Multiview Symbolic Regression

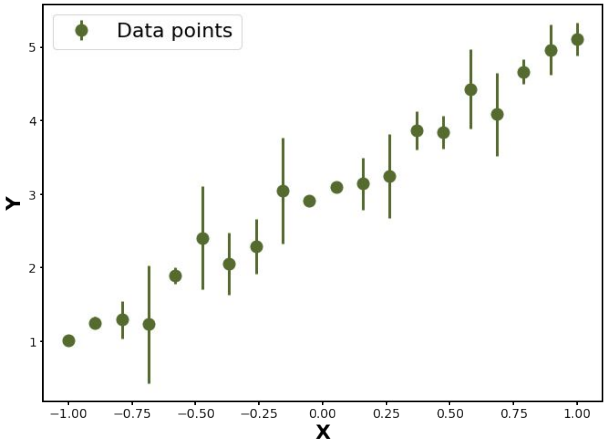
Physical applications

Etienne Russeil - *LPC Université Clermont Auvergne, France*
Emille Ishida - *LPC Université Clermont Auvergne, France*
Konstantin Malanchev - *University of Illinois Urbana–Champaign, USA*
Emmanuel Gangler - *LPC Université Clermont Auvergne, France*
Fabricio Olivetti - *CMCC Federal University of ABC, Brazil*

Traditional Symbolic Regression

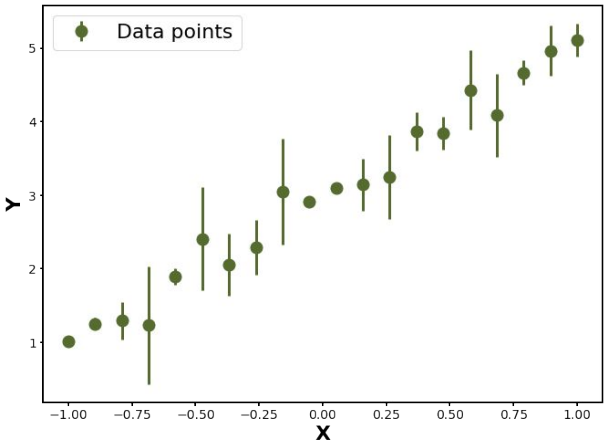
Traditional Symbolic Regression

DATA SET



Traditional Symbolic Regression

DATA SET



RANDOM EQUATIONS

$f(X) = \sin(X) + 2$

$f(X) = X^2 - 1$

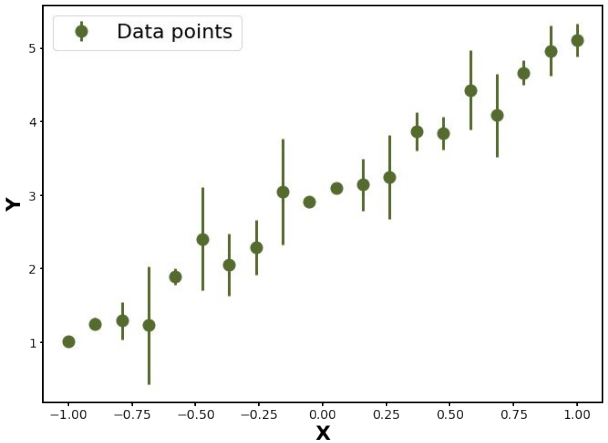
$f(X) = 42$

$f(X) = -4X + 8$

$f(X) = 2X$

Traditional Symbolic Regression

DATA SET



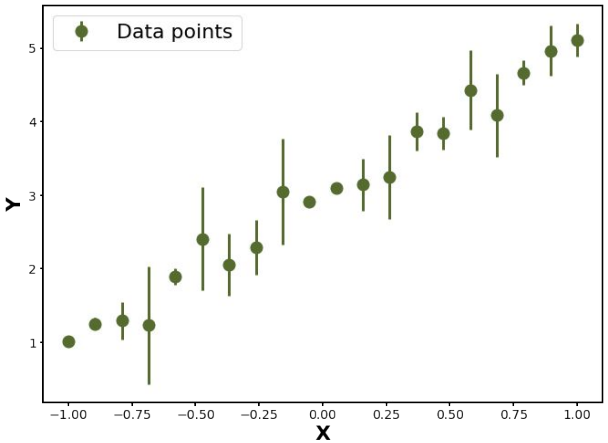
RANDOM EQUATIONS

COST FUNCTION

$f(X) = \sin(X) + 2$	COST = 12
$f(X) = X^2 - 1$	COST = 24
$f(X) = 42$	COST = 43
$f(X) = -4X + 8$	COST = 7
$f(X) = 2X$	COST = 3

Traditional Symbolic Regression

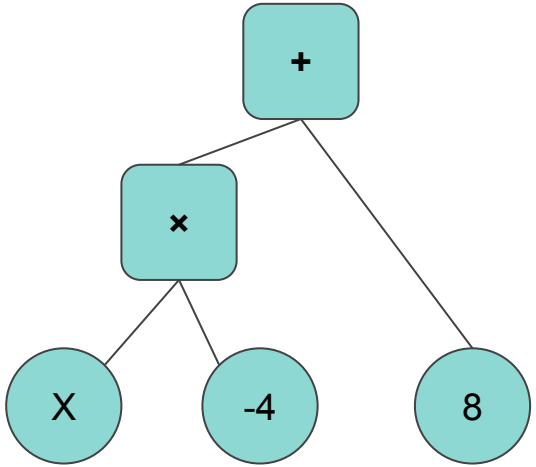
DATA SET

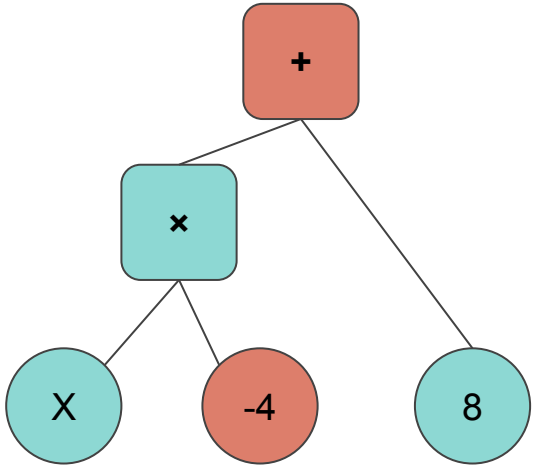


RANDOM EQUATIONS

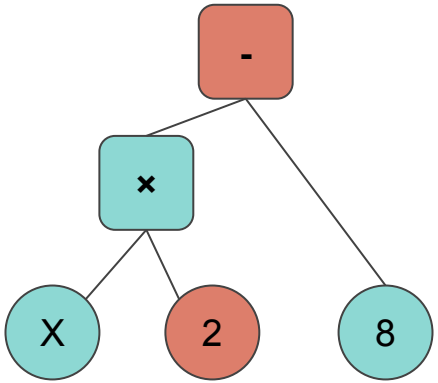
COST FUNCTION

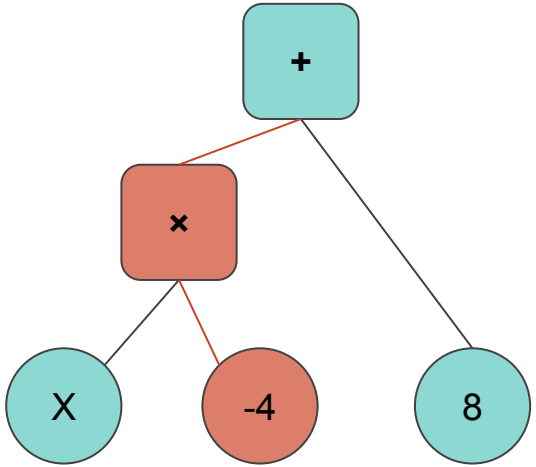
f(X) = sin(X) + 2	COST = 12
f(X) = X² - 1	COST = 24
f(X) = 42	COST = 43
f(X) = -4X + 8	COST = 7
f(X) = 2X	COST = 3


