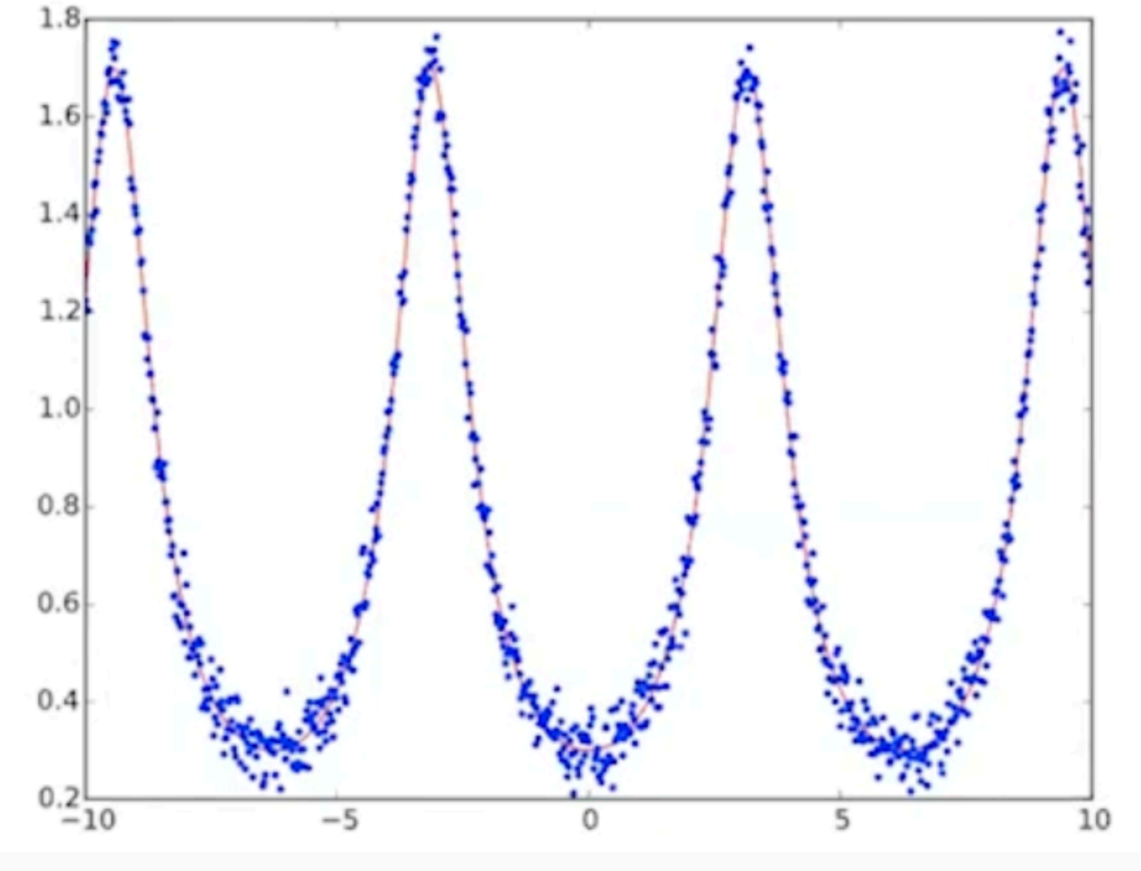
Symbolic Regression and Polynomial Optimization in Scientific Discovery

Karan Srivastava

Aim

- To discover meaningful laws of nature from experimental data

  i.e. Given some data $\{(x_1, \ldots, x_n, y)_i\}_{i \in \mathcal{D}}$ we want to find a function f so that $y = f(x_1, \ldots, x_n)$ for each data point.



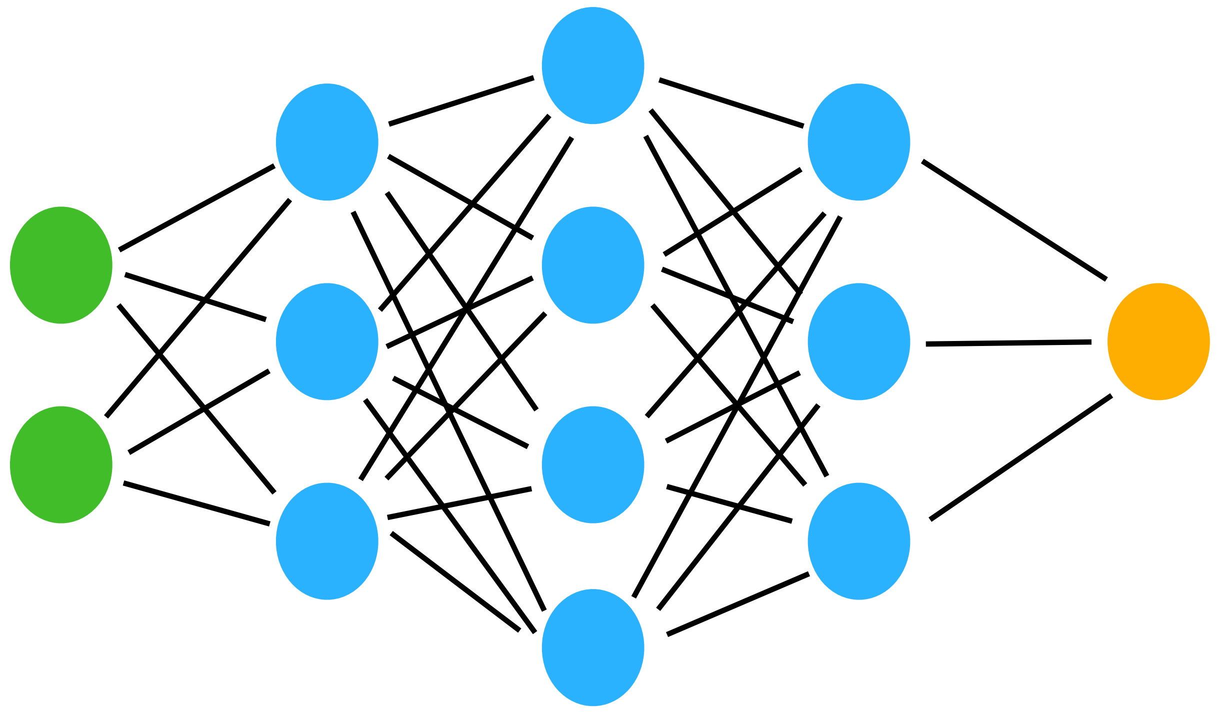
$$r = \frac{a(1 - e^2)}{1 + e\cos(\theta_1 - \theta_2)}$$

Aim

- Given some data $\{(x_1, ..., x_n, y)_i\}_{i \in \mathcal{D}}$ we want to find a function f so that $y = f(x_1, ..., x_n)$ for each data point.

<u>Some ideas</u>

<u>Deep Learning</u>



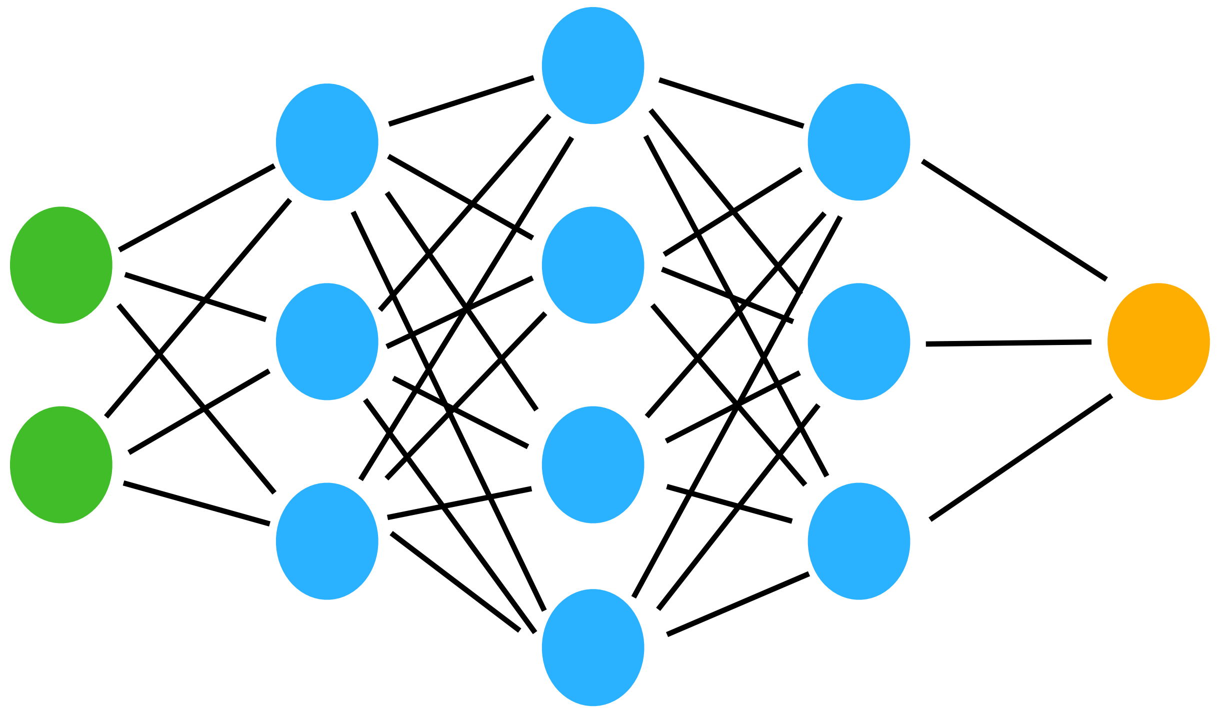
Pros: Good for discovering patterns in data
Drawback: Model itself is uninterpretable

Aim

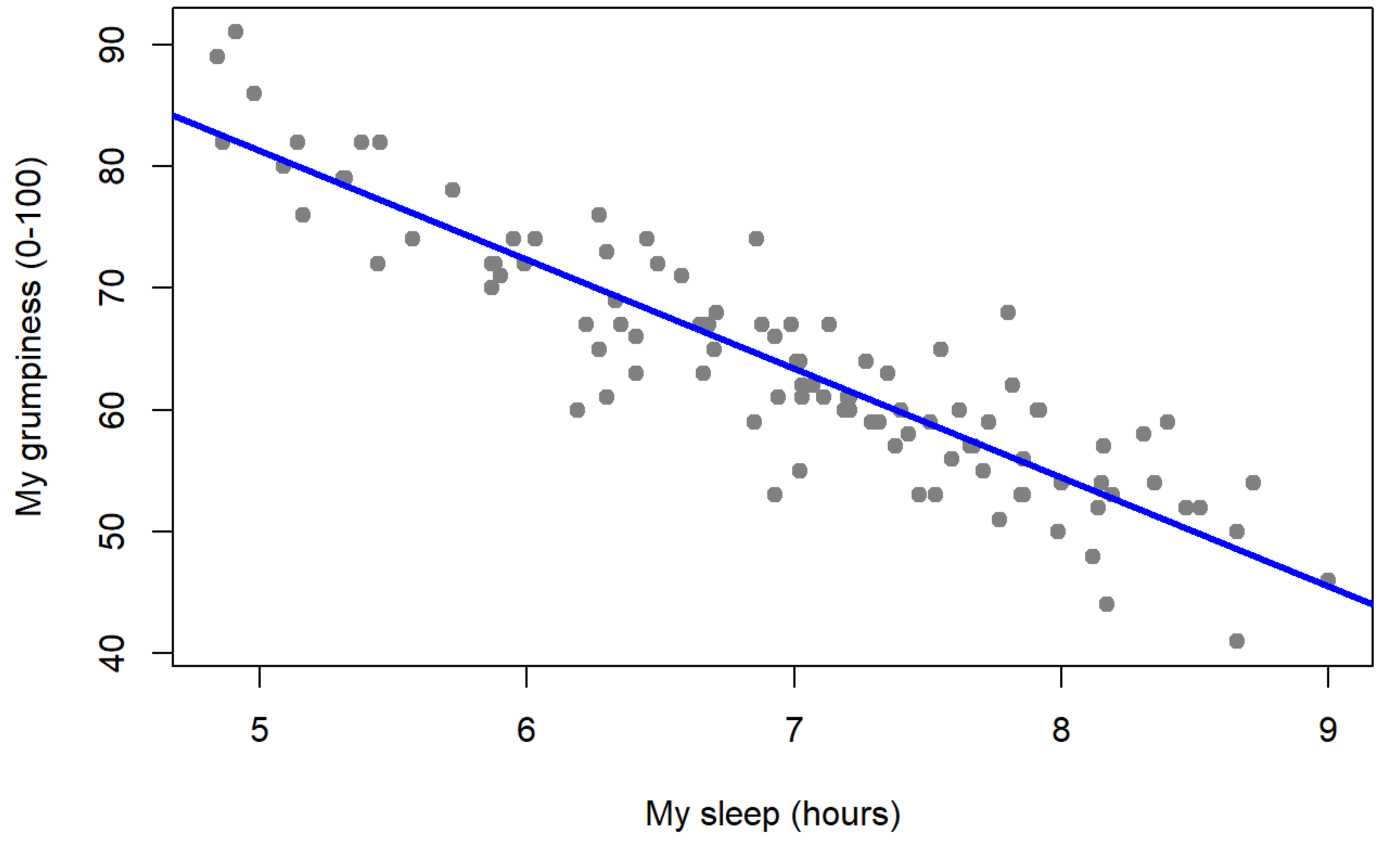
- Given some data $\{(x_1, ..., x_n, y)_i\}_{i \in \mathcal{D}}$ we want to find a function f so that $y = f(x_1, ..., x_n)$ for each data point.

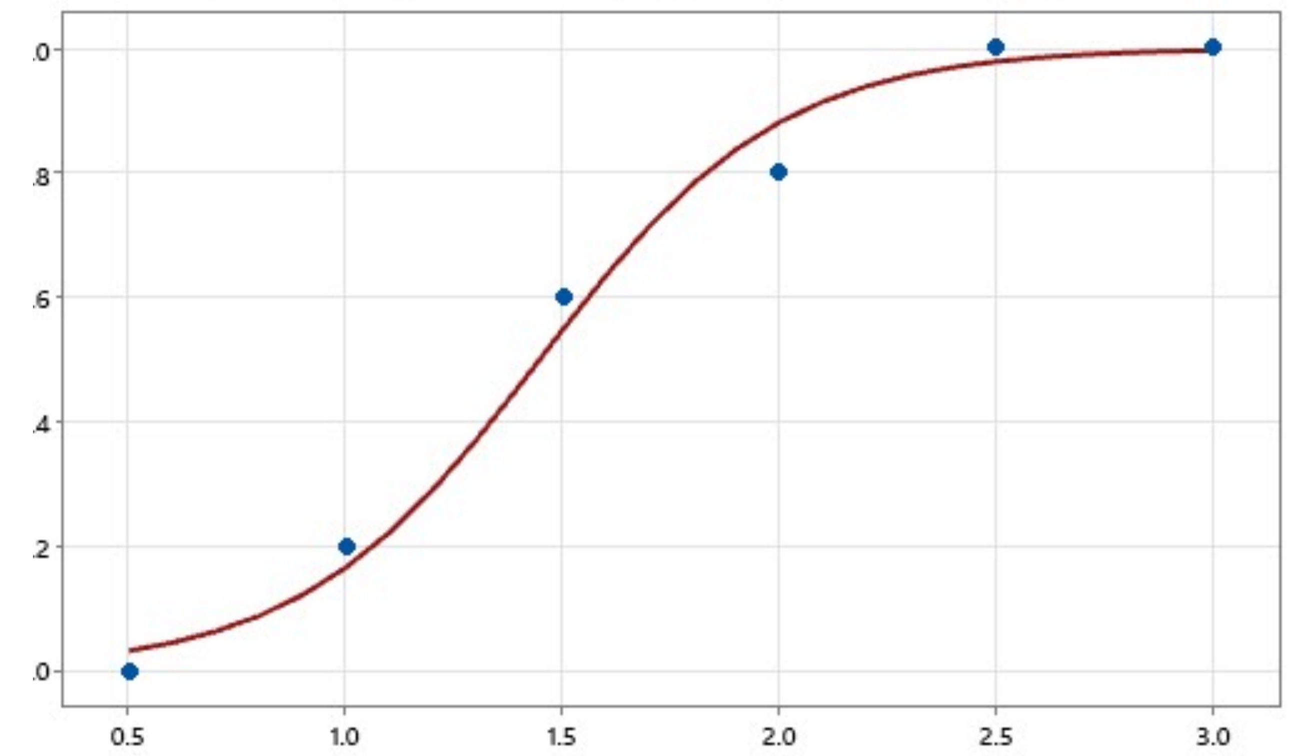
Some ideas

Deep Learning



Standard Regression





Pros: Good for discovering patterns in data
Drawback: Model itself is uninterpretable

Pros: Models are very interpretable
Drawback: Functional form is fixed

Approach 1: Symbolic Regression

Approach 1: Symbolic Regression

In particular, we will look at the approach in this paper:

AI Feynman: a Physics-Inspired Method for Symbolic Regression

Silviu-Marian Udrescu, Max Tegmark*

Dept. of Physics & Center for Brains, Minds & Machines, Massachusetts Institute of Technology, Cambridge, MA 02139; sudrescu@mit.edu and Theiss Research, La Jolla, CA 92037, USA

(Dated: Published in *Science Advances*, 6:eaay2631, April 15, 2020)

A core challenge for both physics and artificial intelligence (AI) is *symbolic regression*: finding a symbolic expression that matches data from an unknown function. Although this problem is likely to be NP-hard in principle, functions of practical interest often exhibit symmetries, separability, compositionality and other simplifying properties. In this spirit, we develop a recursive multidimensional symbolic regression algorithm that combines neural network fitting with a suite of physics-inspired techniques. We apply it to 100 equations from the Feynman Lectures on Physics, and it discovers all of them, while previous publicly available software cracks only 71; for a more difficult physics-based test set, we improve the state of the art success rate from 15% to 90%.

