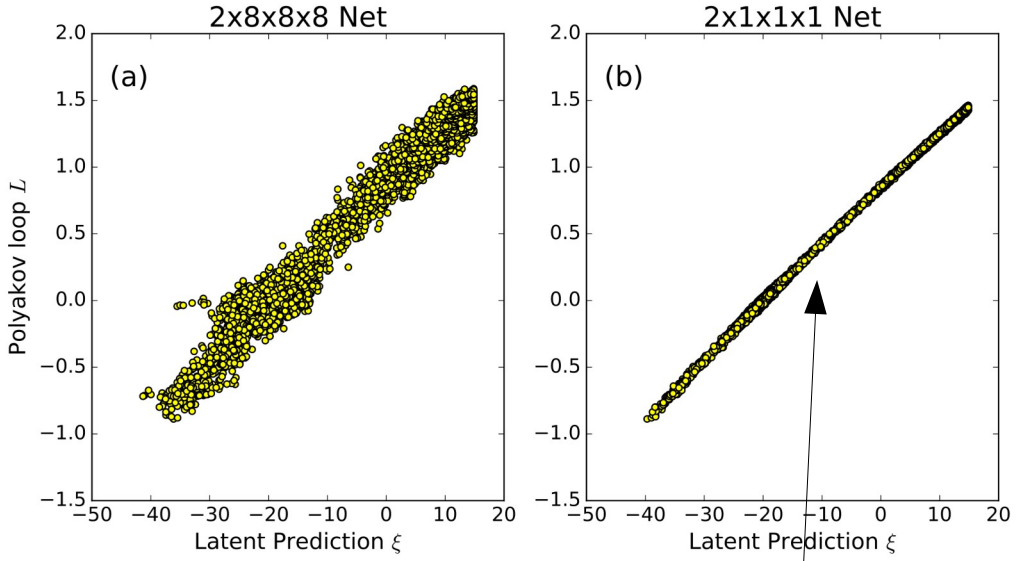
Find phase transition close to lattice calculation



Machine Learning of Explicit Order Parameters: From the Ising Model to SU(2) lattice gauge theory
Wetzel, Scherzer, PRB 2017

Interpretation of Neural Network
SU(2) Lattice Gauge Theory

Deduction confirmed by perfect correlation between NN output and Polyakov Loop order parameter

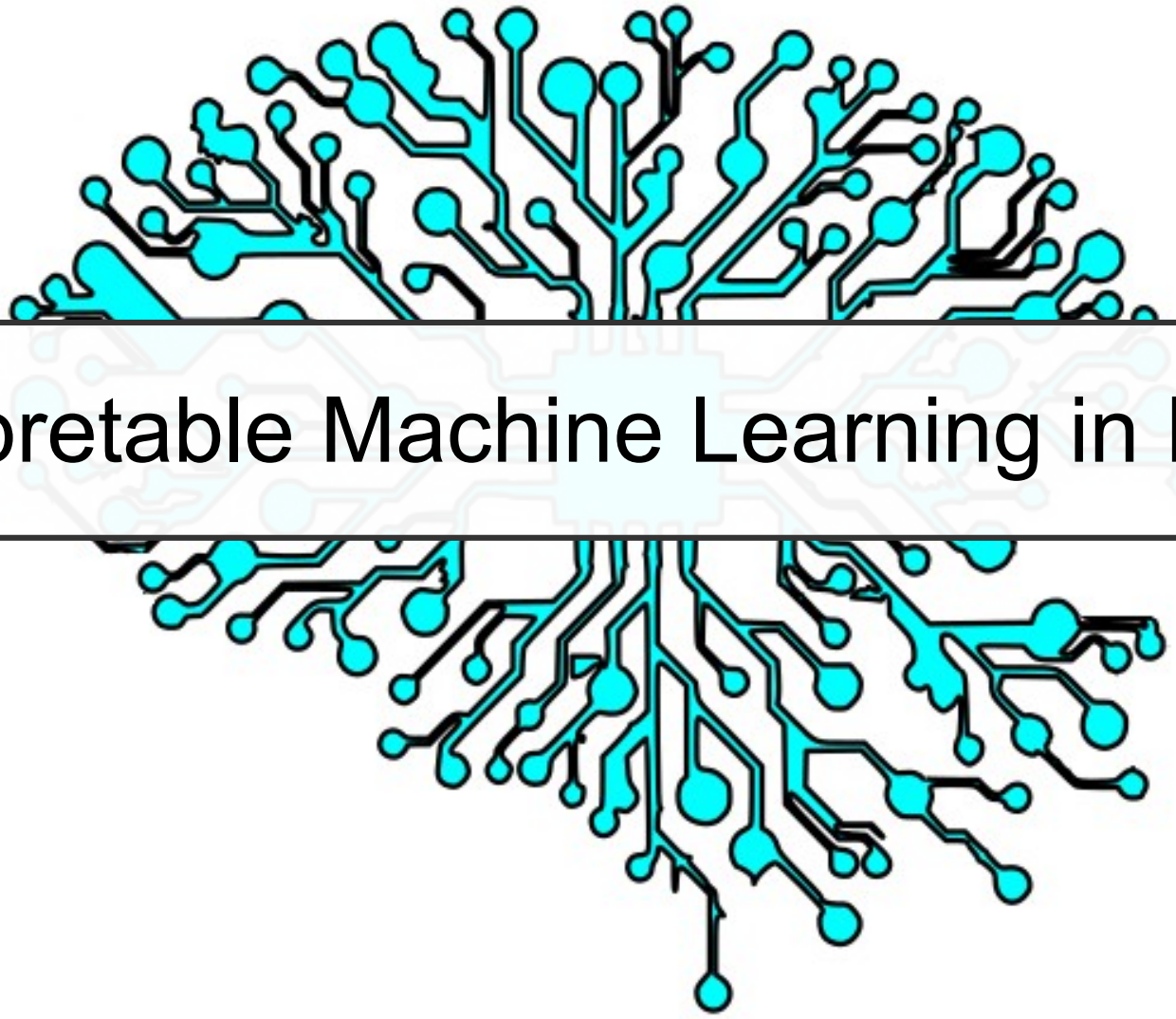


$$F(S) \approx \text{sigmoid}\left(w\left(\frac{2}{N}\sum_{\vec{x}} f(\{U_\mu^{x_0,\vec{x}}\})\right) + b\right)$$

$$f(\{U_\mu^{x_0}\}) = a_\tau^0 a_\tau^1 - b_\tau^0 b_\tau^1 - c_\tau^0 c_\tau^1 - d_\tau^0 d_\tau^1 = \text{tr}\left(U_\tau^0 U_\tau^1\right)$$

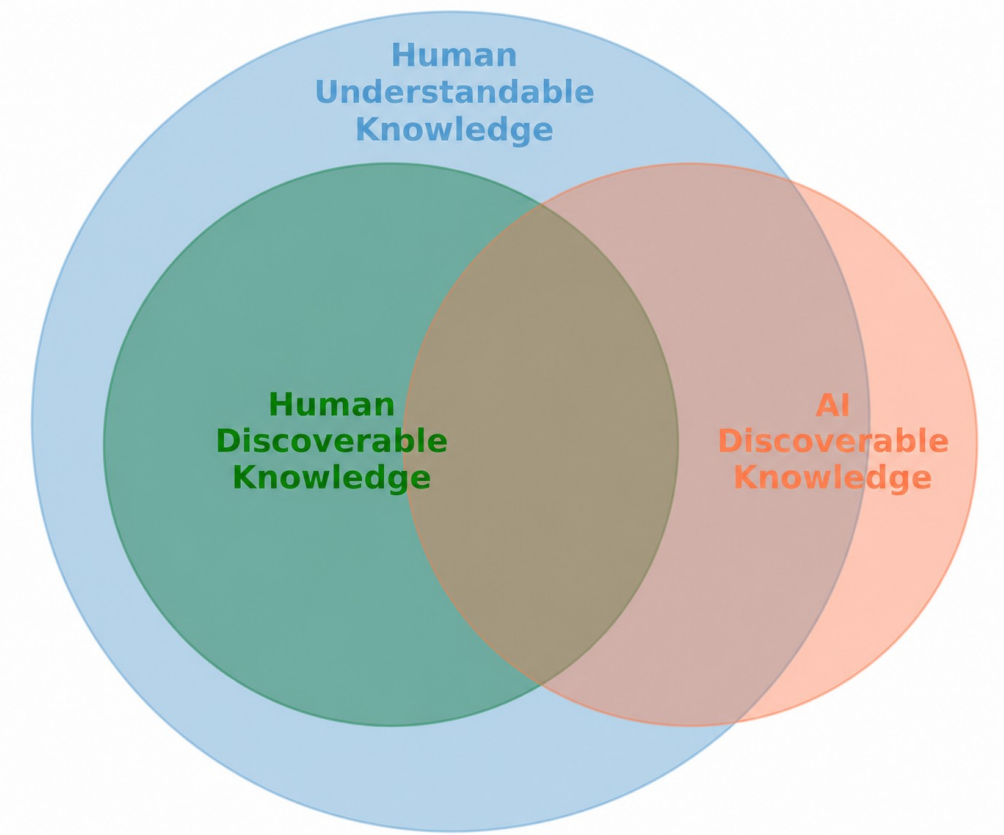
- Polyakov Loop

Interpretation only works for some cases when this line is straight



Interpretable Machine Learning in Physics

Overview



AI is expected to drive scientific discoveries beyond the capabilities of human scientists. In science, the primary goal is to facilitate a human understanding of new and unknown concepts. To bridge the gap between AI-generated insights and human understanding, it is crucial to interpret AI systems.

Interpretation Categories

Interpreting an artificial neural network means formulating A mapping between an abstract mechanism or encoding into a domain that a human can understand.

- ✗ Mechanistic vs. Functional
- ✗ Local vs. Global
- ✗ Verify vs. Discover
- ✗ Low-Level vs. High-Level Features
- ✗ Intrinsic vs Post-hoc
- ✗ Understanding Model vs Understanding Data

Overview

- In order to achieve a full discovering interpretation of neural networks for (scalar) formulas we need to overcome three obstacles:

Symbolic representation

$$y = 2x_1 + x_2^2$$

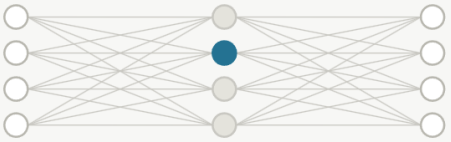
PROBLEM

Neuron activations are continuous, sub-symbolic vectors with no native mapping to a discrete equation.

SOLUTION

Symbolic regression, symbolic search, and networks built from symbolic architectures

Entanglement



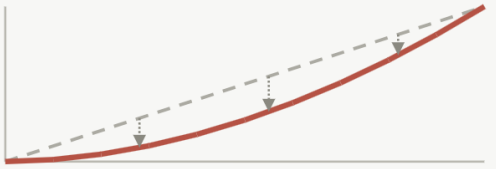
PROBLEM

A single latent axis rarely isolates one generative factor; true factors get mixed together.

SOLUTION

Bottleneck architectures and disentanglement methods, such as β-VAE and sparse autoencoders

Deformation



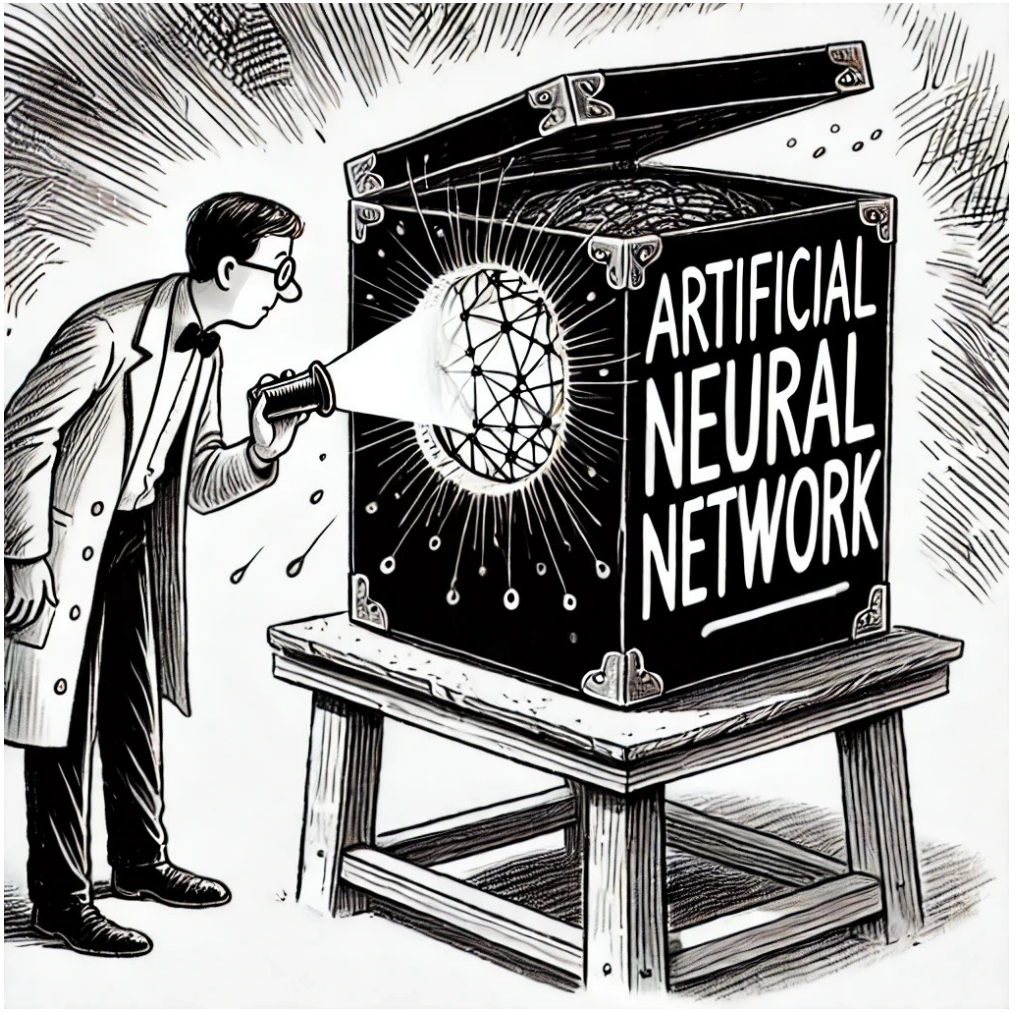
PROBLEM

Even an isolated concept is warped by an unknown, nonlinear, invertible transformation $f(x) = \varphi(g(x))$.

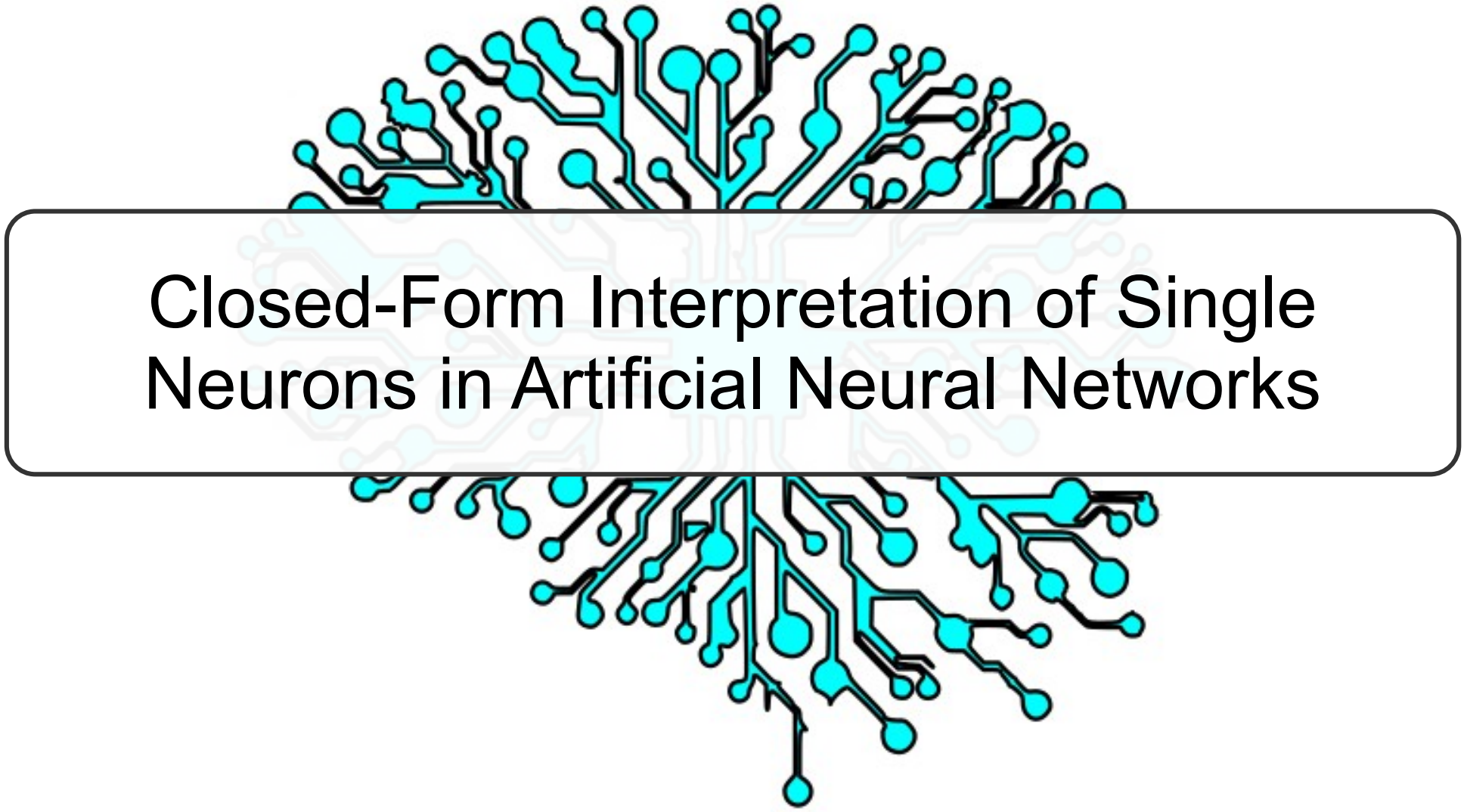
SOLUTION

?

