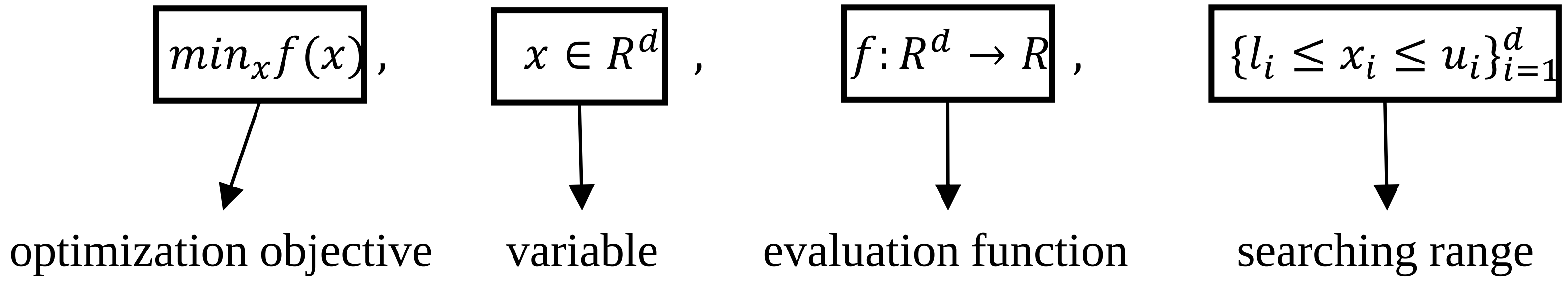
SYMBOL: Generating Flexible Black-Box Optimizers through Symbolic Equation Learning

Jiacheng Chen[1, *], Zeyuan Ma[1, *], Hongshu Guo[1], Yining Ma[2], Jie Zhang[2], Yue-Jiao Gong[1, †]

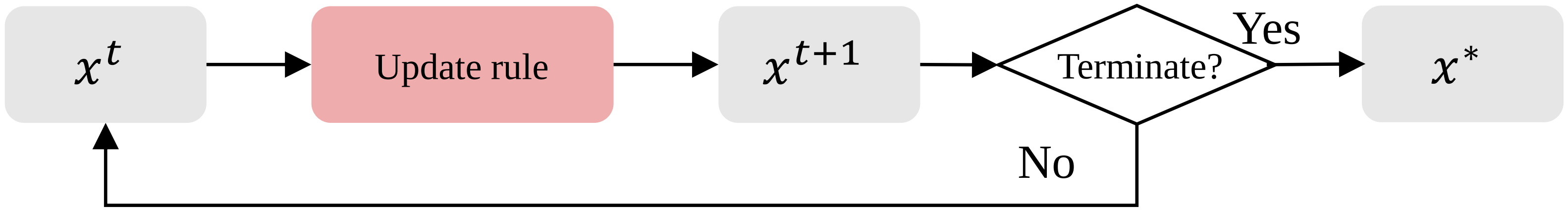
[1]South China University of Technology

[2]Nanyang Technological University

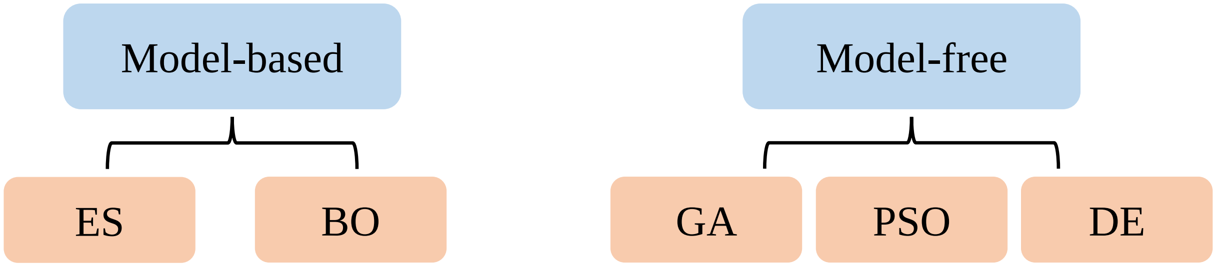
Black-Box Optimization (BBO):



Traditional BBO optimizer workflow:



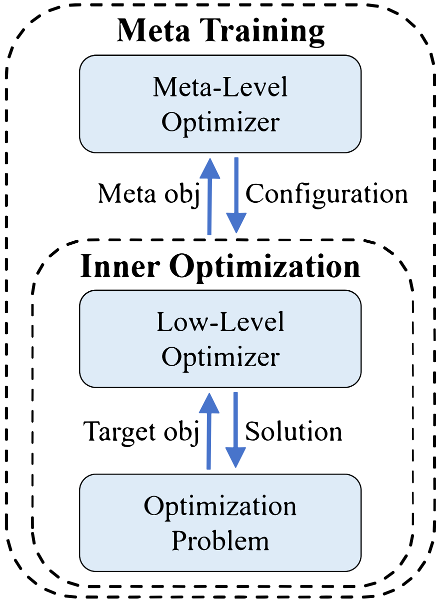
Well-known BBO optimizers:



Meta-Black-Box Optimization (BBO):

Traditional optimizers require heavy human-crafting!!!

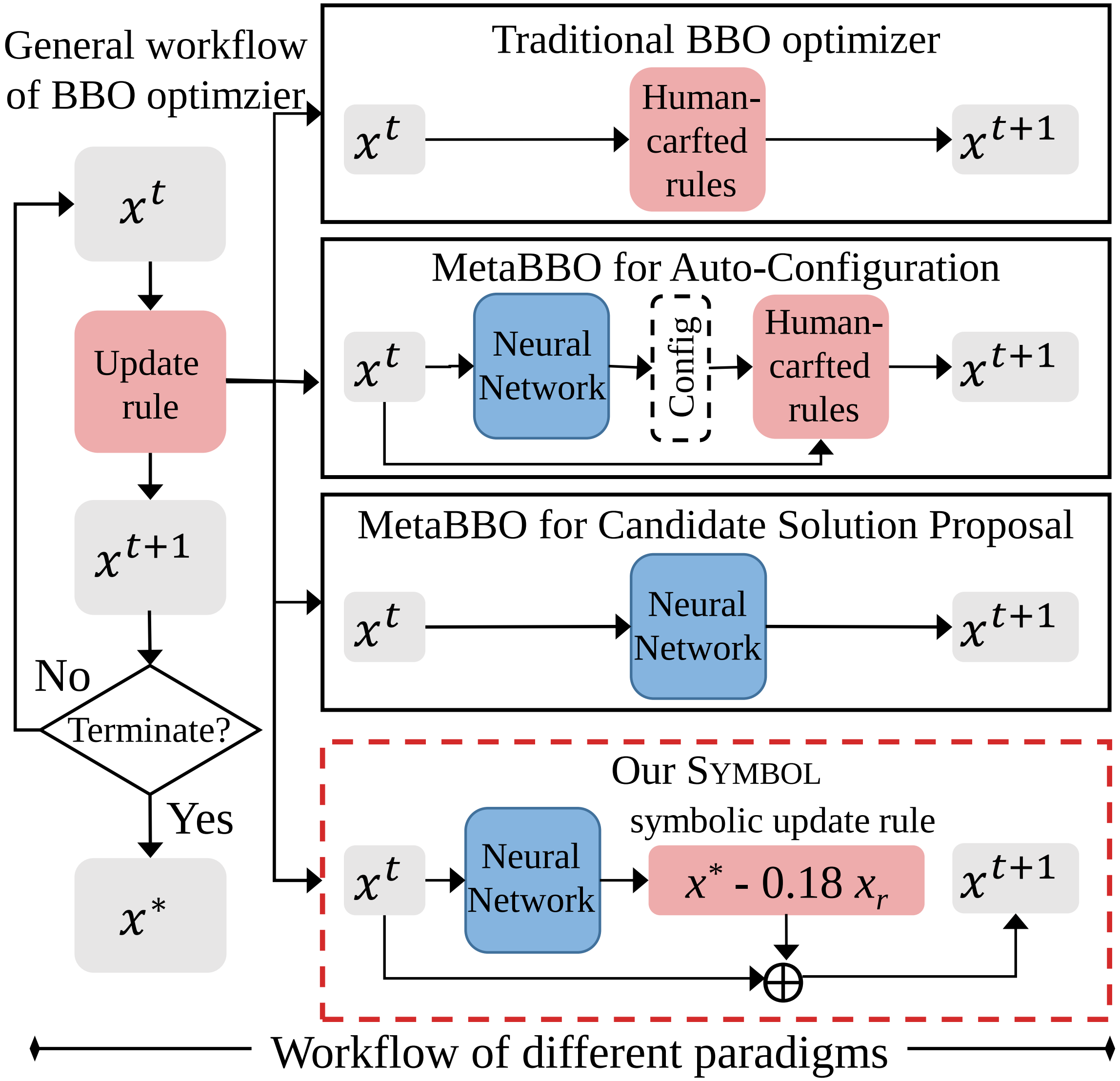
MetaBBO methods introduce a novel meta-level as an automatic decision process. The purpose is to alleviate the need for labor-intensive manual designing/fine-tuning of low-level BBO optimizers.