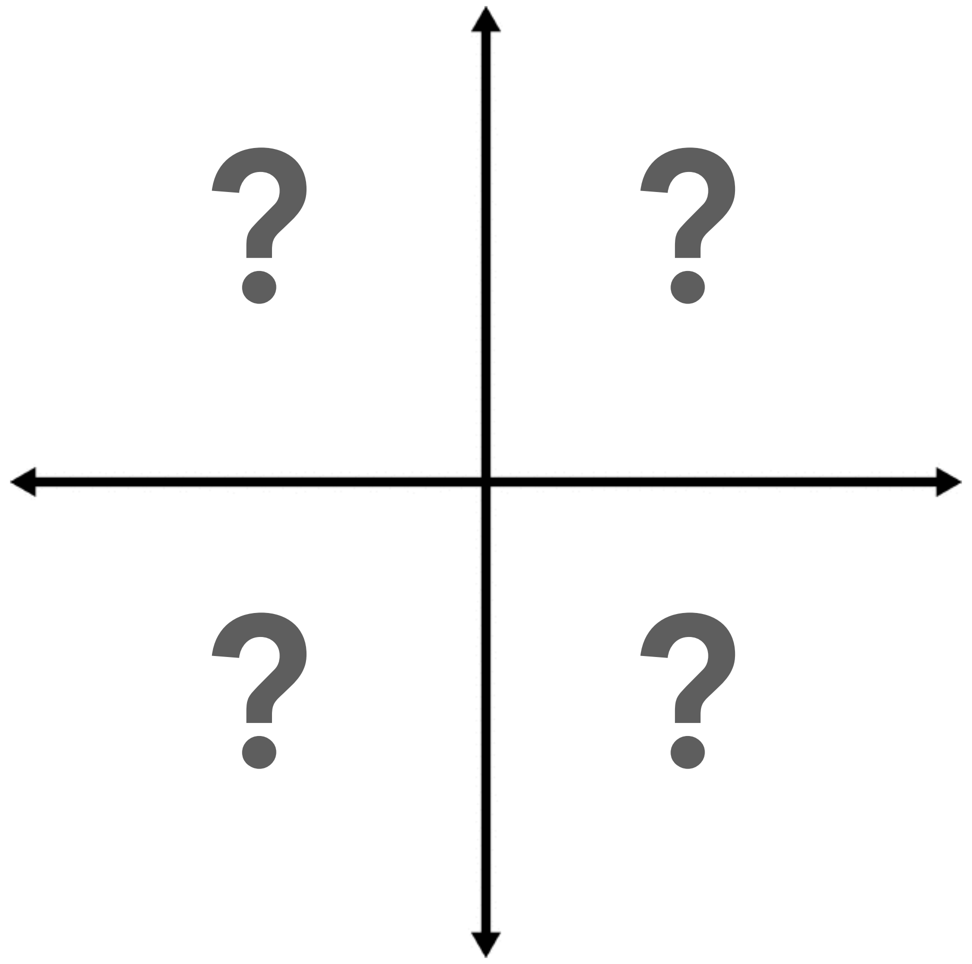
I. INTRODUCTION

In 1601, Johannes Kepler got access to the world's best data tables on planetary orbits, and after 4 years and about 40 failed attempts to fit the Mars data to various ovoid shapes, he launched a scientific revolution by discovering that Mars' orbit was an ellipse [1]. This was an example of *symbolic regression*: discovering a symbolic expression that accurately matches a given data set. More specifically, we are given a table of numbers, whose rows are of the form $\{x_1, ..., x_n, y\}$ where $y = f(x_1, ..., x_n)$, and our task is to discover the correct symbolic expression for the unknown mystery function f, optionally including the complication of noise.

Growing data sets have motivated attempts to automate such regression tasks, with significant success. For the search space characterizes many famous classes of problems, from codebreaking and Rubik's cube to the natural selection problem of finding those genetic codes that produce the most evoutionarily fit organisms. This has motivated *genetic algorithms* [2, 3] for targeted searches in exponentially large spaces, which replace the abovementioned brute-force search by biology-inspired strategies of mutation, selection, inheritance and recombination; crudely speaking, the role of genes is played by useful symbol strings that may form part of the sought-after formula or program. Such algorithms have been successfully applied to areas ranging from design of antennas [4, 5] and vehicles [6] to wireless routing [7], vehicle routing [8], robot navigation [9], code breaking [10], discovering partial differential equations [11], investment strategy [12], marketing [13], classification [14], Rubik's cube [15], program synthesis [16] and metabolic networks [17].

Symbolic Regression

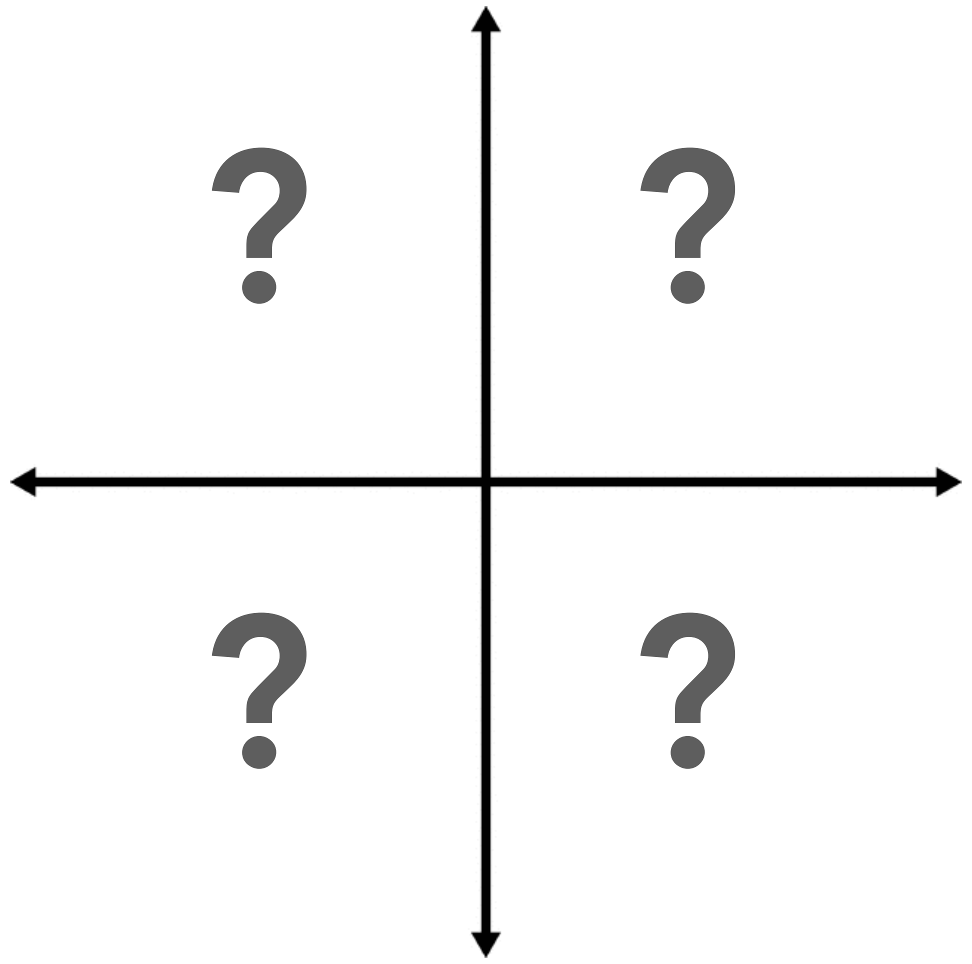
- Given some data $\{(x_1, \ldots, x_n, y)_i\}_{i \in \mathscr{D}}$ we want to find a function f so that $y = f(x_1, \ldots, x_n)$ for each data point **when the functional form of f is unknown**.



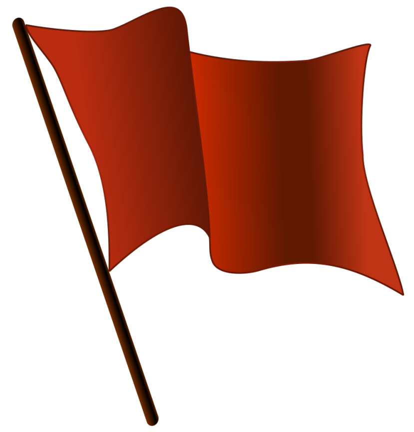
- In particular, given a set S of symbols (e.g. $+, -, \div, \sqrt{}$, etc), find a function (string) f built from these symbols so that $y = f(x_1, \ldots, x_n)$ fits the data.

Symbolic Regression

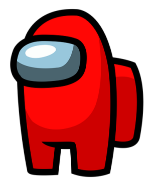
- Given some data $\{(x_1, ..., x_n, y)_i\}_{i \in \mathcal{D}}$ we want to find a function f so that $y = f(x_1, ..., x_n)$ for each data point **when the functional form of f is unknown**.



- In particular, given a set S of symbols (e.g. $+, -, \div, \sqrt{}$,etc), find a function (string) f built from these symbols so that $y = f(x_1, ..., x_n)$ fits the data.

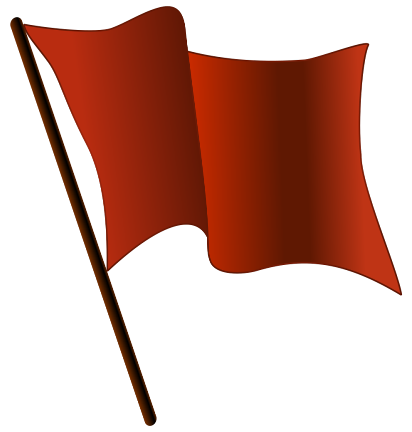


For generic functions $f(x_1, ..., x_n)$, this is NP hard



Symbolic Regression

- In particular, given a set S of symbols (e.g. $+, -, \div, \sqrt{}$, etc), find a function (string) f built from these symbols so that $y = f(x_1, \ldots, x_n)$ fits the data.



For generic functions $f(x_1, \ldots, x_n)$, this is NP hard 

- Search grows exponentially with the number of symbols.

- Brute force search becomes infeasible very quickly

$$f = \frac{e^{-\frac{x^2}{2}}}{\sqrt{2\pi}} \sim 30 \text{ years}, \quad \omega = \frac{1 + \frac{v}{c}}{\sqrt{1 - \frac{v^2}{c^2}}} \sim 10^6 \text{ years, and so on.}$$

Symbolic Regression

Feynman eq.	Equation
I.6.20a	$f = e^{-\theta^2/2}/\sqrt{2\pi}$
I.6.20	$f = e^{-\frac{\theta^2}{2\sigma^2}}/\sqrt{2\pi\sigma^2}$
I.6.20b	$f = e^{-\frac{(\theta-\theta_1)^2}{2\sigma^2}}/\sqrt{2\pi\sigma^2}$
I.8.14	$d = \sqrt{(x_2-x_1)^2+(y_2-y_1)^2}$
I.9.18	$F = \frac{Gm_1m_2}{(x_2-x_1)^2+(y_2-y_1)^2+(z_2-z_1)^2}$
I.10.7	$m = \frac{m_0}{\sqrt{1-\frac{v^2}{c^2}}}$
I.11.19	$A = x_1y_1+x_2y_2+x_3y_3$
I.12.1	$F = \mu N_n$
I.12.2	$F = \frac{q_1q_2}{4\pi\epsilon r^2}$
I.12.4	$E_f = \frac{q_1}{4\pi\epsilon r^2}$
I.12.5	$F = q_2E_f$
I.12.11	$F = q(E_f + Bv\sin\theta)$
I.13.4	$K = \frac{1}{2}m(v^2+u^2+w^2)$
I.13.12	$U = Gm_1m_2(\frac{1}{r_2}-\frac{1}{r_1})$
I.14.3	$U = mgz$
I.14.4	$U = \frac{k_{spring}x^2}{2}$
I.15.3x	$x_1 = \frac{x-ut}{\sqrt{1-u^2/c^2}}$

BUT

SCIENCE IS

NOT

GENERIC

Symbolic Regression

In physics and in lots of science applications, functions we care about tend to be nice in the following ways:

1. Units: f and its variables have to be dimensionally consistent

2. Low degree polynomials: Parts of f tend to have low degree polynomials

3. Compositionality: f is a composition of elementary functions

4. Smoothness: f is continuous and often analytic on its domain

5. Symmetry: f comes with translational, rotational, and scaling symmetry with respect to some variables

6. Separability: f can be written as the sum or product of two parts with no common variables

Symbolic Regression

In physics and in lots of science applications, functions we care about tend to be nice in the following ways:

1. Units: Can use this to transform the problem to fewer independent variables

2. Low degree polynomials: Parts of f tend to have low degree polynomials

3. Compositionality: f is a composition of elementary functions

4. Smoothness: f is continuous and often analytic on its domain

5. Symmetry: f comes with translational, rotational, and scaling symmetry with respect to some variables

6. Separability: f can be written as the sum or product of two parts with no common variables

Symbolic Regression

In physics and in lots of science applications, functions we care about tend to be nice in the following ways:

1. Units: Can use this to transform the problem to fewer independent variables

2. Low degree polynomials: Parts of f tend to have low degree polynomials

3. Compositionality: f is a composition of elementary functions

4. Smoothness: f is continuous and often analytic on its domain

5. Symmetry: f comes with translational, rotational, and scaling symmetry with respect to some variables

6. Separability: f can be written as the sum or product of two parts with no common variables

Symbolic Regression

In physics and in lots of science applications, functions we care about tend to be nice in the following ways:

1. Units: Can use this to transform the problem to fewer independent variables

2. Low degree polynomials: Can use this to do polynomial fits for parts of f

3. Compositionality: f is a composition of elementary functions

4. Smoothness: f is continuous and often analytic on its domain

5. Symmetry: f comes with translational, rotational, and scaling symmetry with respect to some variables

6. Separability: f can be written as the sum or product of two parts with no common variables

Symbolic Regression

In physics and in lots of science applications, functions we care about tend to be nice in the following ways:

1. Units: Can use this to transform the problem to fewer independent variables

2. Low degree polynomials: Can use this to do polynomial fits for parts of f

3. Compositionality: f is a composition of elementary functions

4. Smoothness: f is continuous and often analytic on its domain

5. Symmetry: f comes with translational, rotational, and scaling symmetry with respect to some variables

6. Separability: f can be written as the sum or product of two parts with no common variables

Symbolic Regression

In physics and in lots of science applications, functions we care about tend to be nice in the following ways:

1. Units: Can use this to transform the problem to fewer independent variables

2. Low degree polynomials: Can use this to do polynomial fits for parts of f

3. Compositionality: Can use brute force for parts of the function

4. Smoothness: f is continuous and often analytic on its domain

5. Symmetry: f comes with translational, rotational, and scaling symmetry with respect to some variables

6. Separability: f can be written as the sum or product of two parts with no common variables

Symbolic Regression

In physics and in lots of science applications, functions we care about tend to be nice in the following ways:

1. Units: Can use this to transform the problem to fewer independent variables

2. Low degree polynomials: Can use this to do polynomial fits for parts of f

3. Compositionality: Can use brute force for parts of the function

4. Smoothness: f is continuous and often analytic on its domain

5. Symmetry: f comes with translational, rotational, and scaling symmetry with respect to some variables

6. Separability: f can be written as the sum or product of two parts with no common variables

Symbolic Regression

In physics and in lots of science applications, functions we care about tend to be nice in the following ways:

1. Units: Can use this to transform the problem to fewer independent variables

2. Low degree polynomials: Can use this to do polynomial fits for parts of f

3. Compositionality: Can use brute force for parts of the function

4. Smoothness: f can be learned by a NN

5. Symmetry: f comes with translational, rotational, and scaling symmetry with respect to some variables

6. Separability: f can be written as the sum or product of two parts with no common variables

Symbolic Regression

In physics and in lots of science applications, functions we care about tend to be nice in the following ways:

1. Units: Can use this to transform the problem to fewer independent variables

2. Low degree polynomials: Can use this to do polynomial fits for parts of f

3. Compositionality: Can use brute force for parts of the function

4. Smoothness: f can be learned by a NN

5. Symmetry: f comes with translational, rotational, and scaling symmetry with respect to some variables

6. Separability: f can be written as the sum or product of two parts with no common variables

Symbolic Regression

In physics and in lots of science applications, functions we care about tend to be nice in the following ways:

1. Units: Can use this to transform the problem to fewer independent variables

2. Low degree polynomials: Can use this to do polynomial fits for parts of f

3. Compositionality: Can use brute force for parts of the function

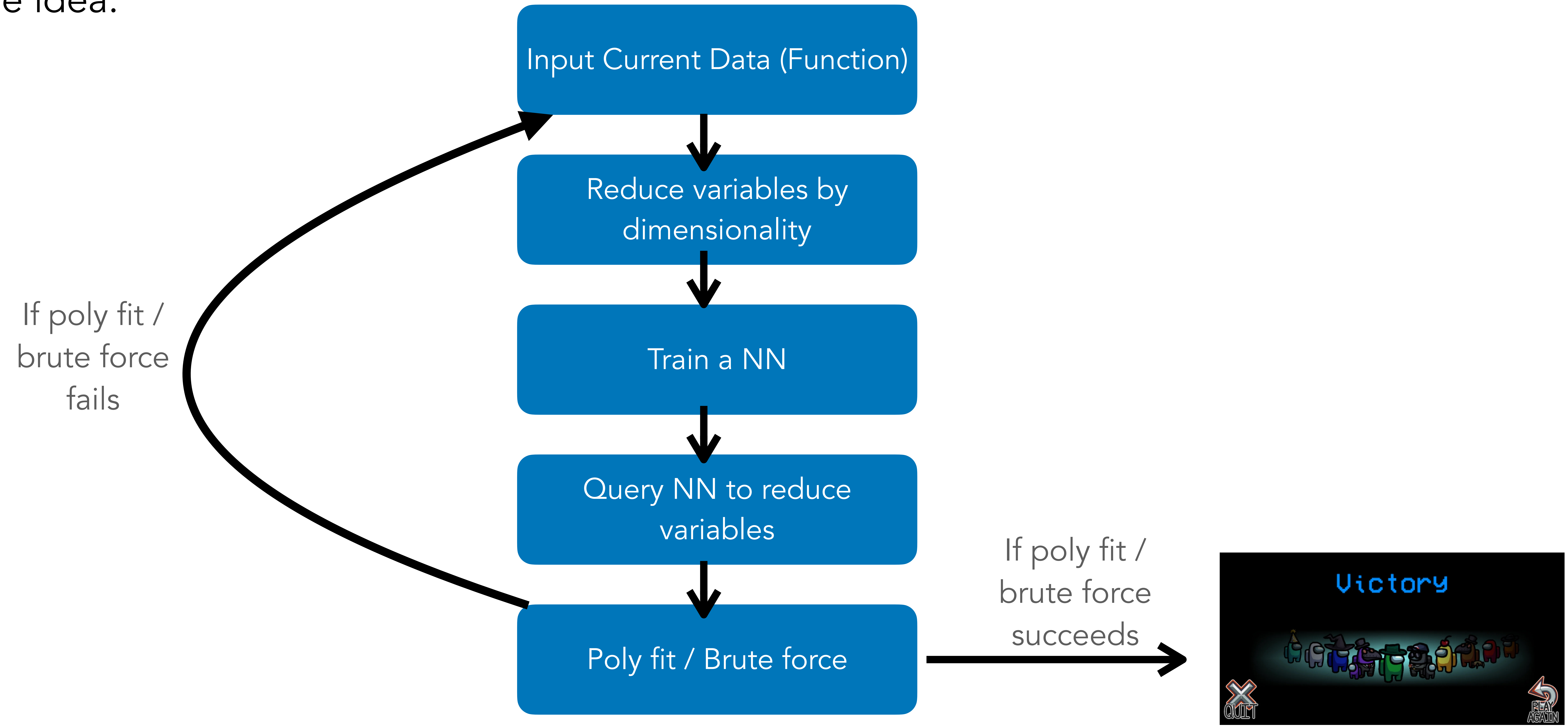
4. Smoothness: f can be learned by a NN

5. Symmetry: Can be tested by a neural network trained on the data

6. Separability: Can be tested by a neural network trained on the data

Symbolic Regression

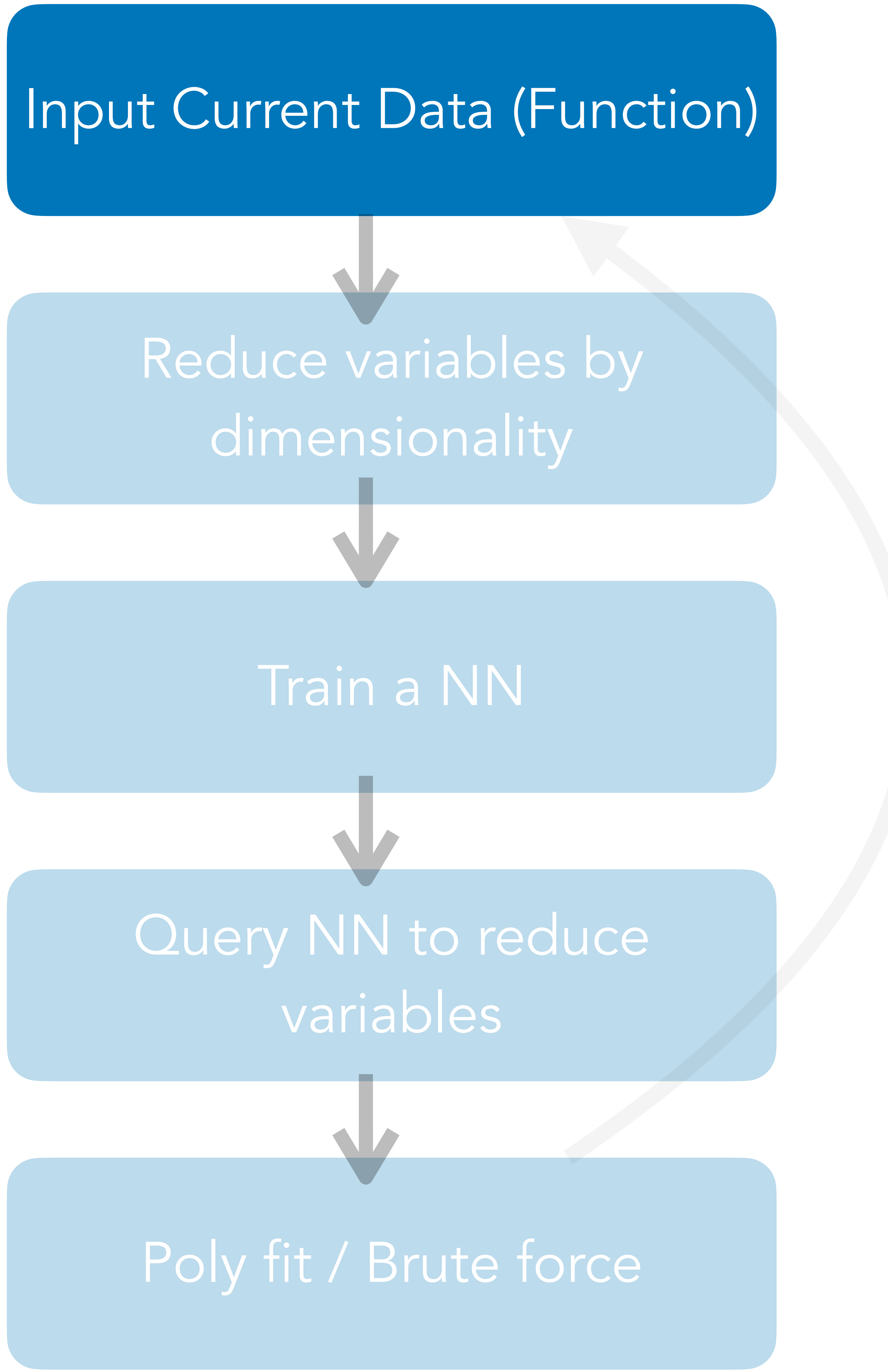
The idea:



Symbolic Regression - Example

$$\frac{Gm_1m_2}{(x_2-x_1)^2+(y_2-y_1)^2+(z_2-z_1)^2}$$

1.1431209959193709	2.7700644791483753	1.7508575540193472	0.23452895063856676
2.4680655653881054	2.2073166947348444	1.7761705838854234	0.15919403345867914
2.7621479700455853	1.4168131204210188	1.5378176974809339	0.14429337334417677
1.9536888384746354	2.7336267945043491	1.2592849110534683	0.15360014539410058
2.1532278876457527	1.5016008010765851	1.4218686278023172	0.18514940761978987
1.9899434091665062	1.4250958039594244	2.5409132056932424	0.17131474581788358
1.2841783534277345	2.5038413591290976	1.2255232096430668	0.18928439532548705
2.3550853261290494	2.2555822345853405	1.4525468706453792	0.15982929556091857
2.7529820467784543	2.6405850369222492	1.6148891450043024	0.13519598787103054
1.2043936184306594	1.4441117081403013	1.4546229392136278	0.33122650381723288
1.3423962980280737	2.0552387587225684	2.8071816262301414	0.25403615776576924
1.4156715177406782	1.0577553334831364	2.5122383795854709	0.16623813635214552
2.0187117984150182	1.0672568295716016	1.4729243386484654	0.19367199411465894
1.5109507435415031	2.5932595628763617	2.9492633493443927	0.25680582940703343
2.5078719313855347	1.0771974670079432	2.7296063077362385	0.12803644285479733
1.4846972064278652	2.7401242687034313	1.3356870260708698	0.17177799061908141
2.1398682204221178	1.5010412666279194	2.0784556152016664	0.17976791449866503
1.7000728811732311	1.5155671571998763	1.5295642030291683	0.23465391617605863
2.3516545980830331	1.0329627134609363	2.4473327905843174	0.14157574334122566
2.6157973581192184	2.0432144044920815	1.4576627239471807	0.14873897153317067

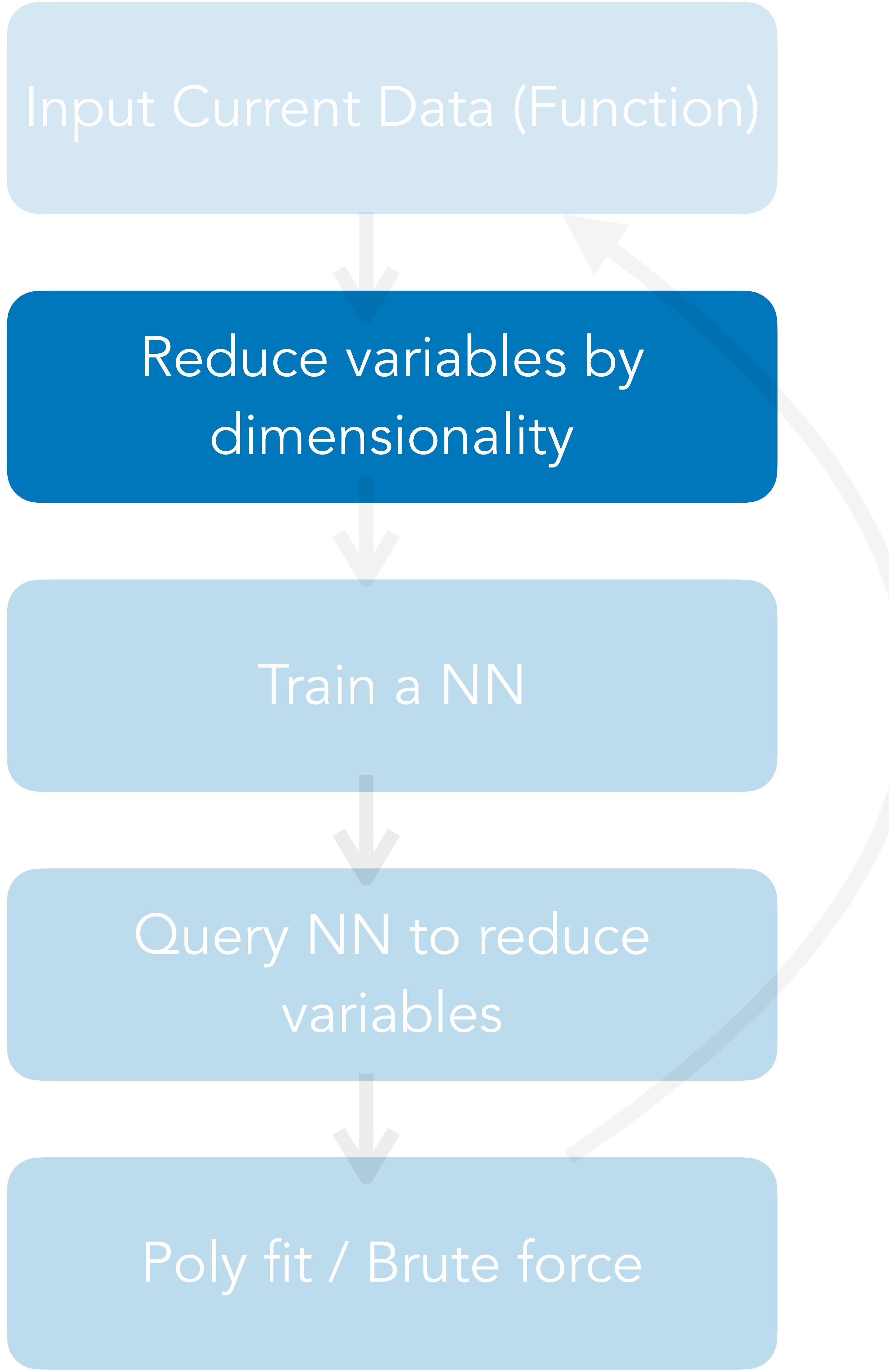


Symbolic Regression - Example

Idea: Because f is dimensionally consistent, we can factor it into a component that encodes the dimension and a dimensionless term

$$\frac{Gm_1m_2}{(x_2-x_1)^2+(y_2-y_1)^2+(z_2-z_1)^2}$$

$$\frac{Gm_1^2}{x_1^2} \qquad \frac{\frac{m_2}{m_1}}{(\frac{x_2}{x_1}-1)^2+(\frac{y_2}{x_1}-\frac{y_1}{x_1})^2+(\frac{z_2}{x_1}-\frac{z_1}{x_1})^2}$$



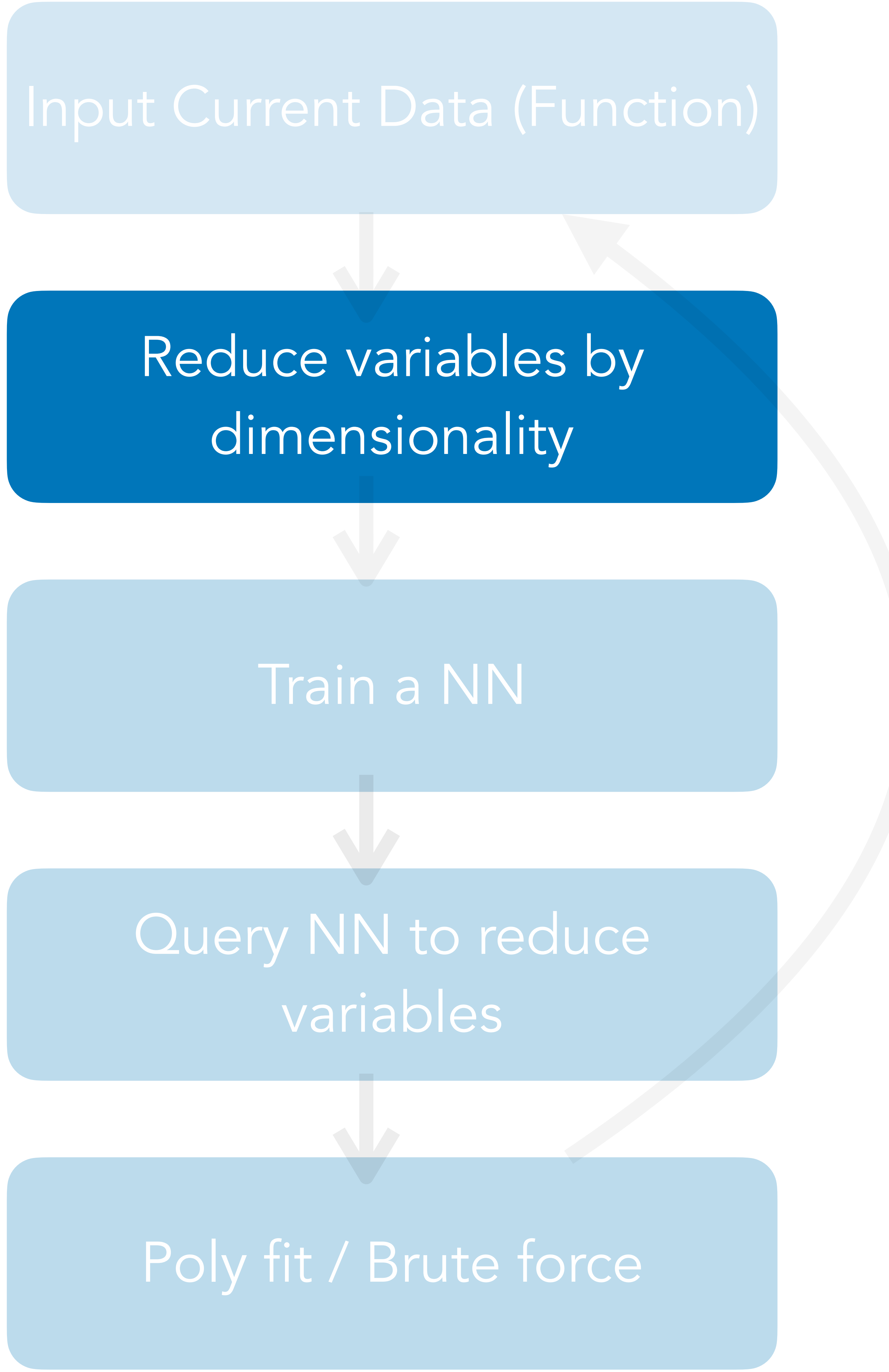
Symbolic Regression - Example

Idea: Because f is dimensionally consistent, we can factor it into a component that encodes the dimension and a dimensionless term

