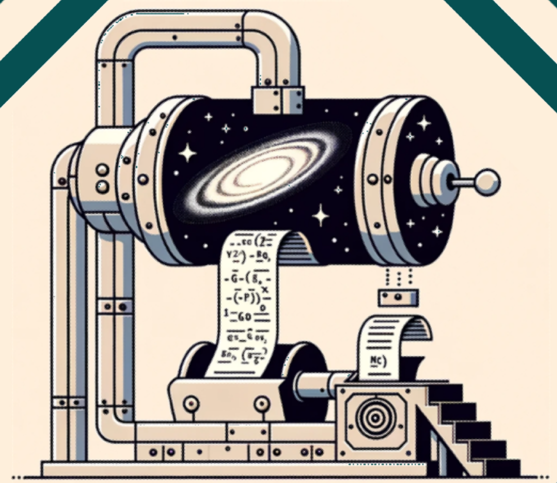
Université de Strasbourg

Observatoire astronomique de Strasbourg



(Machine)-learning (astro)-physical laws

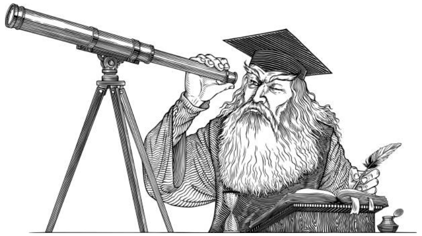
Wassim Tenachi

With Rodrigo Ibata & Foivos Diakogiannis

Nov 27 2023

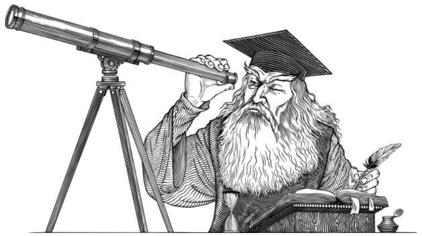
ML IAP/CCA 2023 – Debating the Potential of Machine Learning in Astronomical Surveys

Motivations



Motivations

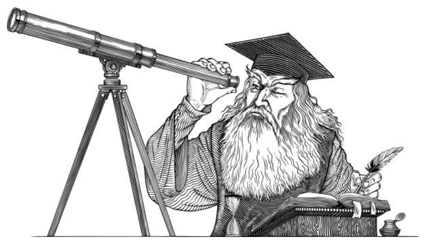
Empirical law



$$T^2 = \frac{4\pi^2}{GM} a^3$$

Motivations

Empirical law



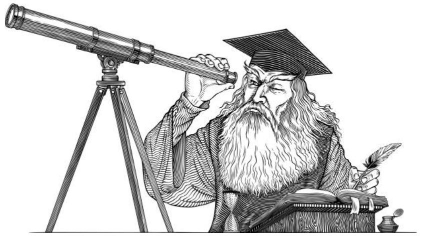
$$T^2 = \frac{4\pi^2}{GM} a^3$$

Took him 4 years !

Motivations

Empirical law

General theory



$$T^2 = \frac{4\pi^2}{GM} a^3$$

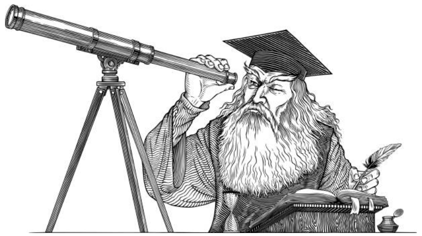
$$m\frac{d\boldsymbol{v}}{dt} = \mathbf{F}$$

Took him 4 years !

Motivations

Empirical law

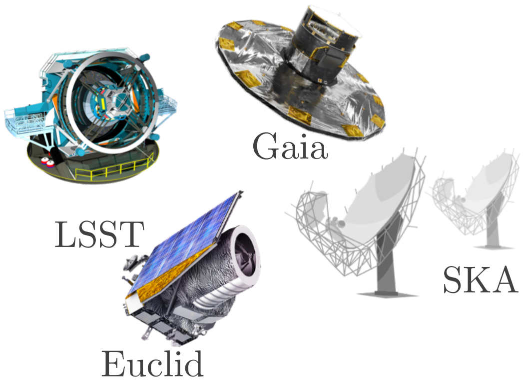
General theory



$$T^2 = \frac{4\pi^2}{GM} a^3$$

$$m\frac{d\boldsymbol{v}}{dt} = \mathbf{F}$$

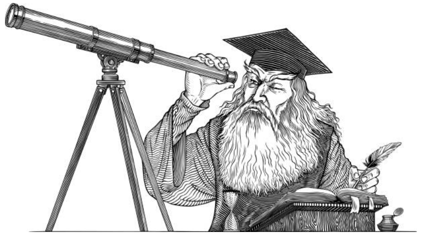
Took him 4 years !



Motivations

Empirical law

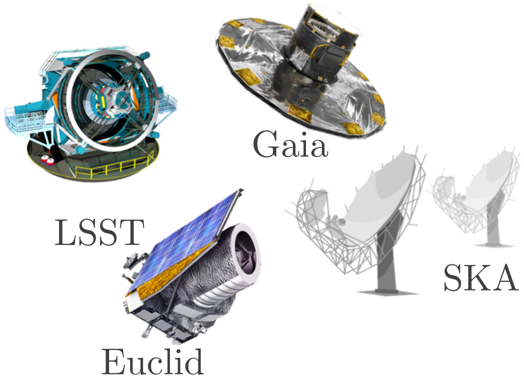
General theory

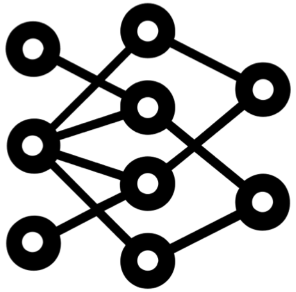


$$T^2 = \frac{4\pi^2}{GM} a^3$$

$$m\frac{d\boldsymbol{v}}{dt} = \mathbf{F}$$

Took him 4 years !

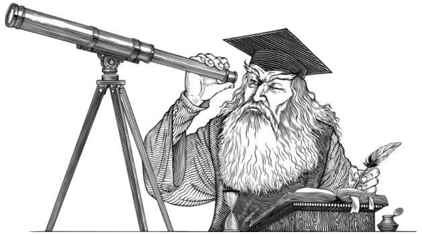




Motivations

Empirical law

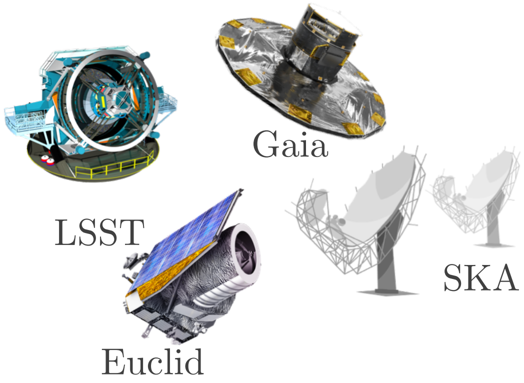
General theory

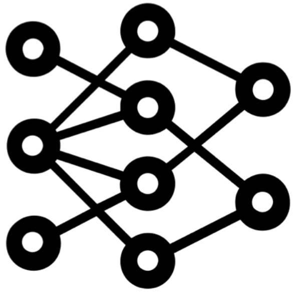


$$T^2 = \frac{4\pi^2}{GM} a^3$$

$$m\frac{dv}{dt} = \mathbf{F}$$

Took him 4 years !



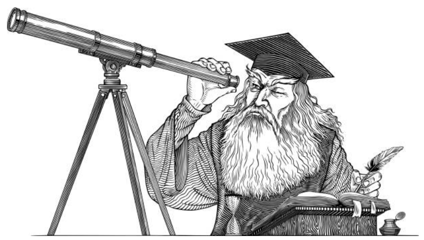


→ Black box

Motivations

Empirical law

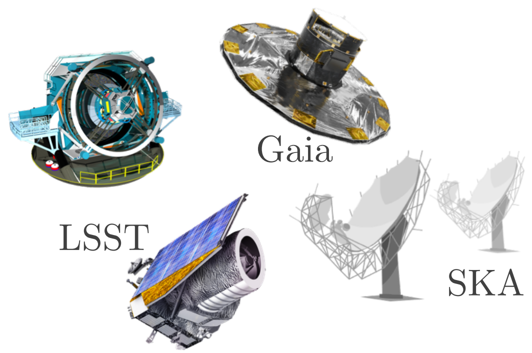
General theory

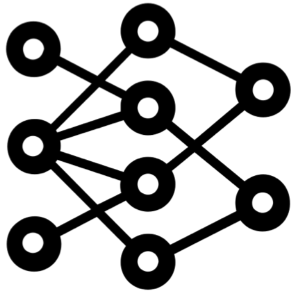


$$T^2 = \frac{4\pi^2}{GM} a^3$$

$$m\frac{dv}{dt} = \mathbf{F}$$

Took him 4 years !





Empirical law

$\rightarrow f(\boldsymbol{x}) = \cdots$

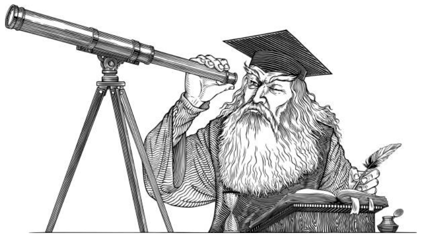
→ Black box

→ Enables connection with theory

Motivations

Empirical law

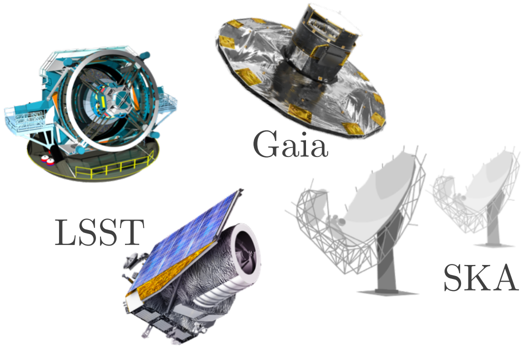
General theory

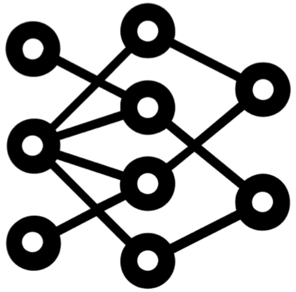


$$T^2 = \frac{4\pi^2}{GM} a^3$$

$$m \frac{d\boldsymbol{v}}{dt} = \mathbf{F}$$

Took him 4 years !





Empirical law

$f(\boldsymbol{x}) = \cdots$

new physics

→ Black box

→ Enables connection with theory

Symbolic regression (SR)

$(\boldsymbol{x}, y)$ data

x_1	x_2	y
0.75582	0.25850	0.02674
0.36786	0.42401	0.06278
0.69507	0.38057	0.74014
0.96493	0.33398	0.81558
0.07139	0.16604	0.07735
0.86413	0.41952	0.87872
0.18012	0.40581	0.63637

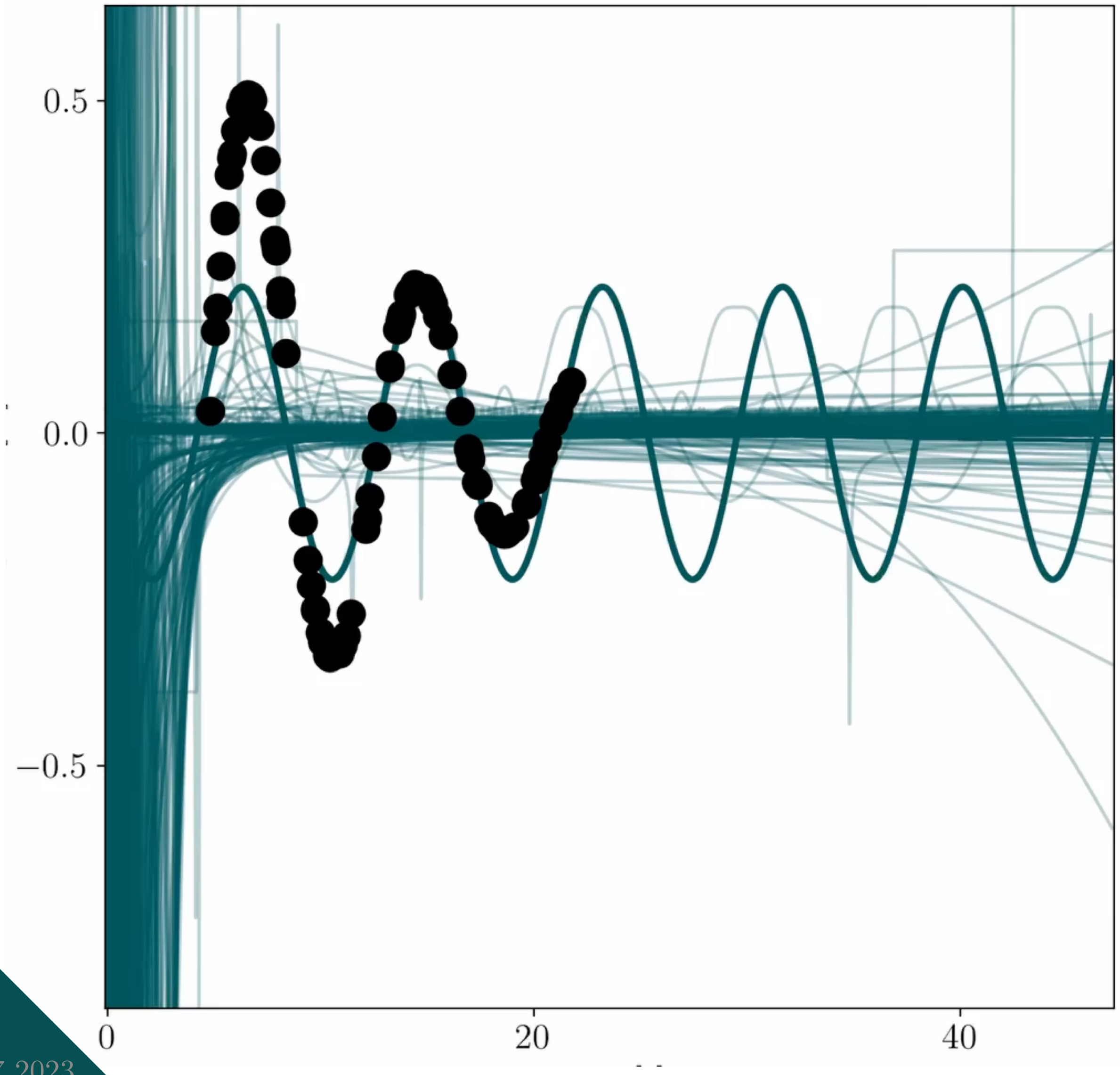
→

$f: \mathbb{R}^n \rightarrow \mathbb{R}$ such that
$$y = f(\boldsymbol{x})$$

Symbolic regression (SR)

Φ-SO

Physical Symbolic Optimization.



Best fit:

$$x(t) = \sin(\alpha t)\cos(\phi^2)$$

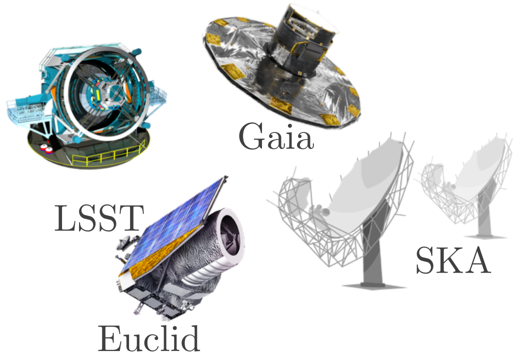
Trying:

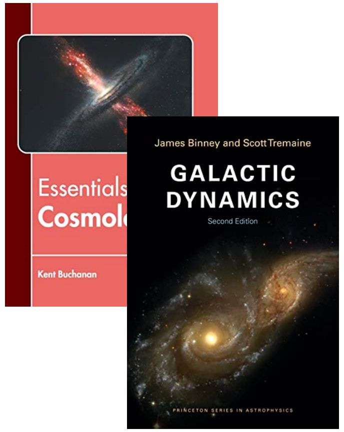
$$x(t) = \sin(\alpha t)\cos(\phi^2)$$

https://youtu.be/wubzZMkoTUY

The virtues of obtaining symbolic models (1)

1. Interpretability

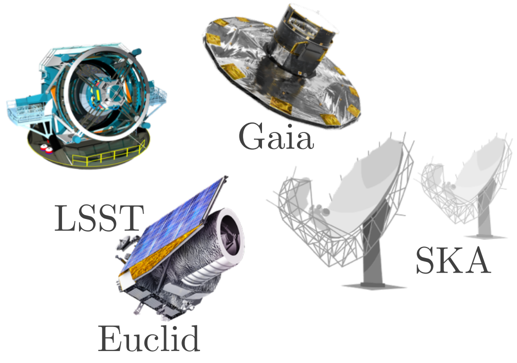


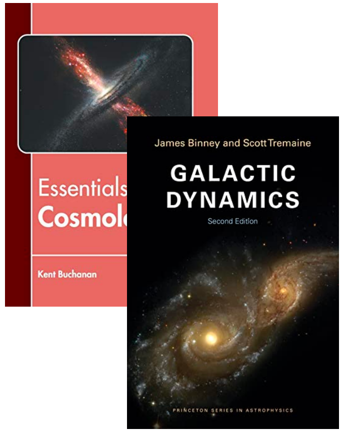


→ Connecting with theory

The virtues of obtaining symbolic models (1)

1. Interpretability



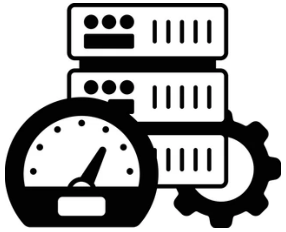


→ Connecting with theory

2. Compactness

$$f(\boldsymbol{x}) = \cdots$$

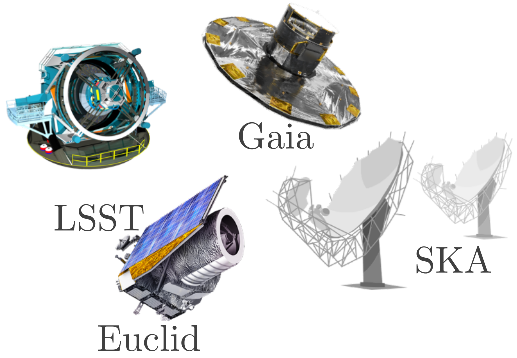
→ Intellegible

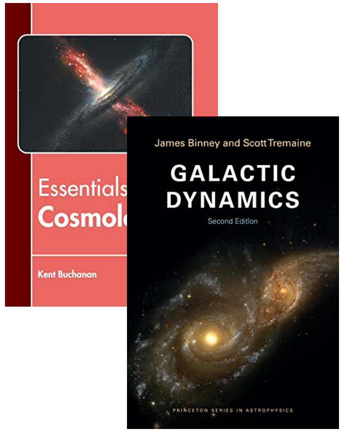


→ Cheap

The virtues of obtaining symbolic models (1)

