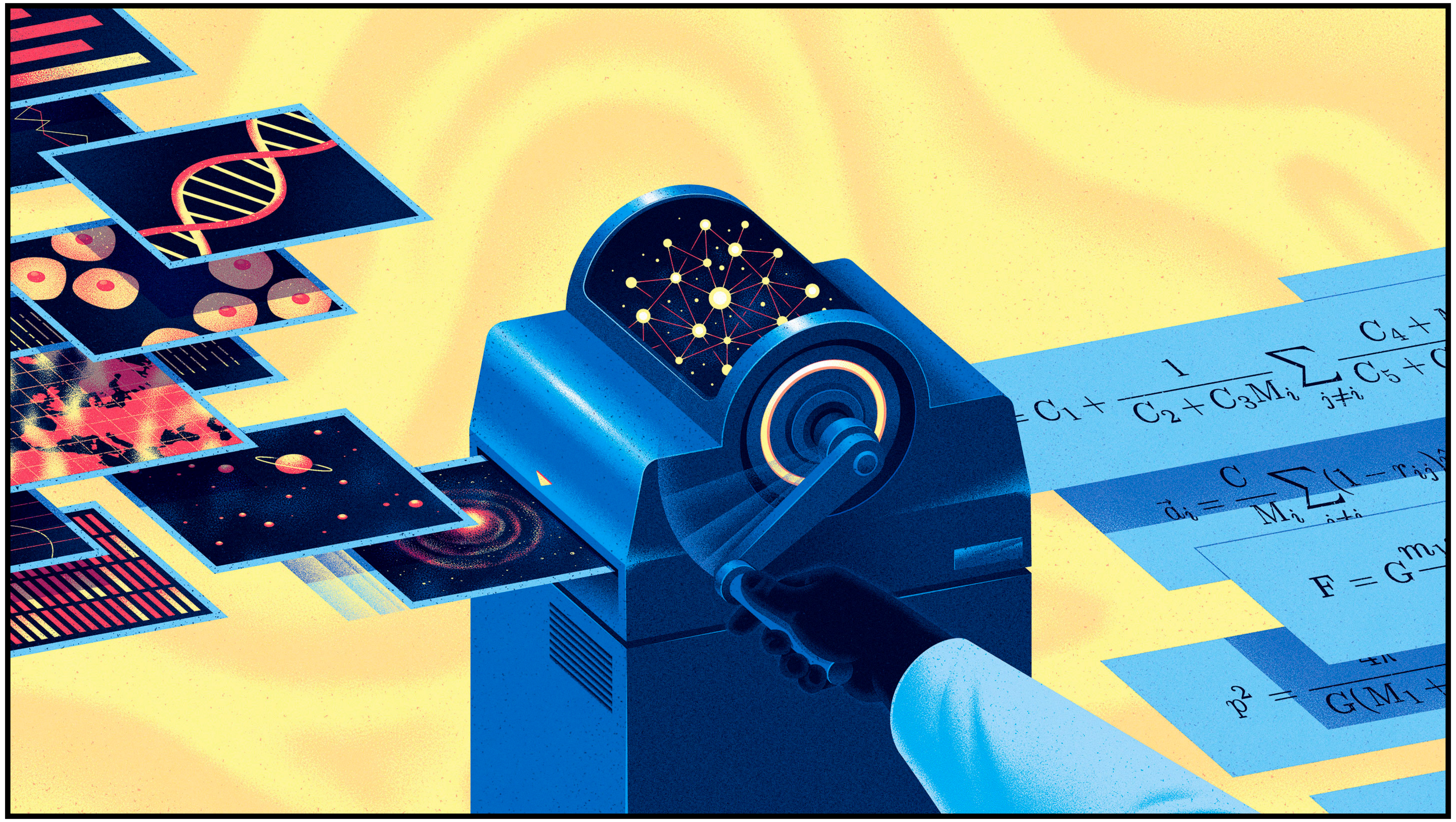
Symbolic Regression for Interpretable Machine Learning



Kouzou Sakai
for Quanta Magazine

Miles Cranmer

University of Cambridge
Assistant Prof, DAMTP & IoA

What I want:

What I want:

I want an AI scientist.

What I want:

I want an AI scientist.

What I want:

I want an AI scientist.

Machine learning research:

What I want:

I want an AI scientist.

Machine learning research:

- Driven mostly by computer vision/NLP benchmarks

What I want:

I want an AI scientist.

Machine learning research:

- Driven mostly by computer vision/NLP benchmarks
  - Motivated by industry interests, robotics

What I want:

I want an AI scientist.

Machine learning research:

- Driven mostly by computer vision/NLP benchmarks
  - Motivated by industry interests, robotics
  - Attempts to reach “human-level performance”

What I want:

I want an AI scientist.

Machine learning research:

- Driven mostly by computer vision/NLP benchmarks
  - Motivated by industry interests, robotics
  - Attempts to reach "human-level performance"
  - Narrow stepping stone benchmarks along the way.

What I want:

I want an AI scientist.

Machine learning research:

- Driven mostly by computer vision/NLP benchmarks
  - Motivated by industry interests, robotics
  - Attempts to reach “human-level performance”
  - Narrow stepping stone benchmarks along the way.

Problem:

What I want:

I want an AI scientist.

Machine learning research:

- Driven mostly by computer vision/NLP benchmarks
  - Motivated by industry interests, robotics
  - Attempts to reach “human-level performance”
  - Narrow stepping stone benchmarks along the way.

Problem:

- Much of ML applied to science takes such approaches, and replaces the datasets with scientific ones.

What I want:

What I want:

Instead of vision/language, want AI to reach human-level performance at ***research in the natural sciences***

What I want:

Instead of vision/language, want AI to reach human-level performance at ***research in the natural sciences***

What I want:

Instead of vision/language, want AI to reach human-level performance at ***research in the natural sciences***

What needs to happen?

What I want:

Instead of vision/language, want AI to reach human-level performance at ***research in the natural sciences***

What needs to happen?

- Natural science is not a regression problem. Need ***understanding***.

What I want:

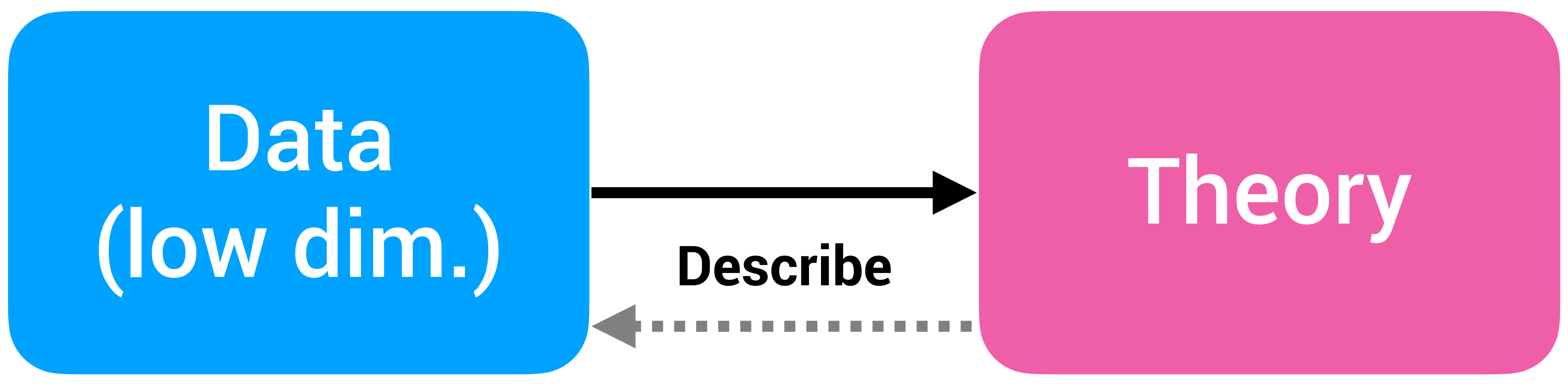
Instead of vision/language, want AI to reach human-level performance at ***research in the natural sciences***

What needs to happen?

- Natural science is not a regression problem. Need ***understanding***.
- We need to be able to use machine learning for discovering universal concepts and theories, and ***representing them in human language***

How?

Traditional approach to physics:



(May be summary statistics)

Empirical fit: Kepler's third law

$$P^2 \propto a^3$$

Empirical fit: Kepler’s third law

$$P^2 \propto a^3$$

$$\downarrow$$

Newton’s law of
gravitation,
to explain it

Empirical fit: Kepler's third law

$$P^2 \propto a^3$$

↓

Newton's law of gravitation, to explain it

Planck's law

$$B = \frac{2h\nu^3}{c^2}\left(\exp\left(\frac{h\nu}{k_\mathrm{B}T}\right) - 1\right)^{-1}$$

Empirical fit: Kepler's third law

$$P^2 \propto a^3$$

Newton's law of gravitation, to explain it

Planck's law

$$B = \frac{2h\nu^3}{c^2}\left(\exp\left(\frac{h\nu}{k_{\mathrm{B}}T}\right) - 1\right)^{-1}$$

(Partially)

Quantum mechanics, to explain it

Empirical fit:

Kepler's third law

$$P^2 \propto a^3$$

Newton's law of gravitation, to explain it

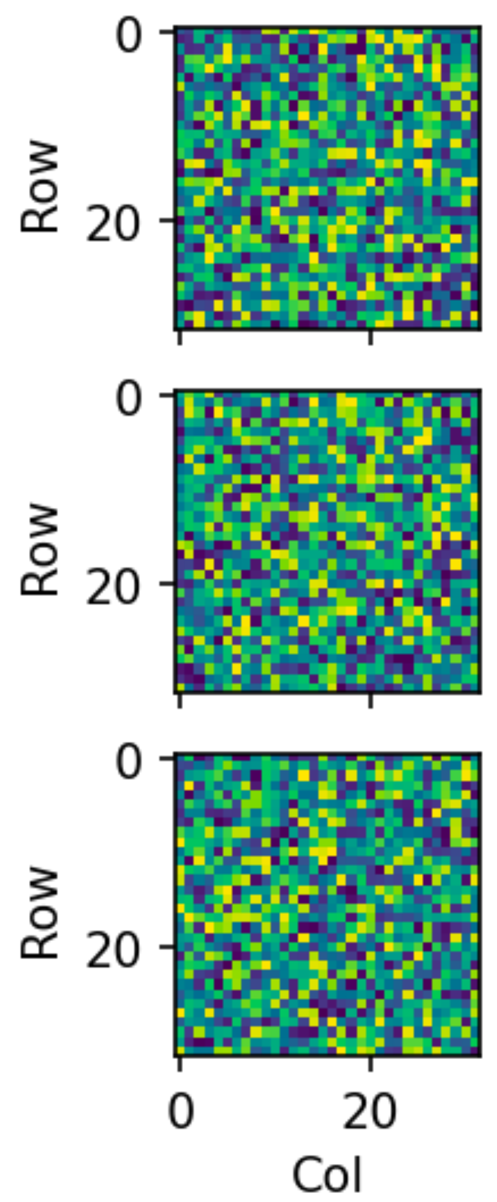
Planck's law

$$B = \frac{2h\nu^3}{c^2}\left(\exp\left(\frac{h\nu}{k_B T}\right) - 1\right)^{-1}$$

(Partially)

Quantum mechanics, to explain it

Neural Network Weights



Empirical fit:

Kepler's third law

$$P^2 \propto a^3$$

Newton's law of gravitation, to explain it

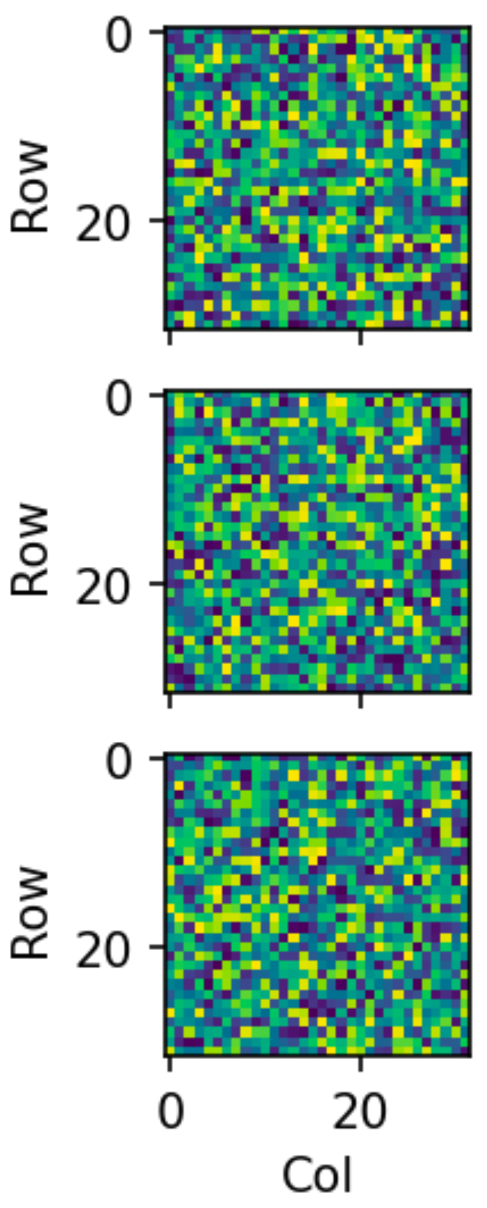
Planck's law

$$B = \frac{2h\nu^3}{c^2}\left(\exp\left(\frac{h\nu}{k_\mathrm{B}T}\right) - 1\right)^{-1}$$

(Partially)

Quantum mechanics, to explain it

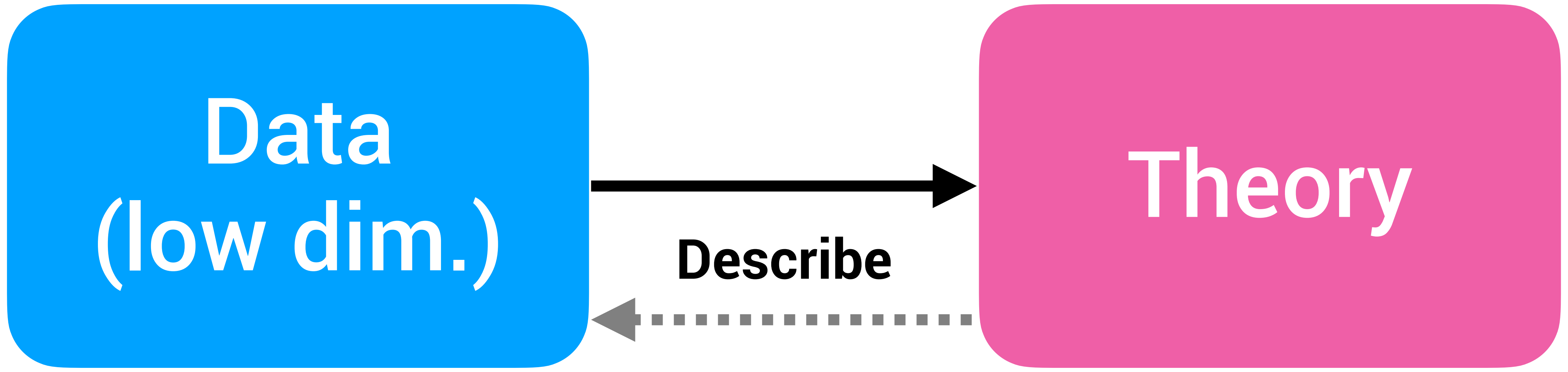
Neural Network Weights



???

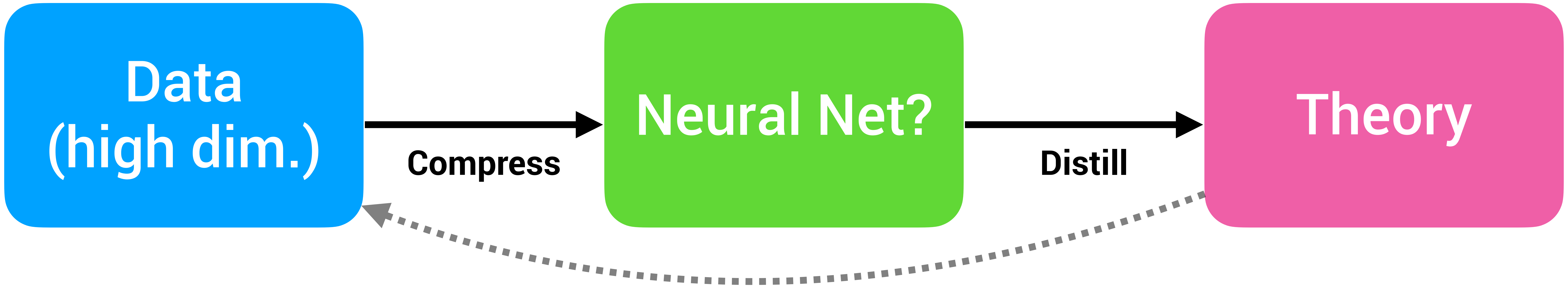
How?

Traditional approach to science:



(May be summary statistics)

Era of AI?



Key point

Key point

Neural nets trained on big datasets can find new insights.

Key point

Neural nets trained on big datasets can find new insights.

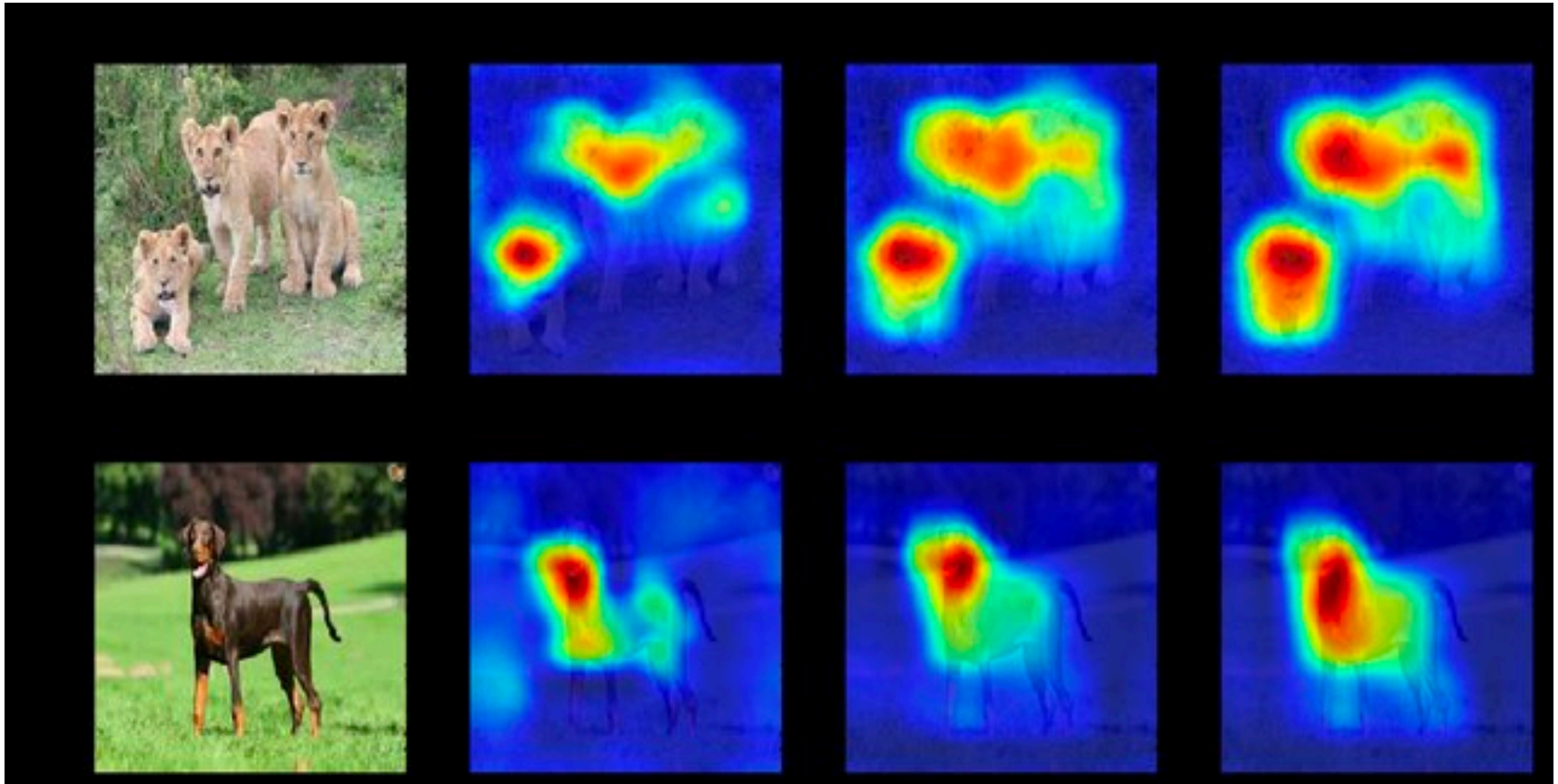
The remaining challenge is distilling the insights to our language.

Outline

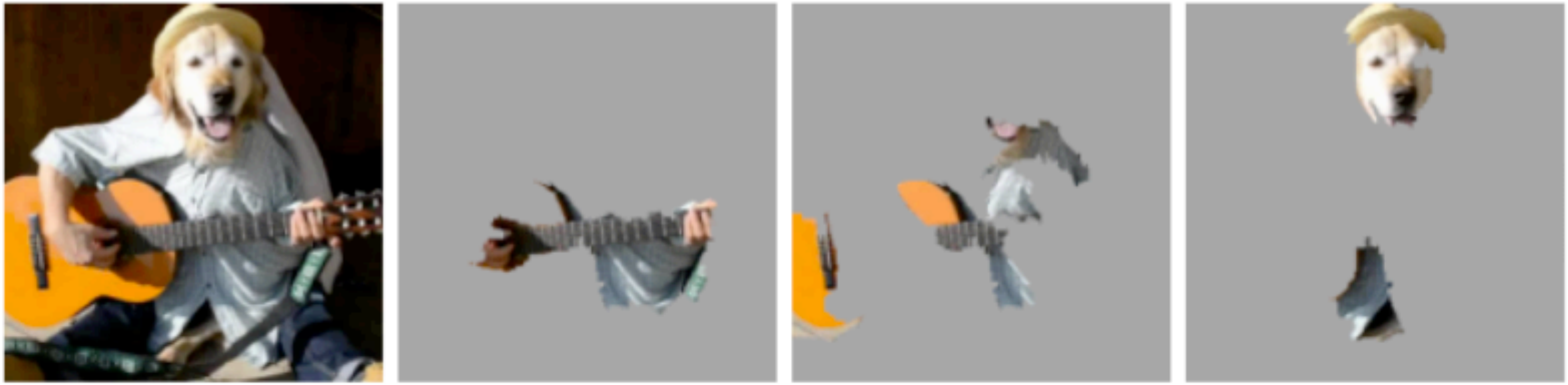
- Interpretability
- Symbolic regression
- Symbolic distillation
- Examples
- Future

CV/NLP strategy of interpretability

Typically involves **feature importance**



Omeiza et al., 2019



(a) Original Image (b) Explaining *Electric guitar* (c) Explaining *Acoustic guitar* (d) Explaining *Labrador*

Figure 4: Explaining an image classification prediction made by Google's Inception network, highlighting positive pixels. The top 3 classes predicted are "Electric Guitar" ($p = 0.32$)**, "Acoustic guitar"** ($p = 0.24$) **and "Labrador"** ($p = 0.21$)

Ribeiro et al., 2016

Science already has a modeling language

Science already has a modeling language

Computer Vision

Science already has a modeling language

Computer Vision

???

Science already has a modeling language

Computer Vision

???

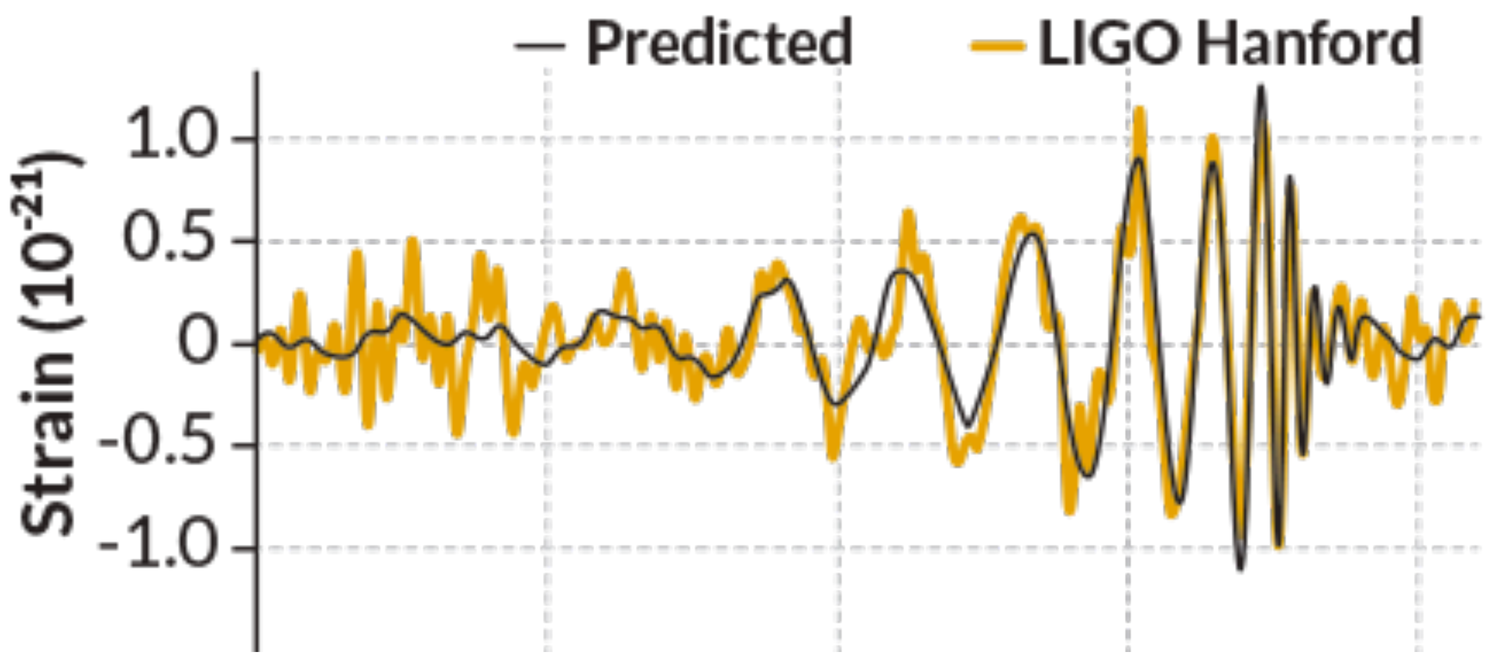
Science

Science already has a modeling language

Computer Vision

???