This reduces the number of variables to consider.

$$\frac{Gm_1m_2}{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

$$\frac{Gm_1^2}{x_1^2} \qquad \frac{a}{(b-1)^2 + (c-d)^2 + (e-f)^2}$$



Symbolic Regression - Example

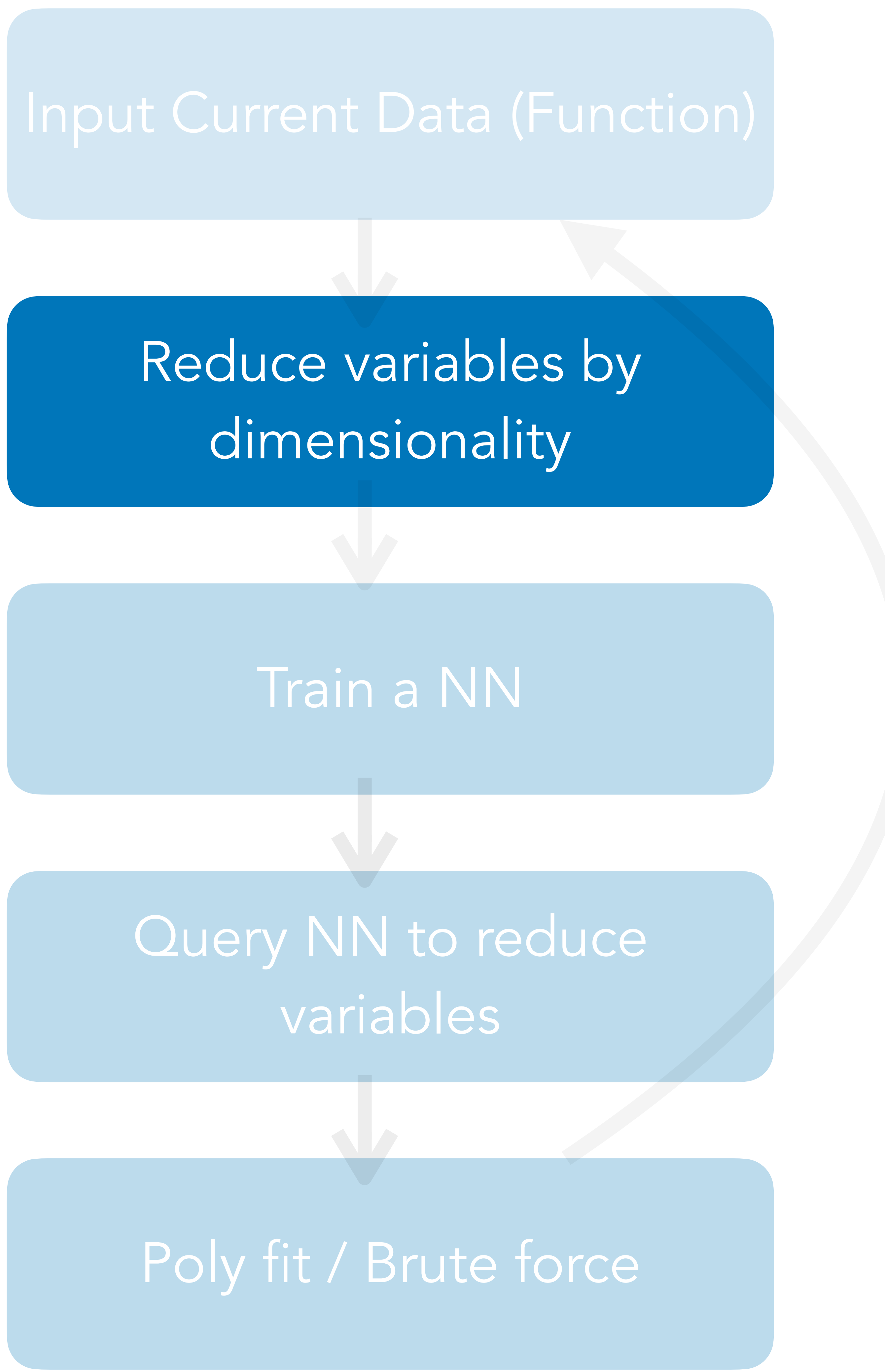
Idea: Because f is dimensionally consistent, we can factor it into a component that encodes the dimension and a dimensionless term

Represent each variable x_i by a 5 integer vector $\mathbf{u_i}$ corresponding to the fundamental units of the vector (being meter, second, kilogram, kelvin, volt). Let $\mathbf{M}$ be a matrix whose ith column is $\mathbf{u_i}$. Define $\mathbf{b}$ to be the corresponding vector for y.

Let $\mathbf{p}$ be a solution to $\mathbf{Mp} = \mathbf{b}$ and $\mathbf{U}$ be a basis for the null space of $\mathbf{M}$

Then apply

$$x_i \mapsto \prod_{i=j}^{n} x_j^{U_i j}, y \mapsto \frac{y}{y_*}, \text{ where } y_* = \prod_{i=1}^{n} x_i^{p_i}$$



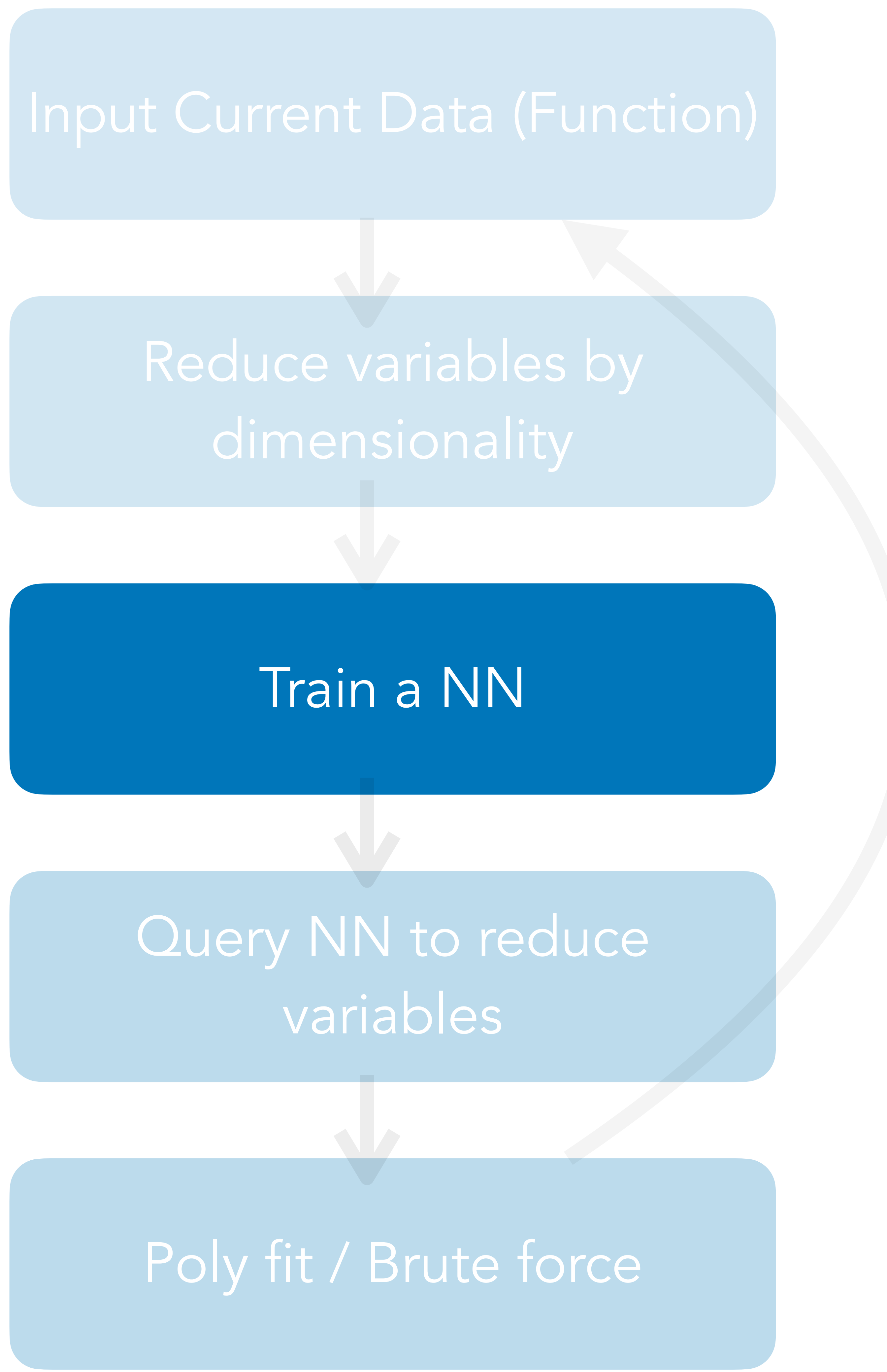
Symbolic Regression - Example

Train a NN on the transformed columns. This is a black box oracle that we can query.

$$\frac{a}{(b-1)^2 + (c-d)^2 + (e-f)^2}$$

Architecture used:

Dimension of layers: (128,128,128,64,64,64)
Epochs: 100
Learning rate: 0.005
Batch Size: 2048
Rms loss and Adam optimization
Weight decay: 10^{-2}



Symbolic Regression - Example

Test for symmetry, separability, and other properties and reduce variables accordingly

$$\frac{a}{(b-1)^2+(c-d)^2+(e-f)^2}$$

Translational Symmetry

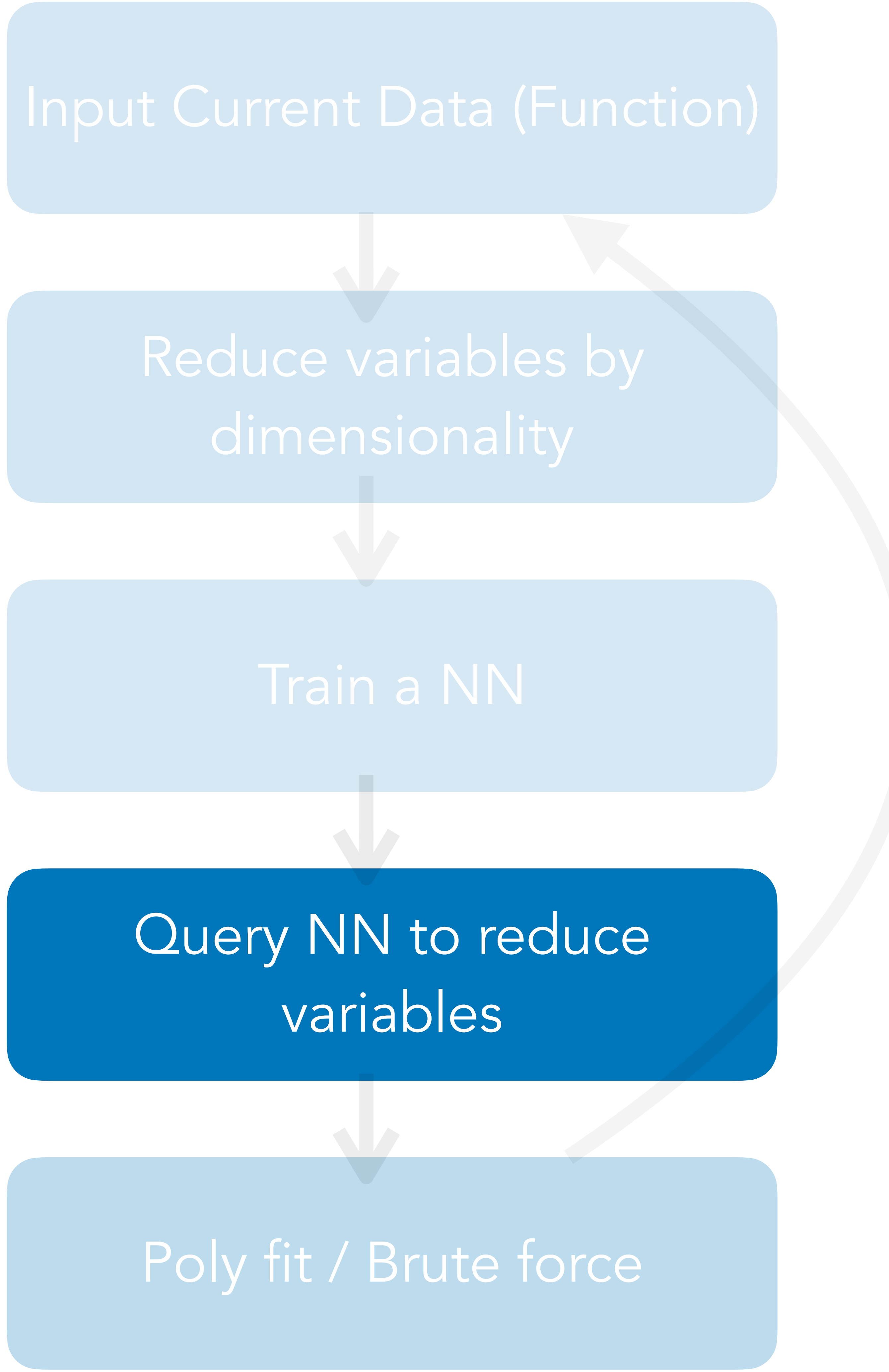
$$\frac{a}{(b-1)^2+g^2+(e-f)^2}$$

Translational Symmetry

$$\frac{a}{(b-1)^2+g^2+h^2}$$

Multiplicative Separability

$$a \qquad \frac{1}{(b-1)^2+g^2+h^2}$$



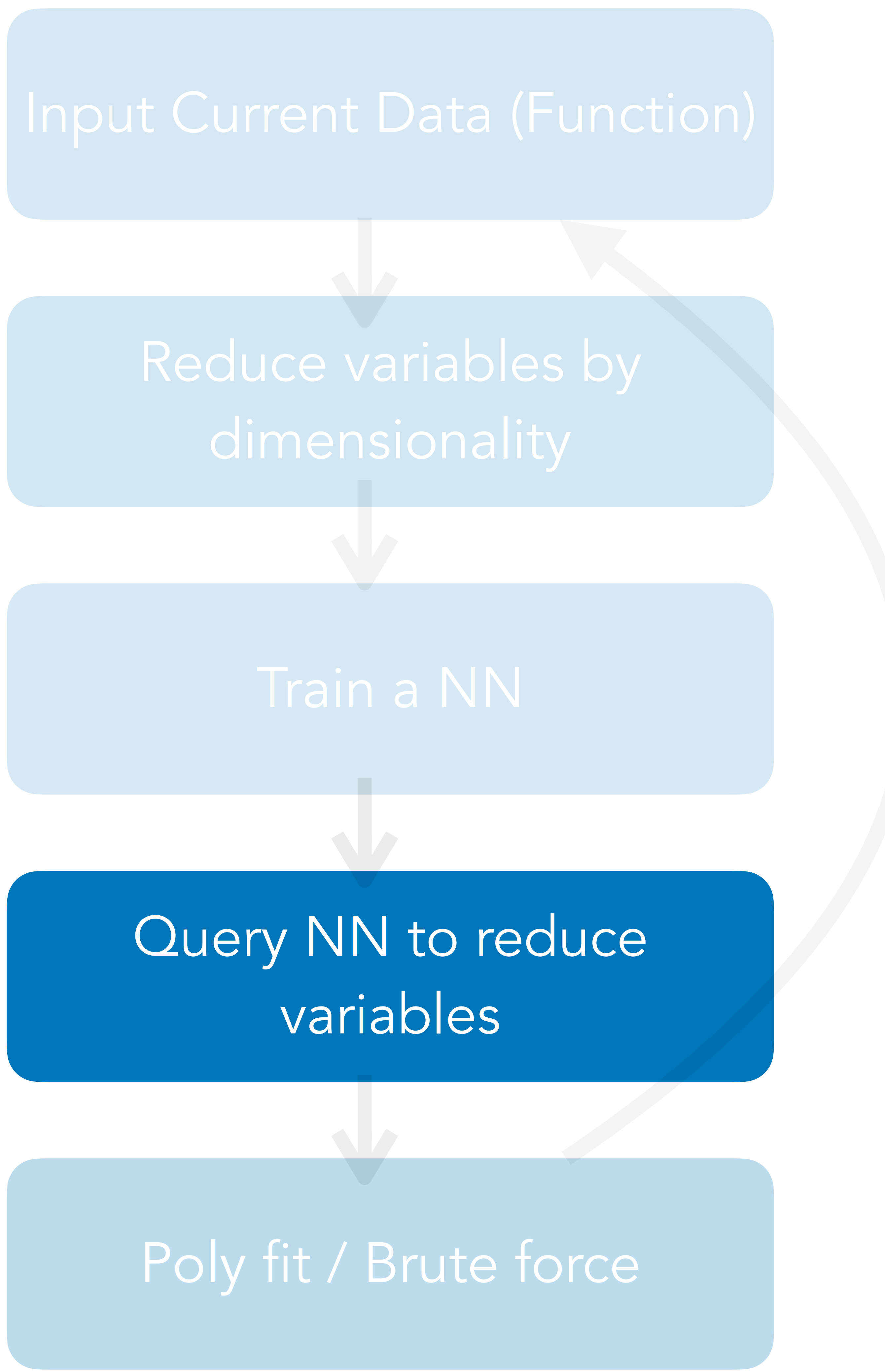
Symbolic Regression - Example

<u>Translational Symmetry</u>

Let $F(x_1, ..., x_n)$ be the function learned by the neural network. Then translational symmetry with respect to x_1 and x_2 corresponds to

$$F(x_1 + a, x_2 + a, x_3, ... x_n) - F(x_1, x_2, ..., x_n) \approx 0$$

If the difference is $< \epsilon_{\text{sym}}$, then apply $x_1 \mapsto x_2 - x_1$



