Symbolic Regression and Equation Learning

by Georg Martius

MPI for Intelligent Systems, Tübingen
Autonomous Learning Group

Max Planck ETH Center for Learning Systems

MAX PLANCK INSTITUTE FOR INTELLIGENT SYSTEMS

Typical loop of understanding...

Theory: something we can understand
involves mechanisms and their interaction

Theory

$$\frac{-\hbar^2}{2m}\frac{\partial^2\Psi}{\partial x^2} = i\hbar\frac{\partial\Psi}{\partial t}$$

istockphoto.com

Prediction

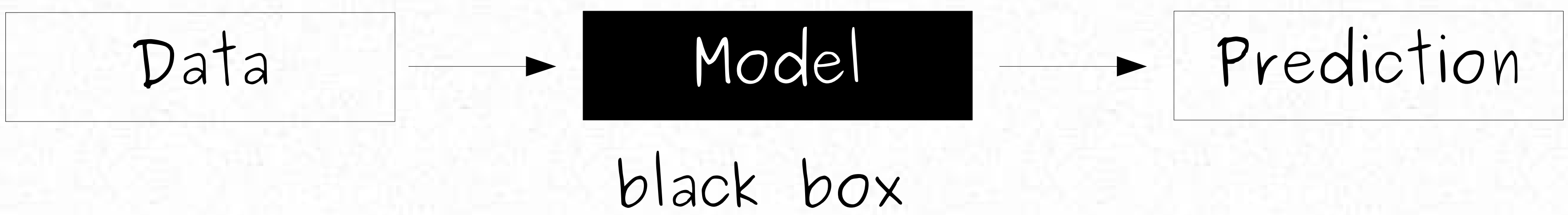
Experiment

martech.org

Observation

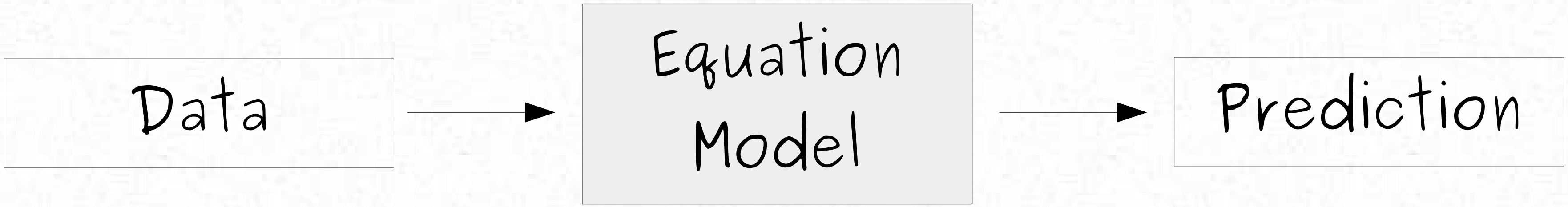
Fotolia/3dkombinat

Machine learning view



Machine learning view

Symbolic Regression



white box

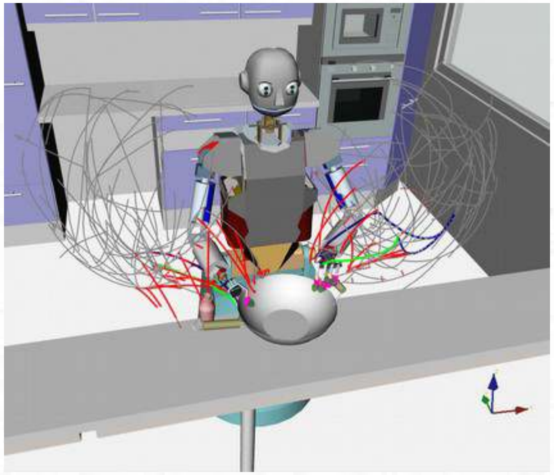
- identify underlying equations (from observed data)
- typically solved by discrete search (e.g. Genetic Algorithms)

Today:

- short overview of Symbolic Regression methods
- Symbolic Regression via Machine Learning
  - Equation Learner (EQL)
  - Transformer-based guided search (NeSymReS)

My Motivation

Model-based Control



(KIT H²T)

- for planning
- data efficiency
- safety

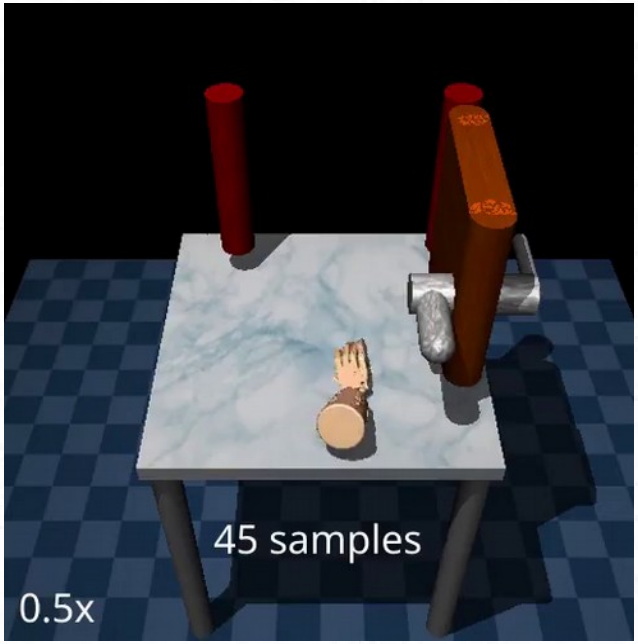
Natural Sciences

(Uni St. Andrews)

- Finding interpretable models of data
- Use prior knowledge of system

Model-based Planning

Instruction/reward: open the door



With an accurate model, planning can be very effective

Pinneri, Sawant, Blaes, …, GM. CORL 2020

Symbolic Regression

Data: $\{(x_1, y_1), (x_2, y_2), \cdots\}$

Ansatz: y = f(x) + noise
f is a concise analytical equation
(known base-functions and compositions)

$$y = \cos(x) \cdot e^{-2x}$$

Simple

x	y
0.1	0.81
0.4	0.41
1.2	0.03
0.9	0.10
...	...

Symbolic Regression

Data: $\{(x_1, y_1), (x_2, y_2), \cdots\}$

Ansatz: y = f(x) + noise

f is a concise analytical equation

(known base-functions and compositions)

$$y = \cos(x) \cdot e^{-2x}$$

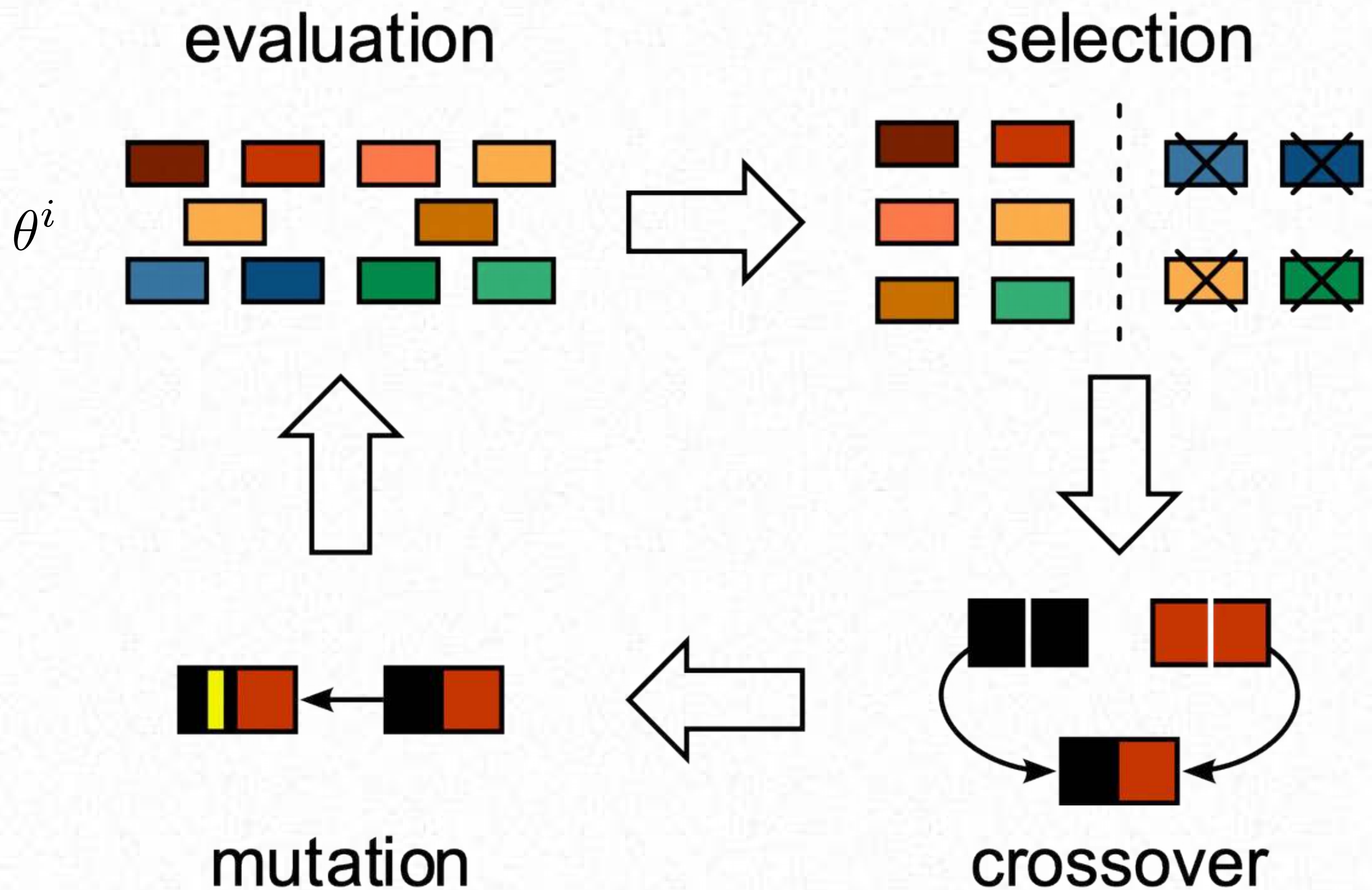
x
0.1
0.4
1.2
0.9
...

y
0.81
0.41
0.03
0.10
...

inference is hard!

Classical Solution: Genetic Algorithm

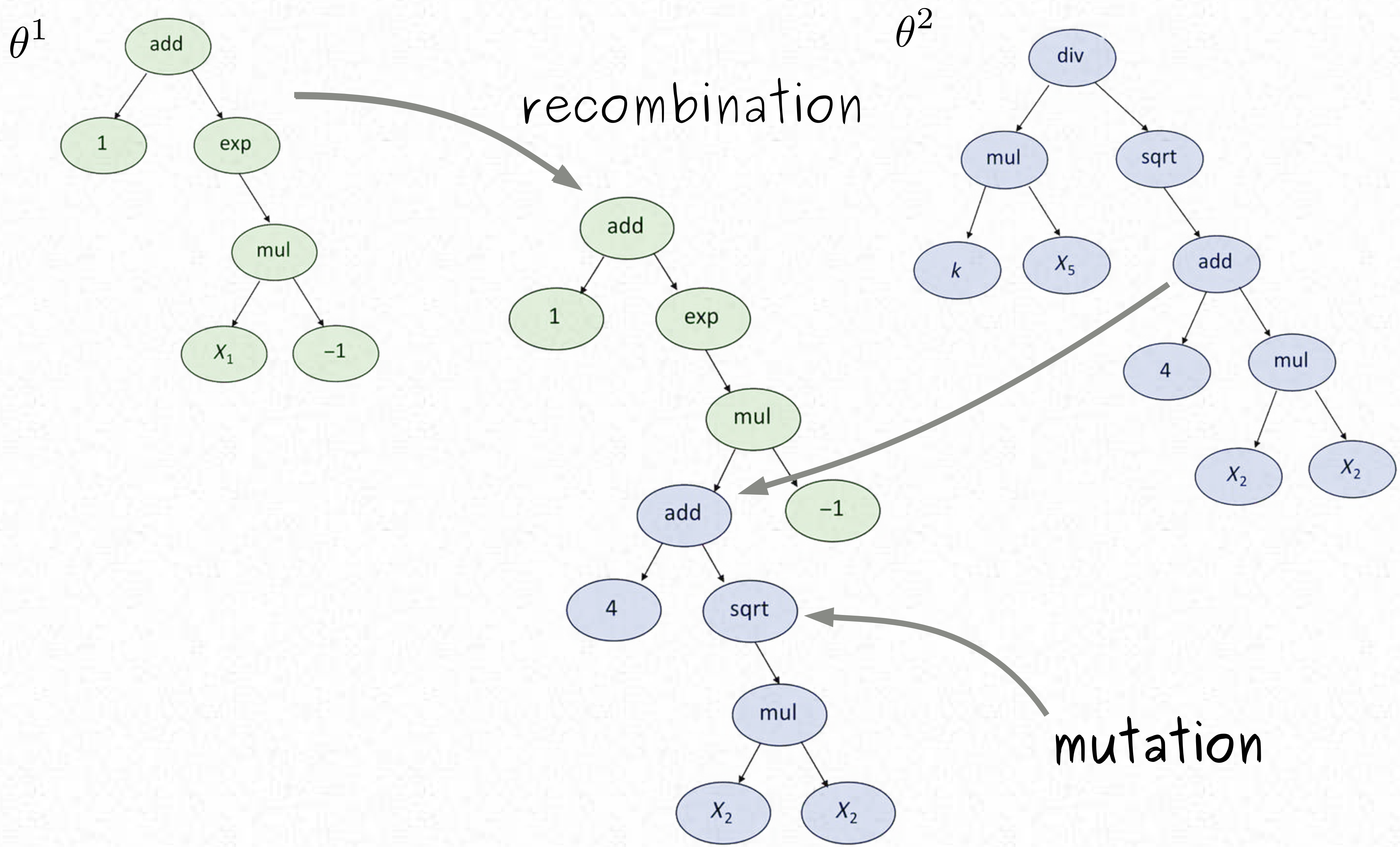
- discrete search method for solving $\arg\min_\theta f(\theta)$
- works for undifferentiable functions f
- works in relatively large spaces



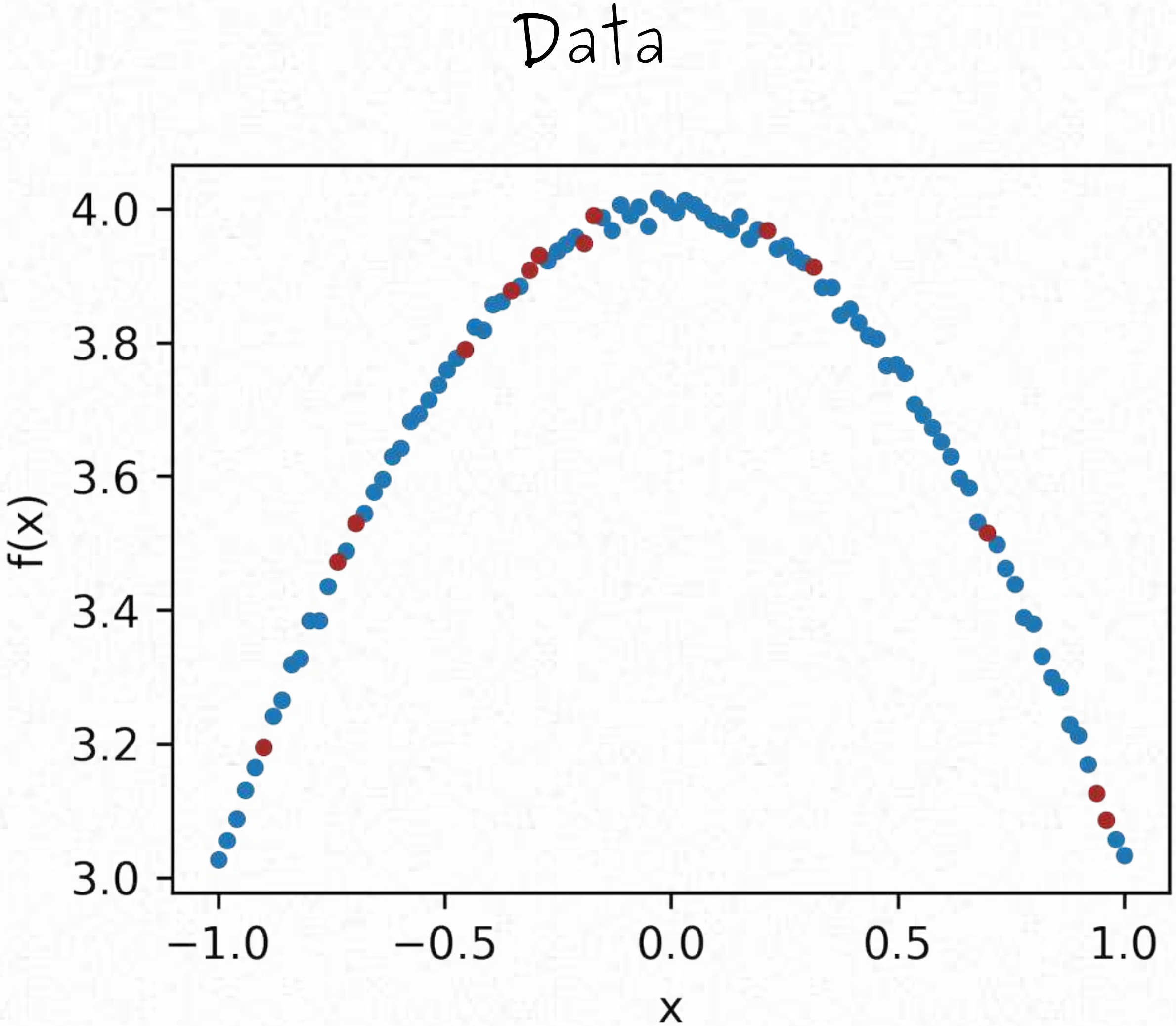
www.strong.io/blog/evolutionary-optimization

Classical Solution: Genetic Algorithm

- search in the space of expression graphs

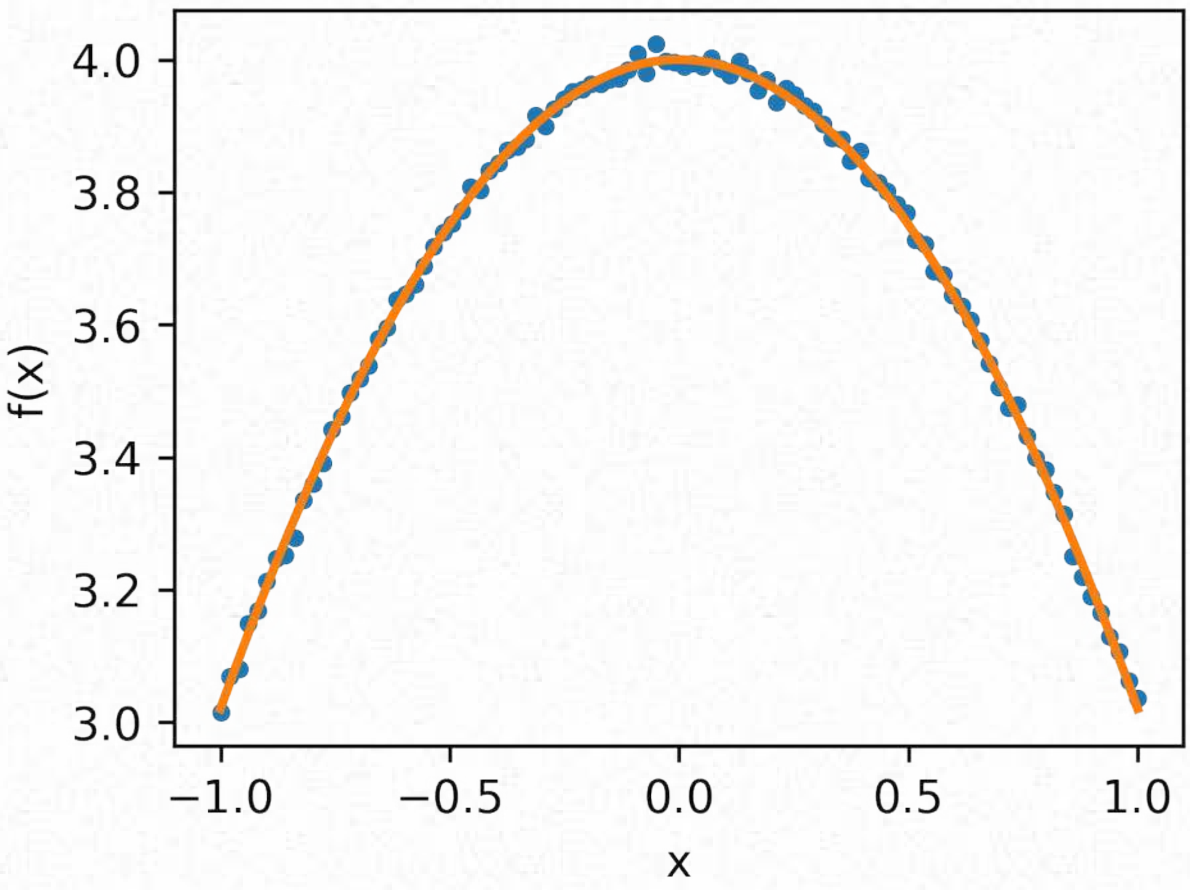


How to select a solution?