Review

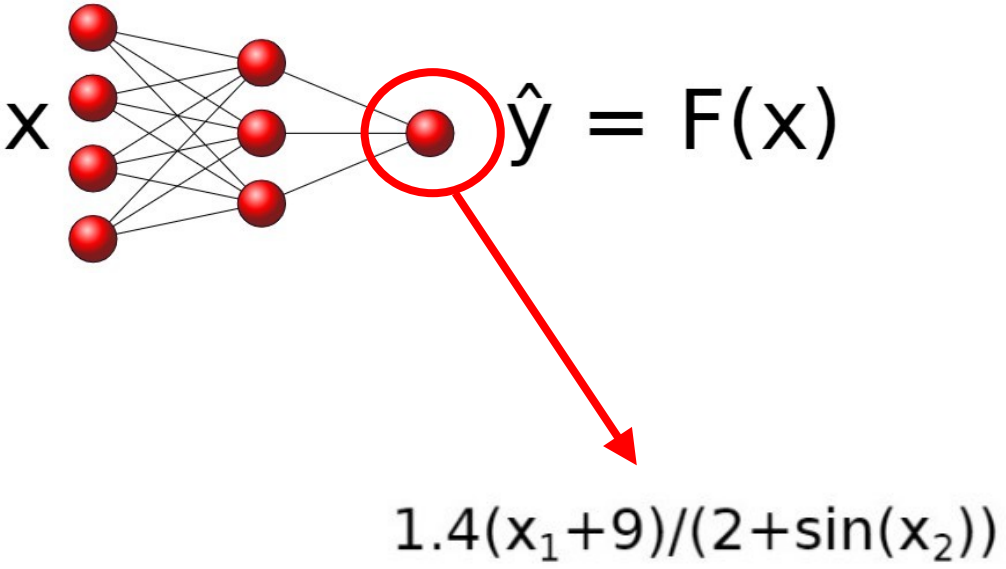


Interpretable Machine Learning in Physics: A Review,
Wetzel, Ha, Iten, Klopotek, Liu
https://arxiv.org/abs/2503.23616



Closed-Form Interpretation of Single Neurons in Artificial Neural Networks

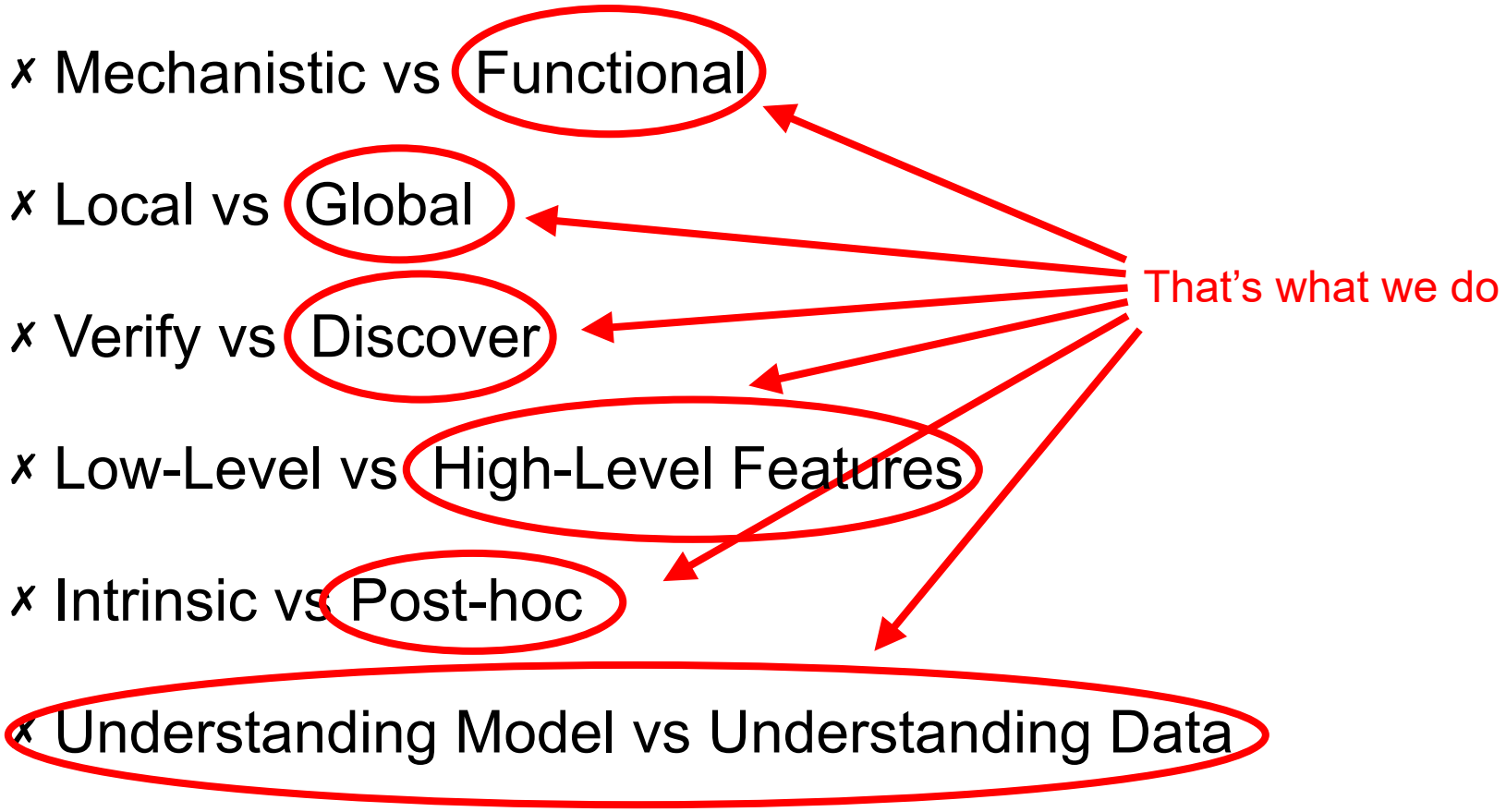
Overview



- Extract Concepts encoded in Neurons of Artificial Neural Networks in the form of human-readable equations.

Overview

Interpreting an artificial neural network means formulating A mapping between an abstract mechanism or encoding into a domain that a human can understand.



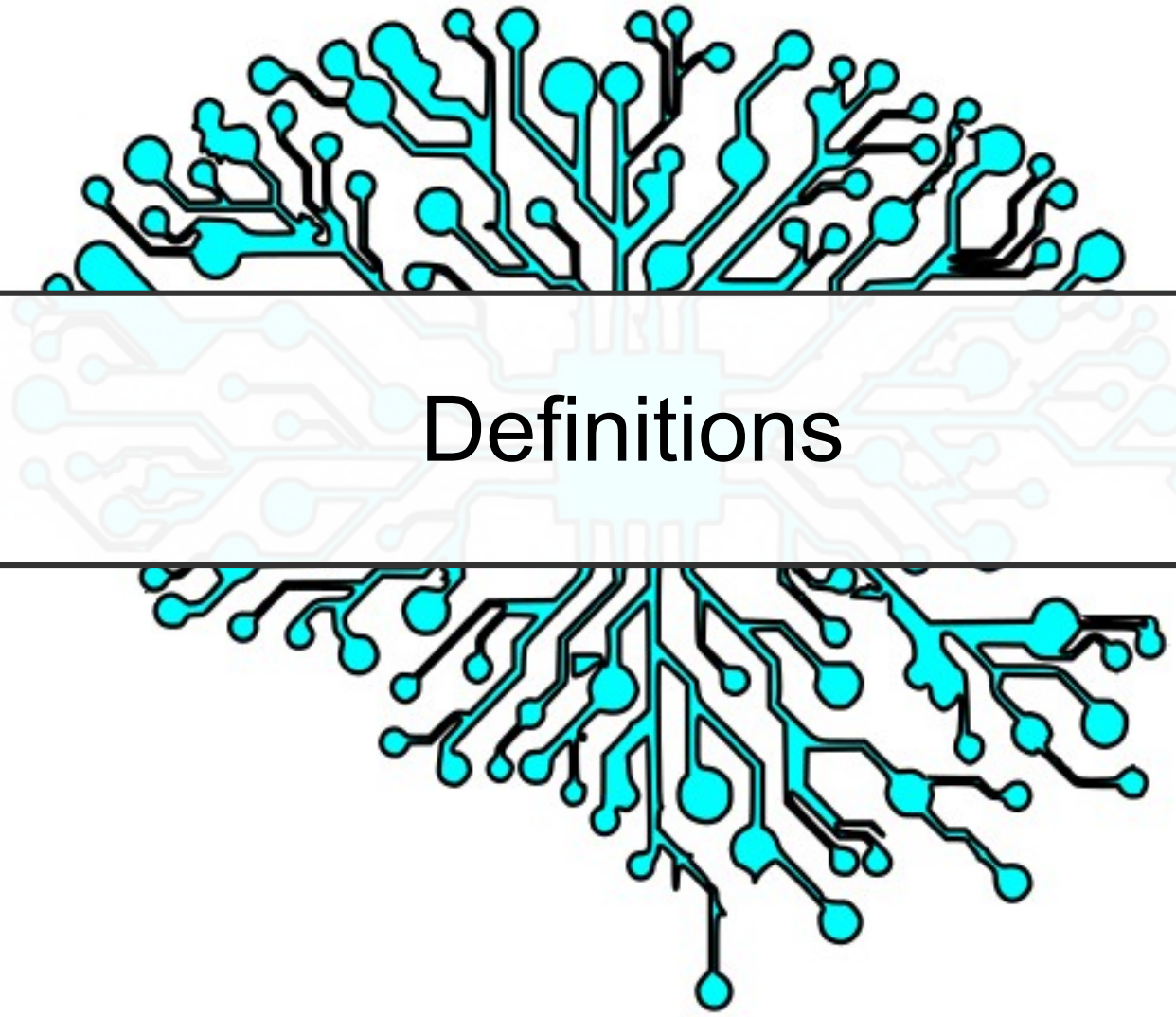
Overview

✗ Theoretical Framework
- Neural Networks Learn Meaningful Concepts in a Form unreadable for Humans
- Equivalence Class – Redefinition of “Concept“
- Interpretation Algorithm

✗ Experiments
- Recover Eqs from Neural Network Classifiers
- Recover Invariants and Conserved Quantities from Neural Network Latent Spaces

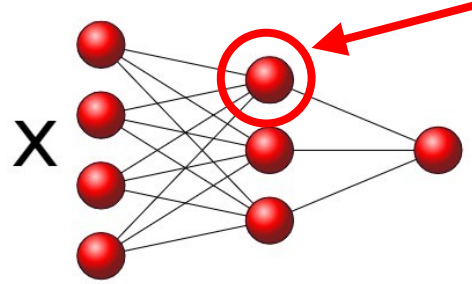
Closed-Form Interpretation of Neural Network Classifiers with Symbolic Gradients, Wetzel, MLST 2025

Closed-Form Interpretation of Neural Network Latent Spaces with Symbolic Gradients, Wetzel, Patel, arXiv:2409.05305



Definitions

Artificial Neural Network



Neuron / Perceptron

Input Variables

Neural Network

$$\hat{\mathbf{y}} = F(\mathbf{x}) = (L^{(l)} \circ \ldots \circ L^{(0)})(\mathbf{x})$$

Layers

Weight Matrix

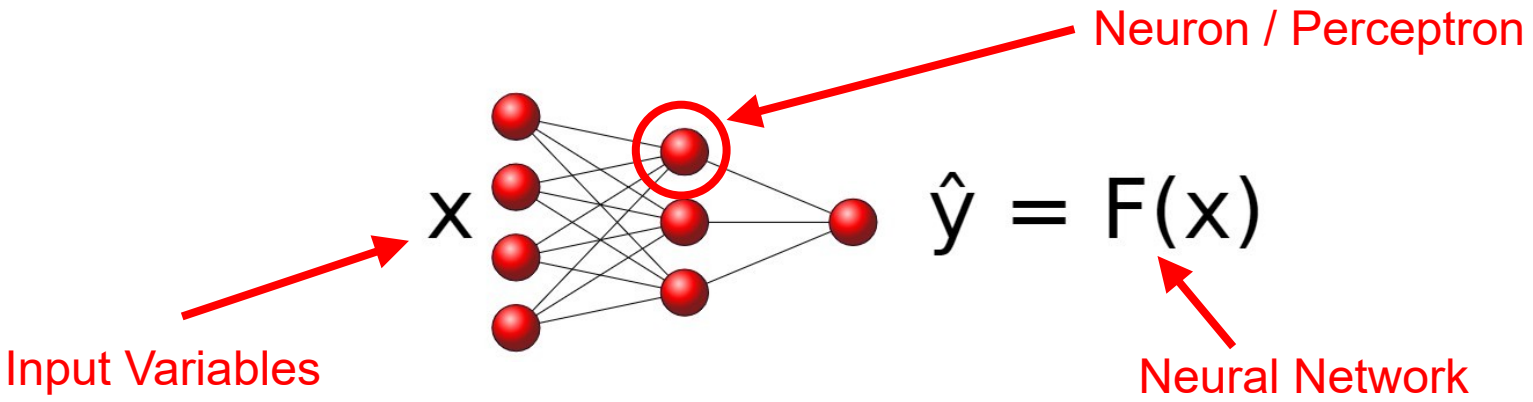
Bias Vector

$$\mathbf{a}^{(l)} = L^{l}(\mathbf{a}^{(l-1)}) = \sigma^{(l)}(\mathbf{W}^{(l)}\mathbf{a}^{(l-1)} + \mathbf{b}^{(l)})$$

$$\mathbf{a}^{(0)} = \mathbf{x} \in \mathbb{R}^{n}$$

Nonlinear/Nonpolynomial Activation Function

Artificial Neural Network



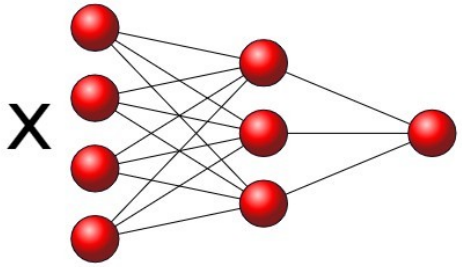
$$\hat{\mathbf{y}} = F(\mathbf{x}) = (L^{(l)} \circ \ldots \circ L^{(0)})(\mathbf{x})$$

$$F(\mathbf{x}) = \text{sigmoid}(f(\mathbf{x}))$$

Activation Function for Binary Classification

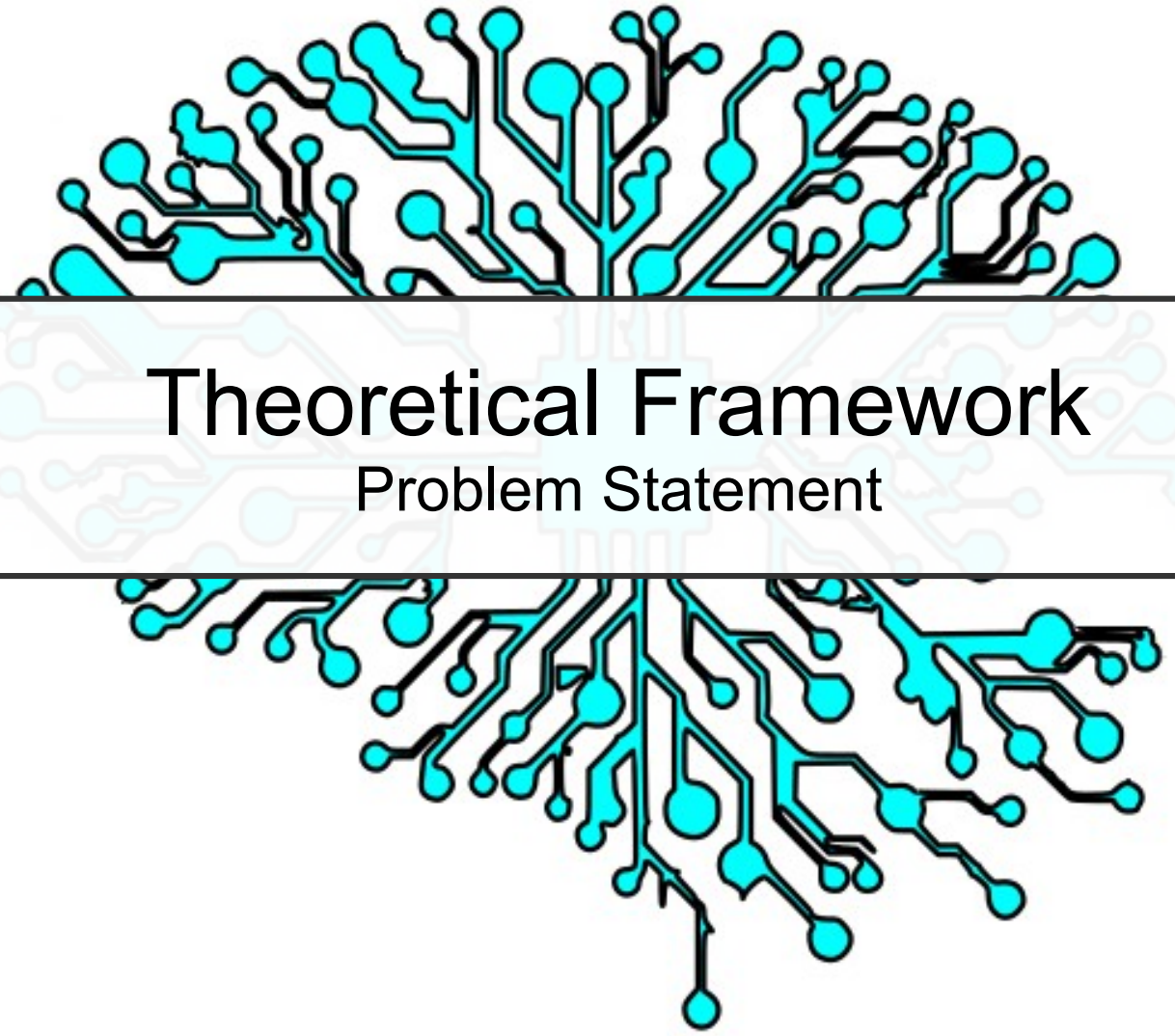
Latent Model

Artificial Neural Network



$$\hat{\mathbf{y}} = F(\mathbf{x}) = (L^{(l)} \circ \ldots \circ L^{(0)})(\mathbf{x})$$

Training: Use Gradient Descent/Backpropagation to adjust the weights and biases of the neural network to minimize an objective function $\mathcal{L}(F(X), Y)$ on a data set (X, Y) to approximate $F(X) \approx Y$ on the data manifold $X \subset D \subset \mathbb{R}^n$

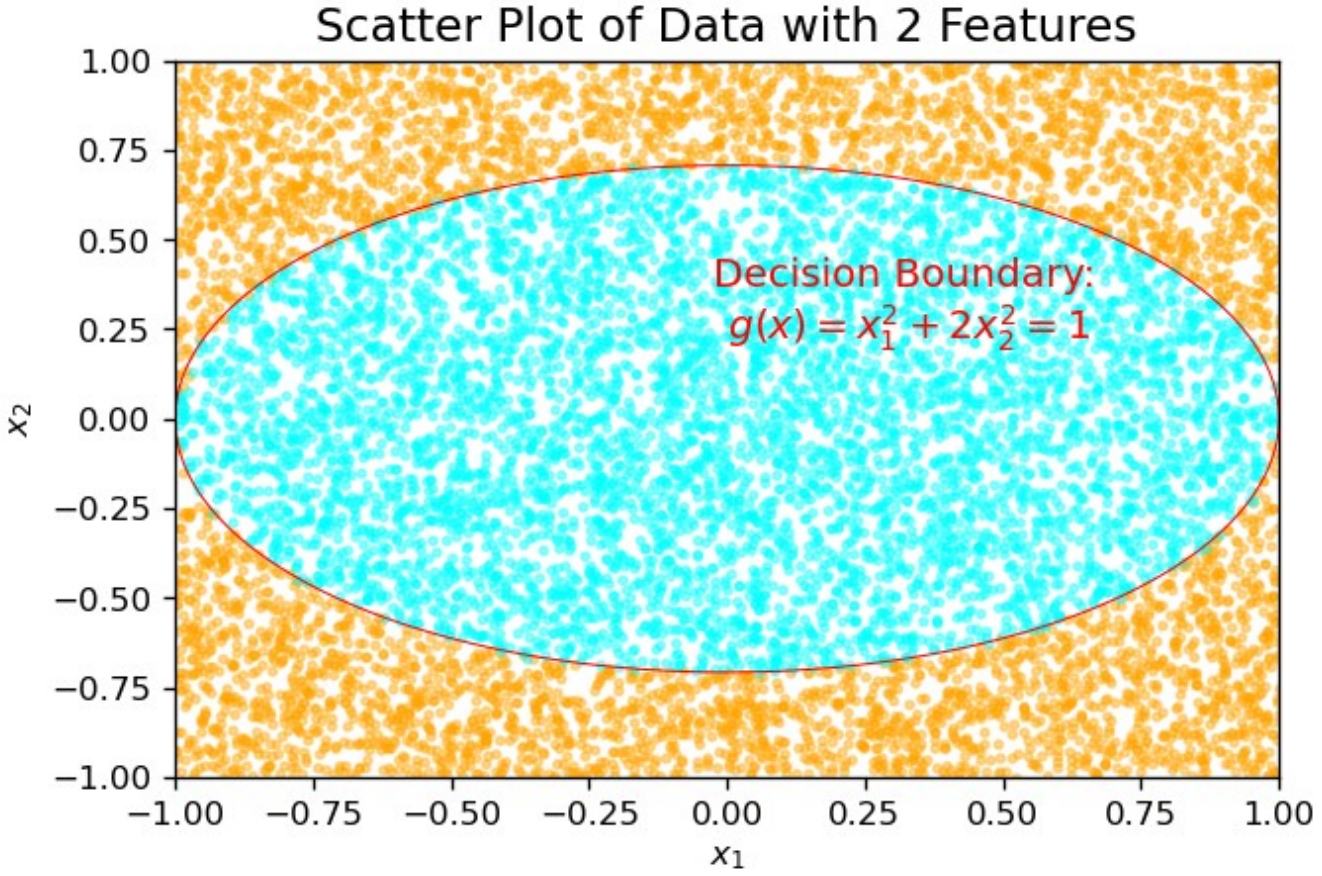


Theoretical Framework

Problem Statement

Binary Classification

Example: Train a Neural Network Classifier to solve a binary Classification problem with decision boundary $x_1^2 + 2x_2^2 = 1$



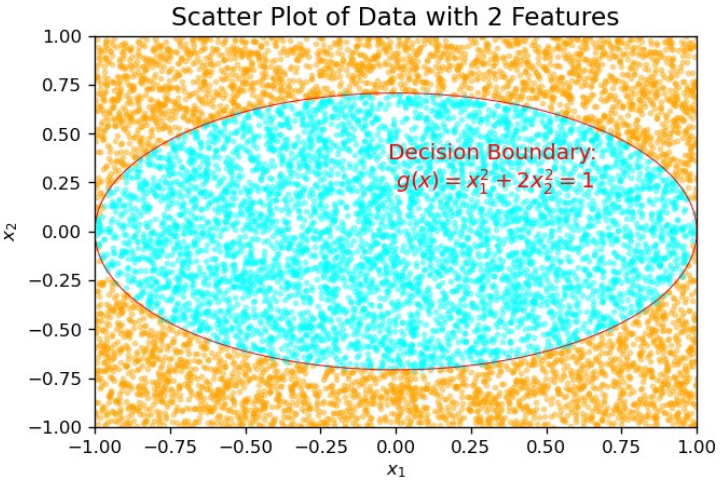
Binary Classification

Question: Assuming we do not know about the decision boundary $x_1^2 + 2x_2^2 = 1$*,can we*

- *Examine the neural network to figure out how it makes decisions*
- *Discover this decision boundary in a form that is understandable by humans (as short equation)?*

Binary Classification

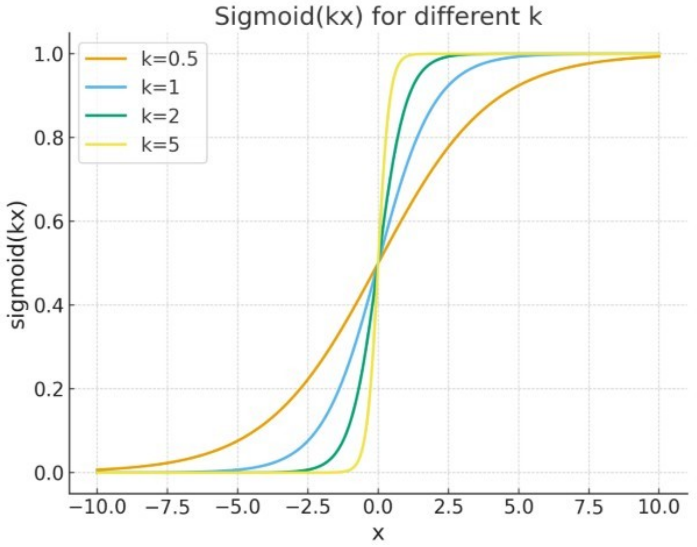
Example: Train a Neural Network Classifier to solve a binary Classification problem with decision boundary $x_1^2 + 2x_2^2 = 1$



Expectation:

$$F(x_1, x_2) = \text{sigmoid}\Big(k\big(x_1^2 + 2x_2^2 - 1\big)\Big)$$

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}}$$



Binary Classification

Example: Train a Neural Network Classifier to solve a binary Classification problem with decision boundary $x_1^2 + 2x_2^2 = 1$

Interpret Neural Network:

inferring the function from the weights is pointless because there is no natural representation beyond linear functions.