1. Interpretability



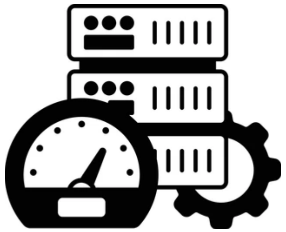


→ Connecting with theory

2. Compactness

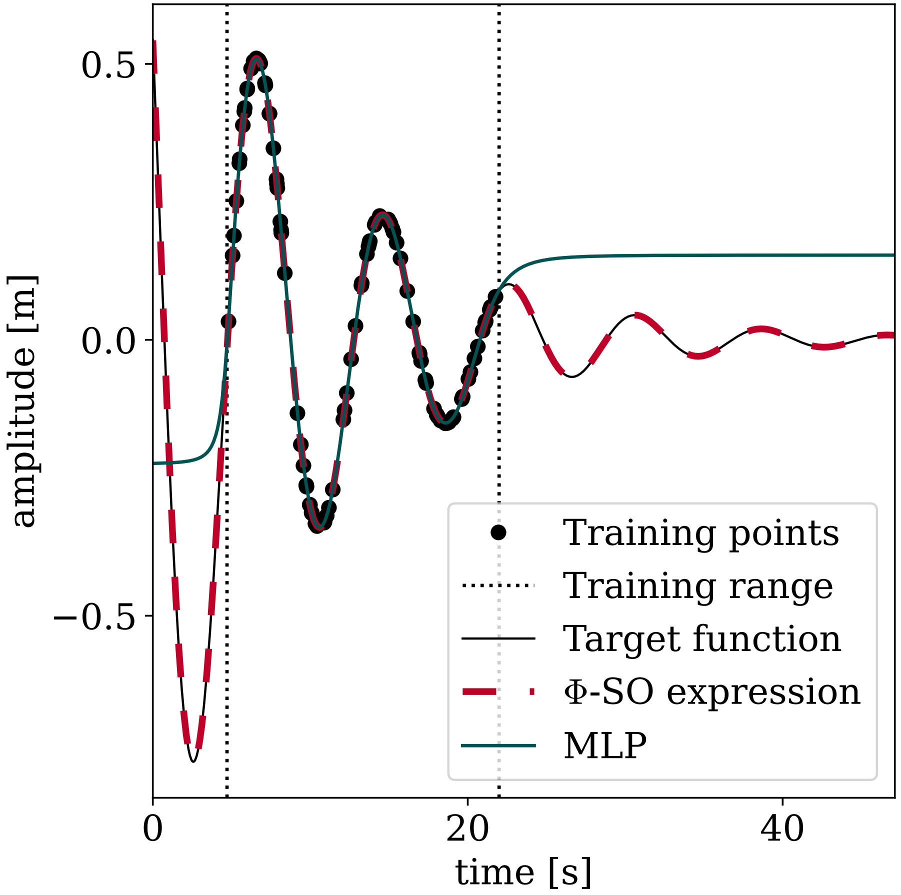
$$f(\boldsymbol{x}) = \cdots$$

→ Intellegible

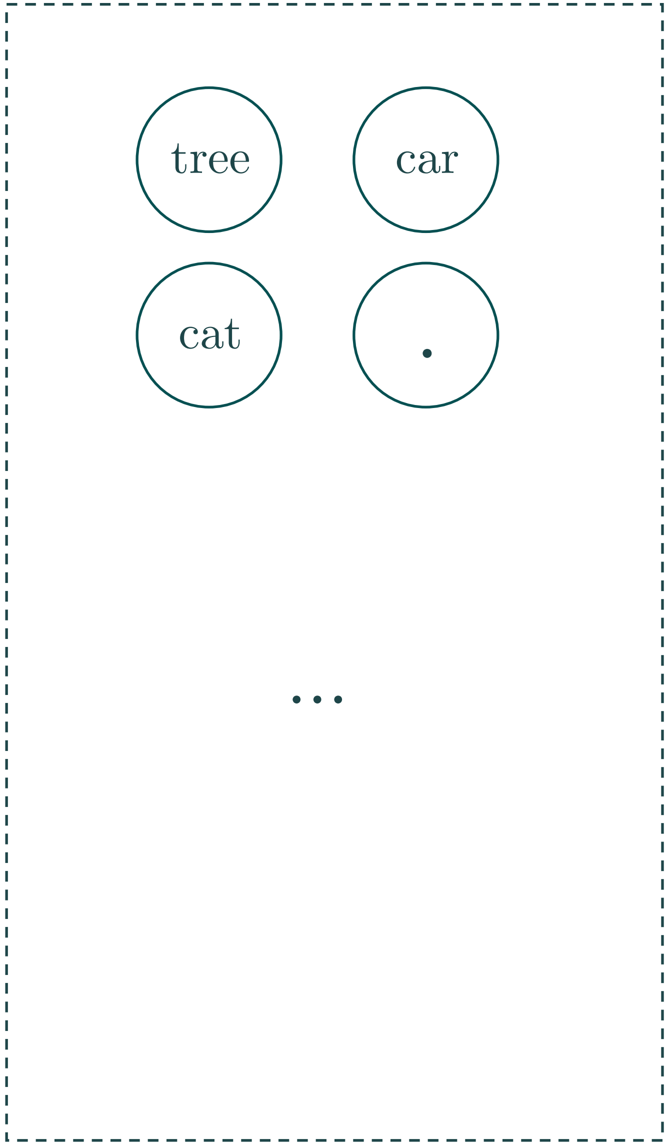


→ Cheap

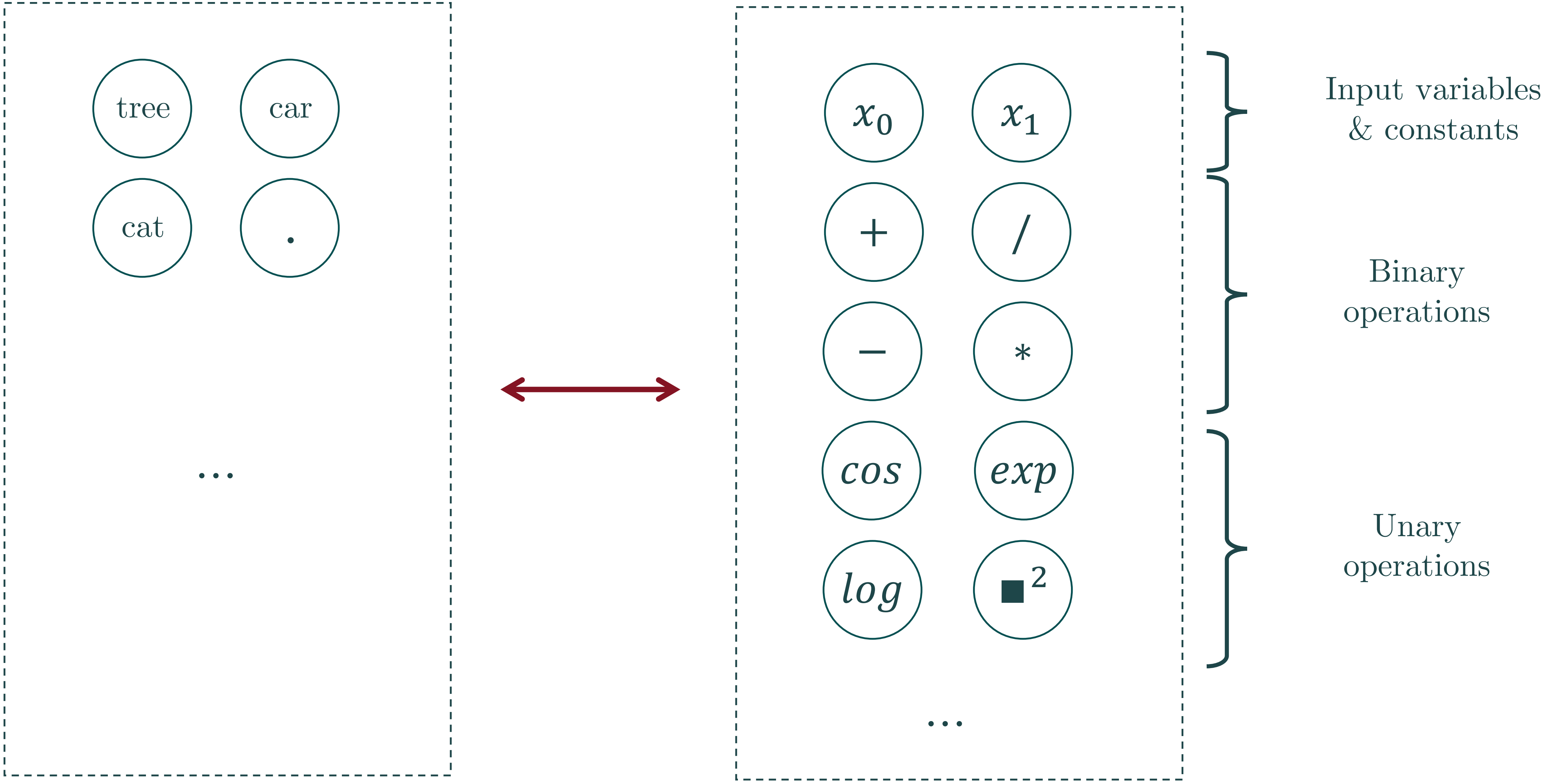
3. Generalization



Language processing for symbolic mathematics



Language processing for symbolic mathematics



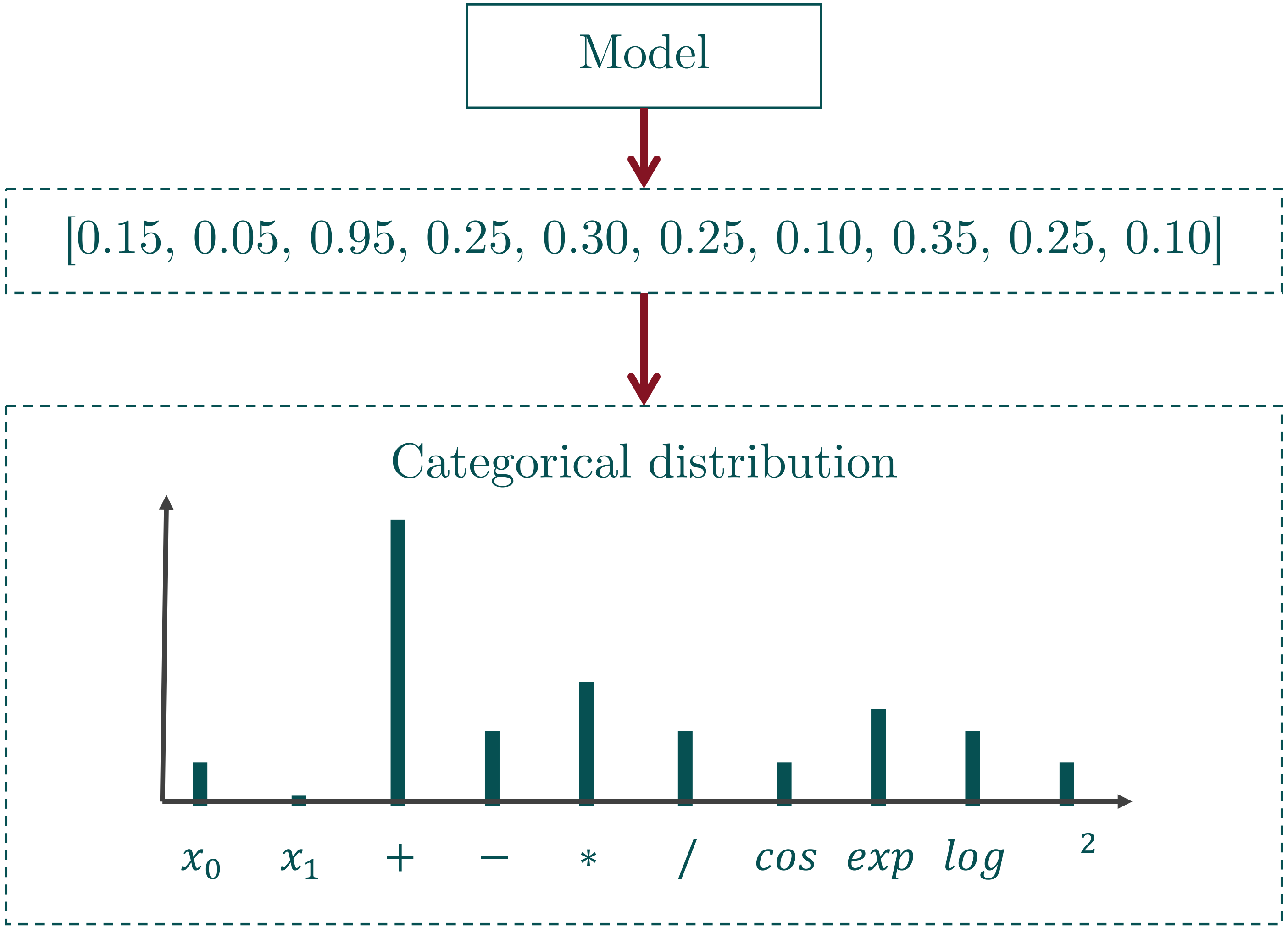
Embedding (1) : how to go from numbers to symbols ?

Model

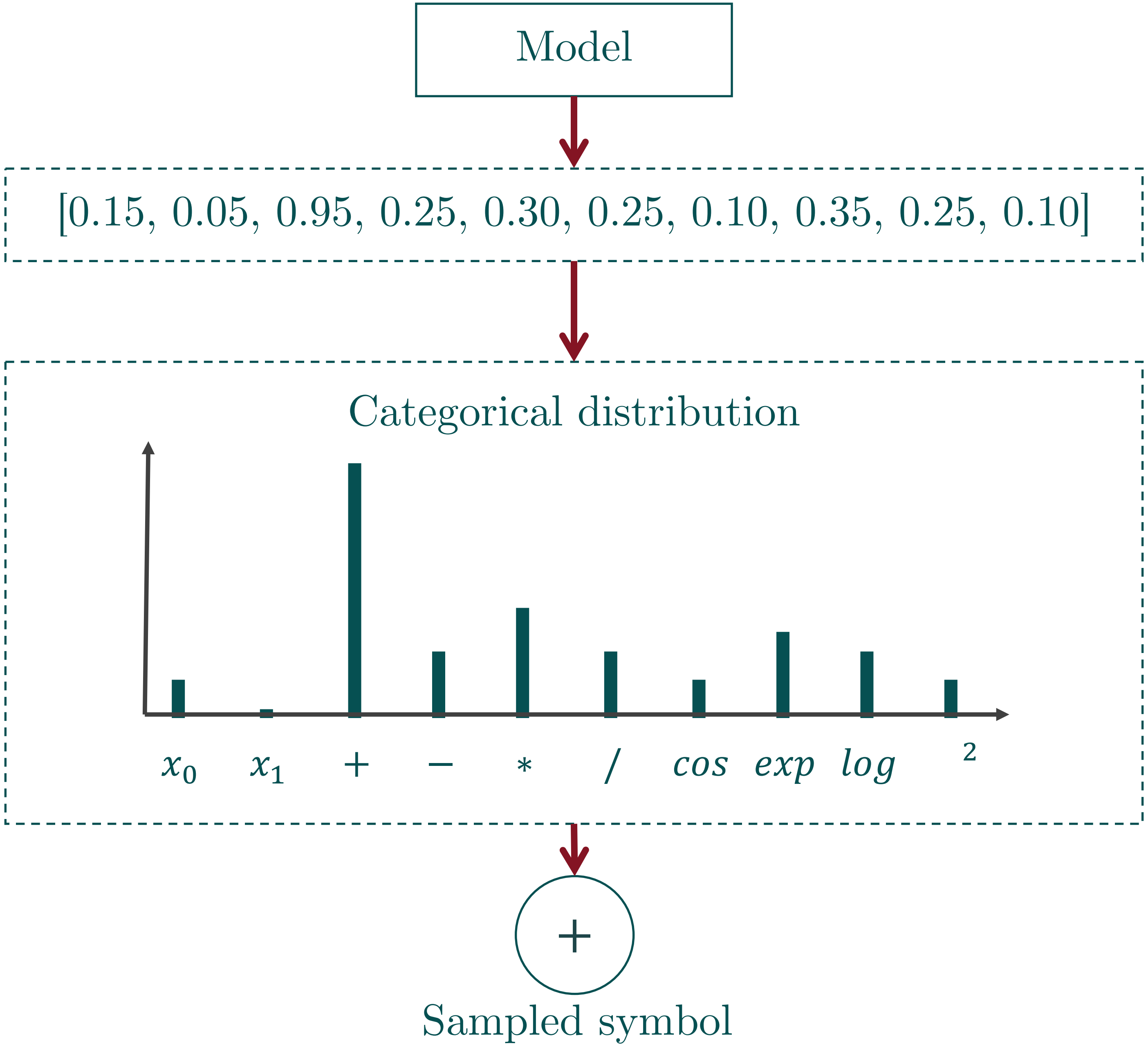
↓

[0.15, 0.05, 0.95, 0.25, 0.30, 0.25, 0.10, 0.35, 0.25, 0.10]

Embedding (1) : how to go from numbers to symbols ?



Embedding (1) : how to go from numbers to symbols ?



Embedding (2) : how to go from vector of symbols to expressions ?

Infix notation

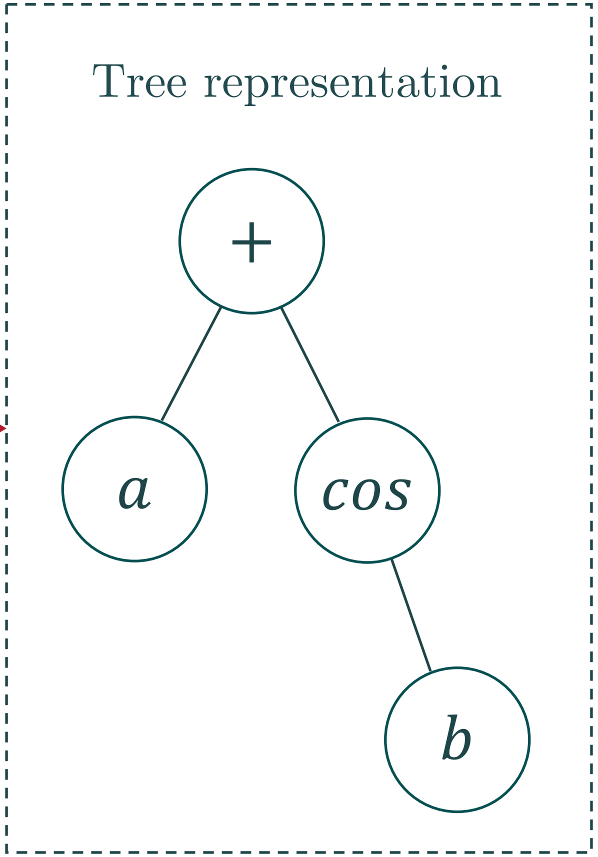
$$a + \cos(b)$$

Embedding (2) : how to go from vector of symbols to expressions ?

Infix notation

$a + \cos(b)$

*

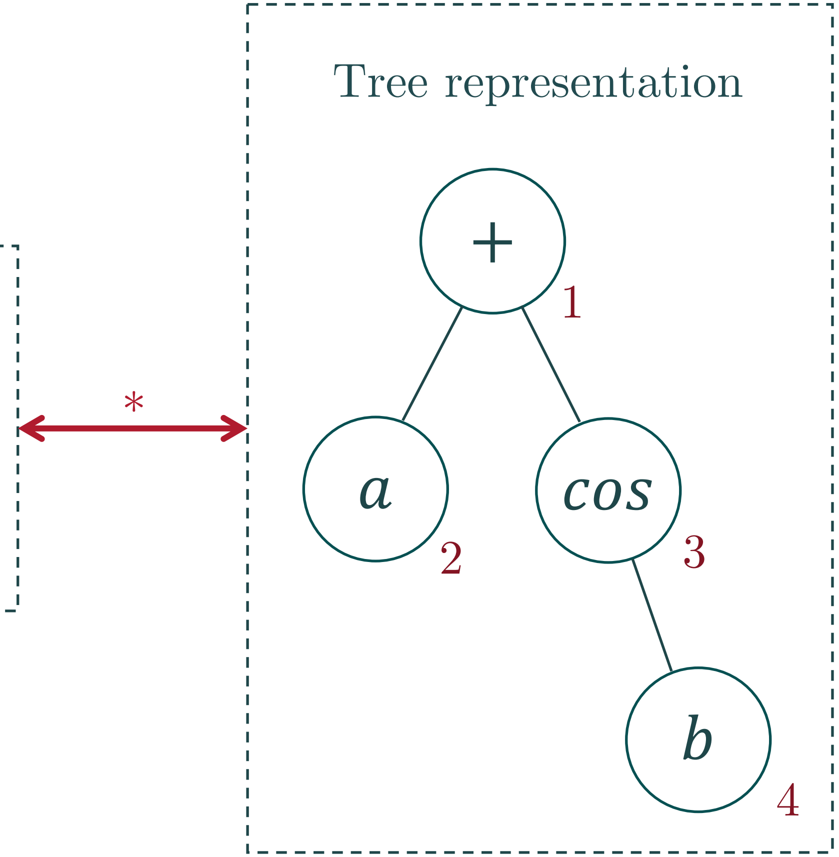


* 1:1 equivalence

Embedding (2) : how to go from vector of symbols to expressions ?

Infix notation

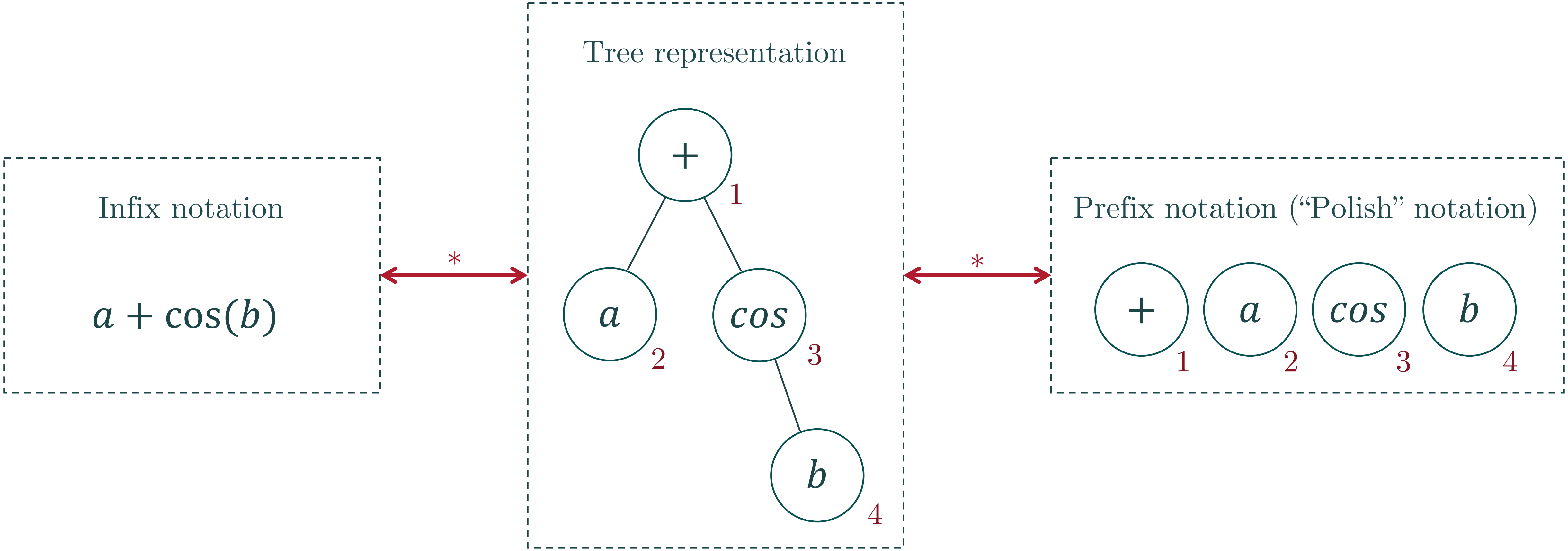
$a + \cos(b)$



Listing each node first in depth and then left to right

* 1:1 equivalence

Embedding (2) : how to go from vector of symbols to expressions ?