Mathematically:

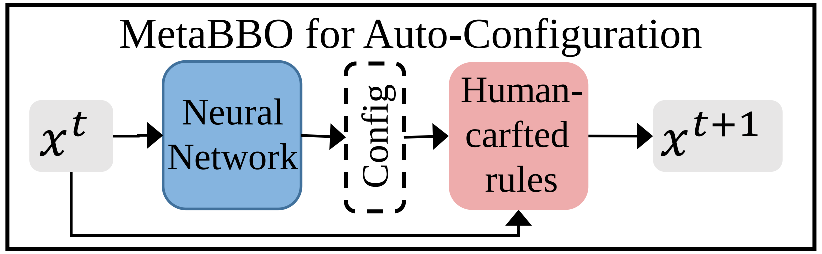
$$max_\theta E_{f\sim D,\pi_\theta}[\sum_{t=0}^{T} r_t]$$

D: optimization task distribution
π_θ: meta level control policy
r_t: performance gain at lower level

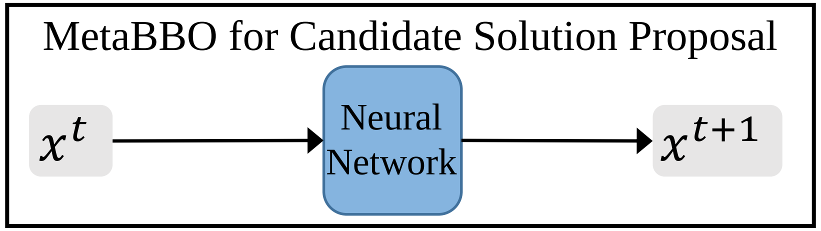


Since existing works either show dependence on hand-crafted optimizers or poor intepretability, we propose Symbol to address these issues.

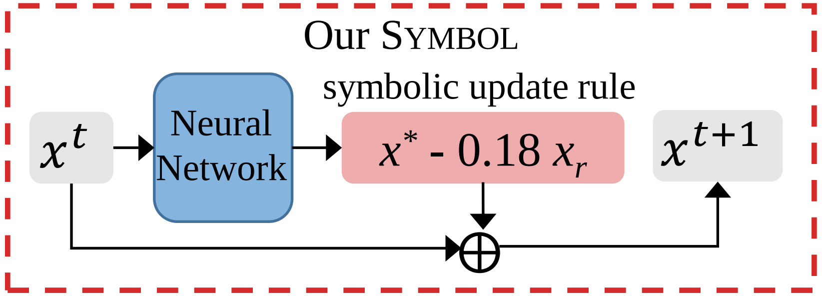
Recall that the workflow of MetaBBO for Auto-Configuration:



and the workflow of MetaBBO for Candidate Solution Proposal:



Our Symbol estabilshes a generating process as:



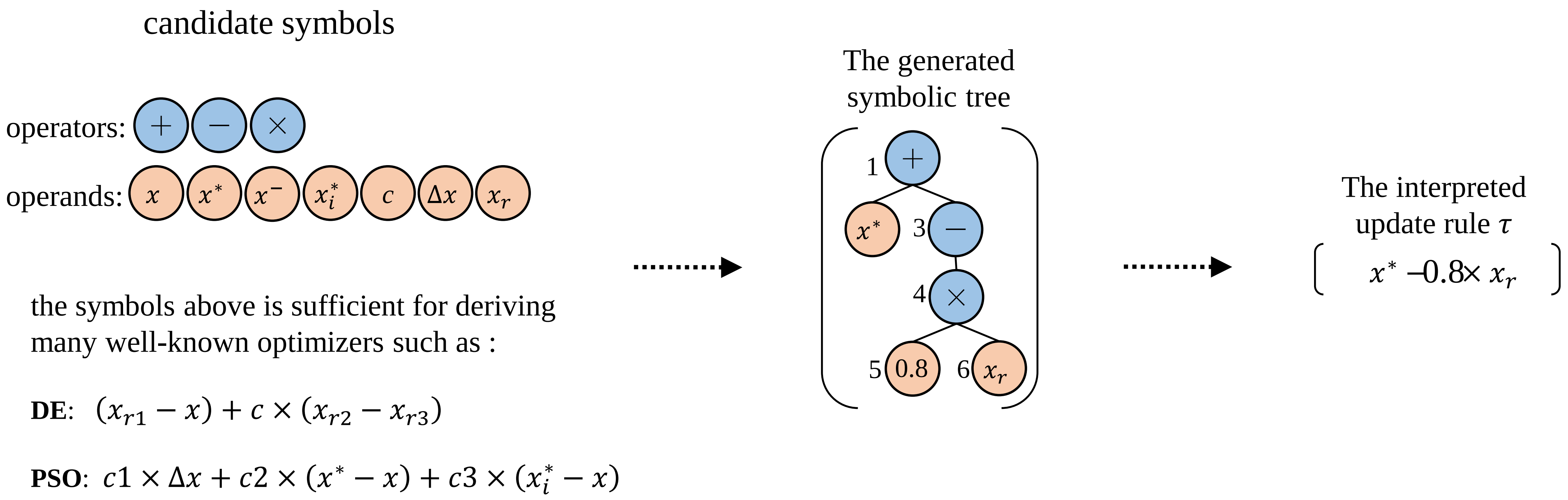
dynamically generate optimization strategy