Since the latent model is a polynomial equation, let's find the best polynomial approximation to what the neural network learns.

$$F(\mathbf{x}) = \text{sigmoid}(f(\mathbf{x}))$$

Latent Model

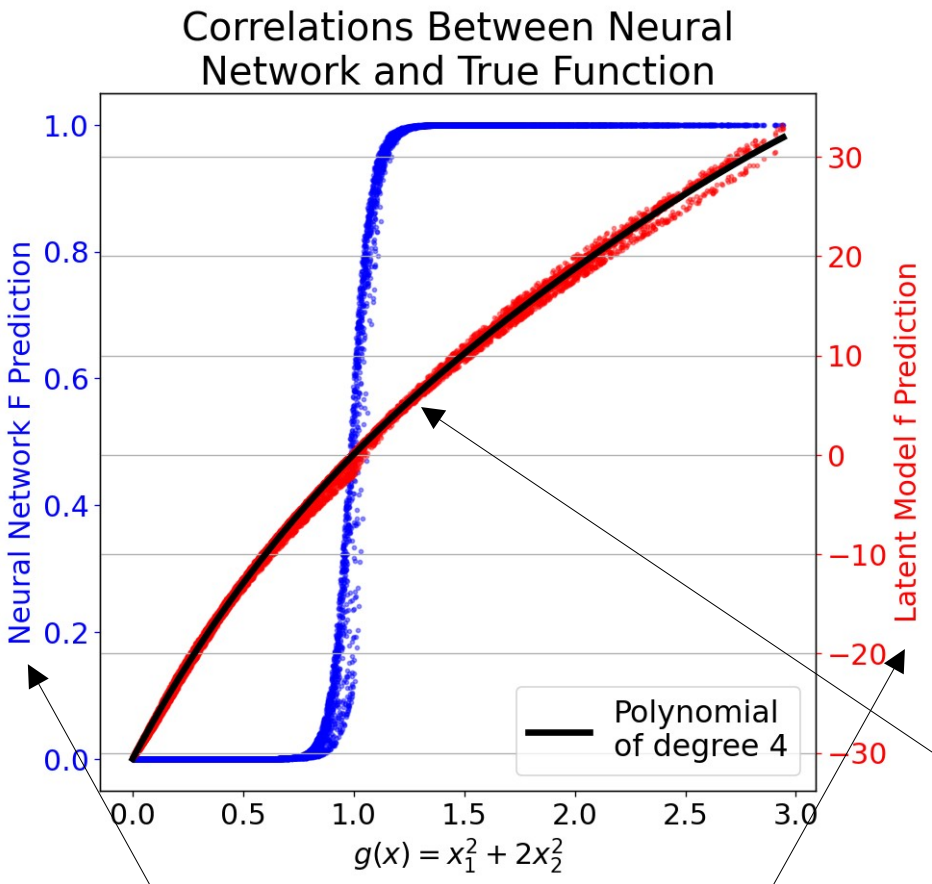
Binary Classification

Neural Network is very well approximated by

$$F(x_1, x_2) = \text{sigmoid}\Big($$
$$-0.44894431\, x_1^8 - 3.59155448\, x_1^6 x_2^2 - 10.77466344\, x_1^4 x_2^4 - 14.36621792\, x_1^2 x_2^6 - 7.18310896\, x_2^8$$
$$+3.87605268\, x_1^6 + 23.25631608\, x_1^4 x_2^2 + 46.51263216\, x_1^2 x_2^4 + 31.00842144\, x_2^6$$
$$+14.45538016\, x_1^4 + 57.82152064\, x_1^2 x_2^2 + 57.82152064\, x_2^4$$
$$+41.66782218\, x_1^2 + 83.33564436\, x_2^2 - 30.5992956\Big)$$

Why can't we recover the decision boundary $x_1^2 + 2x_2^2 = 1$*?*
What went wrong? Can we gain a picture about how neural nets store concepts, how can we retrieve these concepts?

Binary Classification



Observation:
The neuron deforms the concept

$$F(\mathbf{x}) = \text{sigmoid}(f(\mathbf{x}))$$

Binary Classification

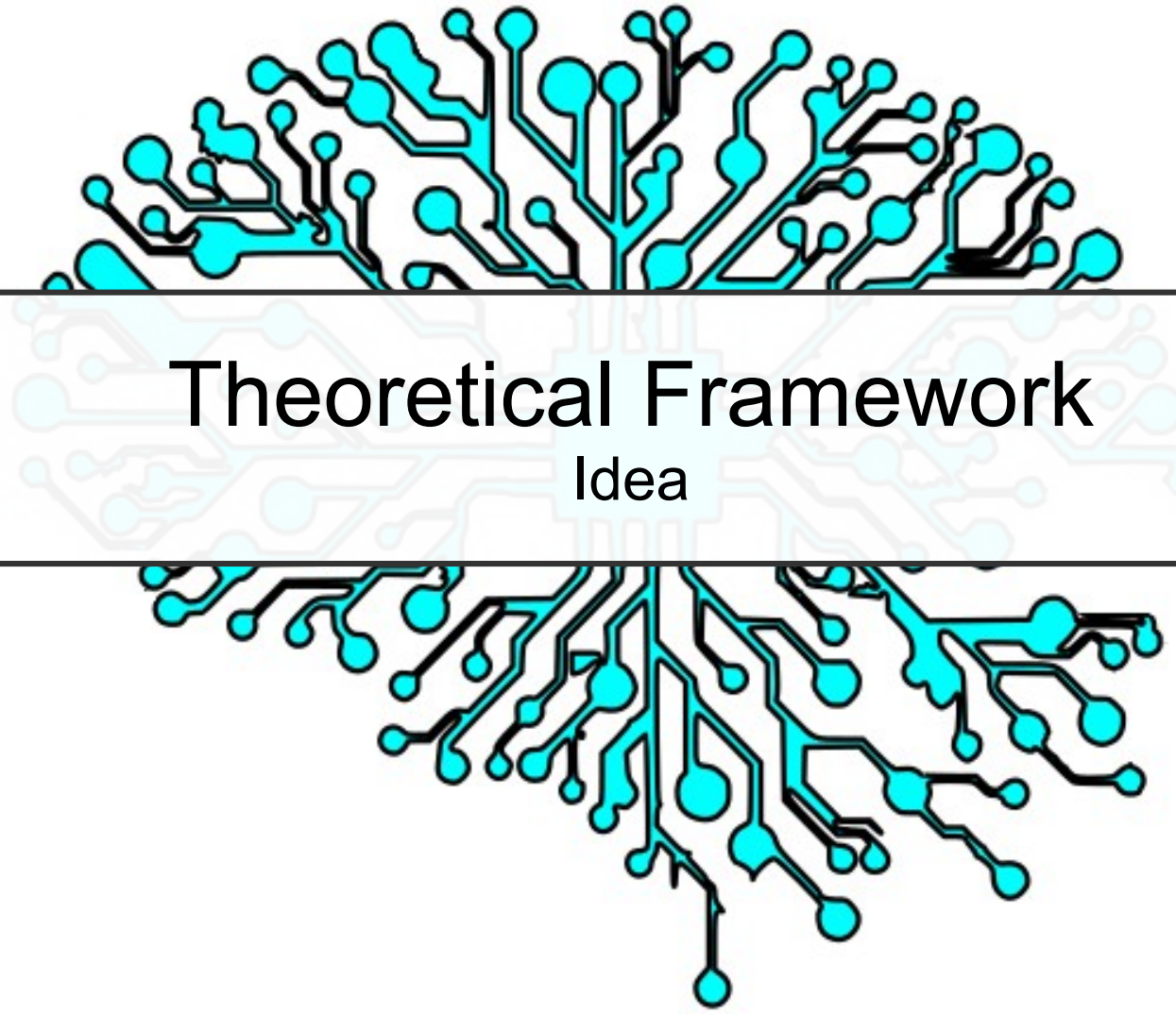
- *The neural network conceals the decision boundary*
- *By looking closely we can disentangle the network*

$$\phi(x) = -0.4489\,x^4 + 3.8761\,x^3 + 14.4554\,x^2 + 41.6678\,x - 30.5993$$

$$g(x_1, x_2) = x_1^2 + 2x_2^2$$

$$F(x_1, x_2) = \text{sigmoid}\big(\phi(g(x_1, x_2))\big)$$

- *Change the paradigm about neural network interpretation, concepts and what information we really want to retrieve.*

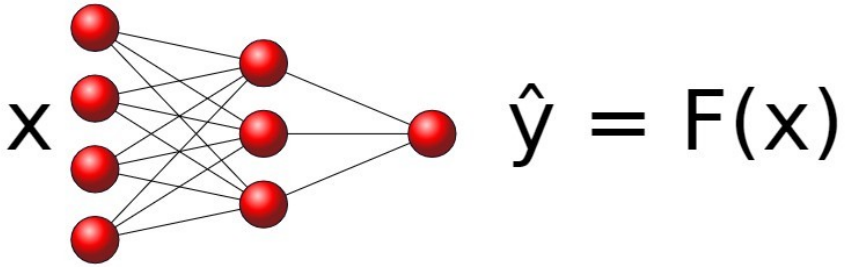


Theoretical Framework

Idea

Idea

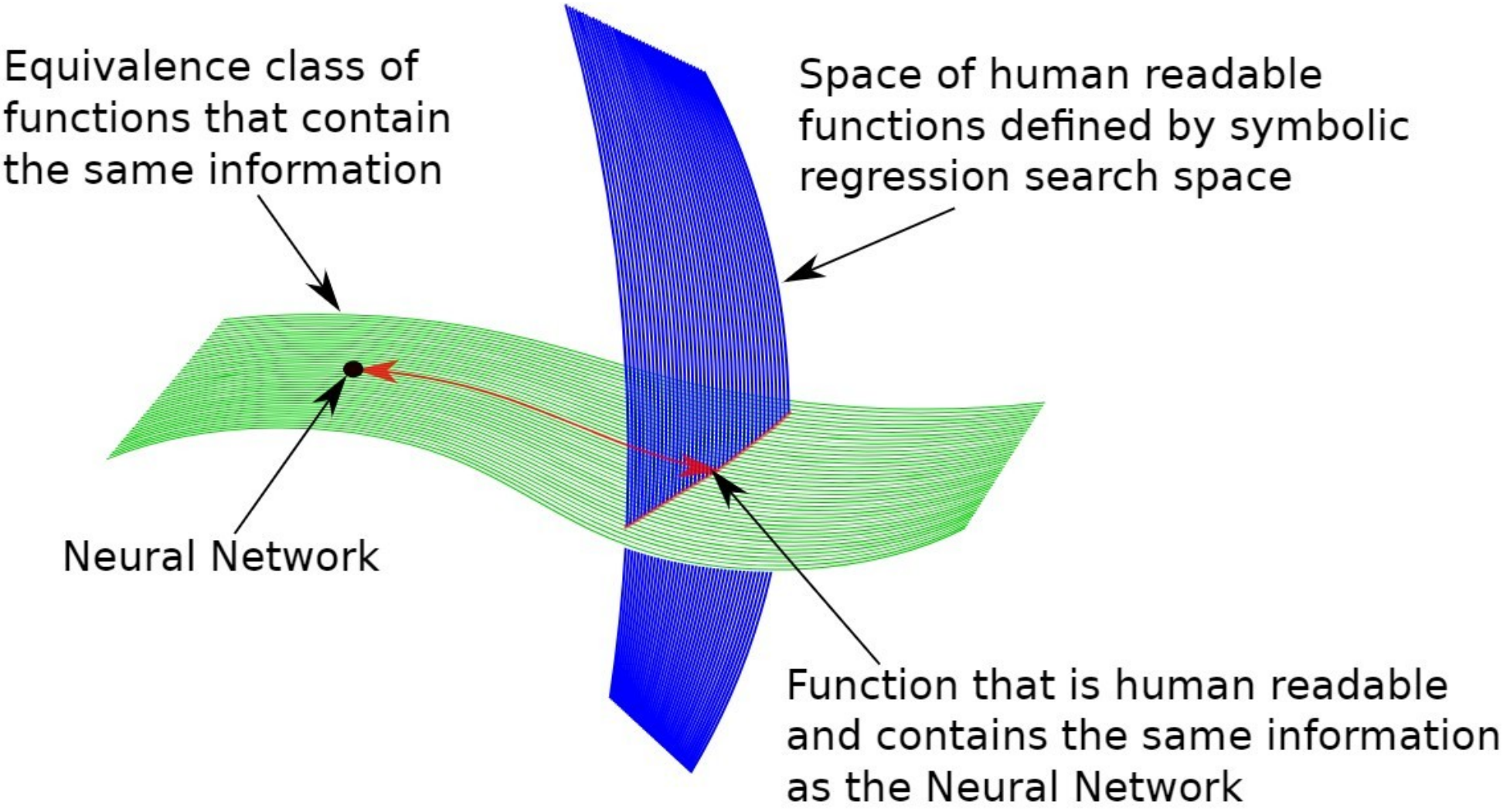
Artificial Neural Network Learns Concepts in a Highly Elusive and Convoluted Manner

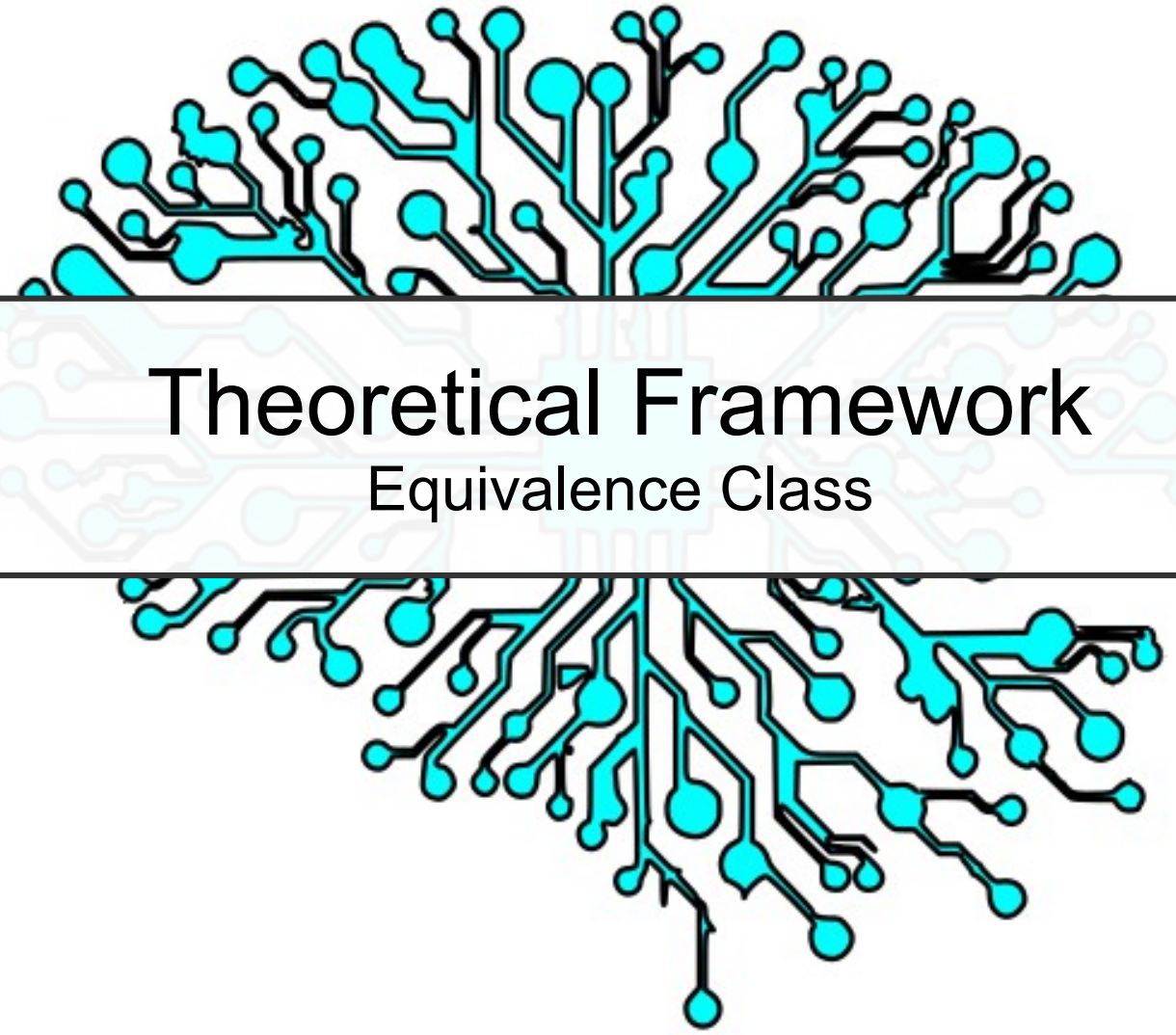


If a Neural Network Learns a Concept that can be written in Human-Readable Form, is it possible to reveal the closed form concept while ignoring any uninterpretable transformations?

$$F(x_1, \ldots, x_n) = \underbrace{\text{sigmoid}}_{\text{activation function}} \left(\underbrace{\phi}_{\text{uninterpretable transformation}} \left(\underbrace{g(x_1, \ldots, x_n)}_{\text{closed form decision function}} \right) \right)$$

Idea


Equivalence class of functions that contain the same information
Space of human readable functions defined by symbolic regression search space
Neural Network
Function that is human readable and contains the same information as the Neural Network