Listing each node first in depth and then left to right

* 1:1 equivalence

Embedding (3) : Encoding a whole expression

Library of possible symbols

- $+$ [**1**, 0, 0, 0, 0]
- log [0, **1**, 0, 0, 0]
- cos [0, 0, **1**, 0, 0]
- a [0, 0, 0, **1**, 0]
- b [0, 0, 0, 0, **1**]

* 1:1 equivalence

Embedding (3) : Encoding a whole expression

Library of possible symbols

- $+$ [**1**, 0, 0, 0, 0]
- log [0, **1**, 0, 0, 0]
- cos [0, 0, **1**, 0, 0]
- a [0, 0, 0, **1**, 0]
- b [0, 0, 0, 0, **1**]

$$a + \cos(b) \;\overset{*}{\longleftrightarrow}\; \begin{pmatrix} + & a & cos & b \\ [\mathbf{1}, 0, 0, 0, 0] & [0, 0, 0, \mathbf{1}, 0] & [0, 0, \mathbf{1}, 0, 0] & [0, 0, 0, 0, \mathbf{1}] \end{pmatrix}$$

* 1:1 equivalence

Embedding (4) : Arity

- $+$, $/$: Binary operations (arity = 2)
- log, cos : Unary operations (arity = 1)
- a, b : Terminal symbols (arity = 0)

		$+$		a		cos		b		
Total arity	=	2	+	0	+	1	+	0	=	3
Length	=	1	+	1	+	1	+	1	=	4

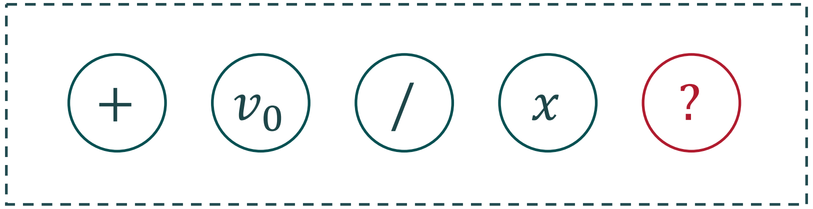
Embedding (4) : Arity

- $+$, $/$: Binary operations (arity = 2)
- log, cos : Unary operations (arity = 1)
- a, b : Terminal symbols (arity = 0)

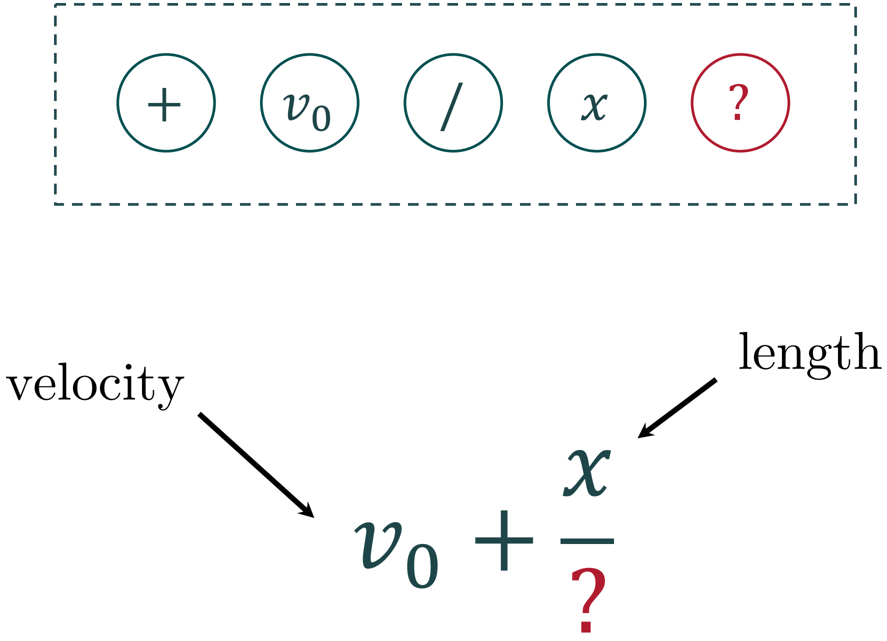
	$+$		a		cos		b		
Total arity =	2	+	0	+	1	+	0	=	3
Length =	1	+	1	+	1	+	1	=	4

An expression is valid $\Leftrightarrow$ length - (total arity) = 1

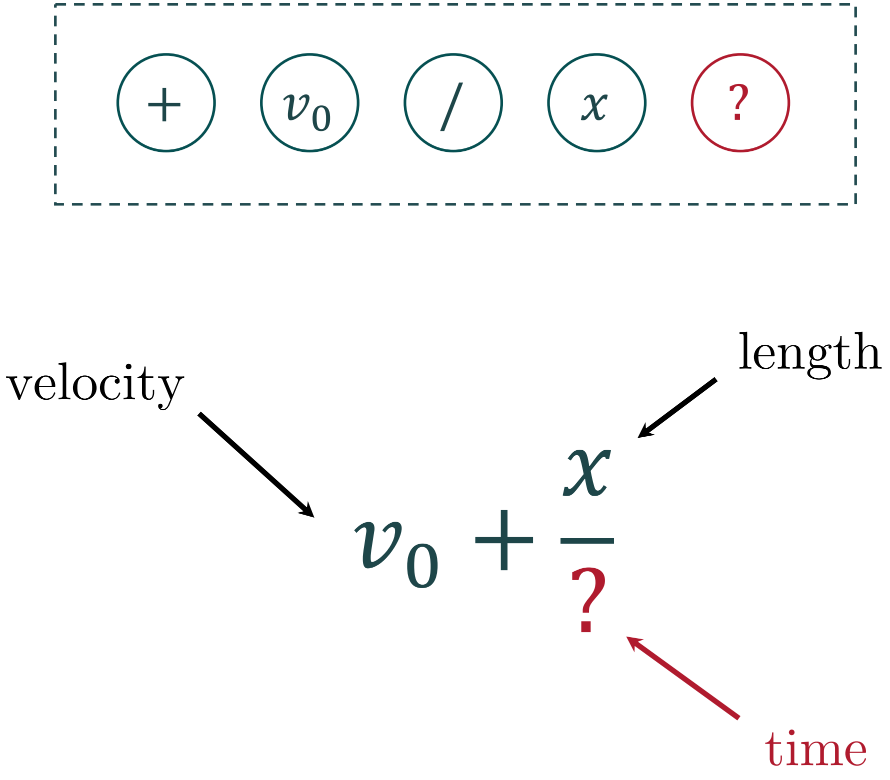
Dimensional analysis constrains the search space



Dimensional analysis constrains the search space

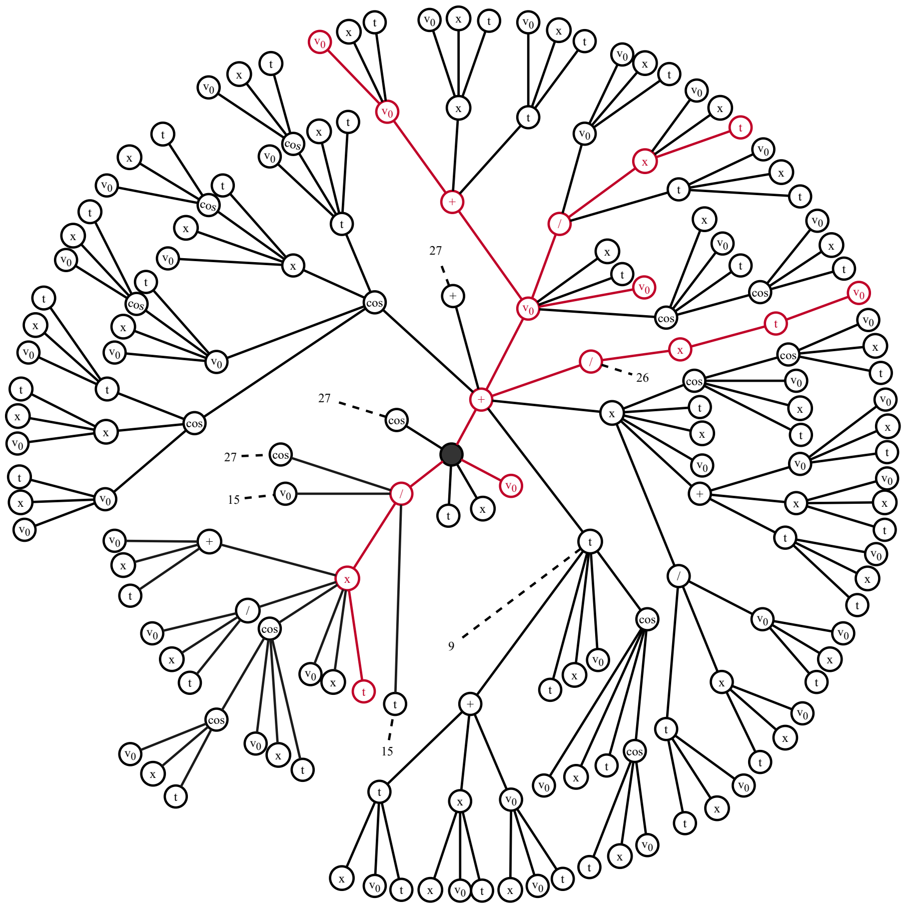


Dimensional analysis constrains the search space



Search space reduction using physical units constraints

268 expressions



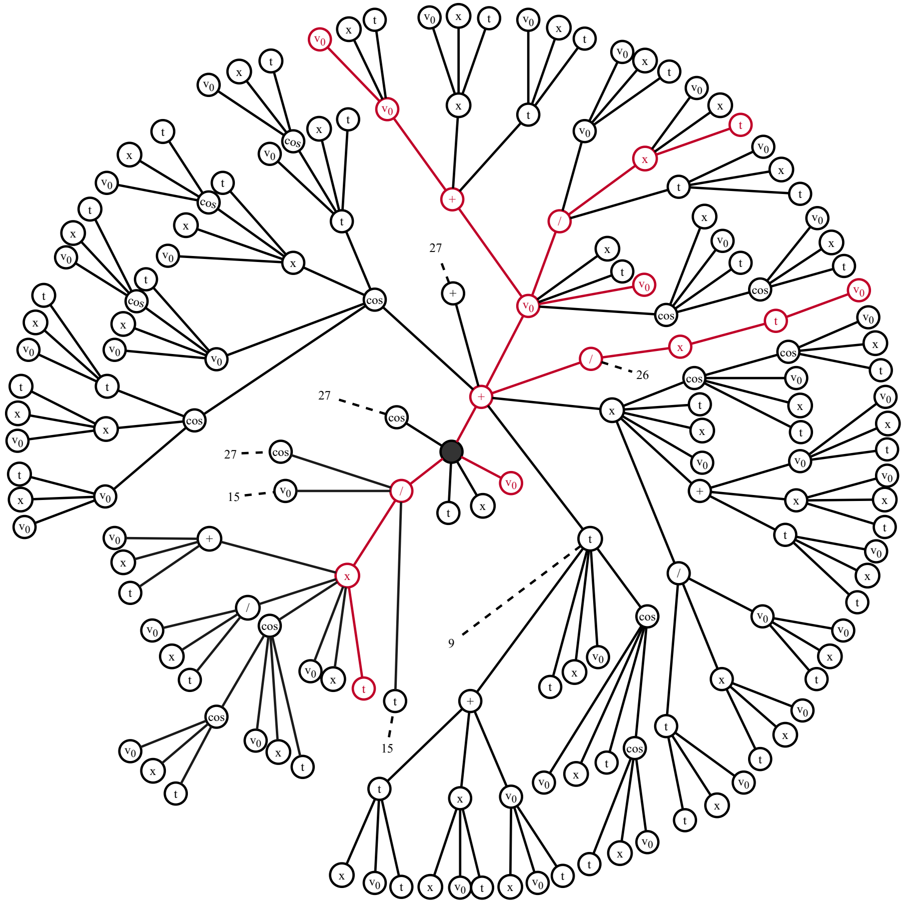
Search space

Prefix notation paths for expressing a velocity v using a library of symbols {+, /, cos, v_0, x, t} where v_0 is a velocity, x is a length, and t is a time (length < 5 for readability).

[Tenachi et al 2023] (this work)

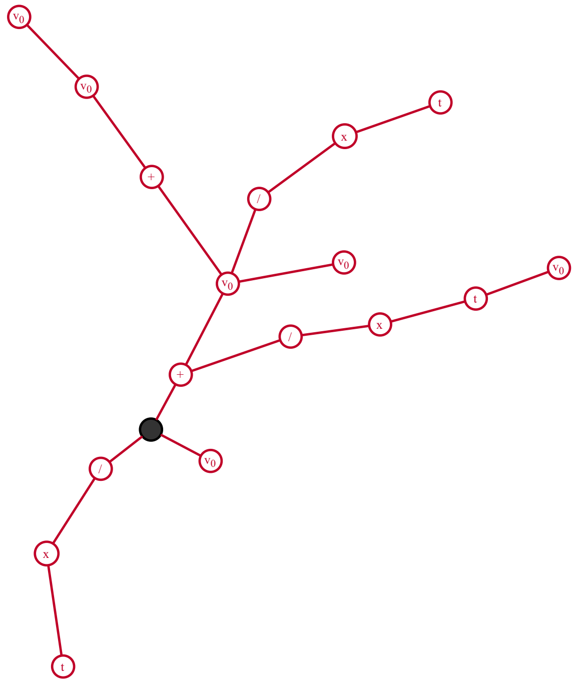
Search space reduction using physical units constraints

268 expressions



Search space

6 expressions



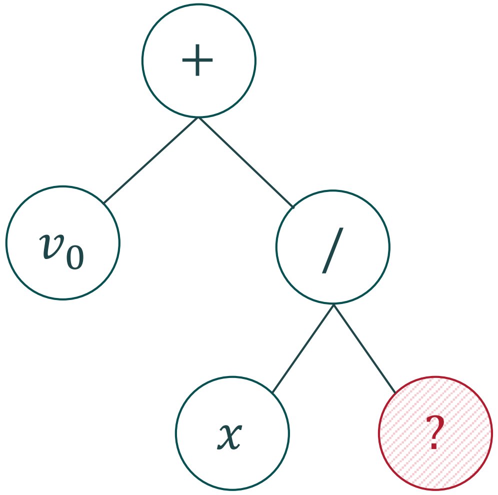
Search space with our *in situ* physical units prior

Prefix notation paths for expressing a velocity v using a library of symbols {+, /, cos, v_0, x, t} where v_0 is a velocity, x is a length, and t is a time (length $<$ 6 for readability).

[Tenachi et al 2023] (this work)

In situ prior

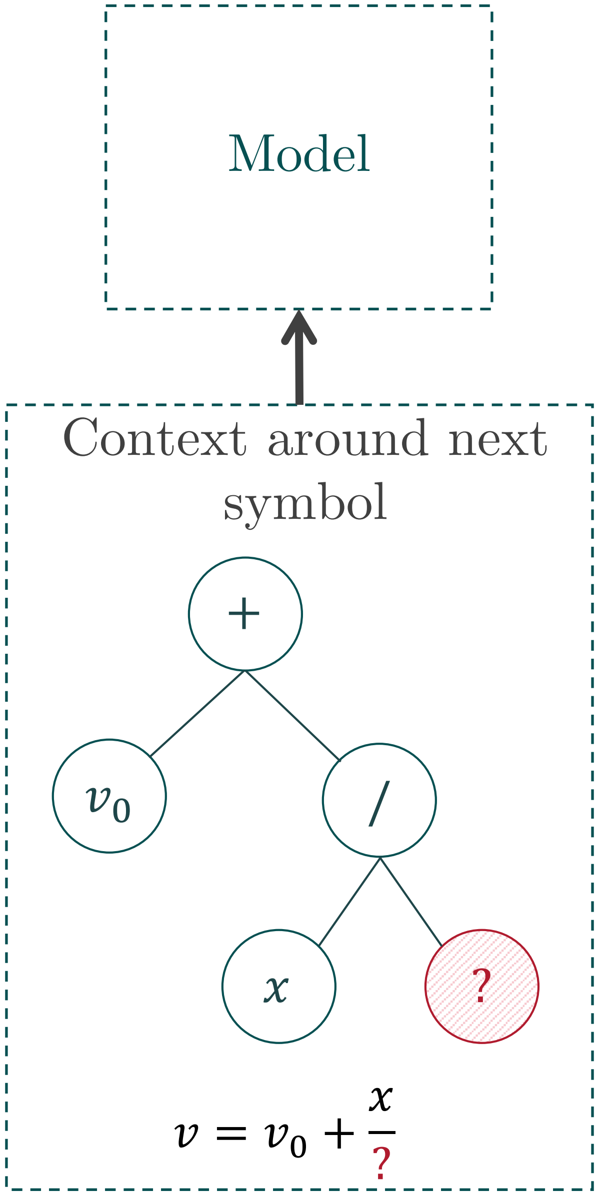
Context around next symbol



$$v = v_0 + \frac{x}{?}$$

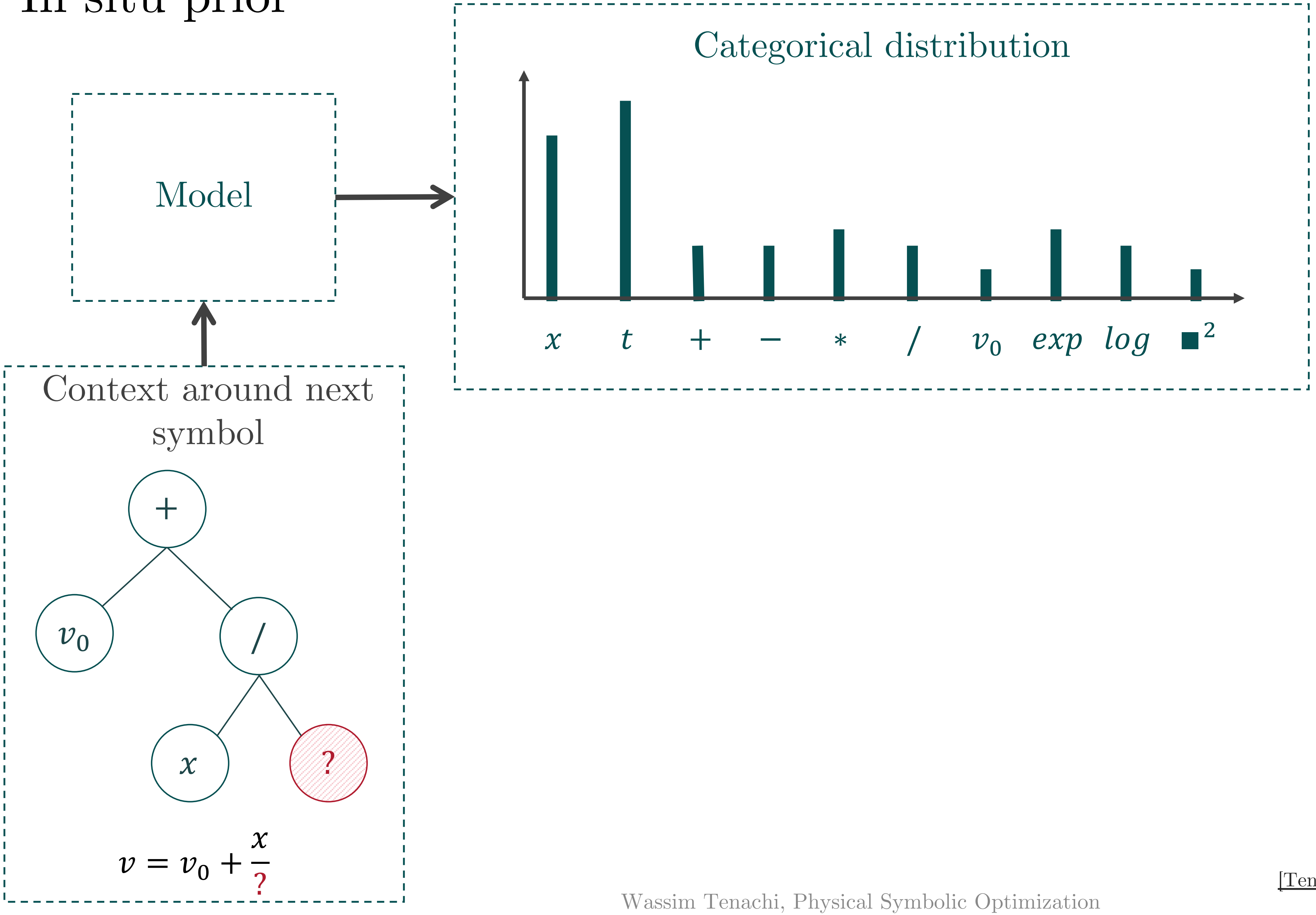
[Tenachi et al 2023] (this work)

In situ prior



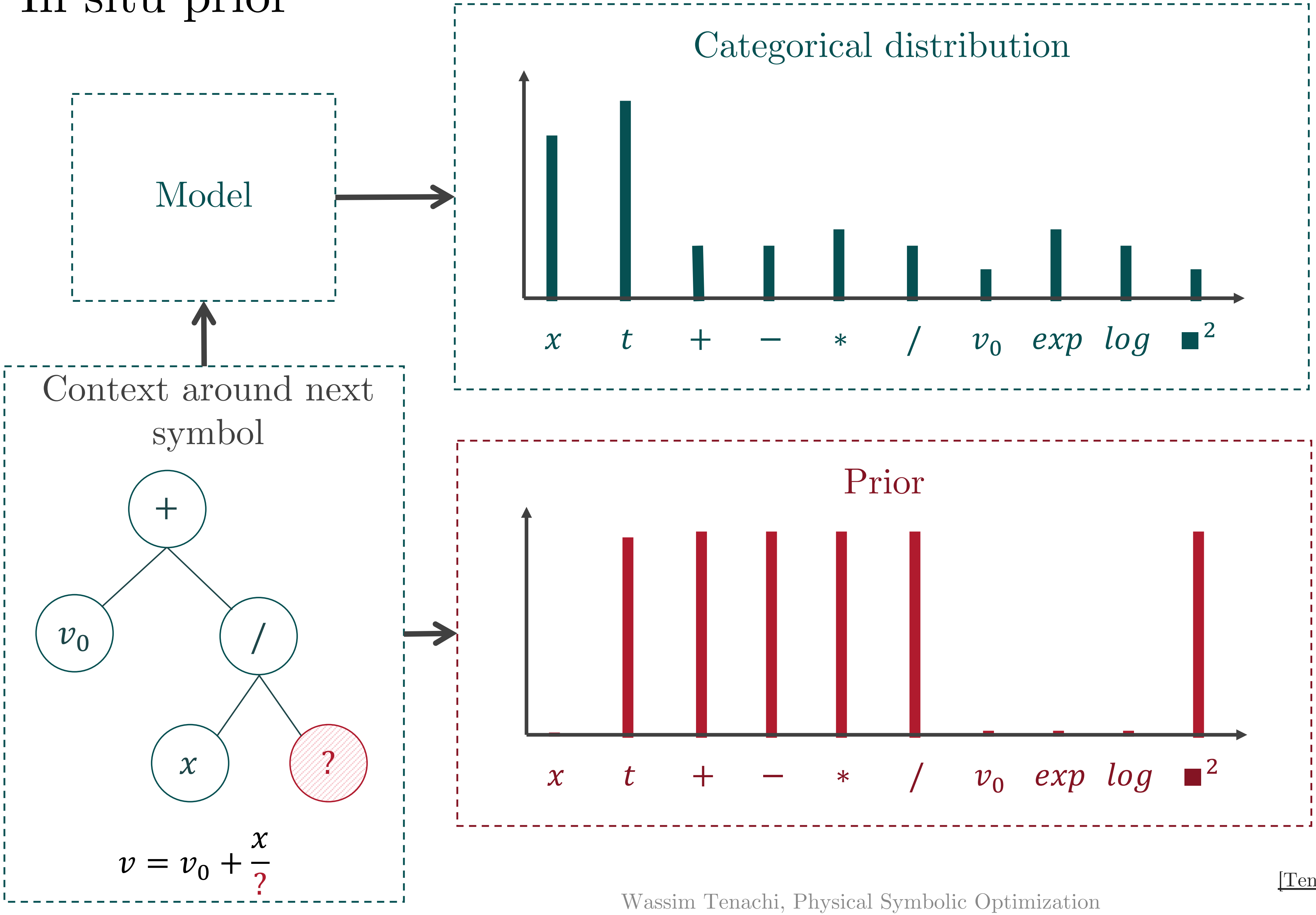
[Tenachi et al 2023] (this work)