<u>more flexible</u>

<u>elegant interpretability</u>

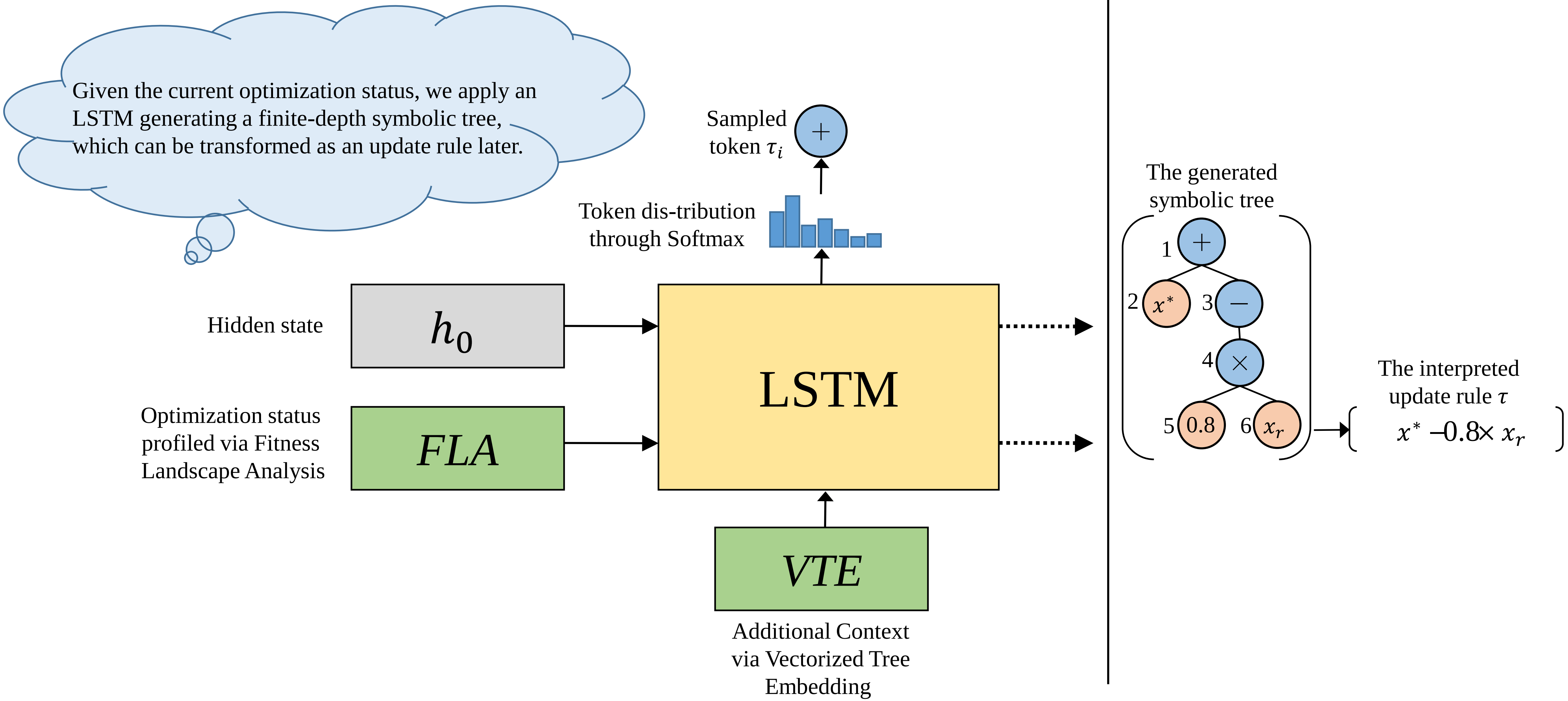
<u>identify undiscovered rules</u>

High-level workflow of Symbol



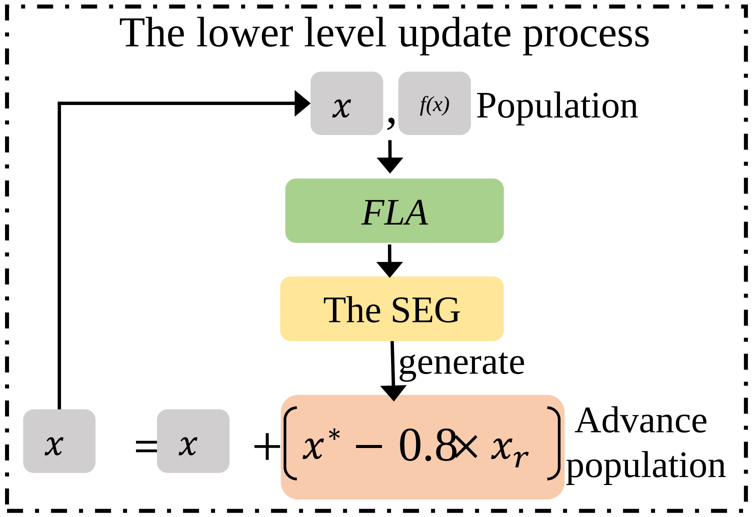
Any off-the-shelf time-series neural network can be used, while we adopt LSTM in our paper.

To construct a symbolic binary tree corresponding to an optimization update rule:

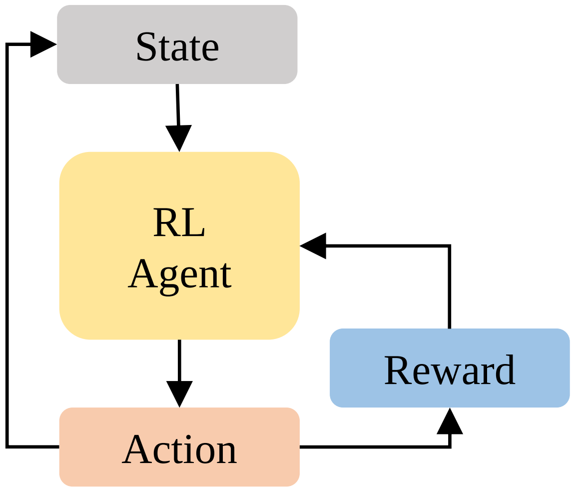


Seems like a perfect generation process, yet how do we learn the LSTM?

recall the low-level workflow:



and analog it to a Markov Decision Process:



If we could define what is the **Reward**, the LSTM would be properly trained by RL:

Symbol-E: Learn to find optimal solution (**exploration**)

$$R_{\text{explore}}\left(\tau^{(t)}, f\right) = (-1) \cdot \frac{y^{*,(t)} - y^{\text{opt}}}{y^{*,(0)} - y^{\text{opt}}}$$

Symbol-G: Learn from an adavanced teacher optimizer (**exploitation**)

$$R_{\text{guided}}\left(\tau^{(t)}, f\right) = \frac{-1}{x_{max} - x_{min}} \cdot \max\left\{\min\left\{|x_i^{(t)} - x_{\kappa,j}^{(t)}|_2\right\}_{j=1}^{|x_\kappa|}\right\}_{i=1}^{|x|}$$

Symbol-S: Synergize the above two (**exploration-exploitation trade-off**)

$$R_{\text{synergized}}(\cdot) = R_{\text{explore}}(\cdot) + \lambda R_{\text{guided}}(\cdot)$$