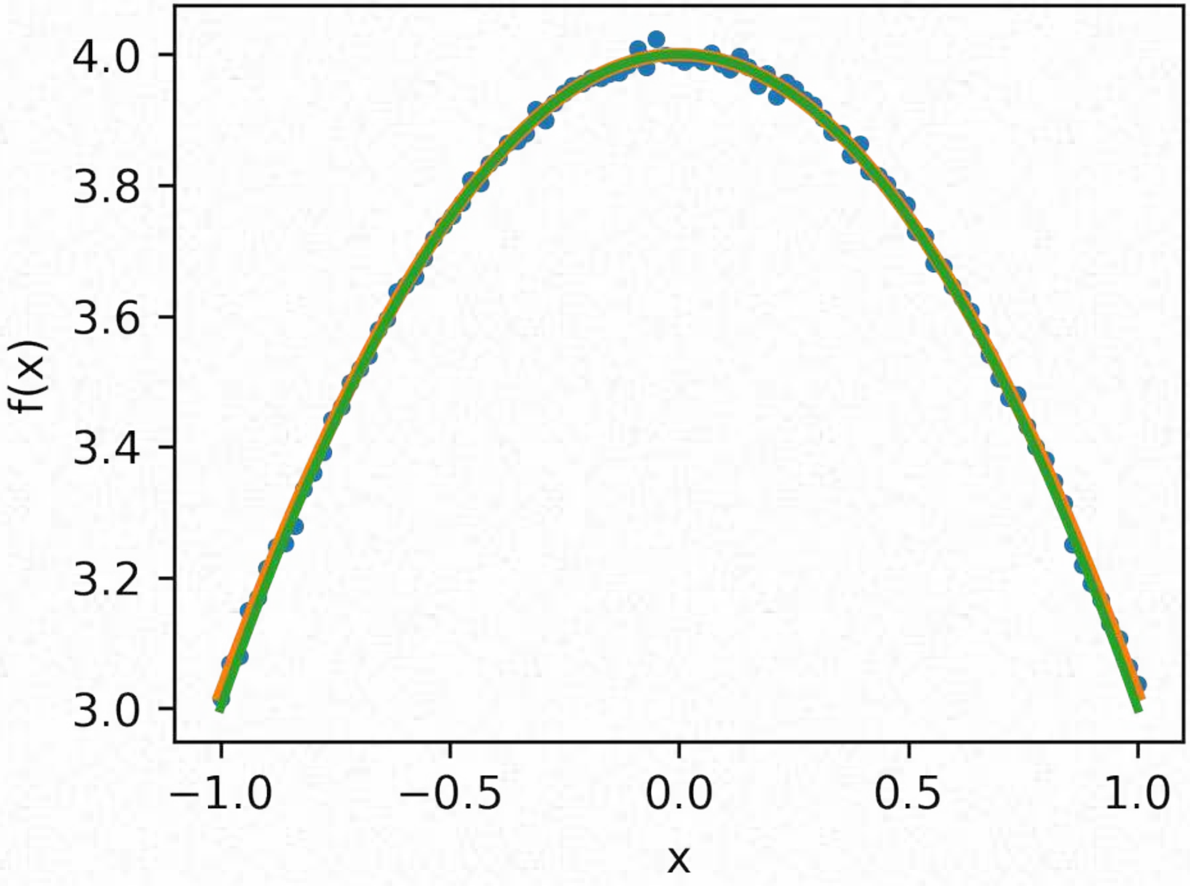
How to select a solution?

Solution 1



How to select a solution?

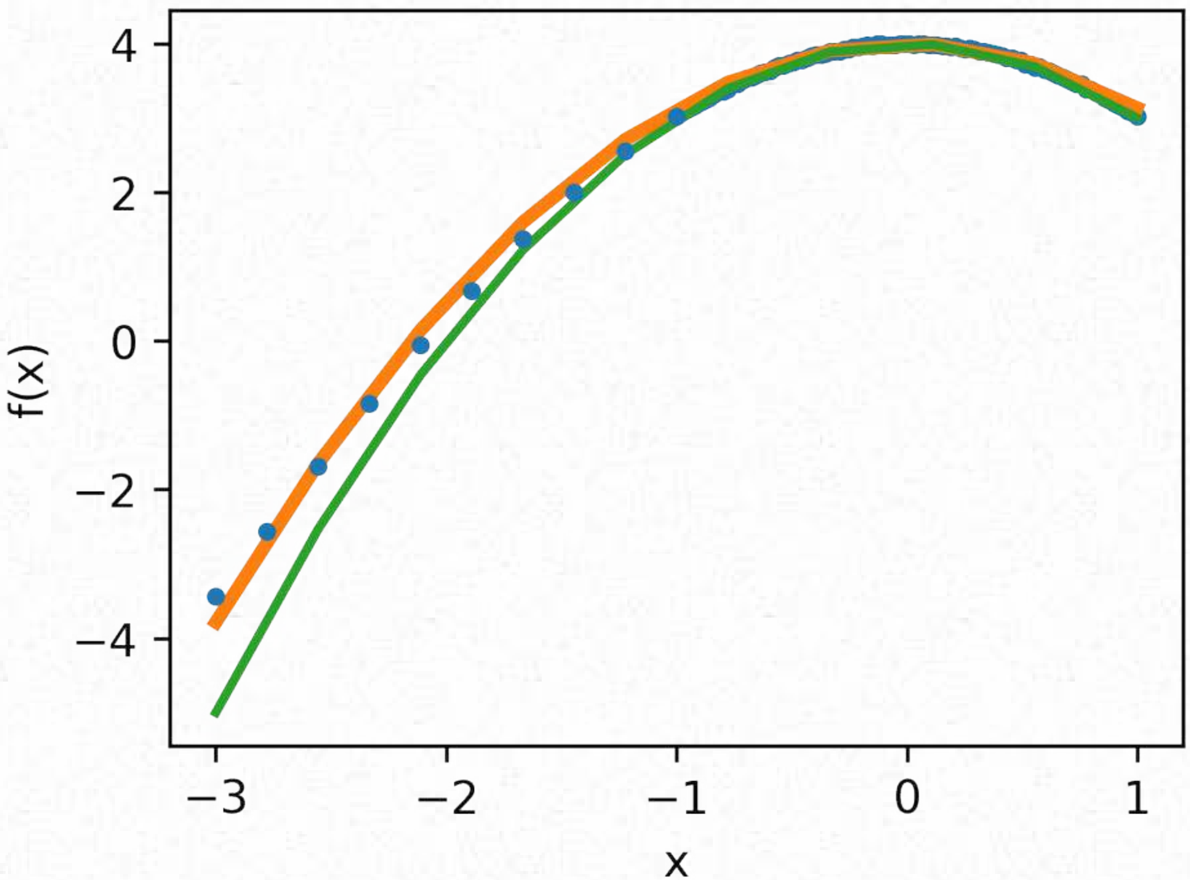
Solution 1 or solution 2?



perform the same with respect to i.i.d. validation data

How to select a solution?

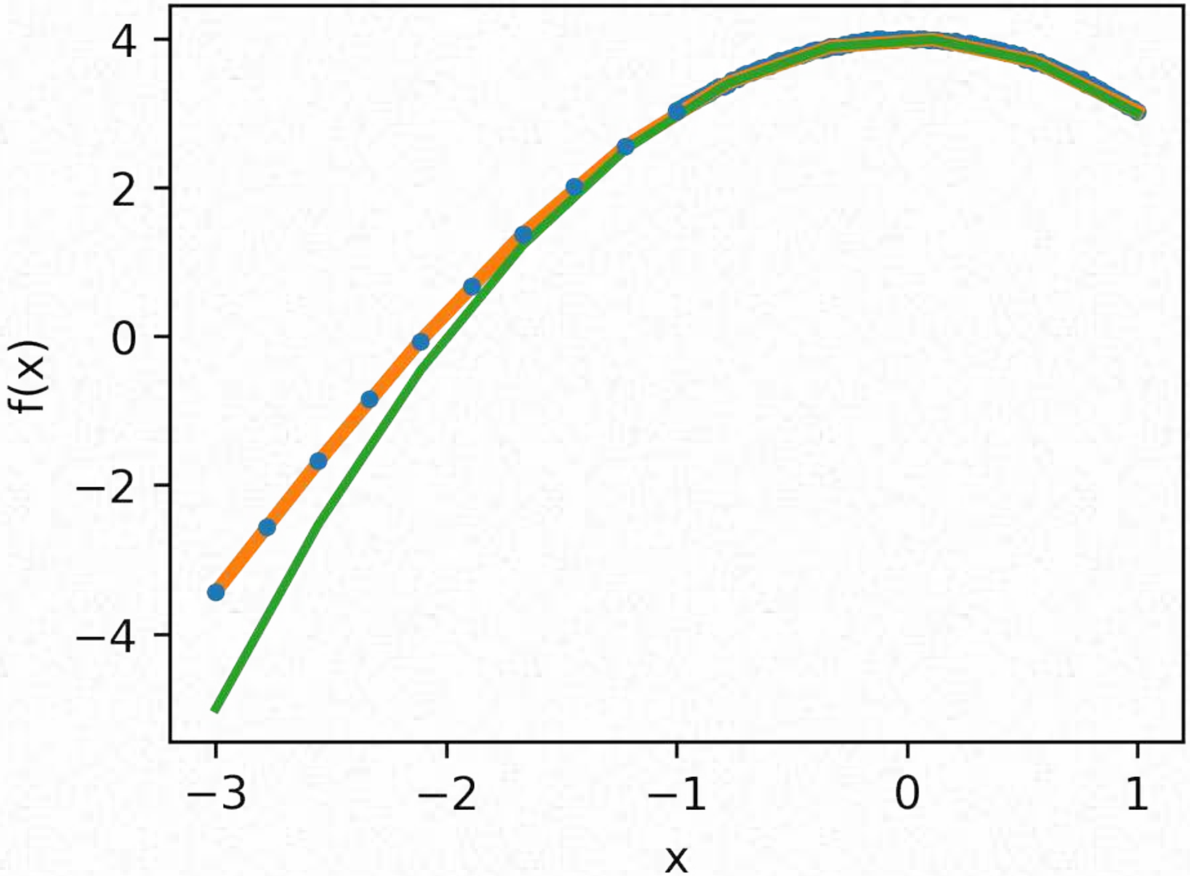
Maybe need more data?



Mh, orange line looks better
(but maybe just noise in data)

How to select a solution?

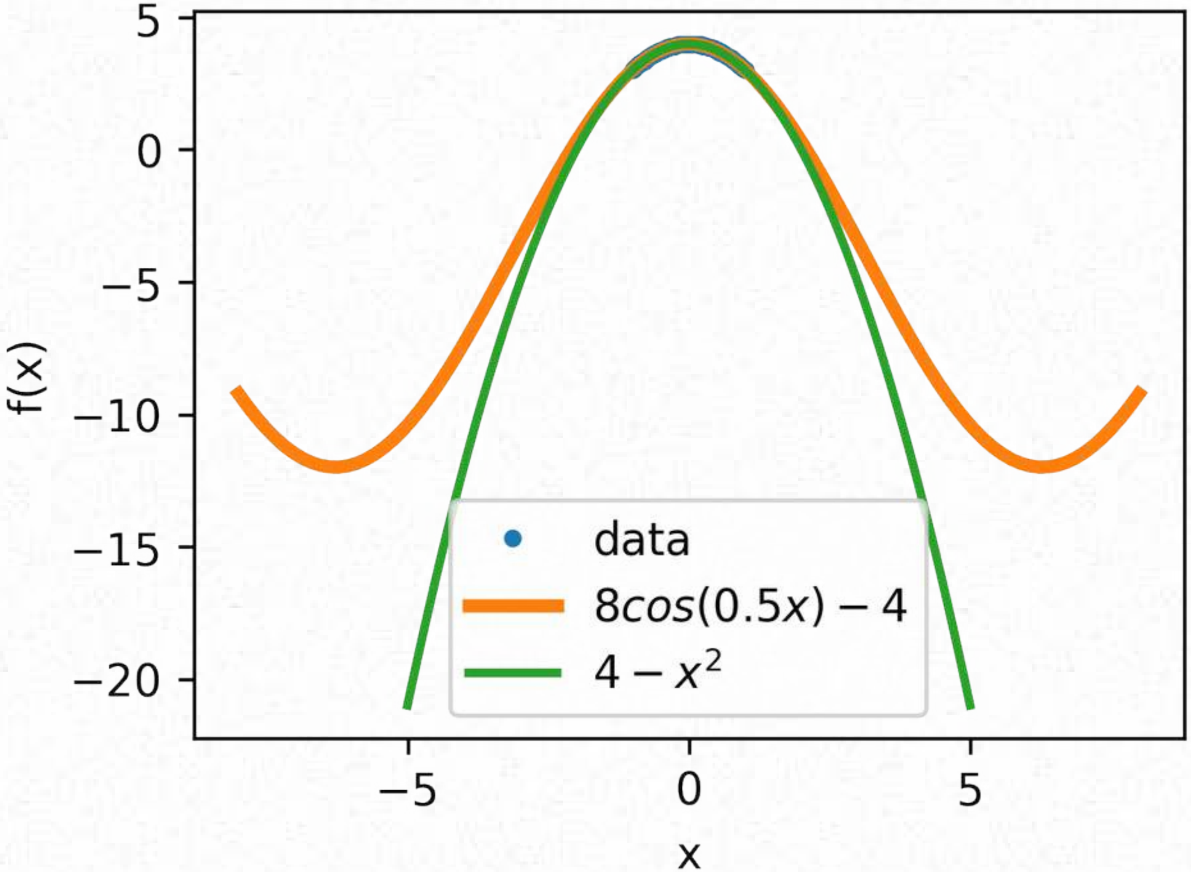
Maybe need more data?



Let's find a good fit...

How to select a solution?

Uff, quite different solutions after all...



Need domain knowledge (e.g. which functions are likely)

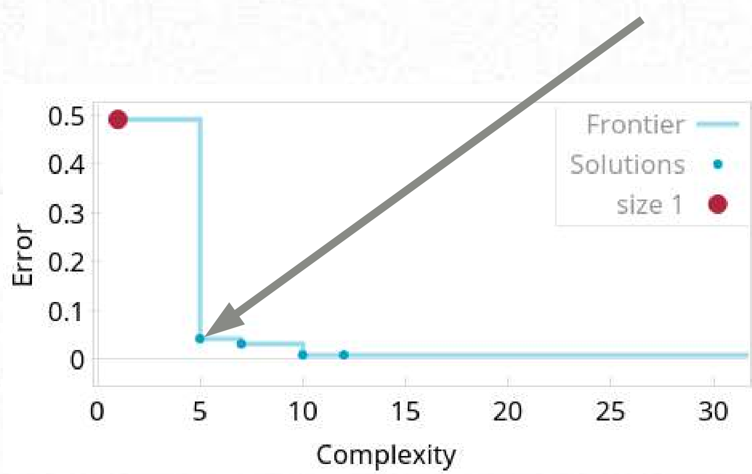
How to select a solution?

- Machine Learning: validation set
- Symbolic regression: take the least complex model

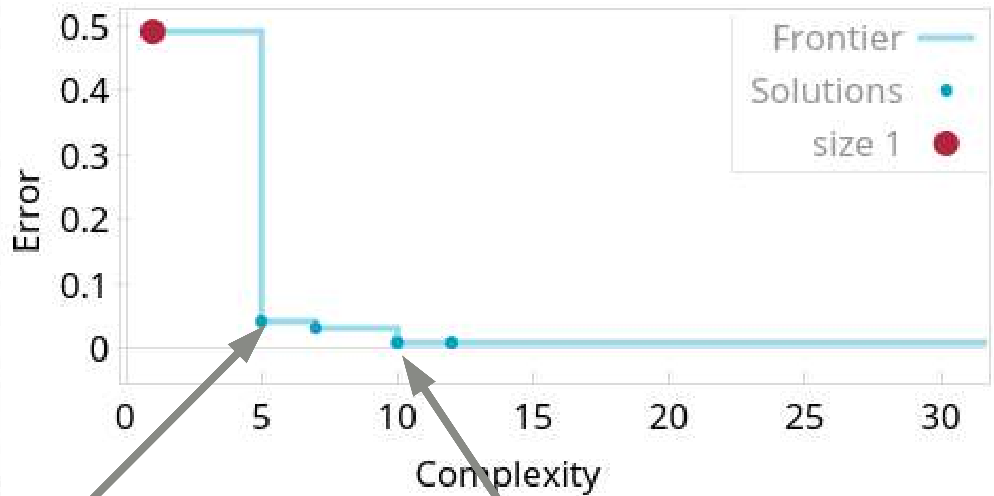
Is there really just one answer?

No → Family of Pareto-optimal solutions

best equation with this complexity



How to select a solution?

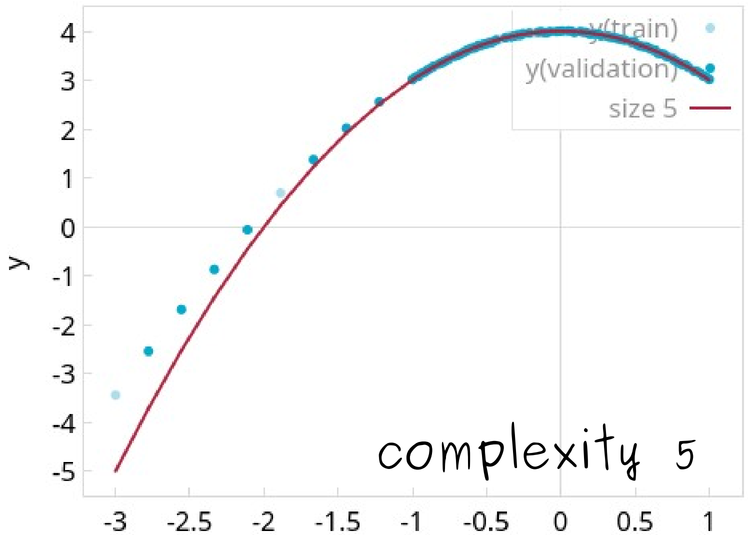


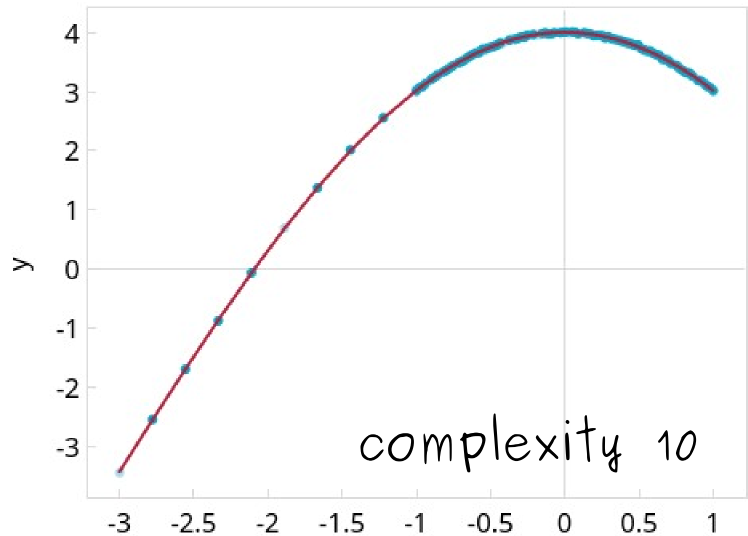
Generating function

$$y = 4 - x^2$$

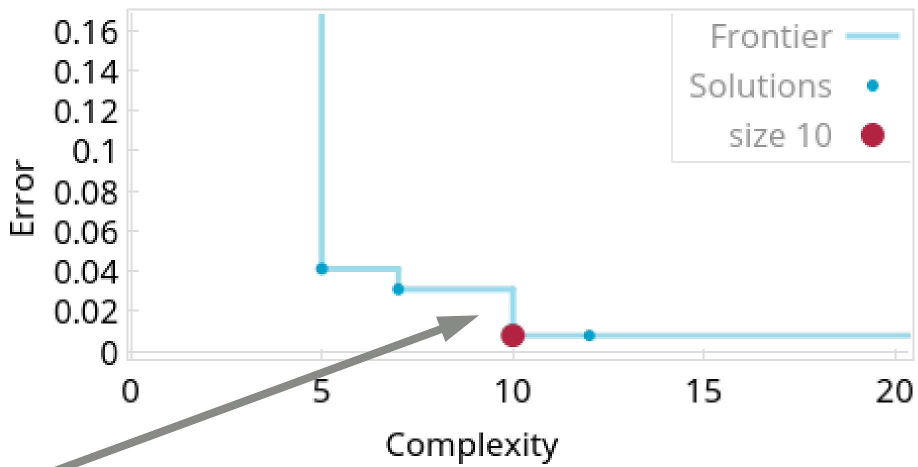
$$y = 7.82\cos(0.51x) - 3.81$$

$$y = 8\cos(0.5x) - 4$$





How to select a solution?