+
×
X
-4
8

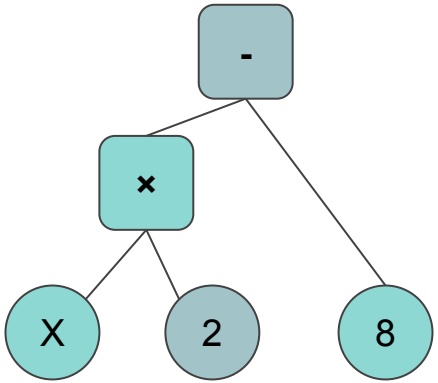


Point mutations

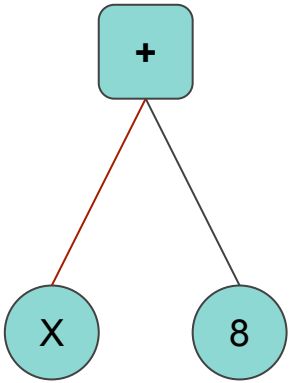


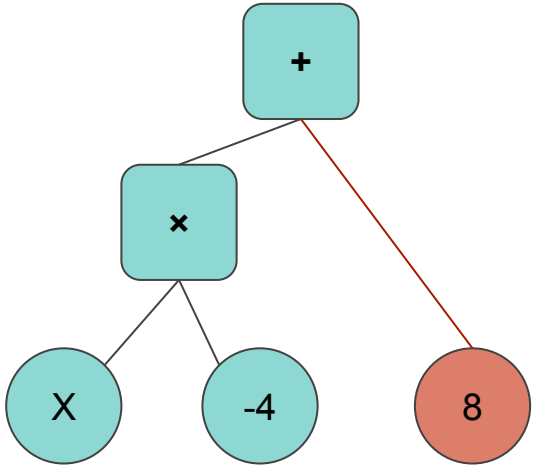


Point mutations

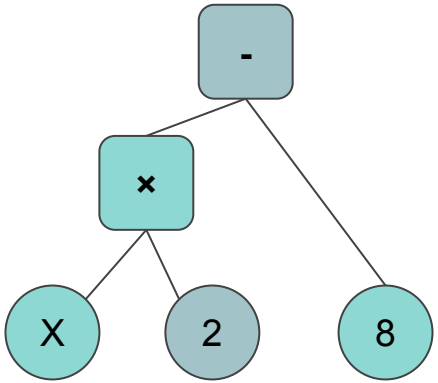


Hoist mutations

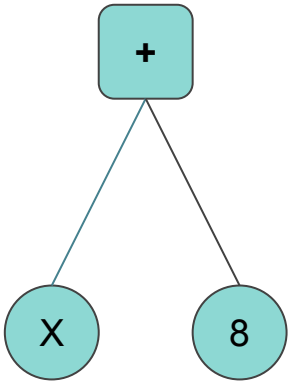




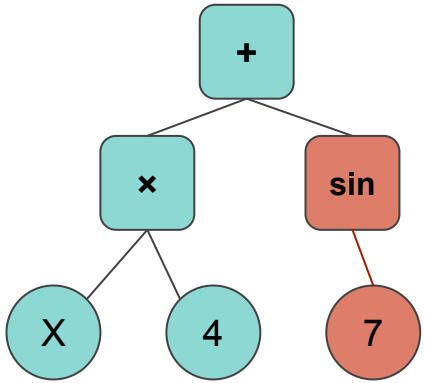
Point mutations

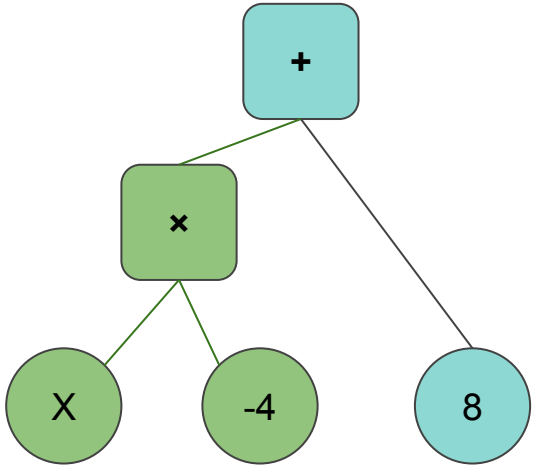


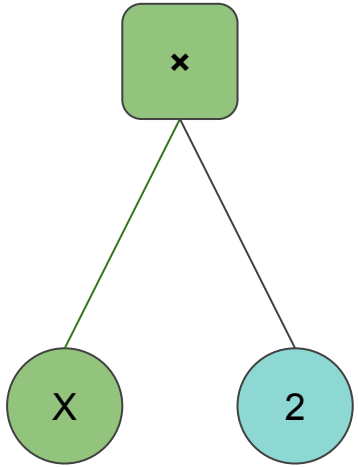
Hoist mutations



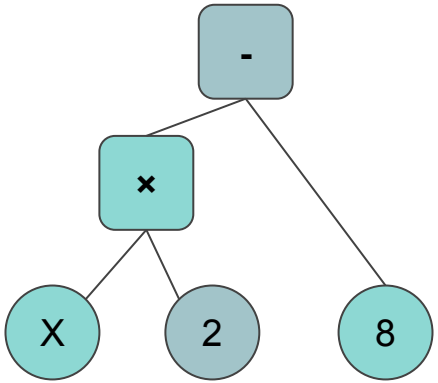
Subtree mutations



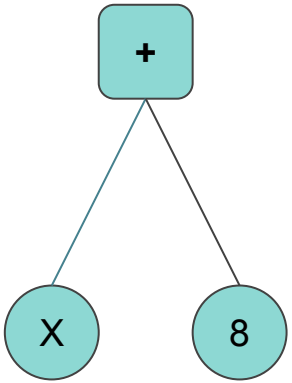




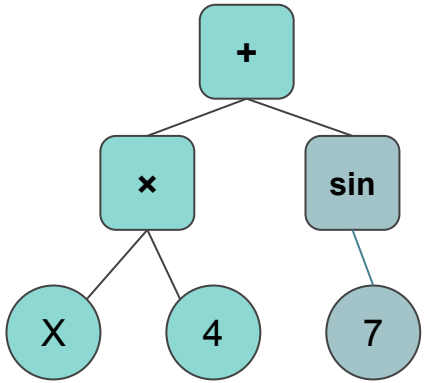
Point mutations



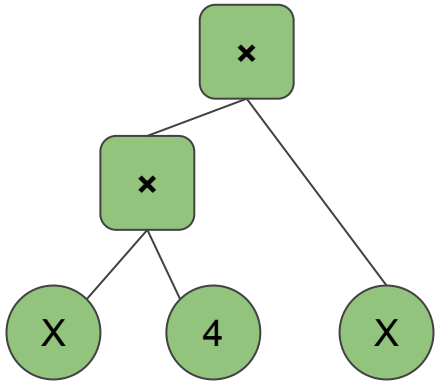
Hoist mutations



Subtree mutations

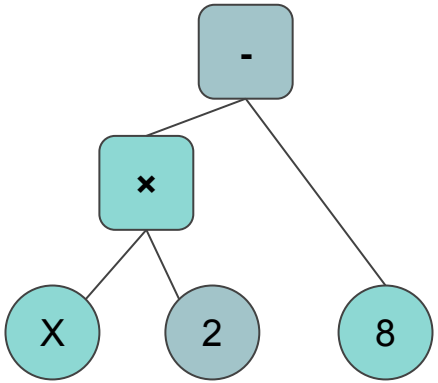


Crossover

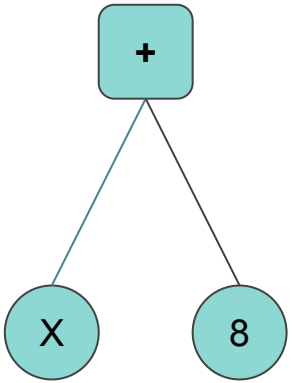


Create a new population from the previous best candidates

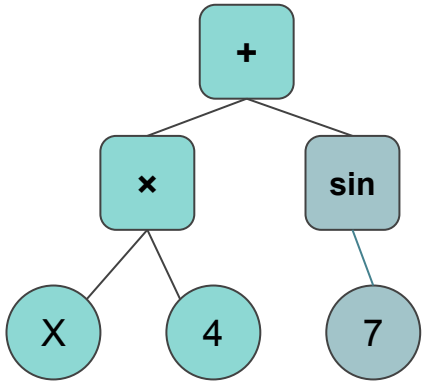
Point mutations



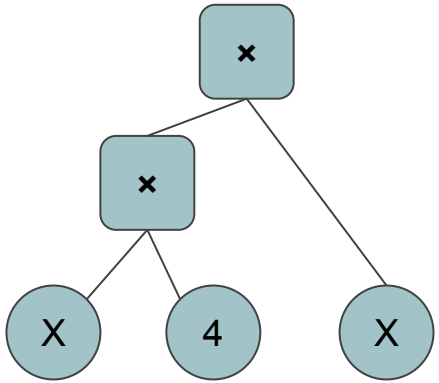
Hoist mutations



Subtree mutations

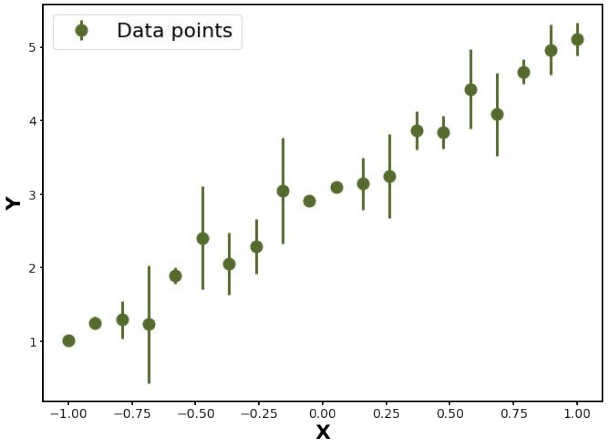


Crossover



Traditional Symbolic Regression

DATA SET



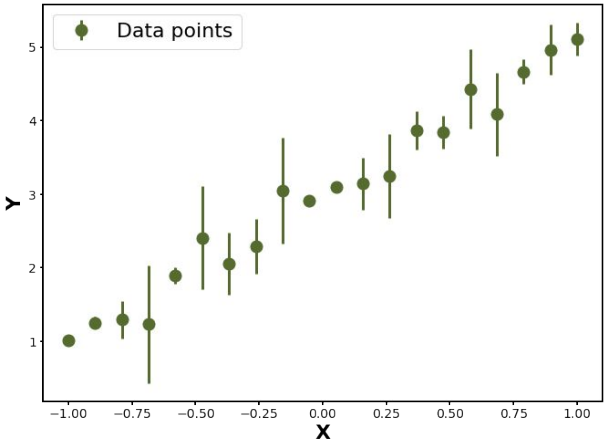
RANDOM EQUATIONS

COST FUNCTION

f(X) = sin(X) + 2	COST = 12
f(X) = X² - 1	COST = 24
f(X) = 42	COST = 43
f(X) = X	COST = 7
f(X) = 2X	COST = 3