In situ prior



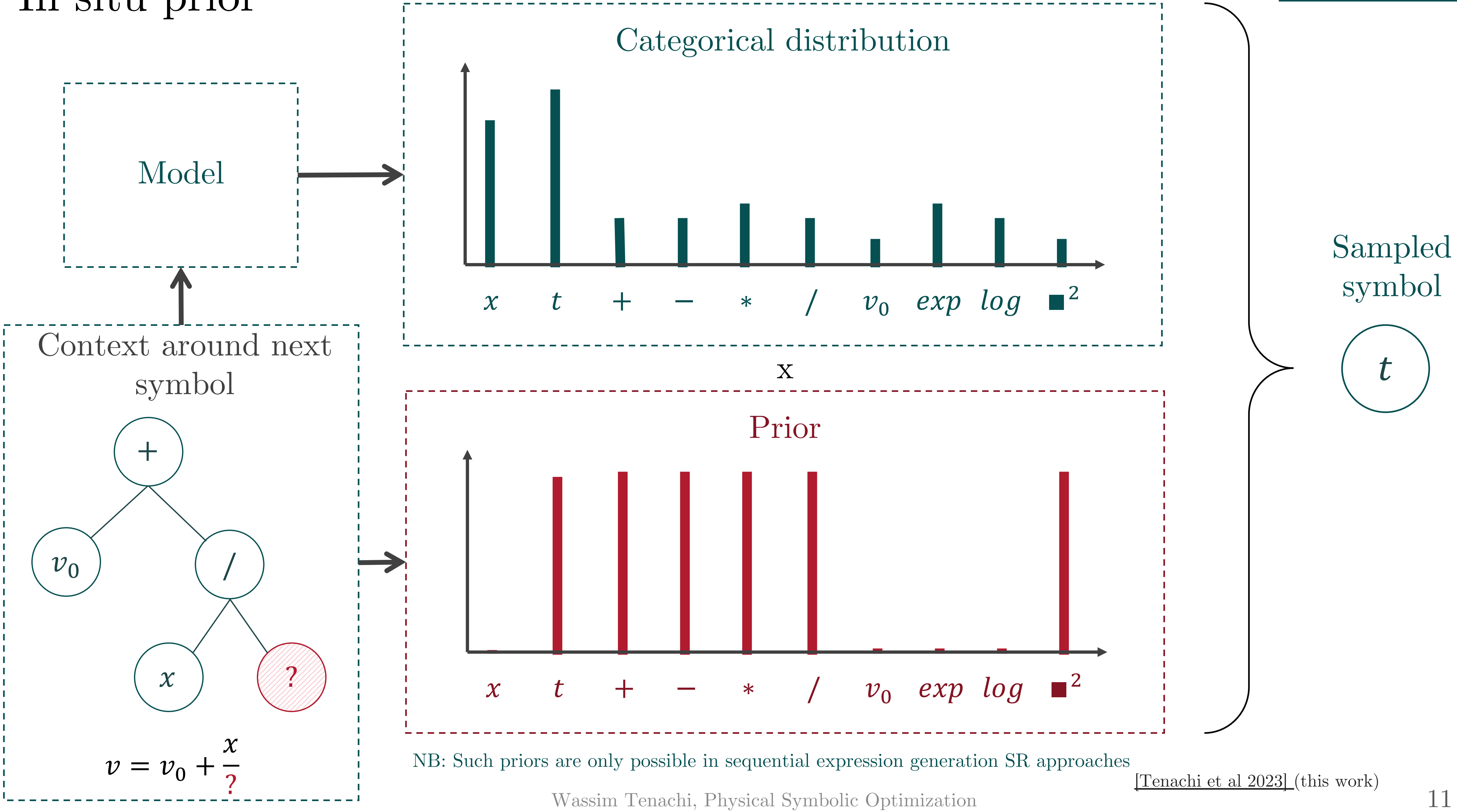
[Tenachi et al 2023] (this work)

In situ prior



[Tenachi et al 2023] (this work)

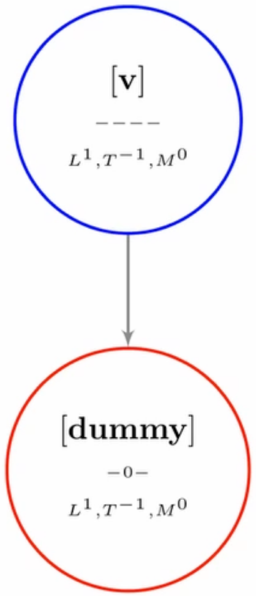
In situ prior



NB: Such priors are only possible in sequential expression generation SR approaches

[Tenachi et al 2023] (this work)

Propagating units constraints (1)

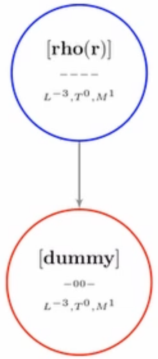


$$v = \square$$

https://youtu.be/clZsLj3oPy8

[Tenachi et al 2023] (this work)

Propagating units constraints (2)



$$\rho(r) = \square$$

https://youtu.be/U_hnFJuZ1dA

[Tenachi et al 2023] (this work)

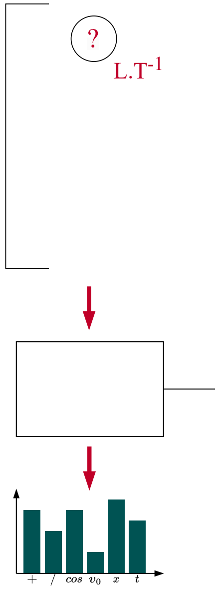
Symbolic function generation

Contextual information around next token

- parent and units
- sibling and units
- previous token and units
- current required units
- # dangling

RNN

Categorical distribution



Expression

$v = ■$

Library of choosable tokens

$[\ +\ \ /\ \ cos\ \ v_0\ \ x\ \ t\ ]$

Symbolic function generation

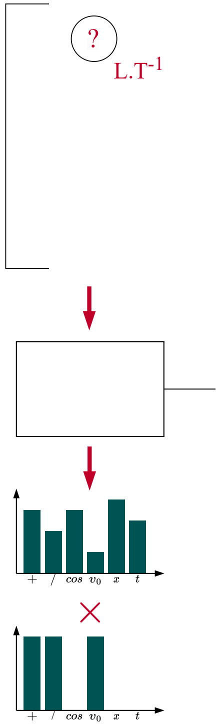
Contextual information around next token

- parent and units
- sibling and units
- previous token and units
- current required units
- # dangling

RNN

Categorical distribution

Physical units prior (+ other priors)



Expression

$v = \blacksquare$

Library of choosable tokens

$[\ +,\ /,\ cos,\ v_0,\ x,\ t\ ]$

Symbolic function generation

Contextual information
around next token

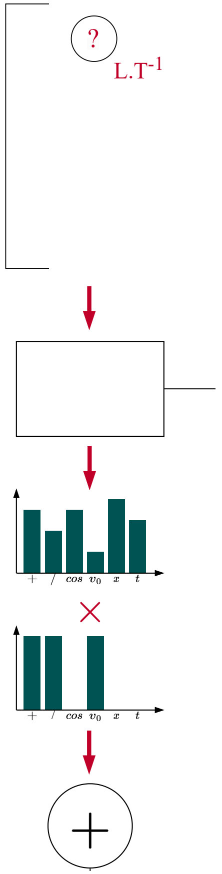
- parent and units
- sibling and units
- previous token and units
- current required units
- # dangling

RNN

Categorical
distribution

Physical units prior
(+ other priors)

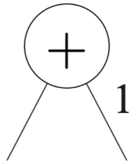
Sampled
token



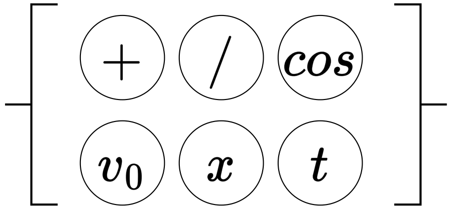
Expression

$$v = \blacksquare + \blacksquare$$

Expression tree



Library of choosable tokens



Symbolic function generation

Contextual information around next token

- parent and units
- sibling and units
- previous token and units

current required units

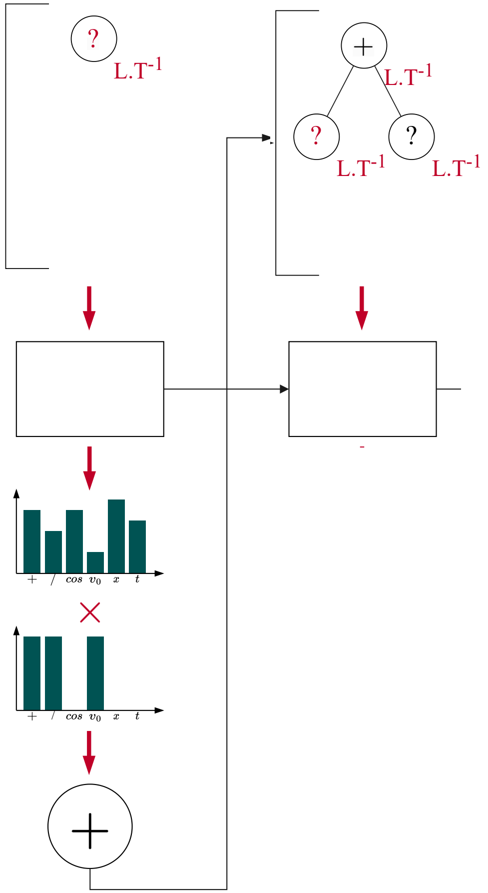
dangling

RNN

Categorical distribution

Physical units prior (+ other priors)