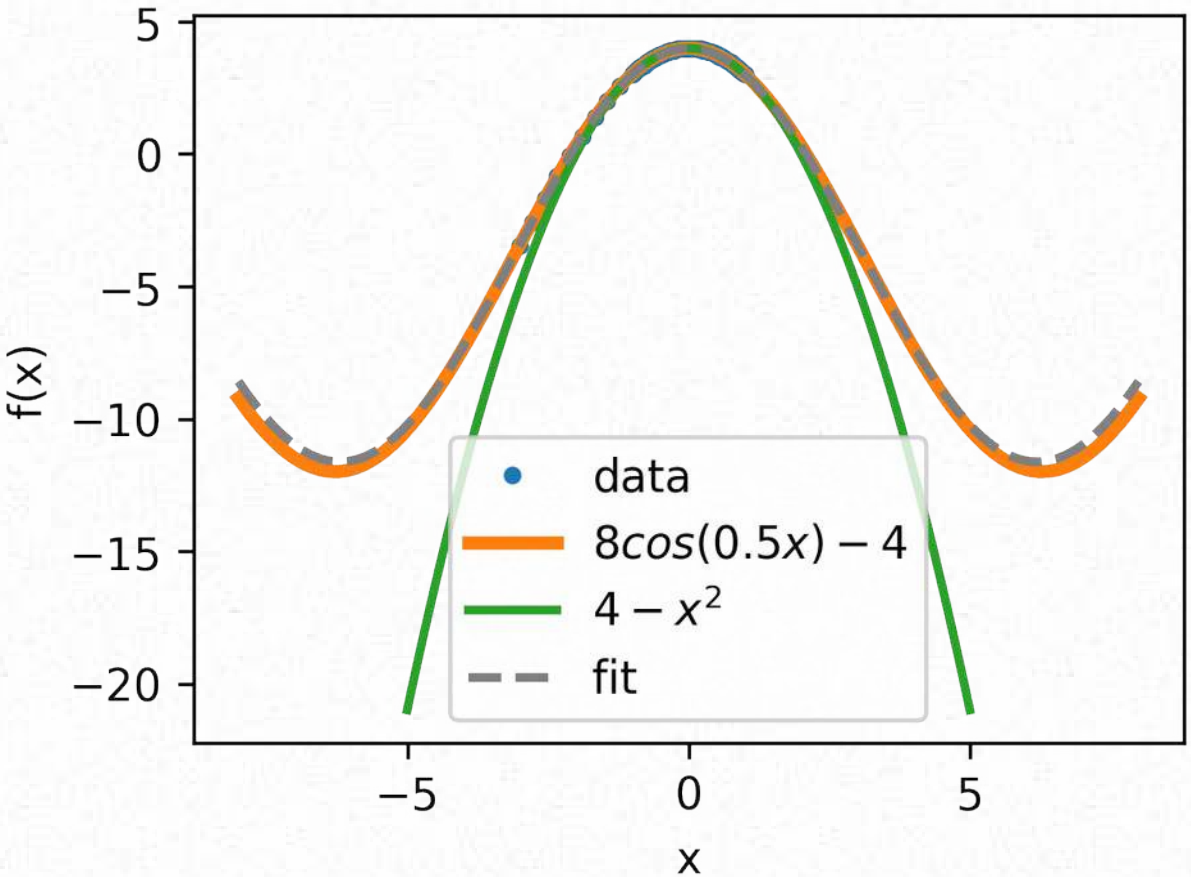


looking for jumps

Demo: run Eureqa

Eureqa: GA-based method, highly optimized
originally by Schmidt and Lipson
discontinued, now: https://www.datarobot.com/nutonian only as online tool

Results from Eureqa



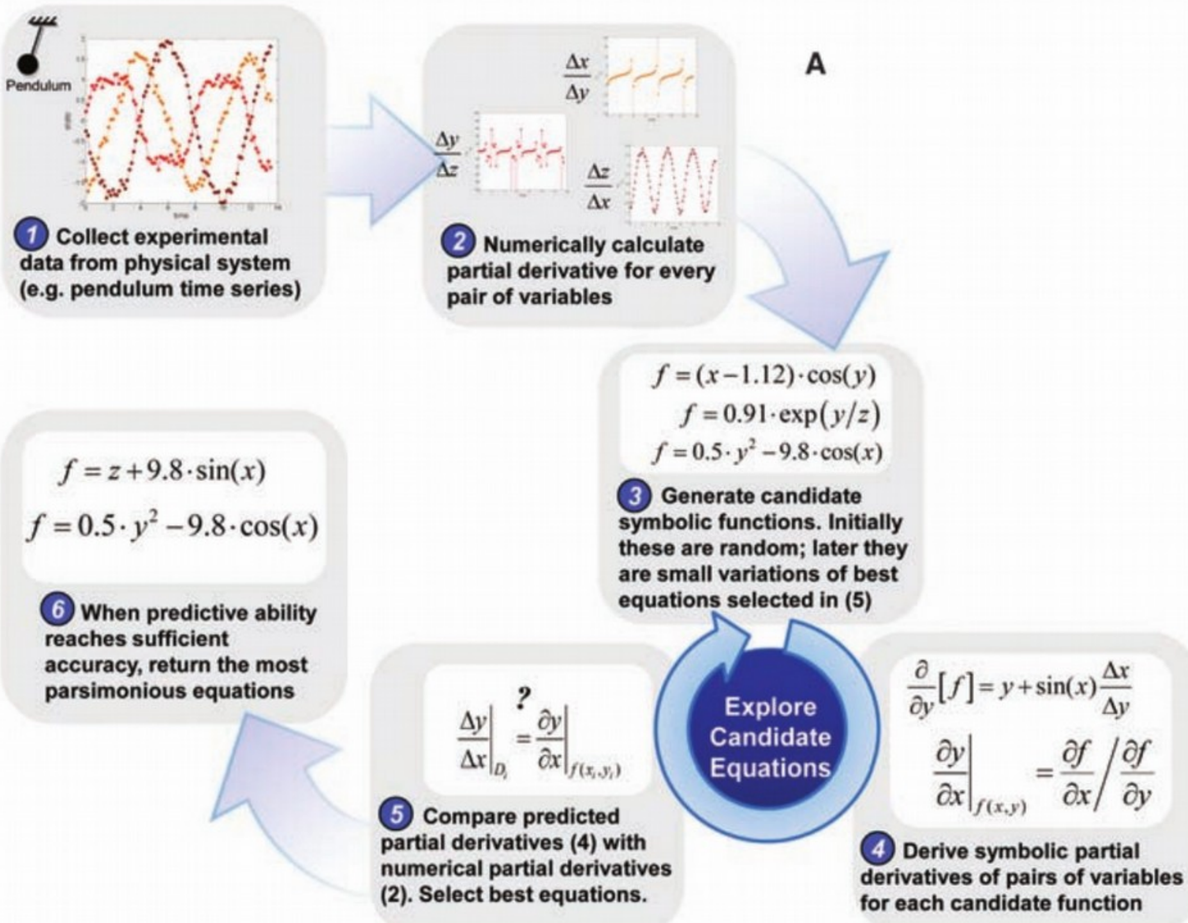
Eureqa: discontinued, now: https://www.datarobot.com/nutonian

Evolutionary search successes

Distilling Free-Form Natural Laws from Experimental Data

Michael Schmidt[1] and Hod Lipson[2,3]*

Science, 2009

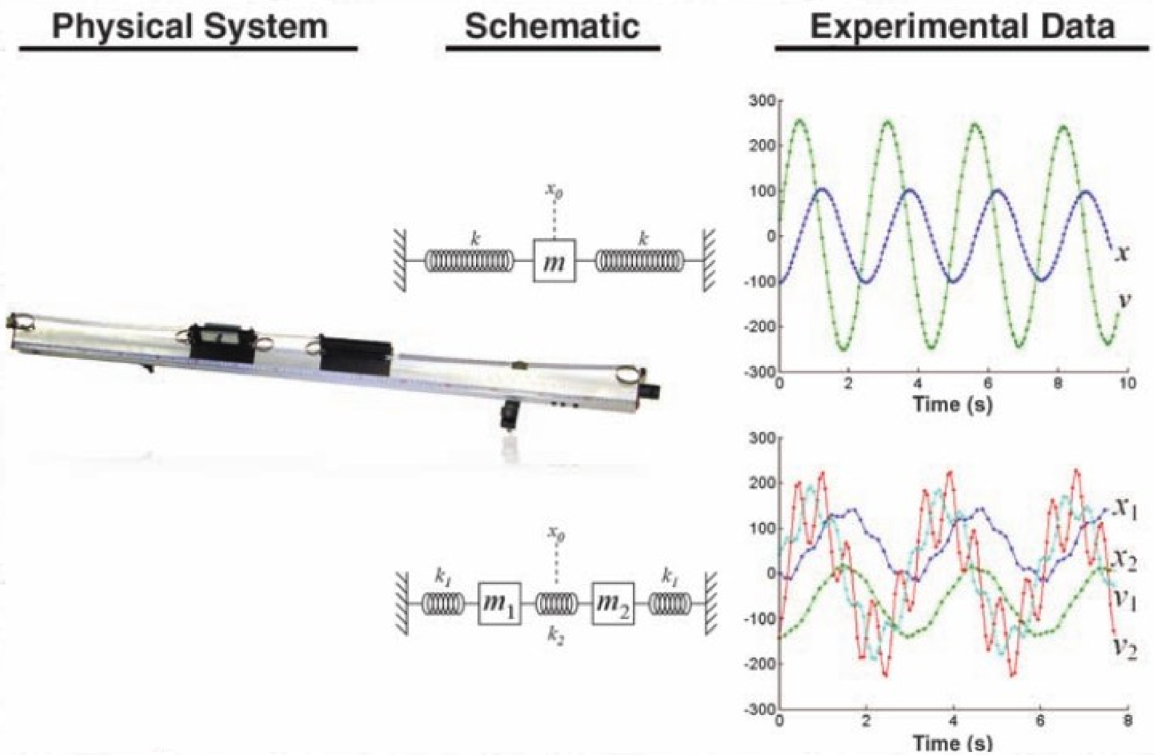


Evolutionary search successes

Distilling Free-Form Natural Laws from Experimental Data

Michael Schmidt[1] and Hod Lipson[2,3]*

Science, 2009



Inferred Laws

$$114.28v^2 + 692.32x^2$$

Hamiltonian

$$v^2 - 6.04x^2$$

Lagrangian

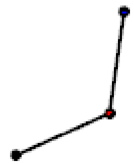
$$a - 0.008v - 6.02x$$

Equation of motion

$$-142.19x_1 - 74.65x_2 + 0.12x_1^2 - 1.89x_1x_2 - 1.51x_2^2 - 0.49v_2^2 + 0.41v_1v_2 - 0.082v_1^2$$

Lagrangian

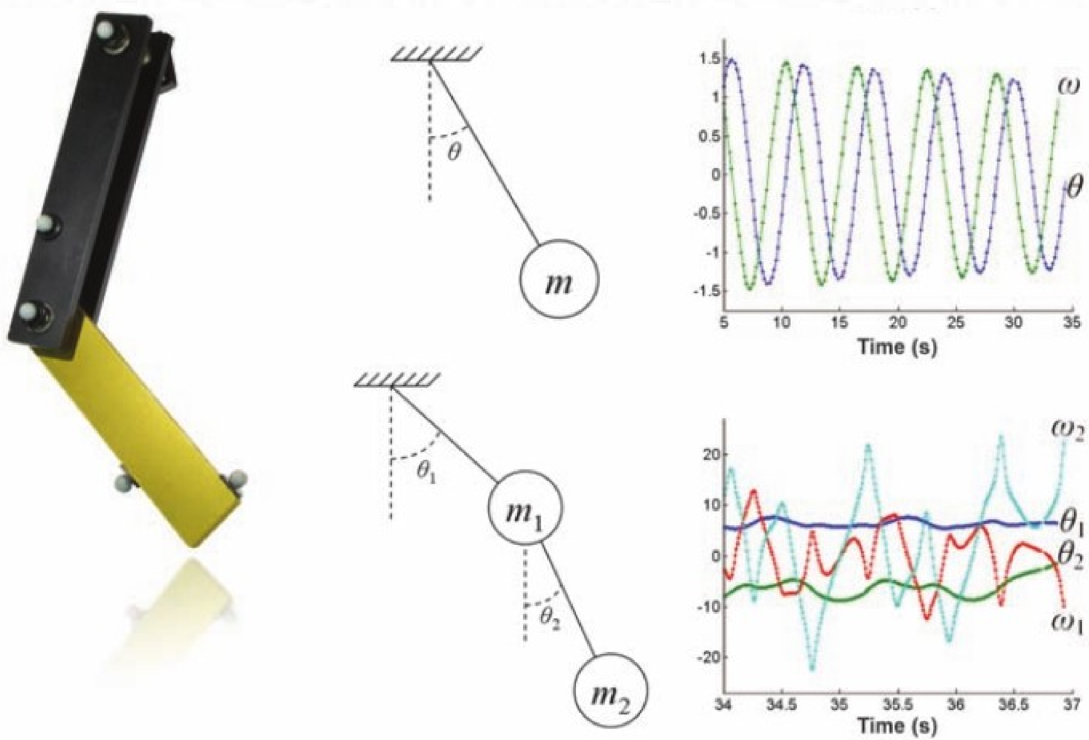
Evolutionary search successes



Distilling Free-Form Natural Laws from Experimental Data

Michael Schmidt[1] and Hod Lipson[2,3]*

Science, 2009



$$1.37\cdot\omega^2 + 3.29\cdot\cos(\theta)$$

Lagrangian

$$2.71\alpha + 0.054\omega - 3.54\sin(\theta)$$

Equation of motion

$$(x - 77.72)^2 + (y - 106.48)^2$$

Circular manifold

$$\omega_1{}^2 + 0.32\omega_2{}^2 - 124.13\cos(\theta_1) - 46.82\cos(\theta_2) + 0.82\omega_1\omega_2\cos(\theta_1 - \theta_2)$$

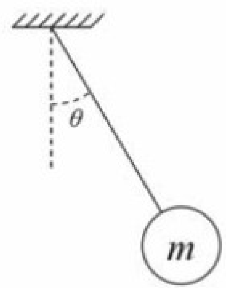
Hamiltonian

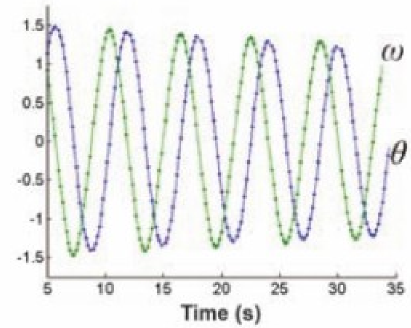
Evolutionary search successes

Distilling Free-Form Natural Laws from Experimental Data

Michael Schmidt[1] and Hod Lipson[2,3]*

Science, 2009





$$1.37 \cdot \omega^2 + 3.29 \cdot \cos(\theta)$$

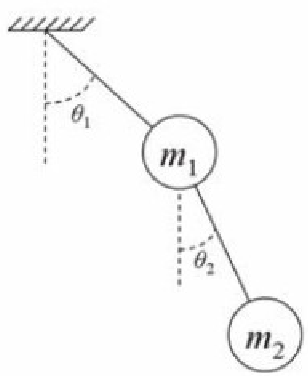
Lagrangian

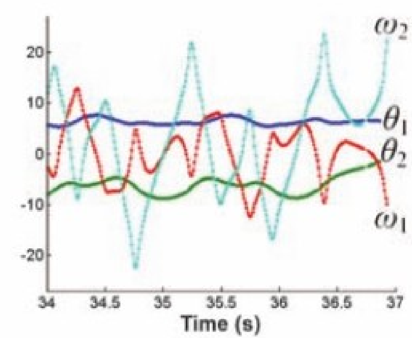
$$2.71\alpha + 0.054\omega - 3.54\sin(\theta)$$

Equation of motion

$$(x - 77.72)^2 + (y - 106.48)^2$$

Circular manifold





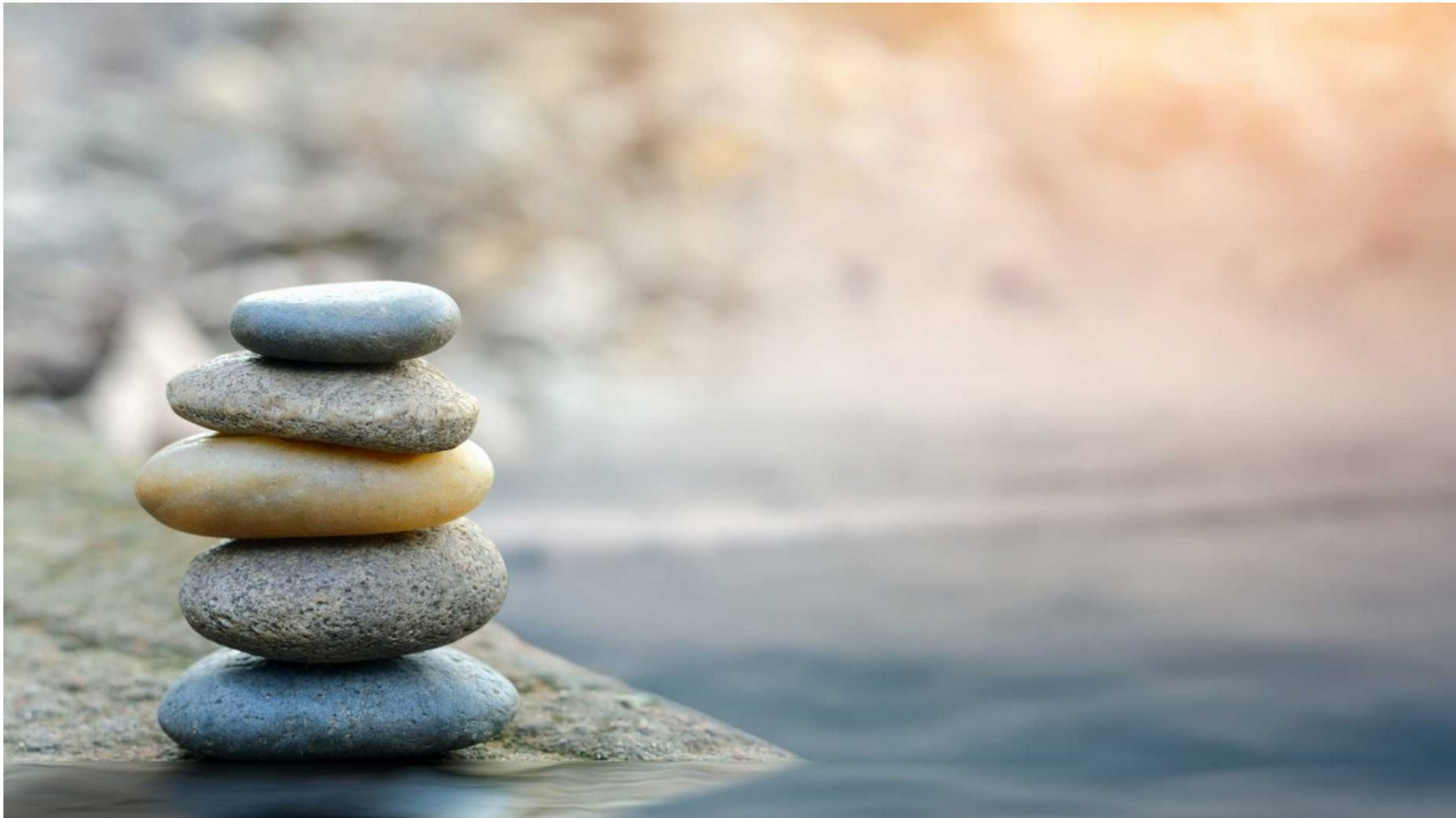
$$\omega_1^2 + 0.32\omega_2^2 - 124.13\cos(\theta_1) - 46.82\cos(\theta_2) + 0.82\omega_1\omega_2\cos(\theta_1 - \theta_2)$$

Hamiltonian

Problems with Evolutionary Search

- does not scale well to
  - high dimensional problems
  - more complex expressions
  - takes long to get all constants right

Short Break



Getty Images

Differentiable Architecture for Equation Learning

Data: $\{(x_1, y_1), (x_2, y_2), \cdots\}$

Assumption: y = f(x) + noise   f is in the model class

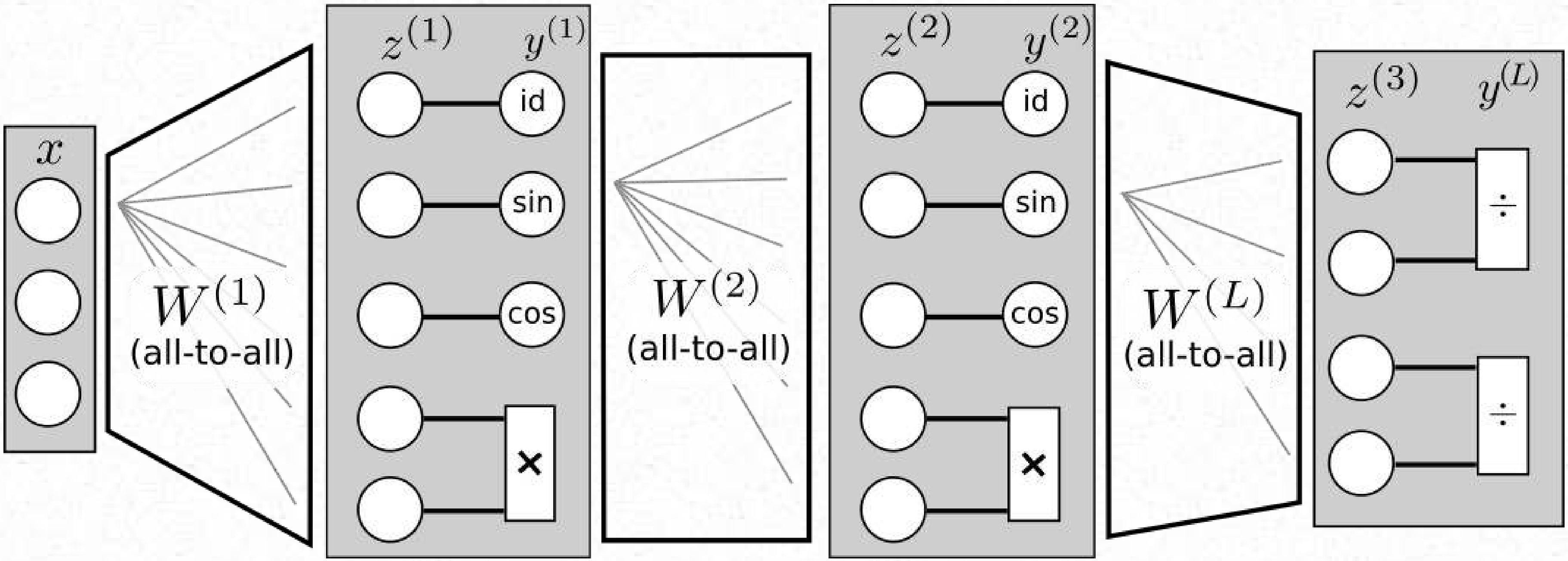
Subham S Sahoo
Google, India

Christoph Lamper
IST Austria

Differentiable Architecture for Equation Learning

Data: $\{(x_1, y_1), (x_2, y_2), \cdots\}$

Assumption: y = f(x) + noise   f is in the model class



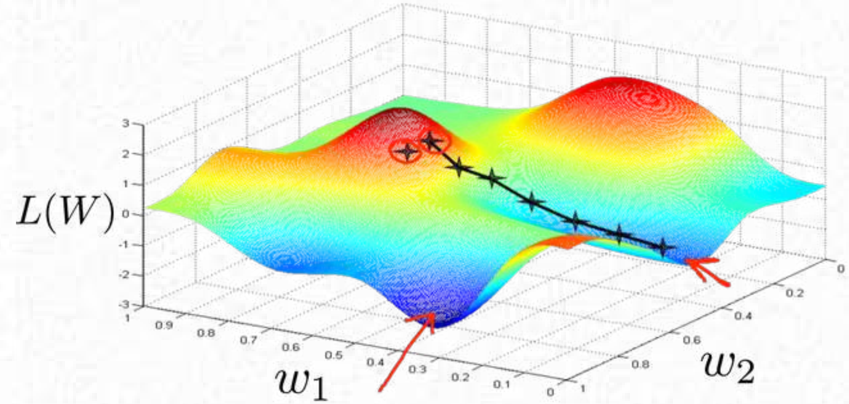
Replace standard units of NN

by id, sin, cos, multiplication and division.