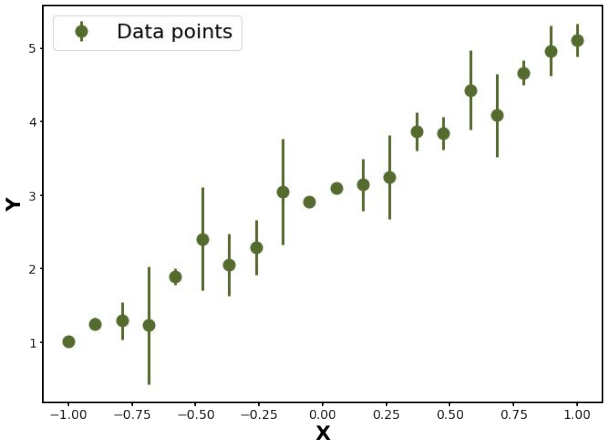
Traditional Symbolic Regression

DATA SET



Traditional Symbolic Regression

DATA SET



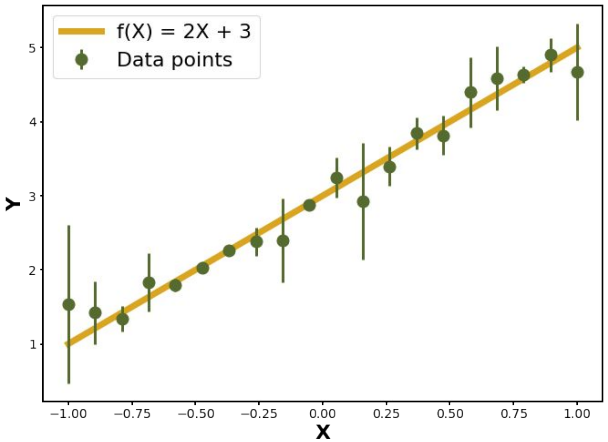
EVOLVED EQUATIONS

COST FUNCTION

f(X) = 2X - 8	COST = 10
f(X) = X	COST = 7
f(X) = 2X + 2	COST = 0.5
f(X) = 2X	COST = 7
f(X) = 1/X	COST = 42

Traditional Symbolic Regression

DATA SET



After many generation

Best answer

f(X) = 2 X + 3

COST = 0

Traditional Symbolic Regression

Modelling the galaxy-halo connection with machine learning

Show affiliations

Delgado, Ana Maria ; Wadekar, Digvijay ; Hadzhiyska, Boryana ; Bose, Sownak ; Hernquist, Lars ; Ho, Shirley

Automated discovery of interpretable gravitational-wave population models

Show affiliations

Wong, Kaze W. K ; Cranmer, Miles

Augmenting astrophysical scaling relations with machine learning : application to reducing the SZ flux-mass scatter

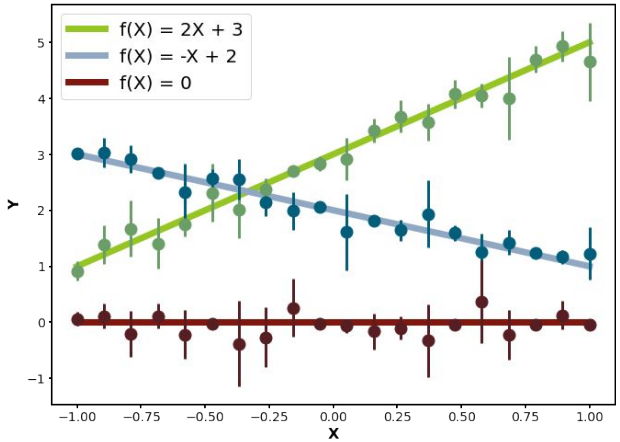
Show affiliations

Wadekar, Digvijay ; Thiele, Leander ; Villaescusa-Navarro, Francisco ; Hill, J. Colin ; Cranmer, Miles ; Spergel, David N. ; Battaglia, Nicholas ; Anglés-Alcázar, Daniel ; Hernquist, Lars ; Ho, Shirley

MultiView Symbolic Regression

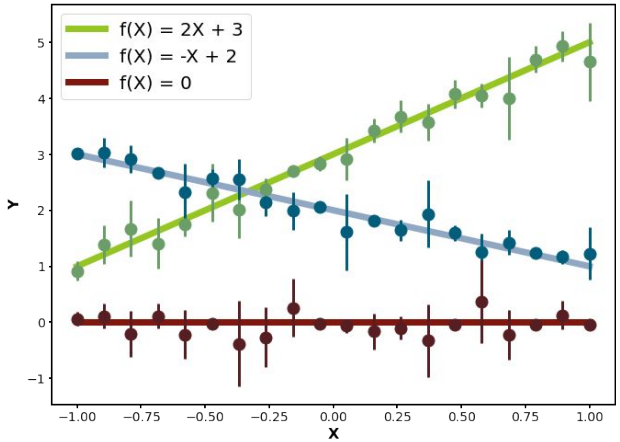
Multiview Symbolic Regression (MvSR)

DATA SETS



Multiview Symbolic Regression (MvSR)

DATA SETS



Best answers

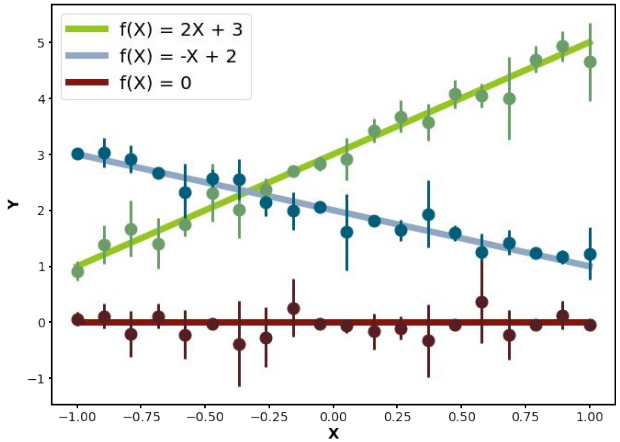
$f(X) = 2X + 3$

$f(X) = -X + 2$

$f(X) = 0$

Multiview Symbolic Regression (MvSR)

DATA SETS



Best answers

f(X) = 2 X + 3

f(X) = -X + 2

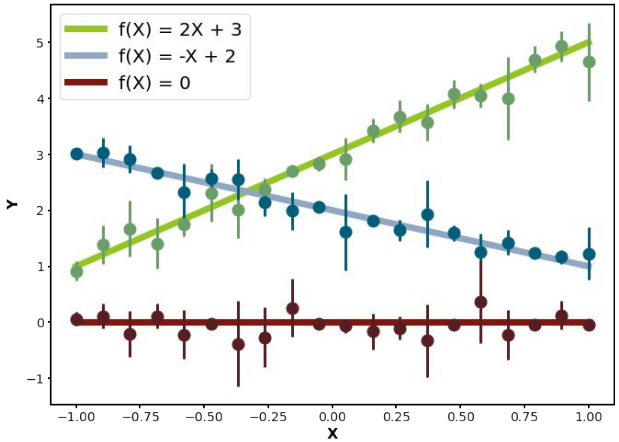
f(X) = 0

Common equation

f(X) = **C1** X + **C2**

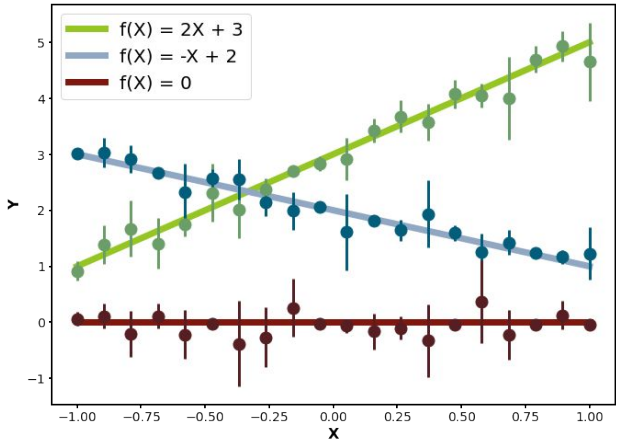
Multiview Symbolic Regression (MvSR)

DATA SETS



Multiview Symbolic Regression (MvSR)

DATA SETS



RANDOM EQUATIONS

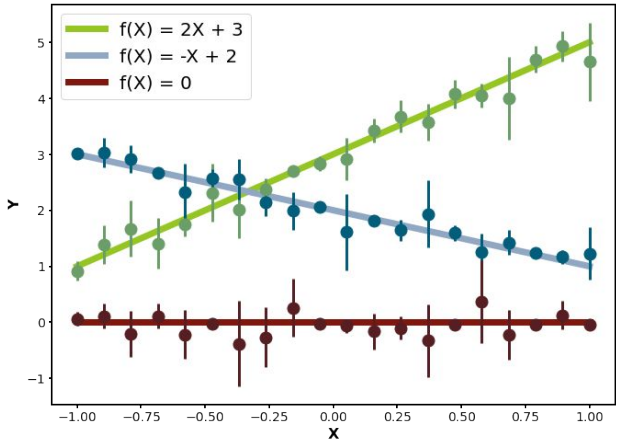
$f(X) = \sin(X) + \mathbf{C1}$

$f(X) = \mathbf{C1} + \mathbf{C2}\ X^2$

$f(X) = \mathbf{C1}$

Multiview Symbolic Regression (MvSR)

DATA SETS



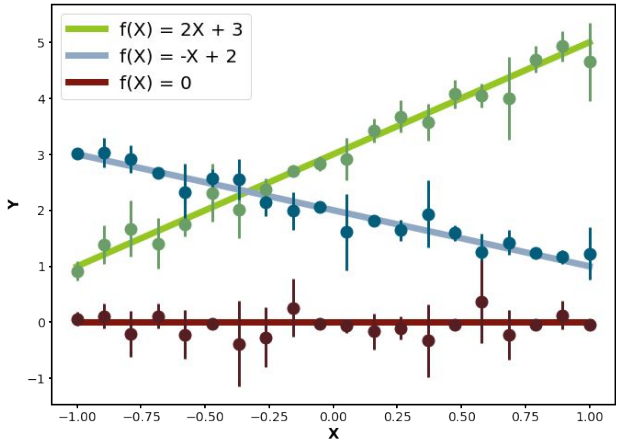
RANDOM EQUATIONS

COST FUNCTION AFTER MINIMIZATION

f(X) = sin(X) + **C1**	COST = 24 32 7
f(X) = **C1** + **C2** X^2	COST = 17 8 0
f(X) = **C1**	COST = 19 10 0