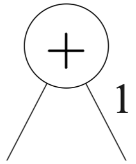
Sampled token



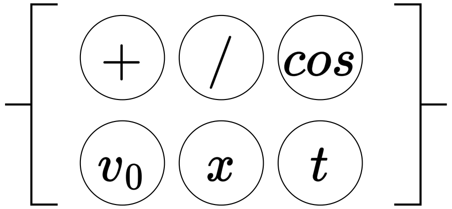
Expression

$v = \blacksquare + \blacksquare$

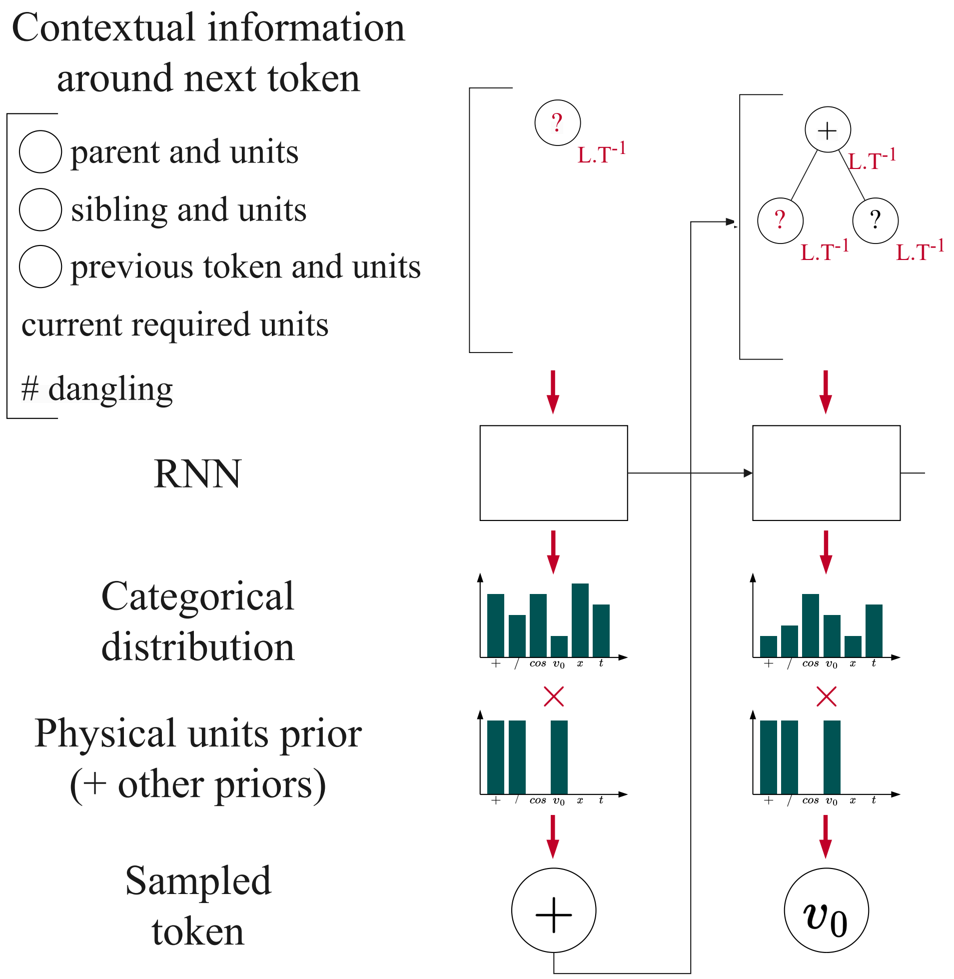
Expression tree



Library of choosable tokens



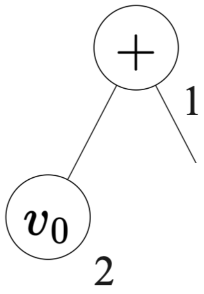
Symbolic function generation



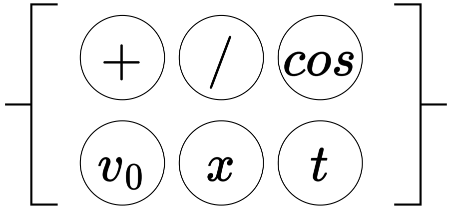
Expression

$$v = v_0 + \blacksquare$$

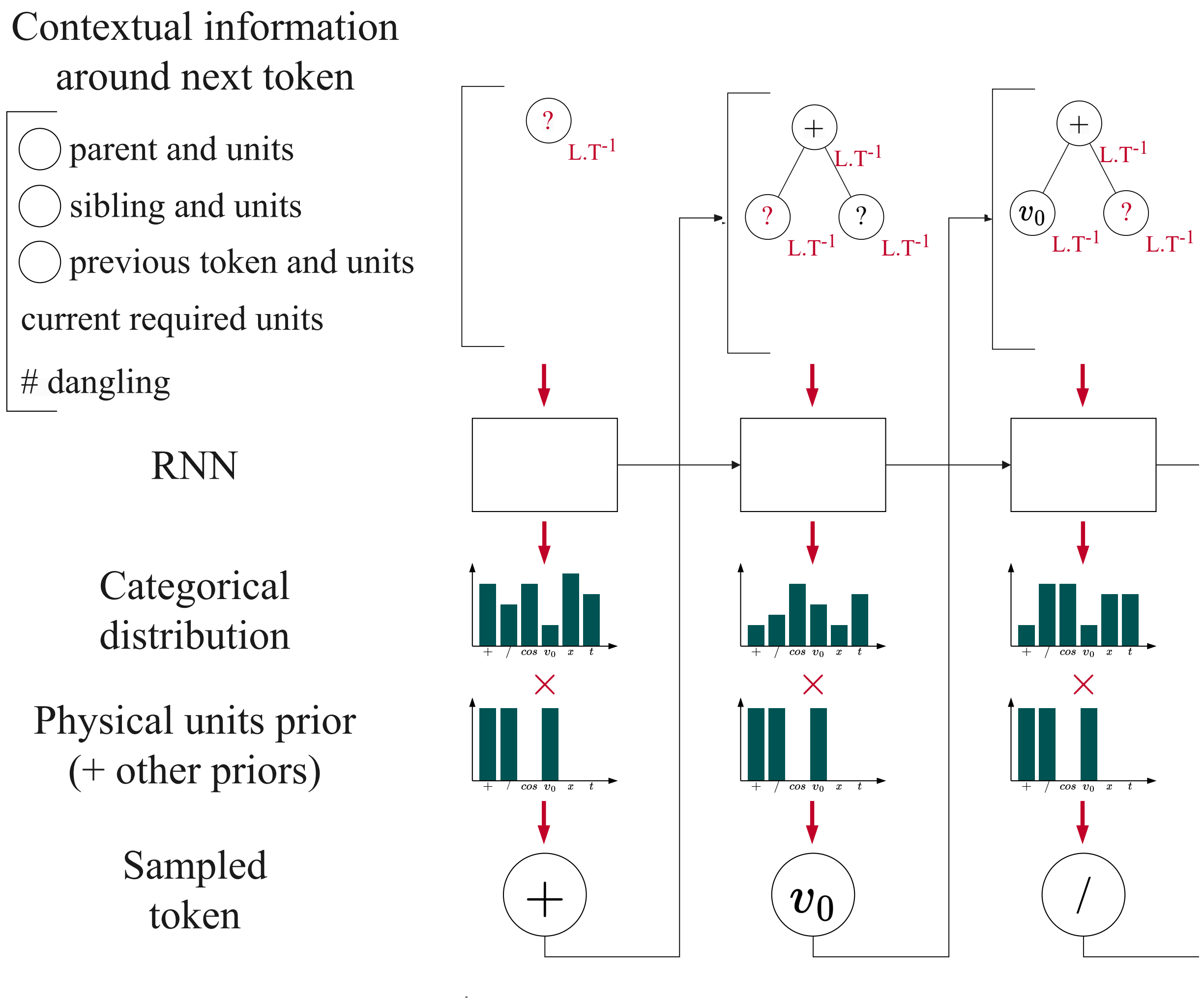
Expression tree

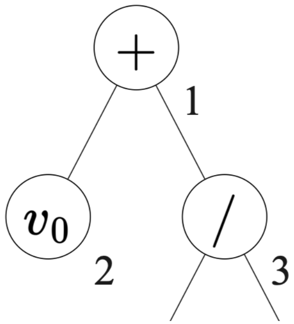


Library of choosable tokens

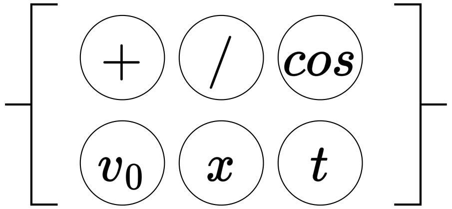


Symbolic function generation

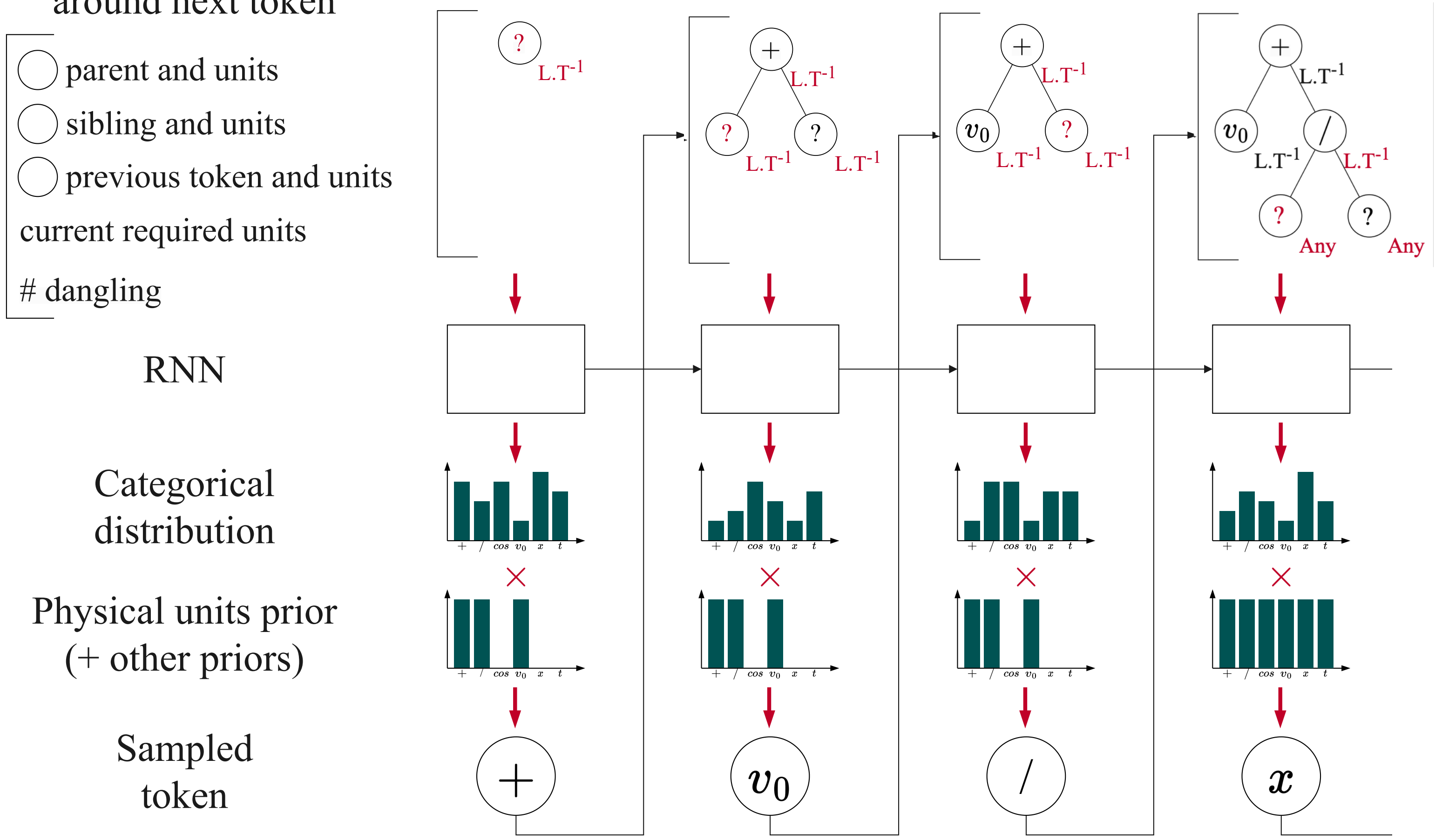




$$v = v_0 + \frac{\blacksquare}{\blacksquare}$$



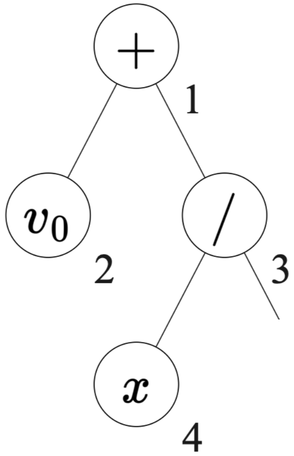
Symbolic function generation



Expression

$$v = v_0 + \frac{x}{\blacksquare}$$

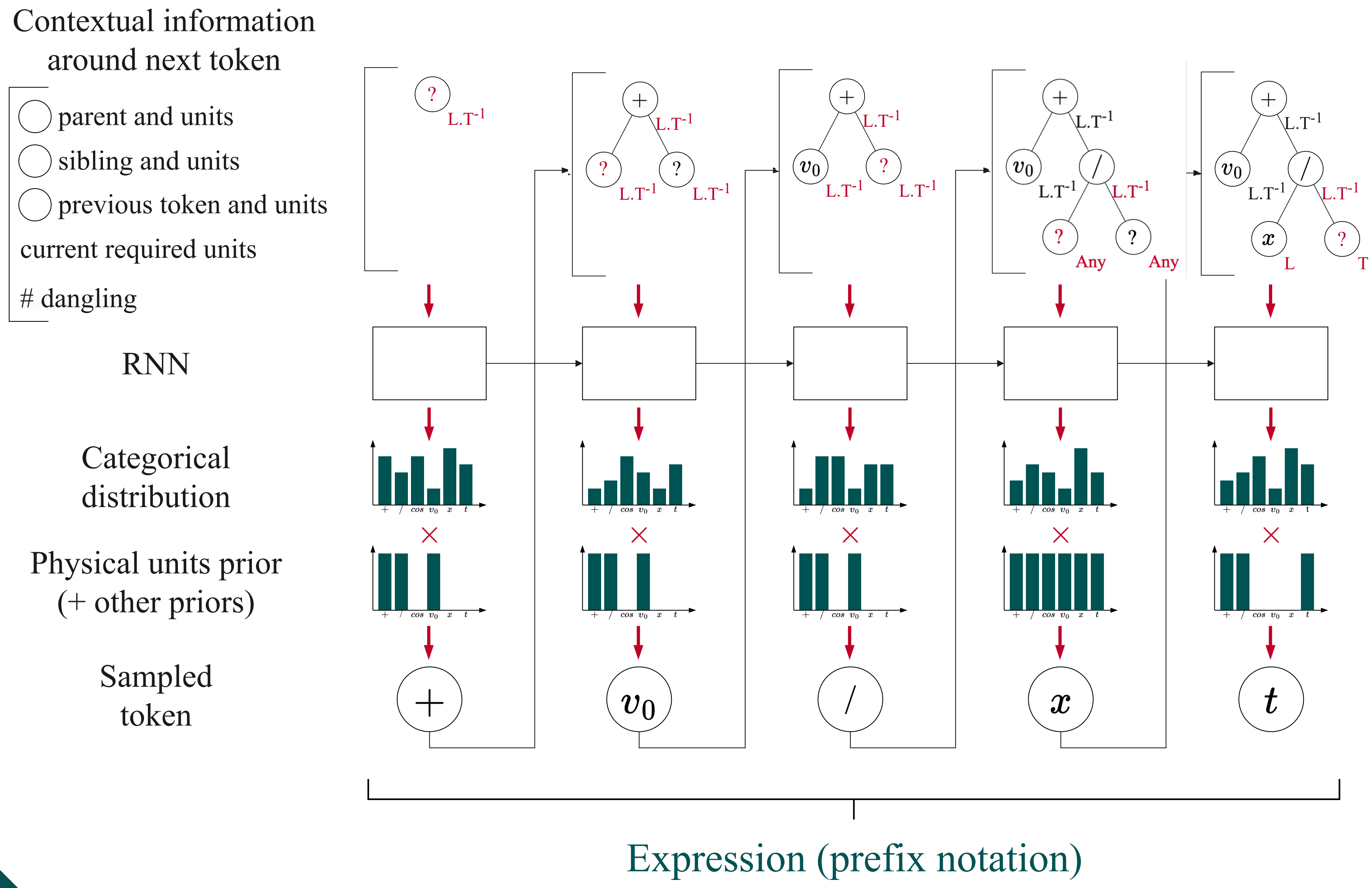
Expression tree



Library of choosable tokens

$$\left[+, /, cos, v_0, x, t \right]$$

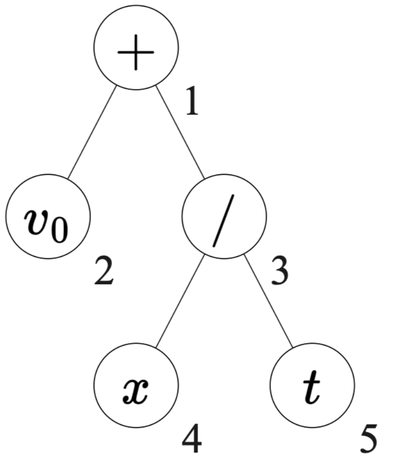
Symbolic function generation



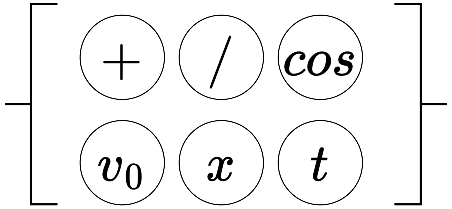
Expression

$$v = v_0 + \frac{x}{t}$$

Expression tree



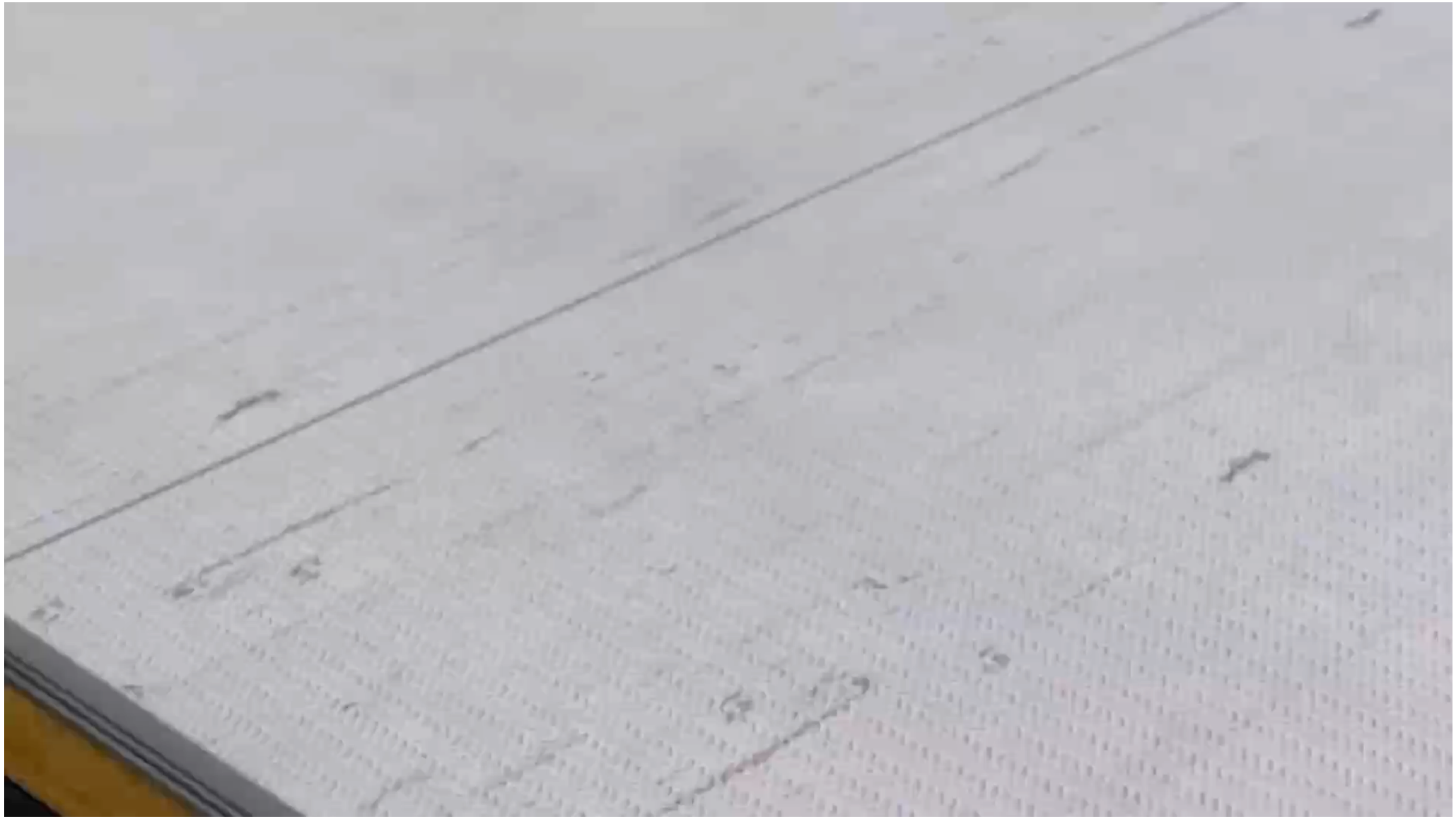
Library of choosable tokens



Reinforcement learning in a nutshell (1)

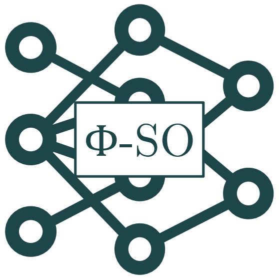
https://youtu.be/spfpBrBjntg

Reinforcement learning in a nutshell (2)



https://youtu.be/igZ6IPQimjQ

Reinforcement learning & risk seeking policy



Categorical distributions

$v = v_0 + x/t$

$v = v_0$

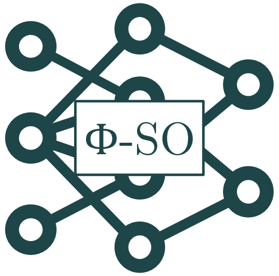
$v = v_0 - x/t$

$v = x/t$

$\vdots$

$$v = \frac{v_0^2}{x/t}$$

Reinforcement learning & risk seeking policy



Categorical distributions

Candidates vs data → reward

$v = v_0 + x/t$ → R = 0.99

$v = v_0$ → R = 0.93

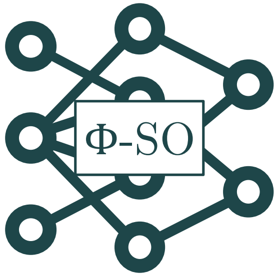
$v = v_0 - x/t$ → R = 0.73

$v = x/t$ → R = 0.28

⋮

$v = \frac{v_0^2}{x/t}$ → R = 0.64

Reinforcement learning & risk seeking policy



Categorical distributions

Candidates vs data → reward

$v = v_0 + x/t$	→ R = 0.99
$v = v_0$	→ R = 0.93
$v = v_0 - x/t$	→ R = 0.73
$v = x/t$	→ R = 0.28
⋮	
$v = \frac{v_0^2}{x/t}$	→ R = 0.64

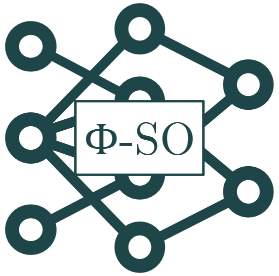
Reinforce on the 5% best candidates

Not punished/rewarded on 95% of candidates → "risk seeking"

[Petersen et al 2019]

Reinforcement learning & risk seeking policy

↳ Black box reward function



Categorical distributions

	Candidates vs data → reward
$v = v_0 + x/t$	→ R = 0.99
$v = v_0$	→ R = 0.93
$v = v_0 - x/t$	→ R = 0.73
$v = x/t$	→ R = 0.28
⋮	
$v = \frac{v_0^2}{x/t}$	→ R = 0.64

Reinforce on the 5% best candidates

Not punished/rewarded on 95% of candidates → "risk seeking"

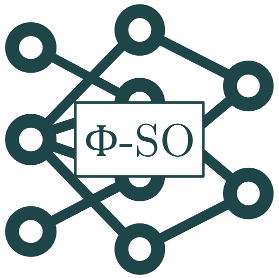
[Petersen et al 2019]

Reward → no auto-differentiation
(unlike most ML methods used in physics)

→ We can apply any physical constraints we want even if it is not differentiable

Reinforcement learning & risk seeking policy

↳ Black box reward function



Categorical distributions

Candidates vs data → reward

$v = v_0 + x/t$	→ R = 0.99
$v = v_0$	→ R = 0.93
$v = v_0 - x/t$	→ R = 0.73
$v = x/t$	→ R = 0.28
⋮	
$v = \frac{v_0^2}{x/t}$	→ R = 0.64

Reinforce on the 5% best candidates

Not punished/rewarded on 95% of candidates → "risk seeking"

[Petersen et al 2019]

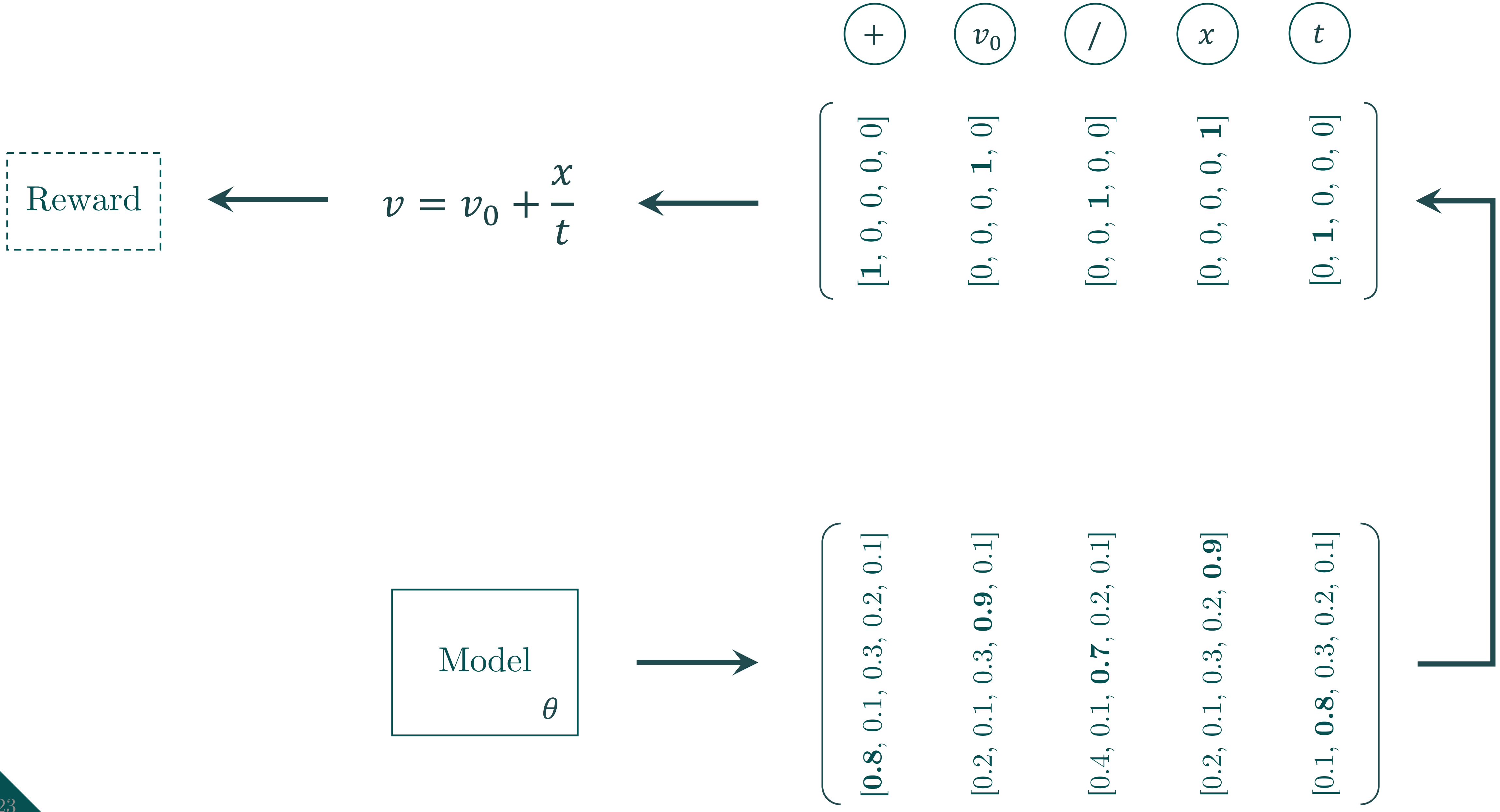
Reward → no auto-differentiation
(unlike most ML methods used in physics)

→ We can apply any physical constraints we want even if it is not differentiable

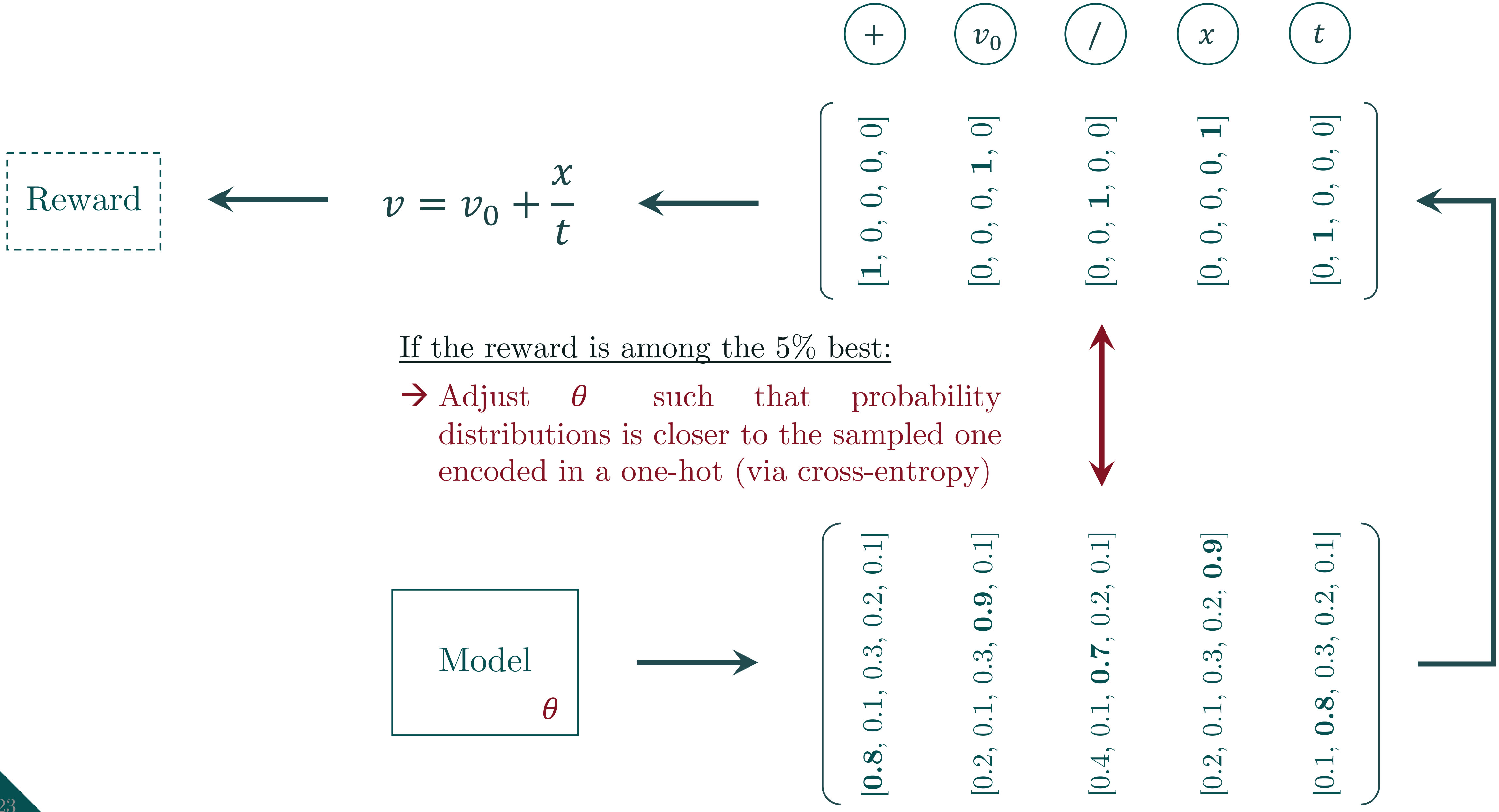
- →Complexity (Occam's razor)
- →Symmetries
- →Constraints on derivatives/primitives
- →Symbolic computing using Sympy/Mathematica
- →Fitness in ODEs, limits values
- →Behavior in N-body simulation

...