Sahoo, Lampert, Martius, ICML 2018

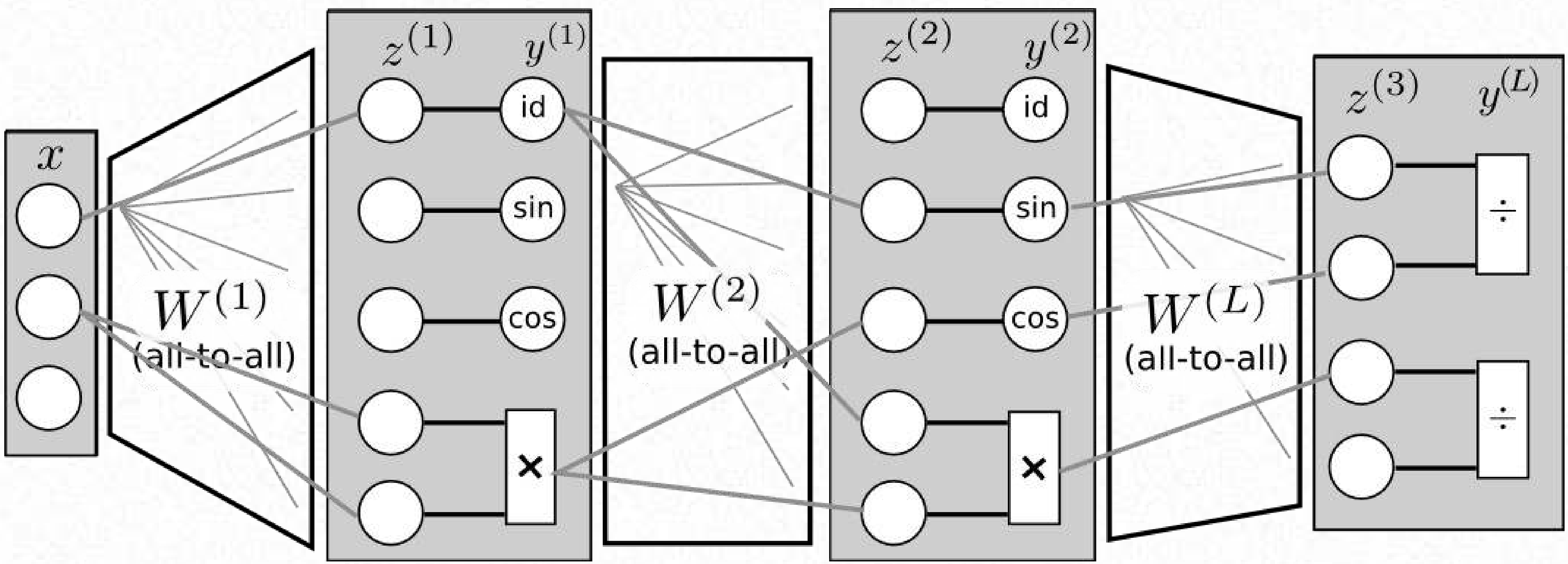
Regression with sparsity regularization

$$L = \sum_{i=1}^{n} |f(x_i, W) - y_i|^2 + \lambda |W|_1$$



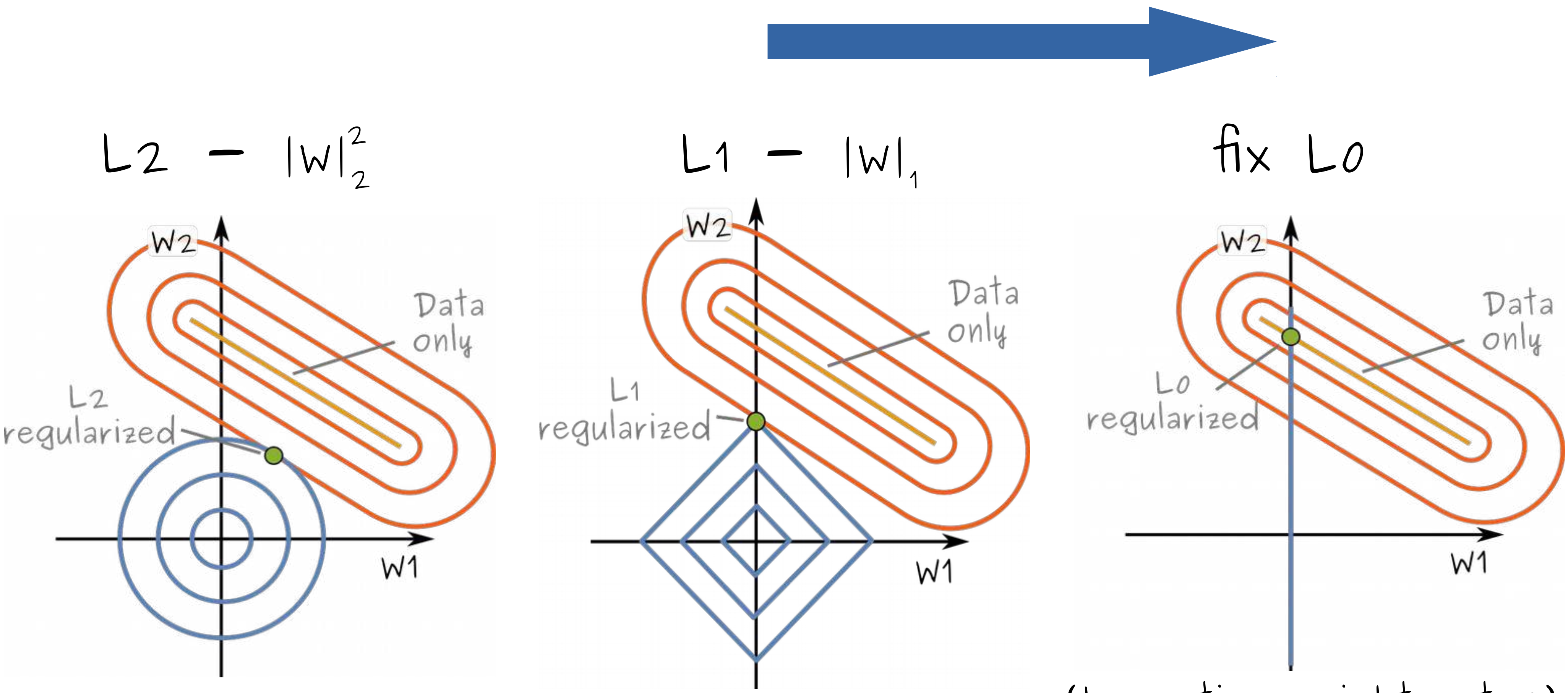
Training by gradient descent

$$\Delta W \propto -\frac{\partial L}{\partial W}$$



Regularization Phases

- Want: sparse solution

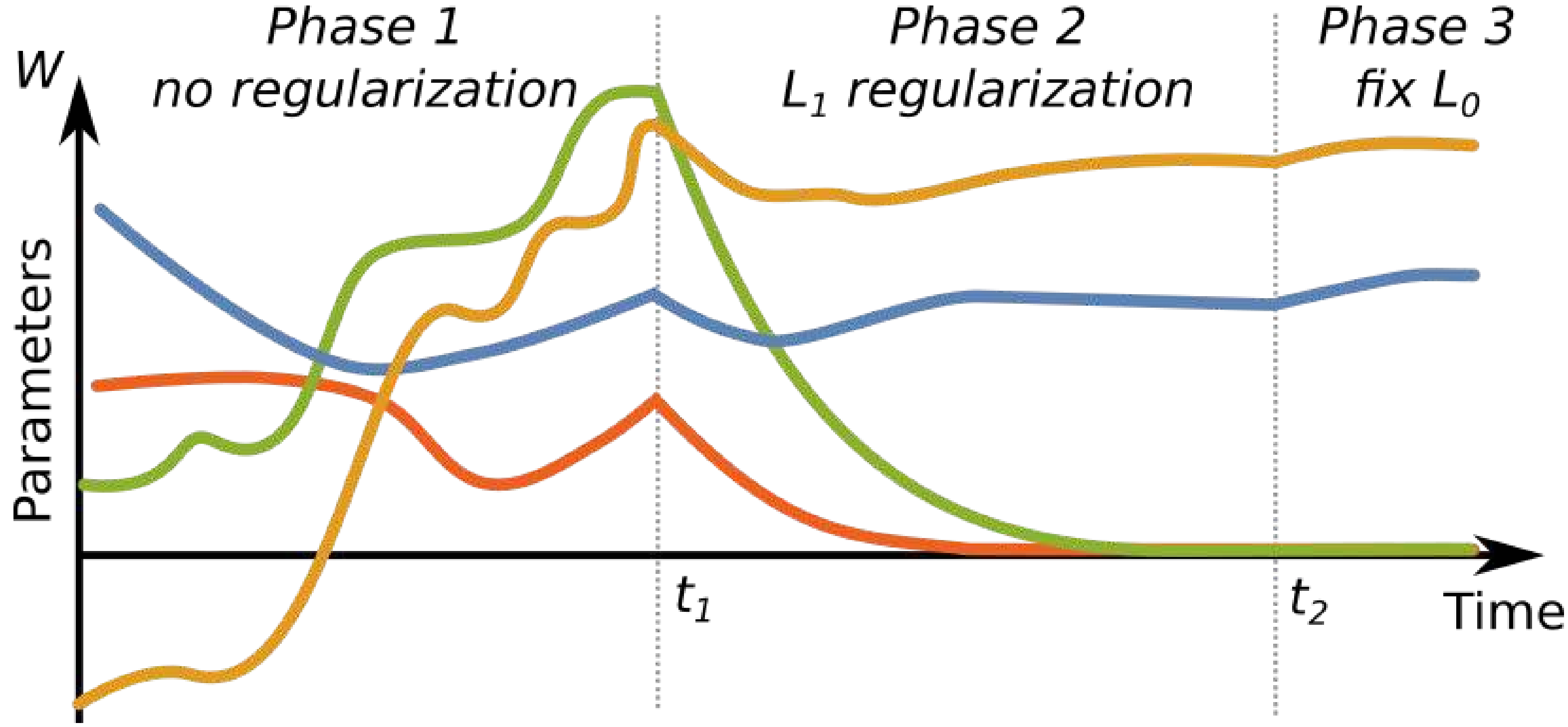


- (keep tiny weights at 0)

- sparse solution without tradeoff

Sahoo, Lampert, Martius, ICML 2018

Regularization Phases



Martius, Lampert arXiv 2016

Directly optimize for sparsity → L0 norm

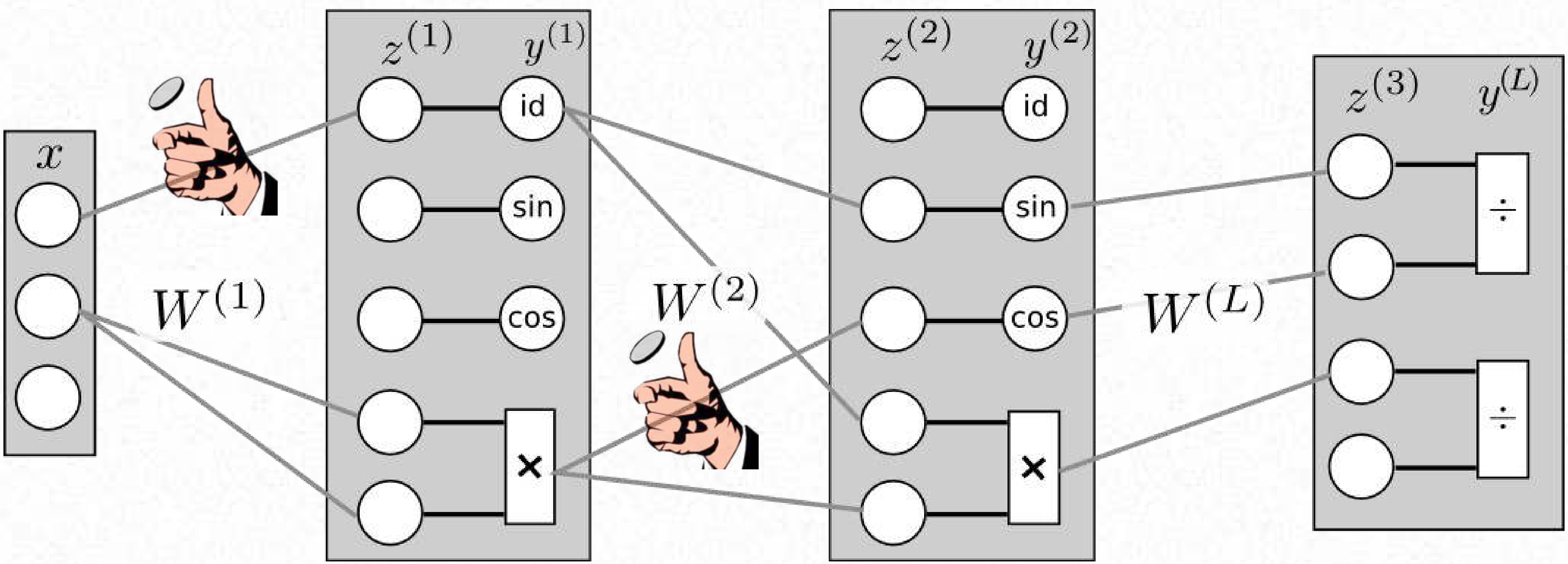
$$L = \sum_{i=1}^{n} |f(x_i, W) - y_i|^2 + \lambda |W|_0$$

How many non-zero entries

~~Training by gradient descent~~

$$\cancel{\Delta W \propto -\frac{\partial L}{\partial W}}$$

Trick: learn probability of usage

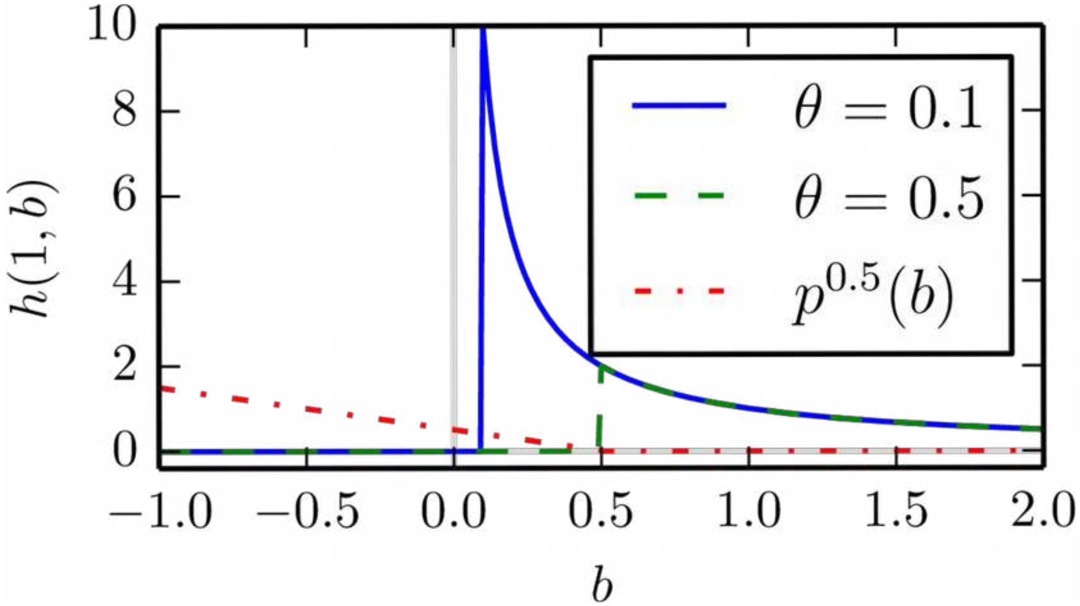


Bayesian compression/learned dropout ICLR 2018, ArXiv: 1712.01312 by Christos Louizos, Max Welling, Diederik P. Kingma

Treating Singular Units

Regularized version of a/b: $h^{\theta}(a,b) := \begin{cases} \frac{a}{b} & \text{if } b > \theta \\ 0 & \text{otherwise} \end{cases}$

Penalty for „wrong side": $p^{\theta}(b) := \max(\theta - b, 0)$



Better: Learn threshold: $\theta = \log(1 + e^{\alpha}) > 0$

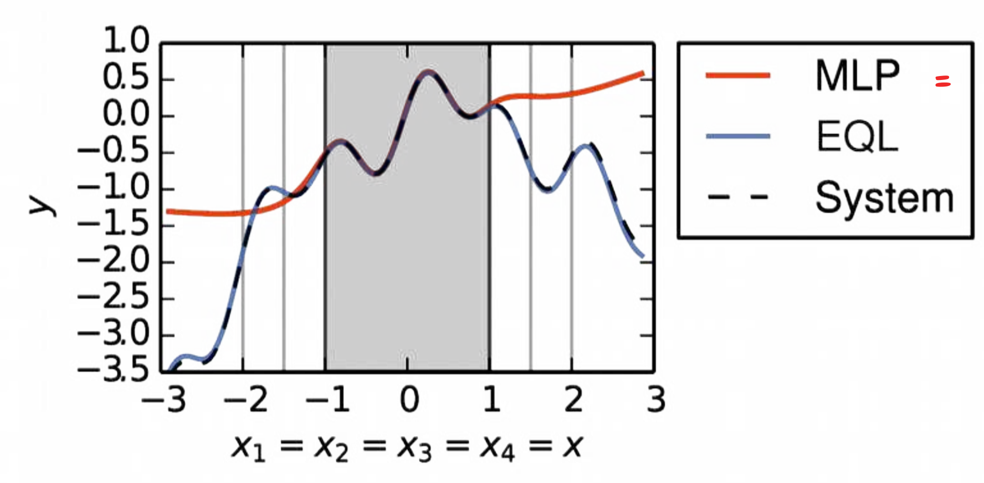
works for: log, 1/a, sqrt (singular derivative)

see Werner et al. arXiv 2105.06331

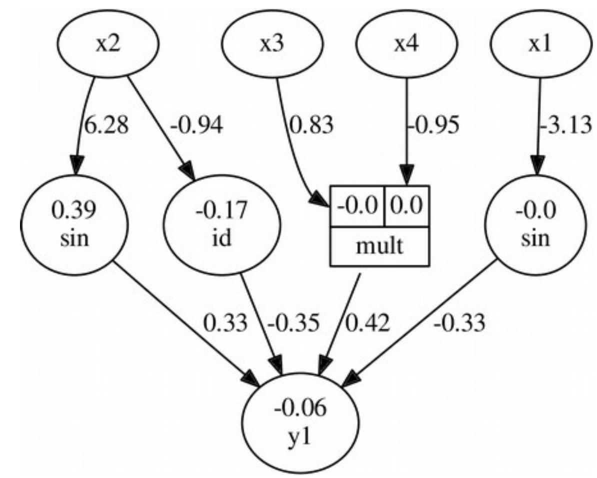
Sahoo, Lampert, Martius, ICML 2018

Function learning and extrapolation

Toy example: $y = {}^{1}/_{3}(\sin(\pi x_1) + \sin(2\pi x_2 + {}^{\pi}/_{8}) + x_2 - x_3 x_4) + \xi$