Multiview Symbolic Regression (MvSR)

DATA SETS



RANDOM EQUATIONS

COST FUNCTION AFTER MINIMIZATION

f(X) = sin(X) + **C1** COST = 24 32 7

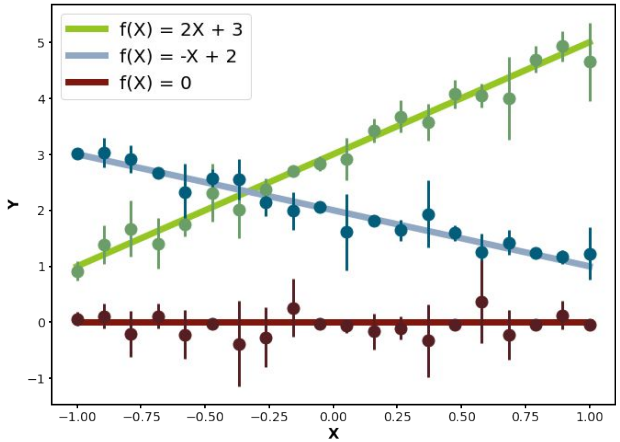
f(X) = **C1** + **C2** X^2 COST = 17 8 0

f(X) = **C1** COST = 19 10 0

Best average COST

Multiview Symbolic Regression (MvSR)

DATA SETS



After many generation

Best answer

f(X) = **C1** X + **C2**

COST = 0 0 0

Multiview Symbolic Regression (MvSR)

Avantages:

→ Computation is performed only once

→ Densest way of performing regression

→ Protection against over fitting

→ Offers interpretable results

MvSR code built from the python module GPlearn : https://gplearn.readthedocs.io/en/stable/

Multiview Symbolic Regression (MvSR)

Smart error computation :

CHI2 → Bayesian Information Criterion (BIC)

$$BIC = \underbrace{\sum_i \frac{(Y_i - f(x_i))^2}{\sigma_i^2}}_{\chi^2} +$$

Multiview Symbolic Regression (MvSR)

Smart error computation :

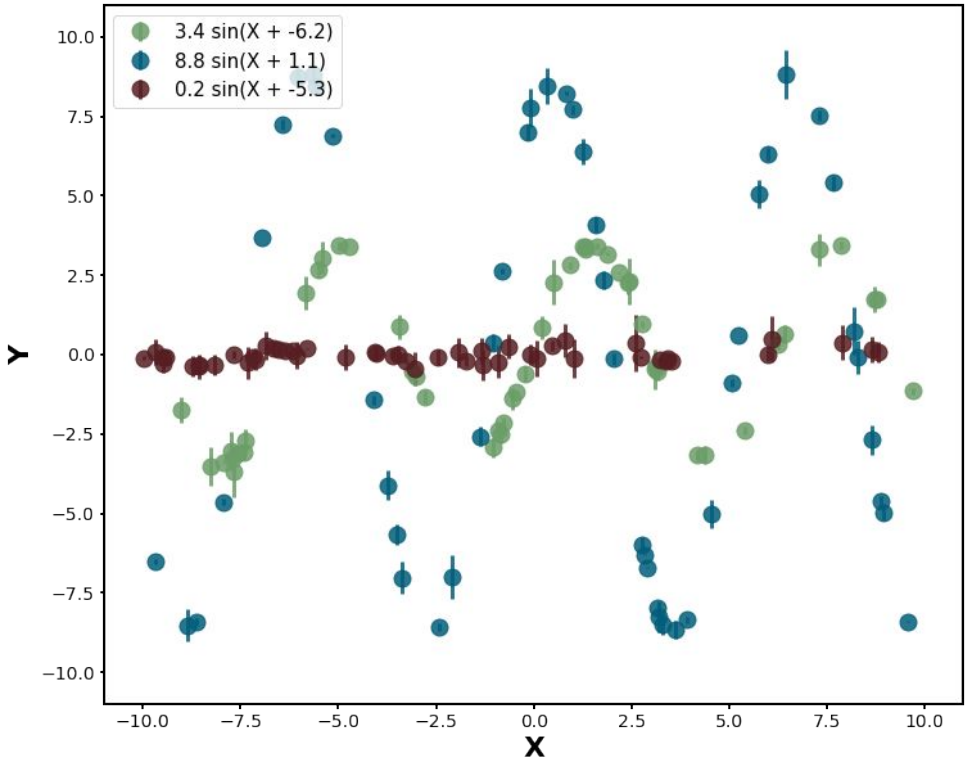
CHI2 → Bayesian Information Criterion (BIC)

$$BIC = \underbrace{\sum_i \frac{(Y_i - f(x_i))^2}{\sigma_i^2}}_{\chi^2} + K \times ln(n)$$

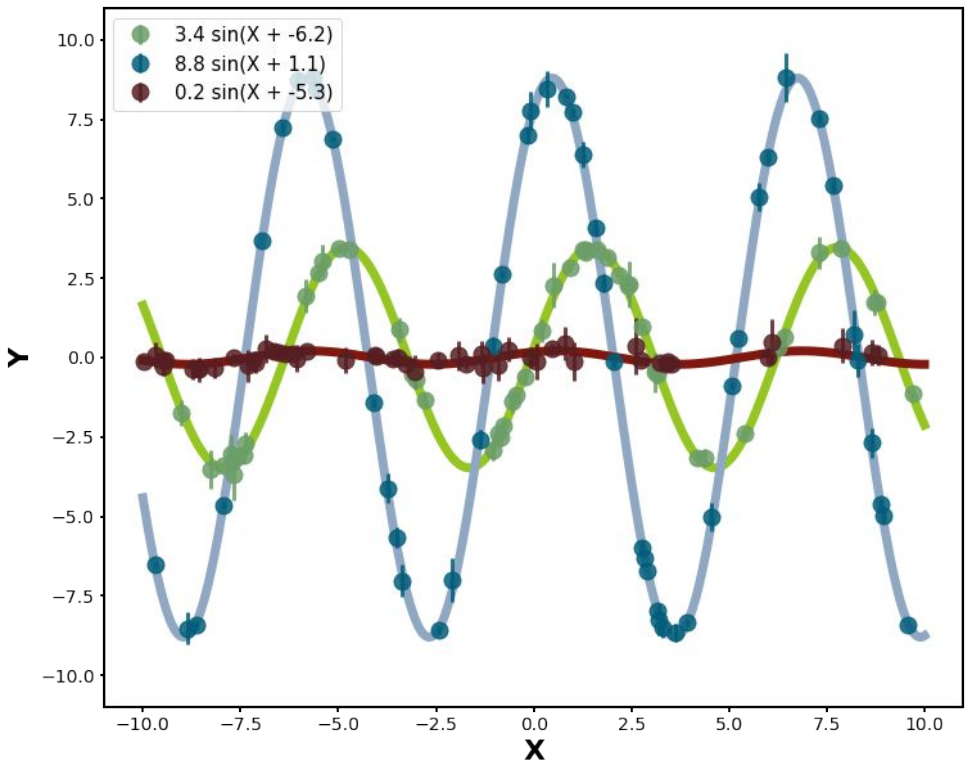
Number of free parameters in the equation

Number of points

Toy data application



	Population Average		Best Individual	
Gen	Length	Fitness	Length	Fitness	Time Left
0	20.97	4756.33	6	1832.67	13.21s
1	15.13	7985.6	4	1832.67	6.26s
2	5.83	2307.86	8	1832.67	2.81s
3	4.63	2208.34	4	1832.67	2.01s
4	4.30	2116.88	4	1832.67	2.15s
5	4.47	2280.56	6	13.7427	1.85s
6	4.77	2202.58	6	13.7427	1.22s
7	6.43	1160.55	6	13.7427	0.82s
8	5.73	1727.43	6	13.7427	0.44s
9	5.70	7197.51	6	13.7427	0.00s

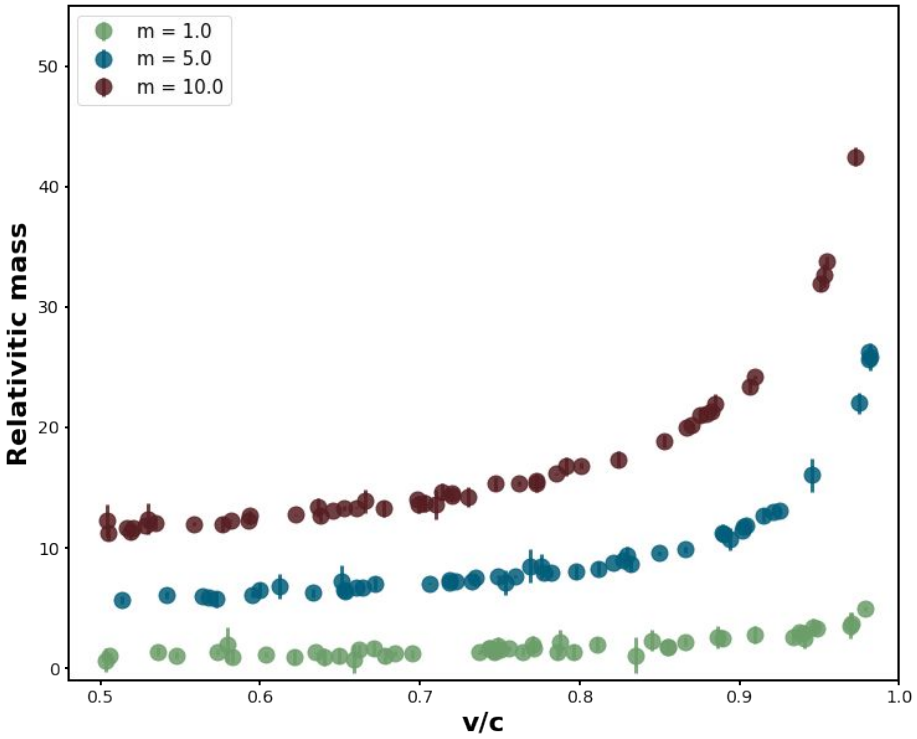


	Population Average		Best Individual	
Gen	Length	Fitness	Length	Fitness	Time Left
0	20.97	4756.33	6	1832.67	13.21s
1	15.13	7985.6	4	1832.67	6.26s
2	5.83	2307.86	8	1832.67	2.81s
3	4.63	2208.34	4	1832.67	2.01s
4	4.30	2116.88	4	1832.67	2.15s
5	4.47	2280.56	6	13.7427	1.85s
6	4.77	2202.58	6	13.7427	1.22s
7	6.43	1160.55	6	13.7427	0.82s
8	5.73	1727.43	6	13.7427	0.44s
9	5.70	7197.51	6	13.7427	0.00s