Symbolic Regression - Example

Multiplicative Separability

f is multiplicatively separable in x_1, x_2 if f can be factored as

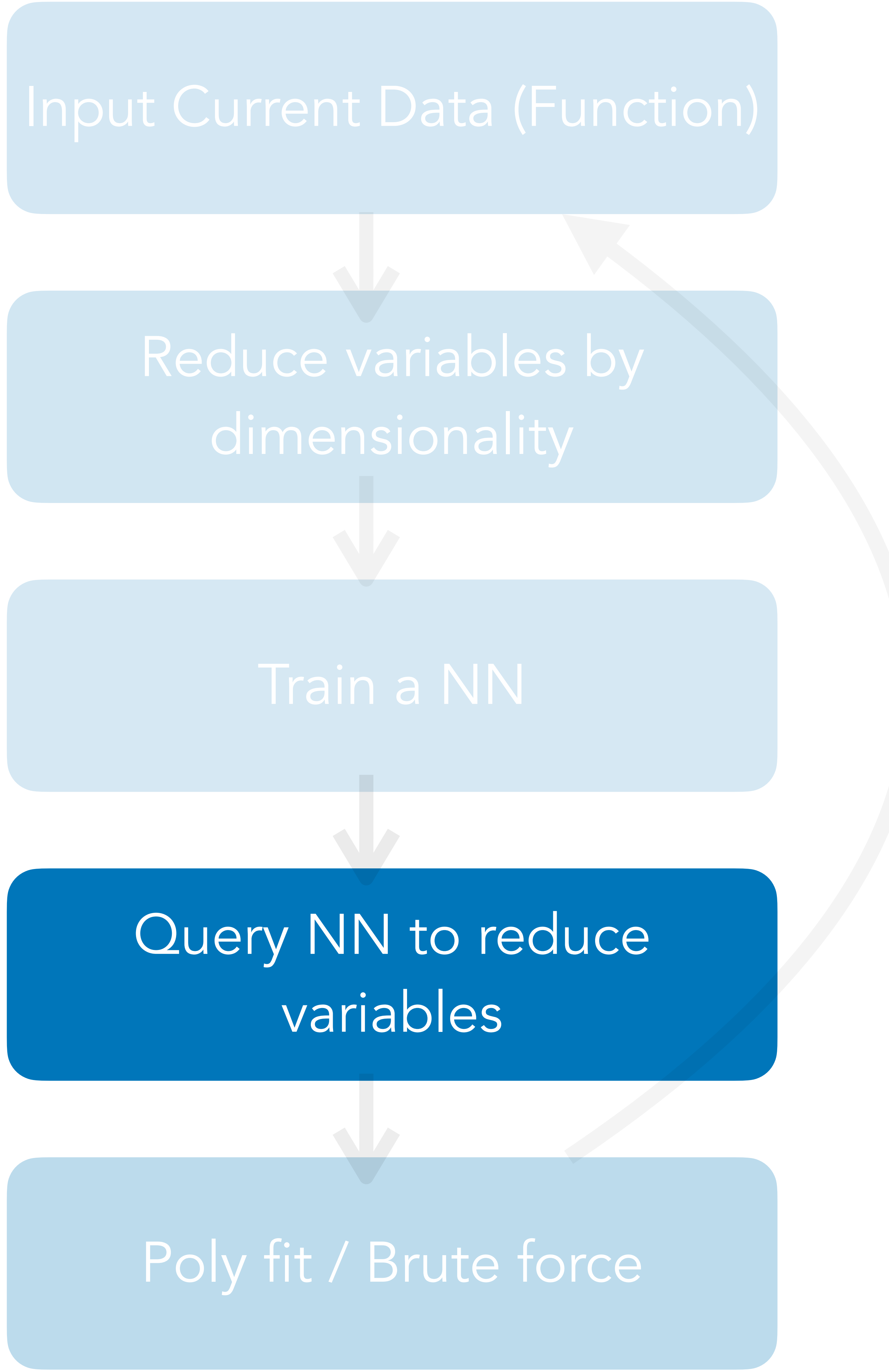
$$f(x_1, x_2) = g(x_1)h(x_2)$$

With no common variables.

To test this, pick constants c_1, c_2 and check if

$$f(x_1, x_2) = \frac{f(x_1, c_2) \cdot f(c_1, x_2)}{f(c_1, c_2)}$$

Up to ϵ_{sep}. If so, then separate the problem of learning f into two subproblems of learning g and h



Symbolic Regression - Example

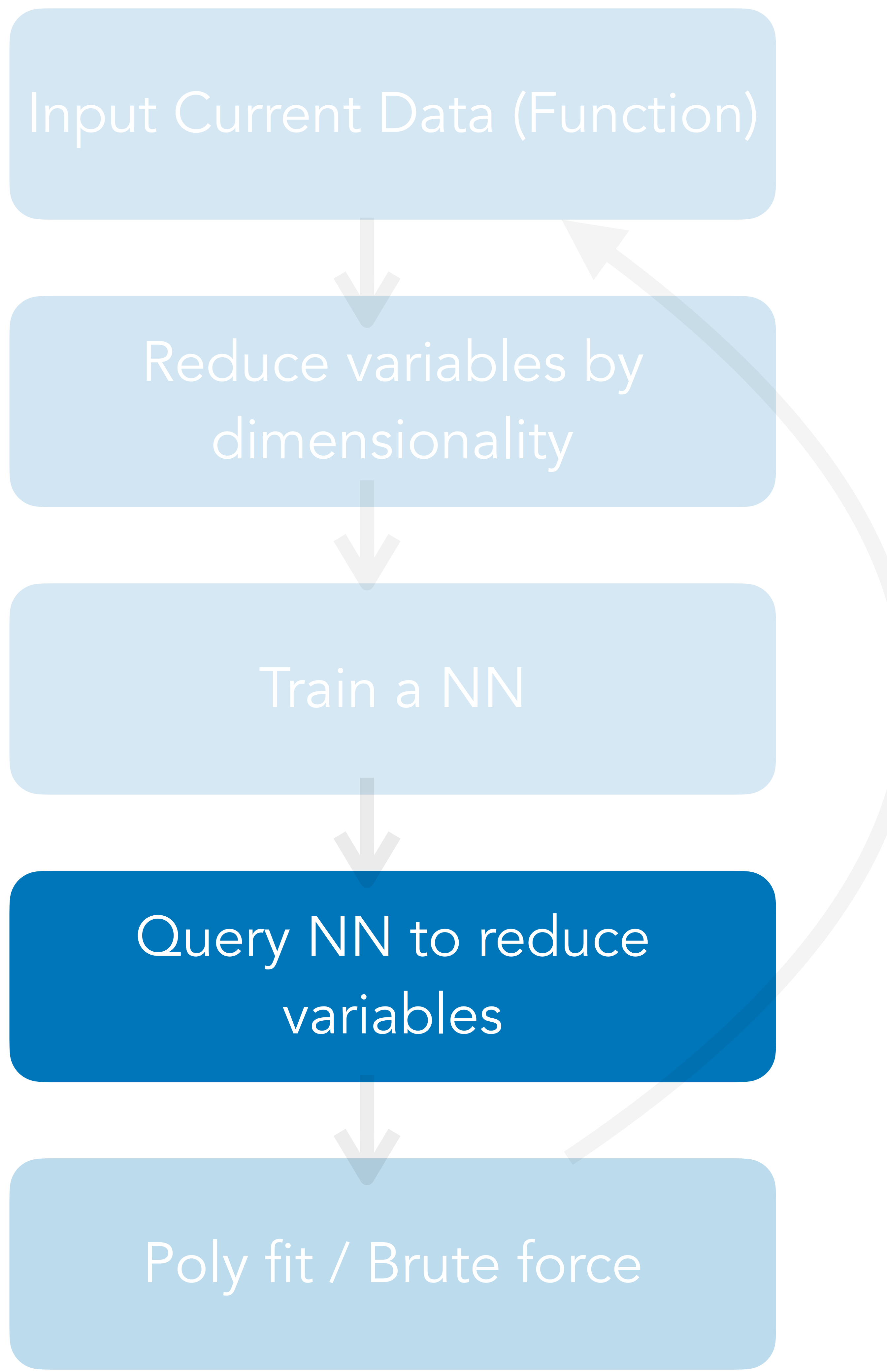
<u>Translational Symmetry</u>

<u>Multiplicative Separability</u>

<u>Rotational Symmetry</u>

<u>Scaling Symmetry</u>

<u>Additive Separability</u>



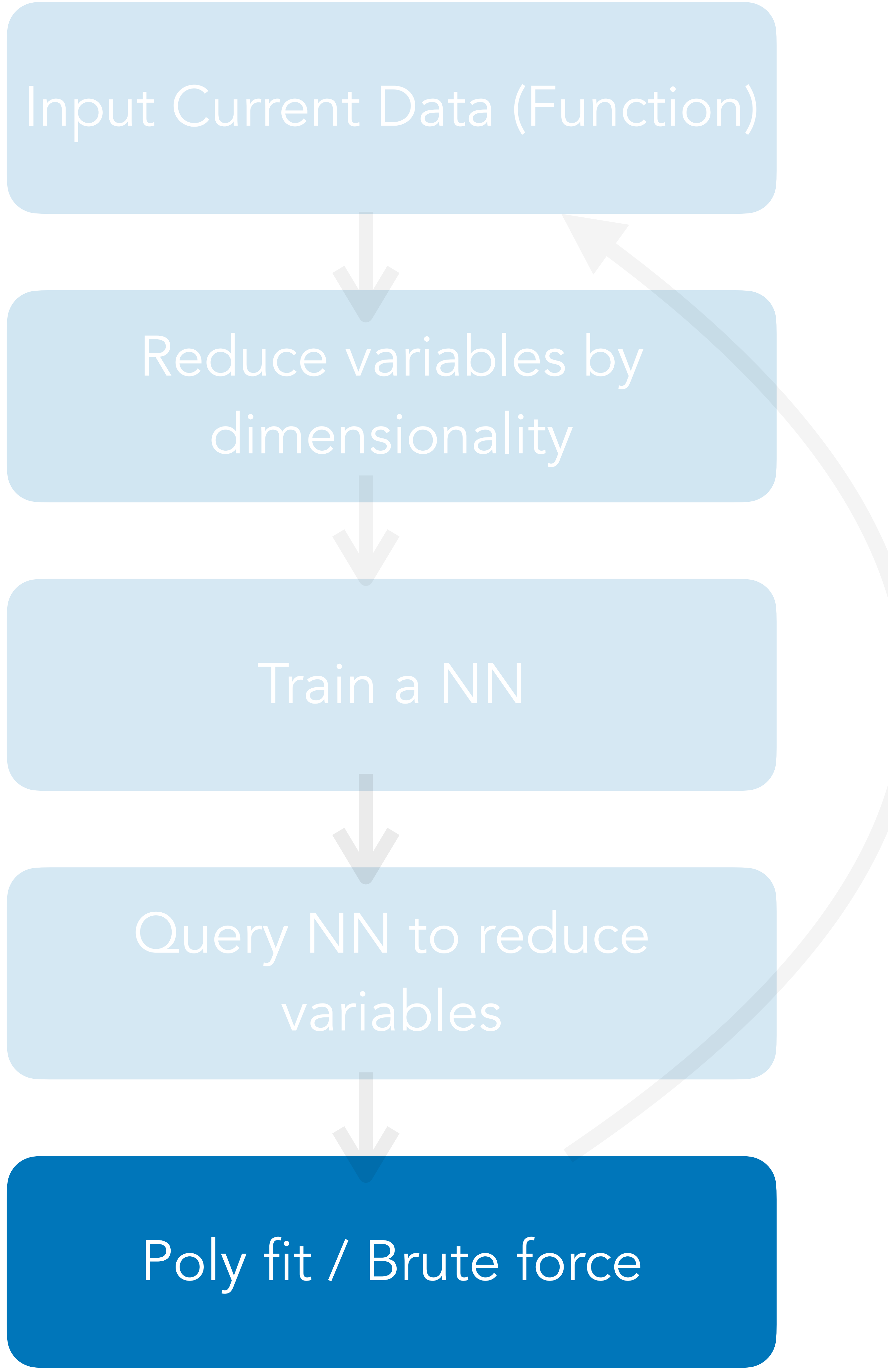
Symbolic Regression - Example

Test brute force / polynomial fit

a ✓

$$\frac{1}{(b-1)^2 + g^2 + h^2}$$

$$(b-1)^2 + g^2 + h^2$$ ✓



Symbolic Regression - Example

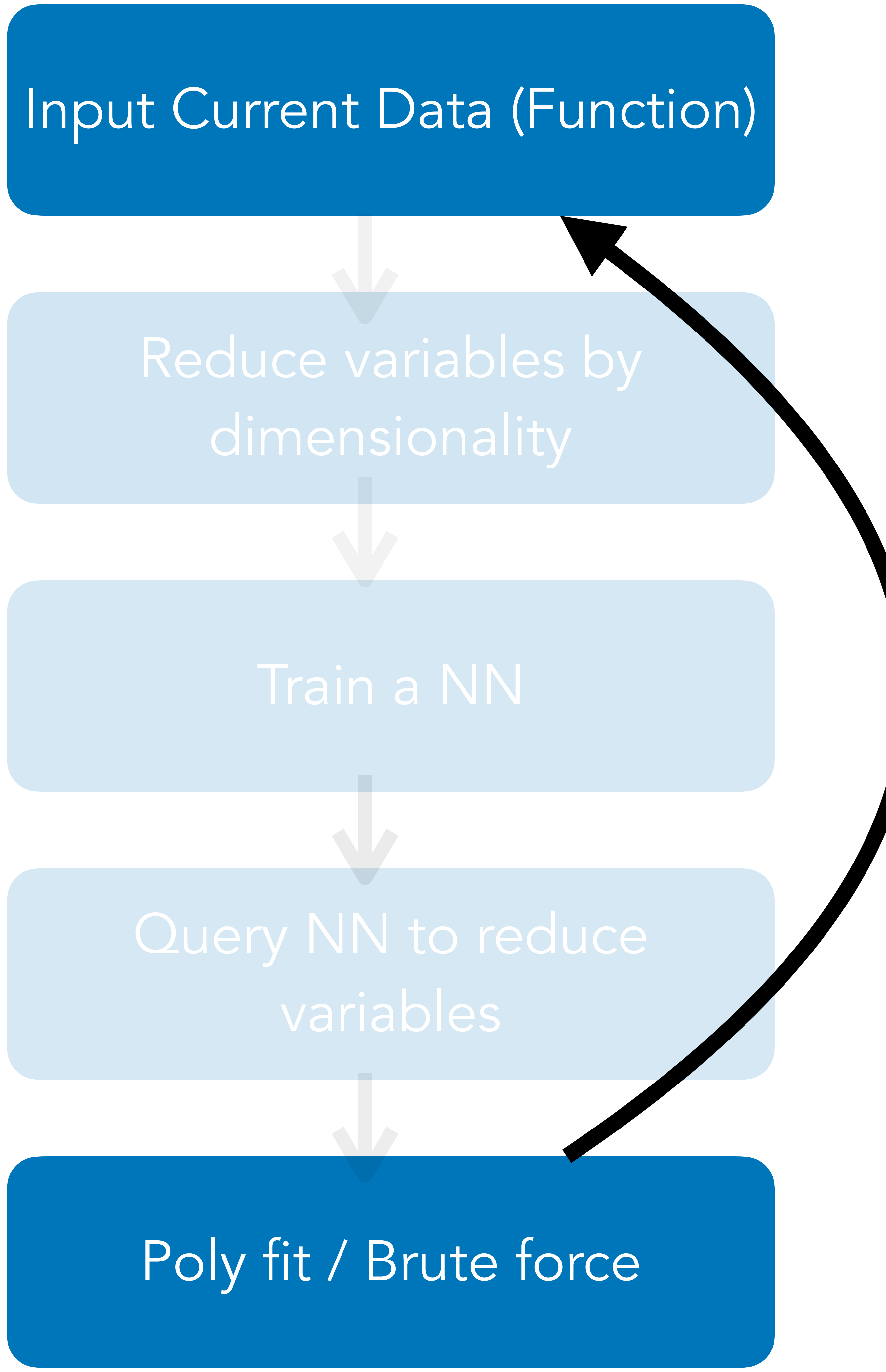
Test brute force / polynomial fit

a ✓

$$\frac{1}{(b-1)^2+g^2+h^2}$$

$$(b-1)^2+g^2+h^2$$ ✓

If this fails, then recursively take the smaller problems and rerun



Symbolic Regression - Example

This was tested on 100 equations from Feynman's lecture notes and 20 harder equations