= Neural Net

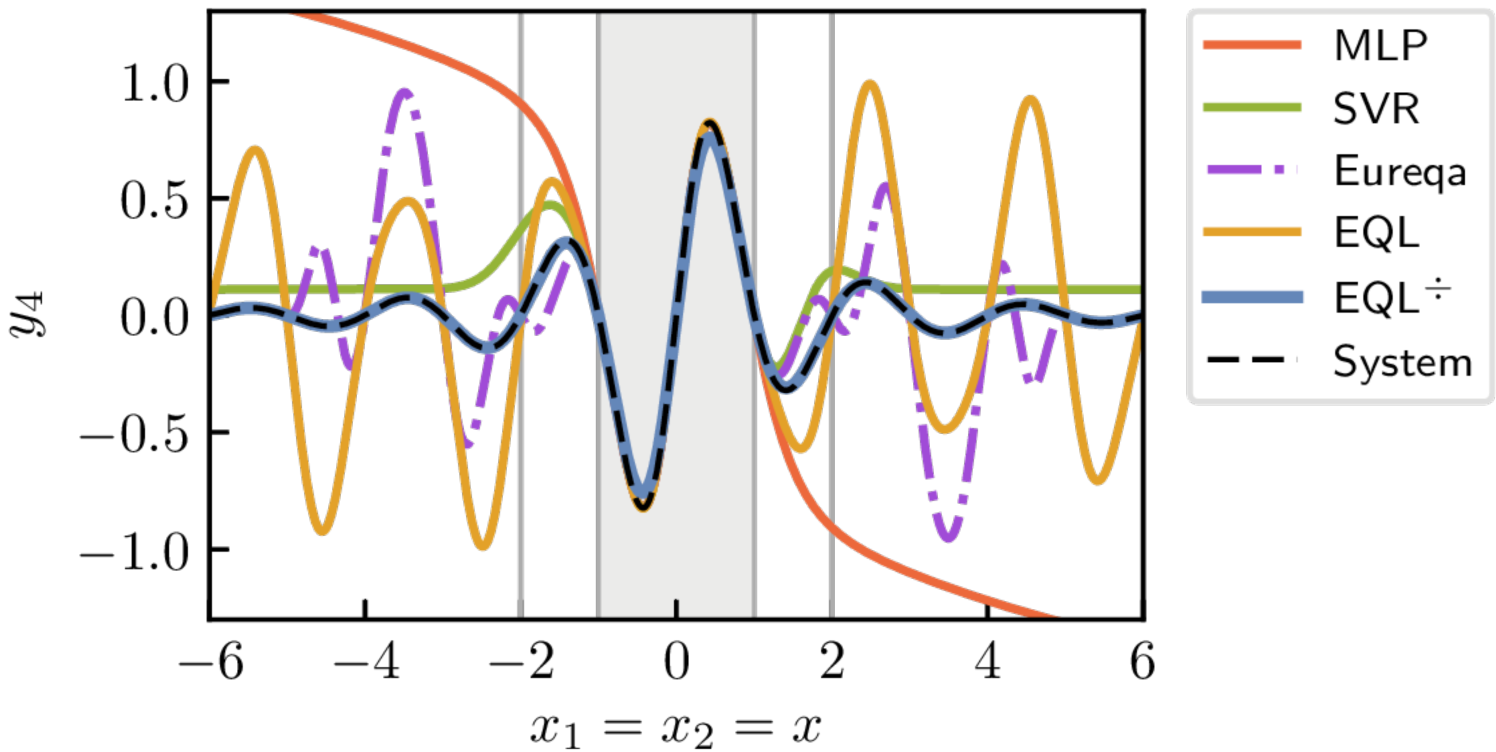


Learned formula: $-0.33\sin(-3.13x_1) + 0.33\sin(6.28x_2 + 0.39) + 0.33x_2 - 0.056 - 0.33x_3x_4$

Sahoo, Lampert, Martius, ICML 2018

Function learning and extrapolation

Toy example 2: $$y = \frac{\sin(\pi x_1)}{(x_2^2 + 1)}$$



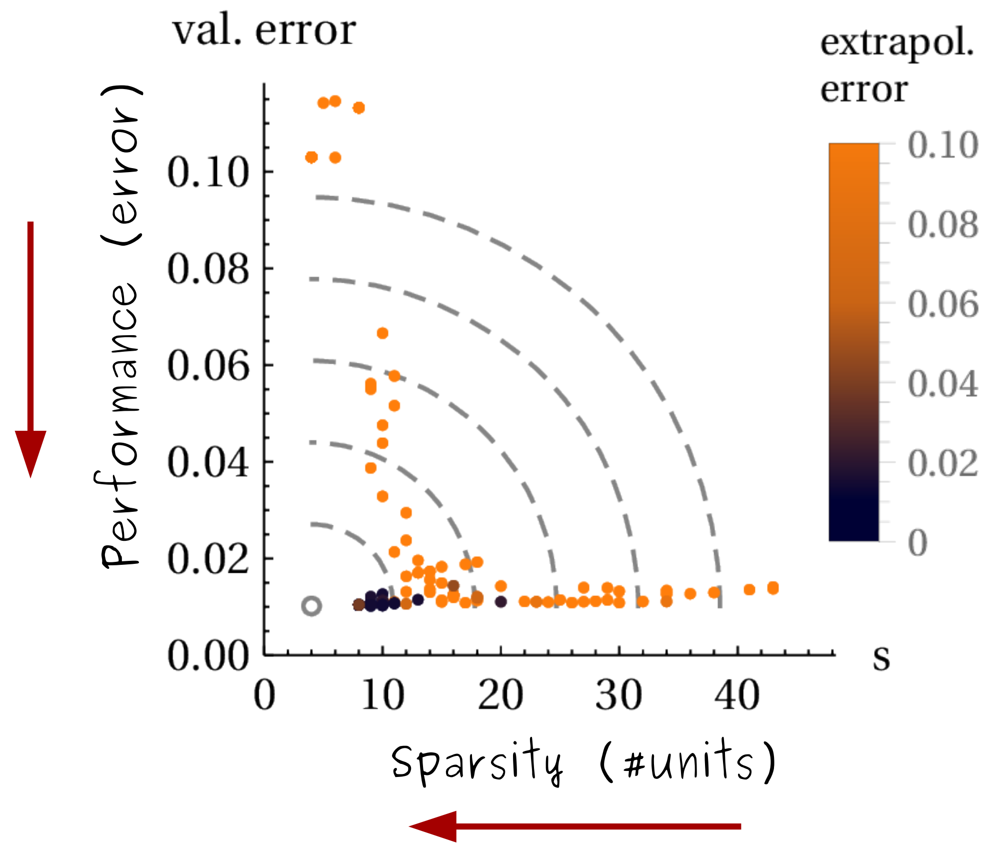
Sahoo, Lampert, Martius, ICML 2018

Model Selection

Occams Razor: Most simple formula is most likely the right one.

But too simple can also be wrong!

Multiobjective: Simple and good performance



different from standard ML!

Source of Interpretability

$$\arg\min_{\phi}[\tilde{v}(\phi)^2 + \tilde{s}(\phi)^2]$$

normalized values

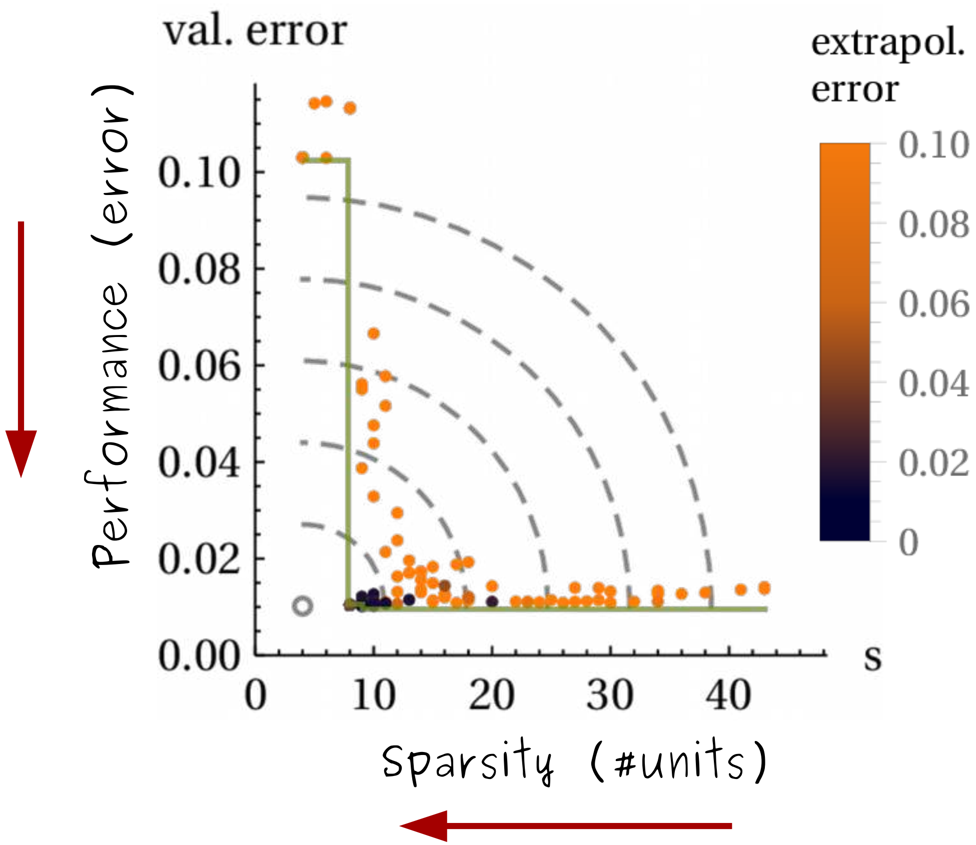
Sahoo, Lampert, Martius, ICML 2018

Model Selection

Occams Razor: Most simple formula is most likely the right one.

But too simple can also be wrong!

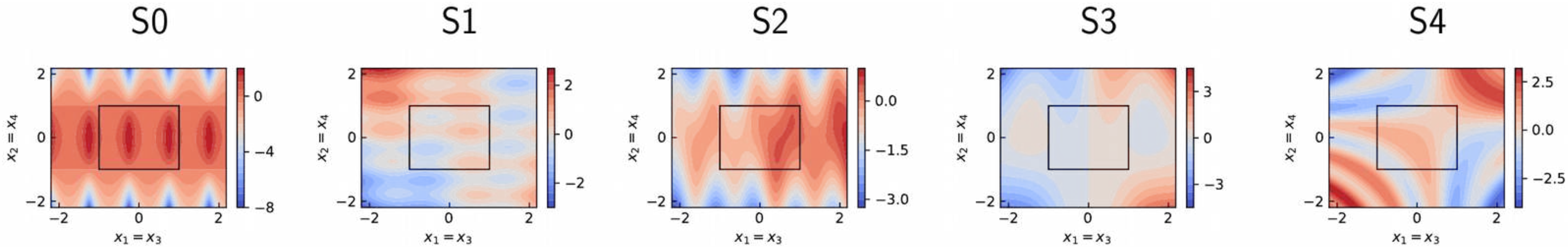
Multiobjective: Simple and good performance

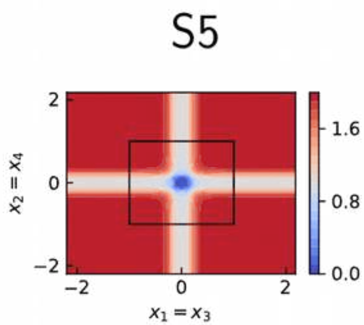


Remark:
jump in Pareto front not always as obvious

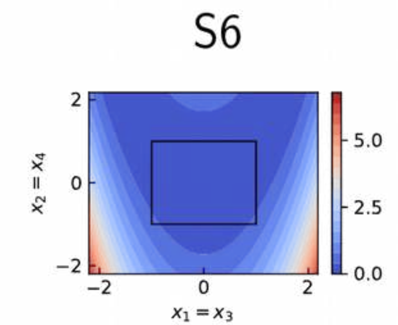
Sahoo, Lampert, Martius, ICML 2018

Some Results Analytical Functions





	S0	S1	S2	S3	S4	S5	S6
MP	1.57	0.66	0.75	0.64	1.17	0.44	0.82
MLP	1.11 (1.17 / 1.04)	0.45 (0.54 / 0.35)	0.41 (0.42 / 0.39)	0.36 (0.37 / 0.34)	0.65 (0.70 / 0.57)	0.07 (0.15 / 0.02)	0.65 (0.66 / 0.65)
GA1	1.07 (1.58 / 0.91)	0.77 (0.77 / 0.77)	0.33 (0.35 / 0.01)	0.01 (0.47 / 0.01)	1.20 (1.23 / 1.07)	0.29 (0.29 / 0.29)	0.82 (0.82 / 0.82)
GA2	0.39 (1.47 / 0.01)	0.66 (0.77 / 0.01)	0.37 (0.77 / 0.01)	0.60 (1.78 / 0.58)	0.01 (0.01 / 0.01)	0.08 (0.42 / 0.01)	4.26 (inf / 0.74)
EQL÷	0.01 (0.12 / 0.01)	0.01 (0.17 / 0.01)	0.01 (0.05 / 0.01)	0.01 (0.01 / 0.01)	0.01 (1.38 / 0.01)	0.01 (0.01 / 0.01)	0.01 (0.02 / 0.01)



S0 $y = (1 - x_2^2)/(\sin(2\pi\, x_1) + 1.5)$

S1 $y = \left[\sin(\pi x_1) + \sin(2\pi x_2 + \pi/8) + x_2 - x_3 x_4\right]/3$

S2 $y = \left[\sin(\pi x_1) + x_2 \cos(2\pi x_1 + \pi/4) + x_3 - x_4^2\right]/3$

S3 $y = \left[(1 + x_2)\sin(\pi x_1) + x_2 x_3 x_4\right]/3$

S4* $y = (3.0375\, x_1 x_2 + 5.5 \sin(9/4\,(x_1 - 2/3)(x_2 - 2/3)))/5$

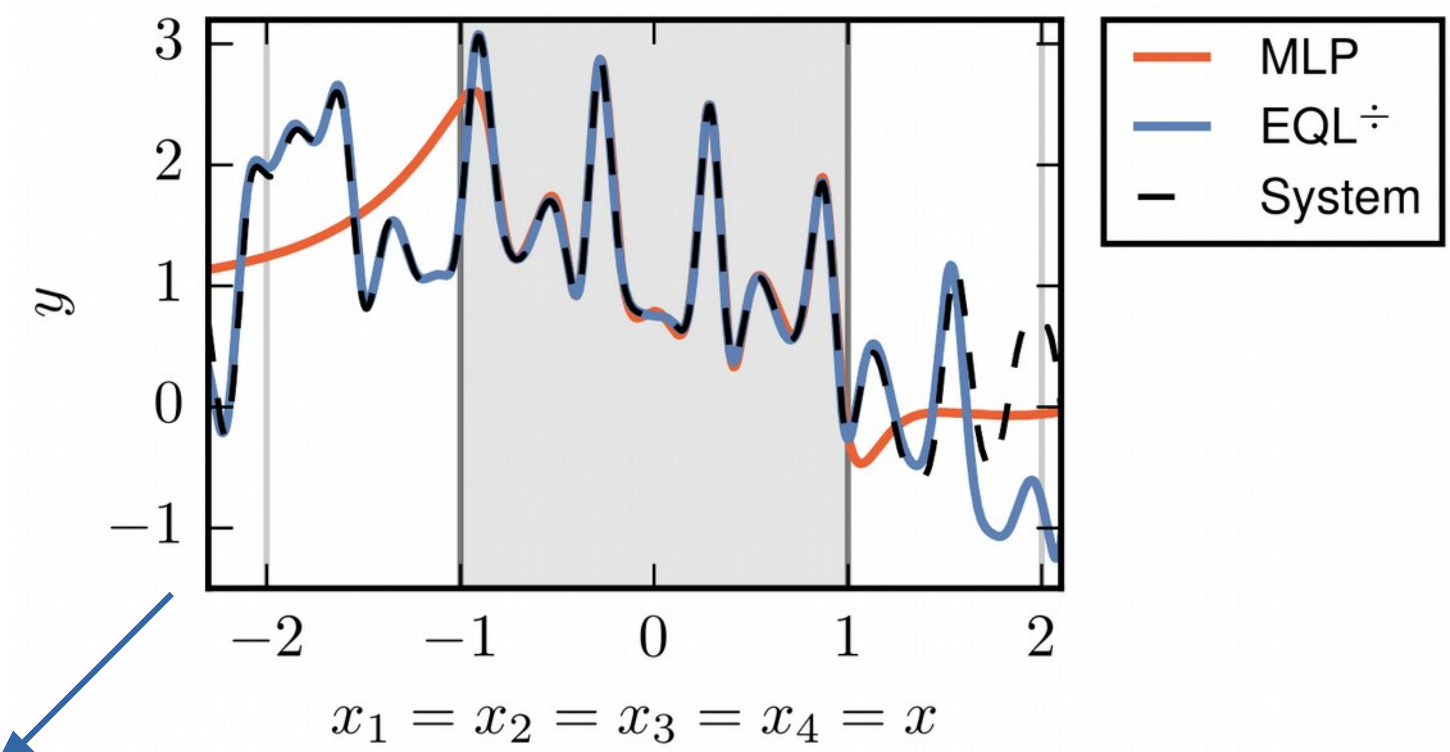
S5* $y = \frac{(5x_1)^4}{(5x_1)^4+1} + \frac{(5x_2)^4}{(5x_2)^4+1}$

S6* $y = ((1 - x_1)^2 + (1 - x_3)^2 + 100(x_2 - x_1^2)^2 + 100(x_4 - x_3^2)^2)/1500$

Werner, Junginger, Hennig, GM. arXiv:2105.06331, 2021

Random formulas

random formula
(RE2-2)



	RE2-1	RE2-2	RE2-3 ✗	RE2-4	RE3-1 ✗	RE3-2	RE3-3	RE3-4
EQL$^{\div}$ $V^{\text{int\&ex}}$	$0.02\,^{0.02}_{0.02}$	$0.04\,^{0.11}_{0.03}$	$0.52\,^{0.82}_{0.48}$	$0.01\,^{0.01}_{0.01}$	$0.46\,^{0.55}_{0.28}$	$0.02\,^{0.06}_{0.01}$	$0.01\,^{0.01}_{0.01}$	$0.03\,^{0.52}_{0.02}$
EQL$^{\div}$ V^{int}-S	$0.27\,^{0.39}_{0.02}$	$0.14\,^{0.14}_{0.14}$	$0.76\,^{2.05}_{0.55}$	$0.01\,^{0.01}_{0.01}$	$0.51\,^{1.23}_{0.31}$	$0.08\,^{4.65}_{0.04}$	$0.01\,^{0.01}_{0.01}$	$0.03\,^{1.64}_{0.02}$
MLP $V^{\text{int\&ex}}$	$1.54\,^{1.66}_{1.43}$	$1.04\,^{1.09}_{0.96}$	$0.90\,^{0.91}_{0.87}$	$0.95\,^{1.12}_{0.86}$	$1.04\,^{1.36}_{0.84}$	$1.85\,^{2.13}_{1.60}$	$0.52\,^{0.58}_{0.40}$	$1.64\,^{1.96}_{1.34}$
MLP V^{int}	$1.60\,^{1.66}_{1.44}$	$1.05\,^{1.10}_{1.01}$	$1.47\,^{1.65}_{1.10}$	$0.99\,^{1.16}_{0.86}$	$1.31\,^{1.59}_{1.07}$	$2.03\,^{2.24}_{1.65}$	$1.16\,^{2.02}_{0.73}$	$1.89\,^{2.12}_{1.61}$
SVR $V^{\text{int\&ex}}$	1.15	1.09	0.59	1.51	0.96	1.81	0.37	1.23
SVR V^{int}	1.20	2.12	17.72	13.89	11.79	11.28	0.37	17.67
Const 0	6.73	2.57	0.50	5.36	1.65	72.26	17.67	3.15

Domain knowledge

- which terms are more likely
  e.g.: cos > x^2, div > exp
- what about exp, sqrt, log?
  needs treatment for gradient optimization
- which combinations are not allowed
  e.g.: cos(cos(.)), exp(exp(.))
- preinitialize network (unexplored)