Model loss : how do we reinforce ?



Model loss : how do we reinforce ?

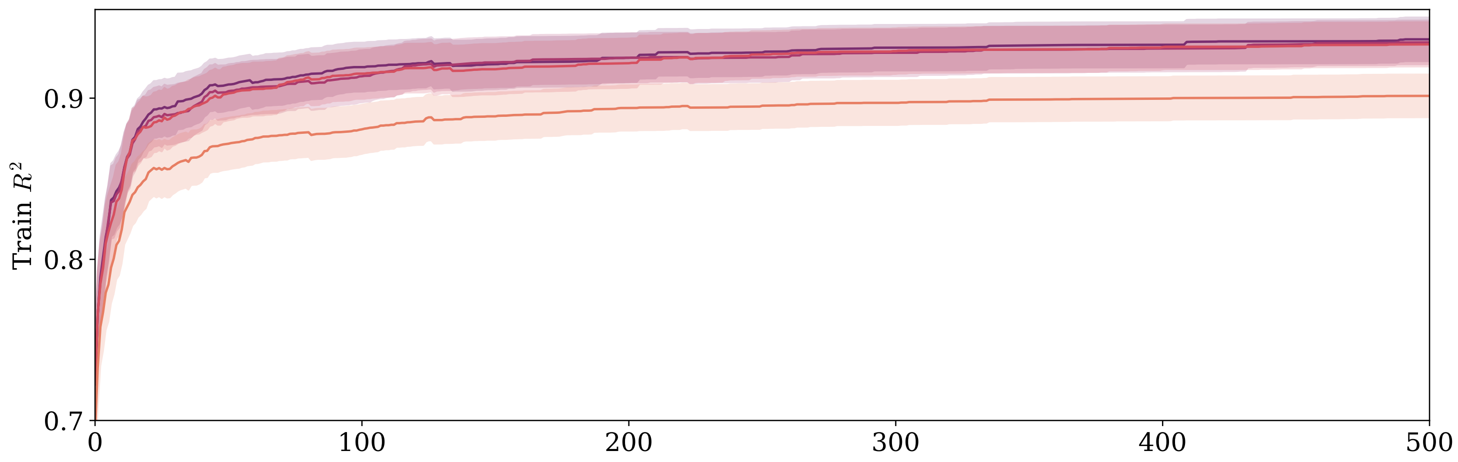


If the reward is among the 5% best:

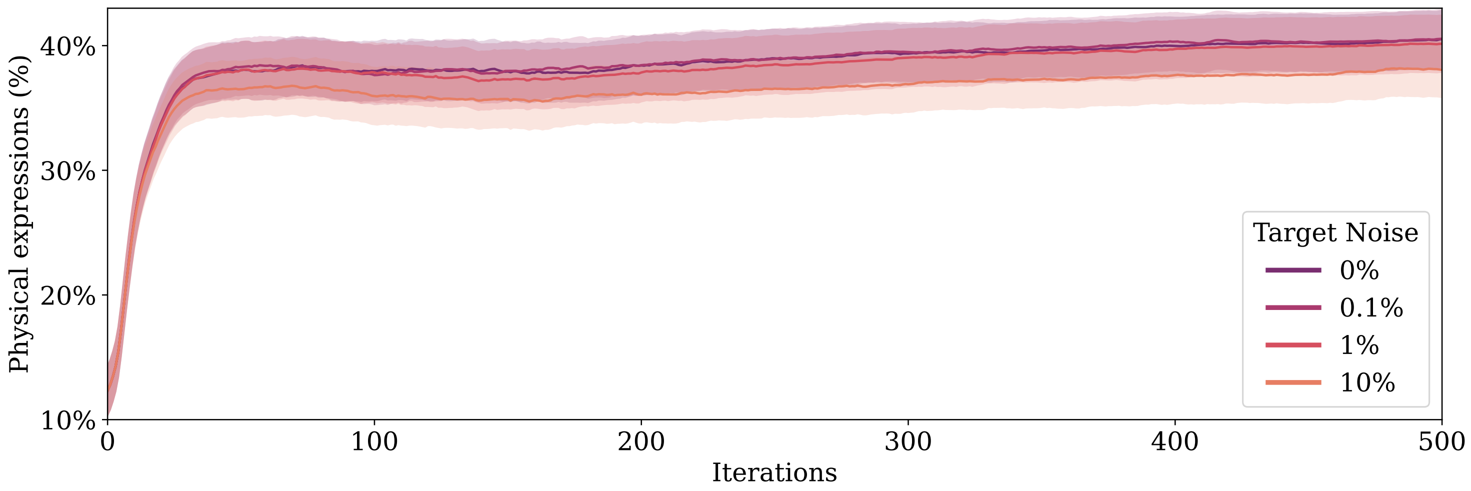
→ Adjust θ such that probability distributions is closer to the sampled one encoded in a one-hot (via cross-entropy)

Learning curves

Learning to produce **accurate** expressions

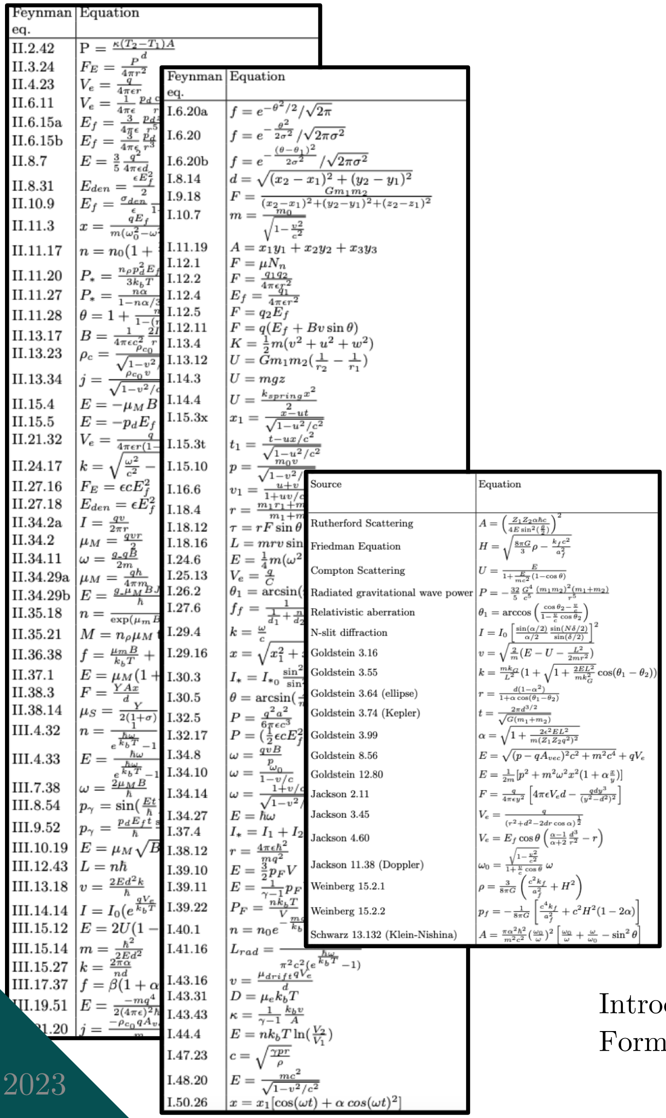


Learning to produce **physical** expressions



Feynman Benchmark

120 equations from the Feynman Lectures on Physics (+ other physics textbooks)



Source	Equation
Rutherford Scattering	$A = \left(\frac{Z_1 Z_2 \alpha \hbar c}{4E \sin^2(\frac{\theta}{2})}\right)^2$
Friedman Equation	$H = \sqrt{\frac{8\pi G}{3}\rho - \frac{k_f c^2}{a_f^2}}$
Compton Scattering	$U = \frac{E}{1+\frac{E}{mc^2}(1-\cos\theta)}$
Radiated gravitational wave power	$P = -\frac{32}{5}\frac{G^4}{c^5}\frac{(m_1 m_2)^2(m_1+m_2)}{r^5}$
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	$\alpha = \sqrt{1 + \frac{2\epsilon^2 EL^2}{m(Z_1 Z_2 q^2)^2}}$
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	$F = \frac{q}{4\pi\epsilon y^2}\left[4\pi\epsilon V_e d - \frac{qdy^3}{(y^2-d^2)^2}\right]$
Jackson 3.45	$V_e = \frac{q}{(r^2+d^2-2dr\cos\alpha)^{\frac{1}{2}}}$
Jackson 4.60	$V_e = E_f\cos\theta\left(\frac{\alpha-1}{\alpha+2}\frac{d^3}{r^2} - r\right)$
Jackson 11.38 (Doppler)	$\omega_0 = \frac{\sqrt{1-\frac{v^2}{c^2}}}{1+\frac{v}{c}\cos\theta}\omega$
Weinberg 15.2.1	$\rho = \frac{3}{8\pi G}\left(\frac{c^2 k_f}{a_f^2} + H^2\right)$
Weinberg 15.2.2	$p_f = -\frac{1}{8\pi G}\left[\frac{c^4 k_f}{a_f^2} + c^2H^2(1-2\alpha)\right]$
Schwarz 13.132 (Klein-Nishina)	$A = \frac{\pi\alpha^2\hbar^2}{m^2c^2}(\frac{\omega_0}{\omega})^2\left[\frac{\omega_0}{\omega} + \frac{\omega}{\omega_0} - \sin^2\theta\right]$

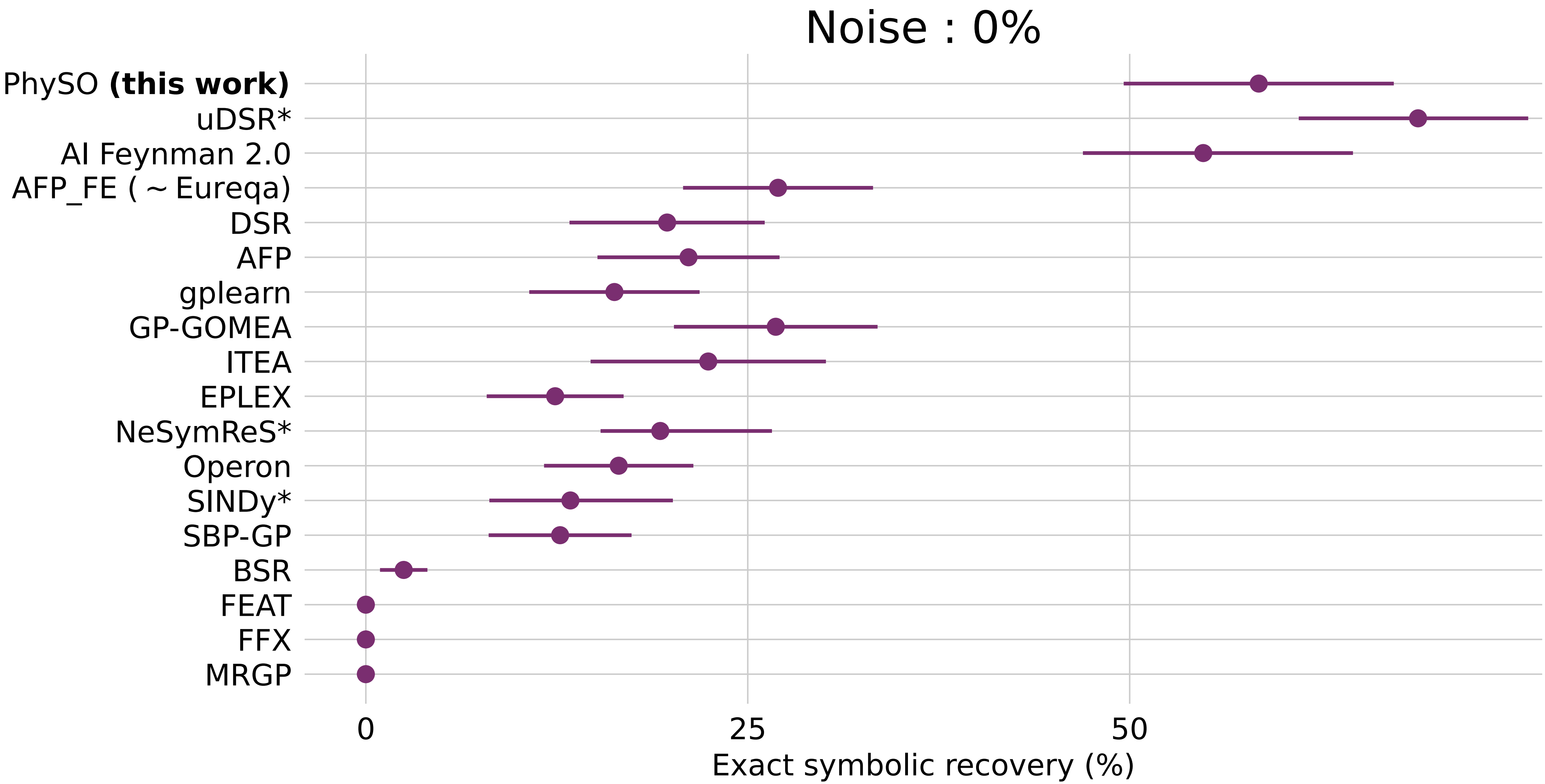
Against 17 other algorithms

Method	Technique(s)	Description	Reference
PhySO	RL, DA	Physical Symbolic Optimization	This work
uDSR	RL, GP, Simp., Sup.	A Unified Framework for Deep Symbolic Regression	Landajuela et al. (2022)
AIFeynman 2.0	Simp., DA	Symbolic regression exploiting graph modularity	Udrescu et al. (2020)
AFP_FE	GP	AFP with co-evolved fitness estimates, Eureqa-esque	Schmidt & Lipson (2009)
DSR	RL	Deep Symbolic Regression	Petersen et al. (2021a)
AFP	GP	Age-fitness Pareto Optimization	Schmidt & Lipson (2011)
gplearn	GP	Koza-style symbolic regression in Python	Stephens (2015)
GP-GOMEA	GP	GP-Optimal Mixing Evolutionary Algorithm	Virgolin et al. (2021)
ITEA	GP	Interaction-Transformation EA	de Franca & Aldeia (2021)
EPLEX	GP	ϵ-lexicase selection	La Cava et al. (2019)
NeSymReS	Sup.	Neural Symbolic Regression that Scales	Biggio et al. (2021)
Operon	GP	SR with Non-linear least squares	Kommenda et al. (2020)
SINDy	NeuroSym	Sparse identification of non-linear dynamics	Brunton et al. (2016)
SBP-GP	GP	Semantic Back-propagation Genetic Programming	Virgolin et al. (2019)
BSR	MCMC	Bayesian Symbolic Regression	Jin et al. (2019)
FEAT	GP	Feature Engineering Automation Tool	Cava et al. (2019)
FFX	Rand.	Fast function extraction	McConaghy (2011)
MRGP	GP	Multiple Regression Genetic Programming	Arnaldo et al. (2014)

Table 3. Summary of baseline symbolic regression methods along with the the underlying techniques they rely on: reinforcement learning (RL), genetic programming (GP), problem simplification schemes (Simp.), end-to-end supervised learning (Sup.), dimensional analysis (DA), neuro-symbolic / auto-differentiation based sparse fitting techniques (NeuroSym), Markov chain Monte Carlo (MCMC) and random search (Rand.).

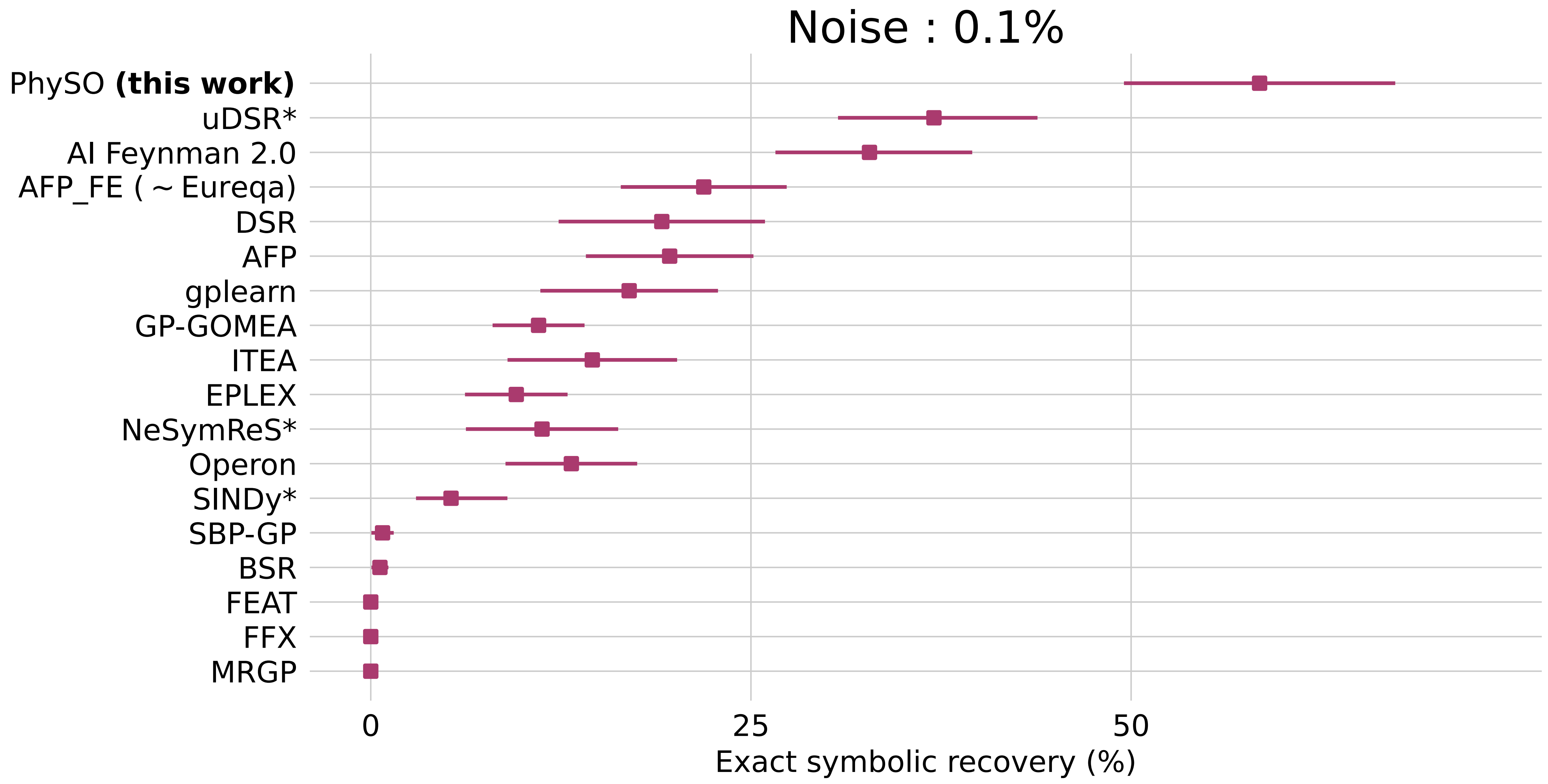
Introduced in [Udrescu & Tegmark 2020]
Formalized benchmark in SRBench [LaCava et al 2021]

Feynman Benchmark



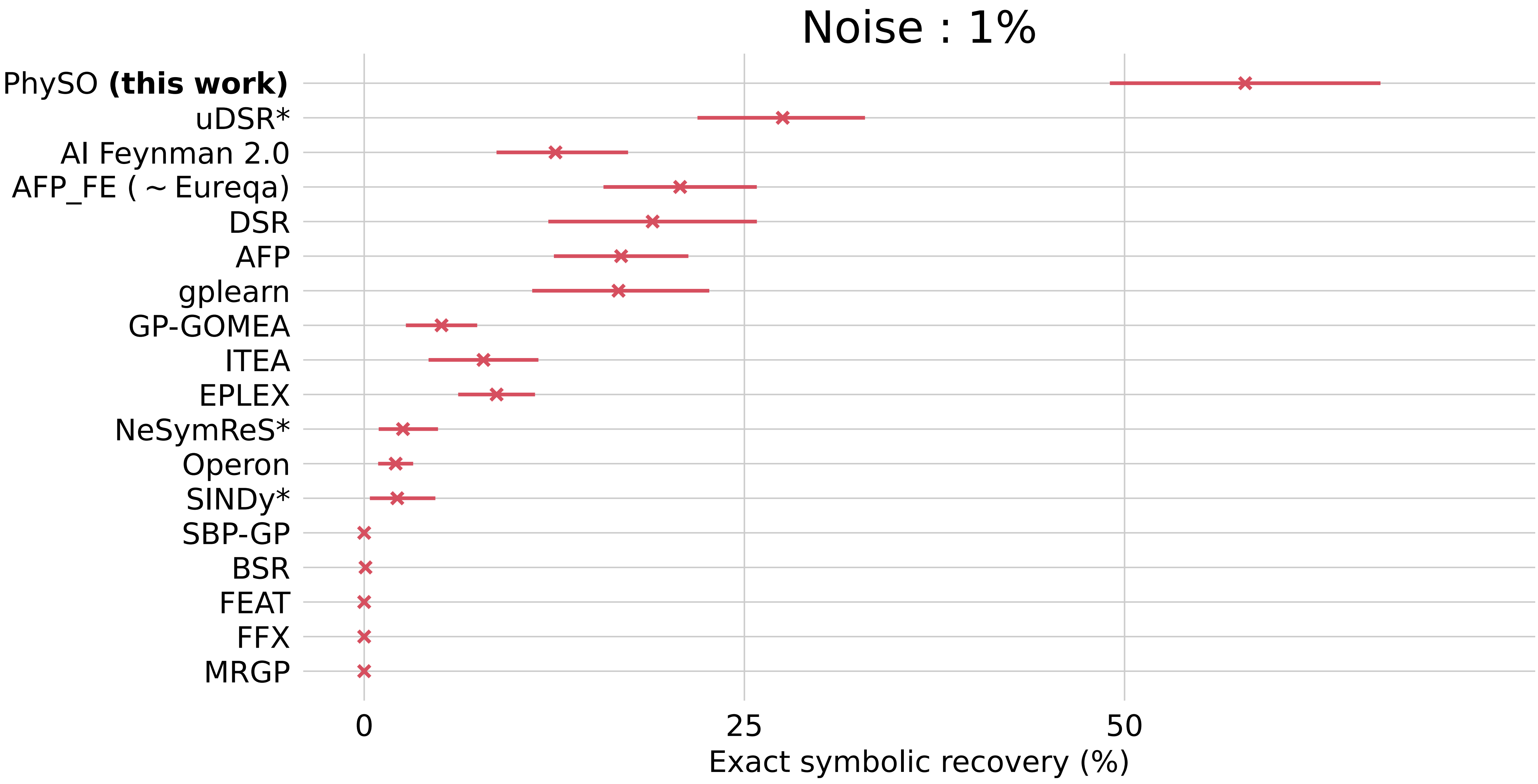
[Tenachi et al 2023]