Science



$$h = \frac{2G}{c^4}\frac{1}{r}\frac{\partial^2 Q}{\partial t^2}$$

1-d motion, constant a

$$a = \frac{v_f - v_0}{t}$$

$$v_{\rm av} = \frac{v_0 + v_f}{2}, \quad (x_f - x_0) = v_{\rm av} t$$

$$(x_f - x_0) = v_0 t + \frac{1}{2} a t^2 = v_f t - \frac{1}{2} a t^2$$

$$\frac{1}{2} v_f^2 - \frac{1}{2} v_0^2 = a(x_f - x_0)$$

Projectile Motion

$$\text{Range} = \frac{v^2}{g} \sin 2\theta, \qquad \text{Max. height} = \frac{v_0^2}{2g} \sin^2 \theta$$

Momentum, Force and Impulse

$$p = mv, \; F = \frac{\Delta p}{\Delta t} = ma, I = F\Delta t = \Delta p$$

Work, Energy and Power

$$W = \vec{F} \cdot (\vec{r} - \vec{r}_0), KE = \frac{1}{2} m v^2, \; P = \frac{\Delta E}{\Delta t}$$

$\gamma = C_p/C_V$=5/3 for monotonic gas=7/5 for diatomic gas
$Q = T\Delta S$, $\Delta S > 0$
Engines: $\epsilon = W/Q_H < (T_H - T_L)/T_H < 1$
Refrigerators and heat pumps: $\epsilon = Q_L/W < T_L/(T_H - T_L)$

Simple Harmonic Motion and Waves

Spring: $F = -kx$, $PE = (1/2)kx^2$, $\omega = \sqrt{k/m}$
$f = \omega/(2\pi)$, $x(t) = A\cos(\omega t) + B\sin(\omega t)$
Pendulum: $T = 2\pi\sqrt{L/g}$
Waves: $y(x,t) = A\sin[2\pi(ft - x/\lambda + \delta)]$, $v = f\lambda$
$I = \text{const} A^2 f^2$, $I_2/I_1 = R_1^2/R_2^2$
Standing waves: $\lambda_n = 2L/n$
Strings: $v = \sqrt{T/\mu}$, Solid/Liquid: $v = \sqrt{B/\rho}$
Sound: $I = E/(A \cdot \Delta t) = \text{Power}/A$
$I_0 \equiv 10^{-12}$ W/m^2, Intensity in decibels=$10\log_{10}(I/I_0)$
Beat freq.=$|f_1 - f_2|$, Doppler:
$f_{\rm obs} = f_{\rm source}(V_{\rm sound} \pm v_{\rm obs})/(V_{\rm sound} \pm v_{\rm source})$
Pipes: same at both ends: $L = \lambda/2, \lambda, 3\lambda/2$
Pipes: open at only one end: $L = \lambda/4, 3\lambda/4 \cdots$

1-d motion, constant a

$$a = \frac{v_f - v_0}{t}$$

$$v_{\text{av}} = \frac{v_0 + v_f}{2}, \quad (x_f - x_0) = v_{\text{av}} t$$

$$(x_f - x_0) = v_0 t + \frac{1}{2} a t^2 = v_f t - \frac{1}{2} a t^2$$

$$\frac{1}{2} v_f^2 - \frac{1}{2} v_0^2 = a(x_f - x_0)$$

Projectile Motion

$$\text{Range} = \frac{v^2}{g} \sin 2\theta, \qquad \text{Max. height} = \frac{v_0^2}{2g} \sin^2 \theta$$

Momentum, Force and Impulse

$$p = mv, \; F = \frac{\Delta p}{\Delta t} = ma, I = F\Delta t = \Delta p$$

Work, Energy and Power

$$W = \vec{F} \cdot (\vec{r} - \vec{r}_0), KE = \frac{1}{2} m v^2, \; P = \frac{\Delta E}{\Delta t}$$

$\gamma = C_p/C_V$=5/3 for monotonic gas=7/5 for diatomic gas
$Q = T\Delta S$, $\Delta S > 0$
Engines: $\epsilon = W/Q_H < (T_H - T_L)/T_H < 1$
Refrigerators and heat pumps: $\epsilon = Q_L/W < T_L/(T_H - T_L)$

Simple Harmonic Motion and Waves
Spring: $F = -kx$, $PE = (1/2)kx^2$, $\omega = \sqrt{k/m}$
$f = \omega/(2\pi)$, $x(t) = A\cos(\omega t) + B\sin(\omega t)$
Pendulum: $T = 2\pi\sqrt{L/g}$
Waves: $y(x,t) = A\sin[2\pi(ft - x/\lambda + \delta)]$, $v = f\lambda$
$I = \text{const} A^2 f^2$, $I_2/I_1 = R_1^2/R_2^2$
Standing waves: $\lambda_n = 2L/n$
Strings: $v = \sqrt{T/\mu}$, Solid/Liquid: $v = \sqrt{B/\rho}$
Sound: $I = E/(A \cdot \Delta t) = \text{Power}/A$
$I_0 \equiv 10^{-12}$ W/m^2, Intensity in decibels=$10\log_{10}(I/I_0)$
Beat freq.=$|f_1 - f_2|$, Doppler:
$f_{\text{obs}} = f_{\text{source}}(V_{\text{sound}} \pm v_{\text{obs}})/(V_{\text{sound}} \pm v_{\text{source}})$
Pipes: same at both ends: $L = \lambda/2, \lambda, 3\lambda/2$
Pipes: open at only one end: $L = \lambda/4, 3\lambda/4 \cdots$

We should build interpretations in this existing language: mathematical expressions!

- For physical problems, even if it is not the “true" expression, analytic models can often generalize better than neural networks! (See M. Cranmer+2020)

- For physical problems, even if it is not the “true" expression, analytic models can often generalize better than neural networks! (See M. Cranmer+2020)

- For physical problems, even if it is not the “true" expression, analytic models can often generalize better than neural networks! (See M. Cranmer+2020)

- This is a type of inductive bias: searching for models represented as sparse combination of analytic operators hold geometrical and physical significance

- For physical problems, even if it is not the “true" expression, analytic models can often generalize better than neural networks! (See M. Cranmer+2020)

- This is a type of inductive bias: searching for models represented as sparse combination of analytic operators hold geometrical and physical significance

$+$	$\times$	$\exp$
Translation	**Length => Area => Volume**	**Solution to common ODE y’ ~ y**

Symbolic regression

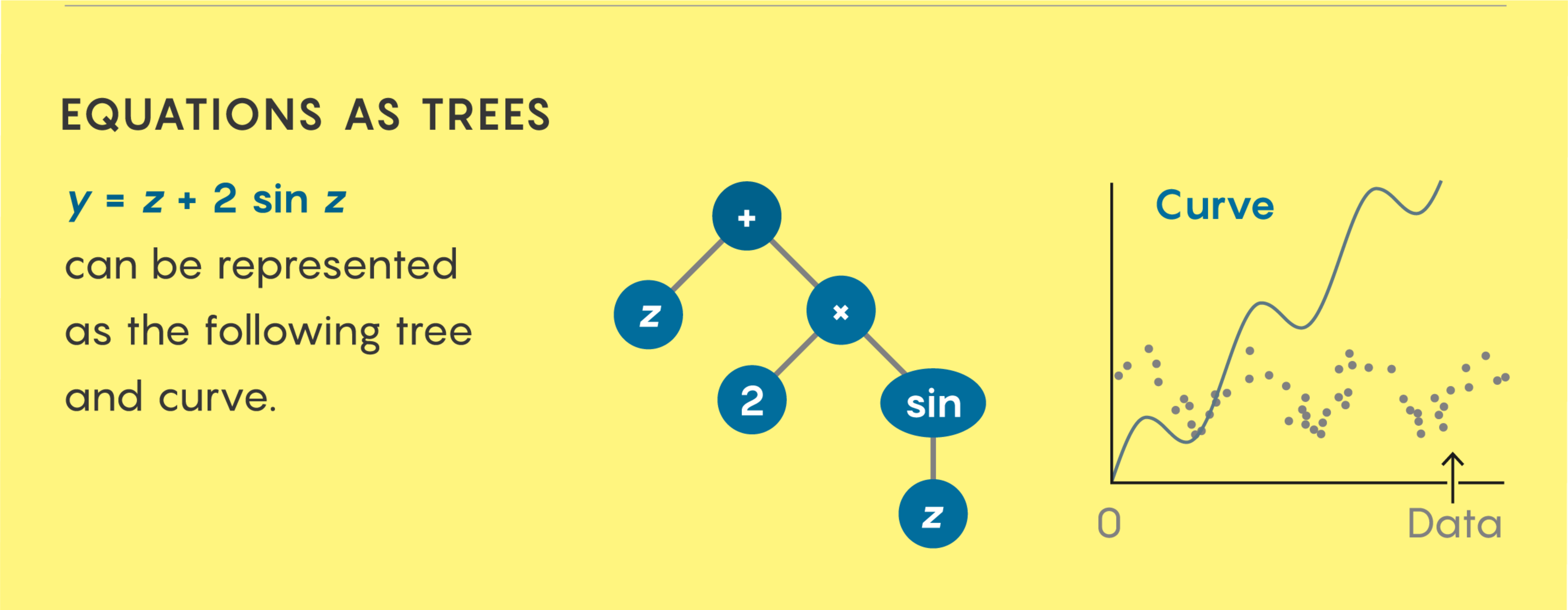
Symbolic regression is a machine learning task, where the objective is to find ***analytic expressions*** that optimize some objective.

- Popularized by Koza (1990s); and its use in science by Lipson (2000s)

Symbolic regression

Symbolic regression is a machine learning task, where the objective is to find ***analytic expressions*** that optimize some objective.

- Popularized by Koza (1990s); and its use in science by Lipson (2000s)

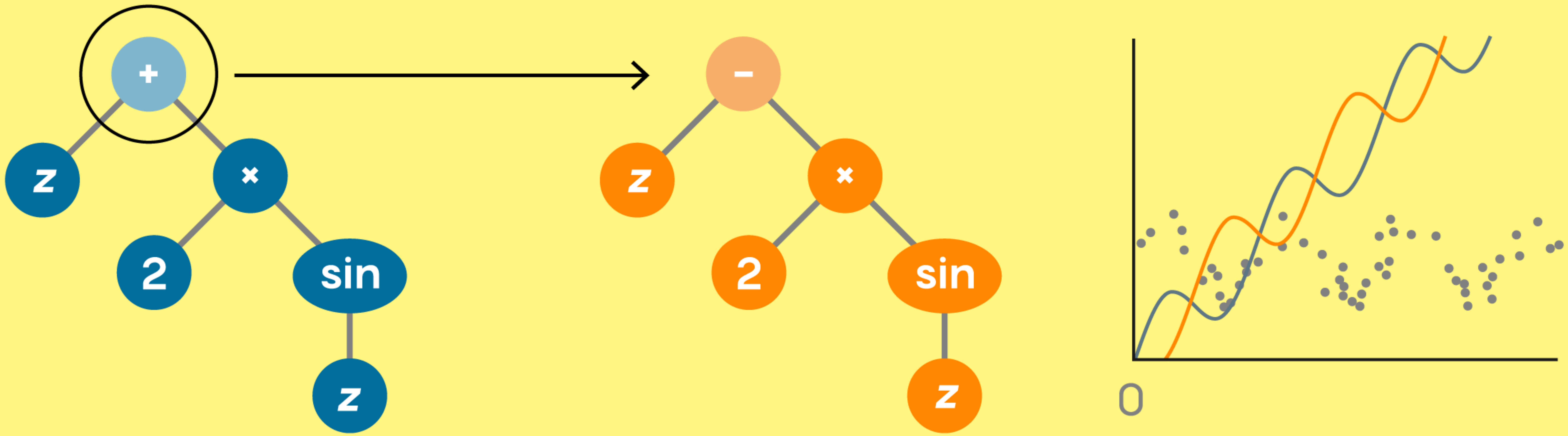


Kouzou Sakai
for Quanta Magazine

SOTA = genetic algorithm

MUTATION

The algorithm might mutate one node of the tree.

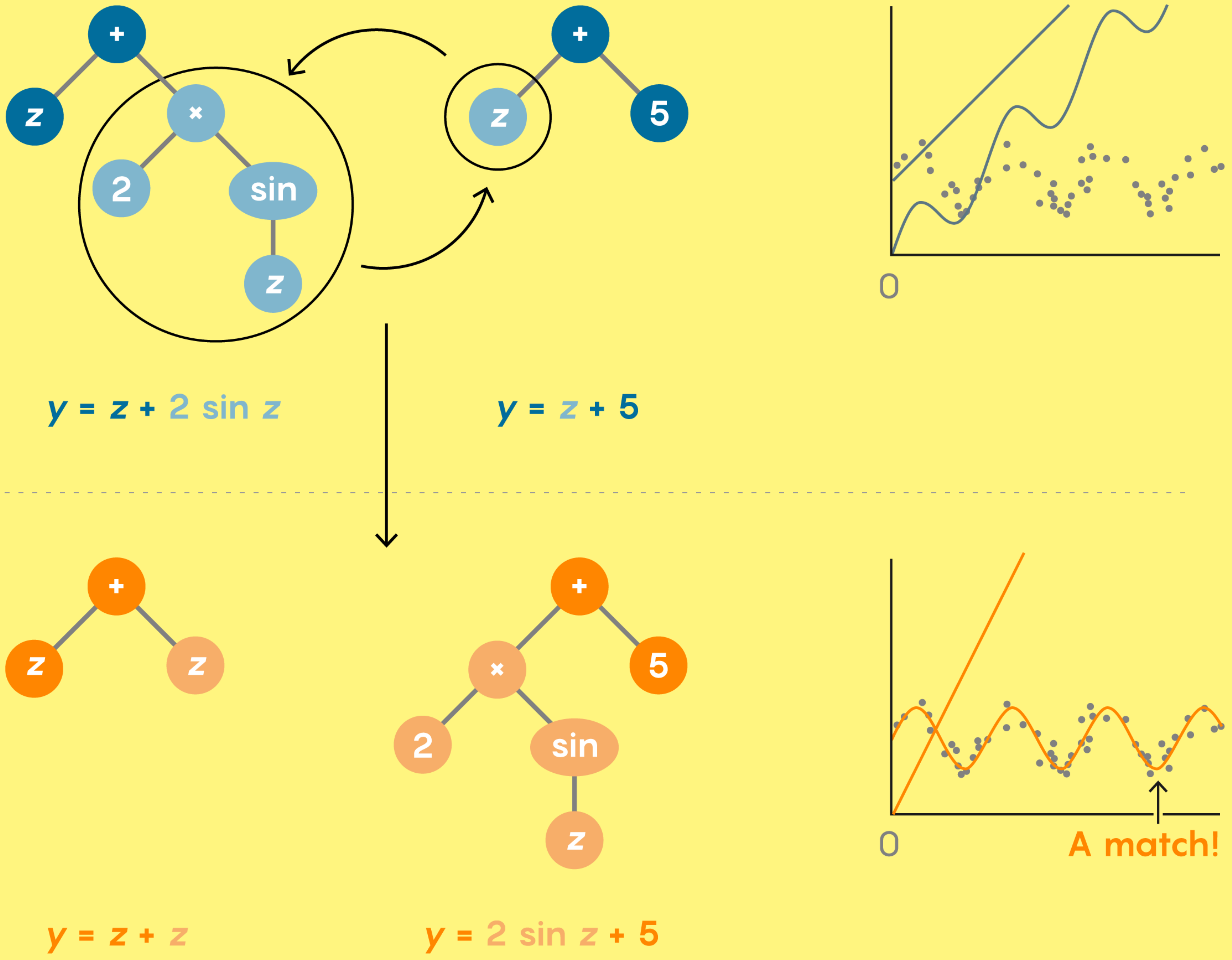


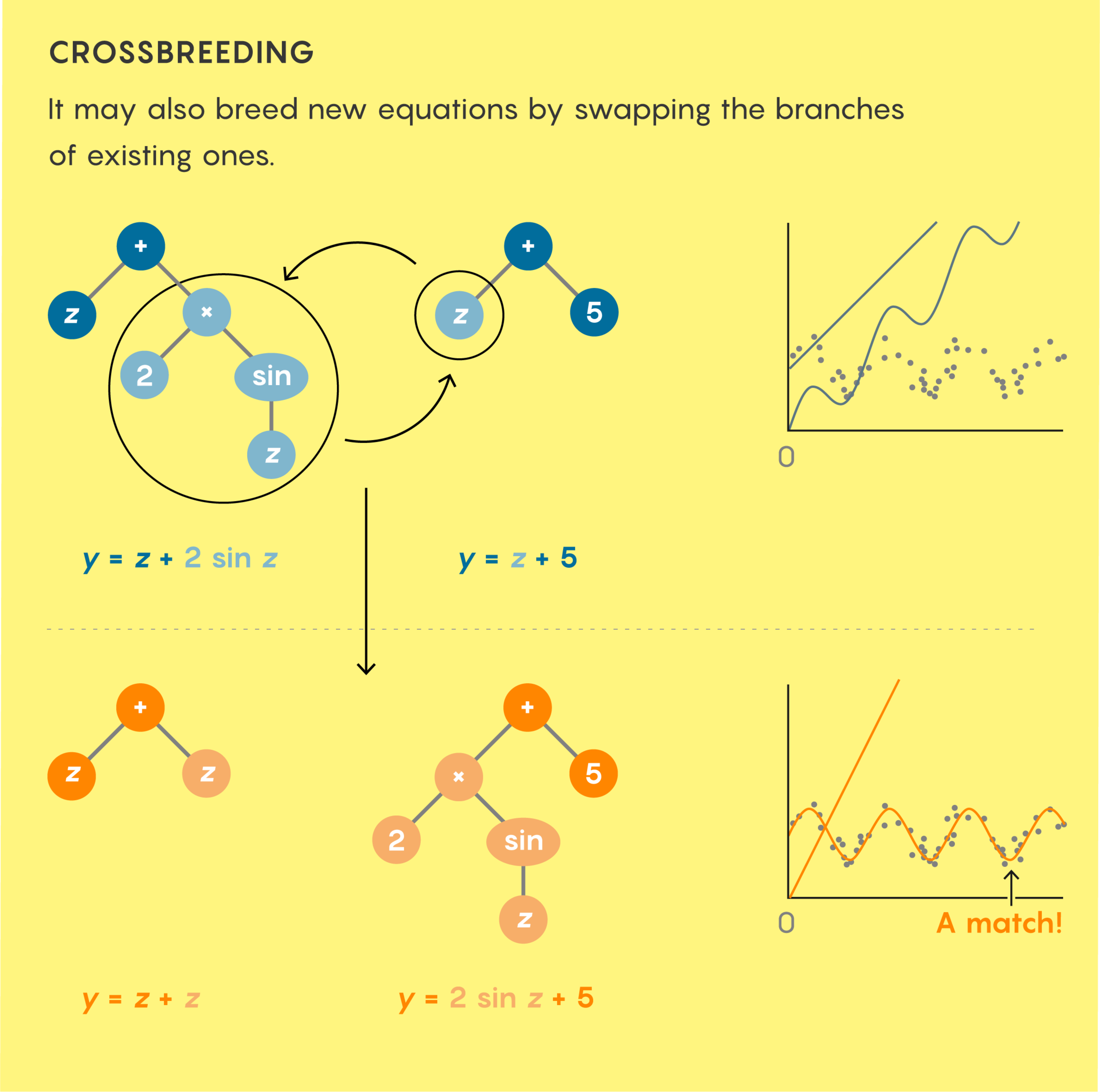
$y = z + 2 \sin z$

$y = z - 2 \sin z$

CROSSBREEDING

It may also breed new equations by swapping the branches of existing ones.





Jointly optimize
accuracy & complexity

Complexity is user-defined,
but usually = number of nodes

High-level open-source frameworks:

github.com/MilesCranmer/SymbolicRegression.jl/

SymbolicRegression.jl Public

Distributed High-Performance Symbolic Regression in Julia

Julia 449 45

= MLJ interface
(main search code)

github.com/MilesCranmer/PySR/

PySR Public

High-Performance Symbolic Regression in Python and Julia

Python 1.4k 141

= Scikit-Learn wrapper

Build your own symbolic regression algorithm!

github.com/SymbolicML/DynamicExpressions.jl/

SymbolicML/DynamicExpressions.jl Public

Ridiculously fast symbolic expressions

Julia ☆ 68 4

DynamicExpressions.jl

github.com/SymbolicML/DynamicQuantities.jl/

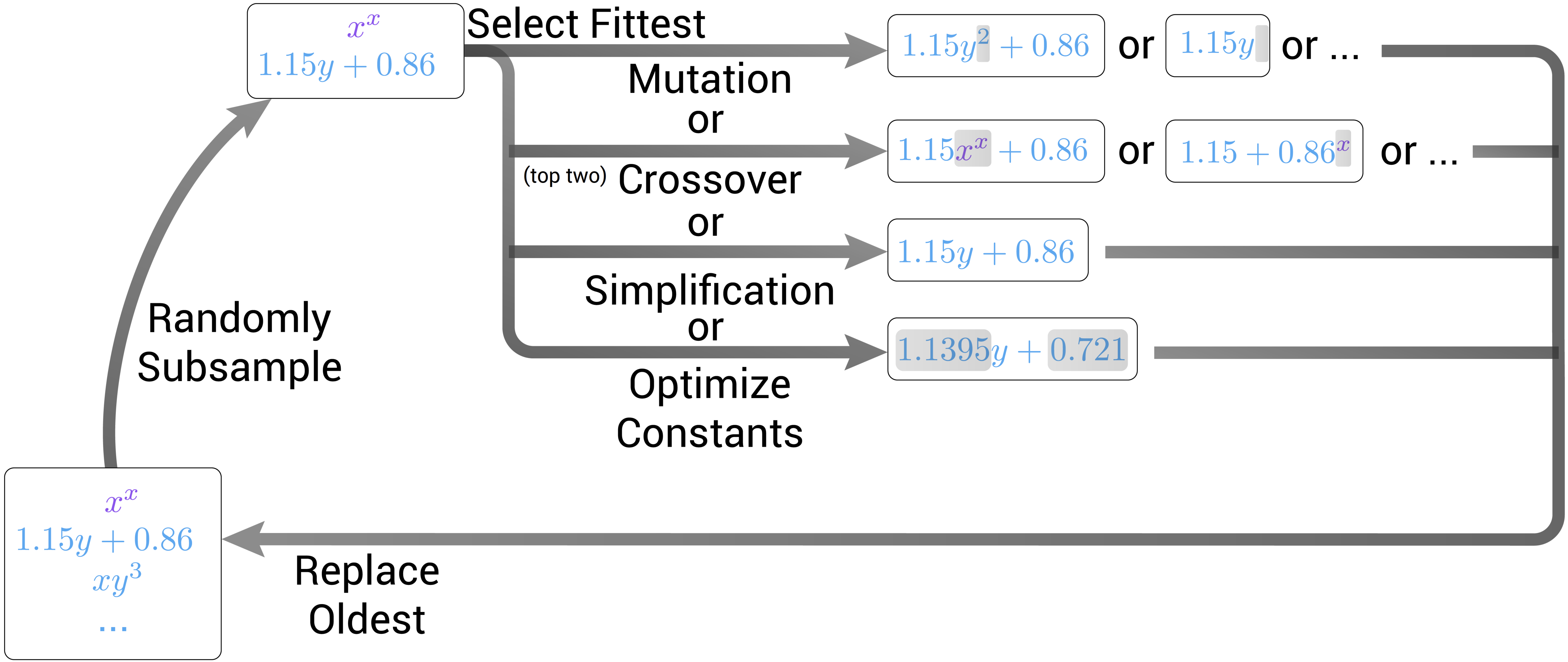
SymbolicML/DynamicQuantities.jl Public

Lightweight + fast physical quantities in Julia

Julia ☆ 44 3

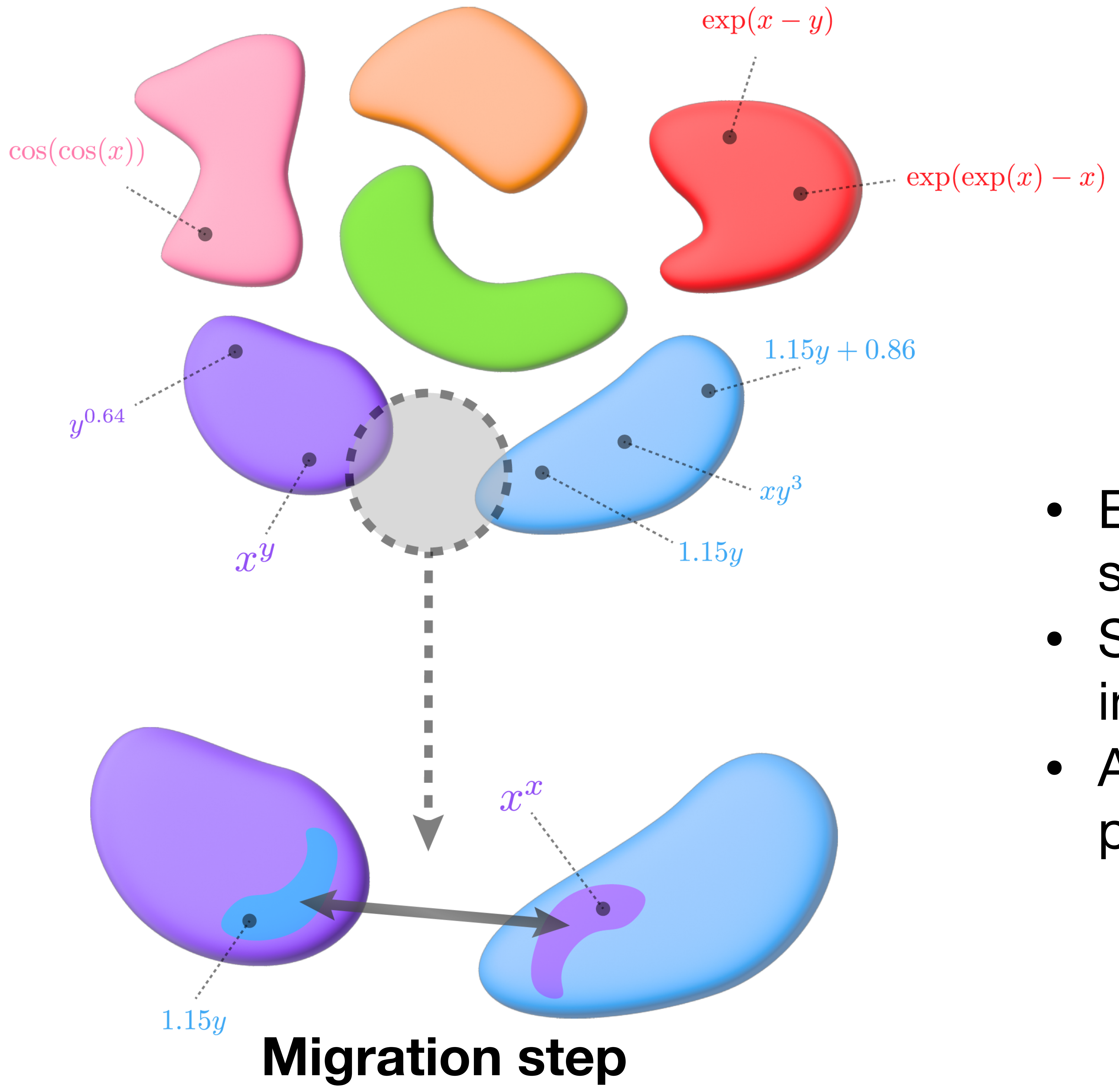
DynamicQuantities.jl

Age-Regularized Multi-Population Evolution in PySR



Cranmer, 2023 - arxiv.org/abs/2305.01582

Model discovery at scale:



- Each island evolves independently on a single core.
- Scale up to ~1000s of cores (=1000s of independent populations)
- Asynchronous migration between populations

```
julia>
```

```
julia>
```

Python API

```python
from pysr import PySRRegressor

model = PySRRegressor(
    niterations=40,  # < Increase me for better results
    binary_operators=["+", "*"],
    unary_operators=[
        "cos",
        "exp",
        "sin",
        "inv(x) = 1/x",
        # ^ Custom operator (julia syntax)
    ],
    extra_sympy_mappings={"inv": lambda x: 1 / x},
    # ^ Define operator for SymPy as well
    loss="loss(prediction, target) = (prediction - target)^2",
    # ^ Custom loss function (julia syntax)
)
```

Dimensional constraints

To do this, we need to use the format of DynamicQuantities.jl.

```
# Get numerical arrays to fit:
X = pd.DataFrame(dict(
    M=M.to("M_sun").value,
    m=m.to("kg").value,
    r=r.to("R_earth").value,
))
y = F.value

model.fit(
    X,
    y,
    X_units=["Constants.M_sun", "kg", "Constants.R_earth"],
    y_units="kg * m / s^2"
)
```

Custom objectives

"Can I make it so that my equation has exactly 2 sinusoids?" Yes!

```julia
function my_objective(tree::Node{T}, dataset::Dataset{T,L}, options::Options) where {T,L}
    prediction, flag = eval_tree_array(tree, dataset.X, options)
    !flag && return convert(L, Inf)

    sin_idx = 1  # Change if you change the order you put `sin`, or use `findfirst(==(sin), options.operators.unaops)::I

    prediction_loss = sum(i -> abs(prediction[i] - dataset.y[i])^3, eachindex(dataset.y)) / length(dataset.y)

    # Count number of sinusoids:
    num_sins = count(node -> node.degree == 1 && node.op == sin_idx, tree)

    # Add penalty of 10 for every sinusoids off from 2:
    regularization = convert(L, 10 * abs(num_sins - 2))

    return prediction_loss + regularization
end
```

https://arxiv.org/abs/2305.01582

		PySR	Eureqa	GPLearn	AI Feynman	Operon	DSR	PySINDy	EQL	QLattice	SR-Transformer	GP-GOMEA	Symbolic Distillation*
Scalability	Compiled	✓	✓	✗	✗	✓	✗	✗	✓	✓	✓	✓	-
	Multi-core	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓
	Multi-node	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	-
	GPU-capable	✗	✗	✗	∗I	✗	✗	✓	✓	✗	✓	✗	✓
Practicality	No pre-training	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	-
	Denoising	✓	✓	✗	✗	✗	✗	∗II	✗	?	✗	✗	✓
	Feature selection	✓	✓	✗	✓	✗	✓	∗II	✗	✓	✗	✗	✓
	Differential equations	✗	✓	✗	✗	✗	✗	✓	✓	✗	✗	✗	✓
	High-dimensional	✗	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗	✓
	Full Pareto curve	✓	✓	✓	✗	✓	✓	∗II	✗	✓	✗	✓	✗
Interfacing	API	✓	✗	✓	✗	✓	✓	✓	✓	✓	✓	✓	-
	SymPy Interface	✓	✗	✗	✓	✓	✗	✗	✗	✓	✓	✓	-
	Deep Learning export	✓	✗	✗	✗	✗	✗	✗	∗III	✗	∗III	✗	-
Extensibility	Expressivity score	4	5	4	3	3	3	1b	2	3	1a	3	6
	Open-source	✓	✗	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓
	Real Constants	✓	✓	✓	✗	✓	✓	∗II	✓	✓	✓	✓	-
	Custom operators	✓	✗	✓	✗	✗	✗	∗II	✗	✗	✗	✗	-
	Discontinuous operators	✓	✓	✓	✗	✗	✗	∗II	✗	✗	✗	✗	-
	Custom losses	✓	✓	✓	✗	✗	✓	✗	✗	✗	✗	✗	✓
	Symbolic Constraints	✓	✗	✓	✗	✗	✓	✗	✗	✗	✗	✗	✓
	Custom complexity	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	-
	Custom types	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
-	Citation	[self]	[11]	-	[73]	[44]	[27]	[74]	[34]	[75]	[30]	[21]	[23]
	Code	🔓	🔒	🔓	🔓	🔓	🔓	🔓	🔓	🔒	🔓	🔓	🔓

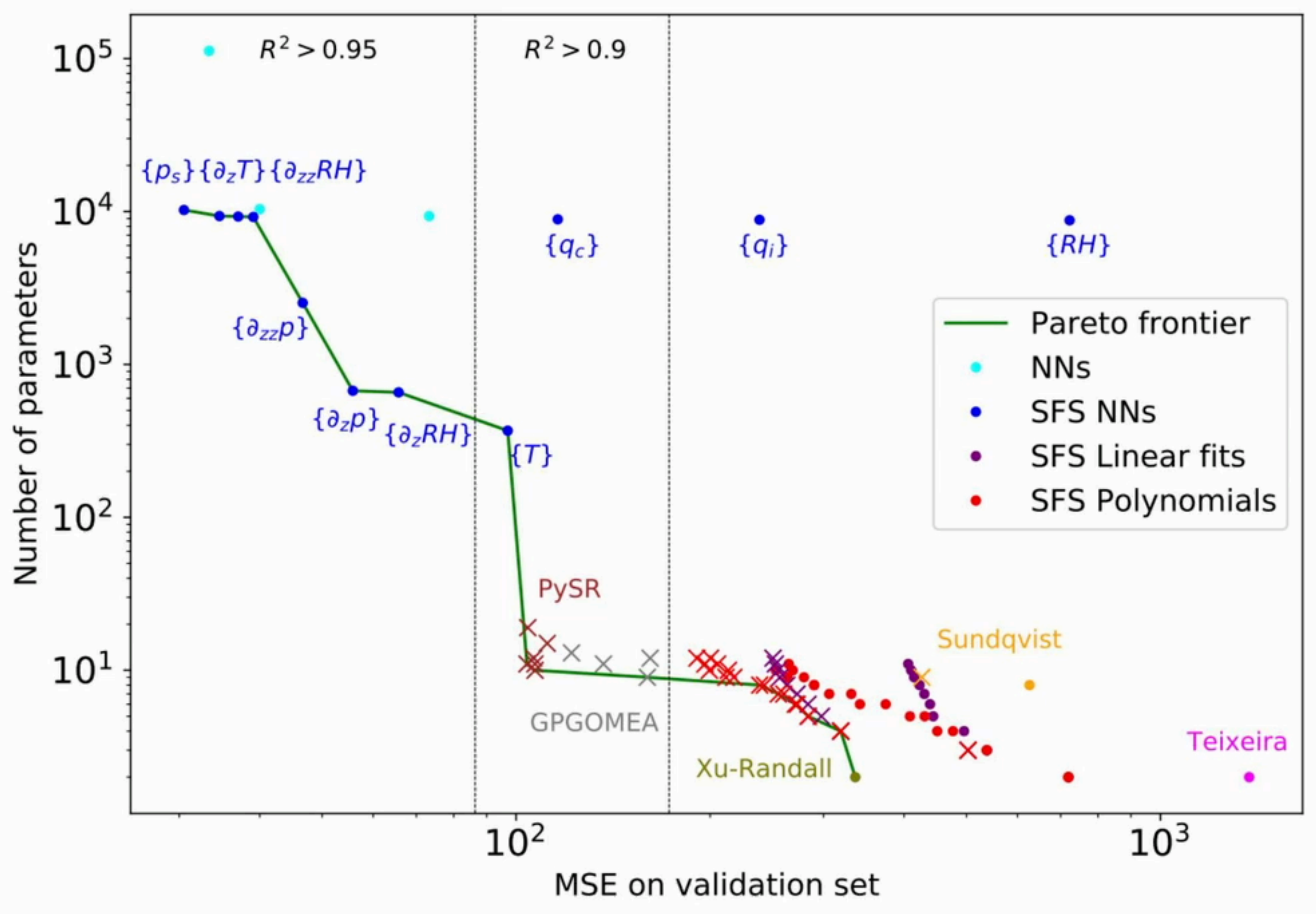
	PySR	Operon	DSR	EQL	QLattice	SR-Transformer
Hubble	5/5 (5, 0, 0, 0)	0/5 (0, 5, 0, 0)	1/5 (1, 0, 4, 0)	0/5 (0, 0, 0, 5)	0/5 (0, 5, 0, 0)	0/5 (0, 0, 0, 5)
Kepler	5/5 (5, 0, 0, 0)	0/5 (0, 5, 0, 0)	4/5 (4, 1, 0, 0)	0/5 (0, 0, 2, 3)	0/5 (0, 0, 0, 5)	0/5 (0, 0, 0, 5)
Newton	5/5 (5, 0, 0, 0)	1/5 (1, 2, 0, 2)	1/5 (1, 0, 4, 0)	0/5 (0, 0, 5, 0)	0/5 (0, 0, 0, 5)	0/5 (0, 0, 0, 5)
Planck	0/5 (0, 0, 0, 5)	0/5 (0, 0, 0, 5)	0/5 (0, 0, 1, 4)	0/5 (0, 0, 5, 0)	0/5 (0, 0, 0, 5)	0/5 (0, 0, 0, 5)
Leavitt	5/5 (5, 0, 0, 0)	0/5 (0, 0, 0, 5)	5/5 (5, 0, 0, 0)	0/5 (0, 0, 5, 0)	0/5 (0, 0, 0, 5)	0/5 (0, 0, 0, 5)
Schechter	5/5 (5, 0, 0, 0)	5/5 (5, 0, 0, 0)	5/5 (5, 0, 0, 0)	0/5 (0, 0, 4, 1)	5/5 (5, 0, 0, 0)	0/5 (0, 0, 0, 5)
Bode	5/5 (5, 0, 0, 0)	3/5 (3, 0, 0, 2)	1/5 (1, 0, 3, 1)	0/5 (0, 0, 4, 1)	0/5 (0, 0, 0, 5)	0/5 (0, 0, 0, 5)
Ideal Gas	5/5 (5, 0, 0, 0)	0/5 (0, 0, 0, 5)	5/5 (5, 0, 0, 0)	0/5 (0, 0, 4, 1)	0/5 (0, 0, 0, 5)	0/5 (0, 0, 0, 5)
Rydberg	0/5 (0, 0, 0, 5)	0/5 (0, 0, 0, 5)	0/5 (0, 0, 5, 0)	0/5 (0, 0, 0, 5)	0/5 (0, 0, 0, 5)	0/5 (0, 0, 0, 5)

Selection of user-contributed publications that have used symbolic distillation/PySR/ SymbolicRegression.jl:

astroautomata.com/PySR/papers

Below is a showcase of papers which have used PySR to discover or rediscover a symbolic model. These are sorted by the date of release, with most recent papers at the top.

If you have used PySR in your research, please submit a pull request to add your paper to this file.



Data-Driven Equation Discovery of a Cloud Cover Parameterization

Arthur Grundner [1,2], Tom Beucler [3], Pierre Gentine [2,3], Veronika Eyring [1,4]

[1]Institut für Physik der Atmosphäre, Deutsches Zentrum für Luft- und Raumfahrt, [2]Center for Learning the Earth with Artificial Intelligence And Physics, Columbia University, [3]Institute of Earth Surface Dynamics, University of Lausanne, [4]Institute of Environmental Physics, University of Bremen

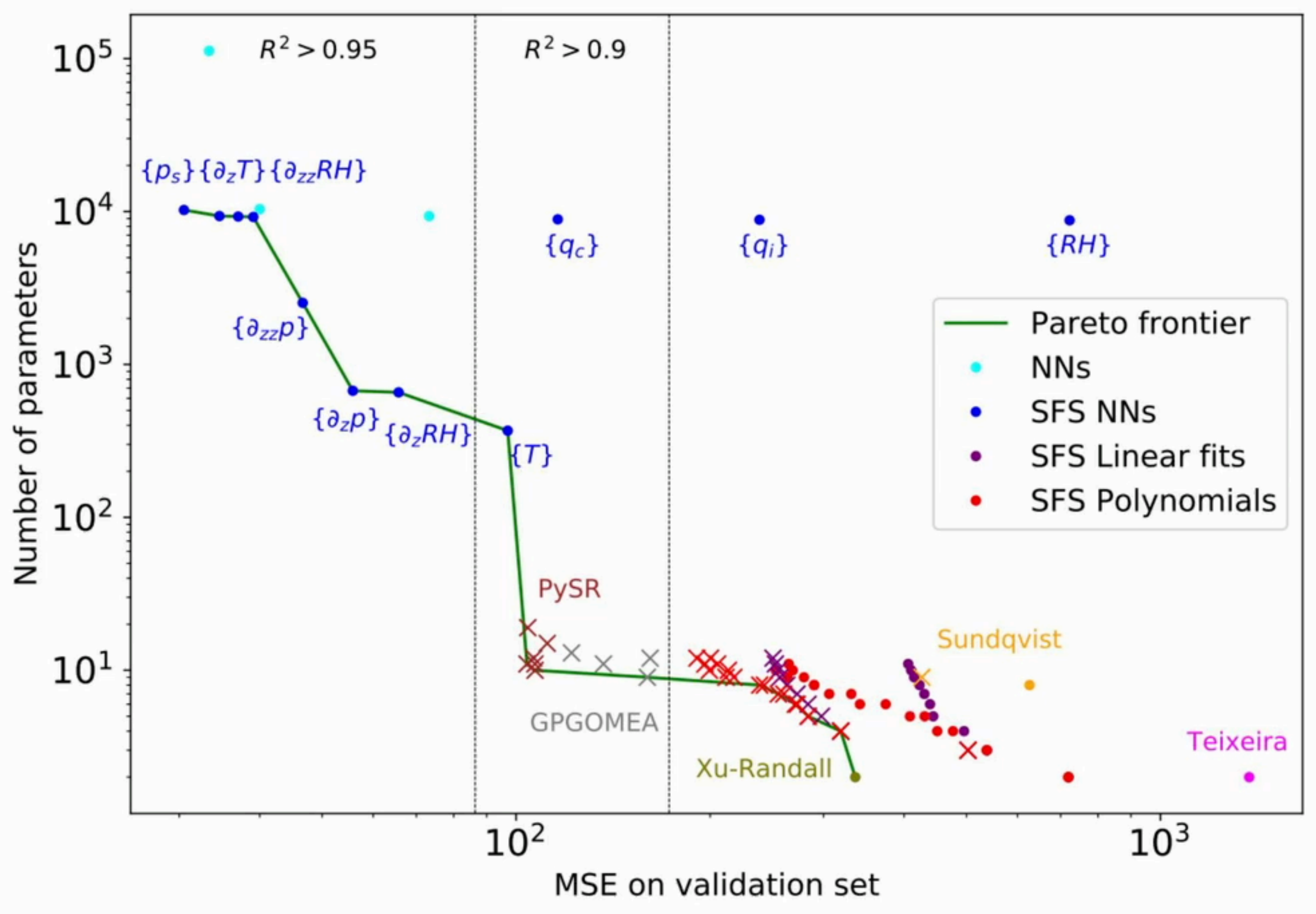
Abstract: A promising method for improving the representation of clouds in climate models, and

Selection of user-contributed publications that have used symbolic distillation/PySR/ SymbolicRegression.jl:

astroautomata.com/PySR/papers

Below is a showcase of papers which have used PySR to discover or rediscover a symbolic model. These are sorted by the date of release, with most recent papers at the top.

If you have used PySR in your research, please submit a pull request to add your paper to this file.



Data-Driven Equation Discovery of a Cloud Cover Parameterization

Arthur Grundner [1,2], Tom Beucler [3], Pierre Gentine [2,3], Veronika Eyring [1,4]

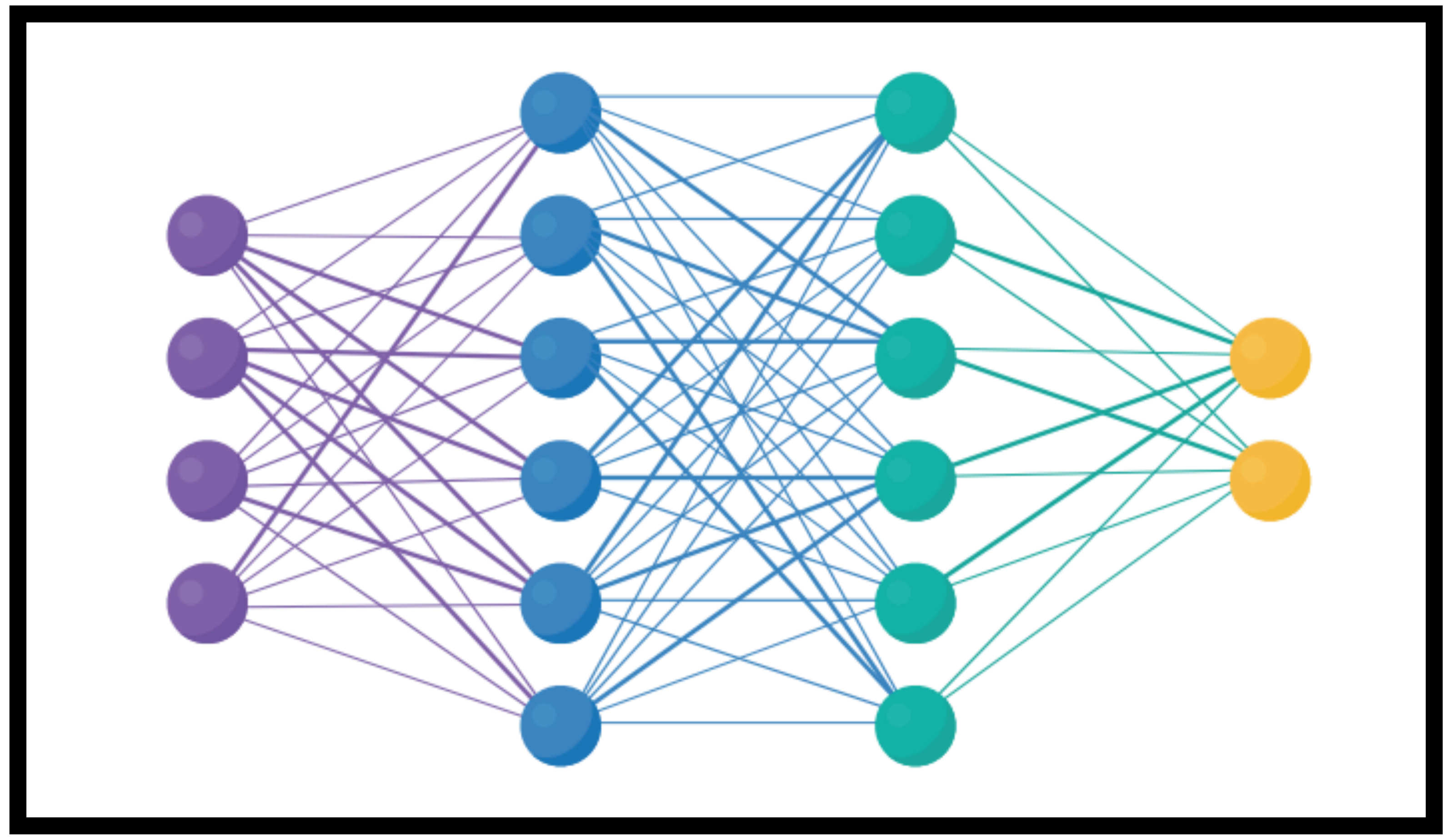
[1]Institut für Physik der Atmosphäre, Deutsches Zentrum für Luft- und Raumfahrt, [2]Center for Learning the Earth with Artificial Intelligence And Physics, Columbia University, [3]Institute of Earth Surface Dynamics, University of Lausanne, [4]Institute of Environmental Physics, University of Bremen

Abstract: A promising method for improving the representation of clouds in climate models, and

We can use Symbolic Regression to **Distill** a Neural Network into an Analytic Expression

How this works:

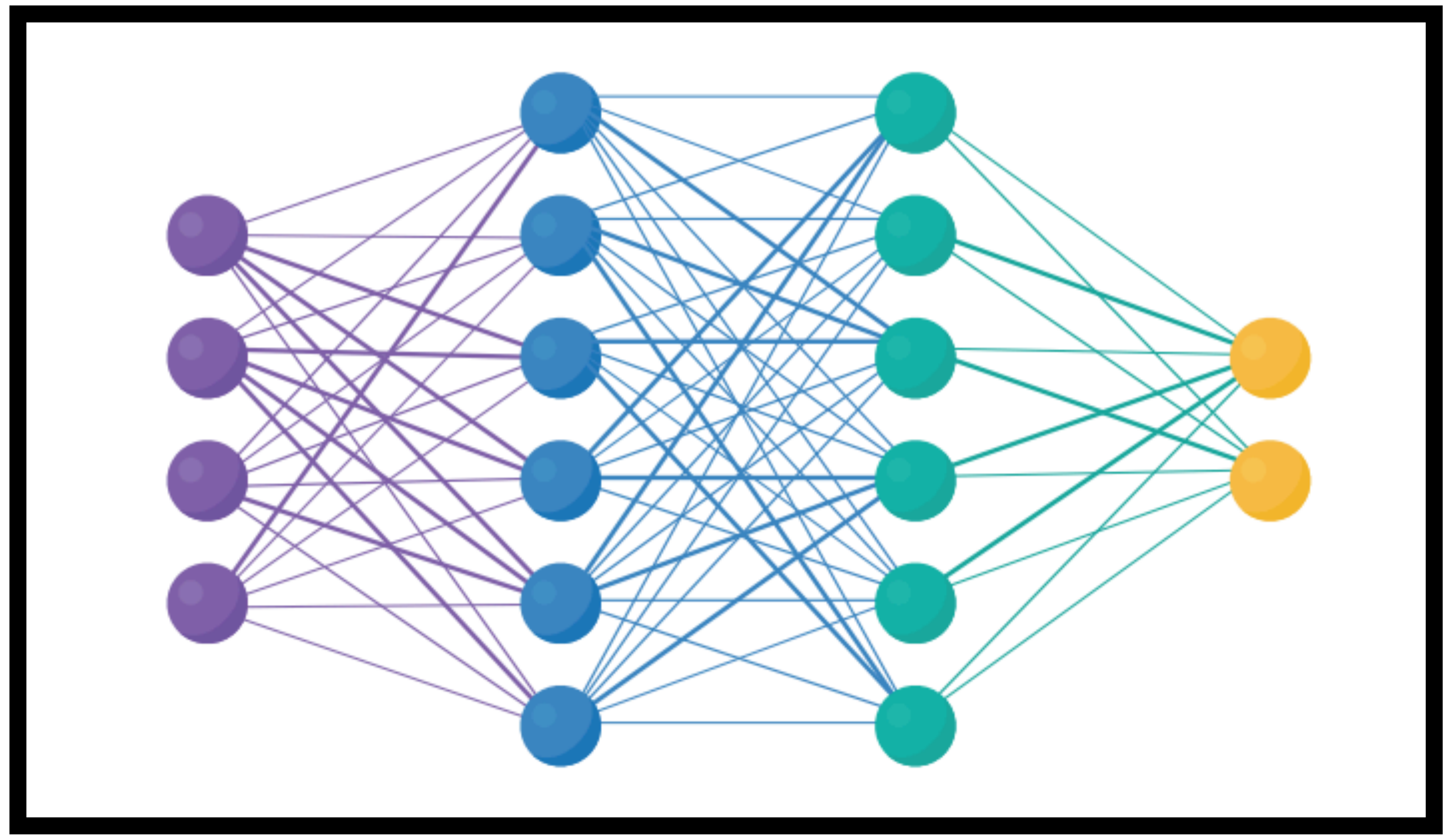
Cranmer et al., 2019, 2020 – Work with: Alvaro Sanchez-Gonzalez, Shirley Ho, Peter Battaglia, Kyle Cranmer, David Spergel, Rui Xu

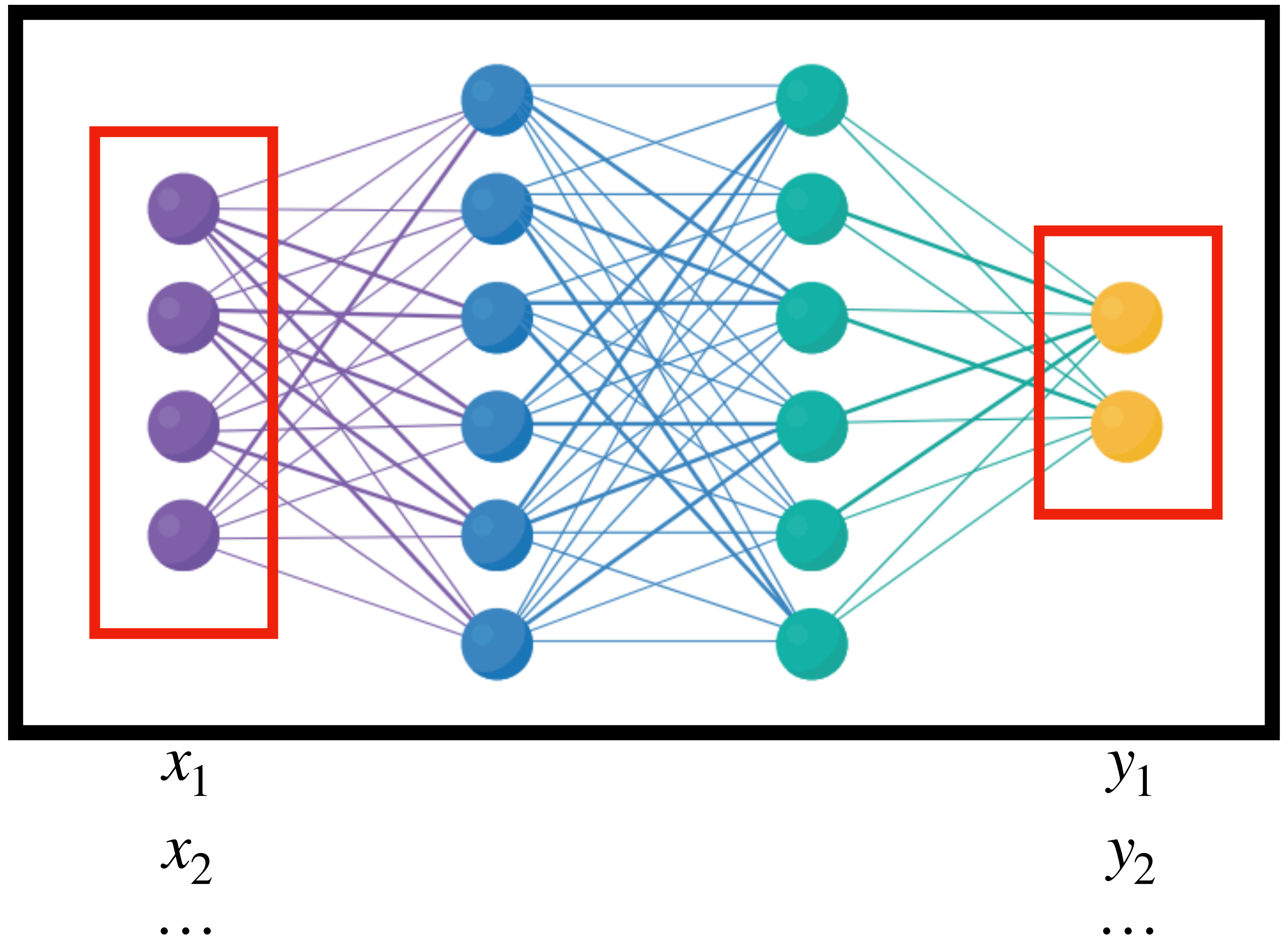


1. **Train NN normally, and freeze parameters.**

How this works:

Cranmer et al., 2019, 2020 – Work with: Alvaro Sanchez-Gonzalez, Shirley Ho, Peter Battaglia, Kyle Cranmer, David Spergel, Rui Xu



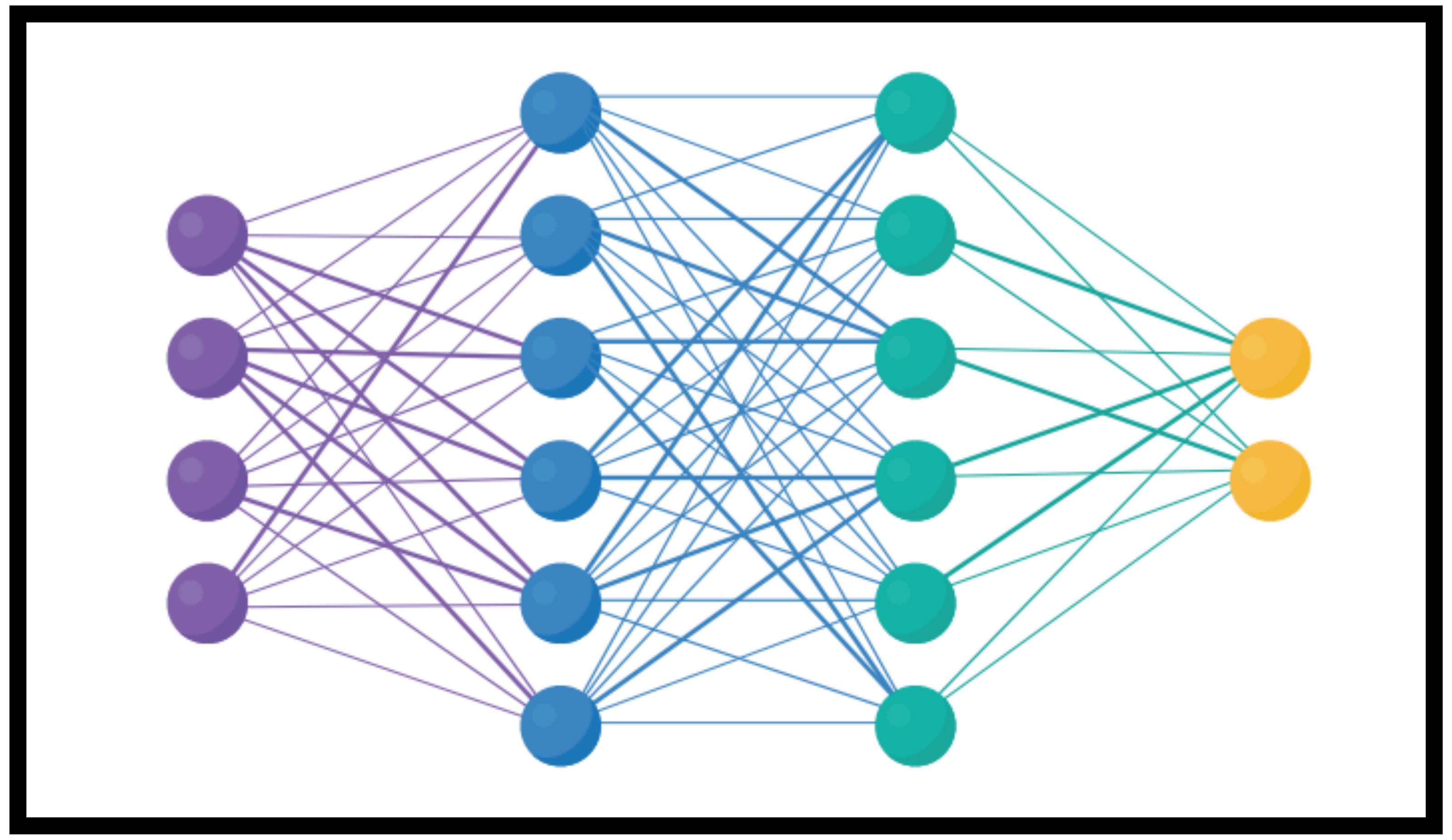


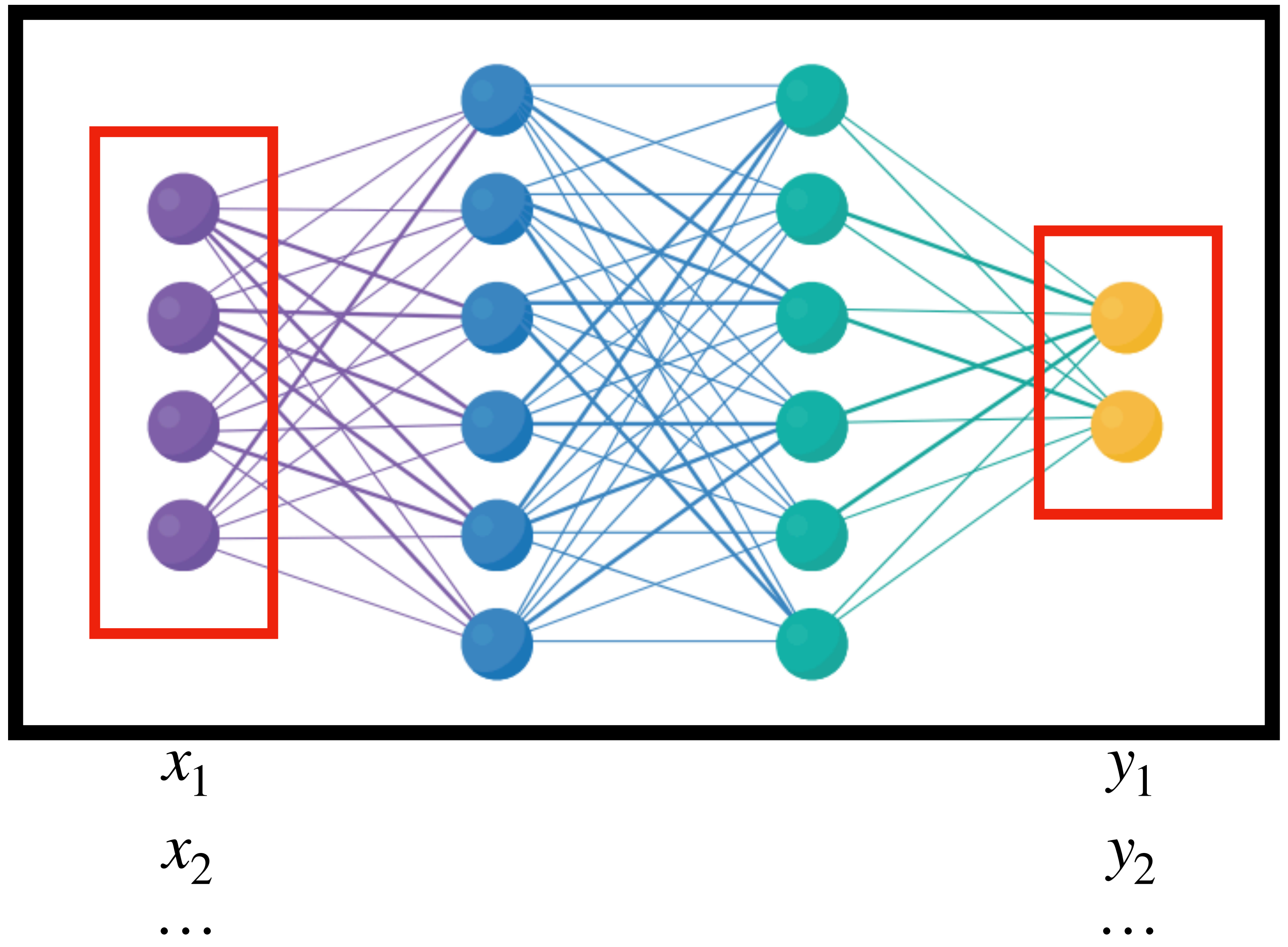
1. **Train NN normally, and freeze parameters.**

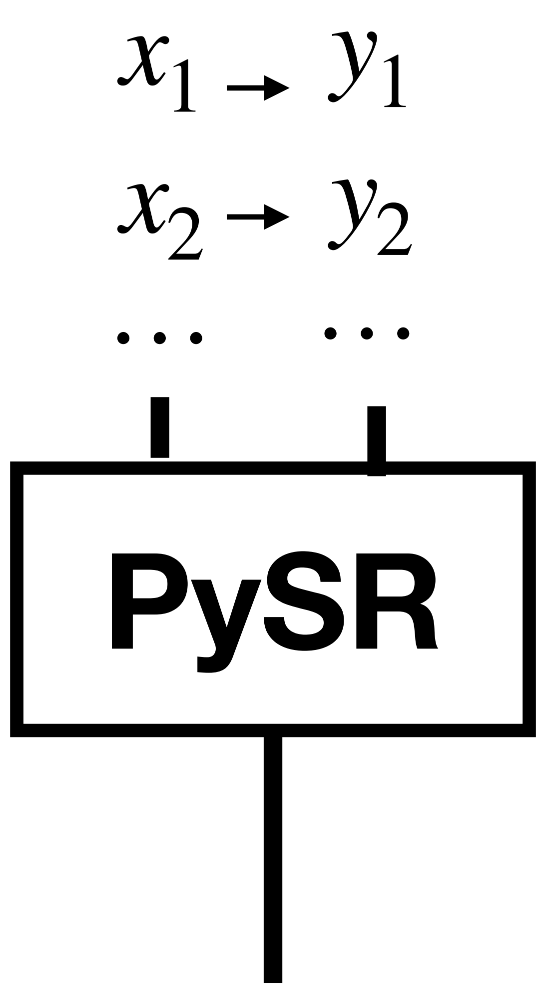
2. Record input/outputs of network over training set.

How this works:

Cranmer et al., 2019, 2020 – Work with: Alvaro Sanchez-Gonzalez, Shirley Ho, Peter Battaglia, Kyle Cranmer, David Spergel, Rui Xu







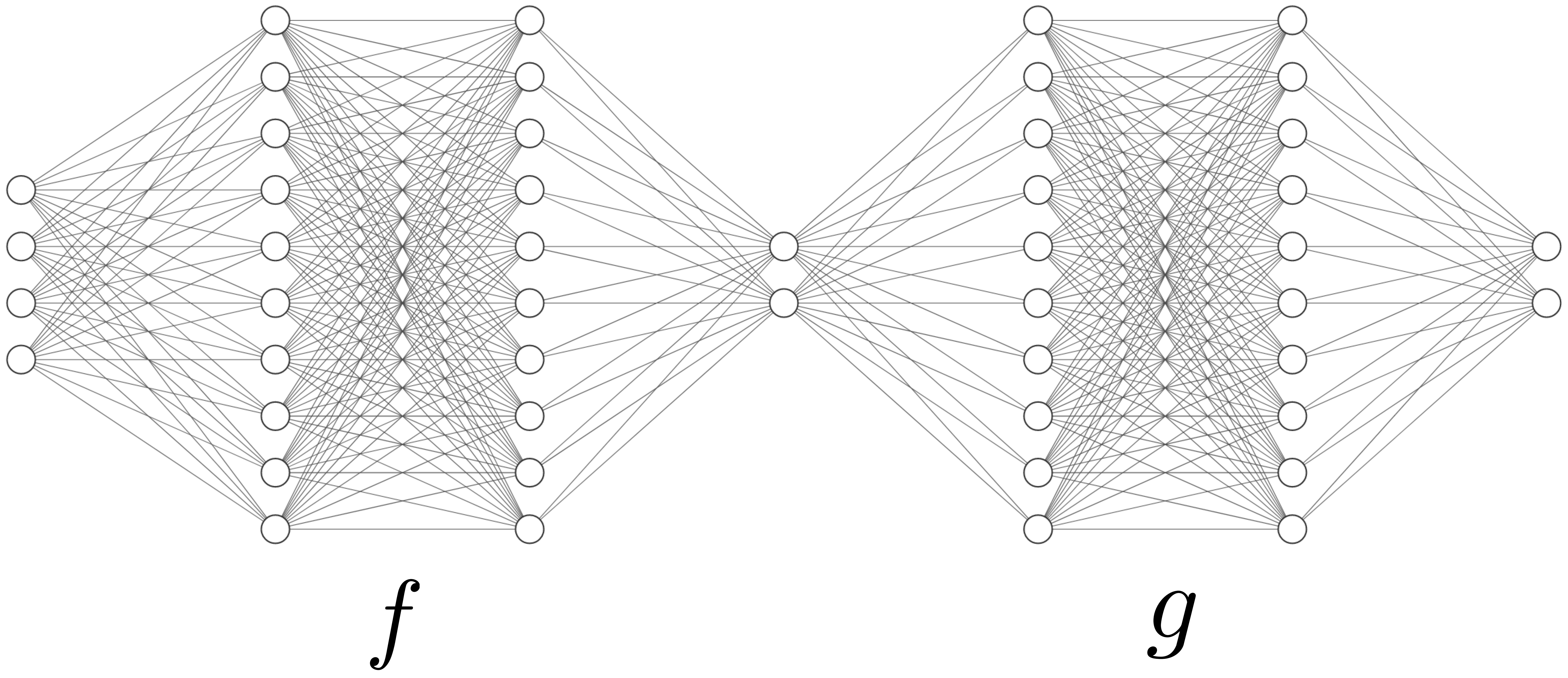
$$[y]_1 = \cos(2.1 \cdot [x]_3) - [x]_4$$

1. **Train NN normally, and freeze parameters.**

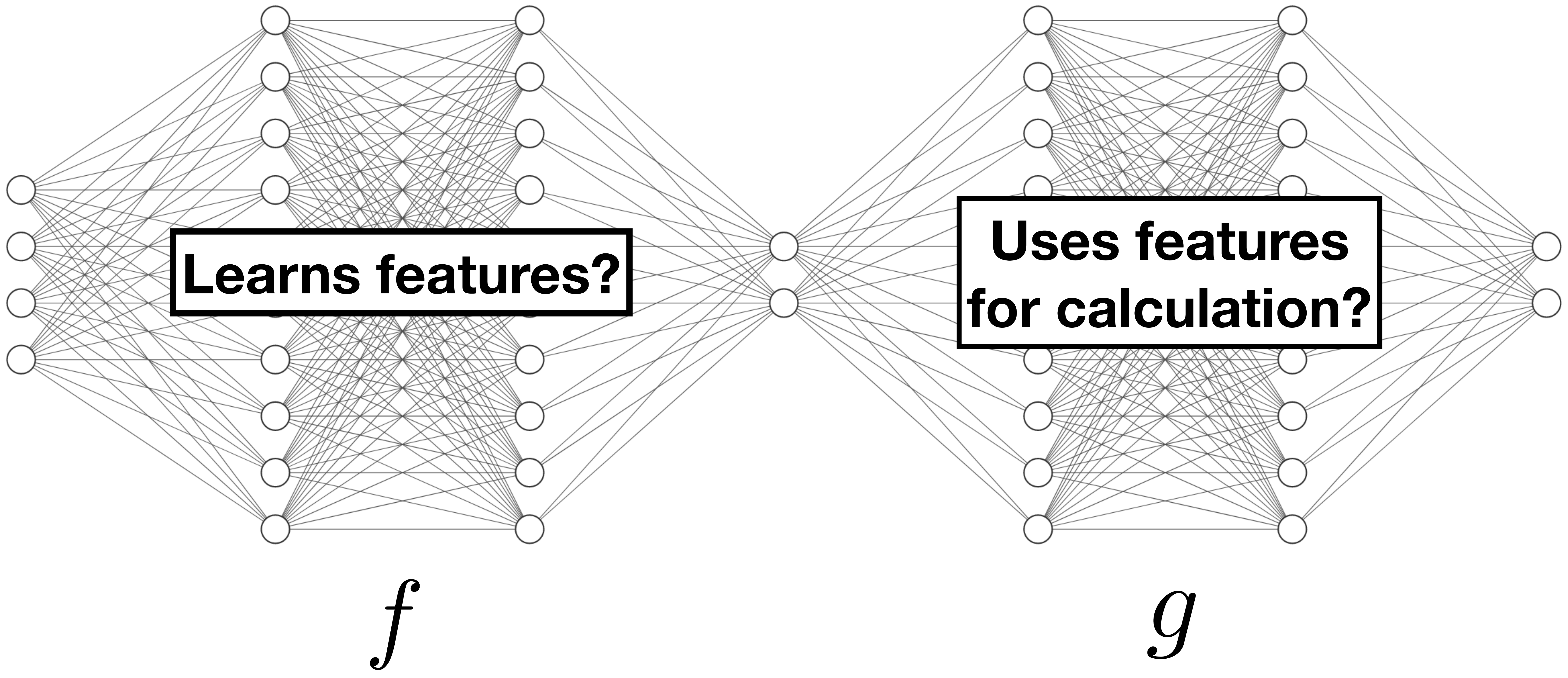
2. Record input/outputs of network over training set.