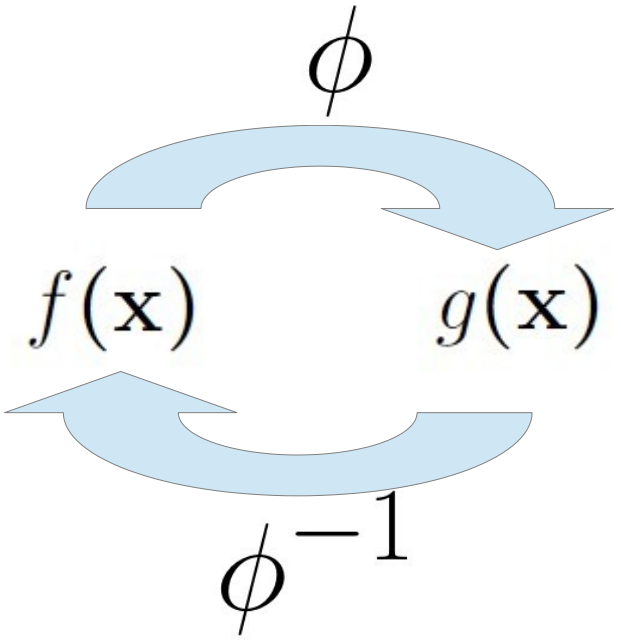


Theoretical Framework

Equivalence Class

Equivalence Class

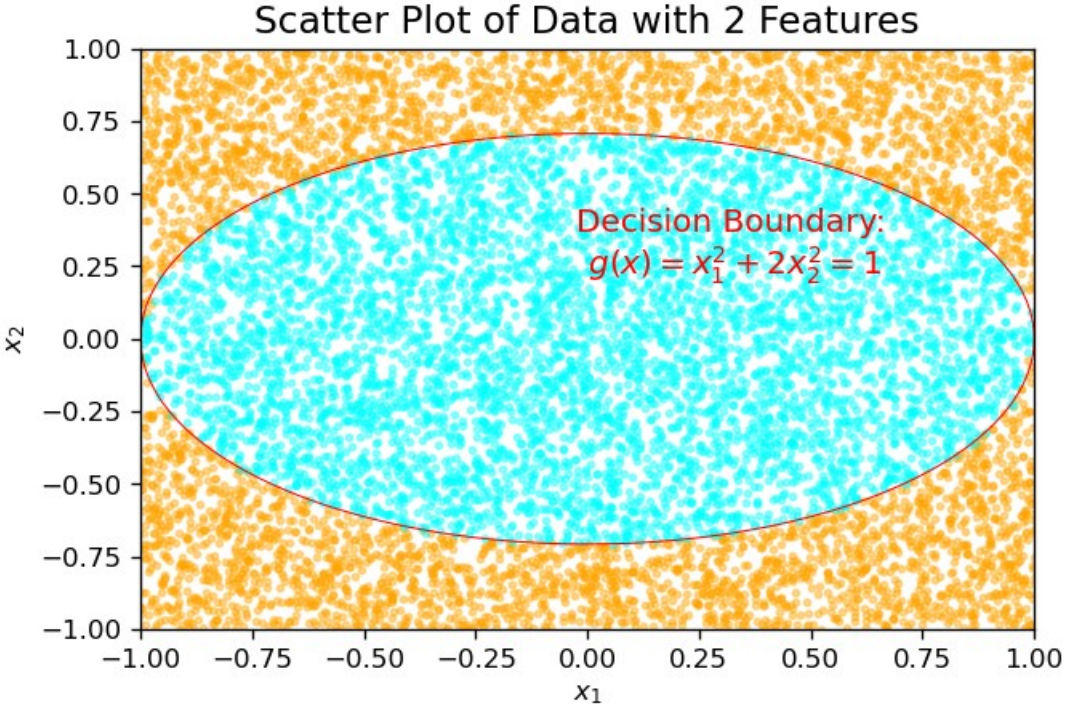
A function $f(\mathbf{x})$ contains the full information about a certain Quantity $g(\mathbf{x})$ if $g(\mathbf{x})$ can be faithfully reconstructed from $f(\mathbf{x})$. Conversely, if $f(\mathbf{x})$ only contains information from $g(\mathbf{x})$ it is possible to reconstruct $f(\mathbf{x})$ from the knowledge of $g(\mathbf{x})$.
In mathematical terms that means that there exists an invertible function such that $f(\mathbf{x}) = \phi(g(\mathbf{x}))$

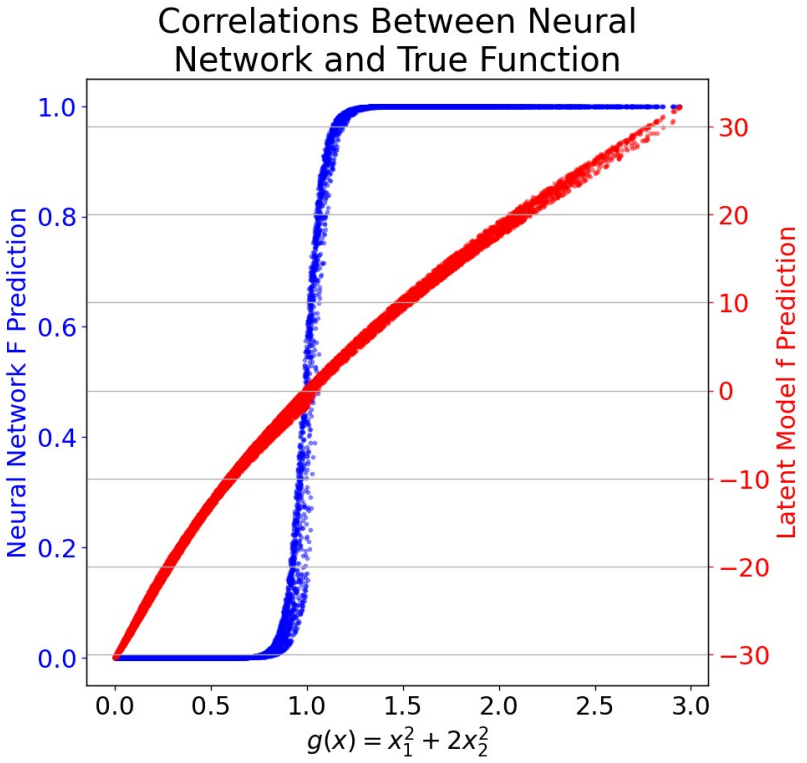


✗ *In the case of a one-dimensional input this is trivial*

Equivalence Class

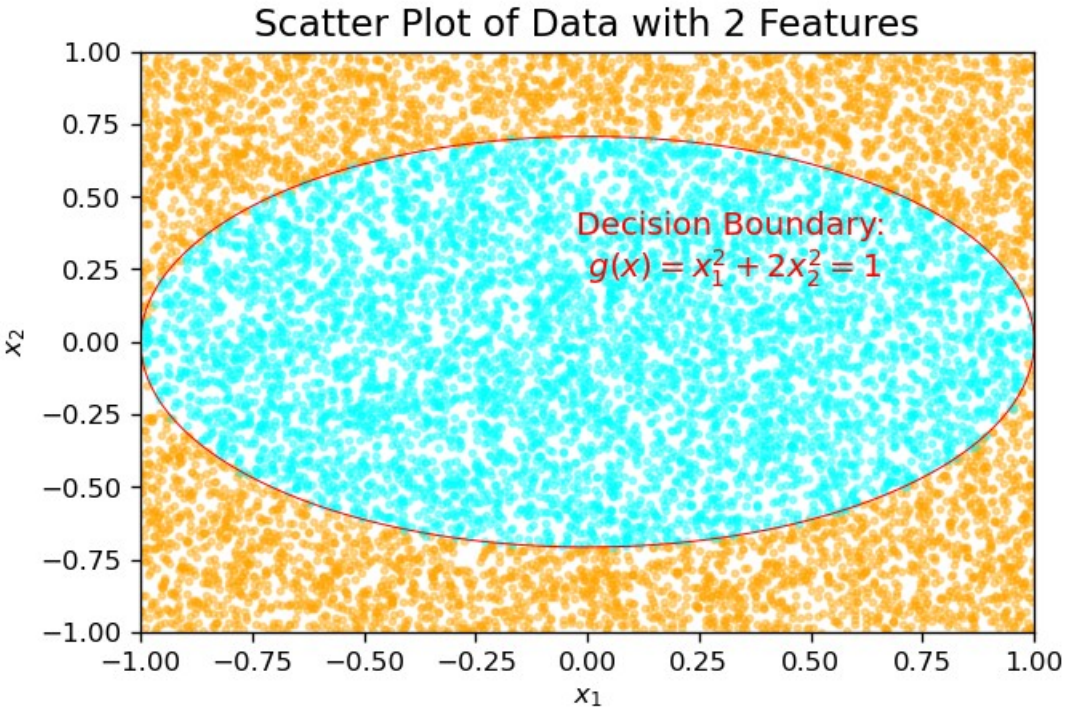
Example: Train a Neural Network Classifier to solve a binary Classification problem with decision boundary $x_1^2 + 2x_2^2 = 1$

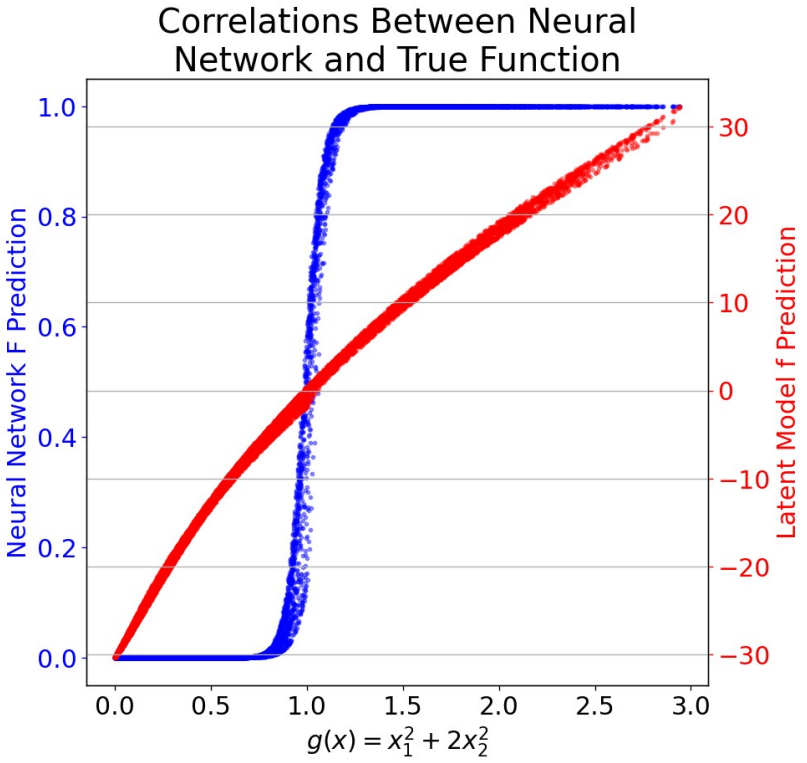




Equivalence Class

Example: Train a Neural Network Classifier to solve a binary Classification problem with decision boundary $x_1^2 + 2x_2^2 = 1$

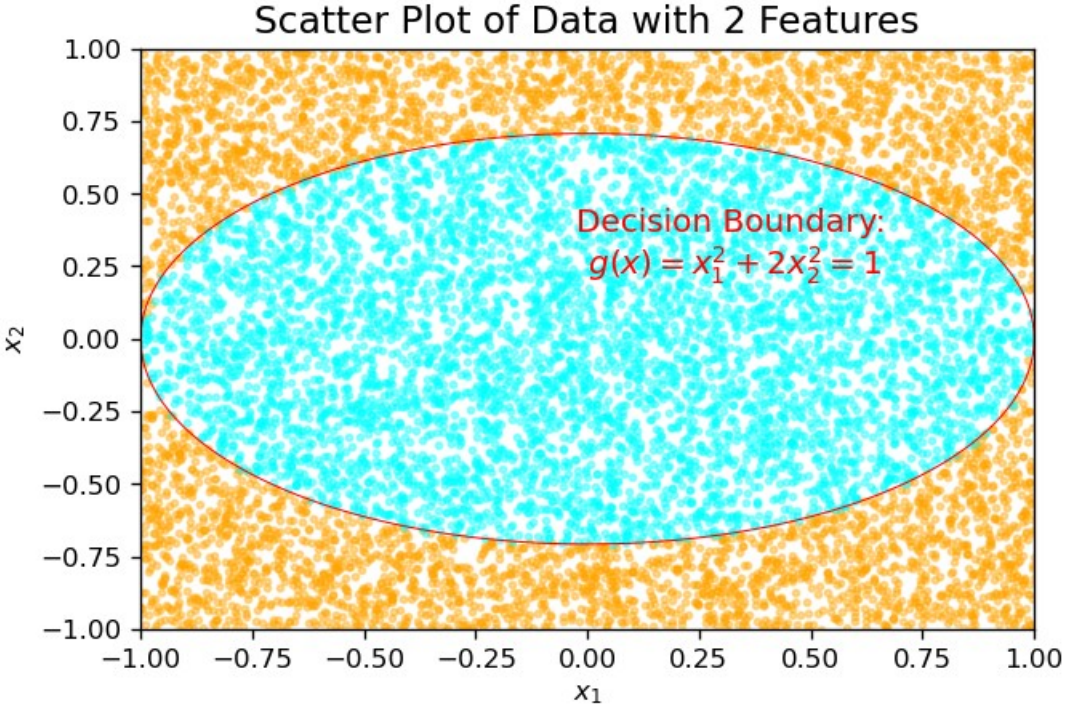


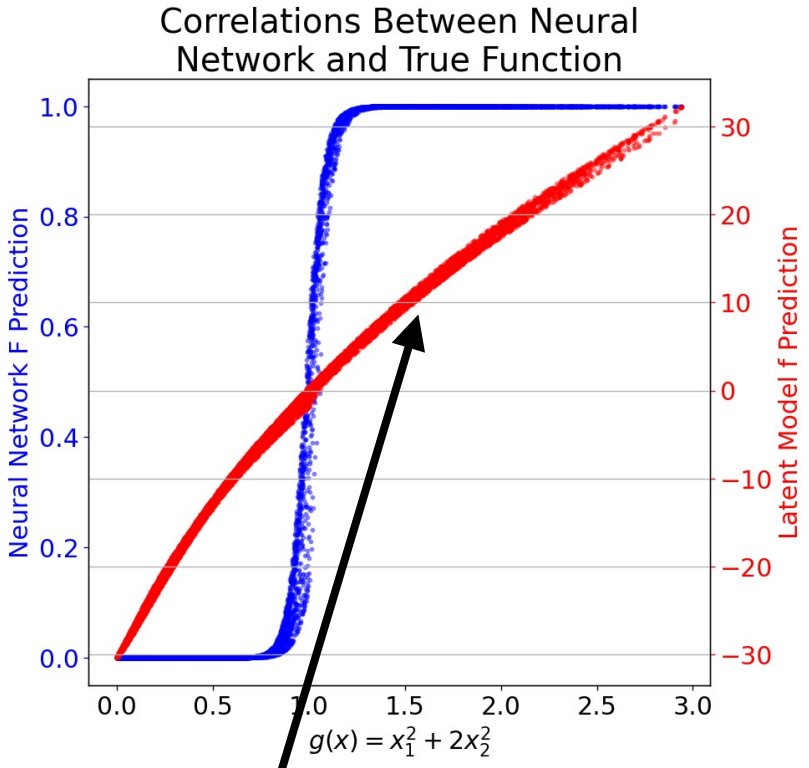


Where is ϕ ?

Equivalence Class

Example: Train a Neural Network Classifier to solve a binary Classification problem with decision boundary $x_1^2 + 2x_2^2 = 1$





There is ϕ !

Equivalence Class

Define the equivalence class of functions $f(\mathbf{x})$ that contain the same information as $g(\mathbf{x})$

$$\tilde{H}_g = \left\{ f(\mathbf{x}) \in C^1(D \subset \mathbb{R}^n, \mathbb{R}) \mid \exists \text{ invertible } \phi \in C^1(\mathbb{R}, \mathbb{R}) \ : \ f(\mathbf{x}) = \phi(g(\mathbf{x})) \right\}$$

This could be seen as the latent model

This could be seen as a representation of a learned concept

Equivalence Class

Define the equivalence class of functions $f(\mathbf{x})$ that contain the same information as $g(\mathbf{x})$

$$\tilde{H}_g = \left\{ f(\mathbf{x}) \in C^1(D \subset \mathbb{R}^n, \mathbb{R}) \mid \exists \text{ invertible } \phi \in C^1(\mathbb{R}, \mathbb{R}) \ : \ f(\mathbf{x}) = \phi(g(\mathbf{x})) \right\}$$

This could be seen as the latent model

This could be seen as a representation of a learned concept

Let us calculate the gradient with respect to the input:

$$\nabla f(\mathbf{x}) = \phi'(g(\mathbf{x}))\nabla g(\mathbf{x})$$

We observe that the Gradients are parallel

$$H_{g\pm} = \left\{ f(\mathbf{x}) \in C^1(D \subset \mathbb{R}^n, \mathbb{R}) \mid \frac{\nabla f(\mathbf{x})}{\|\nabla f(\mathbf{x})\|} = \frac{\pm \nabla g(\mathbf{x})}{\|\nabla g(\mathbf{x})\|} \vee \nabla f(\mathbf{x}) = \nabla g(\mathbf{x}) = 0, \ \forall \mathbf{x} \in D \right\}$$

It is possible to prove that both equivalence classes are the same $\tilde{H}_g = H_g = H_{g+} \cup H_{g-}$ under mild assumptions

Equivalence Class

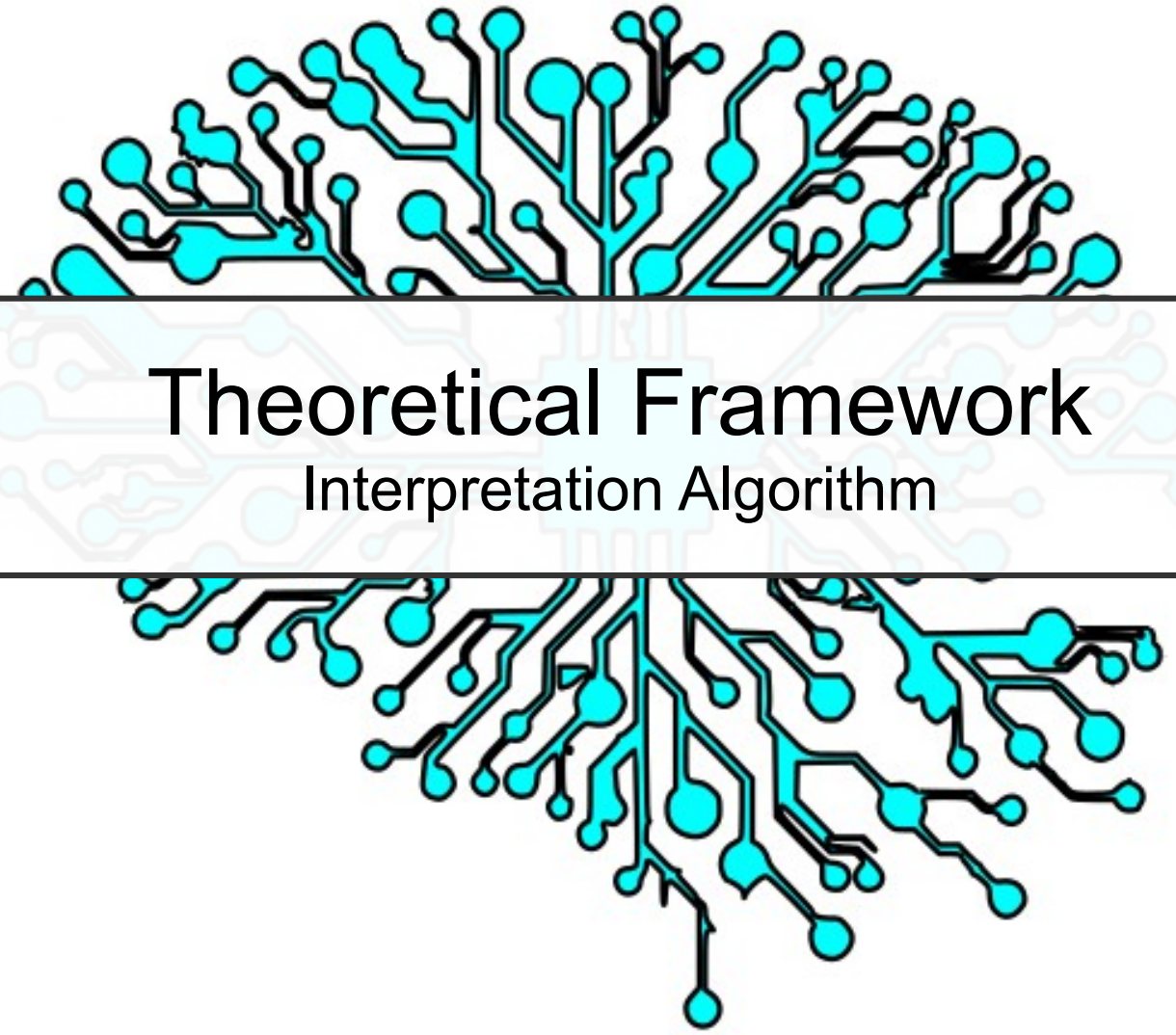
In general the deformation does not need to be invertible or bijective

$$\widetilde{H}_g^{\mathrm{nb}} = \left\{ f(\mathbf{x}) \in C^1(D \subset \mathbb{R}^n, \mathbb{R}) \mid \exists \phi \in C^1(\mathbb{R}, \mathbb{R}) : f(\mathbf{x}) = \phi(g(\mathbf{x})) \right\}$$