▾ SymbolicRegressor

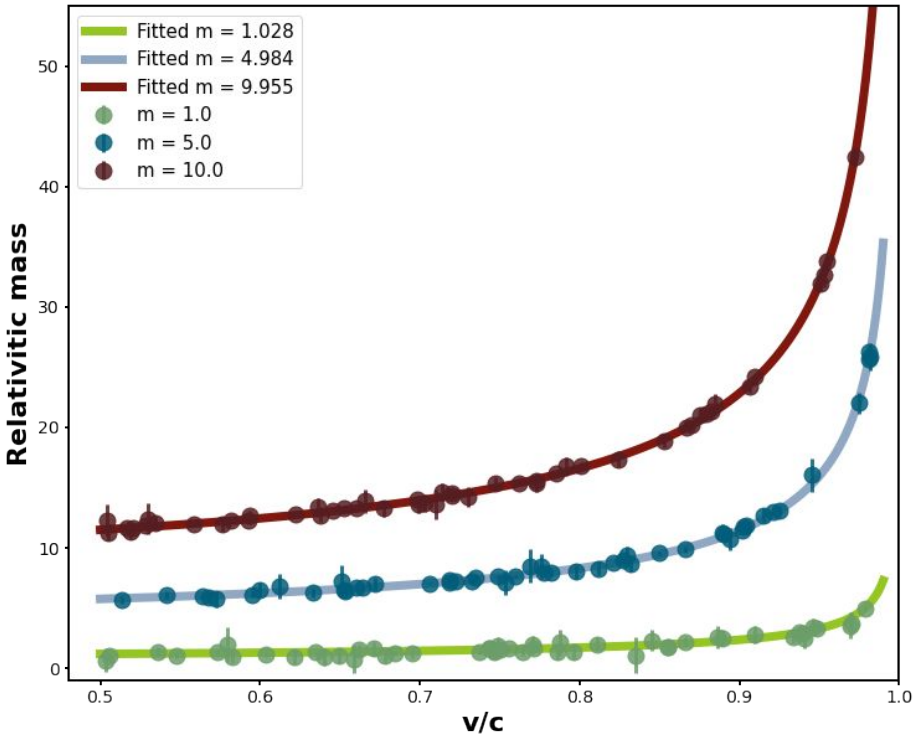
```
mul(C0, sin(sub(C1, X0)))
```

$$C_0 \sin(C_1 - X_0)$$



Data generated from the relativistic mass equation :

$$m_r = \frac{m}{\sqrt{1 - \frac{v^2}{c^2}}} = \frac{m}{\sqrt{1 - X^2}}$$



Data generated from the relativistic mass equation :

$$m_r = \frac{m}{\sqrt{1 - \frac{v^2}{c^2}}} = \frac{m}{\sqrt{1 - X^2}}$$

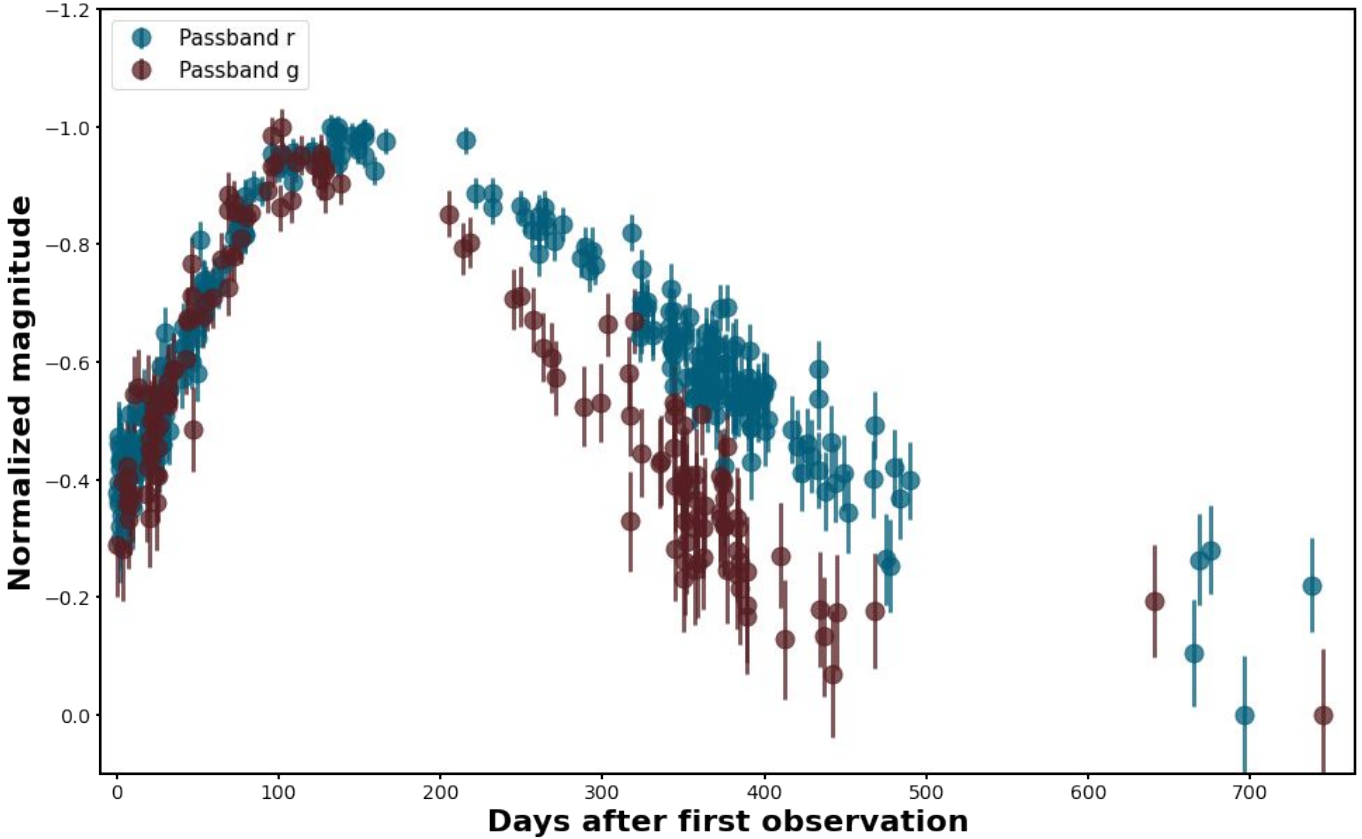
▾ SymbolicRegressor

```
sqrt(div(C0, sub(sqrt(X0), X0)))
```