Informed EQL

Werner, Junginger, Hennig, GM. arXiv:2105.06331, 2021

Domain knowledge

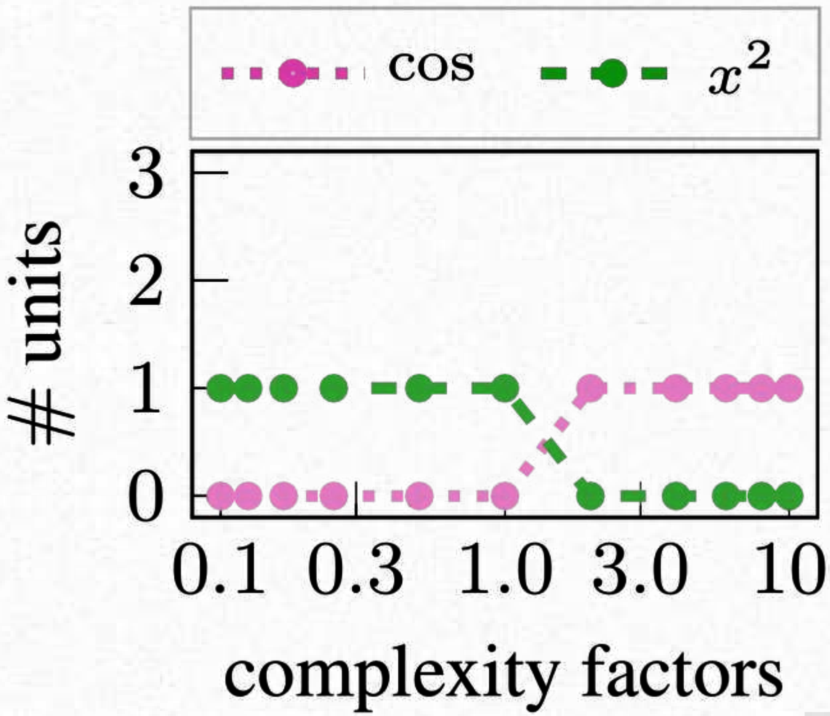
- which terms are more likely

  define complexity per base function: e.g. $w_{cos} = 3$, $w_{*} = 1$

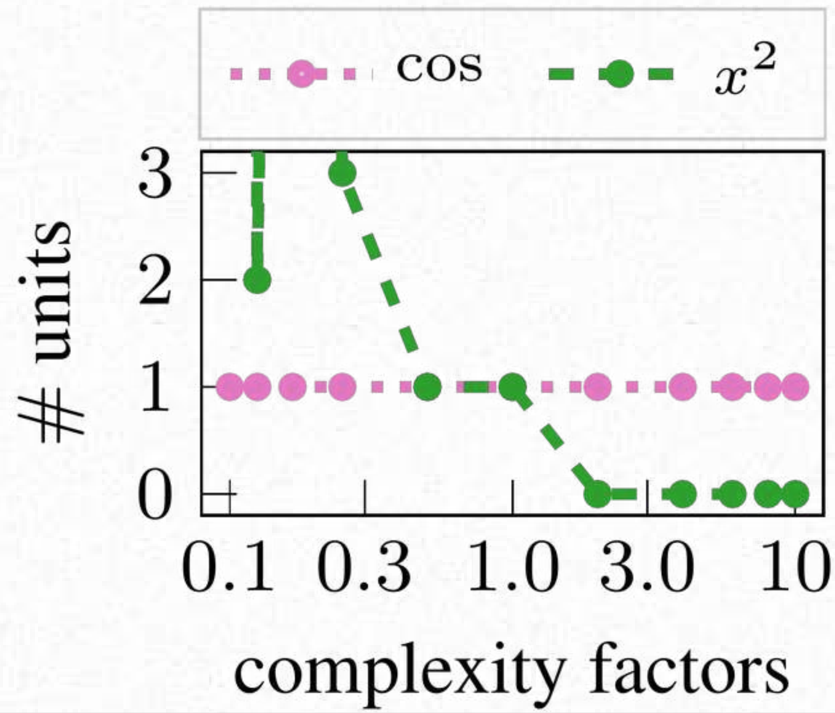
  - weight regularization with w_x
  - use w_x in complexity for model selection

- Example: $y = 8\cos(0.5x) - 4$

range: [−1,1]



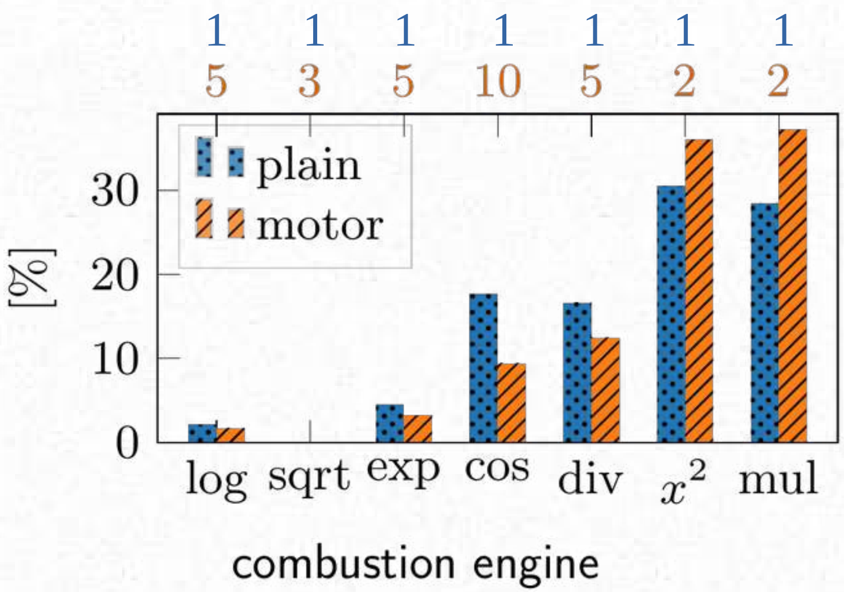
range: [−3,3]

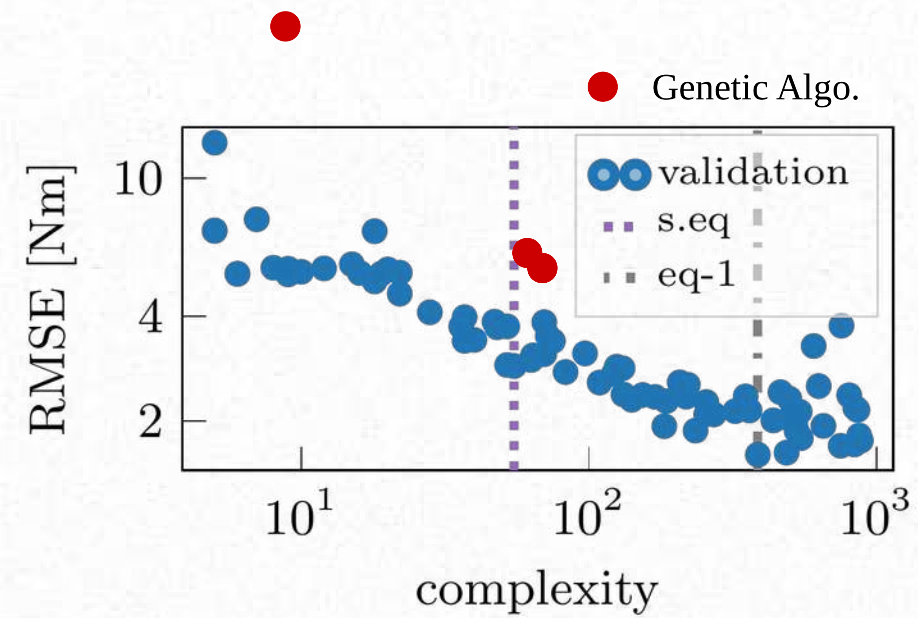


Werner, Junginger, Hennig, GM. arXiv:2105.06331, 2021

Domain knowledge

- example: combustion engine torque prediction



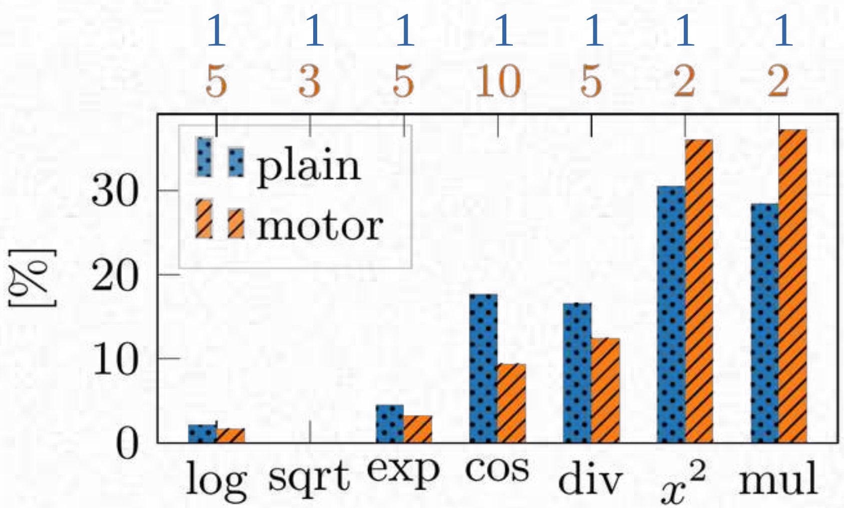


✓ Can influence type of equations

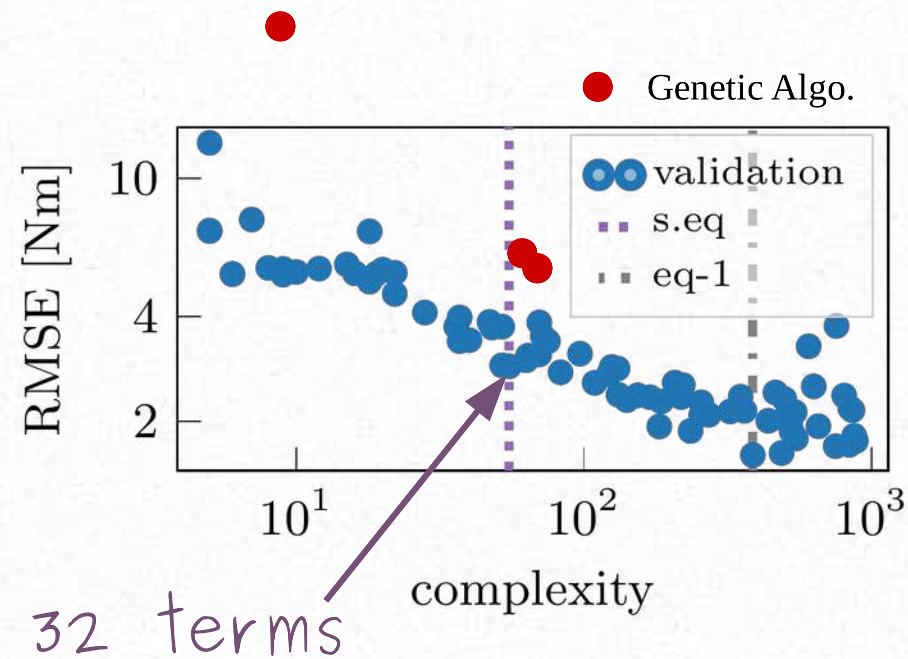
Werner, Junginger, Hennig, GM. arXiv:2105.06331, 2021

Domain knowledge

- example: combustion engine torque prediction



combustion engine



$$y = -0.3x_1 + 2.32x_2 + 0.27x_3 + 0.6c_2 + 0.3c_3 + 0.35\cos(1.12x_3 + 1.42c_2 + 0.59c_3 - 0.23) + 0.51$$

substitutions:

$$c_1 = (0.05 - 0.69x_1)(1.0x_1 + 0.64)$$

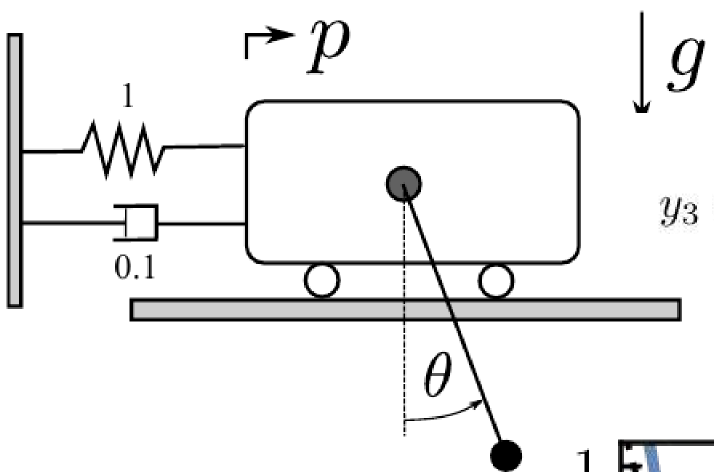
$$c_2 = (-0.95x_2 - 0.69)\left(-1.06x_3 + 0.96(-0.21x_1 + 0.66x_3\right.$$

$$c_3 = \left(-0.74x_3 - 0.76\left(0.6x_1 + 0.73c_1 + 0.47(0.76x_2 + 0.19x_4 - 0.19x_5 - 0.6)^2 - 0.9\right)^2 + 1.36\right)$$
$$(2.0x_3 + 0.88(-0.73x_1 - 1.17c_1 + 0.26)(0.6x_4 - 0.72x_5 - 0.46) - 0.31)$$

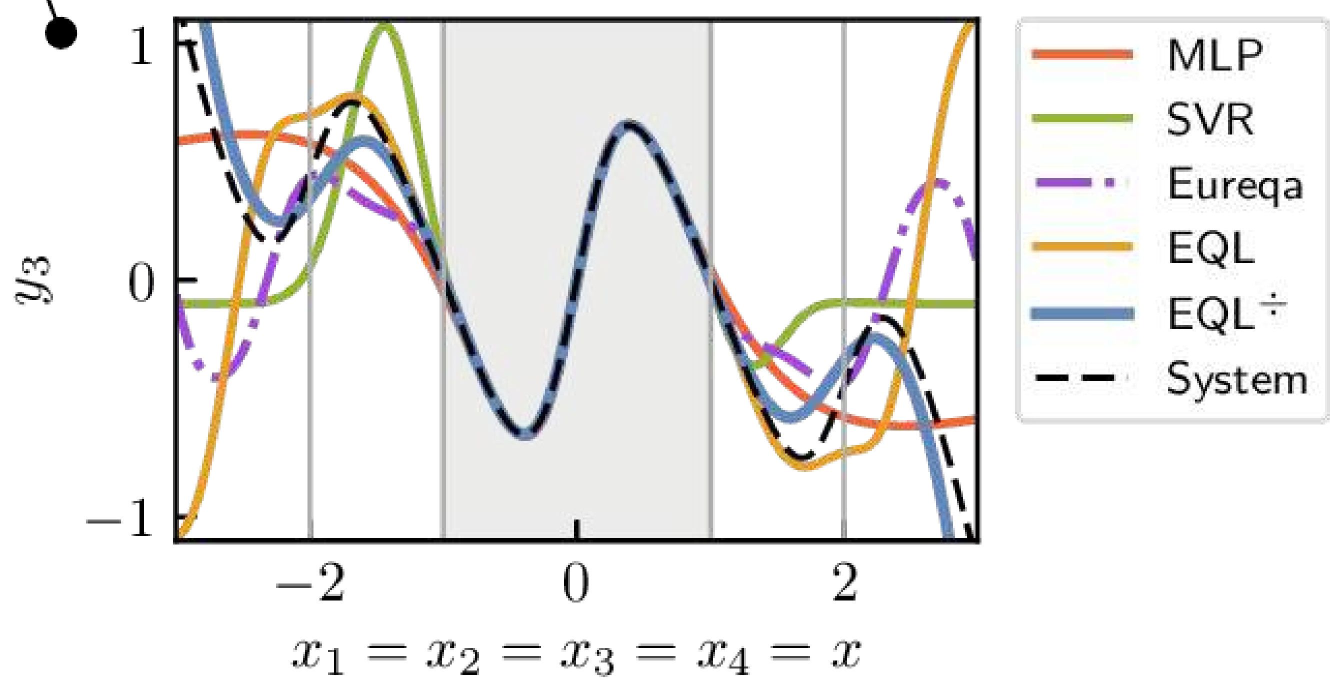
Understanding?
depends...

Werner, Junginger, Hennig, GM. arXiv:2105.06331, 2021

Application: Cart-Pendulum dynamics



$$y_3 = \frac{-x_1 - 0.01x_3 + x_4^2 \sin(x_2) + 0.1x_4 \cos(x_2) + 9.81 \sin(x_2)\cos(x_2)}{\sin^2(x_2) + 1},$$

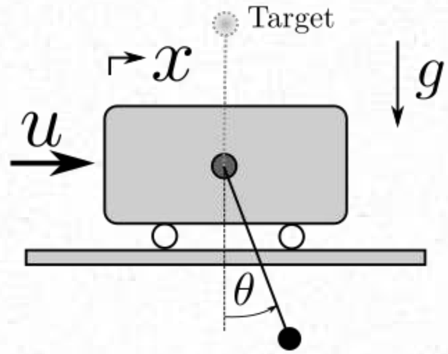


Able to learn dynamics equations

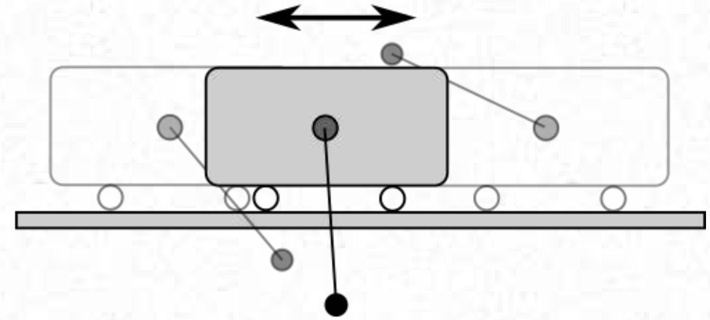
Sahoo, Lampert, Martius, ICML 2018

Learning Cart-Pole swingup

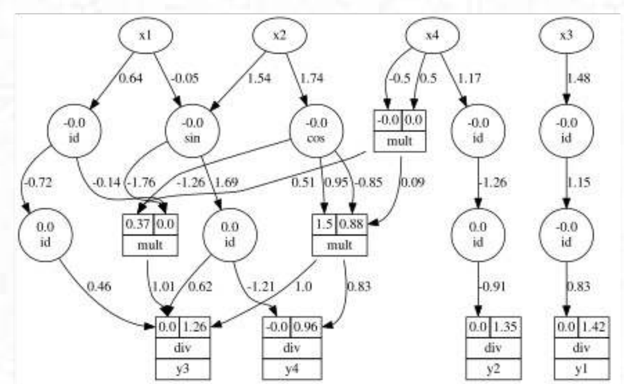
Robot



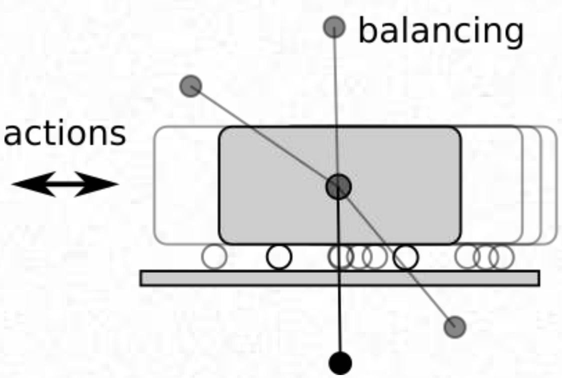
random movements



Learning Equation



Control

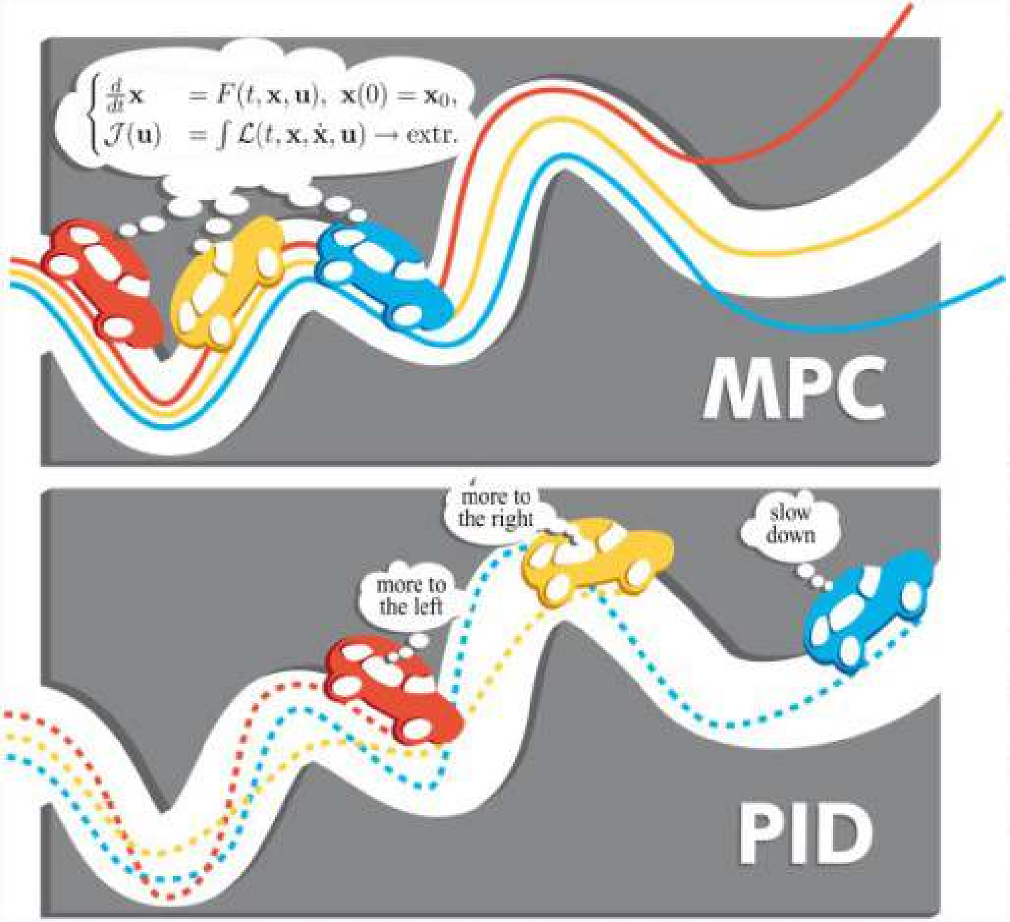


Model predictive Control,
random shooting method

Sahoo, Lampert, Martius, ICML 2018

Model Predictive Control

- plan ahead with model
- take best action
- replan



(openi.nlm.nih.gov)