The gradients are still aligned but they can point in different directions at different inputs

$$\nabla f(\mathbf{x}) = \phi'(g(\mathbf{x}))\nabla g(\mathbf{x})$$

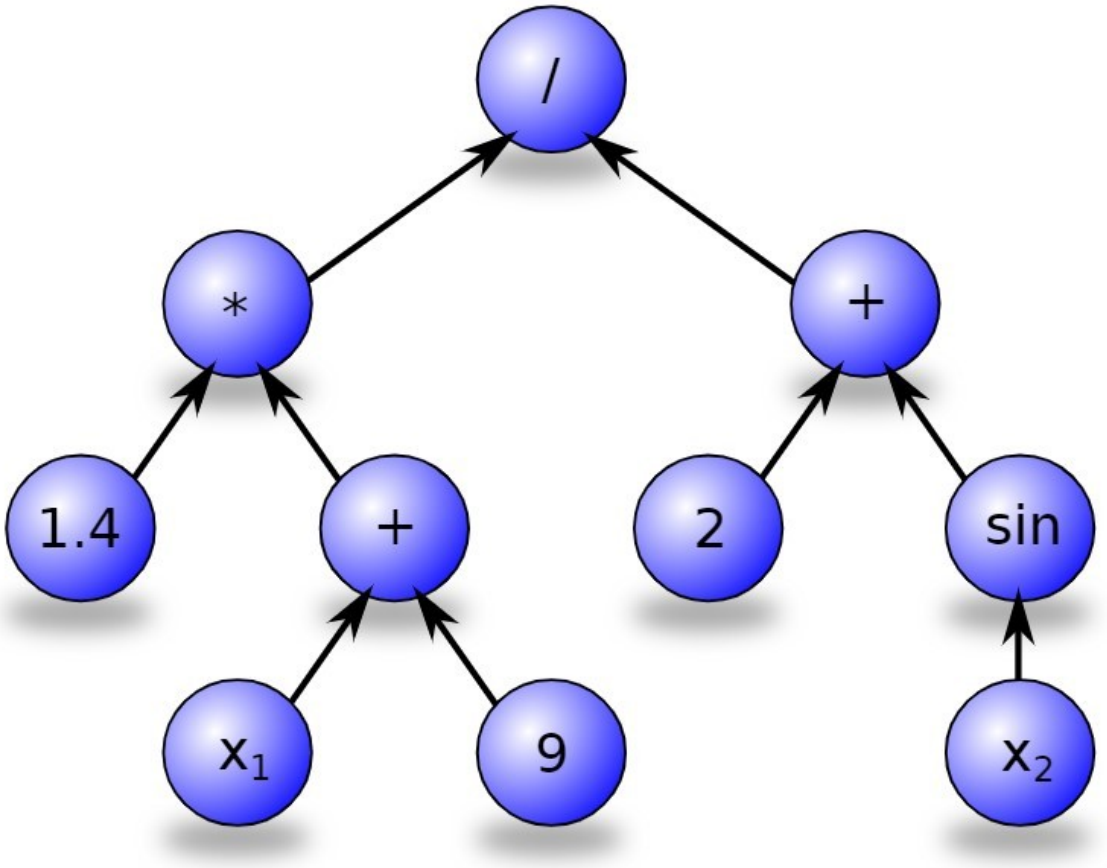
$$H_g^{\mathrm{nb}} = \left\{ f(\mathbf{x}) \in C^1(D \subset \mathbb{R}^n, \mathbb{R}) | \frac{\nabla f(\mathbf{x})}{\|\nabla f(\mathbf{x})\|} = \frac{\pm \nabla g(\mathbf{x})}{\|\nabla g(\mathbf{x})\|} \vee \nabla f(\mathbf{x}) = \nabla g(\mathbf{x}) = 0, \forall \mathbf{x} \in D \right\}$$



Theoretical Framework

Interpretation Algorithm

Symbolic Search

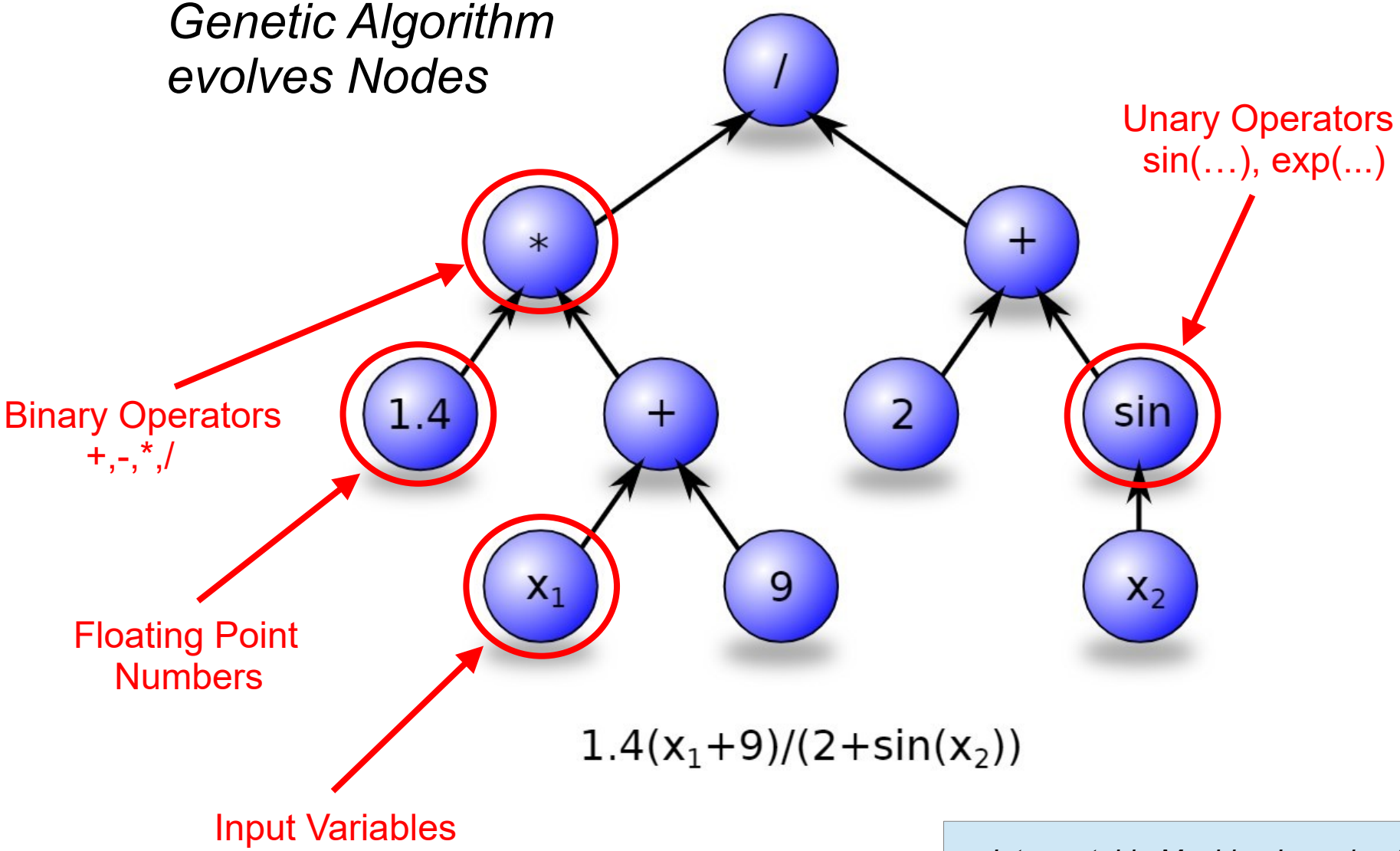


$$1.4(x_1+9)/(2+\sin(x_2))$$

Interpretable Machine Learning for Science with PySR and SymbolicRegression.jl
Cranmer, arXiv:2305.01582

Symbolic Search

Genetic Algorithm evolves Nodes

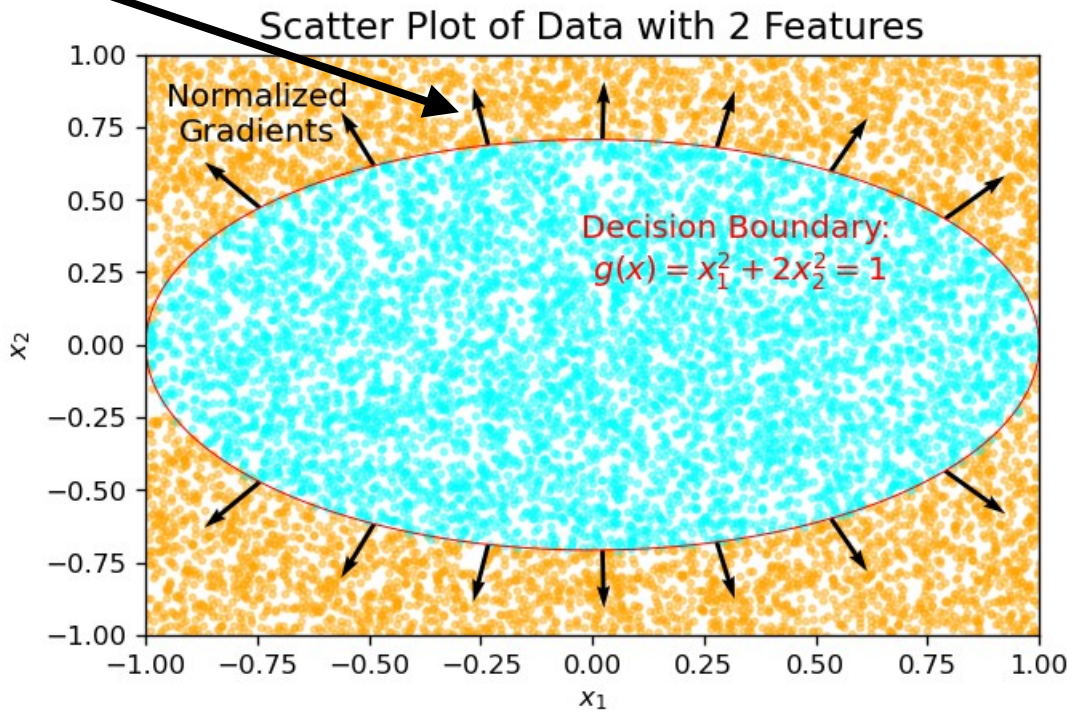


Interpretable Machine Learning for Science with PySR and SymbolicRegression.jl
Cranmer, arXiv:2305.01582

Interpretation Algorithm

In order to find a function that contains the same information as the neural network and is human readable, we calculate the intersection between H_g and the symbolic search space.

We fit a symbolic tree model to replicate the normalized gradients of a latent neural network



Interpretation Algorithm

Minimize Loss functions between normalized gradients of symbolic tree and neuron

- *Bijective Case:*

$$\mathrm{MSE}(g_T(X), g_f(X)) = \frac{1}{n}\sum_{i=1}^{n} \|g_T(\mathbf{x}_i) - g_f(\mathbf{x}_i)\|^2$$

- *Non-bijective Case:*

$$\mathrm{SCSL}(g_T(X), g_f(X)) = -\frac{1}{n}\sum_{i=1}^{n} \left(g_T(\mathbf{x}_i) \cdot g_f(\mathbf{x}_i)\right)^2$$

Interpretation Algorithm

Algorithm 1: Train Neural Network for Binary Classification

Data: Labelled data set $D = (X_{train}, Y_{train})$

Input: Neural Network Hyperparameters

1 Initialize neural network classifier with sigmoid activation at output F

2 Train F on D

3 $f \leftarrow$ remove sigmoid activation of output neuron of F

Output: Trained model F, latent model f

Interpretation Algorithm

Algorithm 2: Obtain Gradients of Latent Model

Data: Labelled data set $D = (X_{train}, Y_{train})$
Unlabelled data set $D_u = (X_u)$ ▷ Artificial data not used for training

Input: Trained model F and latent model f
Selection threshold δ

1 $\tilde{X} \leftarrow (X_{train}, X_u)$ ▷ Add artificial unlabelled data

2 $\tilde{X} \leftarrow \tilde{X}$ **where** $F(\tilde{X}) \in [\delta, 1-\delta]$ ▷ Select data close to decision boundary

3 $G_f \leftarrow [\nabla f(\mathbf{x})$ for $\mathbf{x}$ in $\tilde{X}]$ ▷ Gradients of latent model wrt. input

4 $G_f \leftarrow [$**if** $\nabla f(\mathbf{x}) \neq 0 : \nabla f(\mathbf{x}) / \|\nabla f(\mathbf{x})\|$

5 **else** $\nabla f(\mathbf{x})$ for $\nabla f(\mathbf{x})$ in $G_f]$ ▷ Normalize Gradients

Output: $(\tilde{X}, G_f)$

Interpretation Algorithm

Algorithm 3: Symbolic Search

Data: Gradient data set $(\tilde{X}, G_f)$

Input: Symbolic Regression Hyperparameters
Set of unary and binary operations.

initialize symbolic regression model T

evolve T **with** (

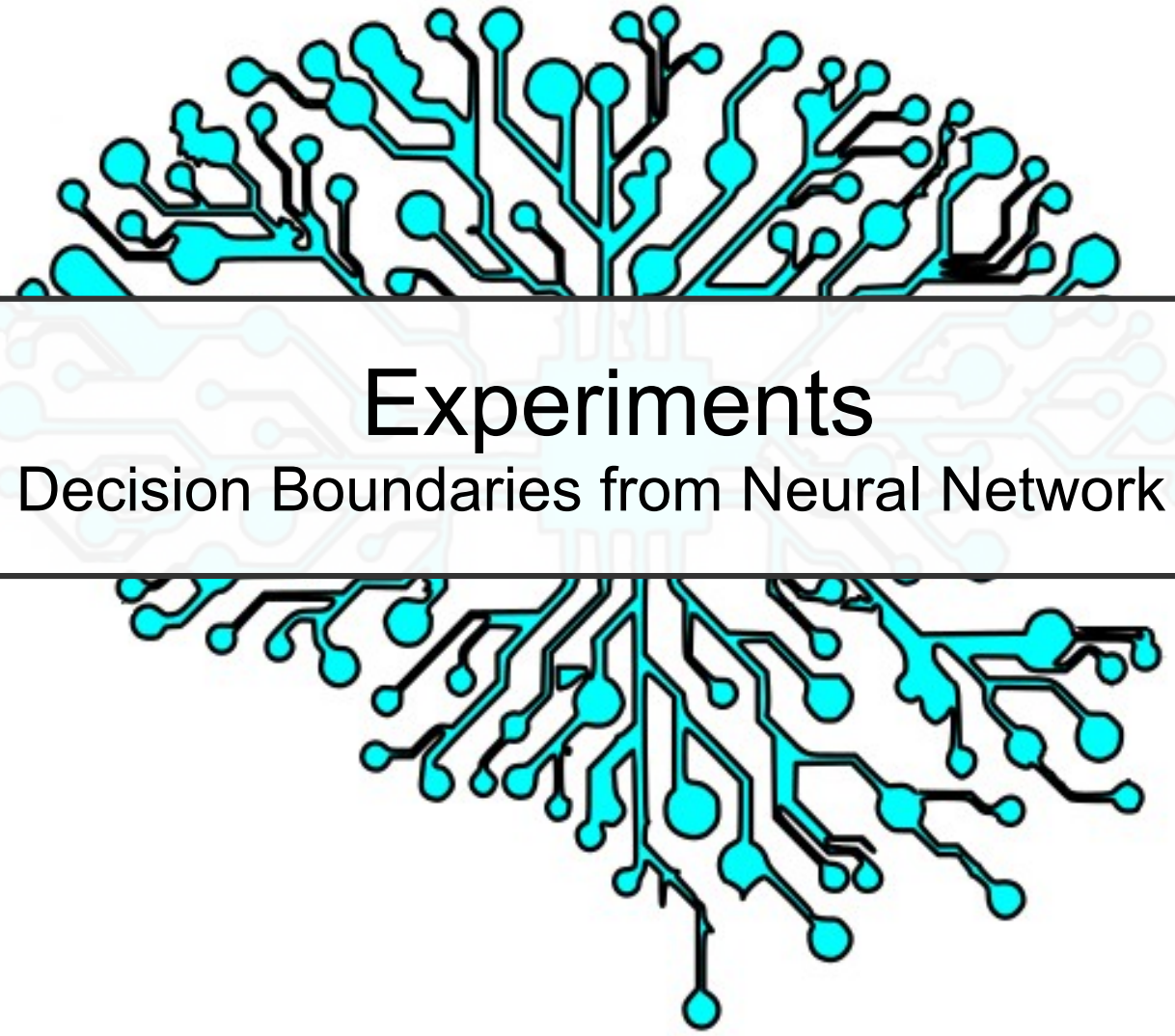
$G_T \leftarrow [\nabla T(\mathbf{x}) \text{ for } \mathbf{x} \text{ in } \tilde{X}]$ ▷ Gradients of symbolic model

$G_T \leftarrow [\textbf{if } \nabla T(\mathbf{x}) \neq 0 : \nabla T(\mathbf{x}) / \|\nabla T(\mathbf{x})\|$

else $\nabla T(\mathbf{x})$ for $\nabla T(\mathbf{x})$ in $G_T]$) ▷ Normalize Gradients

to minimize $\text{MSE}(G_f, G_T)$

Output: Symbolic regression model T



Experiments

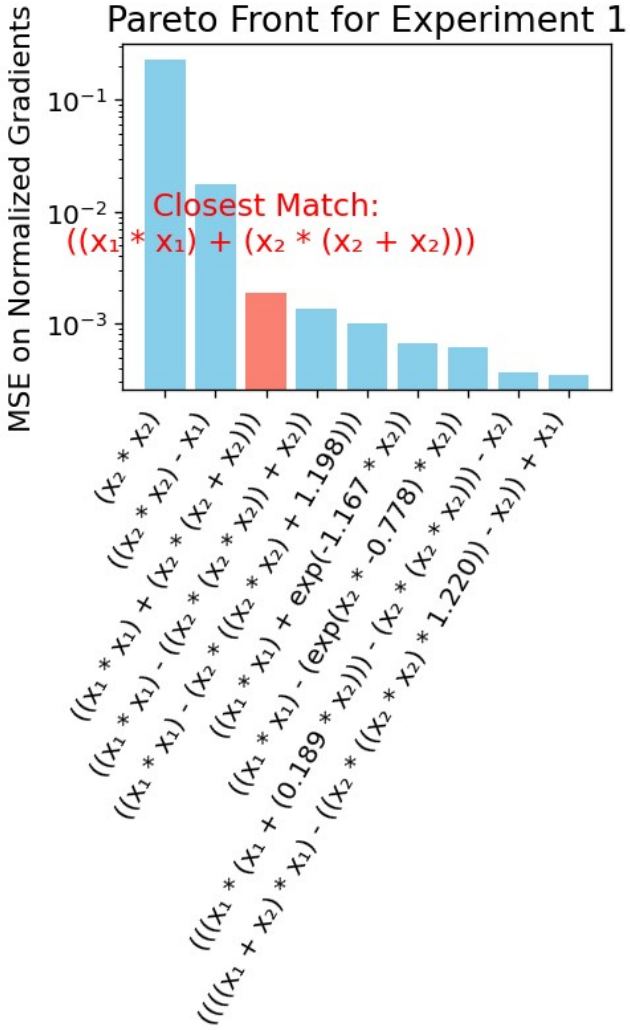
Recover Decision Boundaries from Neural Network Classifiers

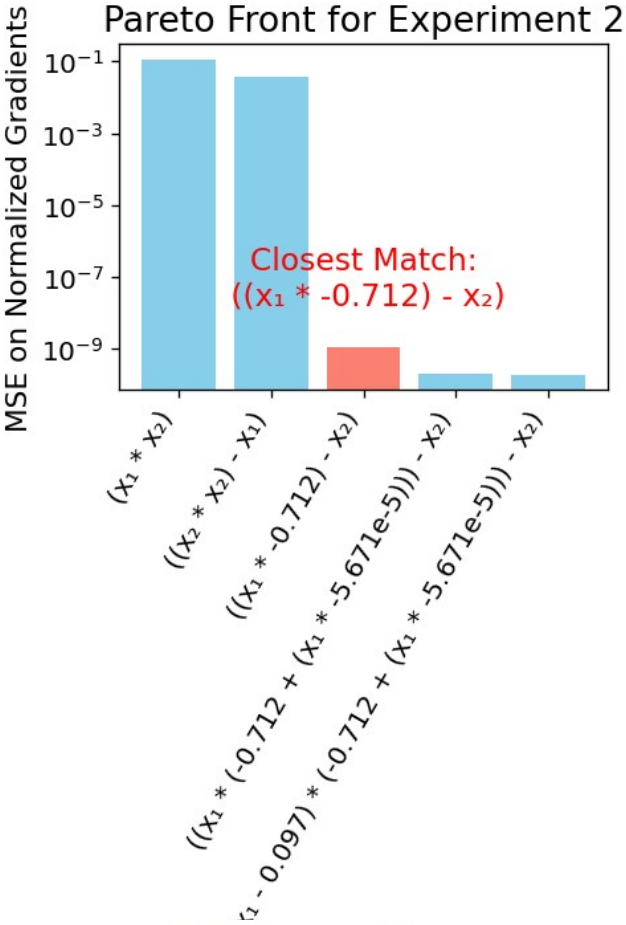
Interpretation (Classification)

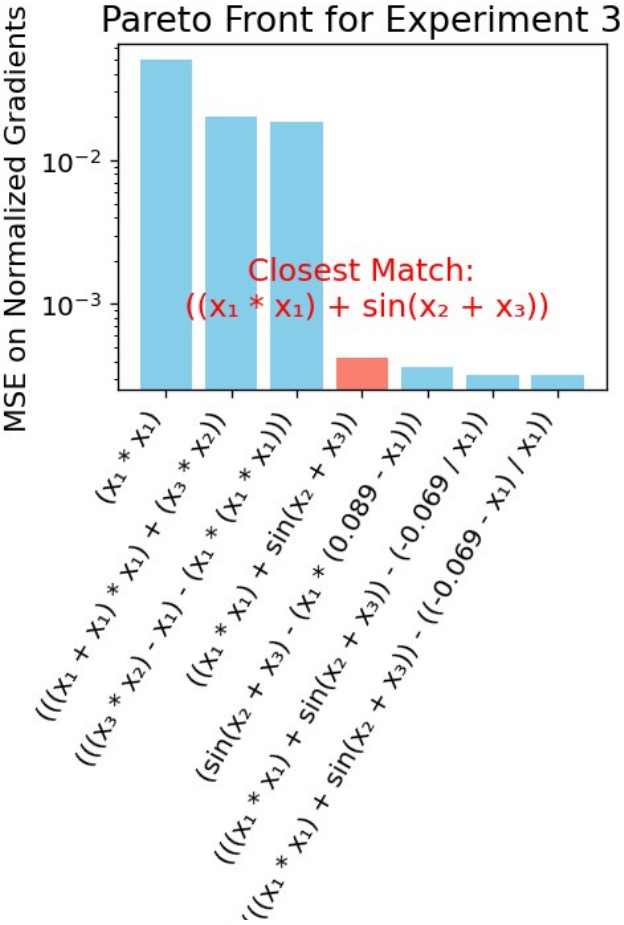
We train a Neural Network Classifier to solve 7 binary classification problems and interpret the learned concepts by applying our interpretation framework