$$\frac{C_0}{\left(1.0 - X_0^2\right)^{0.5}}$$

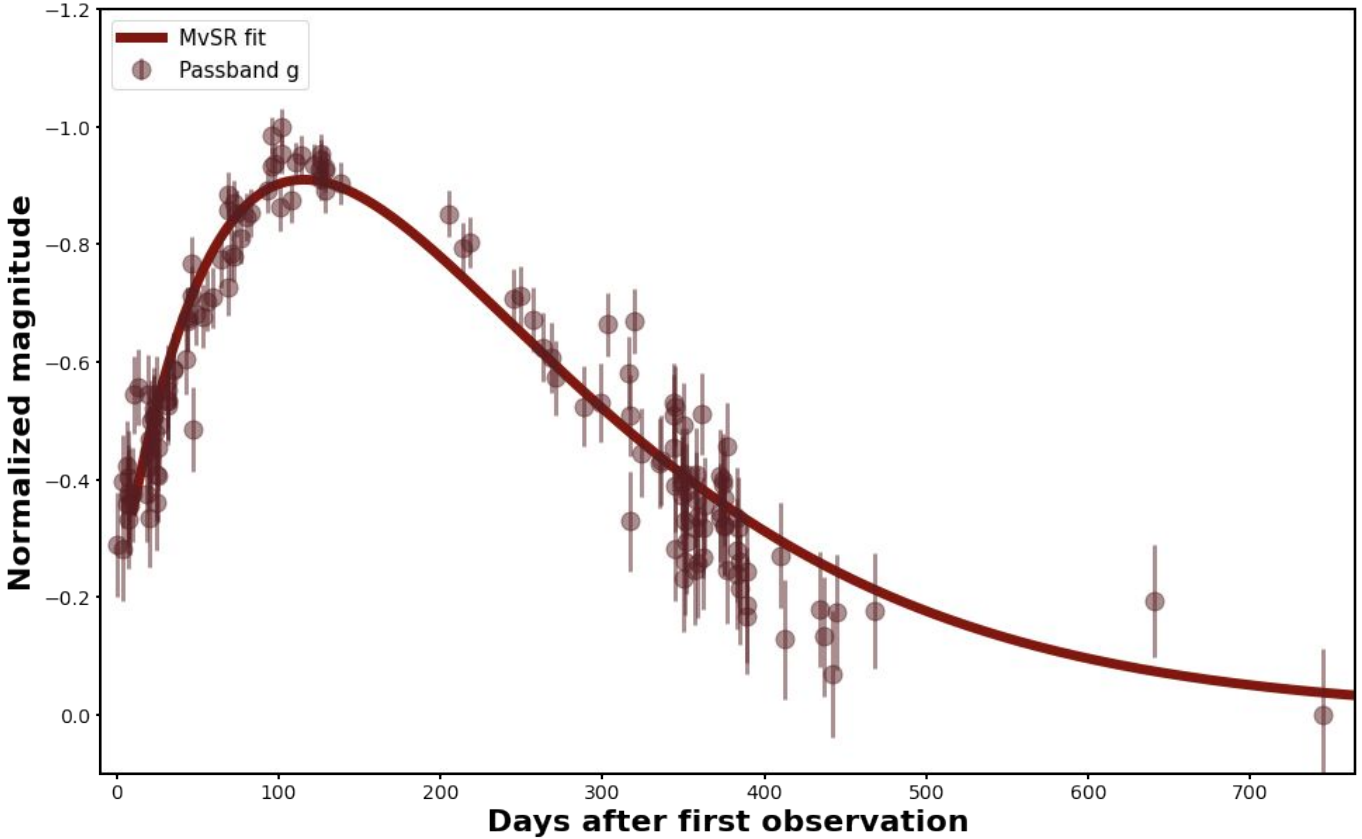
Real data application

https://ztf.snad.space/dr8/view/821207100004043



SNAD 160

https://ztf.snad.space/dr8/view/821207100004043

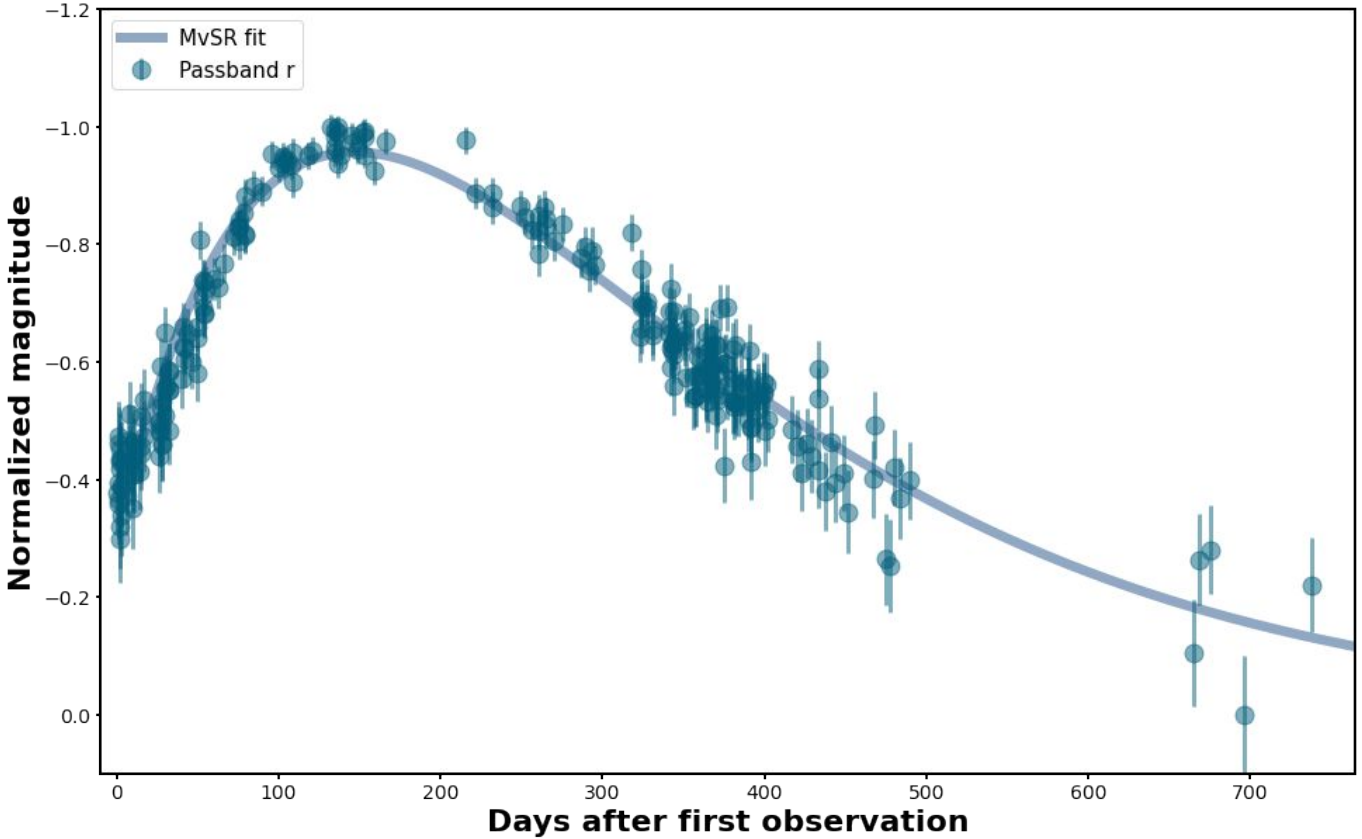


SNAD 160

$$f(t) = C_1(t - C_2) \times e^{C_0 \times (t - C_2)}$$

https://ztf.snad.space/dr8/view/821207100004043

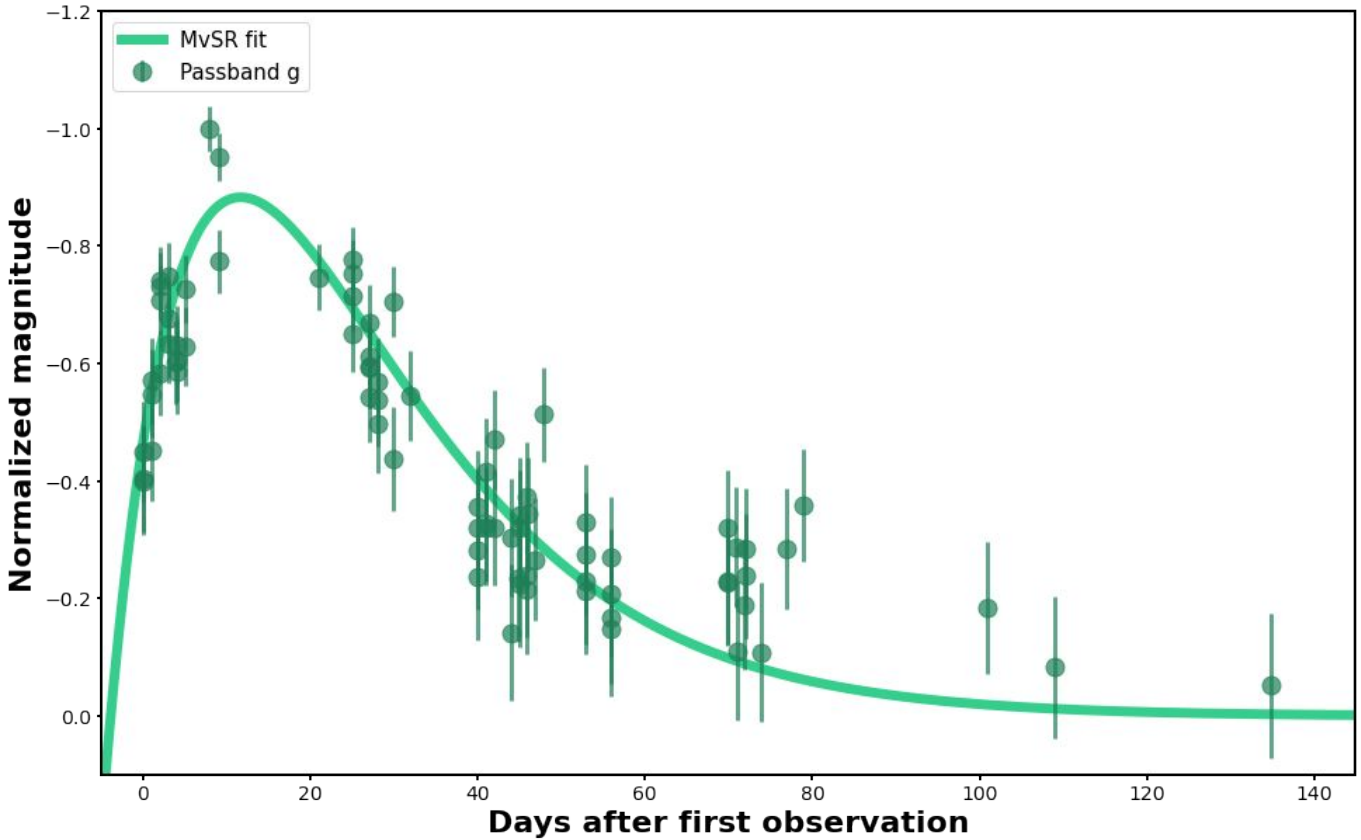
SNAD 160



$$f(t) = C_1(t - C_2) \times e^{C_0 \times (t - C_2)}$$

https://ztf.snad.space/dr8/view/791211100013499

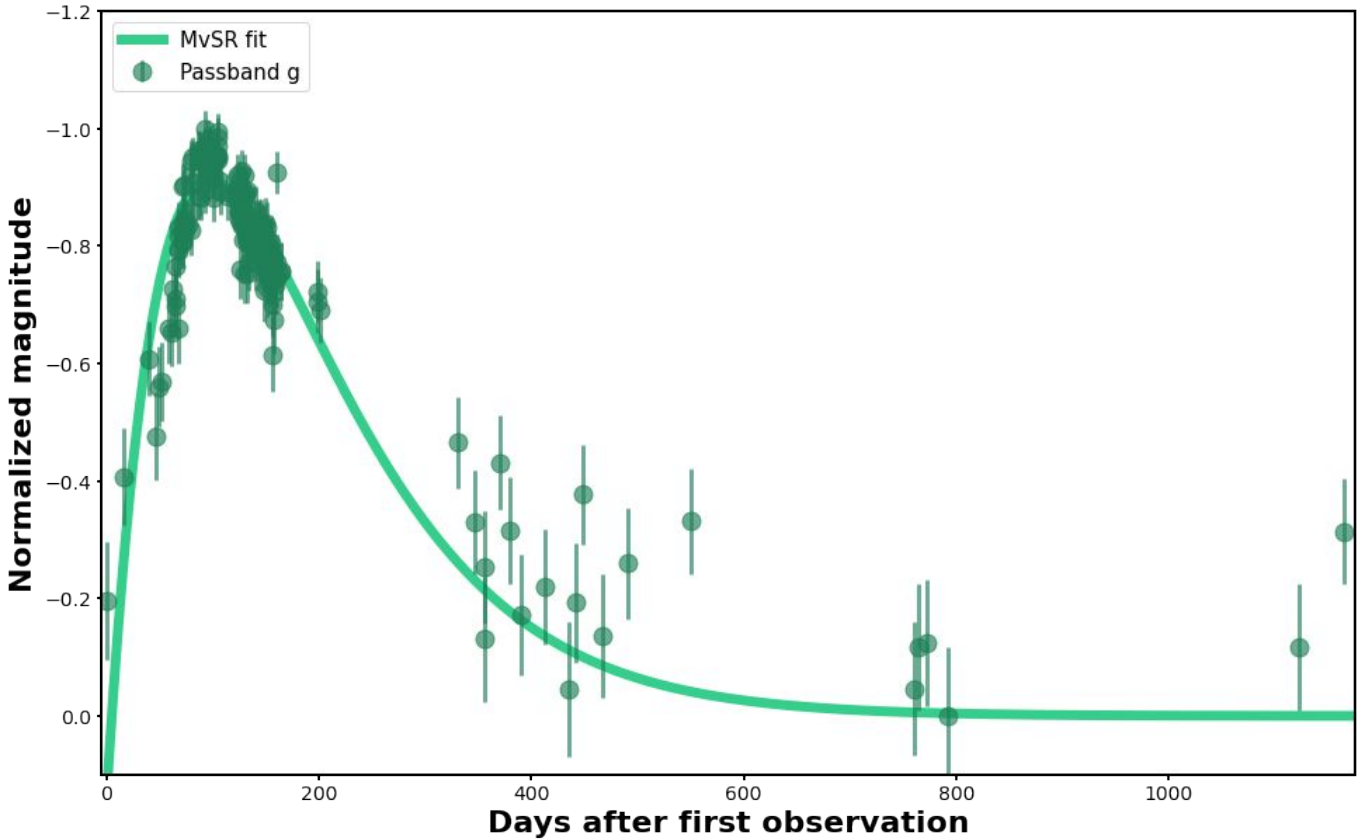
SNAD 161



Apply the equation on a different light curve !

https://ztf.snad.space/dr8/view/791211100013499

SNAD 120



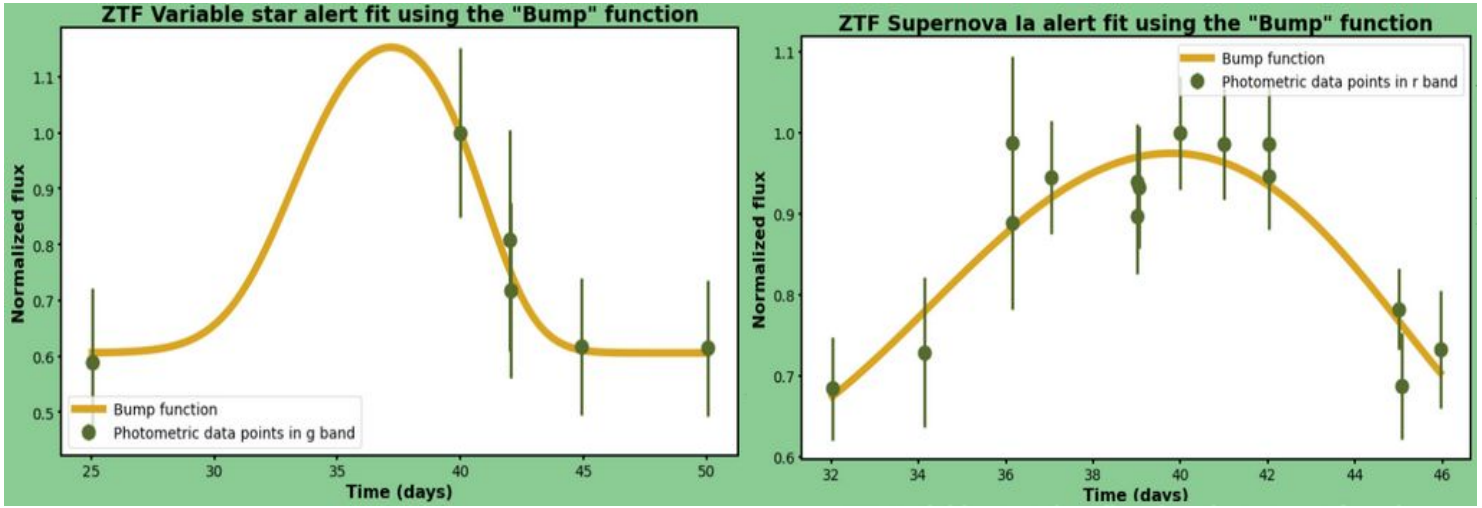
Apply the equation on a different light curve !

Conclusion

- **Method shows great first results on :**
  - Toy data
  - Real astrophysical examples
- **Drawbacks :**
  - Computation time can explode
  - Some data might never converge
- **Future applications :**
  - Implementation in a more efficient way
  - Feature extraction of light curves (to be continued tomorrow)
  - Lightcurve prediction

Thank you for your attention

Extra slides



Bump function

$$f(t) = \frac{1}{1 + exp^{-(At - exp^{Bt} + C)}}$$

Used to compute color for an AGN classifier within the broker Fink

1 Jul 2022