here: planning = many random rollouts

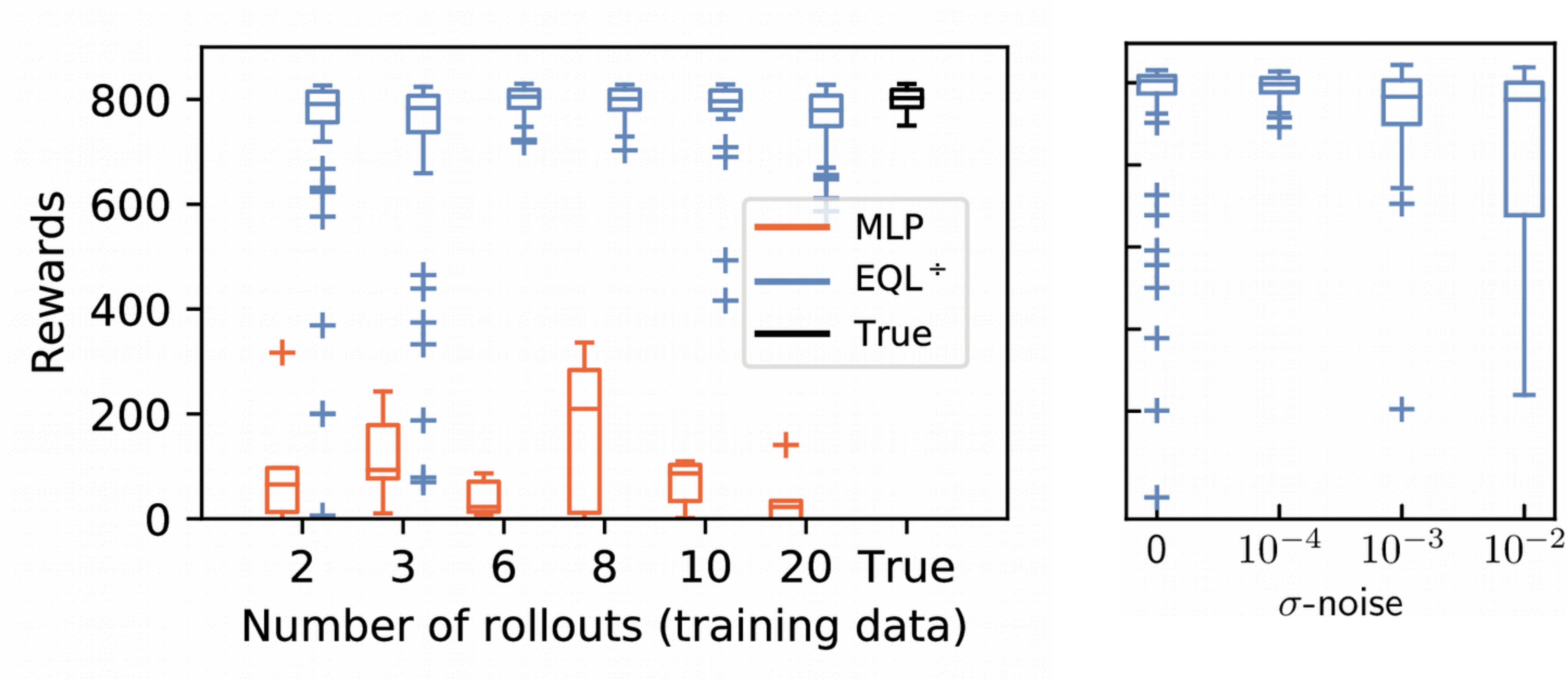
Learning Equations for Extrapolation and Control

by S.S.Sahoo, C.H.Lampert and G.Martius, ICML 2018

Training
1 Random rollout

Validation
1 Random rollout
(stronger actions)

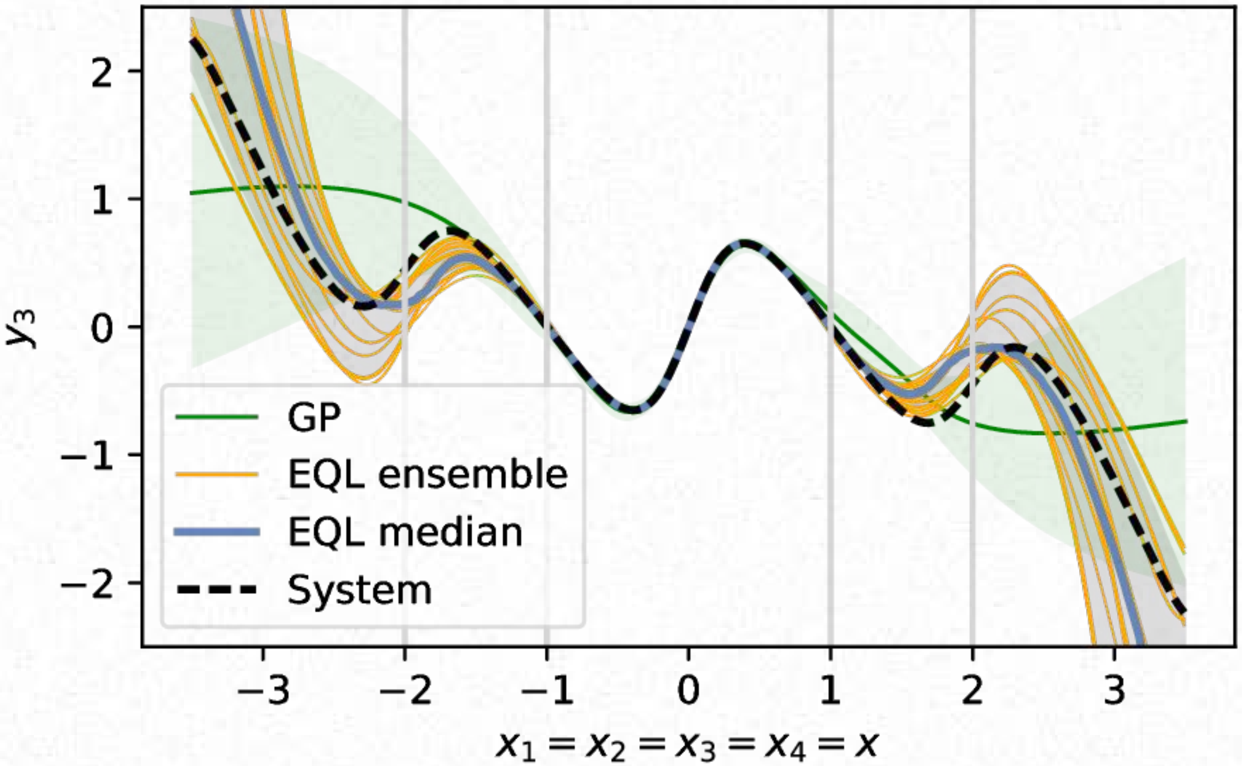
Cart-pole Swingup



Sahoo, Lampert, Martius, ICML 2018

Uncertainty estimates

- get an estimate of uncertainty
  - ensemble of discrete hypotheses

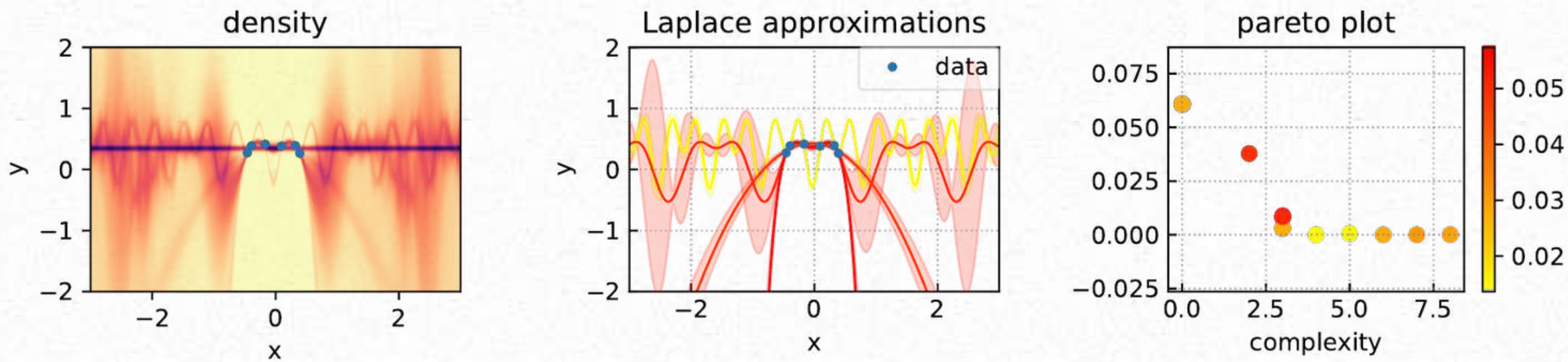


Matthias Werner

work in progress with Matthias Werner

Uncertainty estimates

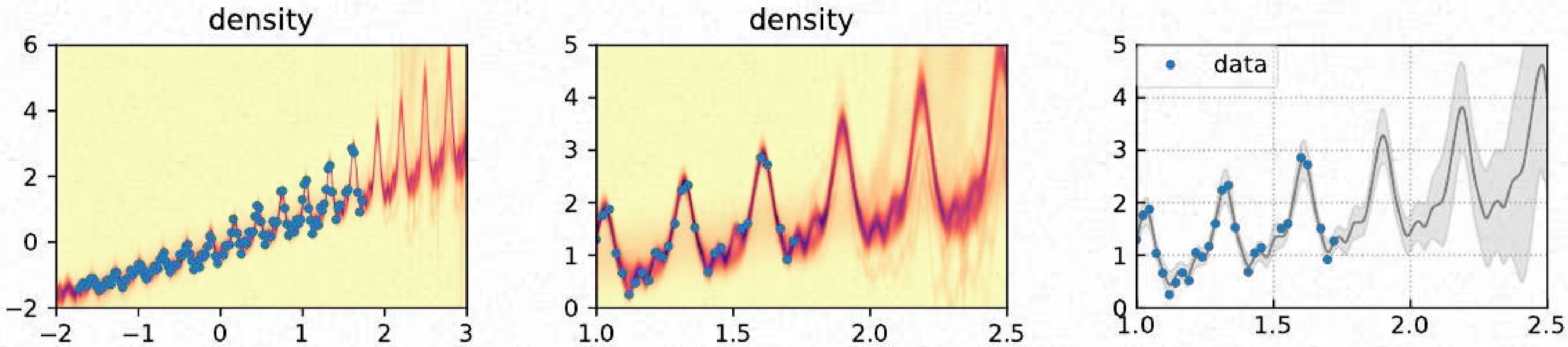
- get an estimate of uncertainty
  - ensemble of discrete hypotheses
  - Laplace approximation of each solution
    (how certain are the parameters estimated)



work in progress with Matthias Werner

Uncertainty estimates

- get an estimate of uncertainty
  - ensemble of discrete hypotheses
  - Laplace approximation of each solution
    (how certain are the parameters estimated)



(b) Airline dataset

work in progress with Matthias Werner

Other approaches:

- AI Feynman
  Udrescu & Tegmark, Science Advances 2020

  Uses physics knowledge

  - physical units
  - symmetries
- DSL (Deep Symbolic Regression)
  Petersen et al, ICLR 2021

  Uses Deep RL to guide the search
- Deterministic search: Prio. Grammer Enumeration
  Worm, Chiu. GECCO 2013, Kronberger et.al. 2018+

Amortizing Data!

- so far: regression/search starts from scratch
- can we train a network to produce good candidate guesses, based on data?

Neural Symbolic Regression that Scales @ICML 2021

Luca Biggio [*12] **Tommaso Bendinelli** [*2] **Alexander Neitz** [3] **Aurelien Lucchi** [1] **Giambattista Parascandolo** [13]

NeSymReS

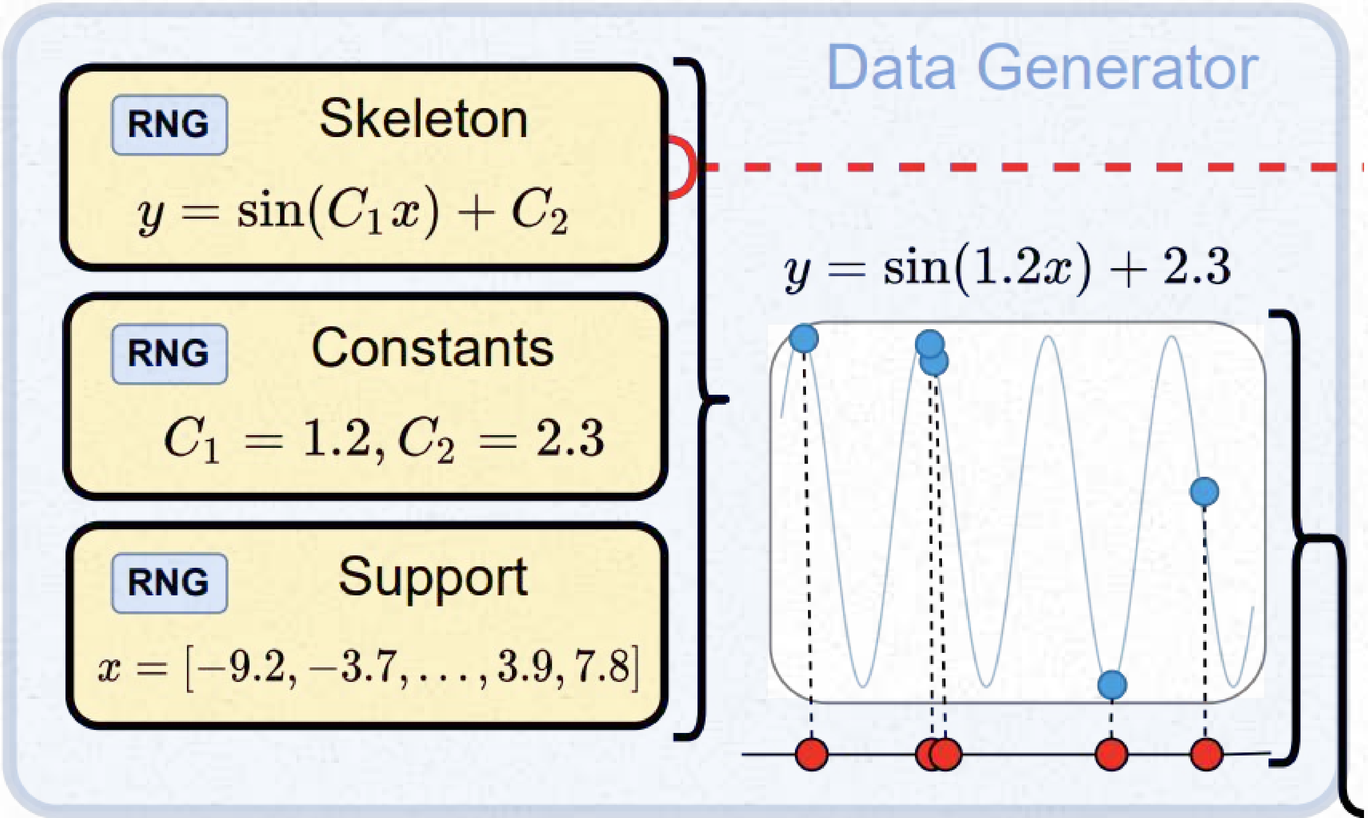
NeSymReS

Approach:

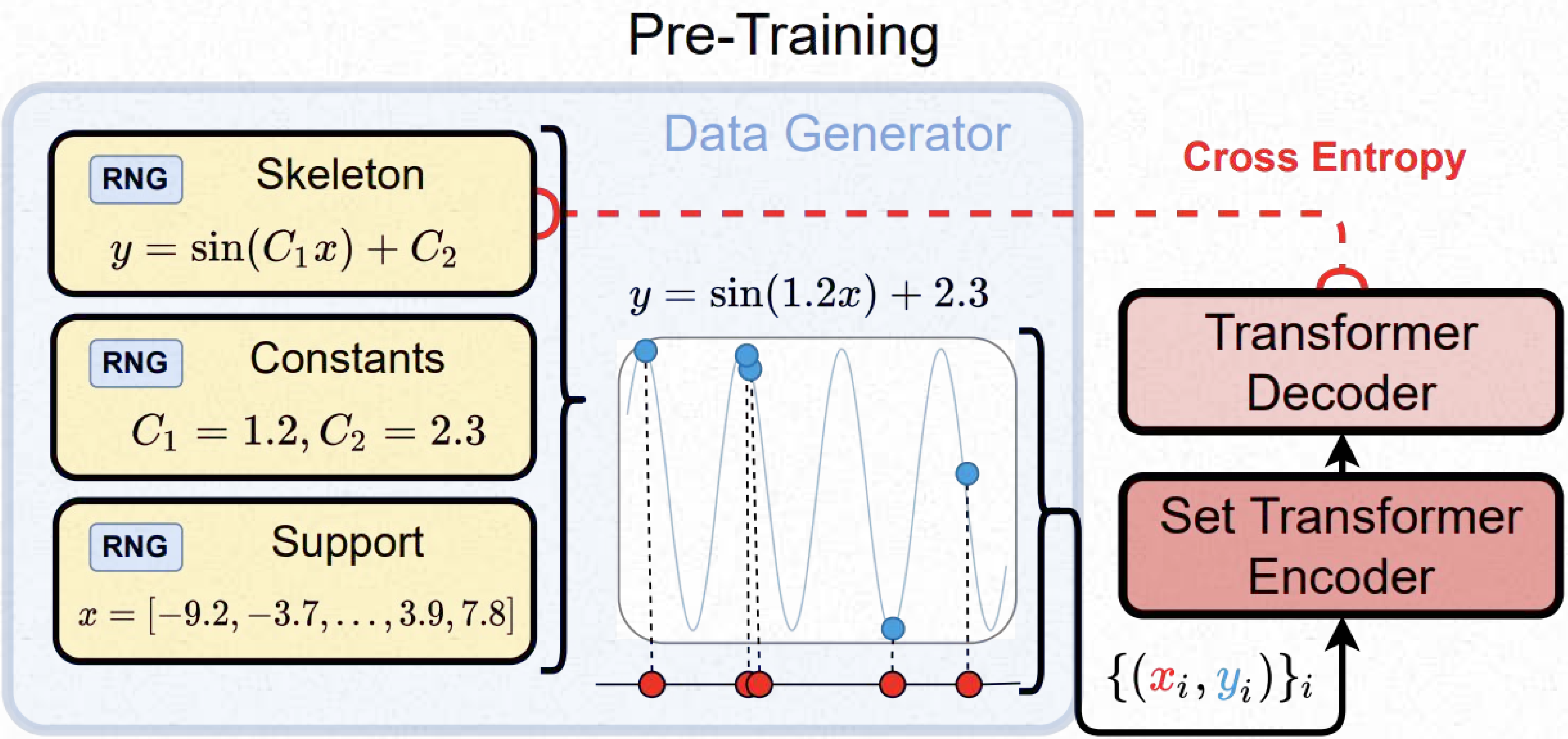
- Generate massive amounts data
- Pre-train end-to-end a big transformer to do exactly what we want to do
- Sample from Transformer
  - Apply beam-search
  - Fine-tune