	Variables	Decision Formula
Experiment 1	2	$x_1^2 + 2x_2^2 > 1$
Experiment 2	2	$x_1^2 + 3x_1x_2 + 2x_2^2 > 5$
Experiment 3	3	$x_1^2 + \sin(x_2 + x_3) > 1$
Experiment 4	3	$\exp(x_1 - x_2) - \pi x_3 > 1$
Experiment 5	3	$x_1 \exp\left(-\frac{1}{2}(x_2^2 + x_3^2)\right) > \frac{1}{5}$
Experiment 6	6	$\frac{x_1x_2}{(x_3-x_4)^2+(x_5-x_6)^2} > 1$
Experiment 7	4	$x_1x_2 > 1 \wedge x_3x_4 > 1$ vs $x_1x_2 < 1 \wedge x_3x_4 < 1$

Interpretation (Classification)

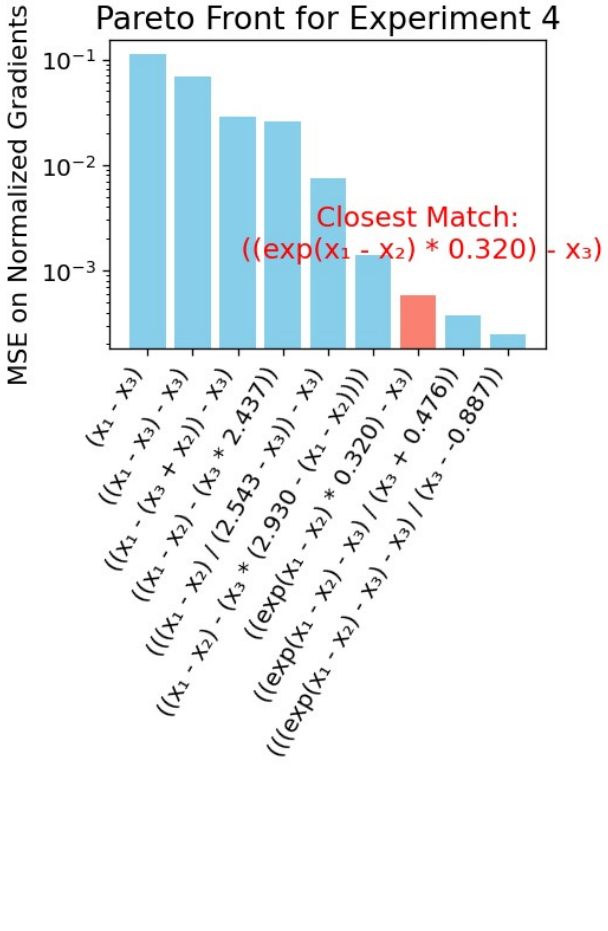


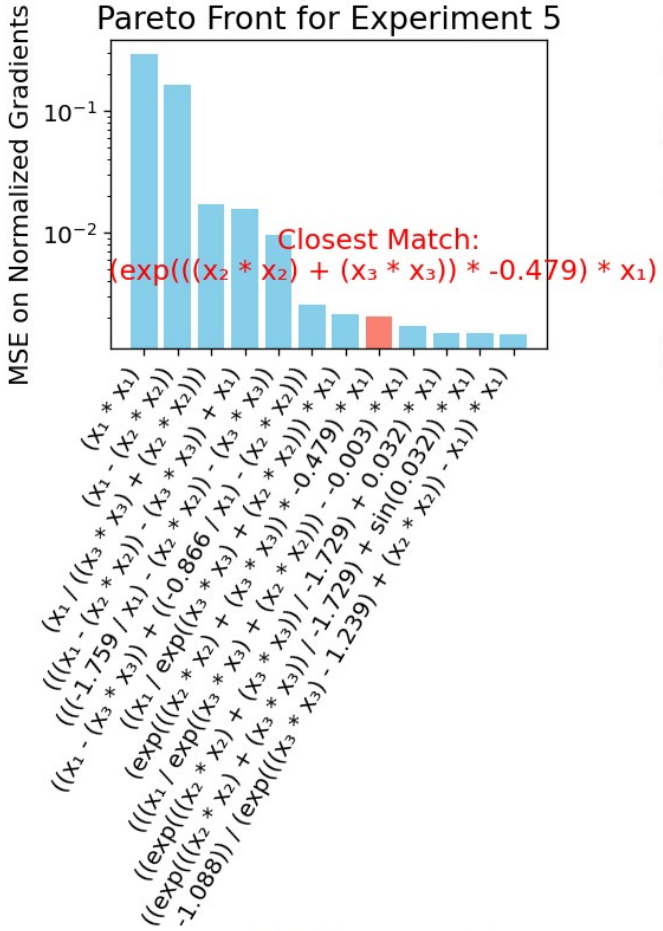


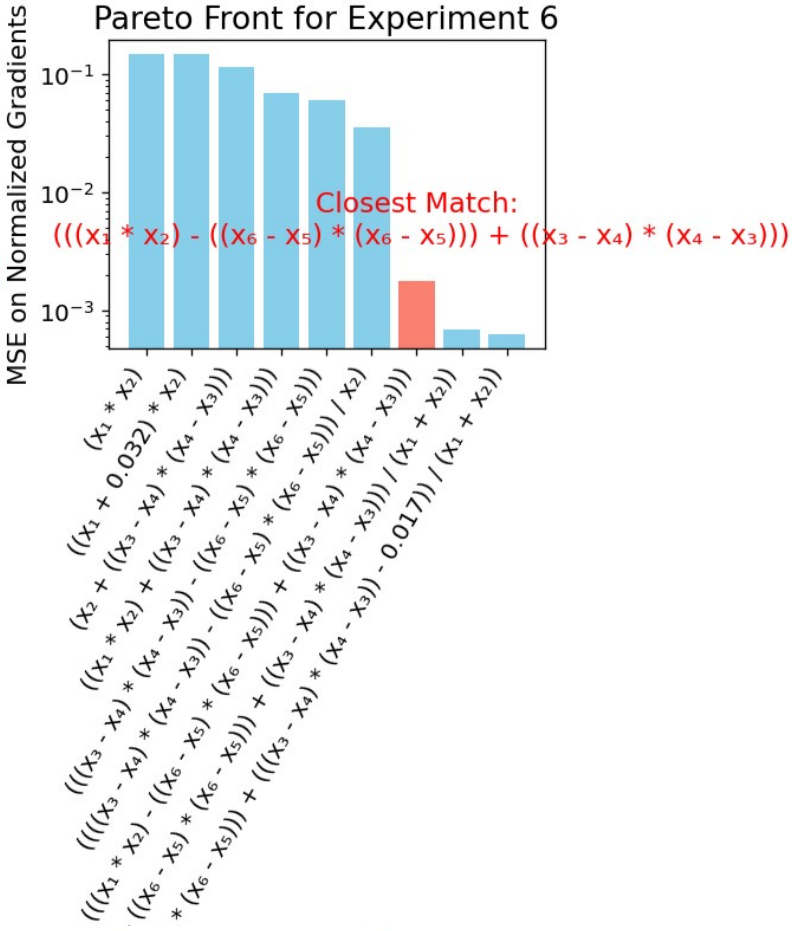


	Variables	Decision Formula
Experiment 1	2	$x_1^2 + 2x_2^2 > 1$
Experiment 2	2	$x_1^2 + 3x_1x_2 + 2x_2^2 > 5$
Experiment 3	3	$x_1^2 + \sin(x_2 + x_3) > 1$

Interpretation (Classification)



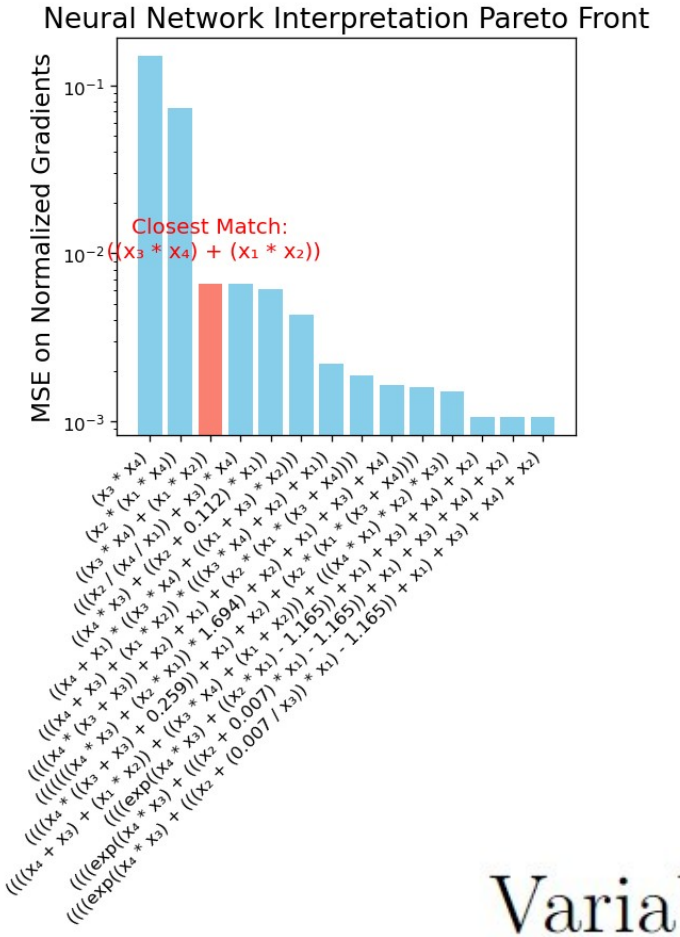


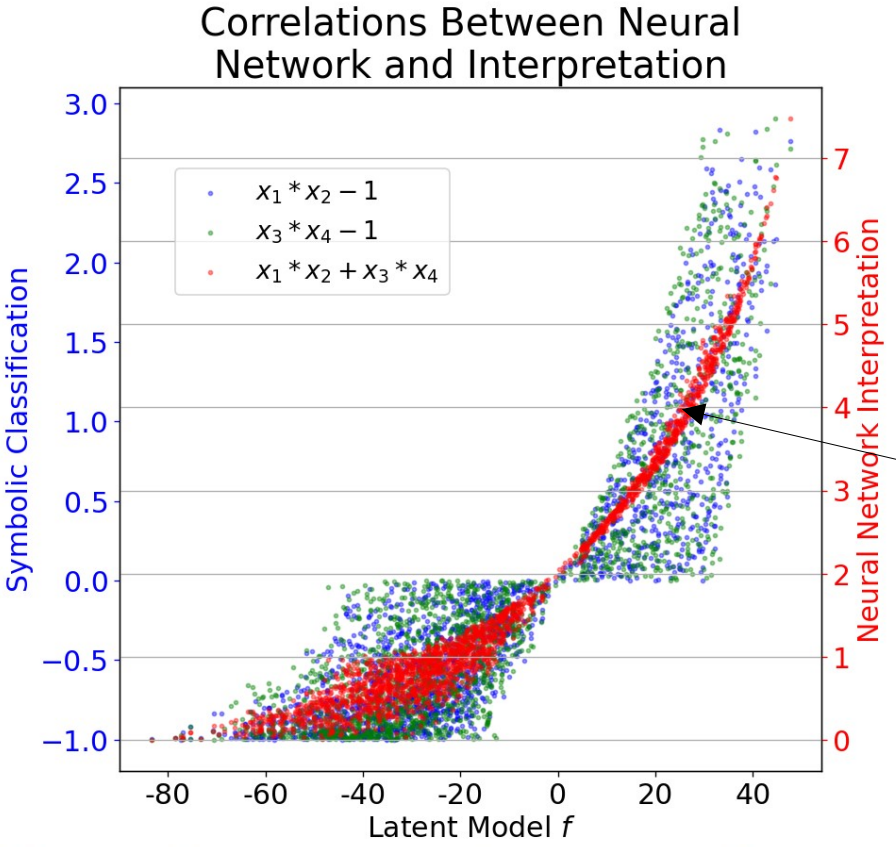


	Variables	Decision Formula
Experiment 4	3	$\exp(x_1 - x_2) - \pi x_3 > 1$
Experiment 5	3	$x_1 \exp\left(-\frac{1}{2}(x_2^2 + x_3^2)\right) > \frac{1}{5}$
Experiment 6	6	$\frac{x_1 x_2}{(x_3 - x_4)^2 + (x_5 - x_6)^2} > 1$

Interpretation (Classification)

We test the interpretation framework on a problem with multiple features that can be learned for a successful classification.

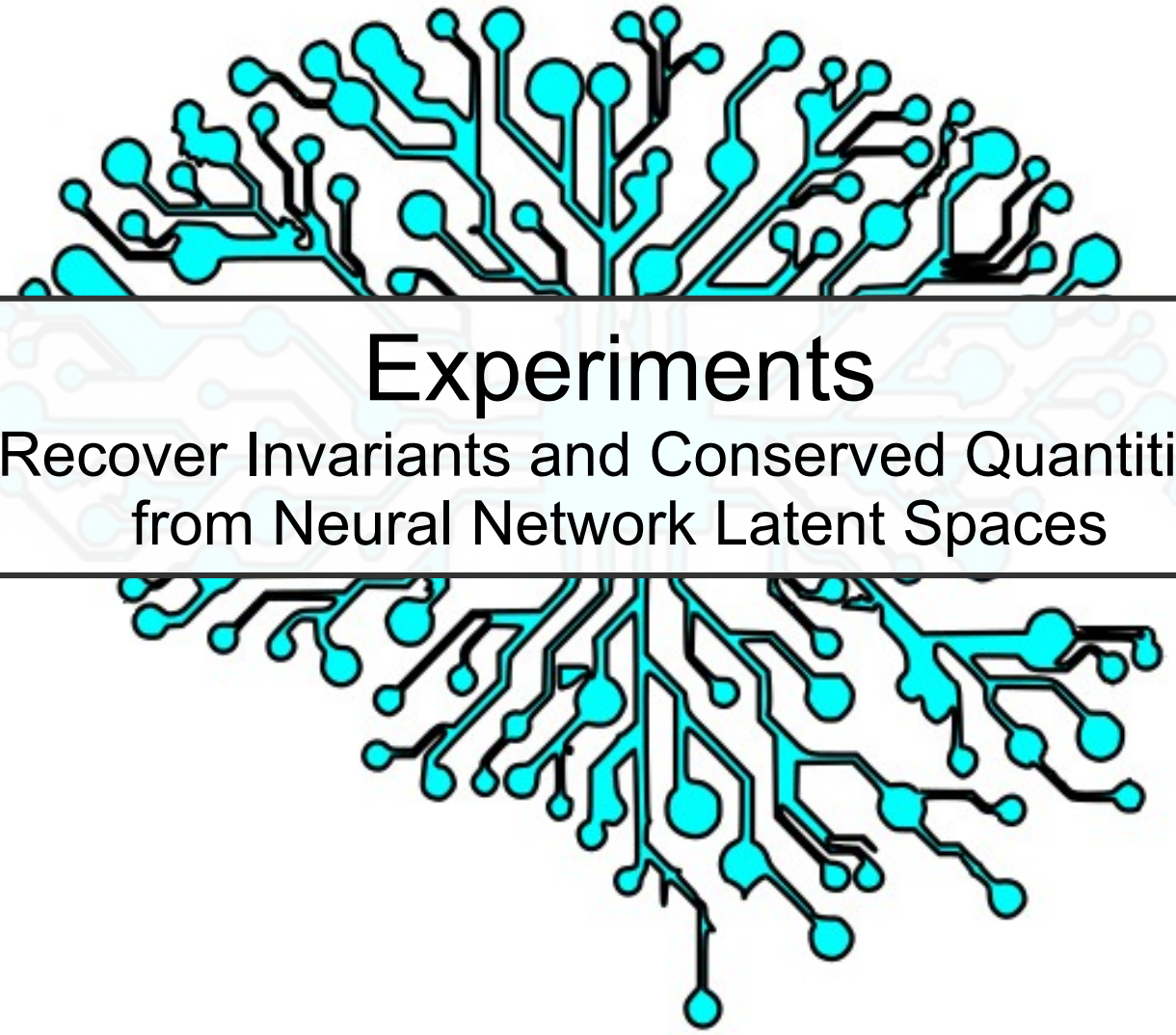




Low Correlation with single features

Strong Correlation with combination of both features

	Variables	Decision Formula
Experiment 7	4	$x_1x_2 > 1 \wedge x_3x_4 > 1$ vs $x_1x_2 < 1 \wedge x_3x_4 < 1$

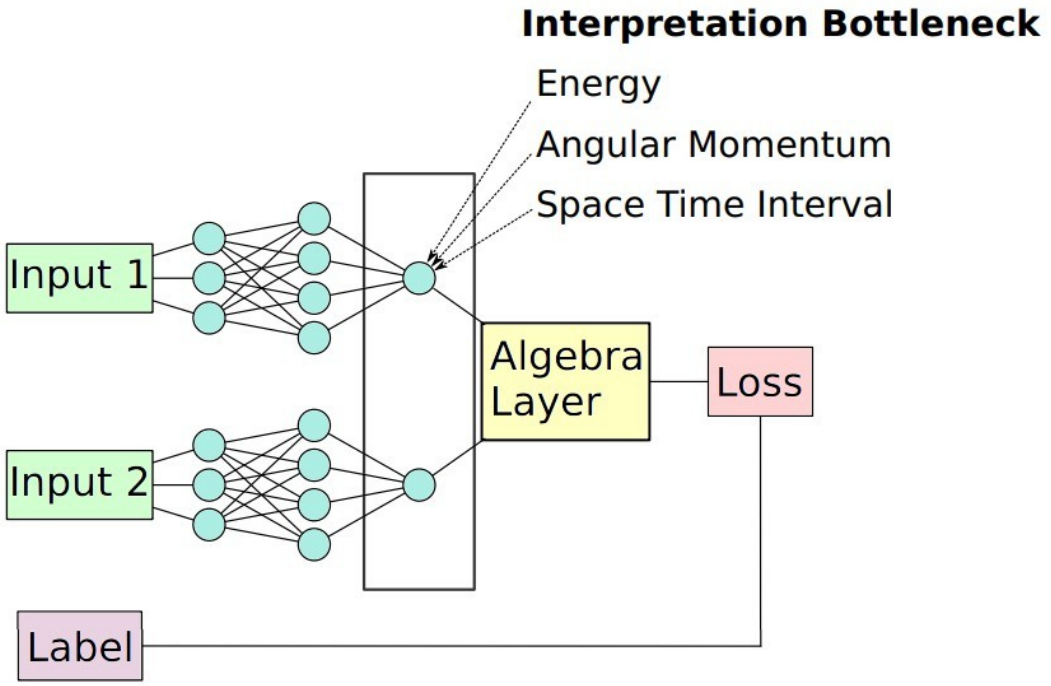


Experiments

Recover Invariants and Conserved Quantities from Neural Network Latent Spaces

Interpretation (Latent Spaces)

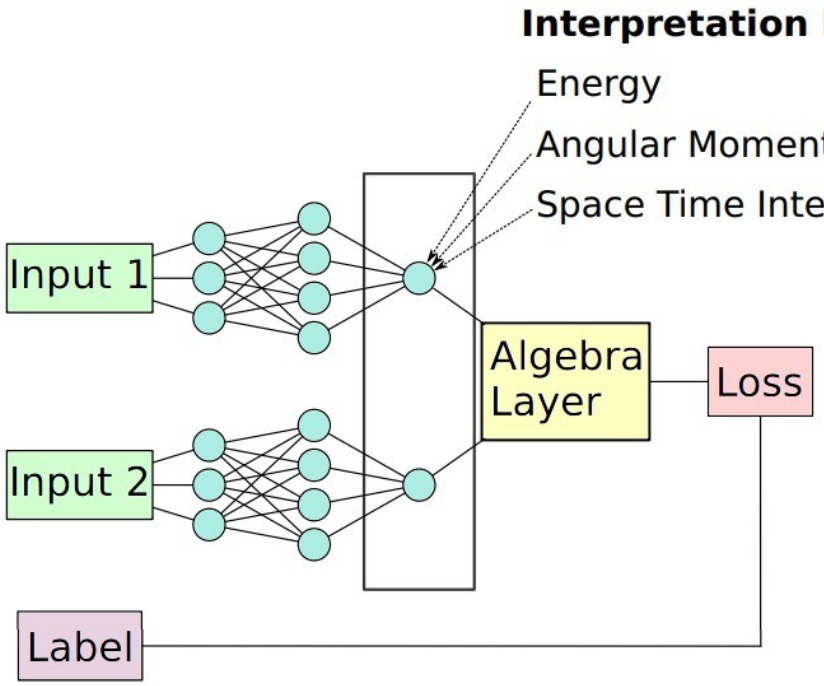
- It has been shown that Siamese Networks can be trained to encode Invariants and Conserved Quantitites in their latent spaces.