Feynman Benchmark



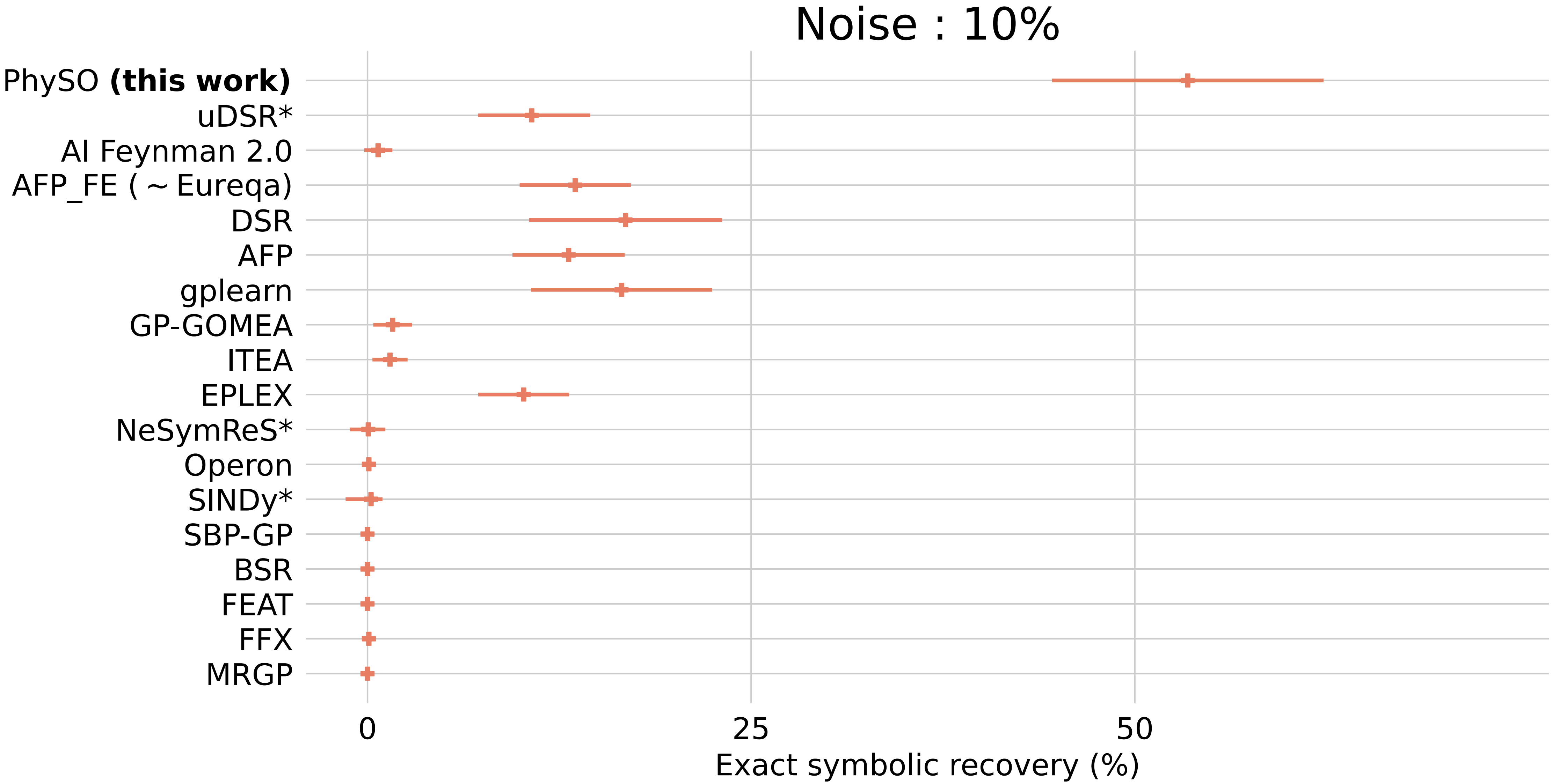
[Tenachi et al 2023]

Feynman Benchmark



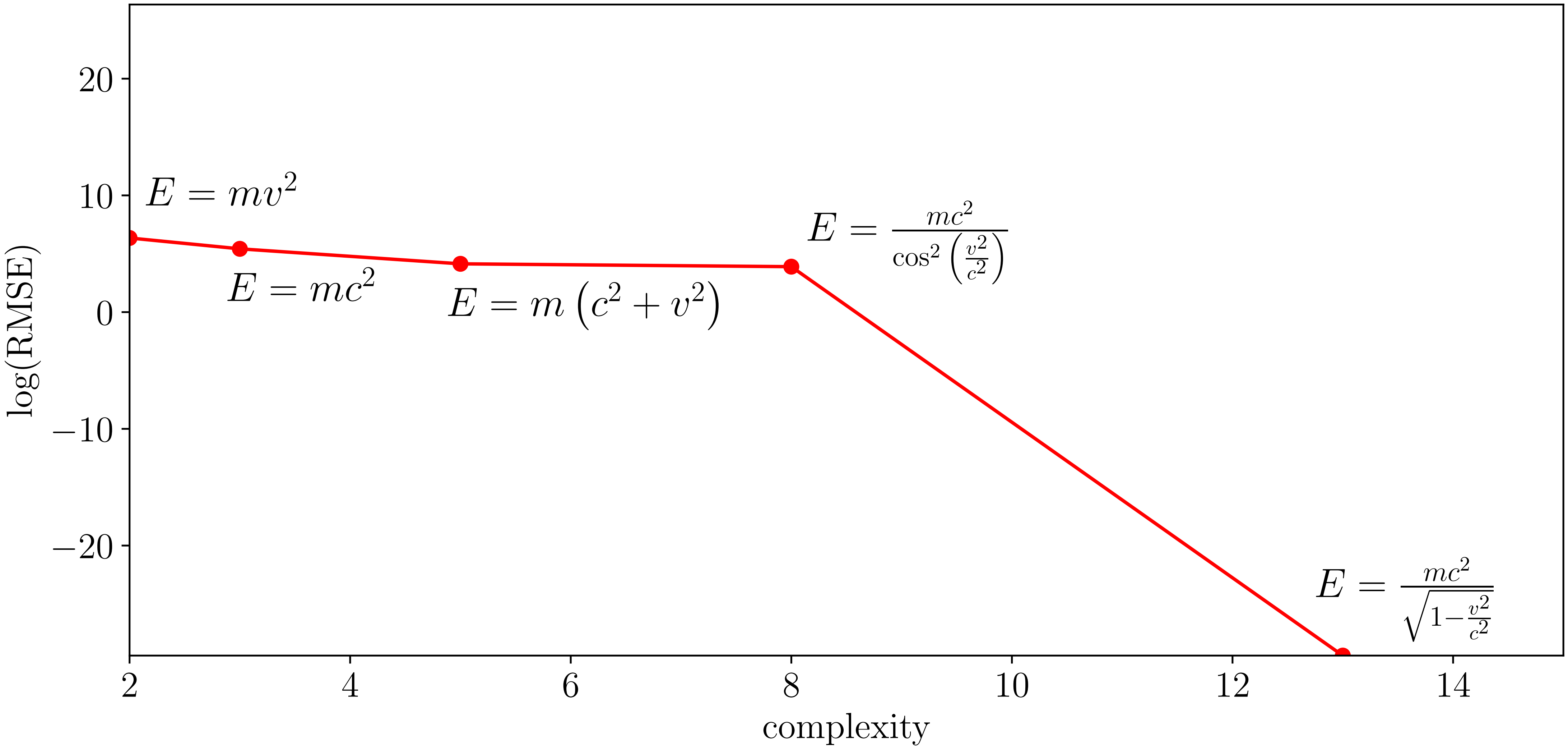
[Tenachi et al 2023]

Feynman Benchmark



[Tenachi et al 2023]

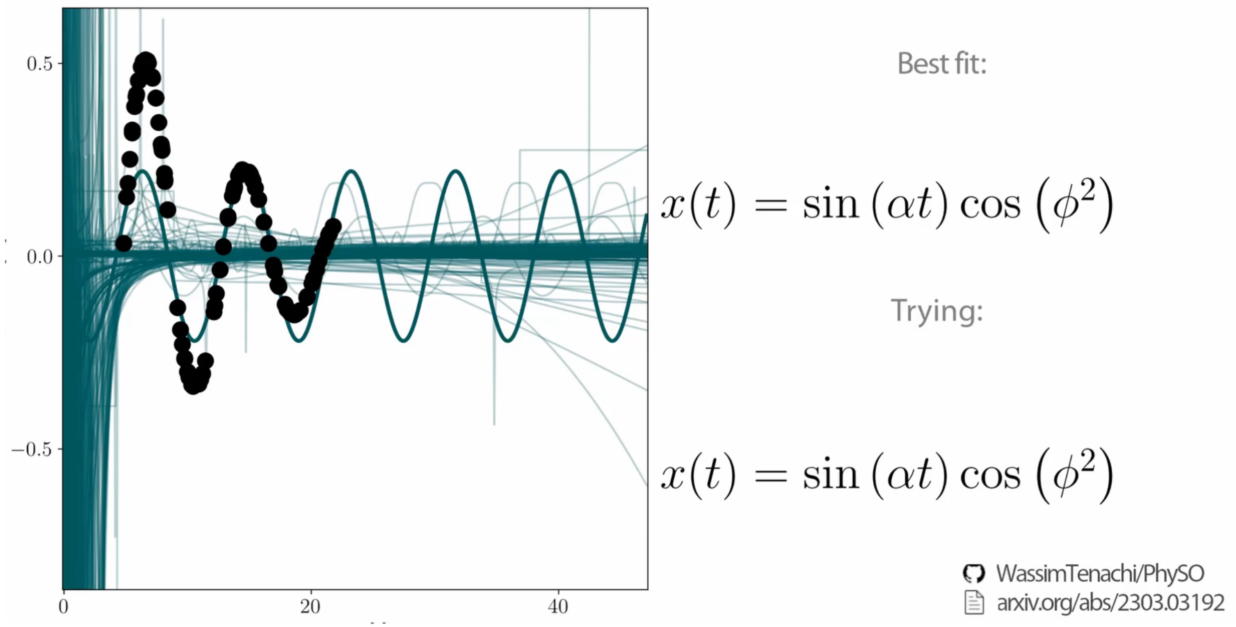
Discovering new physics with SR



Conclusion

Φ-SO is a RL framework that:

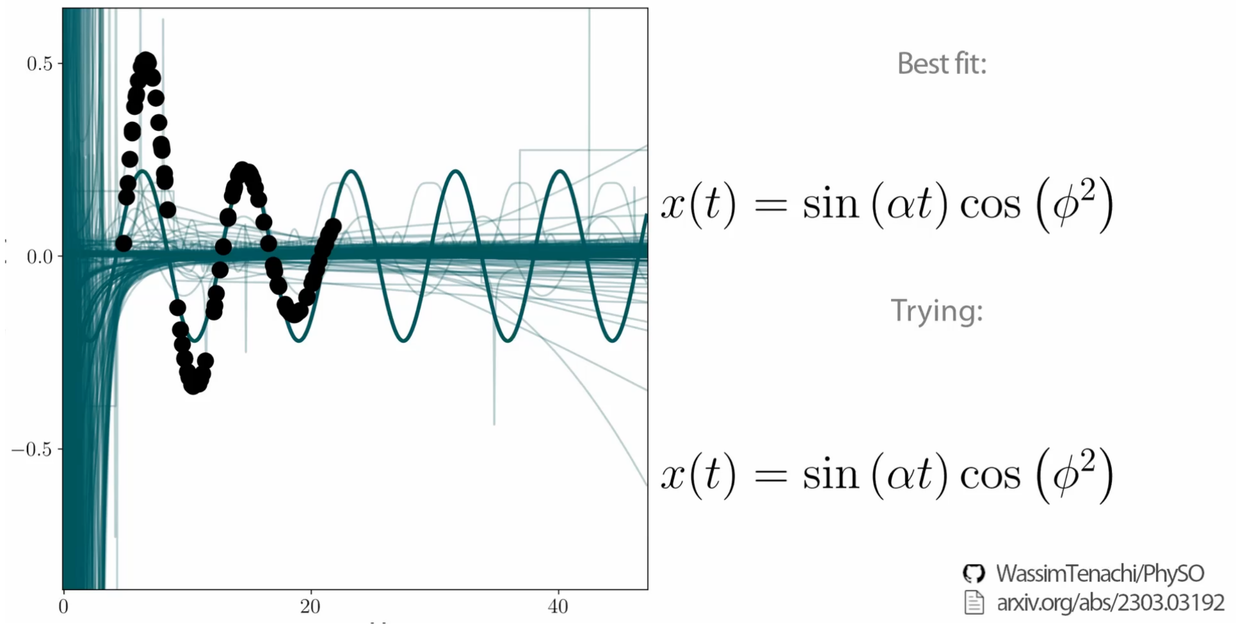
1. Can be used to resolve any symbolic optimization task
2. While taking full advantage of physical units constraints



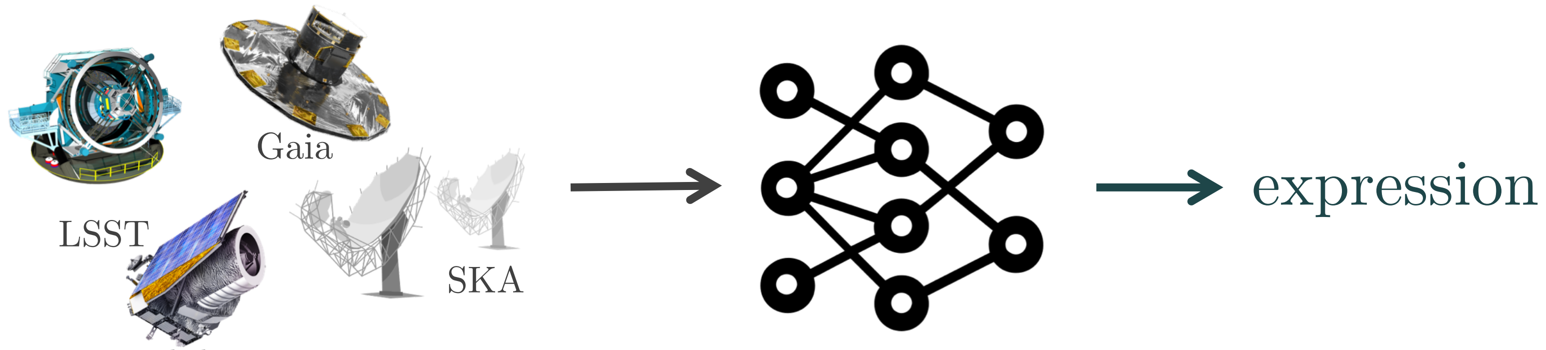
Conclusion

Φ-SO is a RL framework that:

1. Can be used to resolve any symbolic optimization task
2. While taking full advantage of physical units constraints



Big data from surveys & simulations



→ Black box

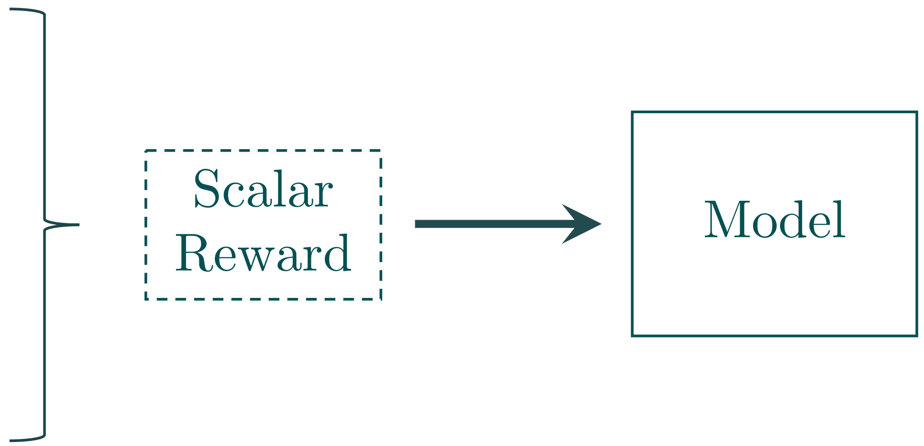
1. Interpretability
2. Compactness
3. Generalization

Perspectives

Curse of accuracy guided SR : "*One can improve fit quality of candidates over learning iterations while getting further away from the correct solution in symbolic arrangement.*"

$(\boldsymbol{X}, y)$ data

x_1	x_2	y
0.75582	0.25850	0.02674
0.36786	0.42401	0.06278
0.69507	0.38057	0.74014
0.96493	0.33398	0.81558
0.07139	0.16604	0.07735
0.86413	0.41952	0.87872
0.18012	0.40581	0.63637



→ Leverage data in a more meaningful way

Perspectives

"Book smart" or "street smart" ?

Trial and error : " *I try to resolve this specific problem by trial and error without benefiting from past experience.*"

Supervised learning: " *I learn a lot beforehand and if what I learned does not help me guess the answer for this specific problem, then too bad there is nothing I can do.*"

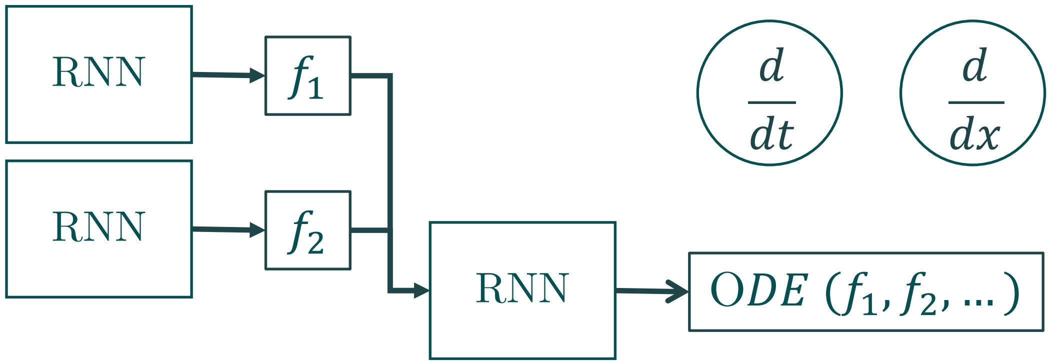
Perspectives

“Book smart” or “street smart” ?

Trial and error : “ *I try to resolve this specific problem by trial and error without benefiting from past experience.*”

Supervised learning: “ *I learn a lot beforehand and if what I learned does not help me guess the answer for this specific problem, then too bad there is nothing I can do.*”

Differential operators & sub-functions



Physical Symbolic Optimization

Φ-SO

Physical Symbolic Optimization.

An open source ...