3. Fit the input/outputs of the neural network with PySR

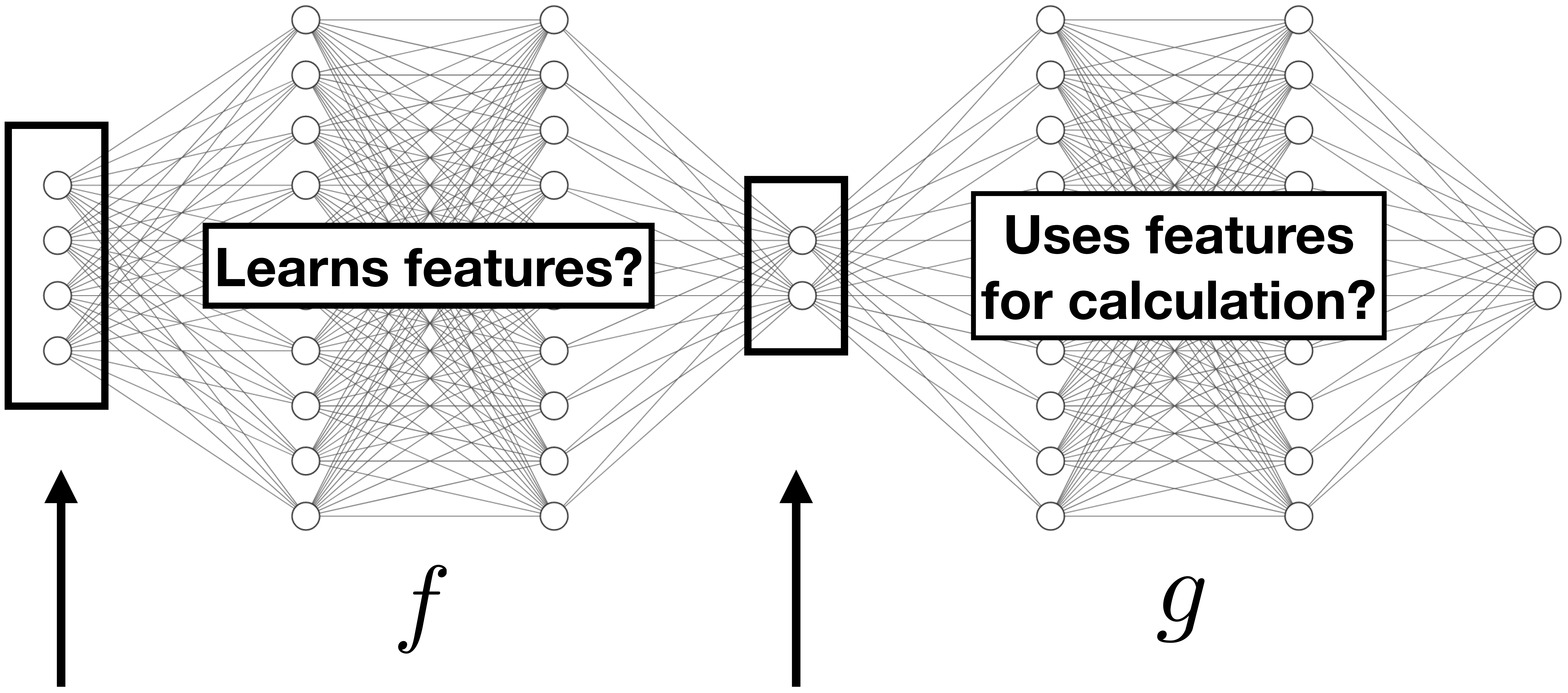
Full Symbolic Distillation



Full Symbolic Distillation



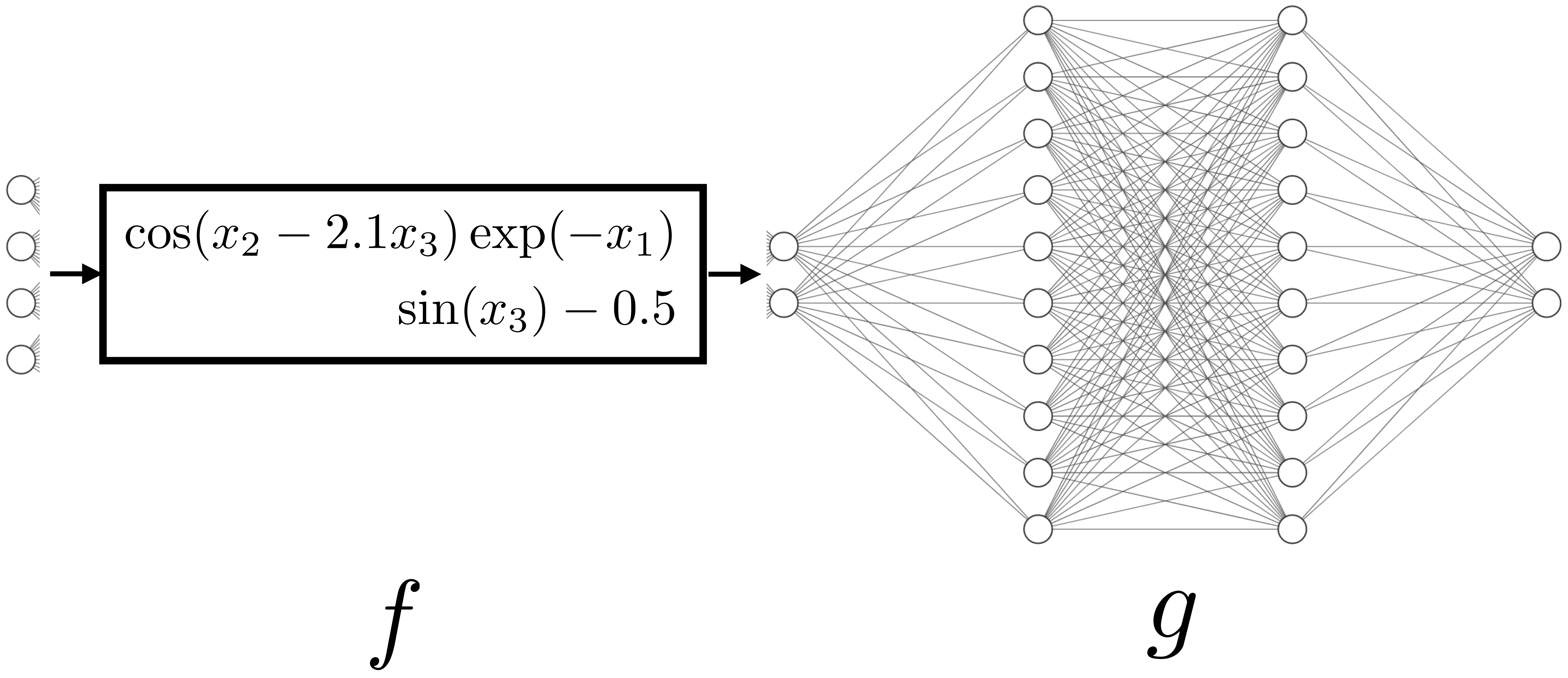
Full Symbolic Distillation



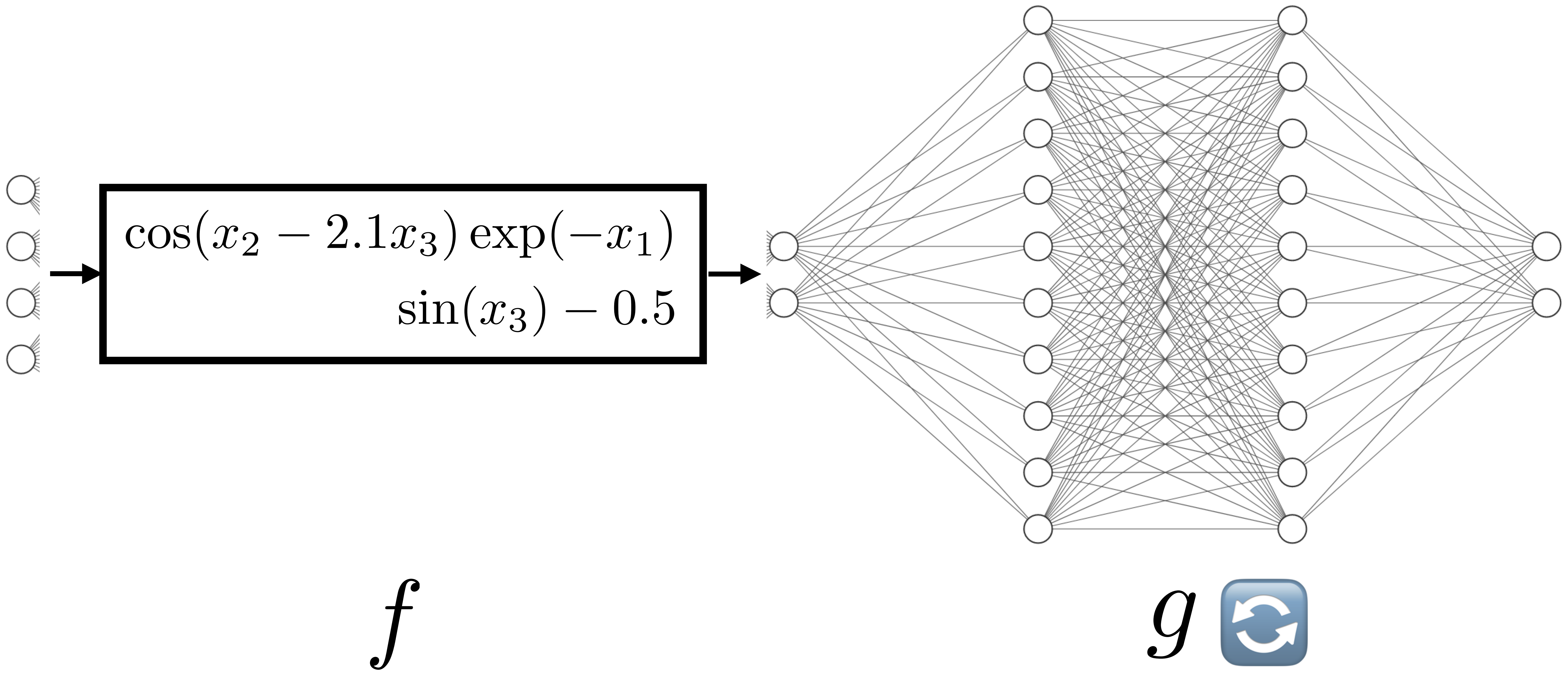
$$y_1 = \cos(x_2 - 2.1x_3)\exp(-x_1)$$

$$y_2 = \sin(x_3) - 0.5$$

Full Symbolic Distillation

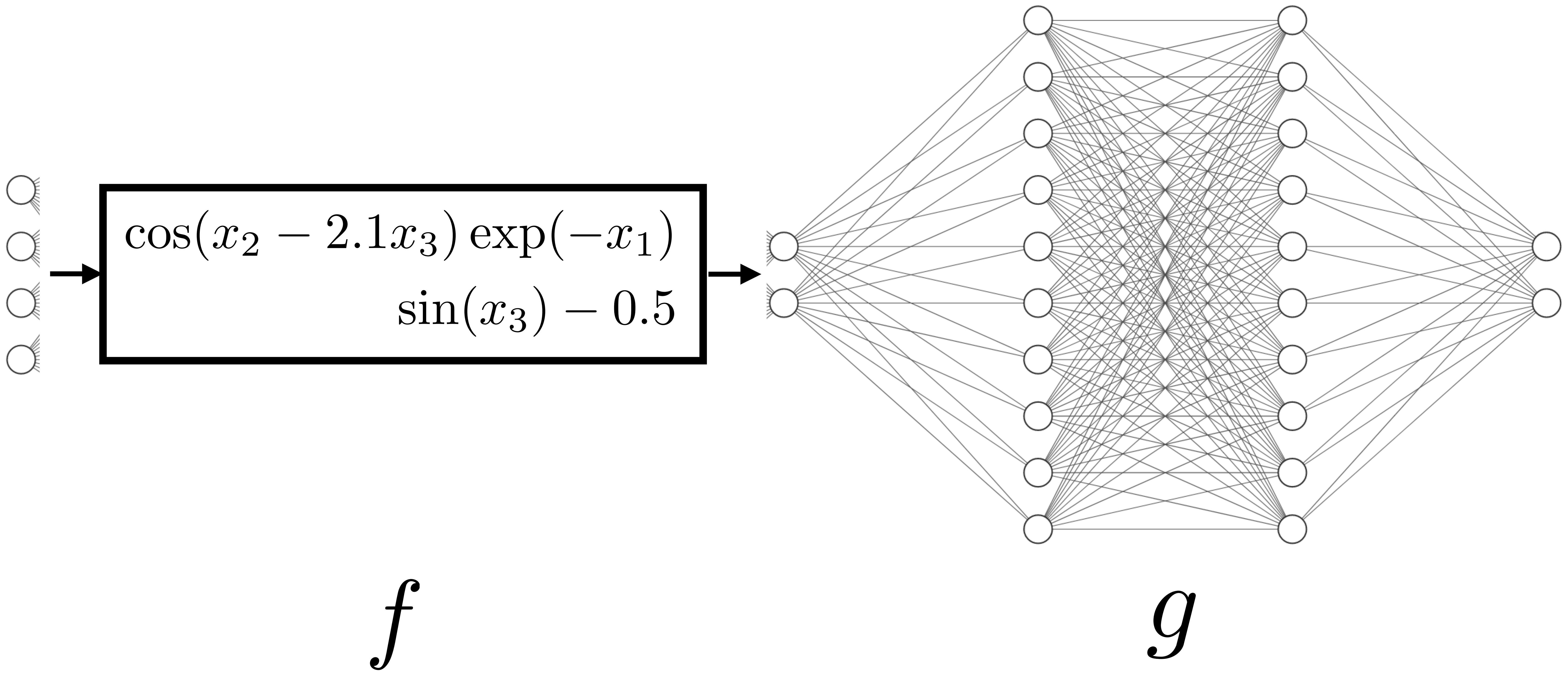


Full Symbolic Distillation

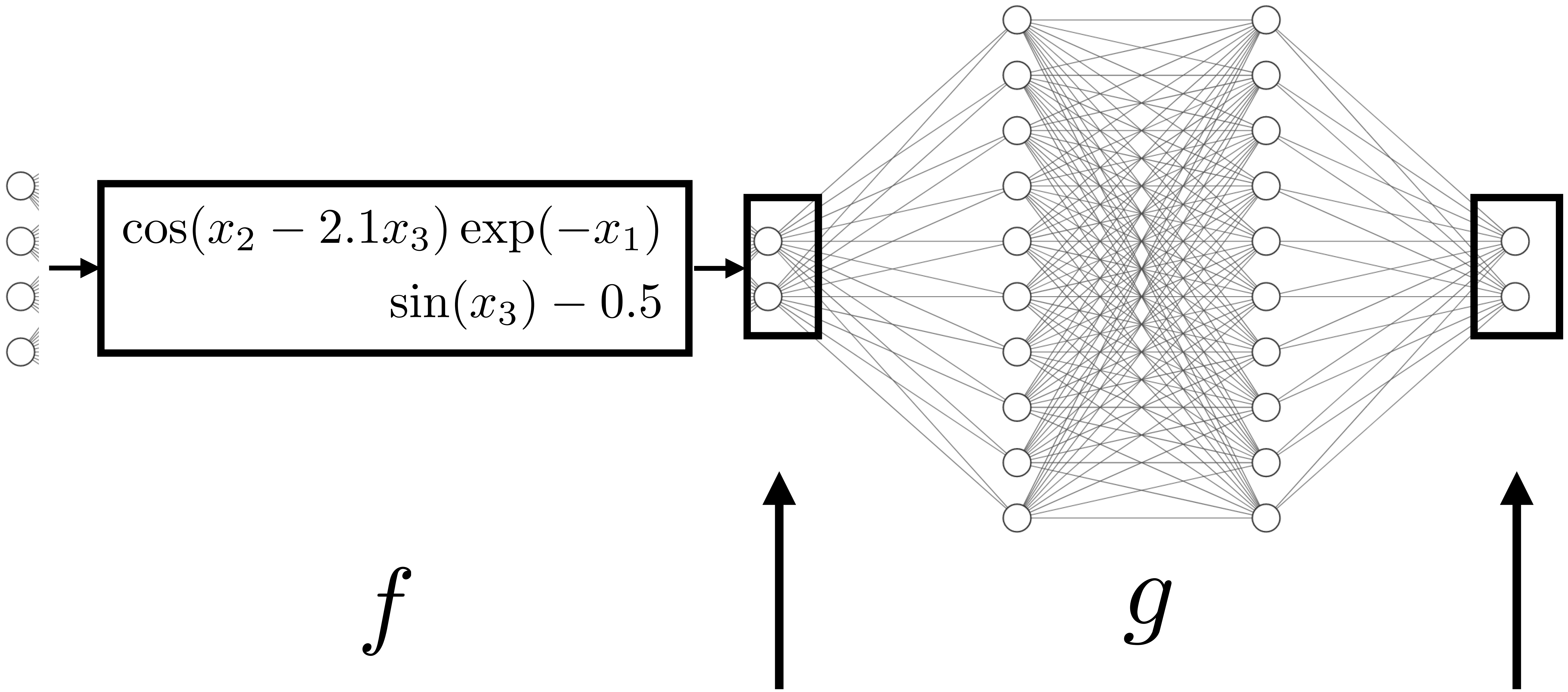


Re-train g, to pick up any errors in the approximation of f

Full Symbolic Distillation



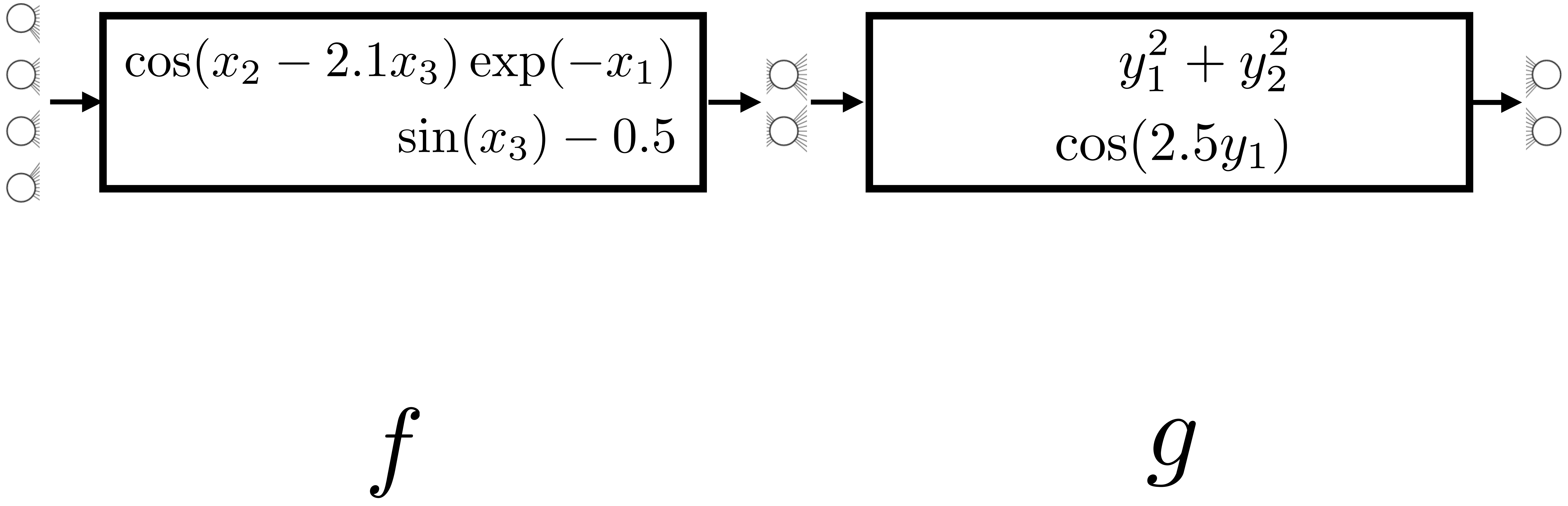
Full Symbolic Distillation



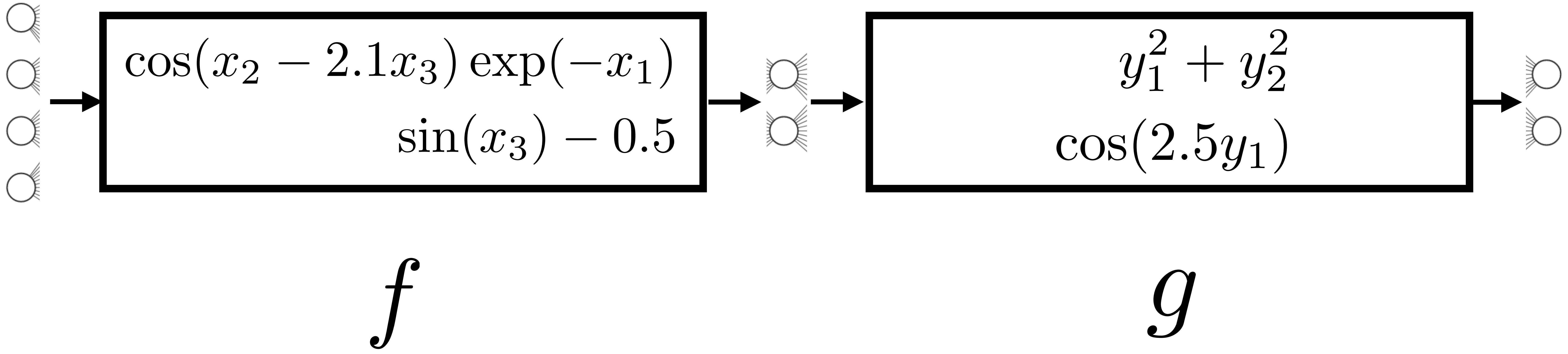
$$z_1 = y_1^2 + y_2^2$$

$$z_2 = \cos(2.5y_1)$$

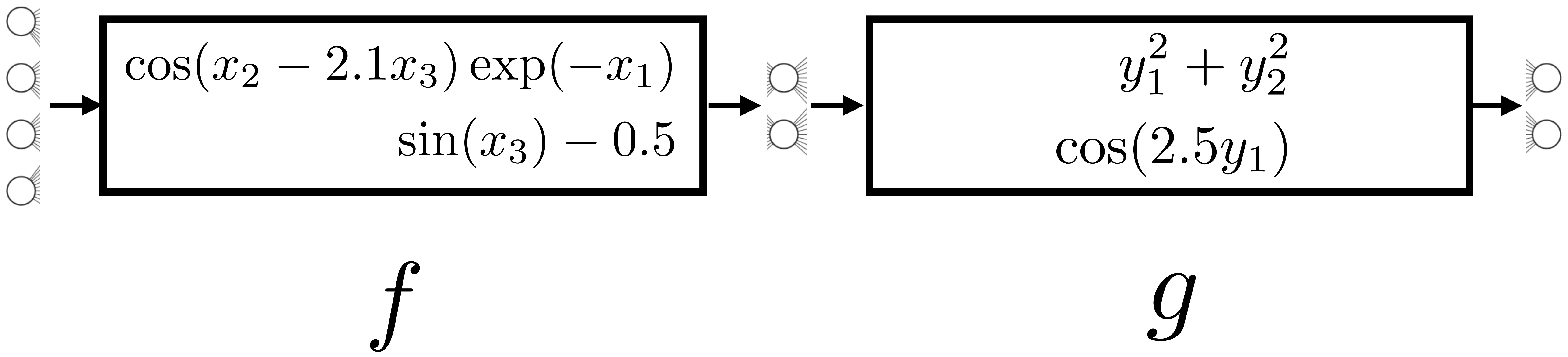
Full Symbolic Distillation



Full Symbolic Distillation

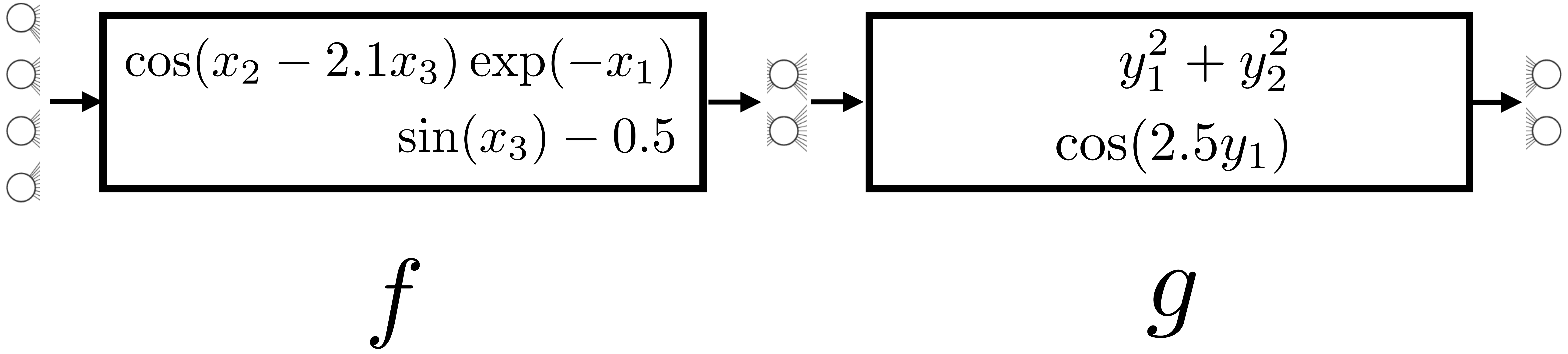


Full Symbolic Distillation



$$(g \circ f)(x_1, x_2, x_3, x_4) = \begin{bmatrix} (\cos(x_2 - 2.1x_3)\exp(-x_1))^2 + (\sin(x_3) - 0.5)^2 \\ \cos(2.5\,(\sin(x_3) - 0.5)) \end{bmatrix}$$

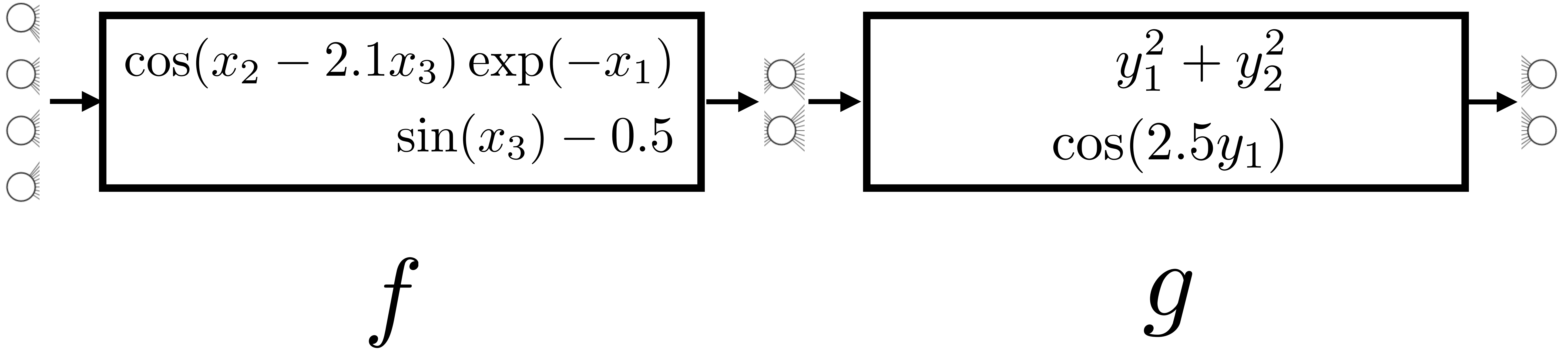
Full Symbolic Distillation



$$(g \circ f)(x_1, x_2, x_3, x_4) = \begin{bmatrix} (\cos(x_2 - 2.1x_3)\exp(-x_1))^2 + (\sin(x_3) - 0.5)^2 \\ \cos(2.5\,(\sin(x_3) - 0.5)) \end{bmatrix}$$

Fully-interpretable approximation of the original neural network!

Full Symbolic Distillation



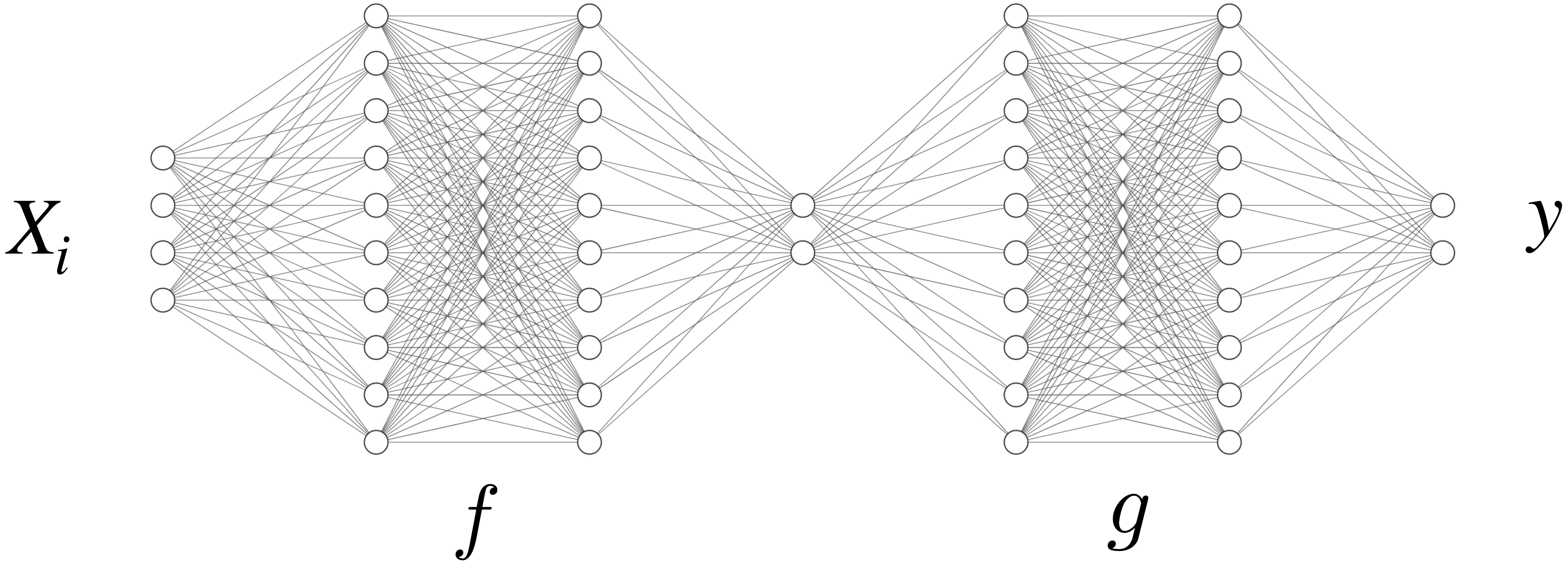
$$(g \circ f)(x_1, x_2, x_3, x_4) = \begin{bmatrix} (\cos(x_2 - 2.1x_3)\exp(-x_1))^2 + (\sin(x_3) - 0.5)^2 \\ \cos(2.5\,(\sin(x_3) - 0.5)) \end{bmatrix}$$

Fully-interpretable approximation of the original neural network!

(Searching over n^2 expressions → Searching over $2n$ expressions)

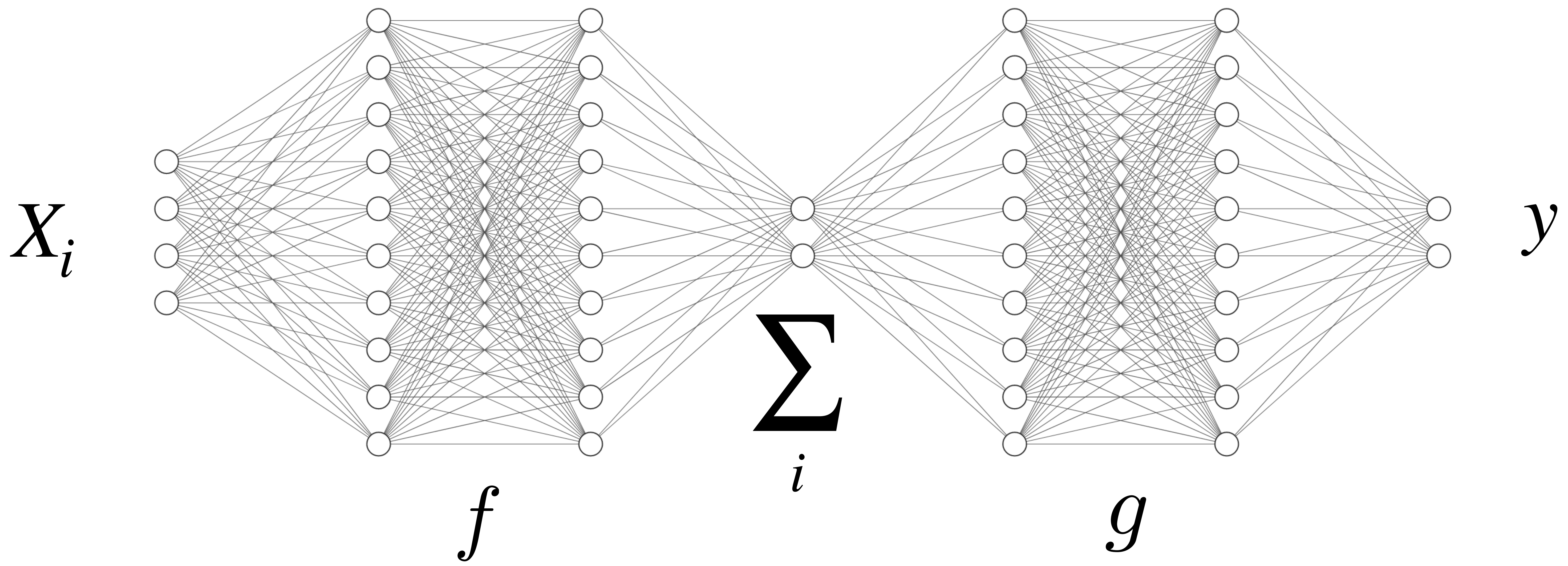
Inductive bias

- Introducing some form of inductive bias is needed to eliminate the functional degeneracy. For example:



Inductive bias

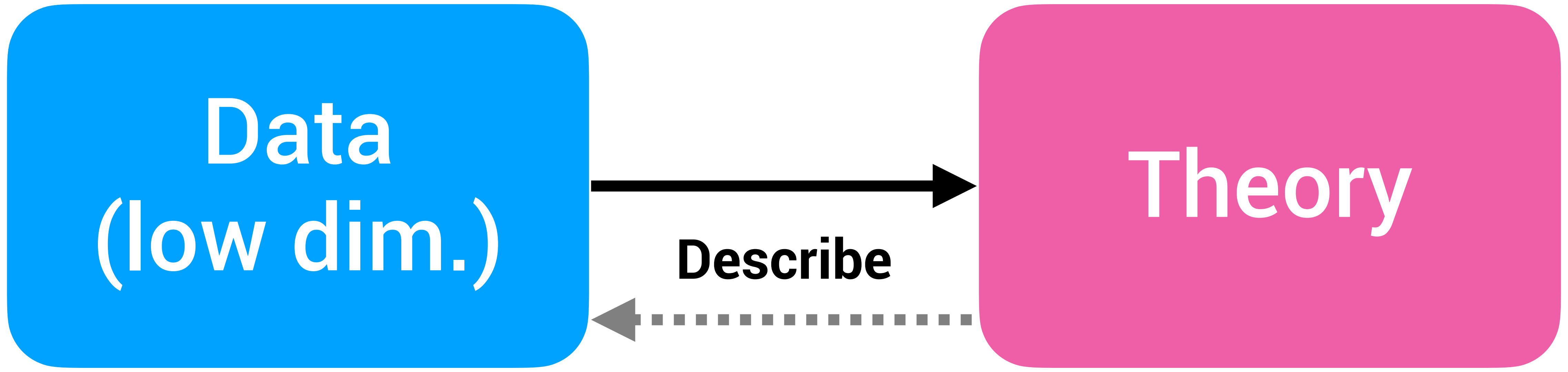
- Introducing some form of inductive bias is needed to eliminate the functional degeneracy. For example:



(the latent space between f and g could have some aggregation over a set)

Recall:

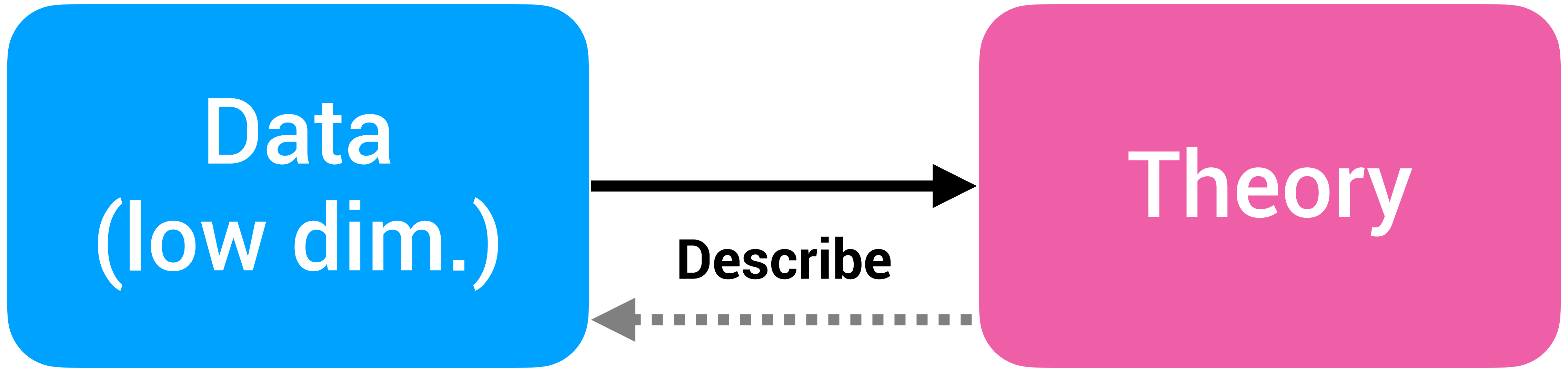
Traditional approach to science:



(May be summary statistics)

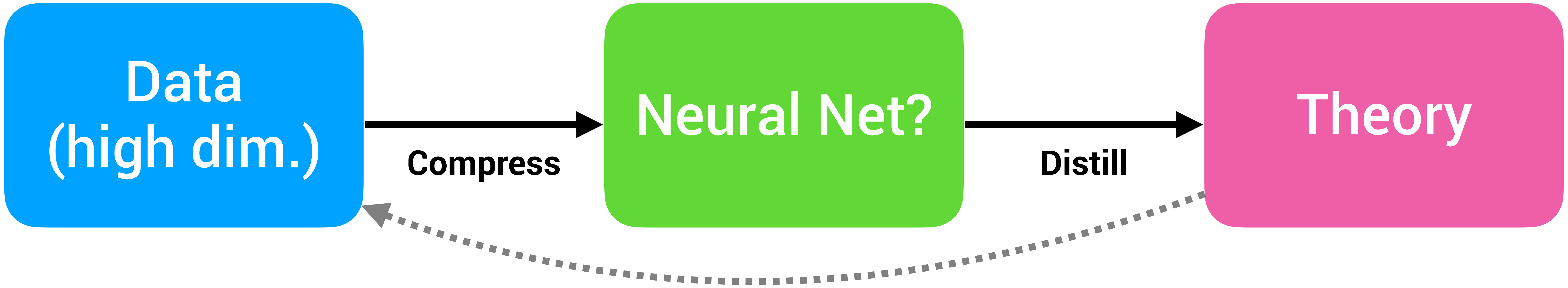
Recall:

Traditional approach to science:



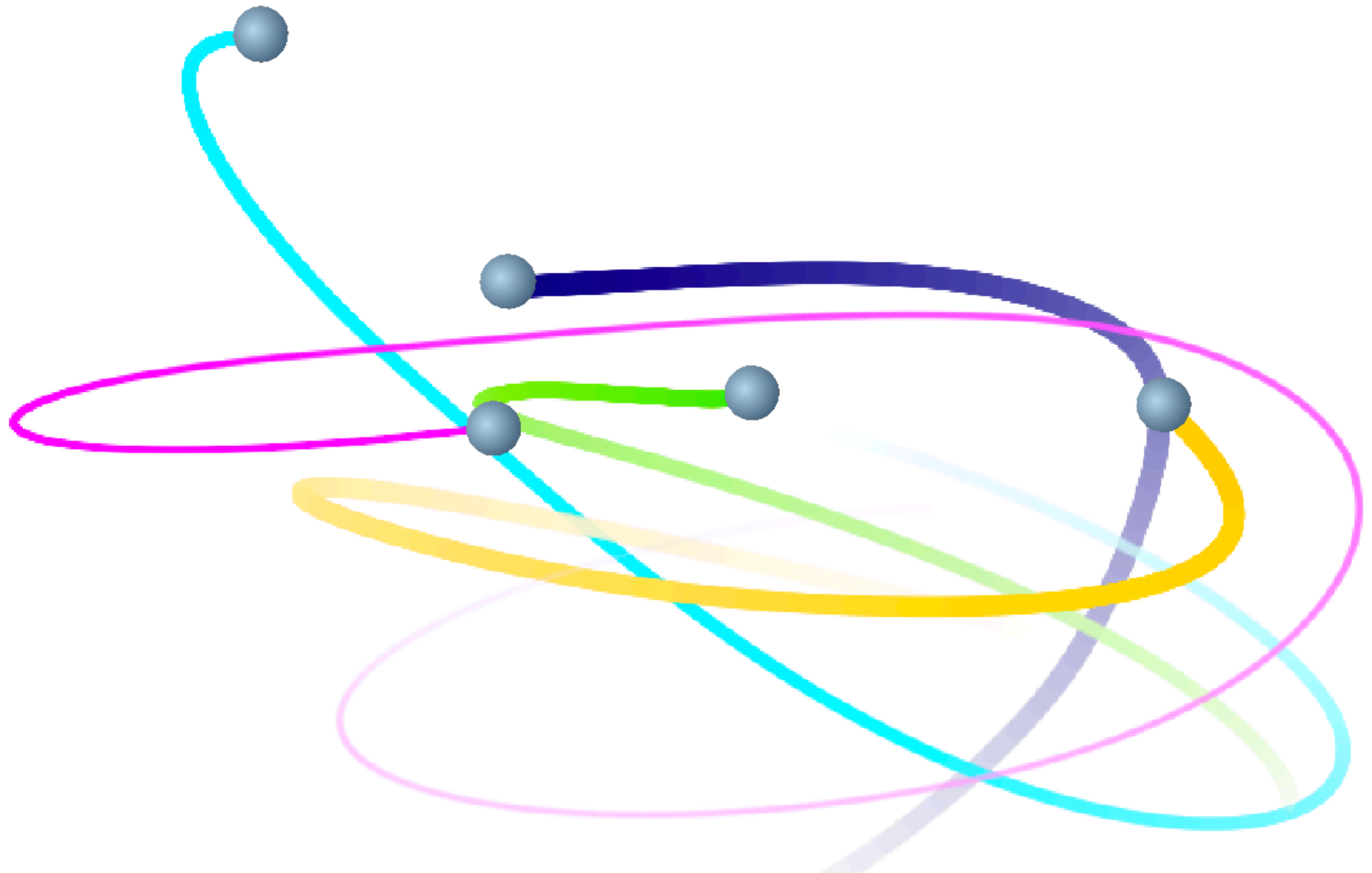
(May be summary statistics)

Era of AI?



Some examples:

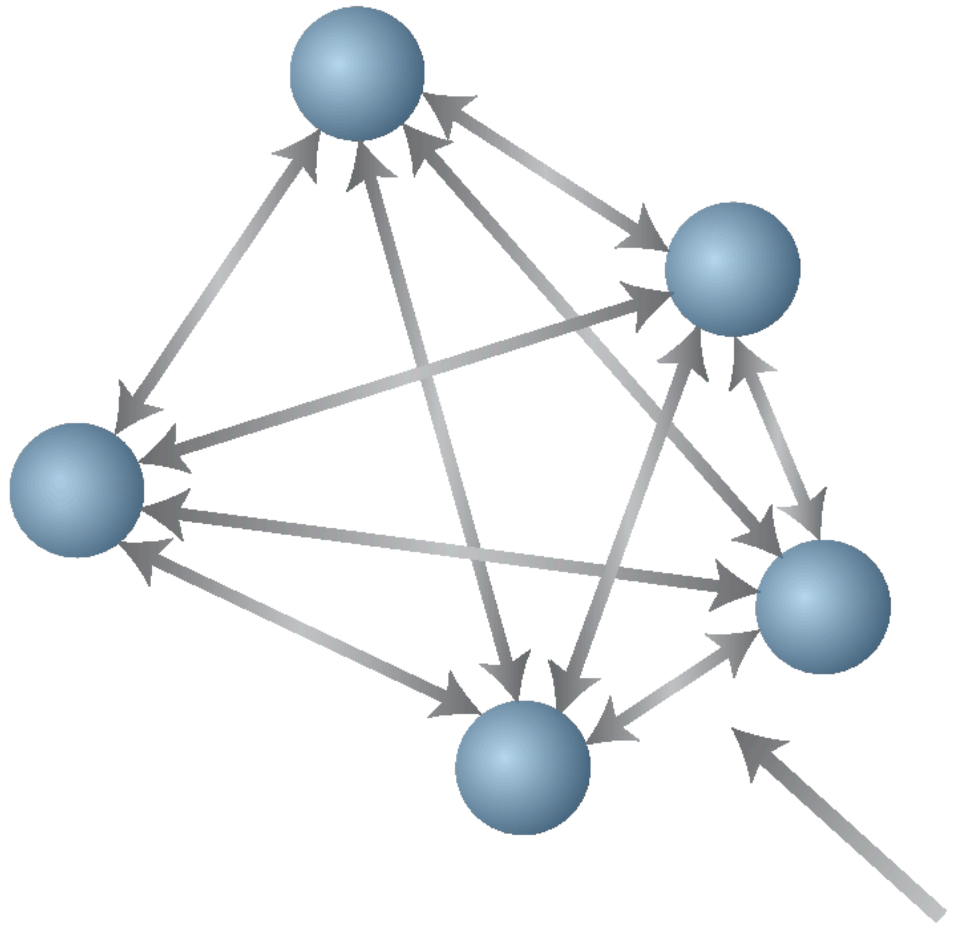
Dataset



Simple Particles

Predict Dynamics

Model with
Graph Neural Network



Encourage Low-Dimensionality

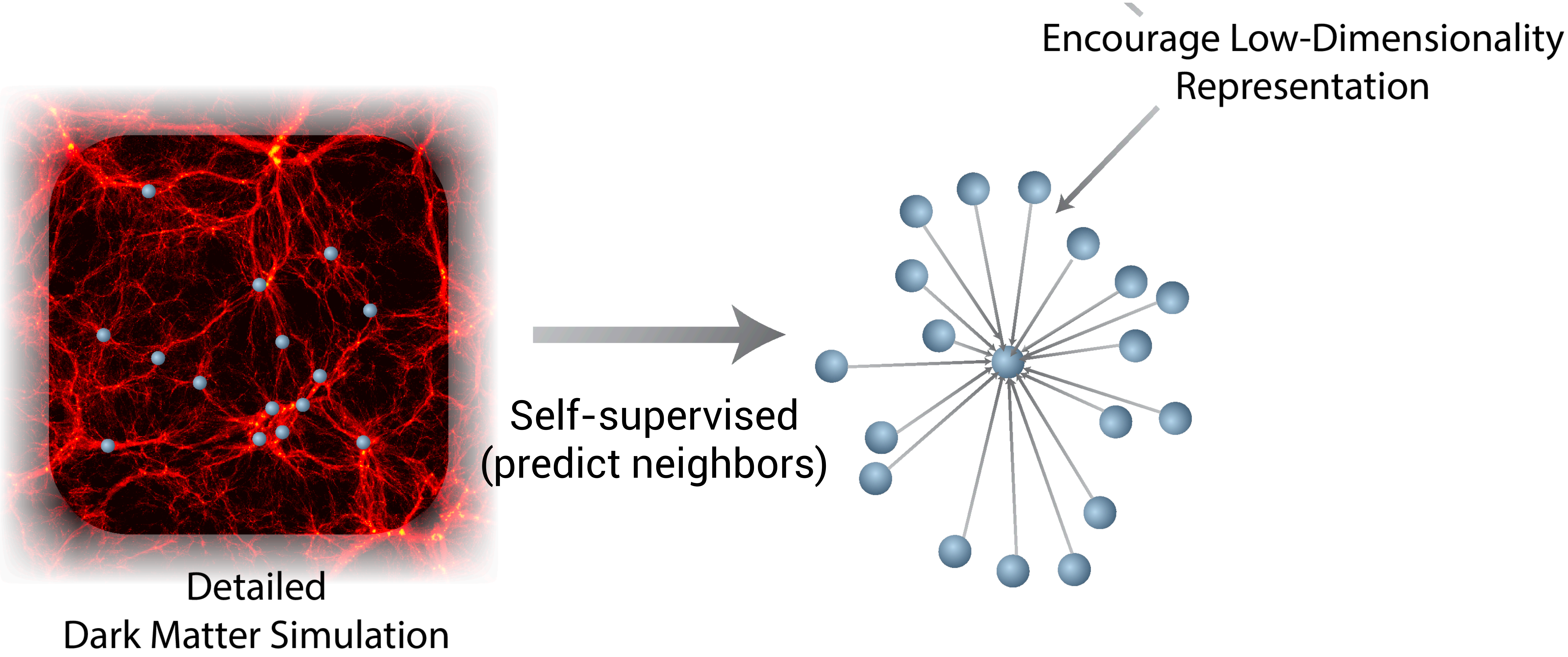
with Alvaro Sanchez Gonzalez, Peter Battaglia, Rui Xu, Kyle Cranmer,
David Spergel, Shirley Ho; (NeurIPS 2020)

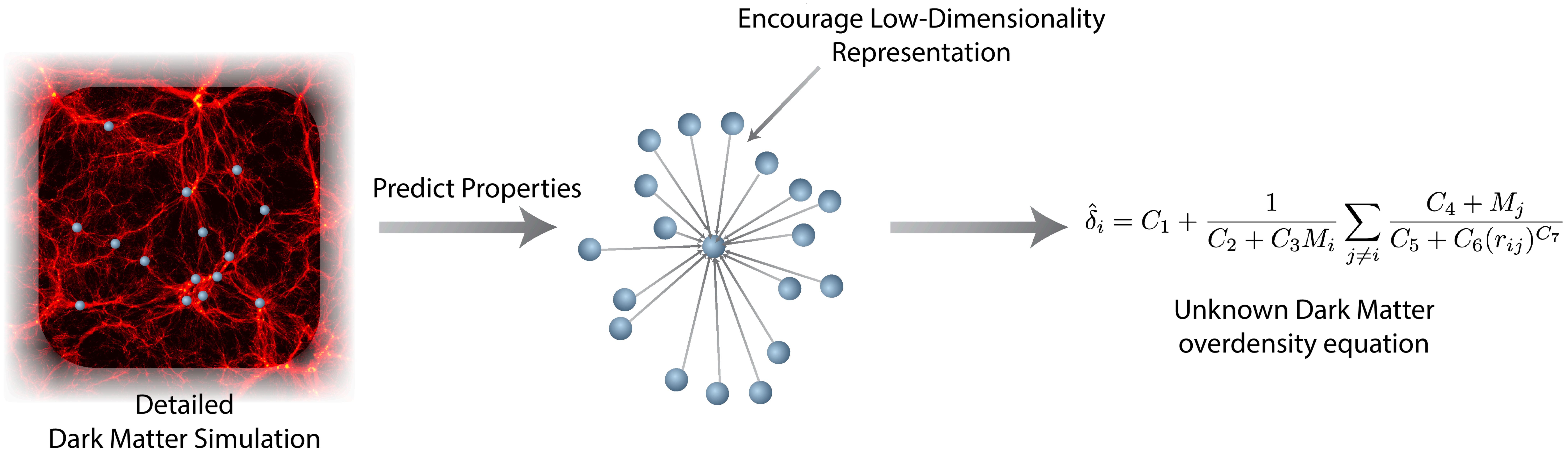
$$
\begin{aligned}
1/r^2 :\ & U_{12} = -m_1 m_2 / r'_{12} \\
1/r :\ & U_{12} = m_1 m_2 \log(r'_{12}) \\
\text{Spring} :\ & U_{12} = (r'_{12} - 1)^2 \\
\text{Damped} :\ & U_{12} = (r'_{12} - 1)^2 + \mathbf{r}_1 \cdot \dot{\mathbf{r}}_1 / n \\
\text{Charge} :\ & U_{12} = q_1 q_2 / r'_{12} \\
\text{Dicontinuous} :\ & U_{12} = \begin{cases} 0, & r'_{12} < 2 \\ (r'_{12} - 1)^2, & r'_{12} \geq 2 \end{cases}
\end{aligned}
$$

$$
\begin{aligned}
1/r^2 :&\; U_{12} = -m_1 m_2 / r'_{12} \\
1/r :&\; U_{12} = m_1 m_2 \log(r'_{12}) \\
\text{Spring} :&\; U_{12} = (r'_{12} - 1)^2 \\
\text{Damped} :&\; U_{12} = (r'_{12} - 1)^2 + \mathbf{r}_1 \cdot \dot{\mathbf{r}}_1 / n \\
\text{Charge} :&\; U_{12} = q_1 q_2 / r'_{12} \\
\text{Dicontinuous} :&\; U_{12} = \begin{cases} 0, & r'_{12} < 2 \\ (r'_{12} - 1)^2, & r'_{12} \geq 2 \end{cases}
\end{aligned}
$$

Knowledge Discovery

- Predict the dark matter properties in a simulation with a graph neural network:




Encourage Low-Dimensionality
Representation
Predict Properties
Detailed
Dark Matter Simulation
$\hat{\delta}_i = C_1 + \frac{1}{C_2 + C_3 M_i} \sum_{j \neq i} \frac{C_4 + M_j}{C_5 + C_6 (r_{ij})^{C_7}}$
Unknown Dark Matter
overdensity equation

Example 2:
Discovering Orbital Mechanics

Example 2:
Discovering Orbital Mechanics

Can we learn Newton's law of gravity by modelling the solar system with a graph neural network?

Unknown masses, and unknown dynamical model.

Example 2:
Discovering Orbital Mechanics

Can we learn Newton's law of gravity by modelling the solar system with a graph neural network?

Unknown masses, and unknown dynamical model.

"Rediscovering orbital mechanics with machine learning" (2022)
Pablo Lemos, Niall Jeffrey, Miles Cranmer, Shirley Ho, Peter Battaglia