Discovering Symmetry Invariants and Conserved Quantitites by Interpreting Siamese Neural Networks
Wetzel, et al, PRE 2017

Interpretation (Latent Spaces)

Train Siamese Network to solve a binary classification Problem



$$F(x_A, x_P) = \text{same}$$
$$F(x_A, x_N) = \text{different}$$

Positive Samples are related to the Anchor by some transformation M

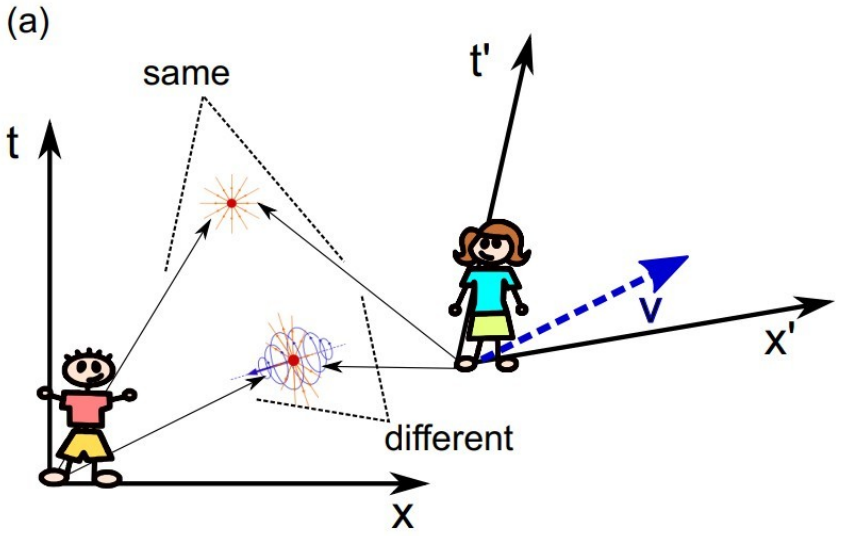
$$x_P = M(x_A)$$

$$x_N$$

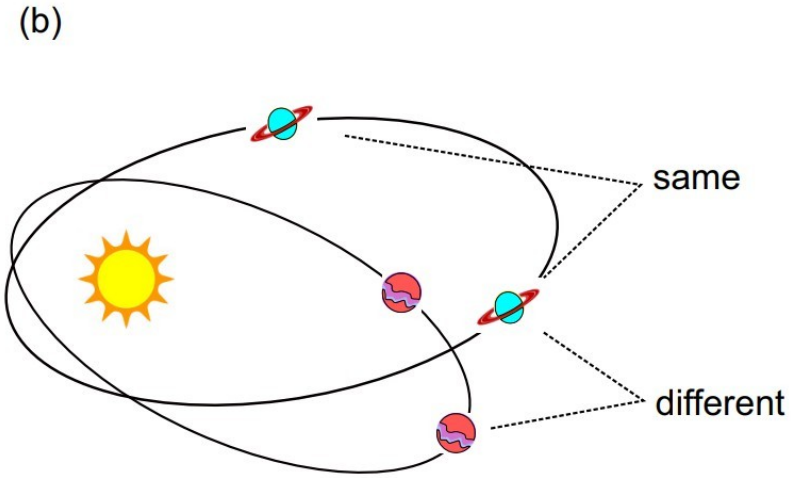
Negative Samples are independent

Interpretation (Latent Spaces)

Examples



Electromagnetism under Lorentz Transformations



Motion under Time Evolution

Interpretation (Latent Spaces)

Results

Table 1: Matrix Invariants

Exp. No.	Name	d	Transformation	Invariant	Analytical Expression	Retrieved Expression
1	2×2	4	Similarity	Trace	$A_{11} + A_{22}$	$-0.677 - A_{11} - A_{22}$
2	2×2	4	Similarity	Determinant	$A_{11}A_{22} - A_{12}A_{21}$	$A_{12}A_{21} - A_{11}A_{22}$
3	3×3	9	Similarity	Trace	$A_{11} + A_{22} + A_{33}$	$A_{11} + A_{22} + A_{33}$
4	3×3 Antisymmetric	9	Orthogonal	Sum of Principal Minors	$A_{12}^2 + A_{23}^2 + A_{13}^2$	$A_{12}A_{21} + A_{23}A_{32} + A_{13}A_{31}$
5	4×4	16	Similarity	Trace	$A_{11} + A_{22} + A_{33} + A_{44}$	$A_{11} + A_{22} + A_{33} + A_{44}$
6	4×4 Field Strenth Tensor	6	Lorentz	Determinant	$E_1B_1 + E_2B_2 + E_3B_3$	$E_1B_1 + E_2B_2 + E_3B_3$

Table 2: Potentials

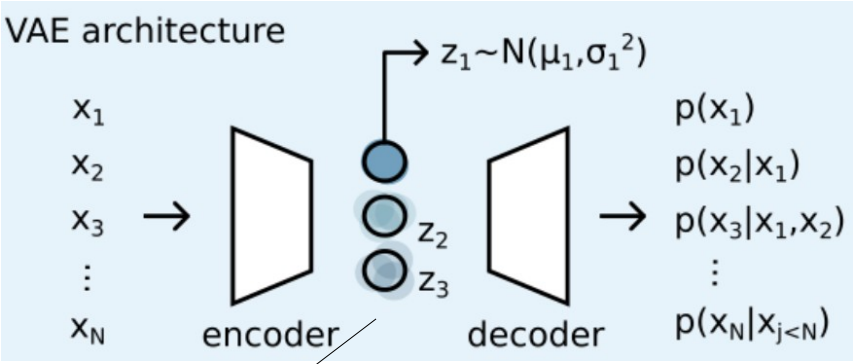
Experiment No.	d	Potential V	Invariant	Analytical Expression	Retrieved Expression
7	2	$\frac{1}{2}x^2$	Energy	$\frac{1}{2}v^2 + \frac{1}{2}x^2$	$0.173/(v^2 + x^2)$
8	2	$\frac{1}{2}x^2 + \frac{1}{4}x^4$	Energy	$\frac{1}{2}v^2 + \frac{1}{2}x^2 + \frac{1}{4}x^4$	$v^2 + \exp(x^2)$
9	2	$\sin(x)$	Energy	$\frac{1}{2}v^2 + \sin(x)$	$-v^2 + \sin(x)/(-0.5)$
10	2	$\frac{1}{2}x^2 + \exp(x+1)$	Energy	$\frac{1}{2}v^2 + \frac{1}{2}x^2 + \exp(x+1)$	$v^2 + x^2 + \exp(x)/\sin(0.184)$
11	4	$-r^{-2}$	Angular Momentum	$x_1v_2 - x_2v_1$	$x_2v_1 - x_1v_2$

Table 3: Spacetime

Experiment No.	d	Transformation	Invariant	Analytical Expression	Retrieved Expression
12	4	Lorentz Transformation	Spacetime Interval	$t^2 - x_1^2 - x_2^2 - x_3^2$	$t^2 - x_1^2 - x_2^2 - x_3^2$

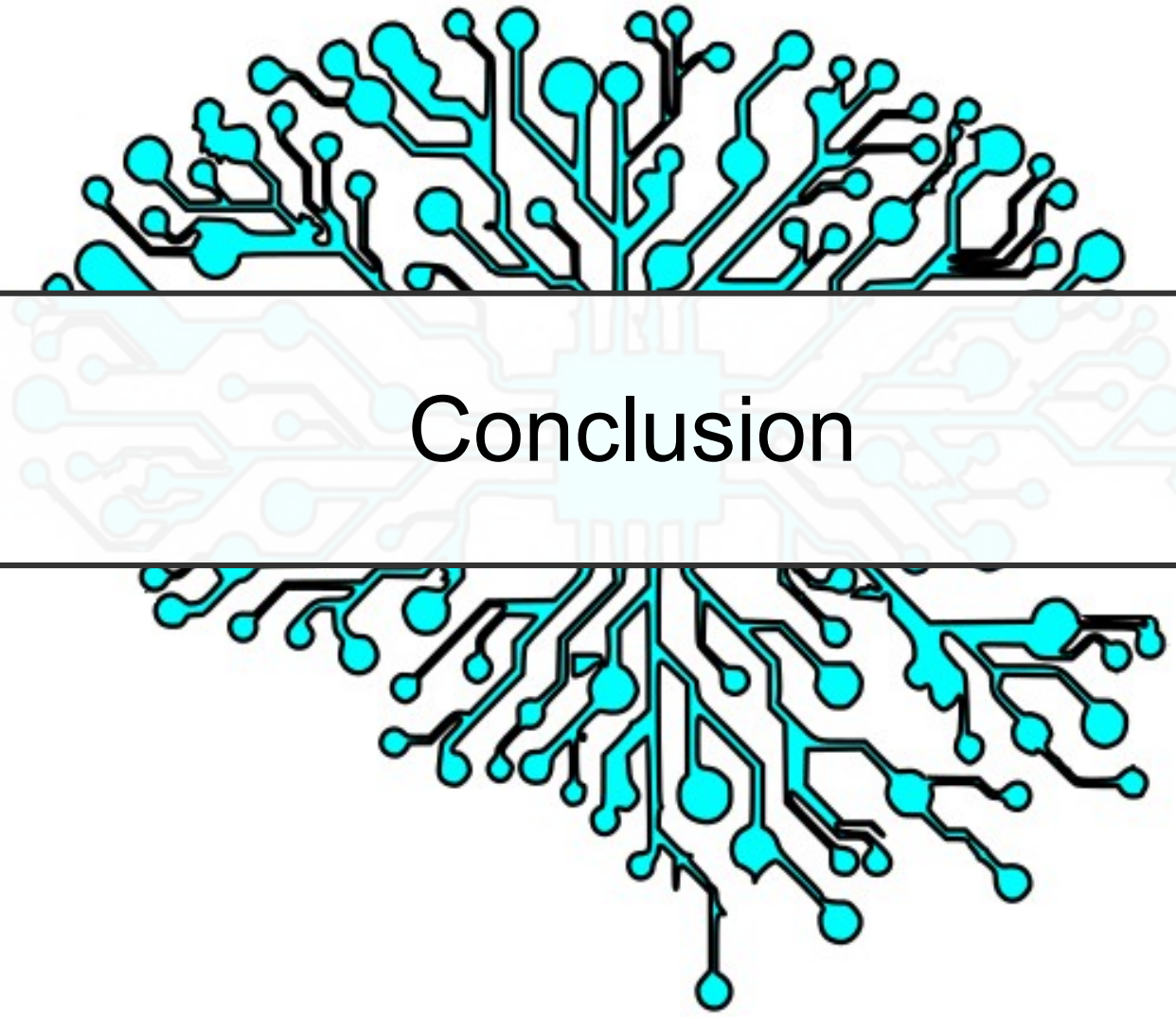
J1J2 Model

$$H = \sum_{\langle i,j \rangle} \sigma_i^z \sigma_j^z + J_2 \sum_{\langle\langle i,j \rangle\rangle} \sigma_i^z \sigma_j^z + h \sum_i \sigma_i^x$$



Discovered Order:
Neel
Striped
Polarized

Discovering quantum phenomena with Interpretable Machine Learning, Schoulepnikoff et al., arXiv:2604.16015



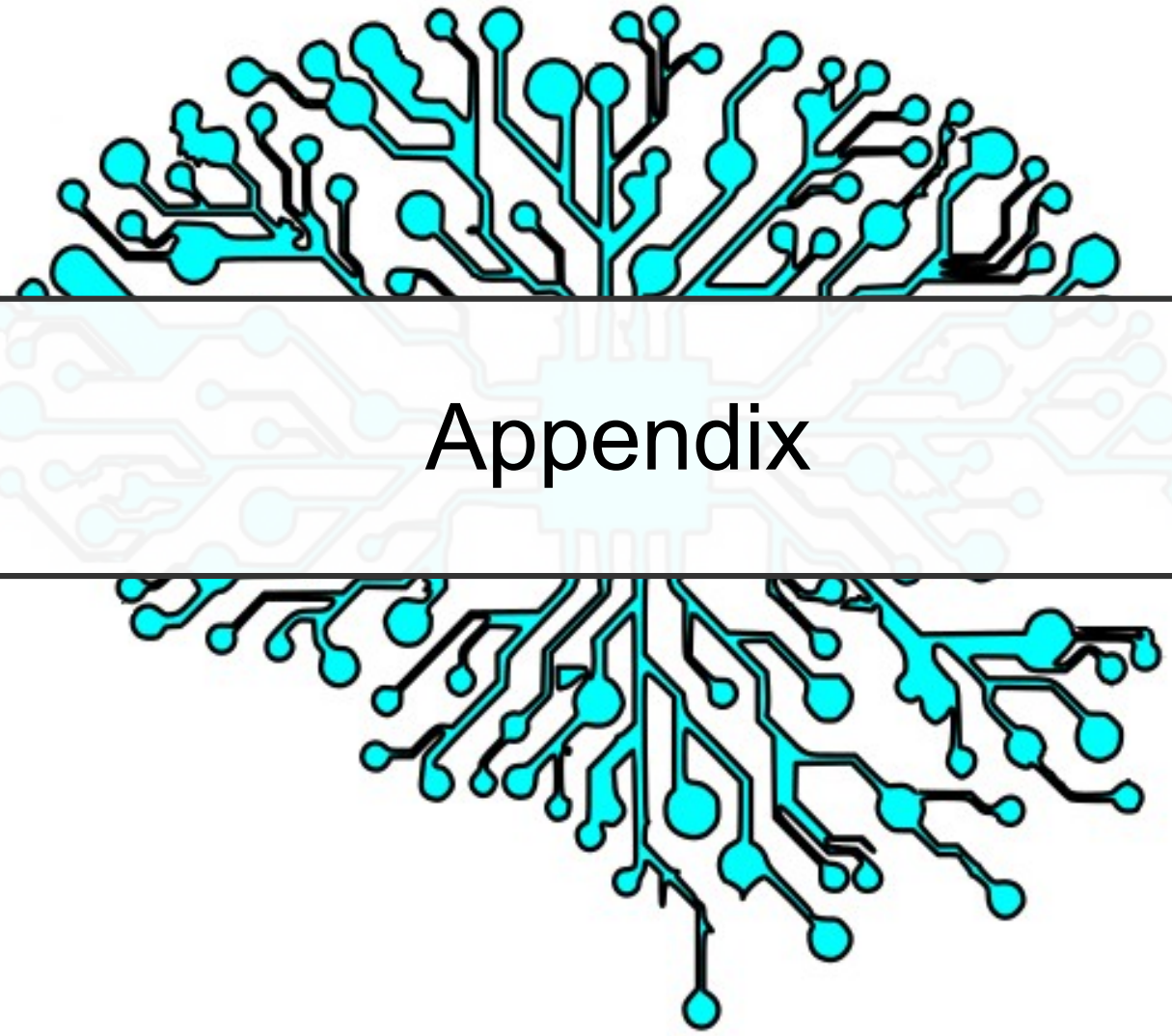
Conclusion

Conclusion

- ✗ Redefine concept as equivalence class of equivalent descriptions.
- ✗ Interpret any neuron by finding a differentiable human-readable representation (eg Polynomial,Symbolic Tree) which approximates the normalized gradients of the neuron wrt the input.
- ✗ Since it is limited to single neurons, best used in combination with bottleneck and decorrelation methods.

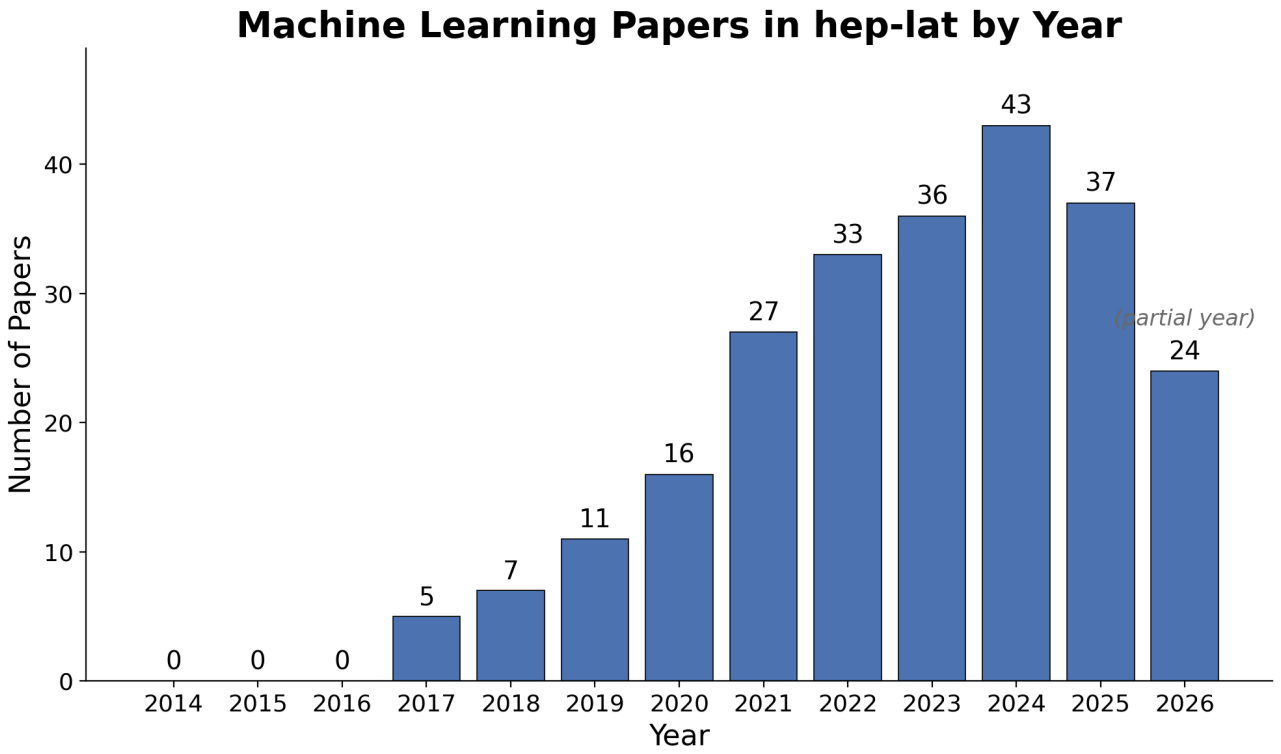
Closed-Form Interpretation of Neural Network Classifiers with Symbolic Gradients, Wetzel, MLST 2025

Closed-Form Interpretation of Neural Network Latent Spaces with Symbolic Gradients, Wetzel, Patel, arXiv:2409.05305