Feynman eq.	Equation
I.6.20a	$f = e^{-\theta^2/2}/\sqrt{2\pi}$
I.6.20	$f = e^{-\frac{\theta^2}{2\sigma^2}}/\sqrt{2\pi\sigma^2}$
I.6.20b	$f = e^{-\frac{(\theta-\theta_1)^2}{2\sigma^2}}/\sqrt{2\pi\sigma^2}$
I.8.14	$d = \sqrt{(x_2-x_1)^2+(y_2-y_1)^2}$
I.9.18	$F = \frac{Gm_1m_2}{(x_2-x_1)^2+(y_2-y_1)^2+(z_2-z_1)^2}$
I.10.7	$m = \frac{m_0}{\sqrt{1-\frac{v^2}{c^2}}}$
I.11.19	$A = x_1y_1 + x_2y_2 + x_3y_3$
I.12.1	$F = \mu N_n$
I.12.2	$F = \frac{q_1q_2}{4\pi\epsilon r^2}$
I.12.4	$E_f = \frac{q_1}{4\pi\epsilon r^2}$
I.12.5	$F = q_2E_f$
I.12.11	$F = q(E_f + Bv\sin\theta)$
I.13.4	$K = \frac{1}{2}m(v^2+u^2+w^2)$
I.13.12	$U = Gm_1m_2(\frac{1}{r_2}-\frac{1}{r_1})$
I.14.3	$U = mgz$
I.14.4	$U = \frac{k_{spring}x^2}{2}$
I.15.3x	$x_1 = \frac{x-ut}{\sqrt{1-u^2/c^2}}$
I.15.3t	$t_1 = \frac{t-ux/c^2}{\sqrt{1-u^2/c^2}}$
I.15.10	$p = \frac{m_0v}{\sqrt{1-v^2/c^2}}$
I.16.6	$v_1 = \frac{u+v}{1+uv/c^2}$
I.18.4	$r = \frac{m_1r_1+m_2r_2}{m_1+m_2}$

Feynman eq.	Equation
II.2.42	$P = \frac{\kappa(T_2-T_1)A}{d}$
II.3.24	$F_E = \frac{P}{4\pi r^2}$
II.4.23	$V_e = \frac{q}{4\pi\epsilon r}$
II.6.11	$V_e = \frac{1}{4\pi\epsilon}\frac{p_d\cos\theta}{r^2}$
II.6.15a	$E_f = \frac{3}{4\pi\epsilon}\frac{p_dz}{r^5}\sqrt{x^2+y^2}$
II.6.15b	$E_f = \frac{3}{4\pi\epsilon}\frac{p_d}{r^3}\cos\theta\sin\theta$
II.8.7	$E = \frac{3}{5}\frac{q^2}{4\pi\epsilon d}$
II.8.31	$E_{den} = \frac{\epsilon E_f^2}{2}$
II.10.9	$E_f = \frac{\sigma_{den}}{\epsilon}\frac{1}{1+\chi}$
II.11.3	$x = \frac{qE_f}{m(\omega_0^2-\omega^2)}$
II.11.17	$n = n_0(1+\frac{p_dE_f\cos\theta}{k_bT})$
II.11.20	$P_* = \frac{n_\rho p_d^2E_f}{3k_bT}$
II.11.27	$P_* = \frac{n\alpha}{1-n\alpha/3}\epsilon E_f$
II.11.28	$\theta = 1 + \frac{n\alpha}{1-(n\alpha/3)}$
II.13.17	$B = \frac{1}{4\pi\epsilon c^2}\frac{2I}{r}$
II.13.23	$\rho_c = \frac{\rho_{c_0}}{\sqrt{1-v^2/c^2}}$
II.13.34	$j = \frac{\rho_{c_0}v}{\sqrt{1-v^2/c^2}}$
II.15.4	$E = -\mu_MB\cos\theta$
II.15.5	$E = -p_dE_f\cos\theta$
II.21.32	$V_e = \frac{q}{4\pi\epsilon r(1-v/c)}$
II.24.17	$k = \sqrt{\frac{\omega^2}{c^2}-\frac{\pi^2}{d^2}}$
II.27.16	$F_E = \epsilon cE_f^2$
II.27.18	$E_{den} = \epsilon E_f^2$
II.34.2a	$I = \frac{qv}{2\pi r}$
II.34.2	$\mu_M = \frac{qvr}{2}$
II.34.11	$\omega = \frac{g\_qB}{2m}$

Source	Equation
Rutherford Scattering	$A = \left(\frac{Z_1Z_2\alpha\hbar c}{4E\sin^2(\frac{\theta}{2})}\right)^2$
Friedman Equation	$H = \sqrt{\frac{8\pi G}{3}\rho - \frac{k_fc^2}{a_f^2}}$
Compton Scattering	$U = \frac{E}{1+\frac{E}{mc^2}(1-\cos\theta)}$
Radiated gravitational wave power	$P = -\frac{32}{5}\frac{G^4}{c^5}\frac{(m_1m_2)^2(m_1+m_2)}{r^5}$
Relativistic aberration	$\theta_1 = \arccos\left(\frac{\cos\theta_2-\frac{v}{c}}{1-\frac{v}{c}\cos\theta_2}\right)$
N-slit diffraction	$I = I_0\left[\frac{\sin(\alpha/2)}{\alpha/2}\frac{\sin(N\delta/2)}{\sin(\delta/2)}\right]^2$
Goldstein 3.16	$v = \sqrt{\frac{2}{m}(E-U-\frac{L^2}{2mr^2})}$
Goldstein 3.55	$k = \frac{mk_G}{L^2}(1+\sqrt{1+\frac{2EL^2}{mk_G^2}}\cos(\theta_1-\theta_2))$
Goldstein 3.64 (ellipse)	$r = \frac{d(1-\alpha^2)}{1+\alpha\cos(\theta_1-\theta_2)}$
Goldstein 3.74 (Kepler)	$t = \frac{2\pi d^{3/2}}{\sqrt{G(m_1+m_2)}}$
Goldstein 3.99	$\alpha = \sqrt{1+\frac{2\epsilon^2EL^2}{m(Z_1Z_2q^2)^2}}$
Goldstein 8.56	$E = \sqrt{(p-qA_{vec})^2c^2+m^2c^4}+qV_e$
Goldstein 12.80	$E = \frac{1}{2m}[p^2+m^2\omega^2x^2(1+\alpha\frac{x}{y})]$
Jackson 2.11	$F = \frac{q}{4\pi\epsilon y^2}\left[4\pi\epsilon V_ed - \frac{qdy^3}{(y^2-d^2)^2}\right]$

Symbolic Regression - Example

This was tested on 100 equations from Feynman's lecture notes and 20 harder equations

The success rate was better than existing state of the art:

Equation Set	Eureqa (Benchmark)	AI Feynman (with DA)	AI Feynman (without DA)
100 Feynman Lecture Equations	**68%**	**100%**	**93%**
20 "Bonus" Equations	**15%**	**90%**	?

Symbolic Regression - Example

Couple of interesting notes:

1. Failure cases. For example, the Radiational Gravitational Waves equation:

$$P = -\frac{32}{5}\frac{G^4}{c^5}\frac{(m_1 m_2)^2(m_1 + m_2)}{r^5} \longrightarrow y = -\frac{32a^2(1+a)}{5b^5}$$

In reverse polish notation, this is the string

$$aaa > * * bbbbb * * * * /$$

Which would take too long to solve. This is separable, but the 5th power in the denominator cause a wide dynamic range and so it was not detected

Symbolic Regression - Example

Couple of interesting notes:

2. Noise: Performance was consistent up to Gaussian noise levels of $\epsilon = 10^{-4}$ but dropped by 50% with noise around $\epsilon = 10^{-2}$.

The thresholds were not adjusted to each problem so the tolerance can probably be improved.

Symbolic Regression - Example

Couple of interesting notes:

3. Constants: Constants were treated as their own variables and symbols in the data. It's not clear otherwise how we would discover relations such as $G\sqrt{\pi}$, $\frac{G^4}{c^5}$, etc.

But how do we learn these specific relationships with real data?

Symbolic Regression

Problems:

1. Sensitivity to noise
2. Constant Relationships are difficult to detect
3. Require lots of data to train a neural network

Incorporate physics knowledge into the search

Approach 2: Symbolic Regression with background knowledge

Approach 2: Symbolic Regression with background knowledge

In particular, we will very briefly look at the approach in this paper:

nature communications

Article https://doi.org/10.1038/s41467-023-37236-y

Combining data and theory for derivable scientific discovery with AI-Descartes

Received: 28 October 2021

Accepted: 8 March 2023

Published online: 12 April 2023

Check for updates

Cristina Cornelio[1,2], Sanjeeb Dash[1], Vernon Austel[1], Tyler R. Josephson[3,4], Joao Goncalves[1], Kenneth L. Clarkson[1], Nimrod Megiddo[1], Bachir El Khadir[1] & Lior Horesh[1,5]

Scientists aim to discover meaningful formulae that accurately describe experimental data. Mathematical models of natural phenomena can be manually created from domain knowledge and fitted to data, or, in contrast, created automatically from large datasets with machine-learning algorithms. The problem of incorporating prior knowledge expressed as constraints on the functional form of a learned model has been studied before, while finding models that are consistent with prior knowledge expressed via general logical axioms is an open problem. We develop a method to enable principled derivations of models of natural phenomena from axiomatic knowledge and experimental data by combining logical reasoning with symbolic regression. We demonstrate these concepts for Kepler's third law of planetary motion, Einstein's relativistic time-dilation law, and Langmuir's theory of adsorption. We show we can discover governing laws from few data points when logical reasoning is used to distinguish between candidate formulae having similar error on the data.

Symbolic Regression with Background Knowledge

Key Idea:

Use symbolic regression methods to generate lots of candidates for the formula

→

Given some background theory B, use an Automated Theorem Prover (ATP) to rank and prove these candidates. Select the

Symbolic Regression with Background Knowledge

Main Contributions:

1. Incorporating a derivability module into the search.

Candidate Formula → **Derivability - Prove the Formula** → **Existential Derivability**

$$f = \frac{p}{1.507x_1 + 0.302x_2}$$

→

$$B \rightarrow \frac{p}{1.507x_1 + 0.302x_2}$$

→

$$\exists c_1, c_2 \text{ s.t. } B \rightarrow \frac{p}{c_1x_1 + c_2x_2}$$

Confirm Functional Form → Run your favorite method of fitting parameters

Symbolic Regression with Background Knowledge

Main Contributions:

2. If not derivable, then have the theorem proved derive bounds on the error between candidate function and functions derivable from the background theory

Point wise reasoning error

$$\beta_2^r = \sqrt{\sum_{i=1}^{m} \frac{f(x_i) - f_{\mathscr{B}}(x_i)}{f_{\mathscr{B}}(x_i)}}$$

Generalized reasoning error

$$\beta_\infty^r = \sqrt{\sum_{i=1}^{m} \frac{f(x_i) - f_{\mathscr{B}}(x_i)}{f_{\mathscr{B}}(x_i)}}$$

Symbolic Regression with Background Knowledge

Results: Ran this on more problems from the Feynman notes as well as supplementary problems.

Label	Formula	AI-Descartes	AI Feynman	PySR	BMS
II.11.20	$n_\rho p_d^2 E_f/(3k_b T)$	✓[1]	X	X	✓[2]
II.11.27	$\frac{n\alpha}{1-n\alpha/3}\varepsilon E_f$	✓[1]	X	X	X
II.11.28	$1+\frac{n\alpha}{1-n\alpha/3}$	✓[1]	X	X	X
II.13.17	$\frac{1}{4\pi\varepsilon c^2}\frac{2I}{r}$	✓[1]	X	✓[1]	X
II.13.23	$\frac{\rho_{c_0}}{\sqrt{1-v^2/c^2}}$	✓[1]	X	X	X
II.13.34	$\frac{\rho_{c_0} v}{\sqrt{1-v^2/c^2}}$	X	X	X	X
II.21.32	$\frac{q}{4\pi\varepsilon r(1-v/c)}$	X	X	X	X
II.24.17	$\sqrt{\frac{\omega^2}{c^2}-\frac{\pi^2}{d^2}}$	X	X	X	X
II.27.16	$\varepsilon c E_f^2$	✓	X	✓	✓
II.27.18	εE_f^2	✓[1]	✓[1]	✓	✓
II.34.2a	$qv/(2\pi r)$	✓[1]	✓[1]	✓[1]	✓[1]
II.34.2	$qvr/2$	✓[1]	✓[1]	✓[1]	✓[1]
II.34.11	$gqB/(2m)$	✓	✓[1]	✓[1]	✓[1]
II.34.29a	$qh/(4\pi m)$	✓[1]	✓[1]	✓[1]	✓[1]
II.34.29b	$g\mu_M B J_z/h$	✓[2]	X	✓[1]	✓[1]
II.35.18	$\frac{n_0}{e^{\frac{mom\times B}{kb\times T}}+e^{\frac{-mom\times B}{(kb\times T)}}}$	X	X	X	X
II.36.38	$\frac{\mu_m B}{k_b T}+\frac{\mu_m \alpha M}{\varepsilon c^2 k_b T}$	X	X	X	X
II.37.1	$\mu_M(1+\chi)B$	✓[1]	X	✓[1]	✓[2]
II.38.3	YAx/d	✓[1]	✓[2]	✓	✓[2]
II.38.14	$\frac{Y}{2(1+\sigma)}$	✓[1]	✓[2]	X	✓[2]
	Accuracy	**86.67%**	80%	73.33%	80%

Symbolic Regression with Background Knowledge

Some Comments:

Failure Cases:

Most of the failure cases came from when the theorem prover could not prove derivability or good bounds on the error.

Approach 3: Polynomial Optimization

Approach 3: Polynomial Optimization

In particular, we will look at the approach in this paper:

nature communications

Article https://doi.org/10.1038/s41467-024-50074-w

Evolving scientific discovery by unifying data and background knowledge with AI Hilbert

Received: 23 September 2023

Accepted: 27 June 2024

Published online: 14 July 2024

Check for updates

Ryan Cory-Wright [1], Cristina Cornelio [2], Sanjeeb Dash[3], Bachir El Khadir[3] & Lior Horesh [3]

The discovery of scientific formulae that parsimoniously explain natural phenomena and align with existing background theory is a key goal in science. Historically, scientists have derived natural laws by manipulating equations based on existing knowledge, forming new equations, and verifying them experimentally. However, this does not include experimental data within the discovery process, which may be inefficient. We propose a solution to this problem when all axioms and scientific laws are expressible as polynomials and argue our approach is widely applicable. We model notions of minimal complexity using binary variables and logical constraints, solve polynomial optimization problems via mixed-integer linear or semidefinite optimization, and prove the validity of our scientific discoveries in a principled manner using Positivstellensatz certificates. We demonstrate that some famous scientific laws, including Kepler's Law of Planetary Motion and the Radiated Gravitational Wave Power equation, can be derived in a principled manner from axioms and experimental data.

Polynomial Optimization

Key Idea: There are a lot of laws in science that can be transformed to polynomial expressions.

Examples:

Kepler's Third Law: $p = \sqrt{\dfrac{4\pi^2(d_1+d_2)^3}{G(m_1+m_2)}} \mapsto p^2G(m_1+m_2) - 4\pi^2(d_1+d_2)^3$

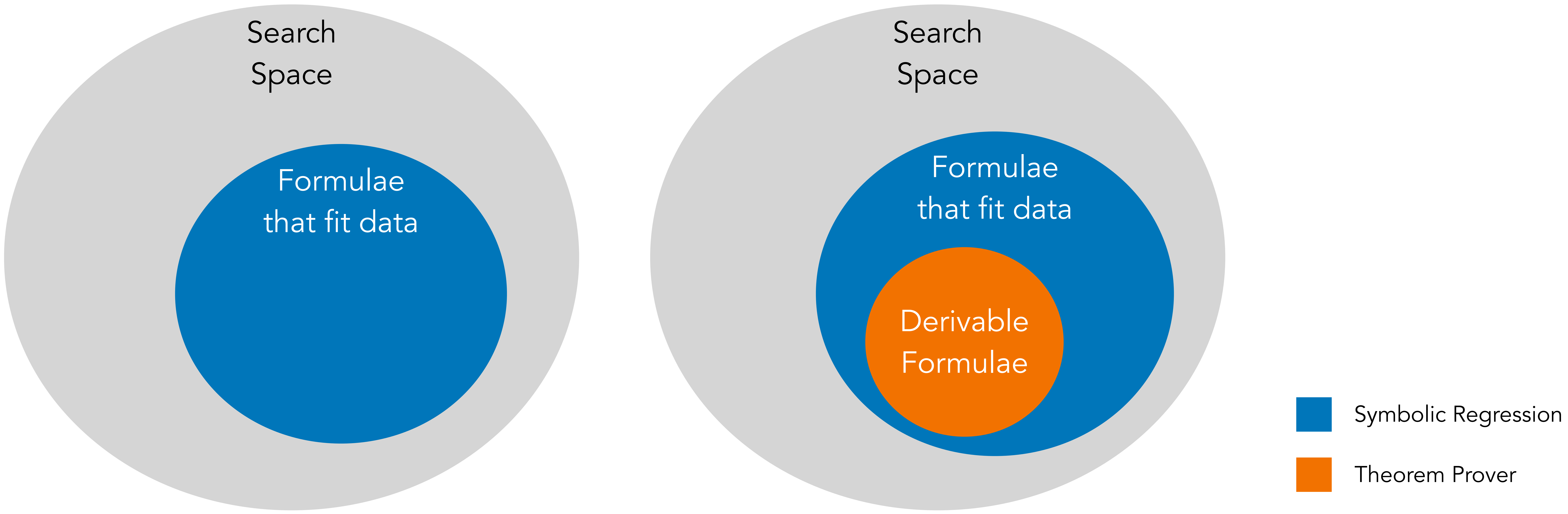
Compton Scattering Equation: $\lambda_2 - \lambda_1 = \dfrac{h}{m_e c}(1-\cos\theta) \mapsto (\lambda_2-\lambda_1)hm_ec^3(1+\cos\theta)$

Note: We will model things after polynomials and polynomial inequalities. We will not deal with diffeqs and trig inequalities directly, but we can get some mileage as we will see.