True
Predicted

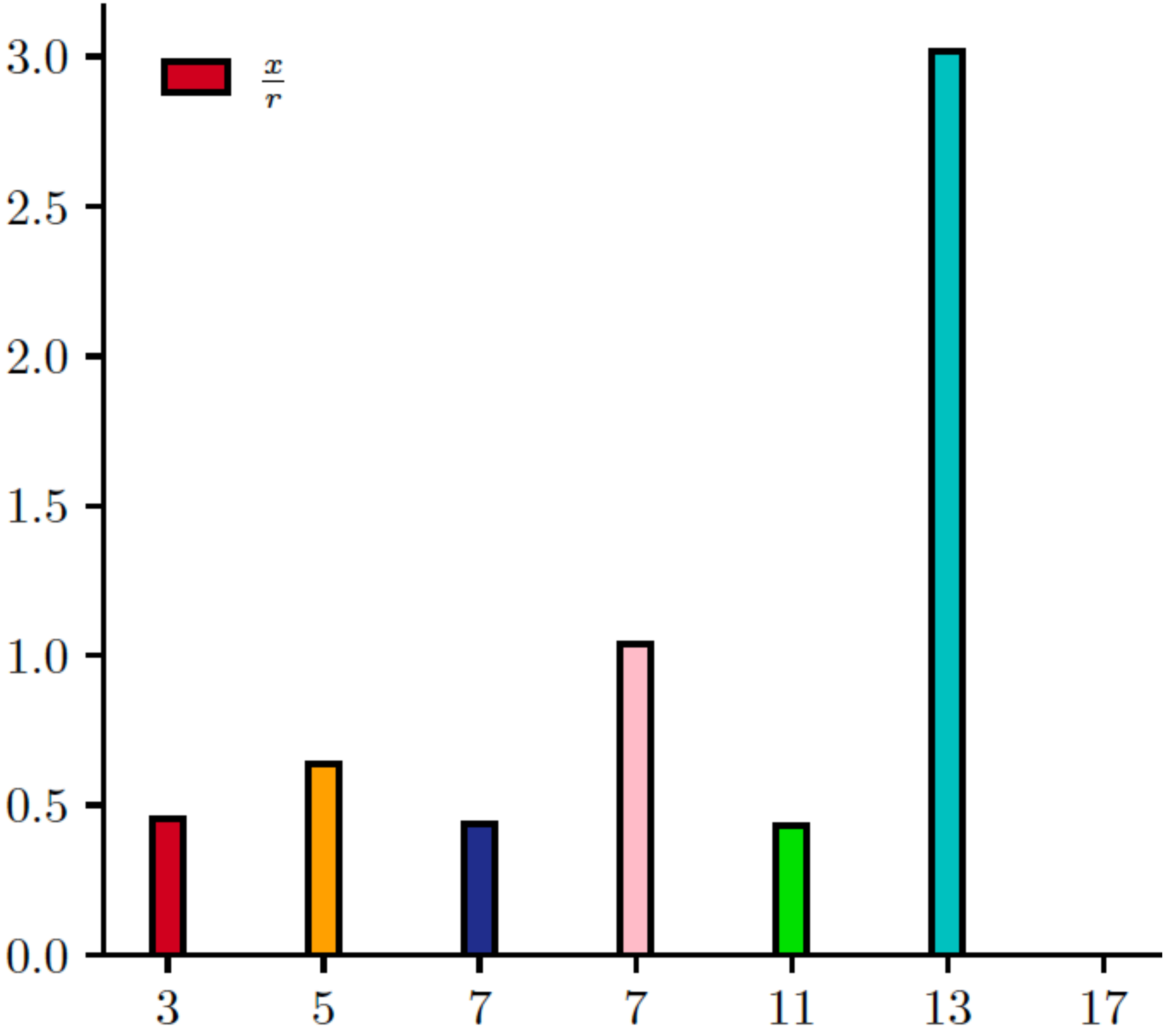

True
Predicted

Next: interpretation

Approximate relation between latent spaces of network with PySR

Interpretation Results for f

Accuracy/Complexity Tradeoff*

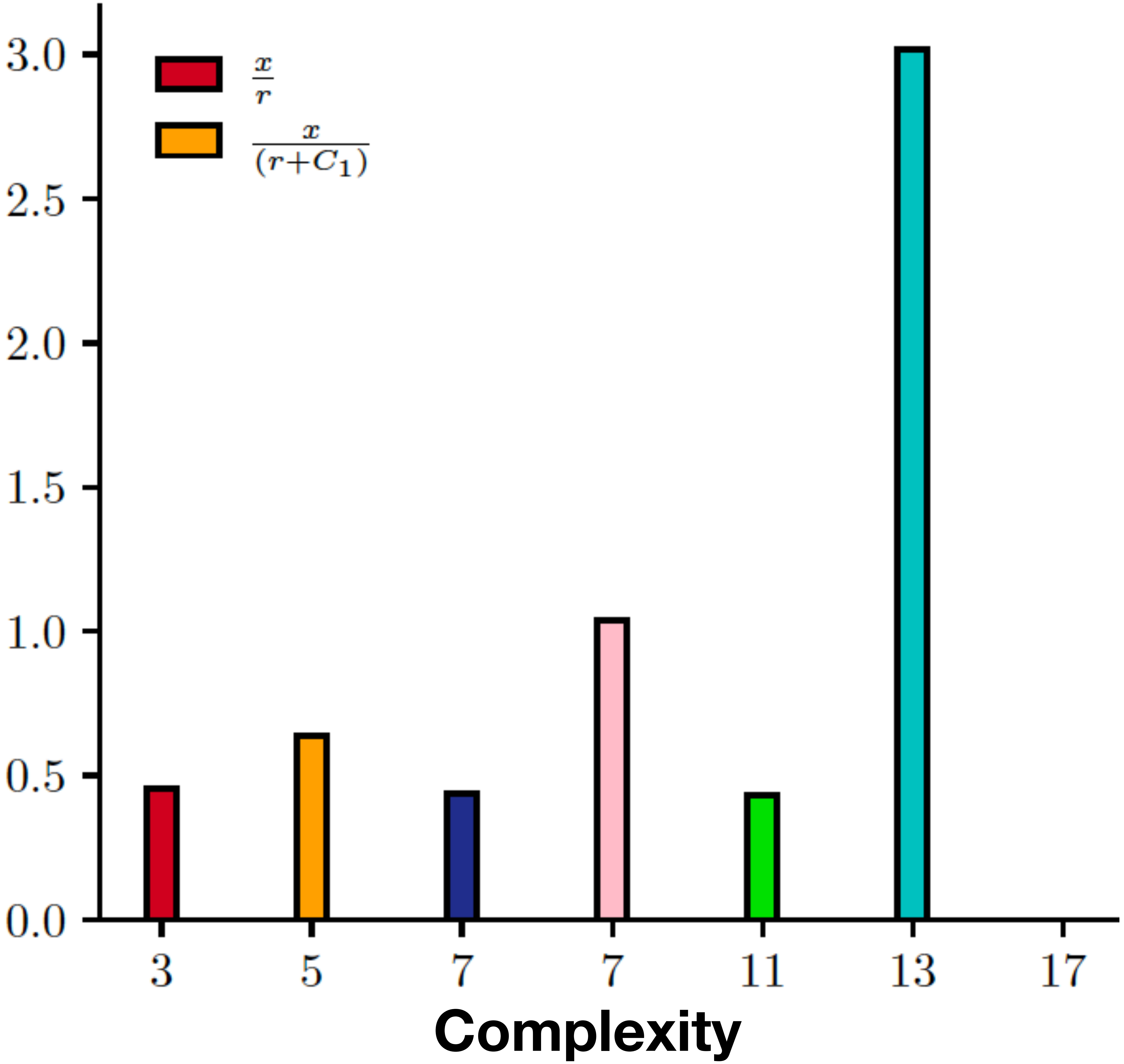


Complexity

***from Cranmer+2020; similar to Schmidt & Lipson, 2009**

Interpretation Results for f

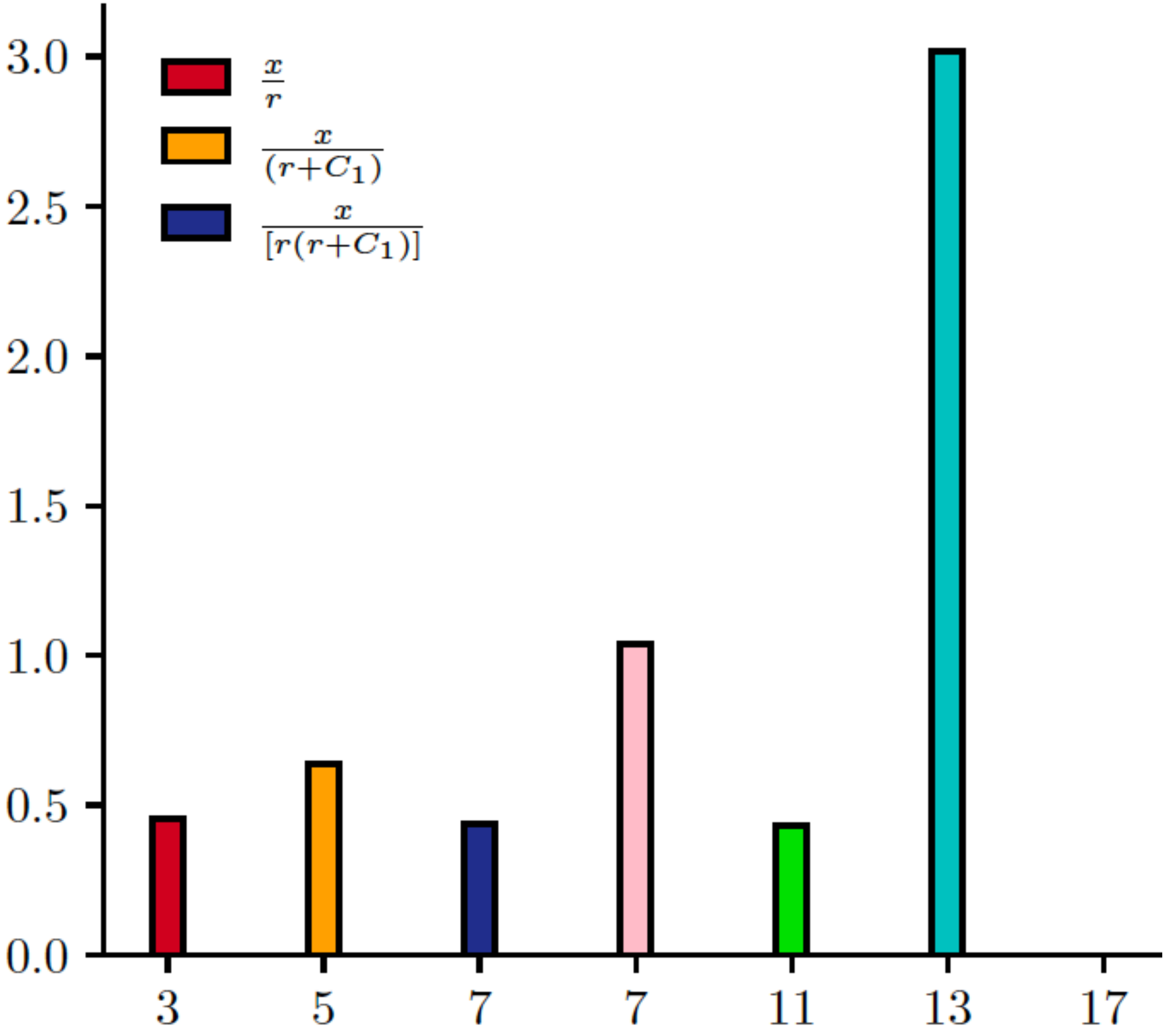
Accuracy/Complexity Tradeoff*



***from Cranmer+2020; similar to Schmidt & Lipson, 2009**

Interpretation Results for f

Accuracy/Complexity Tradeoff*

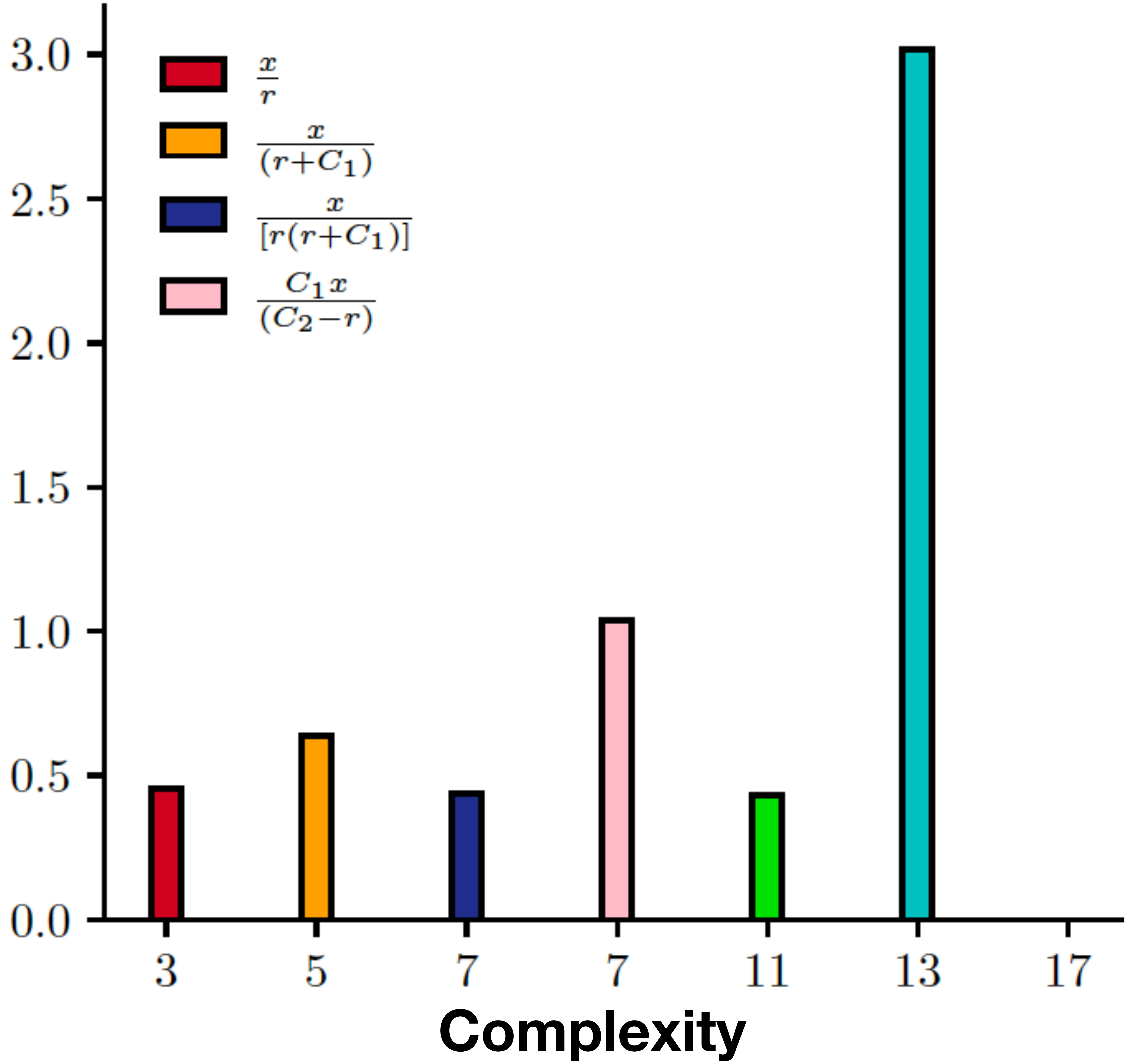


Complexity

*from Cranmer+2020; similar to Schmidt & Lipson, 2009

Interpretation Results for f

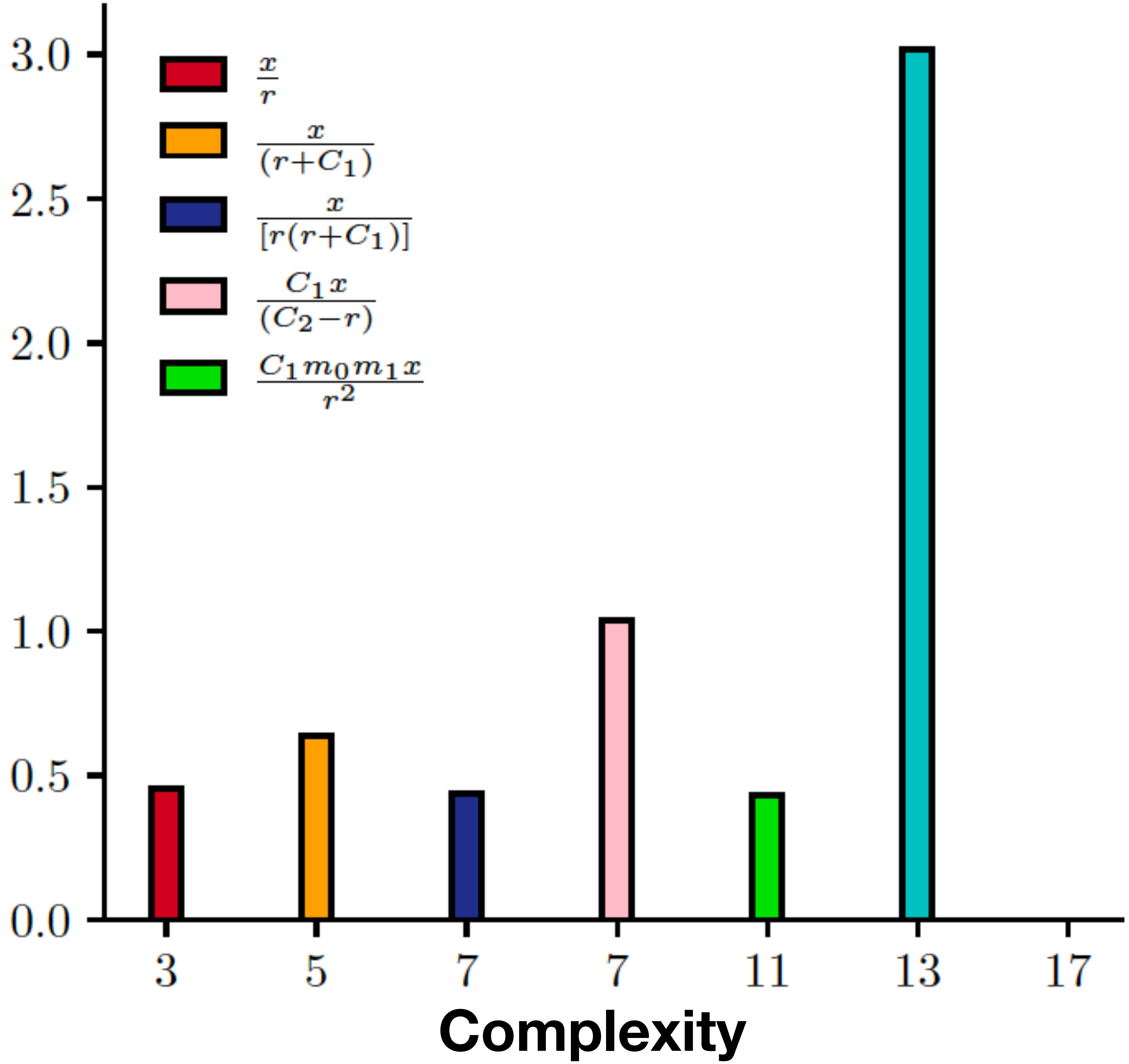
Accuracy/Complexity Tradeoff*



*from Cranmer+2020; similar to Schmidt & Lipson, 2009

Interpretation Results for f

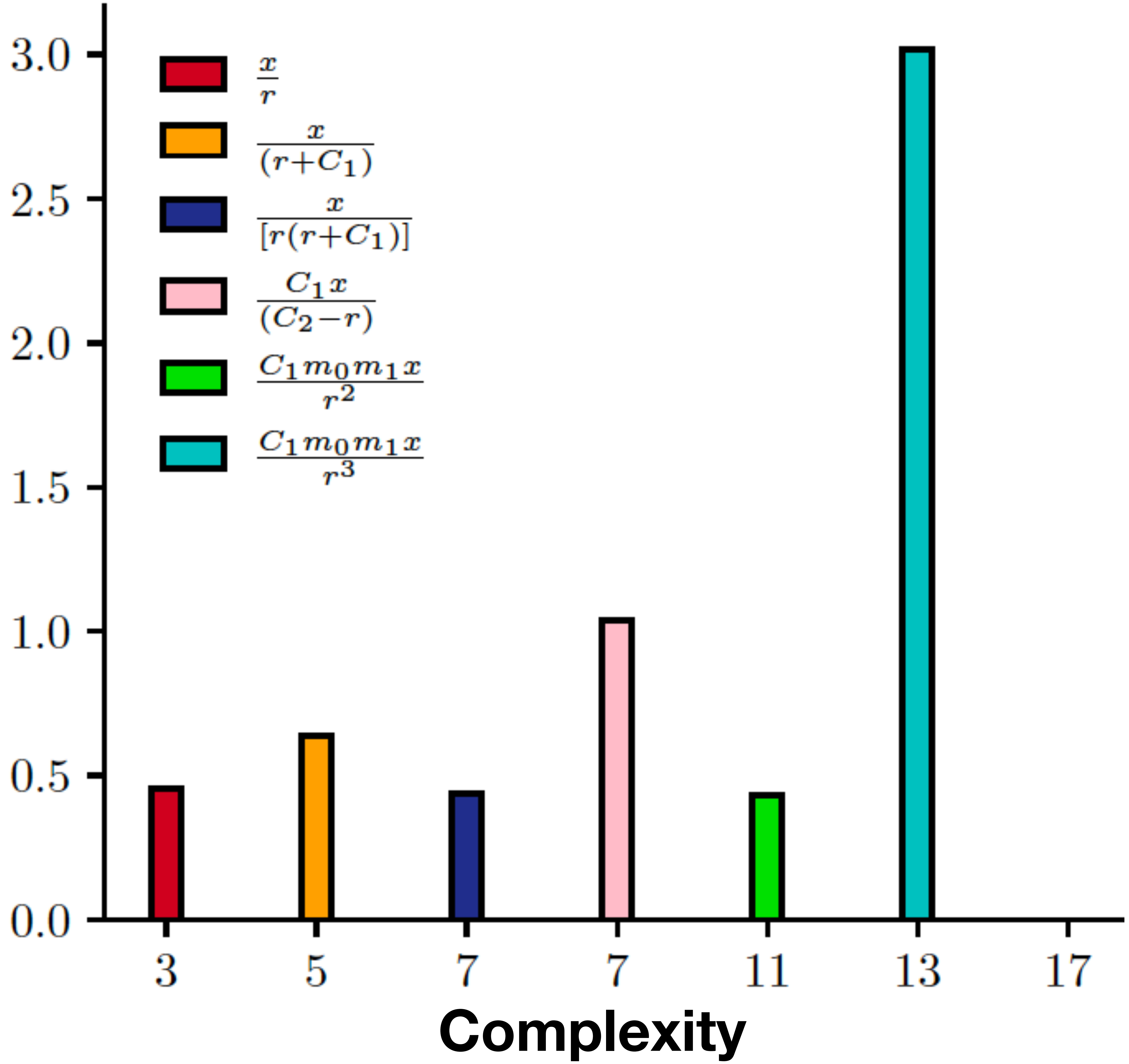
Accuracy/Complexity Tradeoff*



*from Cranmer+2020; similar to Schmidt & Lipson, 2009

Interpretation Results for f

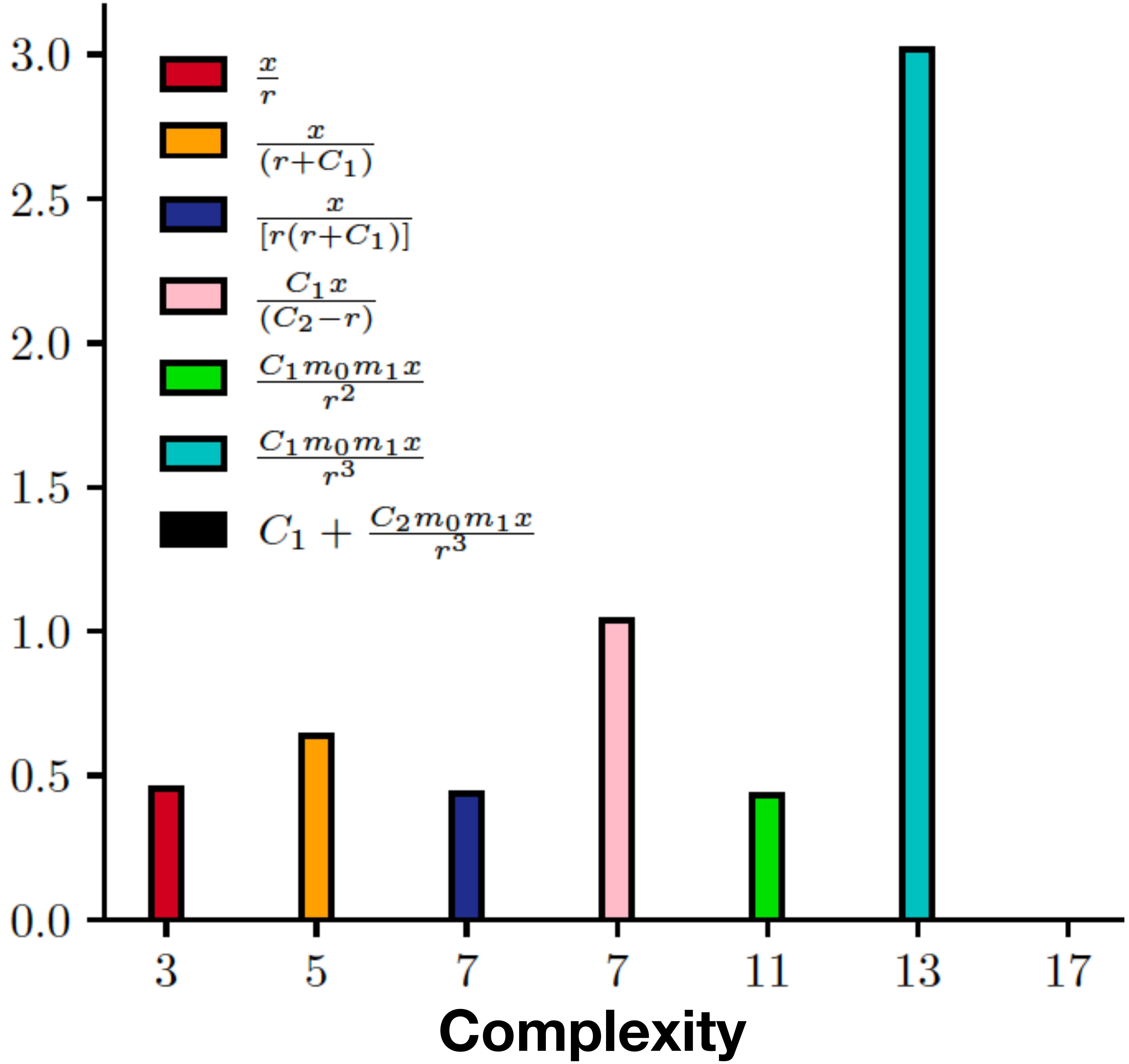
Accuracy/Complexity Tradeoff*



*from Cranmer+2020; similar to Schmidt & Lipson, 2009

Interpretation Results for f

Accuracy/Complexity Tradeoff*

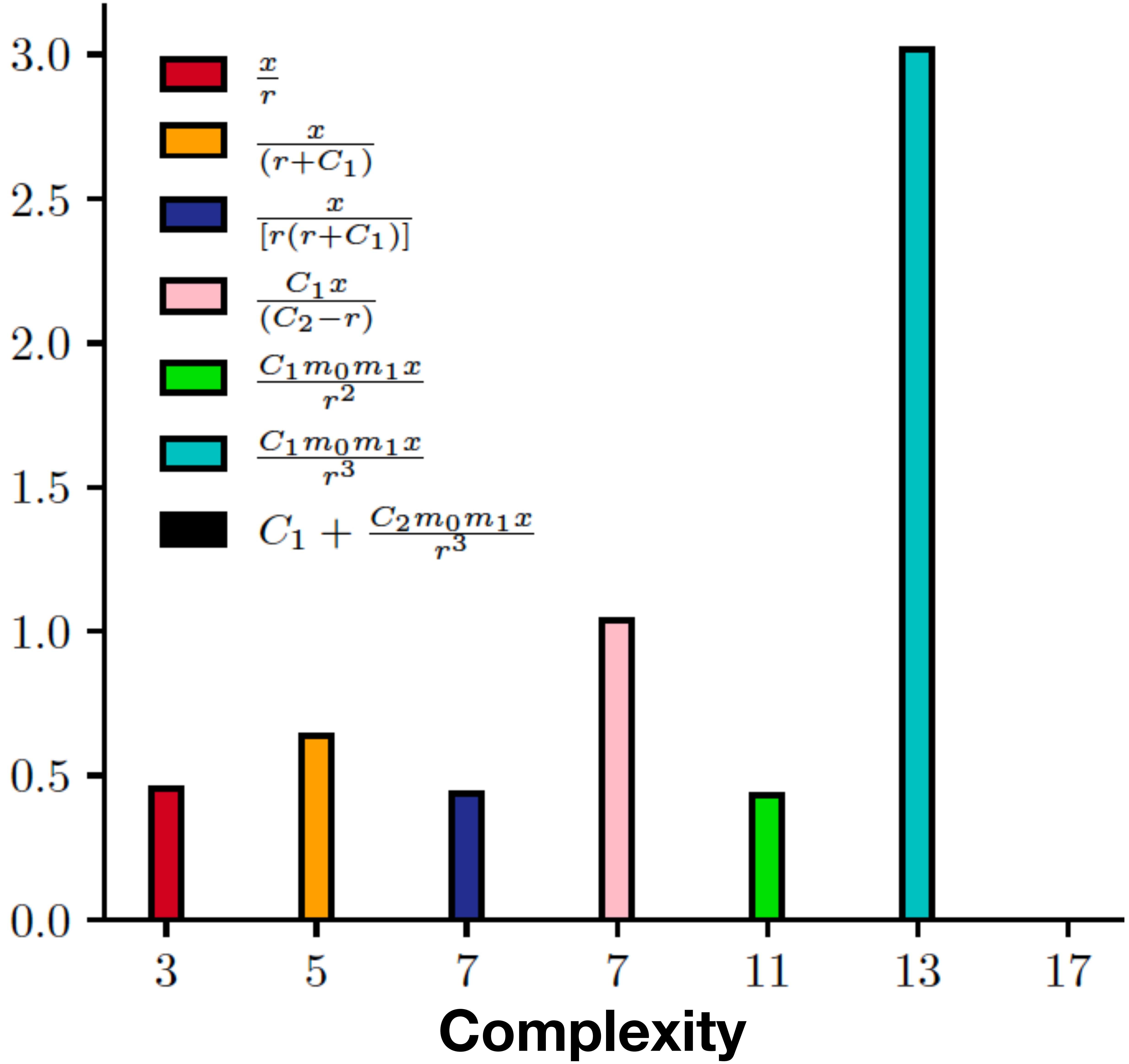


*from Cranmer+2020; similar to Schmidt & Lipson, 2009

Interpretation Results for f

Accuracy/Complexity Tradeoff*

$$= -\frac{d(\log(\textbf{error}))}{d(\textbf{complexity})}$$



*from Cranmer+2020; similar to Schmidt & Lipson, 2009

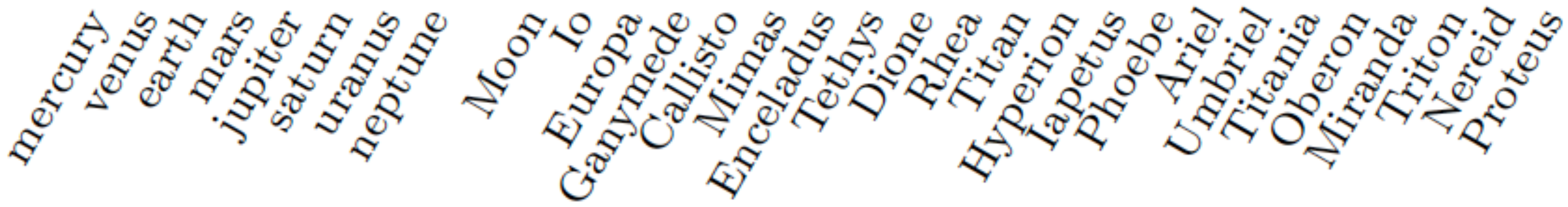
Test the symbolic model:


True
Predicted


True
Predicted

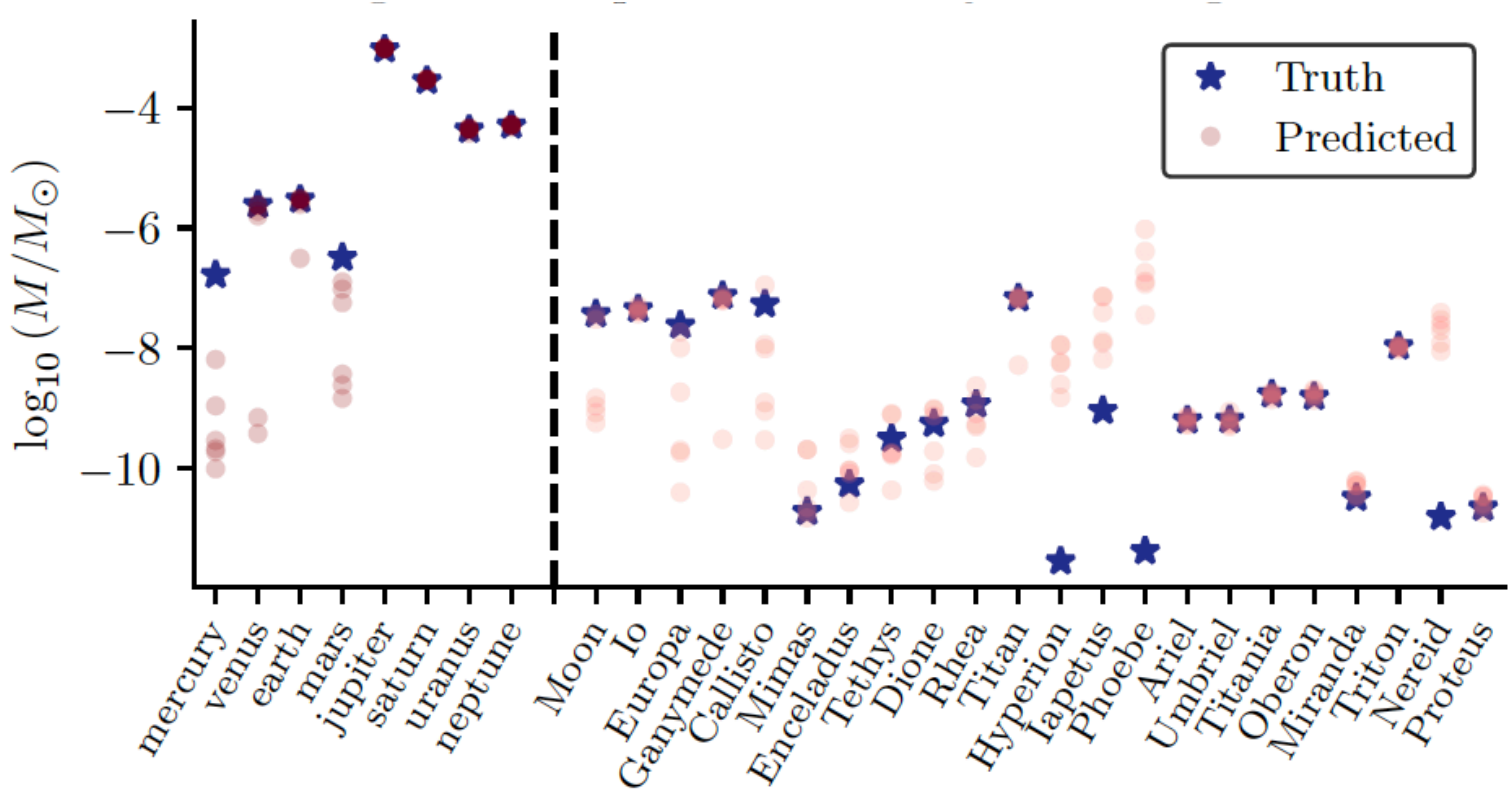
Why isn't this working well?