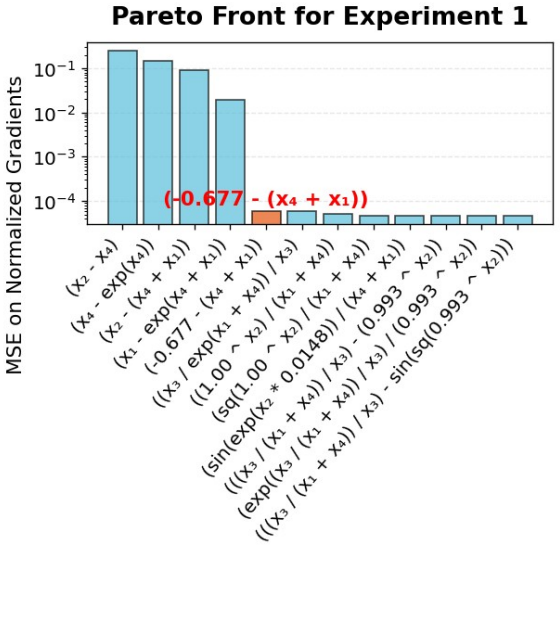


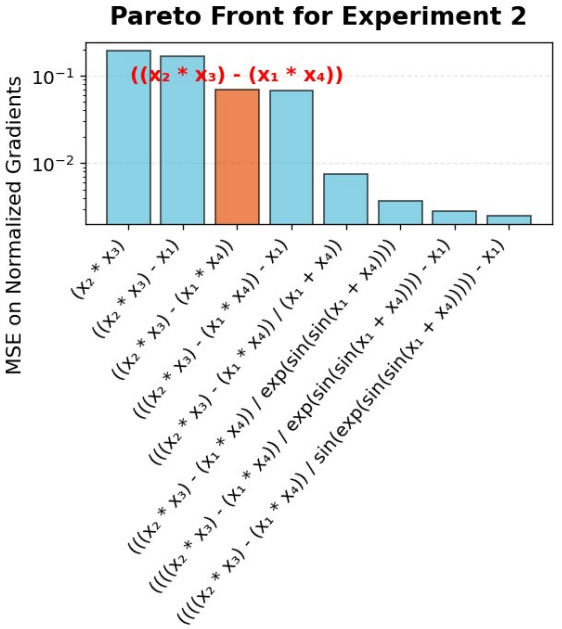
Appendix

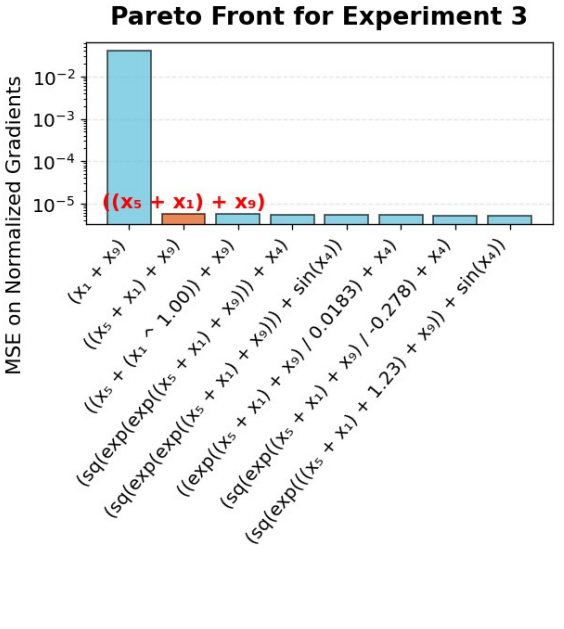
Overview

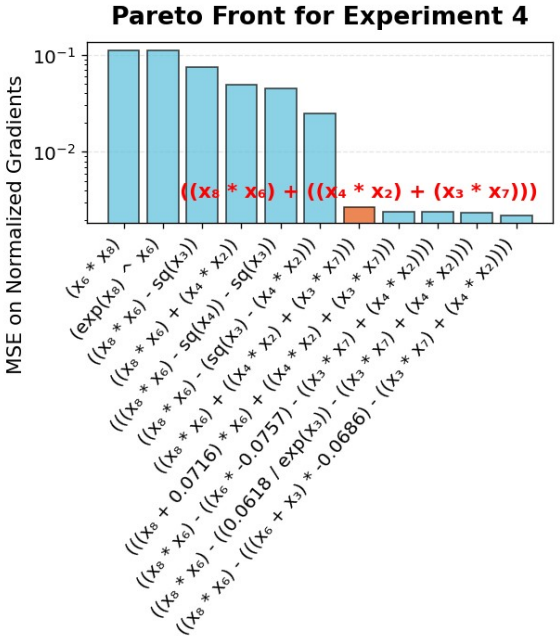

Machine Learning Papers in hep-lat by Year
Number of Papers
Year
0
10
20
30
40
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
0
0
0
5
7
11
16
27
33
36
43
37
24
(partial year)

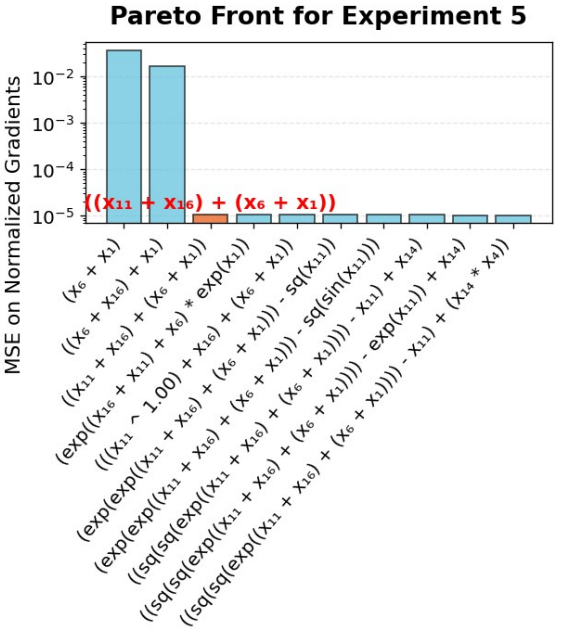
Invariants and Conserved Quantities

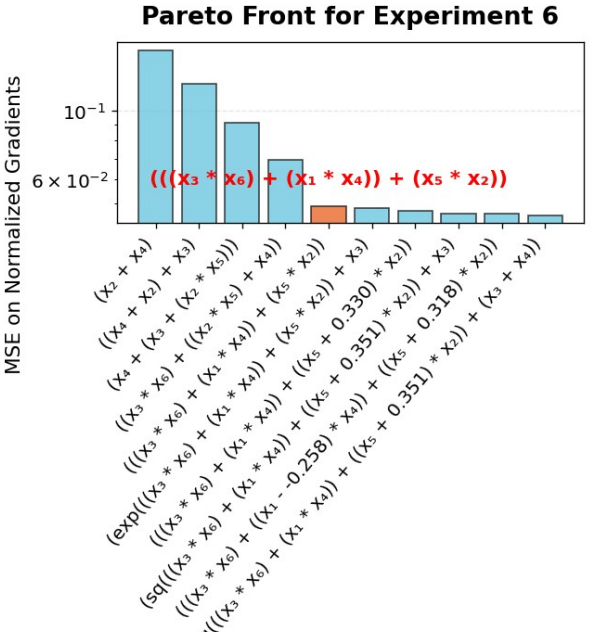




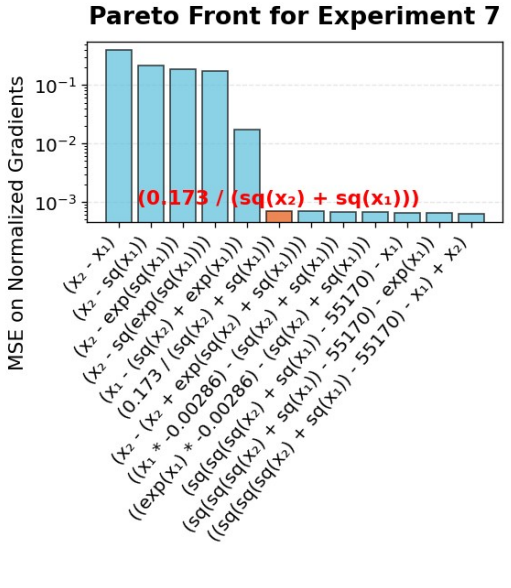


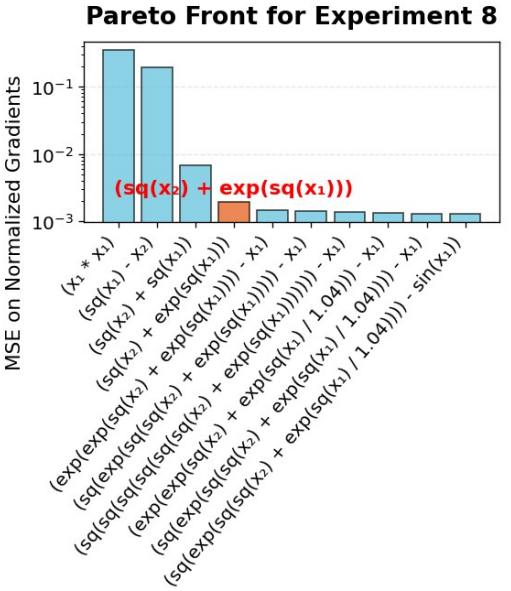


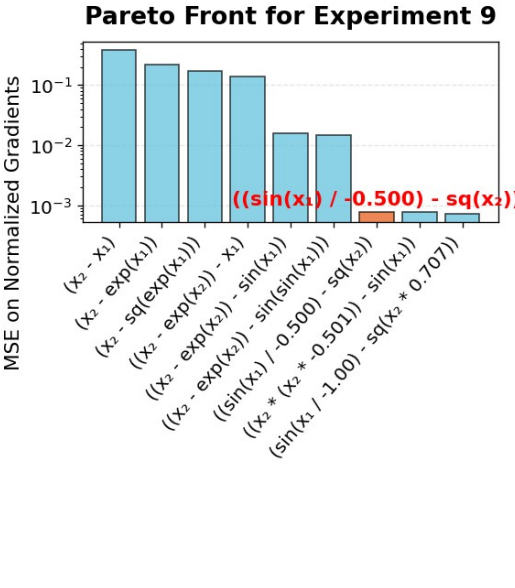


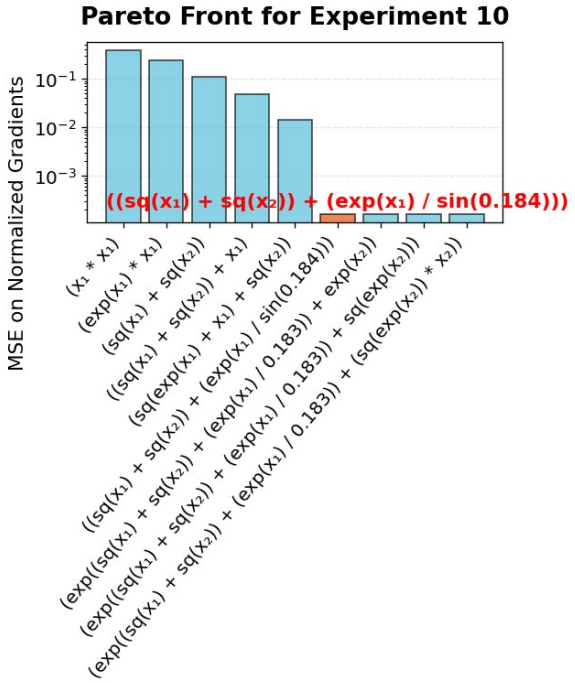


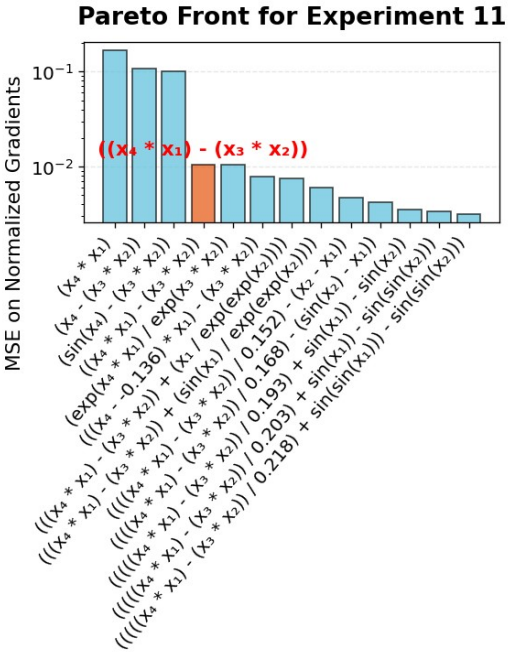
Invariants and Conserved Quantities

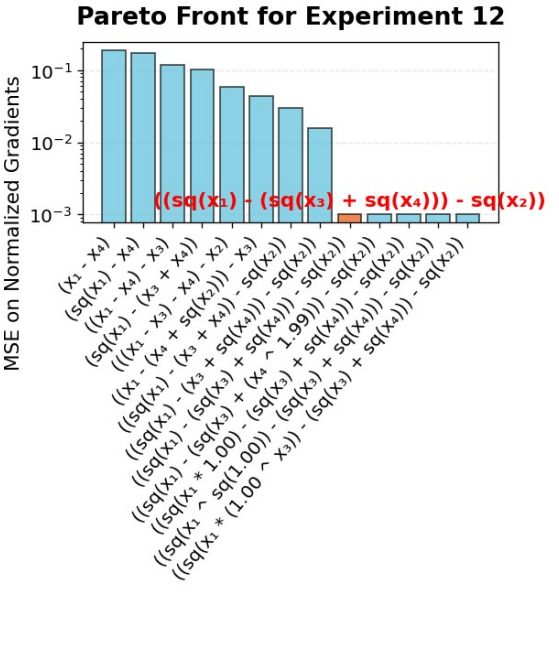




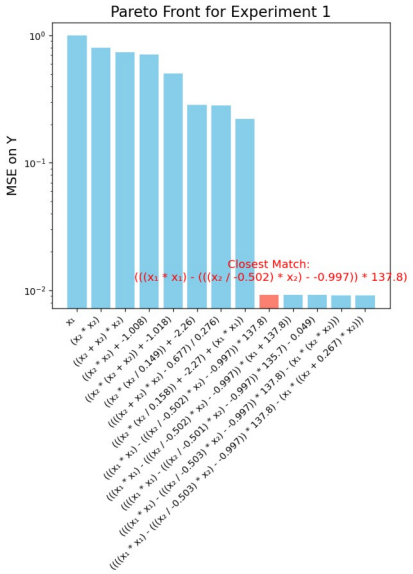


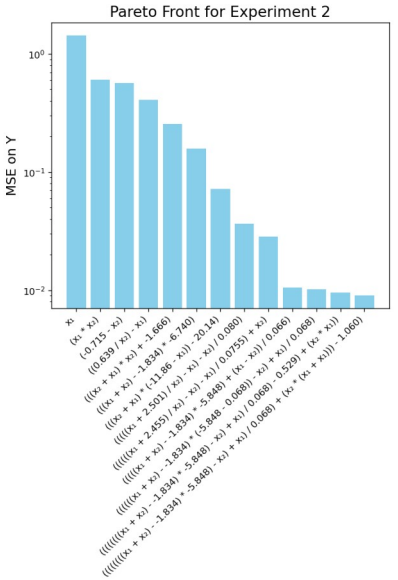


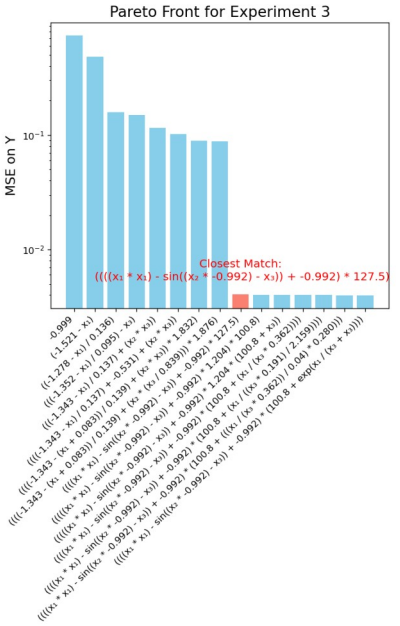


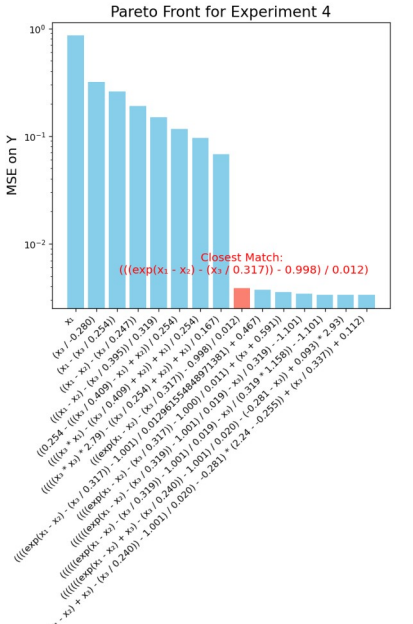


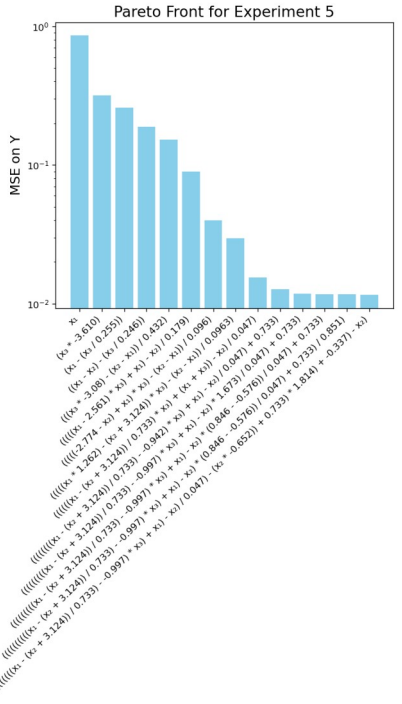
Symbolic Classification

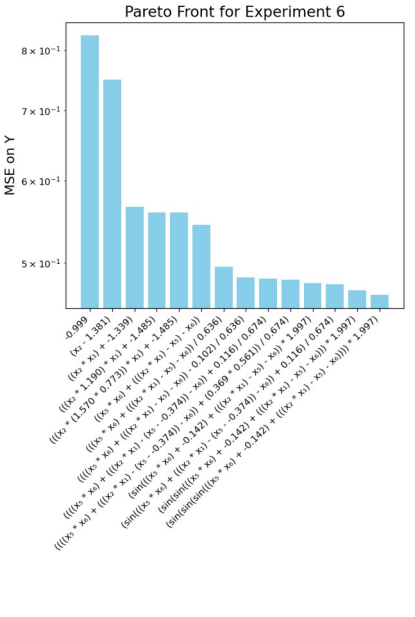








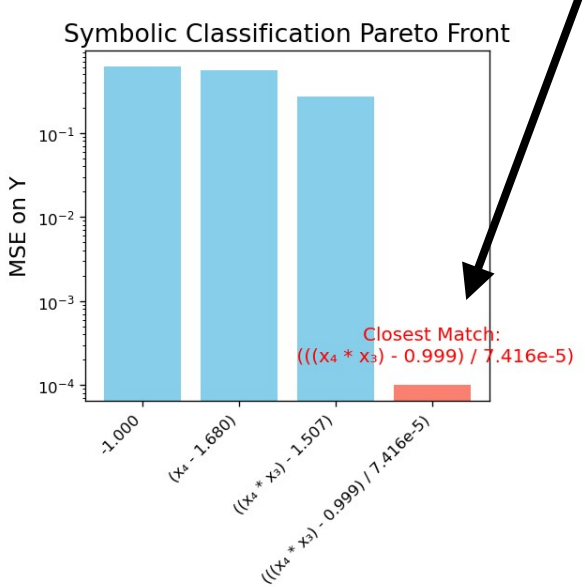


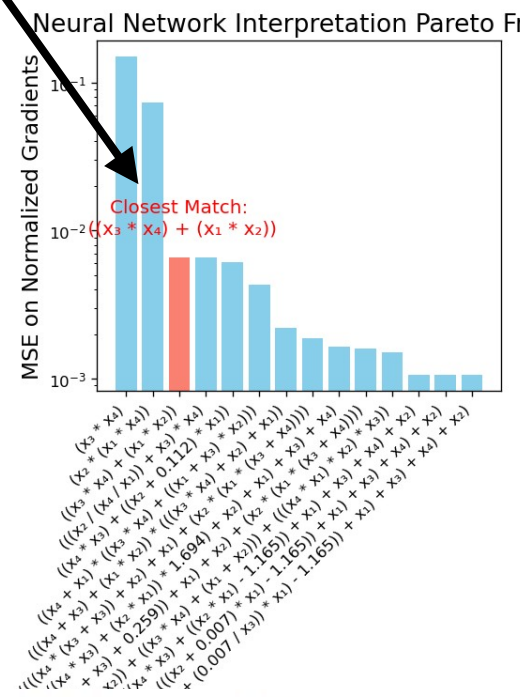


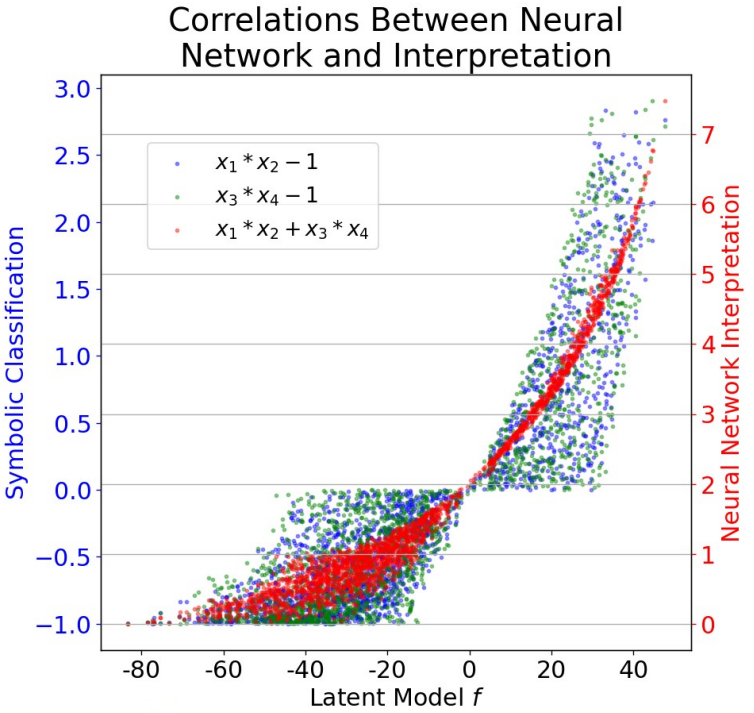
Symbolic Classification

We compare our interpretation framework to direct symbolic classification on a problem with multiple concepts that can be learned for a successful classification.

Neural Network lears combined concept while
Symbolic Classification learns only one







	Variables	Decision Formula
Experiment 7	4	$x_1x_2 > 1 \wedge x_3x_4 > 1$ vs $x_1x_2 < 1 \wedge x_3x_4 < 1$