Polynomial Optimization

Key Idea: There are a lot of laws in science that can be transformed to polynomial expressions.

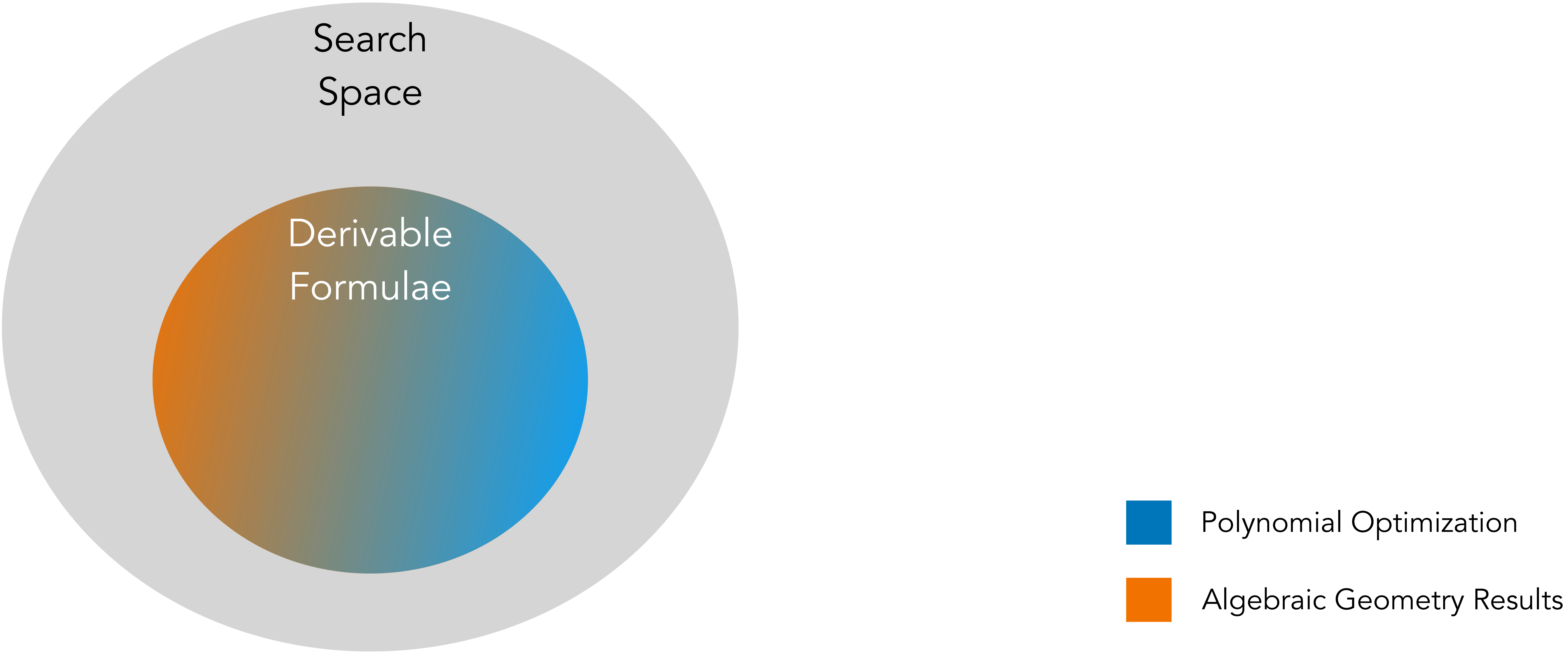
Previous methods:



Polynomial Optimization

Key Idea: There are a lot of laws in science that can be transformed to polynomial expressions.

This work:



Polynomial Optimization

The idea is to solve the following optimization problem

$$\begin{aligned}
\min_{q \in \mathbb{R}_{n,2d}} \quad & \sum_{\bar{\boldsymbol{x}}_i \in \mathcal{D}} |q(\bar{\boldsymbol{x}}_i)| + \lambda \cdot d^c(q, \mathcal{G} \cap \mathcal{H}) \\
\text{s.t.} \quad & \sum_{\boldsymbol{\mu} \in \Omega : \boldsymbol{\mu}_1 \geq 1} a_{\boldsymbol{\mu}} = 1, \\
& a_{\boldsymbol{\mu}} = 0 \;\; \forall \boldsymbol{\mu} \in \Omega : \sum_{j=t+1}^{n} \boldsymbol{\mu}_j \geq 1,
\end{aligned}$$

Polynomial Optimization

The idea is to solve the following optimization problem

$$
\begin{aligned}
\min_{q \in \mathbb{R}_{n,2d}} \quad & \sum_{\bar{\boldsymbol{x}}_i \in \mathcal{D}} |q(\bar{\boldsymbol{x}}_i)| + \lambda \cdot d^c(q, \mathcal{G} \cap \mathcal{H}) \\
\text{s.t.} \quad & \sum_{\boldsymbol{\mu} \in \Omega : \boldsymbol{\mu}_1 \geq 1} a_{\boldsymbol{\mu}} = 1, \\
& a_{\boldsymbol{\mu}} = 0 \;\; \forall \boldsymbol{\mu} \in \Omega : \sum_{j=t+1}^{n} \boldsymbol{\mu}_j \geq 1,
\end{aligned}
$$

Fidelity to data

This can be defined with l_2 or l_∞ loss

Polynomial Optimization

The idea is to solve the following optimization problem

$$\begin{aligned}\min_{q \in \mathbb{R}_{n,2d}} \quad & \sum_{\bar{\boldsymbol{x}}_i \in \mathcal{D}} |q(\bar{\boldsymbol{x}}_i)| + \lambda \cdot d^c(q, \mathcal{G} \cap \mathcal{H}) \\ \text{s.t.} \quad & \sum_{\boldsymbol{\mu} \in \Omega : \boldsymbol{\mu}_1 \geq 1} a_{\boldsymbol{\mu}} = 1, \\ & a_{\boldsymbol{\mu}} = 0 \;\; \forall \boldsymbol{\mu} \in \Omega : \sum_{j=t+1}^{n} \boldsymbol{\mu}_j \geq 1,\end{aligned}$$

Distance between the discovered q and the background theory $\mathscr{G} \cap \mathscr{H}$

Polynomial Optimization

The idea is to solve the following optimization problem

$$
\begin{aligned}
\min_{q \in \mathbb{R}_{n,2d}} \quad & \sum_{\bar{\boldsymbol{x}}_i \in \mathcal{D}} |q(\bar{\boldsymbol{x}}_i)| + \lambda \cdot d^c(q, \mathcal{G} \cap \mathcal{H}) \\
\text{s.t.} \quad & \sum_{\boldsymbol{\mu} \in \Omega : \boldsymbol{\mu}_1 \geq 1} a_{\boldsymbol{\mu}} = 1, \\
& a_{\boldsymbol{\mu}} = 0 \ \ \forall \boldsymbol{\mu} \in \Omega : \sum_{j=t+1}^{n} \boldsymbol{\mu}_j \geq 1,
\end{aligned}
$$

Want a polynomial in x_1

Polynomial Optimization

The idea is to solve the following optimization problem

$$\min_{q \in \mathbb{R}_{n,2d}} \sum_{\bar{\boldsymbol{x}}_i \in \mathcal{D}} |q(\bar{\boldsymbol{x}}_i)| + \lambda \cdot d^c(q, \mathcal{G} \cap \mathcal{H})$$

$$\text{s.t.} \quad \sum_{\boldsymbol{\mu} \in \Omega : \boldsymbol{\mu}_1 \geq 1} a_{\boldsymbol{\mu}} = 1,$$

$$a_{\boldsymbol{\mu}} = 0 \quad \forall \boldsymbol{\mu} \in \Omega : \sum_{j=t+1}^{n} \boldsymbol{\mu}_j \geq 1,$$

Bound on the
complexity of the expression

Polynomial Optimization - Distance to Background

Given some axioms $h_1(\mathbf{x}), \ldots, h_l(\mathbf{x}) = 0$, inequalities $g_1(\mathbf{x}) \geq 0$,....,$g_k(\mathbf{x}) \geq 0$, where $g_i, h_j \in \mathbb{R}[x_1, \ldots, x_n]$ and data $\mathcal{D}$, define the sets

$$\mathcal{G} = \{x \in \mathbb{R}^n \,|\, g_i(x) = 0\} \text{ and } \mathcal{H} = \{x \in \mathbb{R}^n \,|\, h_j(x) \geq 0\}$$

We want to find a polynomial f such that

$$x \in \mathcal{G} \cap \mathcal{H} \implies f(x) \geq 0$$

Theorem: Putinar's Positive Stellensatz

If g_i, h_k satisfy the Archimedean property, i.e. there exists R and sum of squares polynomials $\alpha_0, \ldots, \alpha_l$ such that $R - \sum_{i=1}^{n} x_i^2 = \alpha_0 + \sum_{i=1}^{k} \alpha_i g_i$, then for any degree d polynomial f, the implication above holds if and only if there exist sum of squares polynomials $\alpha_0, \ldots, \alpha_k$ and real polynomials $\beta_1, \ldots, \beta_l$ such that

$$f = \alpha_0 + \sum_{i=1}^{k} \alpha_i g_i + \sum_{j=1}^{l} \beta_j h_j$$

Polynomial Optimization - Distance to Background

Given some axioms $h_1(\mathbf{x}), \ldots, h_l(\mathbf{x}) = 0$, inequalities $g_1(\mathbf{x}) \geq 0$,....,$g_k(\mathbf{x}) \geq 0$, where $g_i, h_j \in \mathbb{R}[x_1, \ldots, x_n]$ and data $\mathcal{D}$, define the sets

$$\mathcal{G} = \{x \in \mathbb{R}^n \,|\, g_i(x) = 0\} \text{ and } \mathcal{H} = \{x \in \mathbb{R}^n \,|\, h_j(x) \geq 0\}$$

We want to find a polynomial f such that

$$x \in \mathcal{G} \cap \mathcal{H} \implies f(x) \geq 0$$

Case 1: Incomplete background knowledge

$$d(q, \mathcal{G} \cap \mathcal{H}) = \min_{\alpha_i \in \Sigma_{n,2d}, \beta_j \in \mathbb{R}_n, d} \|\text{Coeff } (q - \alpha_0 - \sum \alpha_i g_i - \sum \beta_j h_j)\|_2$$

Polynomial Optimization - Distance to Background

Given some axioms $h_1(\mathbf{x}), \ldots, h_l(\mathbf{x}) = 0$, inequalities $g_1(\mathbf{x}) \geq 0$,....,$g_k(\mathbf{x}) \geq 0$, where $g_i, h_j \in \mathbb{R}[x_1, \ldots, x_n]$ and data $\mathscr{D}$, define the sets

$$\mathscr{G} = \{x \in \mathbb{R}^n \,|\, g_i(x) = 0\} \text{ and } \mathscr{H} = \{x \in \mathbb{R}^n \,|\, h_j(x) \geq 0\}$$

We want to find a polynomial f such that

$$x \in \mathscr{G} \cap \mathscr{H} \implies f(x) \geq 0$$

Case 2: Inconsistent background knowledge

$$d(q, \mathscr{G} \cap \mathscr{H}) = \min_{\alpha_i \in \Sigma_{n,2d}, \beta_j \in \mathbb{R}_n, d} \|\text{Coeff } (q - \alpha_0 - \sum \alpha_i g_i - \sum \beta_j h_j)\|_2$$

$$\text{s.t. } \alpha_i = 0 \text{ if } z_i = 0,\ z_i \in \{0,1\}$$
$$\beta_j = 0 \text{ if } y_j = 0, y_j \in \{0,1\}$$
$$\sum z_i + \sum y_j \leq \tau$$

Polynomial Optimization - Example

Let's say we want to discover Kepler's third law

$$p = \sqrt{\frac{4(d_1 + d_2)^3}{G(m_1 + m_2)}}$$

We have data $\{(d_1, d_2, m_1, m_2, p)_i\}$ as well as some knowledge of physics

$$\begin{aligned} d_1 m_1 - d_2 m_2 &= 0, \\ (d_1 + d_2)^2 F_g - G m_1 m_2 &= 0, \\ F_c - m_2 d_2 w^2 &= 0, \\ F_c - F_g &= 0, \\ wp &= 1, \end{aligned}$$

Then running solving the optimization problem on a solver gives

$$m_1 m_2 G p^2 - m_1 d_1 d_2^2 - m_2 d_1^2 d_2 - 2 m_2 d_1 d_2^2 = 0$$

Which is the correct expression post-factoring

Polynomial Optimization - Example

Then running solving the optimization problem on a solver gives

$$m_1 m_2 G p^2 - m_1 d_1 d_2^2 - m_2 d_1^2 d_2 - 2 m_2 d_1 d_2^2 = 0$$

Which is the correct expression post-factoring. What's more is that we also get the corresponding α_i:

$$\begin{aligned}
-d_2^2 p^2 w^2, \\
-p^2, \\
d_1^2 p^2 + 2 d_1 d_2 p^2 + d_2^2 p^2, \\
d_1^2 p^2 + 2 d_1 d_2 p^2 + d_2^2 p^2, \\
m_1 d_1 d_2^2 p w + m_2 d_1^2 d_2 p w + 2 m_2 d_1 d_2^2 p w + m_1 d_1 d_2^2 + m_2 d_1^2 d_2 + 2 m_2 d_1 d_2^2,
\end{aligned}$$

Polynomial Optimization - Example

Then running solving the optimization problem on a solver gives

$$m_1 m_2 G p^2 - m_1 d_1 d_2^2 - m_2 d_1^2 d_2 - 2 m_2 d_1 d_2^2 = 0$$

Which is the correct expression post-factoring. What's more is that we also get the corresponding α_i:

$$\begin{aligned}
-d_2^2 p^2 w^2, \\
-p^2, \\
d_1^2 p^2 + 2 d_1 d_2 p^2 + d_2^2 p^2, \\
d_1^2 p^2 + 2 d_1 d_2 p^2 + d_2^2 p^2, \\
m_1 d_1 d_2^2 p w + m_2 d_1^2 d_2 p w + 2 m_2 d_1 d_2^2 p w + m_1 d_1 d_2^2 + m_2 d_1^2 d_2 + 2 m_2 d_1 d_2^2,
\end{aligned}$$

Polynomial Optimization - Example

Some other examples that were tested:

Radiational Gravitational Wave Power: $P = -\frac{32G^4}{5c^5r^5}(m_1m_2)^2(m_1+m_2)$ with background

$$\omega^2 r^3 - G(m_1+m_2) = 0,$$

$$5(m_1+m_2)^2c^5P + G\mathrm{tr}\left(\frac{d^3}{dt^3}\left(m_1m_2r^2\begin{pmatrix} x^2-\frac{1}{3} & xy & 0 \\ xy & y^2-\frac{1}{3} & 0 \\ 0 & 0 & -\frac{1}{3}\end{pmatrix}\right)\right)^2 = 0,$$

$$x^2+y^2 = 1,$$

Hagen-Poiseulle Equation: $u(r) = \frac{-\Delta p}{4L\mu}(r^2 - R^2)$

$$u = c_0 + c_2r^2,$$

$$\mu\frac{\partial}{\partial r}(r\frac{\partial}{\partial r}u) - r\frac{dp}{dx} = 0,$$

$$c_0 + c_2R^2 = 0,$$

$$L\frac{dp}{dx} = -\Delta p,$$

Polynomial Optimization - Results

Data		AI-Hilbert	AI-Descartes [13]	AI-Feynman [42]	PySR [15]	BMS [24]
S	Hagen Poiseuille	✓	✗	✗	✗	✗
S	Gravitational Wave Power	✓	✗	✗	✗	✗
R	Relat. Time Dilat.	✓	✗	✗	✗	✗
R	Kepler's 3 Law	✓	✓*	✗	✓	✗
S	Bell inequalities†	✓	✗	✗	✗	✗
S	[3.1] I.15.10 FSRD	✓	✓*	✗	✗	✗
S	[3.1] I.27.6 FSRD	✓▷	✓*	✓*	✓	✗
S	[3.1] I.34.8 FSRD	✓▷	✓	✓*	✓	✓*
S	[3.1] I.43.16 FSRD	✓	✓*	✓*	✓	✓*
S	[3.1] II.10.9 FSRD	✓	✓*	✓*	✓*	✓*
S	[3.1] II.34.2 FSRD	✓	✓*	✓*	✓*	✓*
S	[3.2] Inelastic Relativ. Collision	✓	✓*	✗	✗	✗
S	[3.3] Decay of Pion⋄	✓	✓*	✗	✗	✓*
S	[3.4] Radiation Damping	✓	✗	✗	✓*	✓*
S	[3.5] Escape Velocity	✓	✓	✓*	✓*	✓*
S	[3.6] Hall Effect	✓	✓*	✗	✓*	✓*
S	[3.7] Compton Scattering	✓	✓*	✓*	✓*	✓*

Notation used in table:

✓ denotes the successful recovery of a scientific law.