- Let's look at the mass values in comparison with the true masses:



Why isn't this working well?

- Let's look at the mass values in comparison with the true masses:



Solution: re-optimize v_i!

Solution: re-optimize v_i!

- The v_i were optimized for the neural network.

Solution: re-optimize v_i!

- The v_i were optimized for the neural network.
- The symbolic formula is not a *perfect* approximation of the network.

Solution: re-optimize v_i!

- The v_i were optimized for the neural network.
- The symbolic formula is not a *perfect* approximation of the network.
- Thus: we need to re-optimize v_i for the symbolic function f !


True
Predicted


True
Predicted

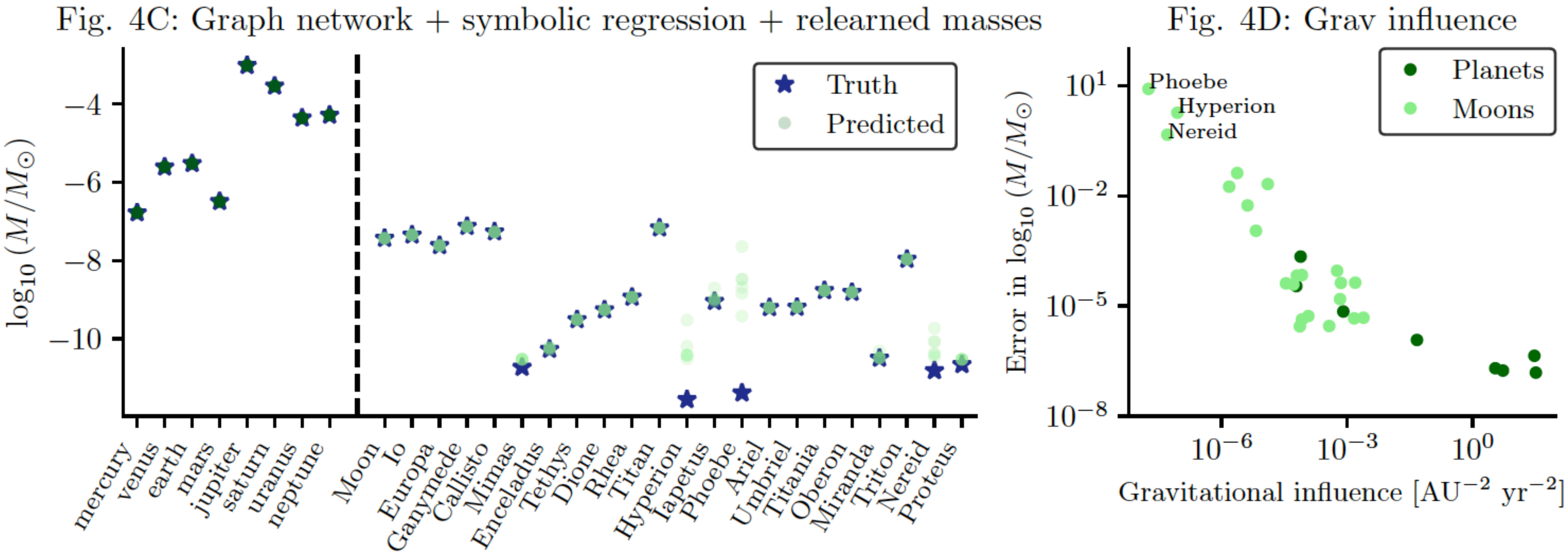
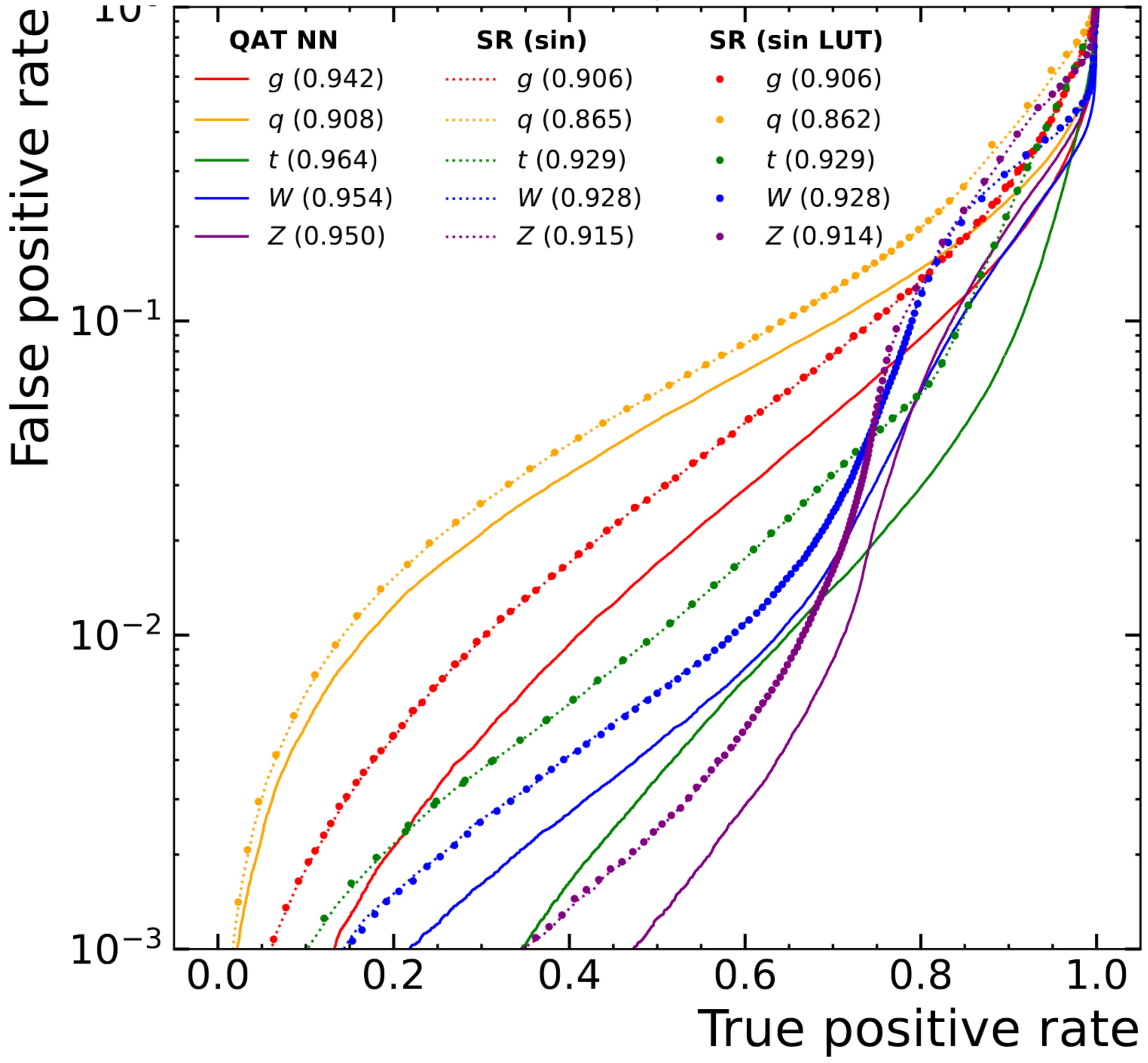


Fig. 4C: Graph network + symbolic regression + relearned masses
Fig. 4D: Grav influence
Truth
Predicted
$\log_{10}(M/M_\odot)$
−4
−6
−8
−10
mercury
venus
earth
mars
jupiter
saturn
uranus
neptune
Moon
Io
Europa
Ganymede
Callisto
Mimas
Enceladus
Tethys
Dione
Rhea
Titan
Hyperion
Iapetus
Phoebe
Ariel
Umbriel
Titania
Oberon
Miranda
Triton
Nereid
Proteus
Planets
Moons
Phoebe
Hyperion
Nereid
Error in $\log_{10}(M/M_\odot)$
10^{1}
10^{-2}
10^{-5}
10^{-8}
10^{-6}
10^{-3}
10^{0}
Gravitational influence [$\mathrm{AU}^{-2}\ \mathrm{yr}^{-2}$]

Symbolic Regression on FPGAs for Fast Machine Learning Inference

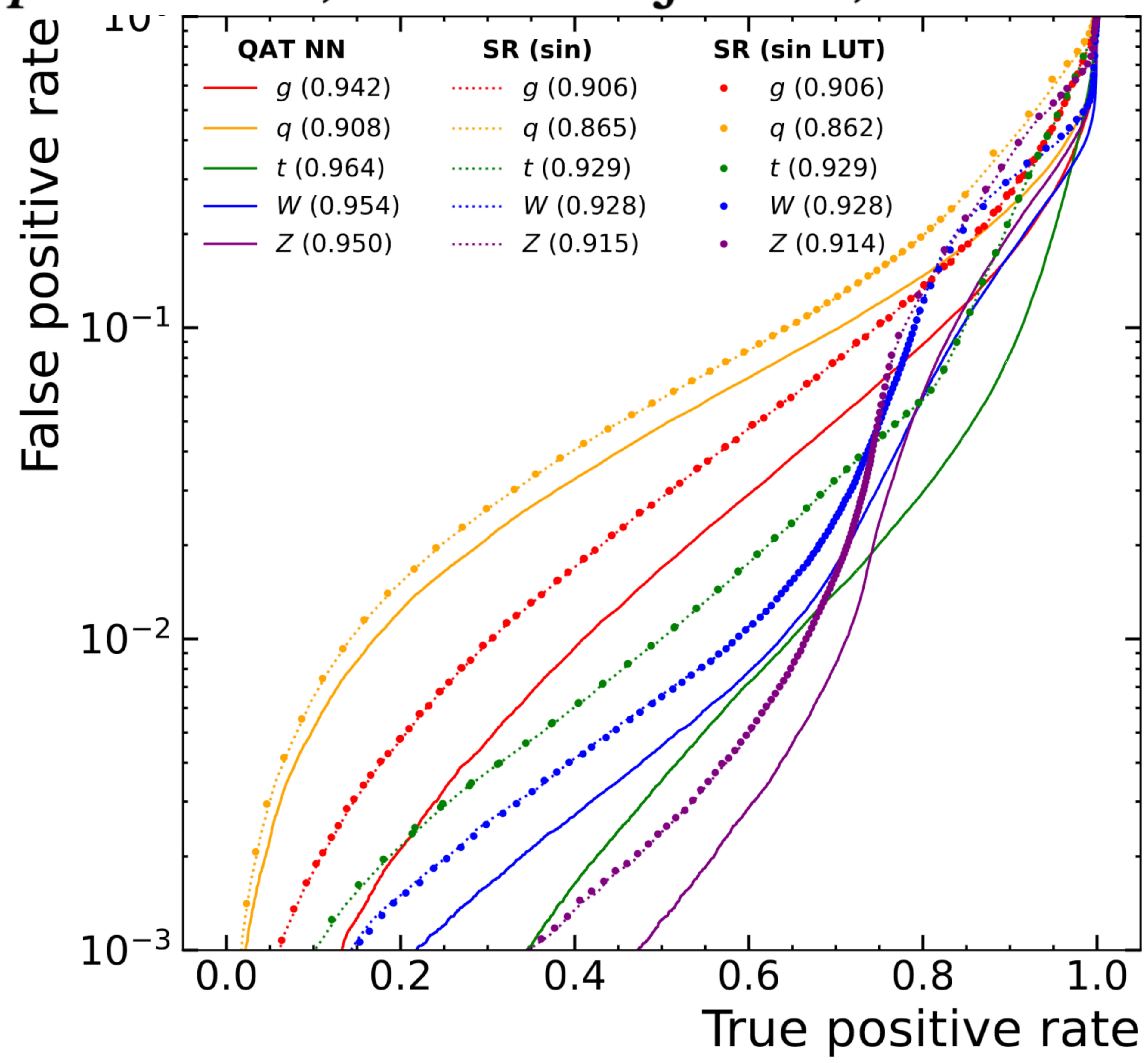
Ho Fung Tsoi[1*], *Adrian Alan* Pol[2], *Vladimir* Loncar[3,4], *Ekaterina* Govorkova[3], *Miles* Cranmer[2,5], *Sridhara* Dasu[1], *Peter* Elmer[2], *Philip* Harris[3], *Isobel* Ojalvo[2], and *Maurizio* Pierini[6]



Symbolic Regression on FPGAs for Fast Machine Learning Inference

Ho Fung Tsoi[1*], *Adrian Alan* Pol[2], *Vladimir* Loncar[3,4], *Ekaterina* Govorkova[3], *Miles* Cranmer[2,5], *Sridhara* Dasu[1], *Peter* Elmer[2], *Philip* Harris[3], *Isobel* Ojalvo[2], and *Maurizio* Pierini[6]

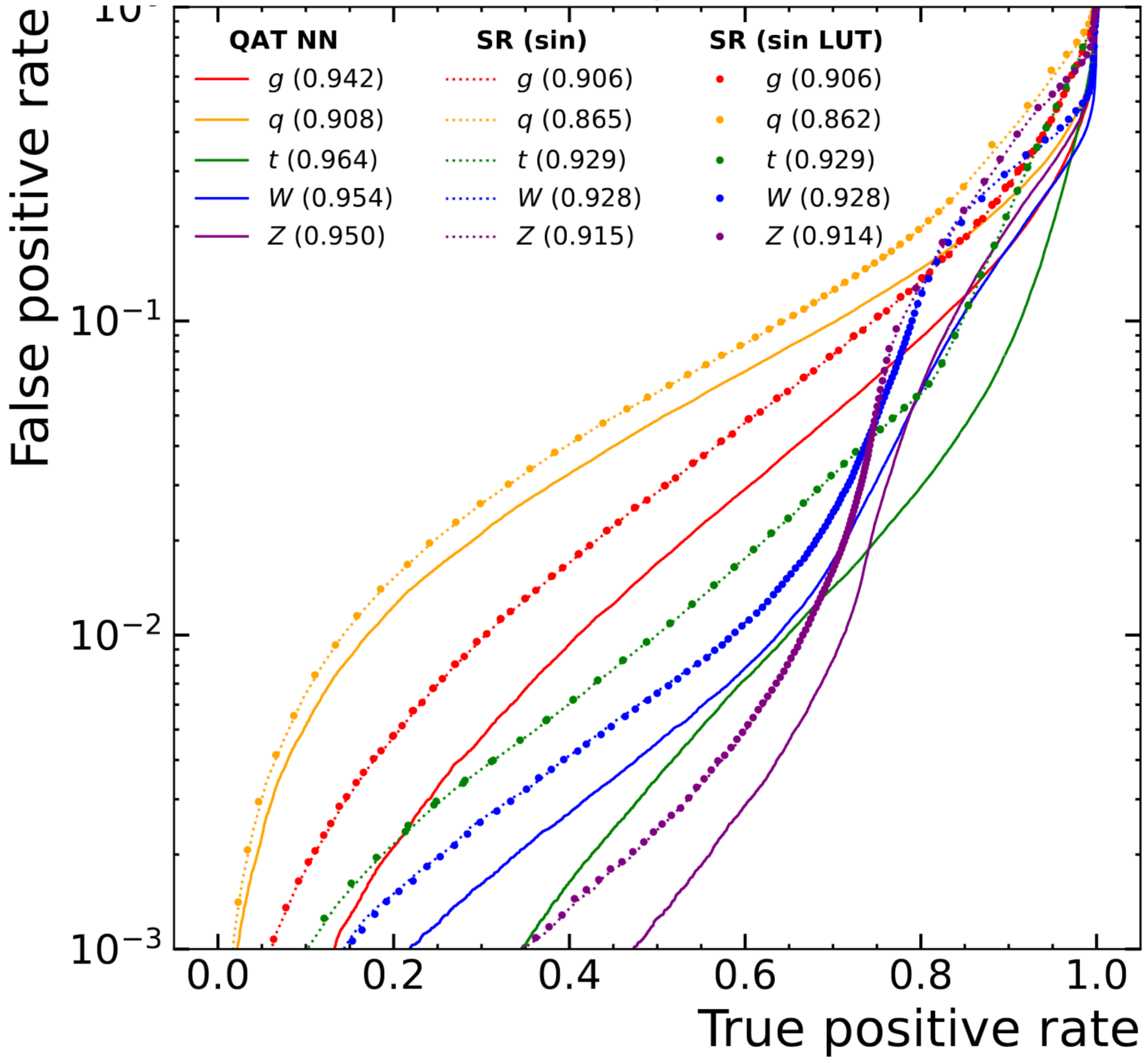
- "Complexity" of an operator = number of clock cycles on FPGA



Symbolic Regression on FPGAs for Fast Machine Learning Inference

Ho Fung Tsoi[1*], *Adrian Alan* Pol[2], *Vladimir* Loncar[3,4], *Ekaterina* Govorkova[3], *Miles* Cranmer[2,5], *Sridhara* Dasu[1], *Peter* Elmer[2], *Philip* Harris[3], *Isobel* Ojalvo[2], and *Maurizio* Pierini[6]

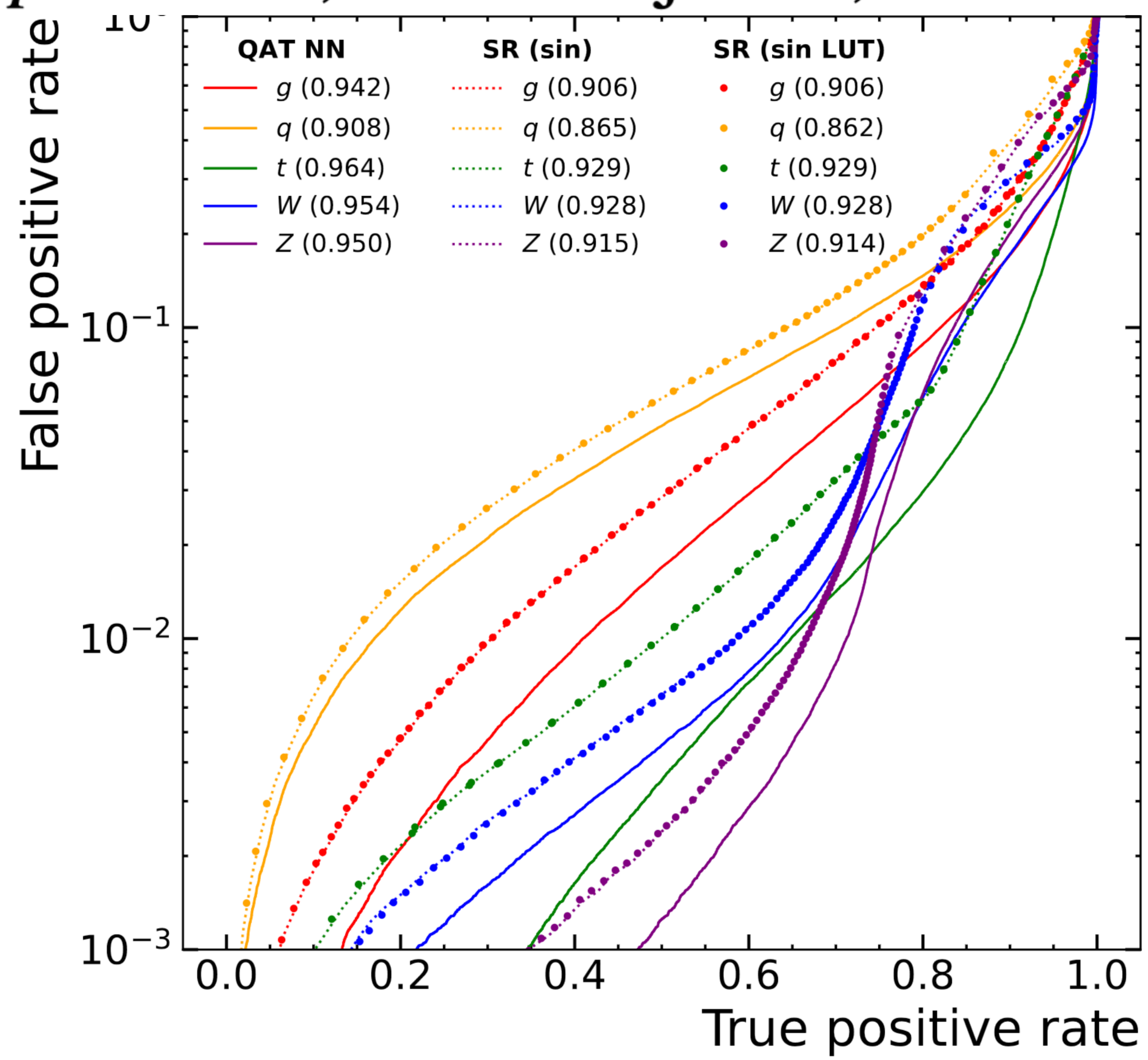
- "Complexity" of an operator = number of clock cycles on FPGA
- Approximate neural net with small expression = 90% accuracy



Symbolic Regression on FPGAs for Fast Machine Learning Inference

Ho Fung Tsoi[1*], *Adrian Alan* Pol[2], *Vladimir* Loncar[3,4], *Ekaterina* Govorkova[3], *Miles* Cranmer[2,5], *Sridhara* Dasu[1], *Peter* Elmer[2], *Philip* Harris[3], *Isobel* Ojalvo[2], and *Maurizio* Pierini[6]

- "Complexity" of an operator = number of clock cycles on FPGA
- Approximate neural net with small expression = 90% accuracy
- **5 ns inference time** on FPGA!



Discussion/Future

Discussion/Future

- Is a pure neural net approach to AI for science (i.e., no interpretation) possible? How would you get the same level of generalization as we have had from theory

Discussion/Future

- Is a pure neural net approach to AI for science (i.e., no interpretation) possible? How would you get the same level of generalization as we have had from theory
  - General relativity derived from only a few postulates/data points, yet can predict the existence of black holes. Is it hopeless to expect that level of generalization from foundation models?

Discussion/Future

- Is a pure neural net approach to AI for science (i.e., no interpretation) possible? How would you get the same level of generalization as we have had from theory
  - General relativity derived from only a few postulates/data points, yet can predict the existence of black holes. Is it hopeless to expect that level of generalization from foundation models?
- How do we distill very large models, like large language models, into the language of science?

Discussion/Future

- Is a pure neural net approach to AI for science (i.e., no interpretation) possible? How would you get the same level of generalization as we have had from theory
  - General relativity derived from only a few postulates/data points, yet can predict the existence of black holes. Is it hopeless to expect that level of generalization from foundation models?
- How do we distill very large models, like large language models, into the language of science?
  - These models may have learned some new unifying principles across domains. How can we find it?

Discussion/Future

- Is a pure neural net approach to AI for science (i.e., no interpretation) possible? How would you get the same level of generalization as we have had from theory
  - General relativity derived from only a few postulates/data points, yet can predict the existence of black holes. Is it hopeless to expect that level of generalization from foundation models?
- How do we distill very large models, like large language models, into the language of science?
  - These models may have learned some new unifying principles across domains. How can we find it?
- Can you use this symbolic regression technique to interpret language models directly?

FAQ: Why not fit directly?

FAQ: Why not fit directly?

- Constraints:

FAQ: Why not fit directly?

- Constraints:
  - Neural Networks require ~1M evaluations of a loss function to train.

FAQ: Why not fit directly?

- Constraints:
  - Neural Networks require ~1M evaluations of a loss function to train.
  - Genetic algorithm-based symbolic regression requires ~1B evaluations to find a complex+accurate expression.

FAQ: Why not fit directly?

- Constraints:
  - Neural Networks require ~1M evaluations of a loss function to train.
  - Genetic algorithm-based symbolic regression requires ~1B evaluations to find a complex+accurate expression.
  - Need symbolic regression loss to be **extremely efficient**!

FAQ: Why not fit directly?

- Constraints:
  - Neural Networks require ~1M evaluations of a loss function to train.
  - Genetic algorithm-based symbolic regression requires ~1B evaluations to find a complex+accurate expression.
  - Need symbolic regression loss to be **extremely efficient**!
- Offline vs online learning:

FAQ: Why not fit directly?

- <u>Constraints:</u>
  - Neural Networks require ~1M evaluations of a loss function to train.
  - Genetic algorithm-based symbolic regression requires ~1B evaluations to find a complex+accurate expression.
  - Need symbolic regression loss to be **extremely efficient**!
- <u>Offline vs online learning:</u>
  - Full loss is too expensive.

FAQ: Why not fit directly?

- <u>Constraints:</u>
  - Neural Networks require ~1M evaluations of a loss function to train.
  - Genetic algorithm-based symbolic regression requires ~1B evaluations to find a complex+accurate expression.
  - Need symbolic regression loss to be **extremely efficient**!
- <u>Offline vs online learning:</u>
  - Full loss is too expensive.
  - So, we do “online” learning of the neural net, and then fit the inputs/outputs of the network afterwards