Deep Learning and Symbolic Regression for Discovering Parametric Equations

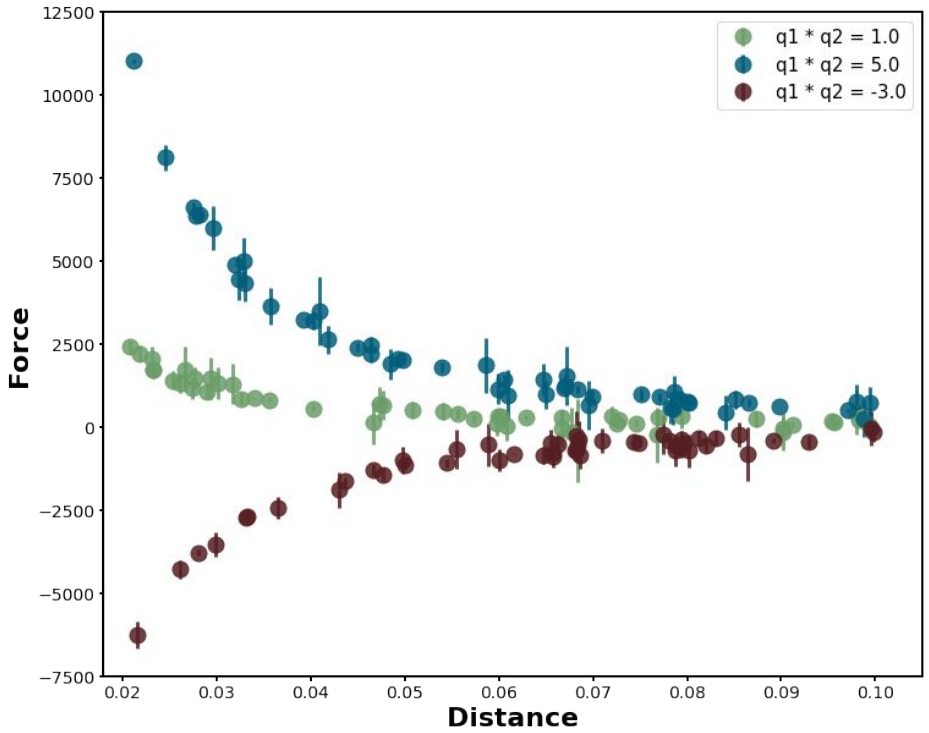
Michael Zhang*[1], Samuel Kim*[1a], Peter Y. Lu[2], Marin Soljačić[2b]

***Abstract*—Symbolic regression is a machine learning technique that can learn the governing formulas of data and thus has the potential to transform scientific discovery. However, symbolic regression is still limited in the complexity and dimensionality of the systems that it can analyze. Deep learning on the other hand has transformed machine learning in its ability to analyze extremely complex and high-dimensional datasets. We propose a neural network architecture to extend symbolic regression to parametric systems where some coefficient may vary but the structure of the underlying governing equation remains constant. We demonstrate our method on various analytic expressions, ODEs, and PDEs with varying coefficients and show that it extrapolates well outside of the training domain. The neural network-based architecture can also integrate with other deep learning architectures so that it can analyze high-dimensional data while being trained end-to-end. To this end we integrate our architecture with convolutional neural networks to analyze 1D images of varying spring systems.**

***Index Terms*—Symbolic regression, deep learning, neural network, parametric, PDE, varying coefficient, high-dimensional**