✗ denotes the failure to recover a scientific law.

✓* denotes recovery up to constants, but not exact recovery.

✓▷ denotes successful recovery when some variables that are potentially not observable are still declared to be observable.

Polynomial Optimization

Some thoughts:

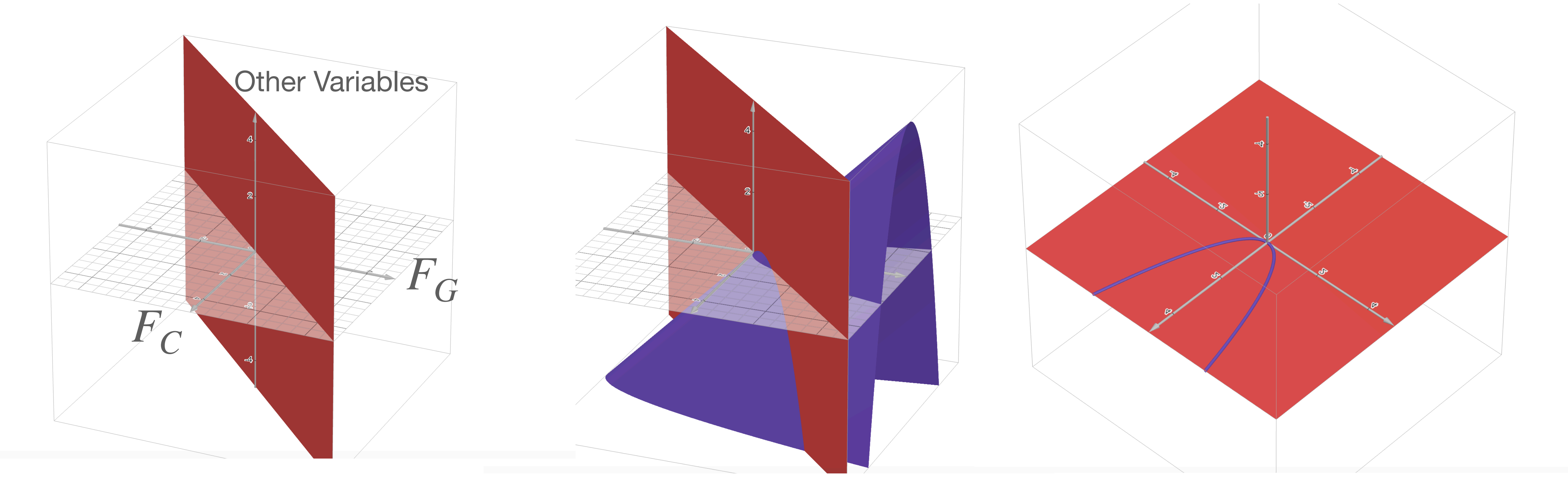
1. Solving the LP can be slow. In the worst case (for gravitational waves)

Ongoing work

Projecting varieties

Aim: Discover Kepler's Third Law using just background theory

$$\begin{aligned} d_1m_1 - d_2m_2 &= 0, \\ (d_1+d_2)^2F_g - Gm_1m_2 &= 0, \\ F_c - m_2d_2w^2 &= 0, \\ F_c - F_g &= 0, \\ wp &= 1, \end{aligned} \implies p = \sqrt{\frac{4\pi^2(d_1+d_2)^3}{G(m_1+m_2)}},$$



$$\pi : V(f_1, f_2, f_3, f_4, f_5) \subseteq \mathbb{R}^8 \longrightarrow \bar{V} \subseteq \mathbb{R}^5$$

Algebraically: Eliminating Variables

Encode axioms as an ideal: $I = \langle f_1, \ldots, f_5 \rangle \subseteq \mathbb{R}[m_1, d_1, m_2, d_2, F_c, F_g, w, p]$

Compute the intersection $I \cap \mathbb{R}[m_1, d_1, m_2, d_2, p]$

This can be done using Groebner bases:

Encode axioms as an ideal: $G = \langle g_1, \ldots, g_k \rangle = I$

Compute the intersection $G \cap \mathbb{R}[m_1, d_1, m_2, d_2, p]$

Abductive inference

We know we can recover a formula from a complete axiom system

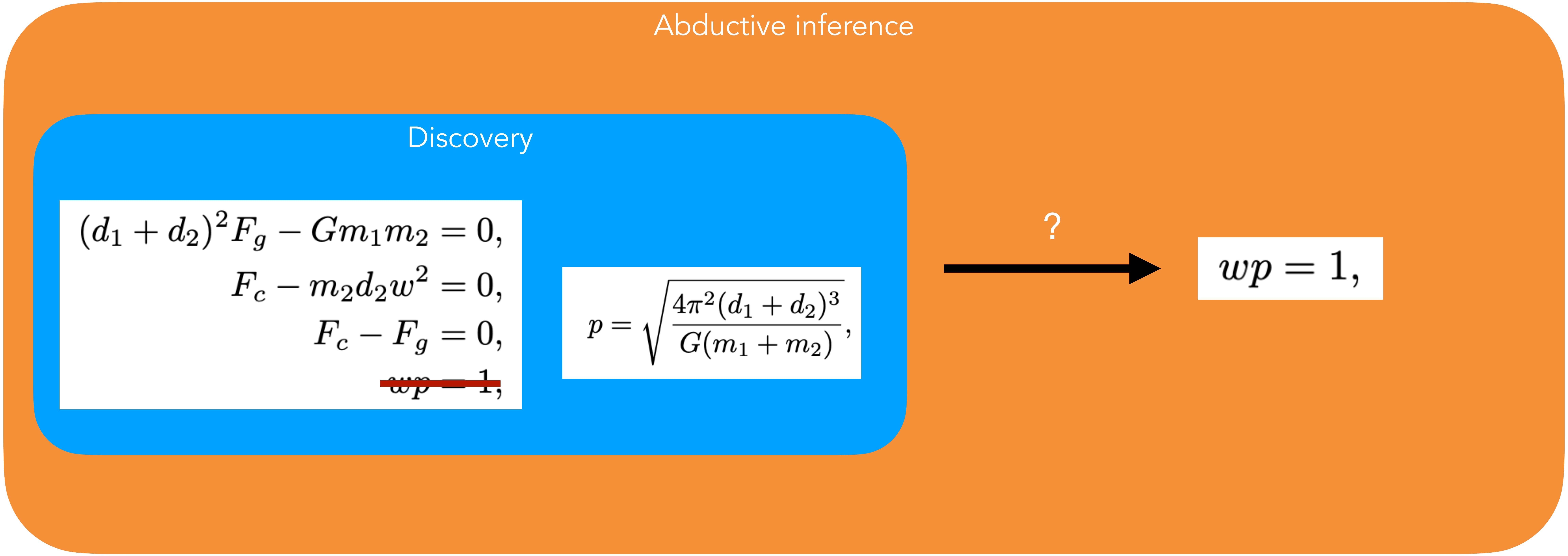
Discovery

$$(d_1 + d_2)^2 F_g - G m_1 m_2 = 0,$$
$$F_c - m_2 d_2 w^2 = 0,$$
$$F_c - F_g = 0,$$
$$wp = 1,$$

$$p = \sqrt{\frac{4\pi^2 (d_1 + d_2)^3}{G(m_1 + m_2)}},$$

Abductive inference

In practice, axiom systems can be incomplete. Can we rediscover a missing axiom in the discovery process?



Abductive inference

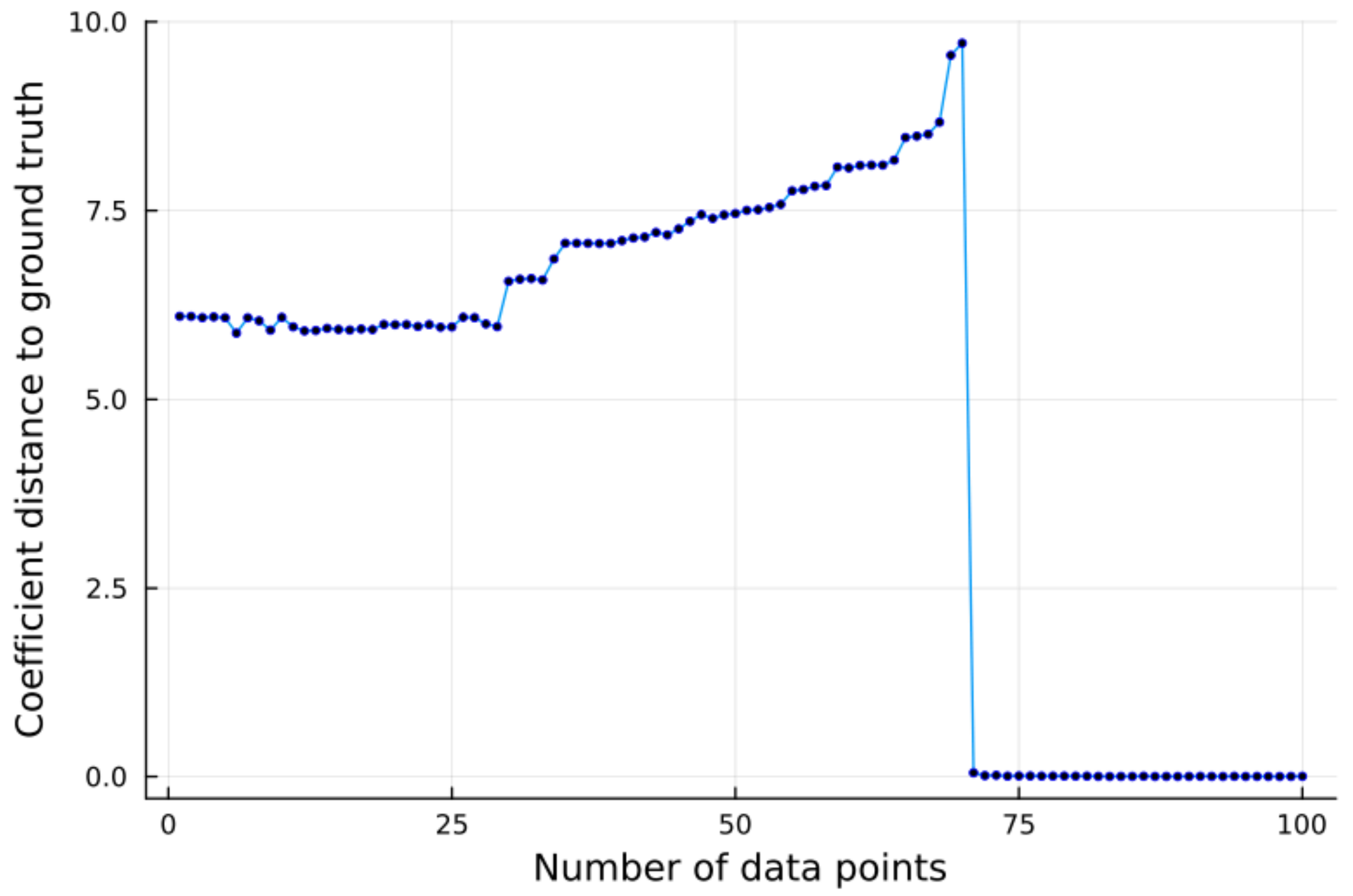
Key idea: Using data and variable elimination, we can still discover formulae with AI Hilbert

Assumption: The discovered formula q can be proven if we had an additional axiom.

i.e. $q = \alpha_1 f_1 + \alpha_2 f_2 + \alpha_3 f_3 + \alpha_4 F_4$

Where $\alpha_4 F_4$ is unknown.

Then $\langle f_1, f_2, f_3, q \rangle = \langle f_1, f_2, f_3, \alpha_4 F_4 \rangle$.



Rediscover Kepler with 70 data points if we omit $wp - 1$

$$p = \sqrt{\frac{4\pi^2(d_1 + d_2)^3}{G(m_1 + m_2)}},$$

Take the new "discovered" surface and break it into its irreducible components

$$V(f_1, f_2, f_3, q) = \bigcup V(p_i)$$

Where each $V(p_i)$ is irreducible. This is known as "primary decomposition" in commutative algebra.

Abductive inference

For Kepler's third law with $wp - 1$ missing, we get the following primary decomposition:

$$\langle f_1, f_2, f_3, q\rangle = \begin{array}{c} \langle m_2, F_g, F_c\rangle \cap \langle d_2, m_1, F_g, F_c\rangle \cap \langle d_1 + d_2, m_1, F_c - F_g, w^2 m_2 d_2 - F_g\rangle \cap \\ \langle F_g - Fc, (wp-1), m_1 p^2 - d_1^2 d_2 - 2d_1 d_2^2 - d_2^3, F_g(d_1+d_2)^2 - m_1 m_2, wm_2 d_2 - F_g p\rangle \cap \\ \cap \langle F_g - F_c, (wp+1), m_1 p^2 - d_1^2 d_2 - 2d_1 d_2^2 - d_2^3, F_g(d_1+d_2)^2 - m_1 m_2, wm_2 d_2 + F_g p\rangle \end{array}$$

$$\begin{array}{lr} f_1 & (d_1 + d_2)^2 F_g - G m_1 m_2 = 0, \\ f_2 & F_c - m_2 d_2 w^2 = 0, \\ f_3 & F_c - F_g = 0, \\ q & p^2 m_1 m_2 - d_1^2 d_2 m_2 - 2 d_1 d_2^2 m_2 - d_2^3 m_2 \end{array}$$

Abductive inference results

Problem & Axiom #	Axiom	Recovered*
Kepler 1	$(d_1 + d_2)^2 F_g - m_1 m_2$	✓
Kepler 2	$F_c - m_2 d_2 w^2$	✓
Kepler 3	$F_c - F_g$	✓
Kepler 4	$wp - 1$	✓
Compton 1	$E_1 + Ee_1 - E_2 - Ee_2$	✓
Compton 2	$E_1 - hf_1$	✓
Compton 3	$E_2 - hf_2$	✓
Compton 4	$p_1 c - hf_1$	✓
Compton 5	$p_2 c - hf_2$	✓
Compton 6	$\lambda_2 f_1 - c$	✓
Compton 7	$\lambda_2 f_2 - c$	✓
Compton 8	$Ee1 - m_c^2$	✓
Compton 9	$Ee2^2 - c^2 pe2^2 - me^2 c^4$	X^+
Compton 10	$pe2^2 - p2^2 - p1^2 + 2p1p2cos$	X^+
Einstein 1	$cdt_0 - 2 * d$	✓
Einstein 2	$4L^2 - 4d^2 - v^2 dt^2$	✓
Einstein 3	$f_0 dt_0 - 1$	✓
Einstein 4	$fdt - 1$	✓
Einstein 5	$c * dt - 2L$	✓
Escape Velocity 1	$K_i - \frac{1}{2} m v_e^2$	✓
Escape Velocity 2	$K_f = 0$	✓
Escape Velocity 3	$U_i r + GMm$	✓
Escape Velocity 4	$U_f = 0$	✓
Escape Velocity 5	$K_i + U_i - (K_f + U_f)$	✓
Light Damping 1	$Sr^2 - q_c^2 a_p^2 \sin\theta^2$	✓
Light Damping 2	$dA - 2\pi r^2 \sin\theta d\theta$	✓
Light Damping 3	$P - \int_0^\pi SdA$	✓
Light Damping 4	$\frac{4}{3} - \int_0^\pi \sin\theta^3 d\theta$	✓
Light Damping 5	$a_p^2 - \frac{1}{2} w^4 x_0^2$	✓
Hagen Poiseuille 1	$u - c_0 - c_2 r^2$	✓
Hagen Poiseuille 2	$\mu \frac{\partial}{\partial r}(r \frac{\partial}{\partial r} u) - r \frac{dp}{dx}$	✓
Hagen Poiseuille 3	$c_0 + c_2 R^2$	✓
Hagen Poiseuille 4	$L \frac{dp}{dx} = -\Delta p$	✓
Neutrino Decay 1	$p_v - p_\mu$	✓
Neutrino Decay 2	$E_\pi - m_\pi$	✓
Neutrino Decay 3	$E_v - p_v$	✓
Neutrino Decay 4	$E_\pi - E_\mu - E_v$	✓
Neutrino Decay 5	$E_\mu^2 - p_\mu^2 - m_\mu^2$	X^+
Hall Effect 1	$F_m - q_e v B$	✓
Hall Effect 2	$F_e - q_e E$	✓
Hall Effect 3	$F_m - F_e$	✓
Hall Effect 4	$Eh - U_H$	✓
Hall Effect 5	$vdt - L$	✓
Hall Effect 6	$Idt - Q$	✓
Hall Effect 7	$Q - Nq_e$	✓
Hall Effect 8	$nV - N$	X
Hall Effect 9	$V - Lhd$	X

Other goings on

Dataset and Benchmarking

Aim: Creating a synthetic dataset of polynomial axioms, data, and discovered formulae for benchmarking progress in the field

New Discoveries

Aim: Apply the tools developed to research problems in physics to discover new scientific facts. Current work ongoing in cosmology.

Non-polynomial Systems

Aim: Extending the current framework to systems involving differential equations, black box axioms, and trig functions

Karan Srivastava
ksrivastava4@wisc.edu