Github repository: WassimTenachi/PhySO

PhySO Public

Physical Symbolic Optimization

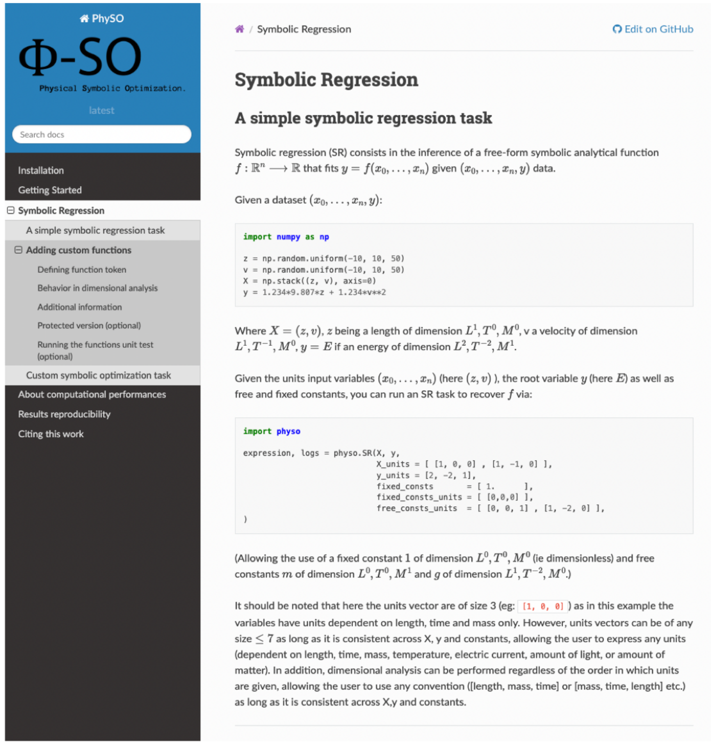
Python ☆ 1.5k 205

Creating a physical free-form symbolic analytical function $f: \mathbb{R}^n \rightarrow \mathbb{R}$ fitting $y = f(\boldsymbol{X})$ given $(\boldsymbol{X}, y)$ data:

```
expression, logs = physo.SR(X, y,
                            X_units = [ [1, 0, 0] , [1, -1, 0] ],
                            y_units = [2, -2, 1],
                            fixed_consts       = [ 1.      ],
                            fixed_consts_units = [ [0,0,0] ],
                            free_consts_units  = [ [0, 0, 1] , [1, -2, 0] ],
)
```

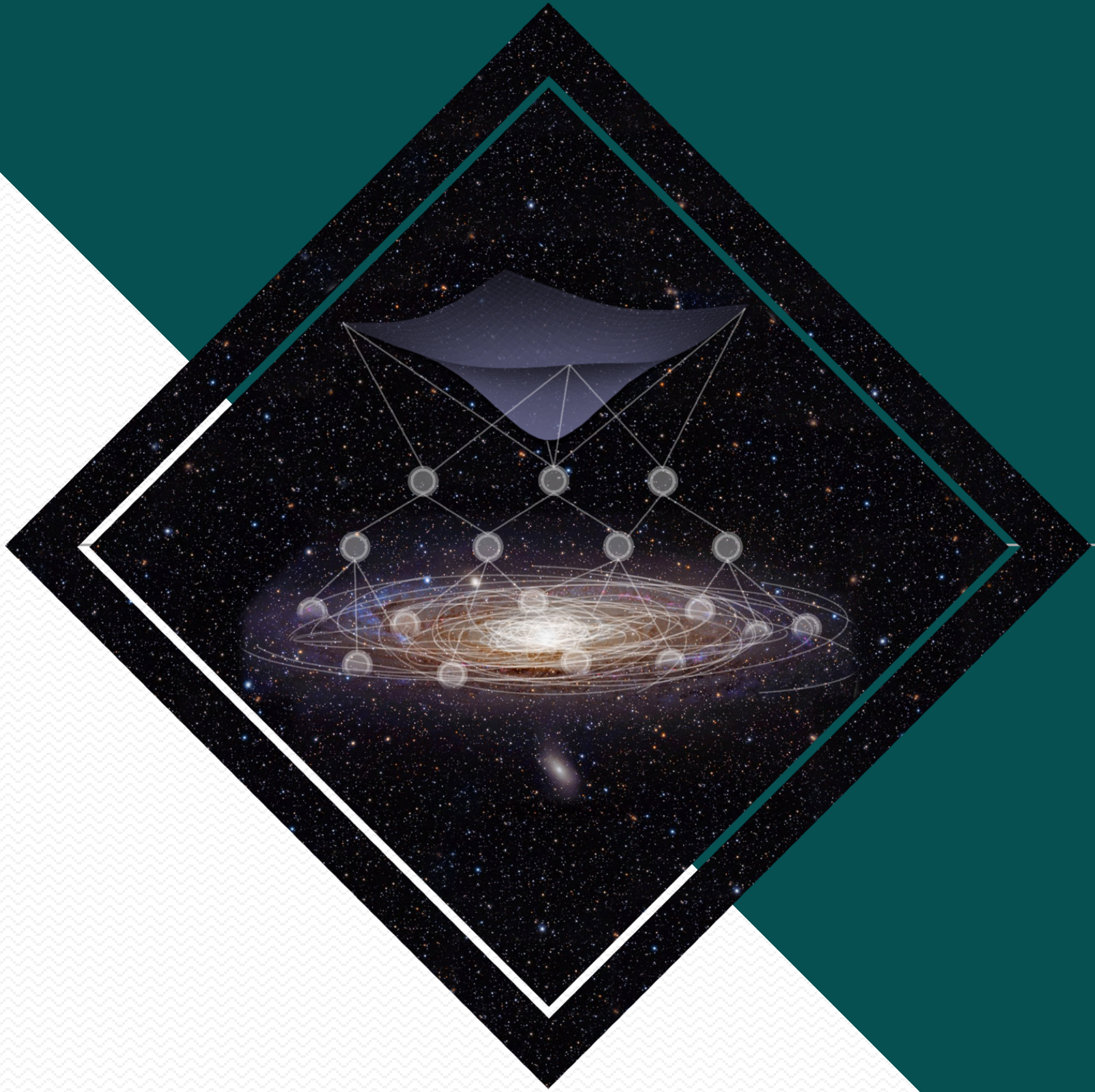
... and documented package

Documentation: physo.readthedocs.io



[Tenachi et al 2023]

Thank you for your attention !



Evaluating a candidate function f

1. Fitting free constants (α)

$$f_\alpha(\boldsymbol{x}) = \hat{y} \quad \text{vs} \quad y \longrightarrow \alpha$$

(using the Broyden–Fletcher–Goldfarb–Shanno (BFGS) optimization algorithm)

Evaluating a candidate function f

1. Fitting free constants (α)

$$f_\alpha(\boldsymbol{x}) = \hat{y} \quad \text{vs} \quad y \longrightarrow \alpha$$

(using the Broyden–Fletcher–Goldfarb–Shanno (BFGS) optimization algorithm)

2. Reward (R)

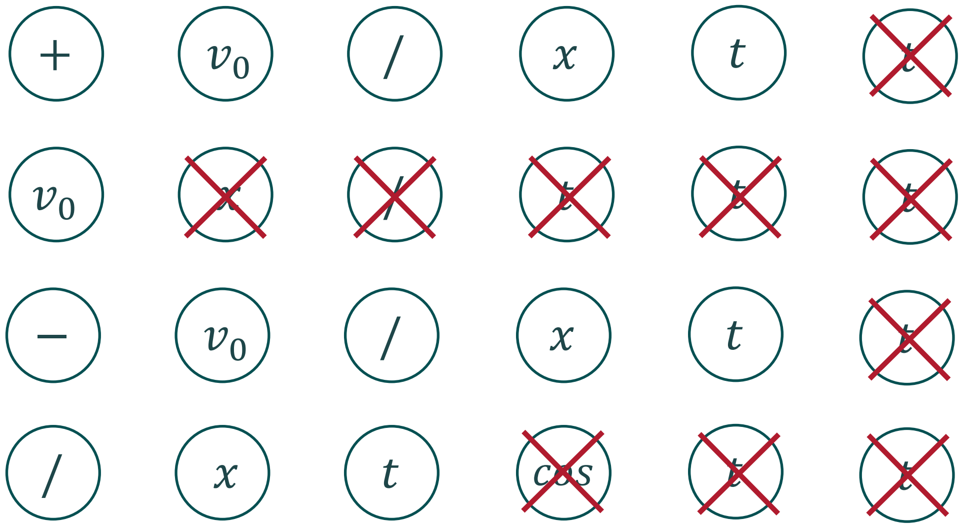
Normalized Root Mean Square Error

$$\text{NRMSE} = \frac{1}{\sigma_y}\sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - f_\alpha(\boldsymbol{x}_i))^2} \longrightarrow R = \frac{1}{1 + NRMSE}$$

$R \in [0,1]$

Loss details (1)

- <u>Expressions of a given batch have different sizes</u>



$$v = v_0 + x/t$$

$$v = v_0$$

$$v = v_0 - x/t$$

$$v = x/t$$

→ Placeholder out of valid expression range are not taken into account

Loss details (2)

- Weighting is different along sequence dimension: via a γ^t coefficient

$+$	v_0	$/$	x	t	~~t~~
γ^0	γ^1	γ^2	γ^3	γ^4	γ^5

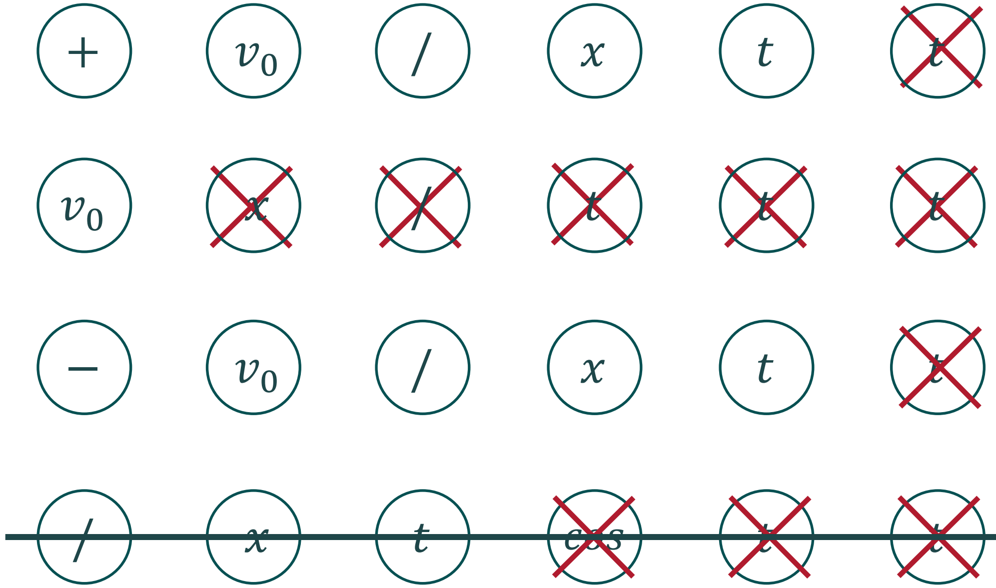
$$v = v_0 + x/t$$

→ We give more importance to first symbols via $\gamma = 0.7 < 1$ to avoid searching around the same initial nodes

[Landajuela et al 2021]

Loss details (3)

- <u>Weighting gradients accordingly with reward value</u>



5% best

$v = v_0 + x/t$ → R = 0.99

$v = v_0$ → R = 0.93

$v = v_0 - x/t$ → R = 0.73

~~$v = x/t$ → R = 0.28~~

(R – baseline)

// $R_{lim} = 0.73$

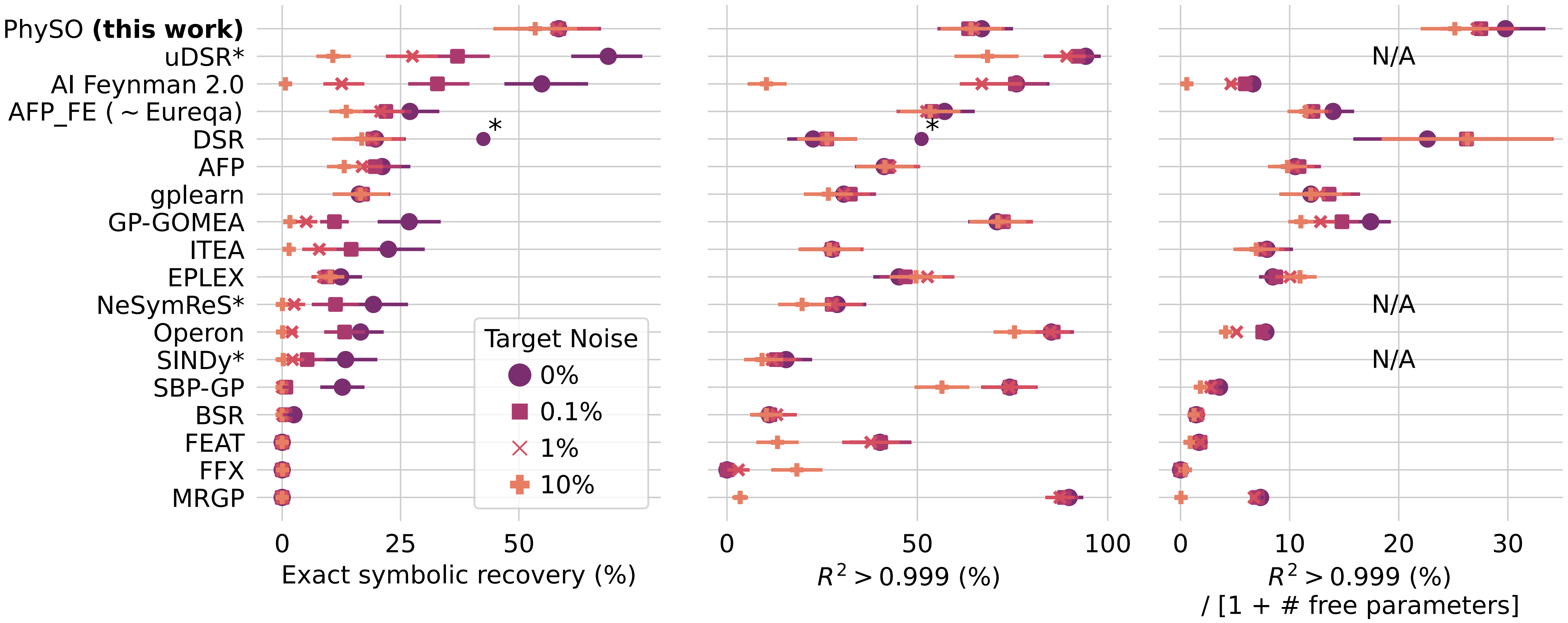
0.26

0.20

0.00

[Petersen et al 2019]

Feynman Benchmark



[Tenachi et al 2023]

(Astro)-physical showcases & ablation study

- ✓ RNN : vanilla model
- ✓ Φ-RNN : model can observe physical units context
- ✓ Φ-prior : physical units prior

Expression	# Trial expressions	Φ-SO ≡ {Φ-prior, Φ-RNN}	{Φ-RNN}	{Φ-prior, RNN}	{RNN}	{Φ-prior, RNG}	{RNG}
$E = \frac{mc^2}{\sqrt{1-v^2/c^2}}$	10M	100 %	0 %	60 %	0 %	20 %	0 %
$J_r = \frac{GM}{\sqrt{-2E}} - \frac{1}{2}\left(L + \frac{1}{2}\sqrt{L^2 - 4GMb}\right)$	4M	100 %	0 %	80 %	0 %	60 %	0 %
$\rho = \rho_0 / \left(\frac{r}{R_s}(1 + \frac{r}{R_s})^2\right)$	2M	100 %	100 %	40 %	100 %	20 %	100 %
$y = e^{-\alpha t}\cos(ft + \Phi)$	1M	100 %	0 %	0 %	0 %	0 %	0 %
$F = \frac{Gm_1 m_2}{r^2}$	100K	100 %	80 %	100 %	20 %	80 %	0 %
$H^2(x \equiv 1 + z) = {H_0}^2(\Omega_m x^3 + (1 - \Omega_m))$	100K	100 %	100 %	100 %	100 %	40 %	40 %

Allowing: $\{+, -, \times, /, 1/\square, \sqrt{\square}, \square^2, \exp, \log, \cos, \sin, 1\}$, free constants & input variables

(Astro)-physical showcases & ablation study

- ✓ RNN : vanilla model
- ✓ Φ-RNN : model can observe physical units context
- ✗ Φ-prior : physical units prior

Expression	# Trial expressions	Φ-SO ≡ {Φ-prior, Φ-RNN}	{Φ-RNN}	{Φ-prior, RNN}	{RNN}	{Φ-prior, RNG}	{RNG}
$E = \frac{mc^2}{\sqrt{1-v^2/c^2}}$	10M	100 %	0 %	60 %	0 %	20 %	0 %
$J_r = \frac{GM}{\sqrt{-2E}} - \frac{1}{2}\left(L + \frac{1}{2}\sqrt{L^2 - 4GMb}\right)$	4M	100 %	0 %	80 %	0 %	60 %	0 %
$\rho = \rho_0/\left(\frac{r}{R_s}(1+\frac{r}{R_s})^2\right)$	2M	100 %	100 %	40 %	100 %	20 %	100 %
$y = e^{-\alpha t}\cos(ft + \Phi)$	1M	100 %	0 %	0 %	0 %	0 %	0 %
$F = \frac{Gm_1m_2}{r^2}$	100K	100 %	80 %	100 %	20 %	80 %	0 %
$H^2(x \equiv 1+z) = {H_0}^2(\Omega_m x^3 + (1-\Omega_m))$	100K	100 %	100 %	100 %	100 %	40 %	40 %

Allowing: $\{+, -, \times, /, 1/\square, \sqrt{\square}, \square^2, \exp, \log, \cos, \sin, 1\}$, free constants & input variables

(Astro)-physical showcases & ablation study

- ✓ RNN : vanilla model
- ✗ Φ-RNN : model can observe physical units context
- ✓ Φ-prior : physical units prior

Expression	# Trial expressions	Φ-SO ≡ {Φ-prior, Φ-RNN}	{Φ-RNN}	{Φ-prior, RNN}	{RNN}	{Φ-prior, RNG}	{RNG}
$E = \frac{mc^2}{\sqrt{1-v^2/c^2}}$	10M	100 %	0 %	60 %	0 %	20 %	0 %
$J_r = \frac{GM}{\sqrt{-2E}} - \frac{1}{2}\left(L + \frac{1}{2}\sqrt{L^2 - 4GMb}\right)$	4M	100 %	0 %	80 %	0 %	60 %	0 %
$\rho = \rho_0 / \left(\frac{r}{R_s}(1+\frac{r}{R_s})^2\right)$	2M	100 %	100 %	40 %	100 %	20 %	100 %
$y = e^{-\alpha t}\cos(ft + \Phi)$	1M	100 %	0 %	0 %	0 %	0 %	0 %
$F = \frac{Gm_1m_2}{r^2}$	100K	100 %	80 %	100 %	20 %	80 %	0 %
$H^2(x \equiv 1+z) = {H_0}^2(\Omega_m x^3 + (1-\Omega_m))$	100K	100 %	100 %	100 %	100 %	40 %	40 %

Allowing: $\{+, -, \times, /, 1/\square, \sqrt{\square}, \square^2, \exp, \log, \cos, \sin, 1\}$, free constants & input variables

(Astro)-physical showcases & ablation study

- ✓ RNN : vanilla model
- ✗ Φ-RNN : model can observe physical units context
- ✗ Φ-prior : physical units prior

Expression	# Trial expressions	Φ-SO ≡ {Φ-prior, Φ-RNN}	{Φ-RNN}	{Φ-prior, RNN}	{RNN}	{Φ-prior, RNG}	{RNG}
$E = \frac{mc^2}{\sqrt{1-v^2/c^2}}$	10M	100 %	0 %	60 %	0 %	20 %	0 %
$J_r = \frac{GM}{\sqrt{-2E}} - \frac{1}{2}\left(L + \frac{1}{2}\sqrt{L^2 - 4GMb}\right)$	4M	100 %	0 %	80 %	0 %	60 %	0 %
$\rho = \rho_0 / \left(\frac{r}{R_s}(1 + \frac{r}{R_s})^2\right)$	2M	100 %	100 %	40 %	100 %	20 %	100 %
$y = e^{-\alpha t} \cos(ft + \Phi)$	1M	100 %	0 %	0 %	0 %	0 %	0 %
$F = \frac{Gm_1 m_2}{r^2}$	100K	100 %	80 %	100 %	20 %	80 %	0 %
$H^2(x \equiv 1 + z) = {H_0}^2(\Omega_m x^3 + (1 - \Omega_m))$	100K	100 %	100 %	100 %	100 %	40 %	40 %

Allowing: $\{+, -, \times, /, 1/\square, \sqrt{\square}, \square^2, \exp, \log, \cos, \sin, 1\}$, free constants & input variables

(Astro)-physical showcases & ablation study

- ✗ RNN : vanilla model
- ✗ Φ-RNN : model can observe physical units context
- ✓ Φ-prior : physical units prior

Expression	# Trial expressions	Φ-SO ≡ {Φ-prior, Φ-RNN}	{Φ-RNN}	{Φ-prior, RNN}	{RNN}	{Φ-prior, RNG}	{RNG}
$E = \frac{mc^2}{\sqrt{1-v^2/c^2}}$	10M	100 %	0 %	60 %	0 %	20 %	0 %
$J_r = \frac{GM}{\sqrt{-2E}} - \frac{1}{2}\left(L + \frac{1}{2}\sqrt{L^2 - 4GMb}\right)$	4M	100 %	0 %	80 %	0 %	60 %	0 %
$\rho = \rho_0 / \left(\frac{r}{R_s}(1+\frac{r}{R_s})^2\right)$	2M	100 %	100 %	40 %	100 %	20 %	100 %
$y = e^{-\alpha t}\cos(ft + \Phi)$	1M	100 %	0 %	0 %	0 %	0 %	0 %
$F = \frac{Gm_1 m_2}{r^2}$	100K	100 %	80 %	100 %	20 %	80 %	0 %
$H^2(x \equiv 1+z) = {H_0}^2(\Omega_m x^3 + (1-\Omega_m))$	100K	100 %	100 %	100 %	100 %	40 %	40 %

Allowing: $\{+, -, \times, /, 1/\square, \sqrt{\square}, \square^2, \exp, \log, \cos, \sin, 1\}$, free constants & input variables

(Astro)-physical showcases & ablation study

- ✗ RNN : vanilla model
- ✗ Φ-RNN : model can observe physical units context
- ✗ Φ-prior : physical units prior

Expression	# Trial expressions	Φ-SO ≡ {Φ-prior, Φ-RNN}	{Φ-RNN}	{Φ-prior, RNN}	{RNN}	{Φ-prior, RNG}	{RNG}
$E = \frac{mc^2}{\sqrt{1-v^2/c^2}}$	10M	100 %	0 %	60 %	0 %	20 %	0 %
$J_r = \frac{GM}{\sqrt{-2E}} - \frac{1}{2}\left(L + \frac{1}{2}\sqrt{L^2 - 4GMb}\right)$	4M	100 %	0 %	80 %	0 %	60 %	0 %
$\rho = \rho_0 / \left(\frac{r}{R_s}(1+\frac{r}{R_s})^2\right)$	2M	100 %	100 %	40 %	100 %	20 %	100 %
$y = e^{-\alpha t}\cos(ft + \Phi)$	1M	100 %	0 %	0 %	0 %	0 %	0 %
$F = \frac{Gm_1m_2}{r^2}$	100K	100 %	80 %	100 %	20 %	80 %	0 %
$H^2(x \equiv 1+z) = {H_0}^2(\Omega_m x^3 + (1-\Omega_m))$	100K	100 %	100 %	100 %	100 %	40 %	40 %

Allowing: $\{+, -, \times, /, 1/\square, \sqrt{\square}, \square^2, \exp, \log, \cos, \sin, 1\}$, free constants & input variables