NeSymReS

Pre-Training

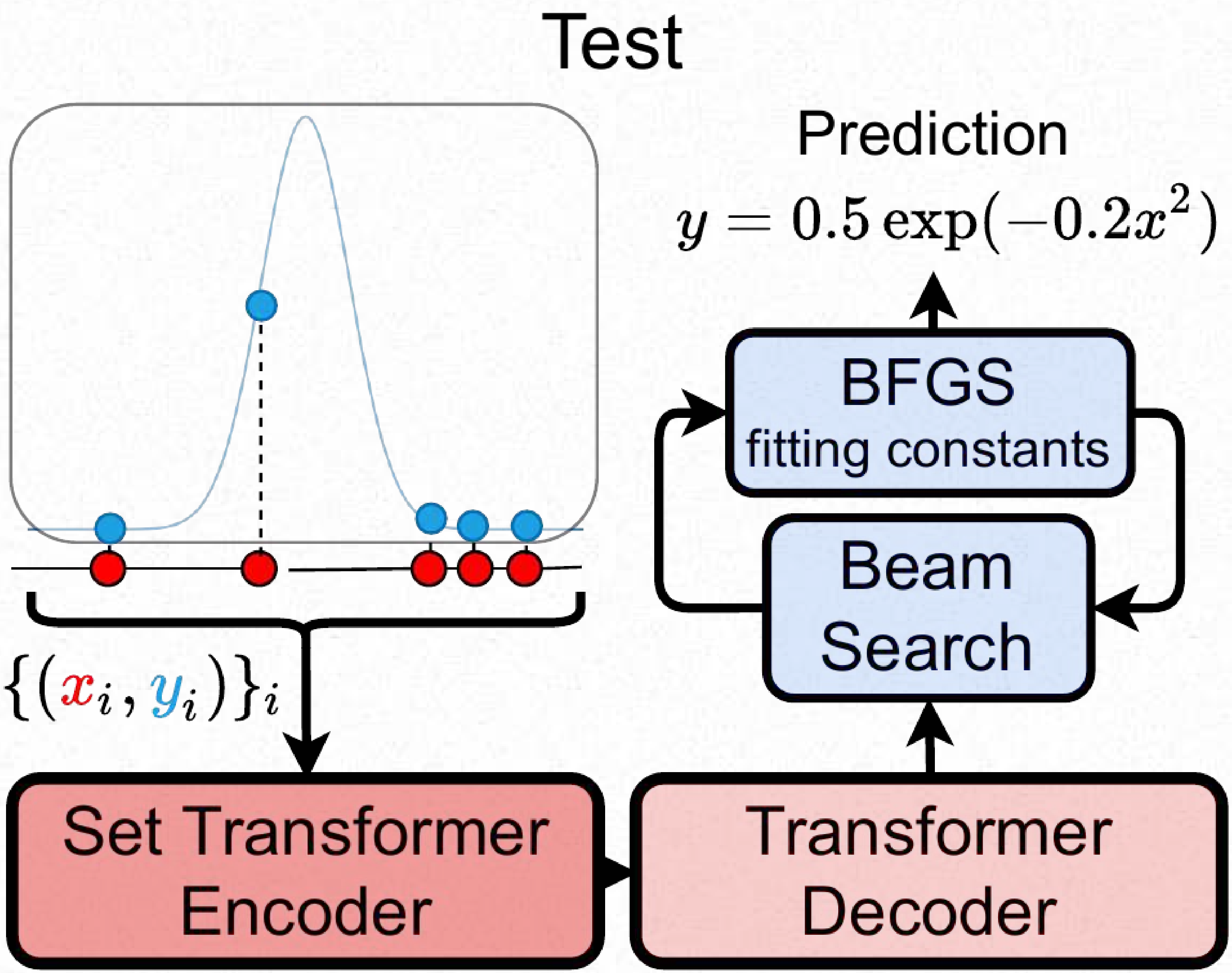


NeSymReS



learn to guess the right functional form
without constants

NeSymReS

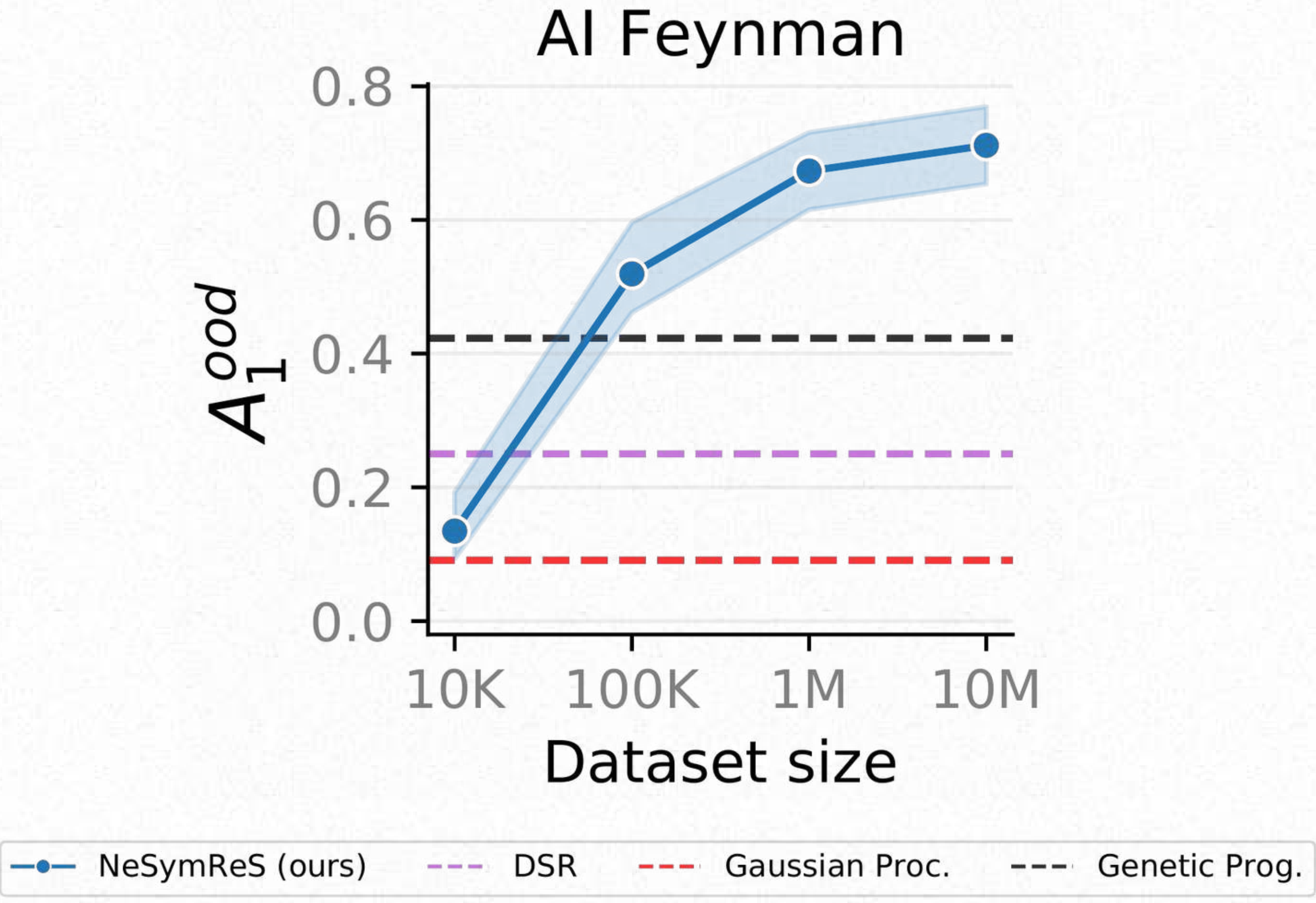

Test
Prediction
$y = 0.5\exp(-0.2x^2)$
BFGS
fitting constants
Beam
Search
$\{(x_i, y_i)\}_i$
Set Transformer
Encoder
Transformer
Decoder

How much pretraining do we need?

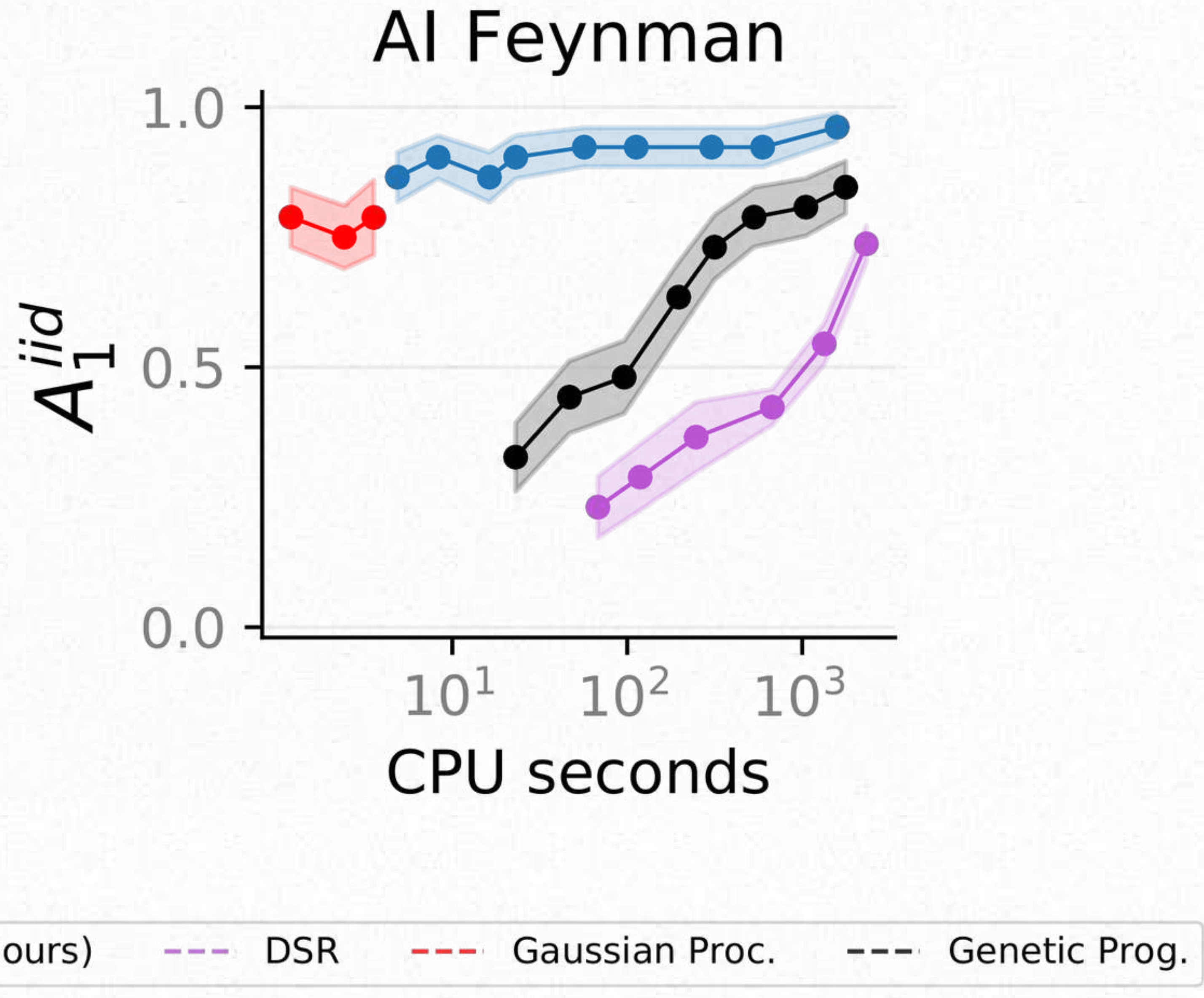
AI Feynman: Equations/Dataset

$\sqrt{-x_1 + \frac{x_2 x_3}{x_1}}$	$\frac{x_3 + \frac{3.797 \sin(x_1)}{x_2}}{x_3}$
$\sin(4.84x_3(2.3x_1 - 3.494x_2 + 1))$	$x_1 - 4.843x_2x_3 + x_2 + \cos(x_3)$
$\sin(x_3) + \sin\left(\frac{x_3}{x_1 - x_2}\right)$	$\cos\left(x_1 + 1.504x_2 + (x_2 + x_3)^2\right)$
$x_1(2.683x_1 + x_2\cos(x_3)) + 1$	$x_1\left(-4.641x_1 + \cos^2(4.959\sqrt{x_2})\right)$
$4.631\sin\left(4.419\sin\left(\frac{x_2x_3}{x_1^2}\right)\right)$	$x_2\left(x_2 - \frac{-x_1 - 1}{x_2}\right)$
$3.874x_3 + 4.12 - \frac{1}{x_1 + 4.322x_2x_3}$	$4.47x_1 + 1.193\cos\left(1 + \frac{1}{x_2}\right)$
$\frac{1.858x_1x_3}{-x_1 + x_2} - 3.661x_3$	$3.63x_1\cos(1.427x_2^3 + x_2)$
$2.846x_2 + \sin(x_1^5 + 2.258x_3)$	$-x_1(1.196x_1 + \sin(x_1 + x_2))$
$\frac{x_2 + x_3 + \frac{x_2 - 4.615}{x_2}}{x_1}$	$x_3 + \frac{x_3}{x_1 + 2.318x_2 + x_3}$
$-x_3 + \frac{0.221(-x_1 + x_2)}{\log(x_2)} - 1$	$x_1 - \frac{7.74\sqrt{0.383x_1 + x_2}}{x_2}$
$(1.261x_1 + 3.29\cos(1))\log(4.169x_2)$	$1 + \frac{0.221\tan(3.972x_2)}{x_2(-3.549x_1 + x_2)}$

How much pretraining do we need?



Inference speed



Comparison

	Easy to implement	Differentiable	Pro	Cons	Domain knowledge
Genetic Algorithm [1]	✓ ?	✗	easy to implement	constants are hard, only small equations	base-functions + complexity
AI Feynman [2]	✗	✗	finds very plausible eqns.	restricted to physics?	physics (units etc)
RL-based search (DSR) [3]	✗	✗	faster than GAs	constants are hard	base-functions + complexity
Transformer (NeSymReS) [4]	✗ / ✓	○	fast, accurate	max # variables little probl. data	pretraining, base-functions
Sparse regression of Library (SINDy) [5]	✓	○	fast large systems	no function composition	library of blocks
Deep Network EQL / iEQL [6,7,8]	○	✓	large systems (many DoF)	complicated expressions? slow	base-functions + complexity

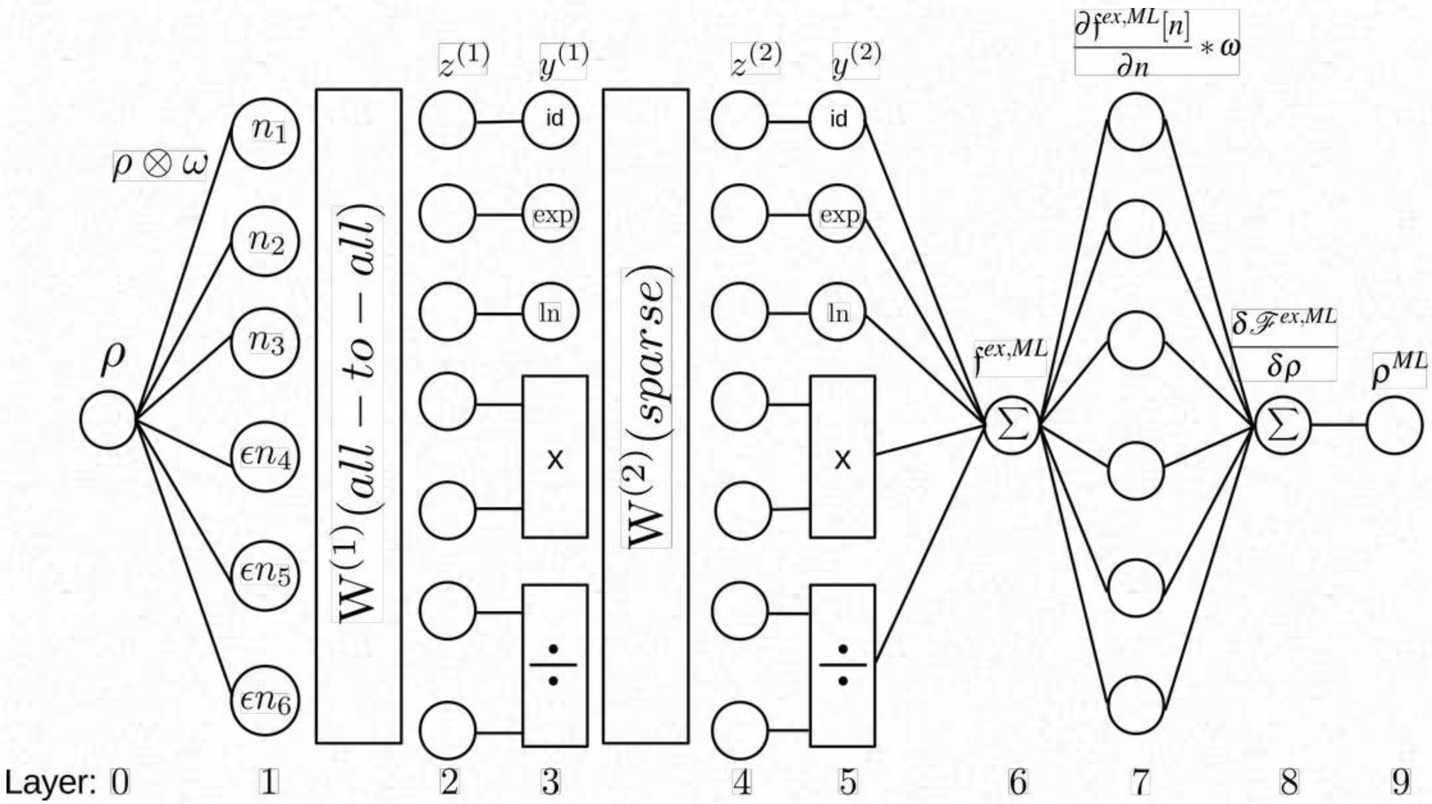
[1] Schmidt, Lipson. Distilling Free-Form Natural Laws from Experimental Data. Science, 2009
[2] Udrescu, Tegmark, AI Feynman: A physics-inspired method for symbolic regression, Science Advances, 2020
[3] Petersen et al. Deep symbolic regression: Recovering mathematical expressions from data via risk-seeking policy gradients. ICLR 2021
[4] Biggio et al. Neural Symbolic Regression that Scales. ArXiv 2106.06427, 2021
[5] Brunton, Proctor, Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. PNAS 2016
[6] Martius, Lampert, ArXiv 1610.02995, 2016
[7]Sahoo, Lampert, Martius. Learning Equations for Extrapolation and Control. ICML 2018
[8] Werner, Junginger, Hennig,Martius. Informed Equation Learning. ArXiv 2105.06331, 2021

Why is differentiability relevant

- allows to put the Symb. Regression module inside of deep architecture
- e.g.: Vision input → model object movement

Application to Physics

Find analytical expression of the classical free energy functional of simple fluids



Lin, **GM**, Oettel. JCP 2020

Joined project with Martin Oettel

- Promising direction for understanding relevant fluids

Application to Physics

Find analytical expression of the classical free energy functional of simple fluids

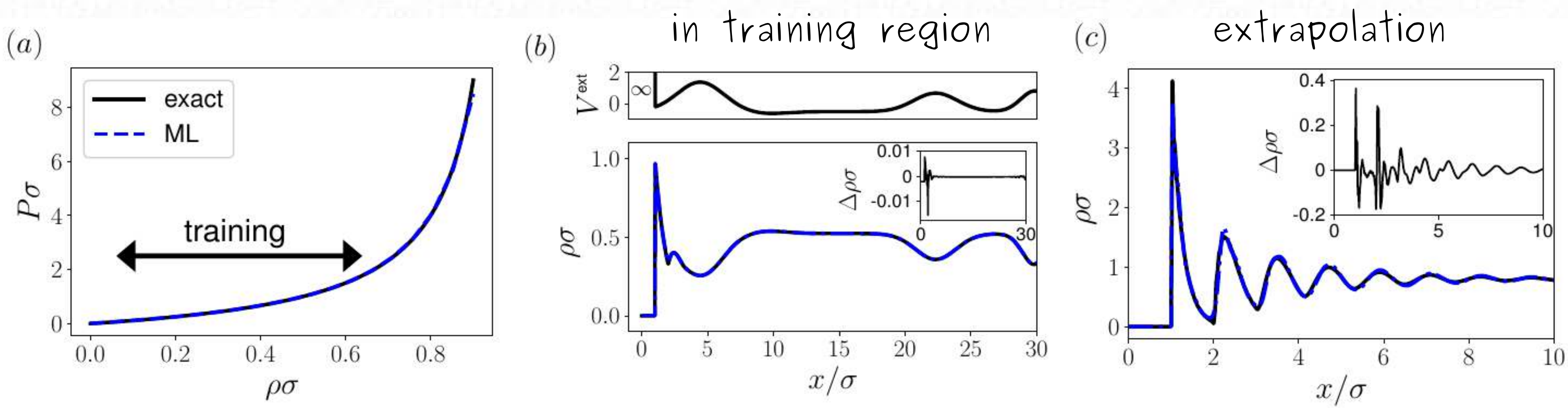


FIG. 2: FEQL results for hard rods. Dark solid lines are exact solutions from $\mathscr{F}^{HR}$ and blue dashed lines are ML results. (a) eos, $P(\rho)$. (b) density profile for $\rho_0 = 0.49$ inside the training region but V^{ext} not in the training data. (c) density profile at hard wall for $\rho_0 = 0.80$ outside the training region. Insets in (b) and (c) show $\Delta\rho = \rho^{exact} - \rho^{ML}$.

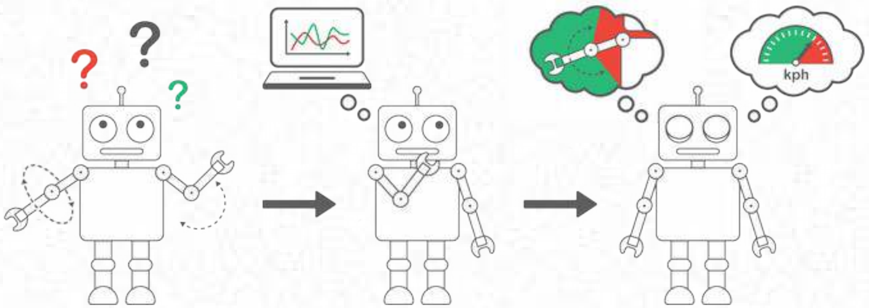
It works

Joined project with Martin Oettel

- Promising direction for understanding relevant fluids

Lin, **GM**, Oettel. *JCP* 2020

Summary



- Symbolic regression
  - find smallest fitting equation/formula
  - mimics model discovery
  - with domain knowledge can lead to great out-of-distribution generalization
- Many methods by now:
  - discrete search (GA's, DSL...)
  - differentiable methods (EQL)
  - neurally guided search (NeSymReS)