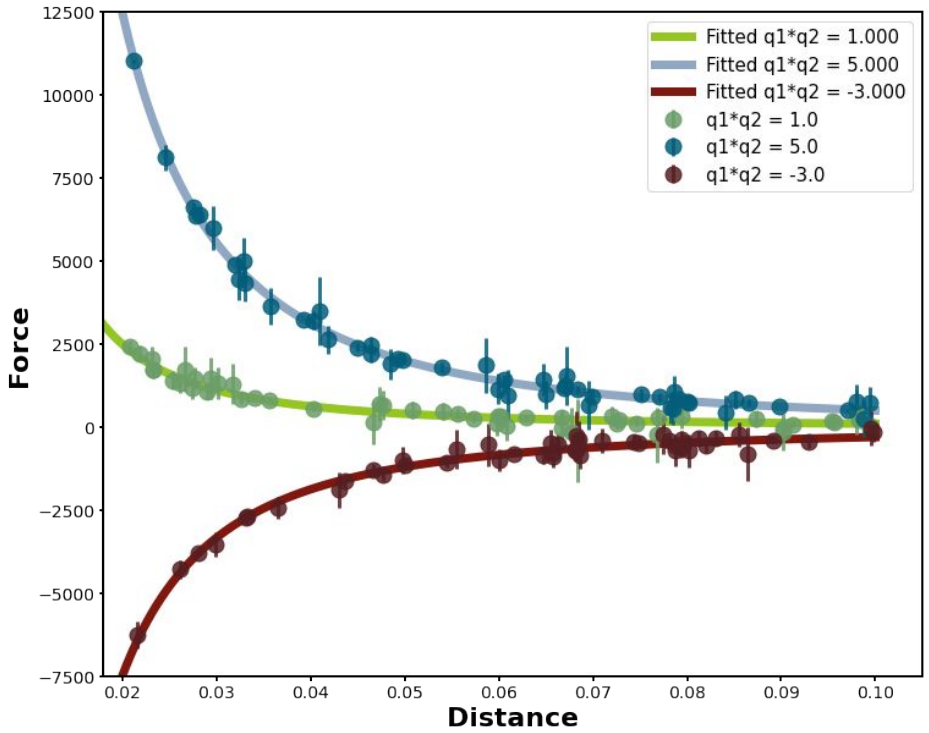
heuristics [3]. The equations are pieced together through basic building blocks known as primitive functions, which include constants and simple functions (e.g. addition, multiplication, sine). Ref. [4], one of the most popular earlier works in this direction, demonstrated how symbolic regression could discover equations of motions including Hamiltonians and Lagrangians for various physical systems. However, these approaches do not scale well to high-dimensional problems and often require numerous hand-built heuristics and rules.

One type of complexity we explore in this work are datasets described by parametric equations in which the underlying equation may stay the same but coefficients may vary along one or more dimensions. For example, the diffusion constant may vary over time or space as the system governed by the diffusion equation evolves. Various approaches have been proposed to discover parametric PDEs, including genetic algorithms combined with averaging over local windows [5], linear regression with kernel smoothing over adjacent coefficients



Data generated from Coulomb law:

$$F(d) \propto \frac{q_1 \times q_2}{d^2} = \frac{Q}{d^2}$$



Data generated from Coulomb law:

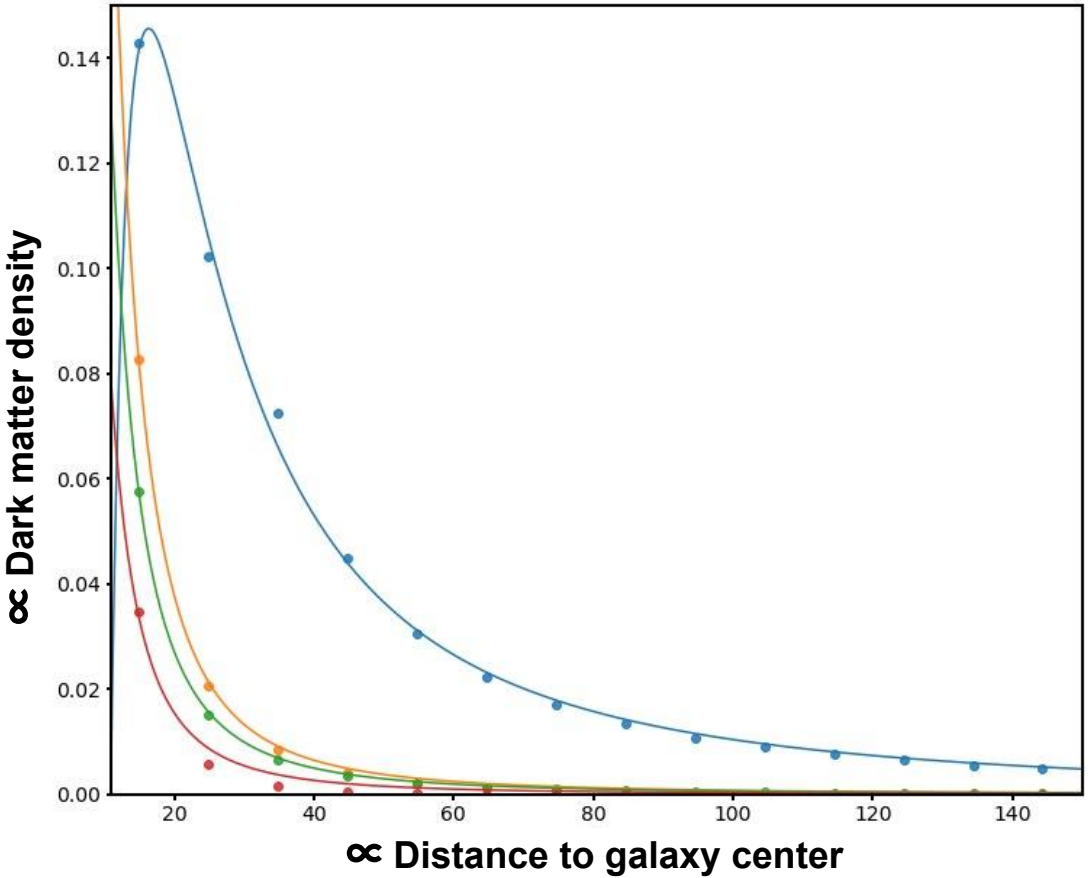
$$F(d) \propto \frac{q_1 \times q_2}{d^2} = \frac{Q}{d^2}$$

SymbolicRegressor

```
div(div(C0, X0), X0)
```

$$\frac{C_0}{X_0^2}$$

SIMULATION DATA

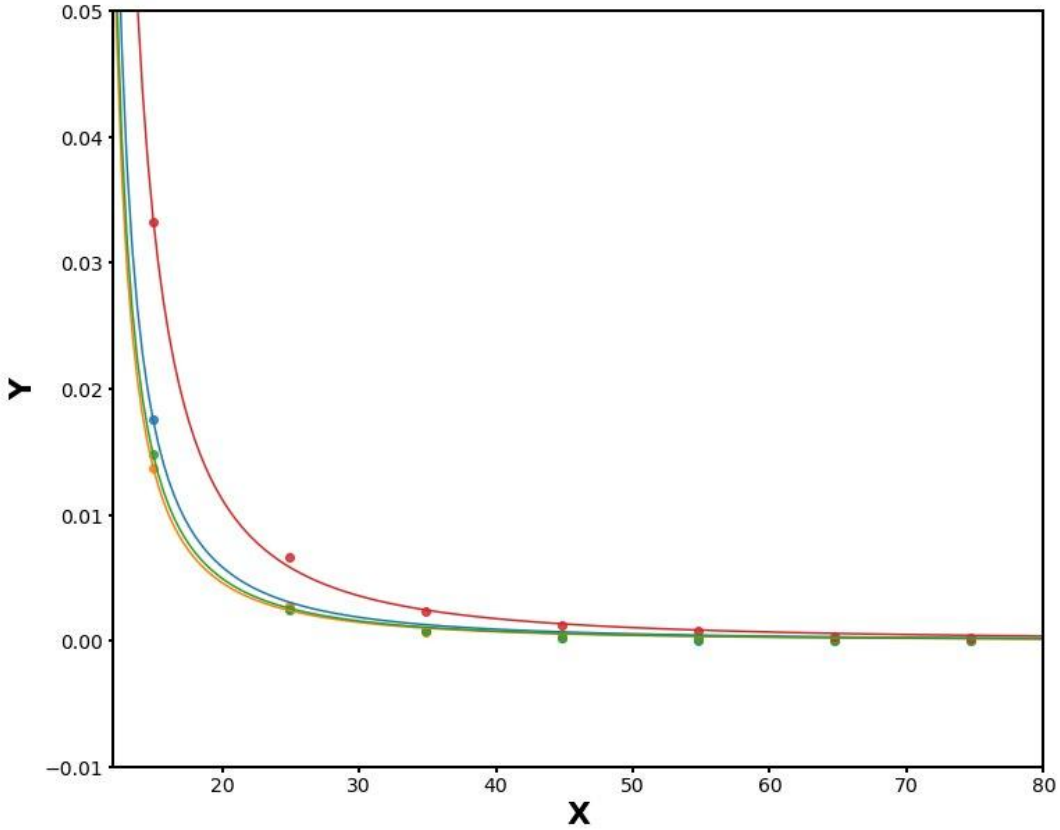


Trying to recover NFW equation from simulation :

NFW : $$\rho(r) = \frac{\rho_0}{\frac{r}{R_s}(1+\frac{r}{R_s})^2}$$

MvSR attempt : $$\rho(r) = \frac{\frac{C_0}{r} + 0.288}{C_1 \times r^2}$$

Trying to recover NFW equation from simulation :



NFW : $$\rho(r) = \frac{\rho_0}{\frac{r}{R_s}(1+\frac{r}{R_s})^2}$$

MvSR attempt (1 parameter): $$\rho(r) = \frac{C_0}{r(r+10.8)}$$