We train Symbol using the proposed three reward functions, teacher optimizer is MadDE.
Training algorithm is PPO, a dominating policy gradient method.
Training tasks come from well-known BBO synthetic benchmarks.
During the experimental analysis, we have the following observations:

Superior in-distribution and out-of-distribution generalization ability

	Baselines	Meta Test ($\mathbb{D}$) #Ps=100, #Dim=10, #FEs=50000 Mean↑±(Std)	Time	Rank	Meta Generalization (HPO-B) #Ps=5, #Dim=2∼16, #FEs=500 Mean↑±(Std)	Time	Rank	Meta Generalization (Protein-docking) #Ps=10, #Dim=12, #FEs=1000 Mean↑±(Std)	Time	Rank
BBO	RS	0.932±(0.007)	-/0.03s	11	0.908±(0.004)	-/0.02s	8	0.996±(0.000)	-/0.003s	4
	MadDE	0.940±(0.009)	-/0.8s	6	0.932±(0.004)	-/0.2s	6	0.991±(0.001)	-/0.4s	8
	sep-CMA-ES	0.935±(0.017)	-/1.3s	9	0.870±(0.017)	-/0.1s	9	0.971±(0.000)	-/0.3s	10
	ipop-CMA-ES	0.970±(0.012)	-/1.4s	3	0.938±(0.013)	-/0.1s	5	0.996±(0.003)	-/0.3s	5
	SMAC	0.937±(0.019)	-/1.1m	7	**0.979**±(0.005)	-/0.7m	1	0.998 ±(0.000)	-/3.8m	2
MetaBBO	LDE	0.970±(0.006)	9h/0.9s	2	fail			fail		
	DEDDQN	0.959±(0.007)	38m/1.1m	5	0.862±(0.026)	-/0.6s	10	0.993±(0.000)	-/1.3s	7
	Meta-ES	0.936±(0.012)	12h/0.4s	8	0.949±(0.002)	-/0.1s	3	0.984±(0.000)	-/0.3s	9
	MELBA	0.846±(0.012)	4h/2.6m	13	fail			fail		
	RNN-Opt	0.923±(0.010)	11h/0.4m	12	fail			fail		
	SYMBOL-E	0.934±(0.008)	6h/0.9s	10	0.920±(0.007)	-/0.5s	7	0.996±(0.000)	-/0.5s	3
	SYMBOL-G	0.964±(0.012)	10h/1.0s	4	0.940±(0.011)	-/0.5s	4	0.995±(0.000)	-/0.5s	6
	SYMBOL-S	**0.972**±(0.011)	10h/1.3s	1	0.963±(0.006)	-/0.7s	2	**0.999**±(0.000)	-/0.7s	1

Note: For meta generalization, we test on HPO-B and protein-docking tasks featuring different task dimensions (#Dim), population sizes (#Ps), and optimization horizons (#FEs). Note that several MetaBBO methods fail to generalize to these two realistic tasks: RNN-Opt and MELBA are not generalizable across different task dimensions; LDE is not generalizable across different population sizes.

Flexible exploration-exploitation trade-off with certain intepretability

Table 2: Generated update rules and corresponding frequencies.

Update rule ($x + \underline{\tau}$)	Frequency
$x + \underline{0.18 \times (x^* - x_r) + 0.42 \times (x_i^* - x_r)}$	42.207%
$x + \underline{0.18 \times (x^* - x) + 0.42 \times (x_i^* - x)}$	39.448%
$x + \underline{0.18 \times (x^* - x_i^*)}$	7.448%
$x + \underline{0.6 \times (x^* - x_r) + 0.6 \times (x_i^* - x) + 0.18 \times (x^* - x_i^*)}$	2.601%
$x + \underline{0.78 \times (x^* - x_r) + 0.42 \times (x_i^* - x_r)}$	1.586%

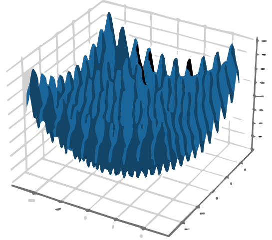


Table 3: 2D Rastrigin.

Our Symbol intelligently apply:

a) $\underline{0.18 \times (x^* - x_r) + 0.42 \times (x_i^* - x_r)}$ at $20 \leq t \leq 25$
to introduce random exploration, hence avoid the pre-mature.

b) $\underline{0.18 \times (x^* - x) + 0.42 \times (x_i^* - x)}$ at $25 \leq t \leq 30$
to prompt the population converging to the real optimal area, hence accelerate the optimization process.

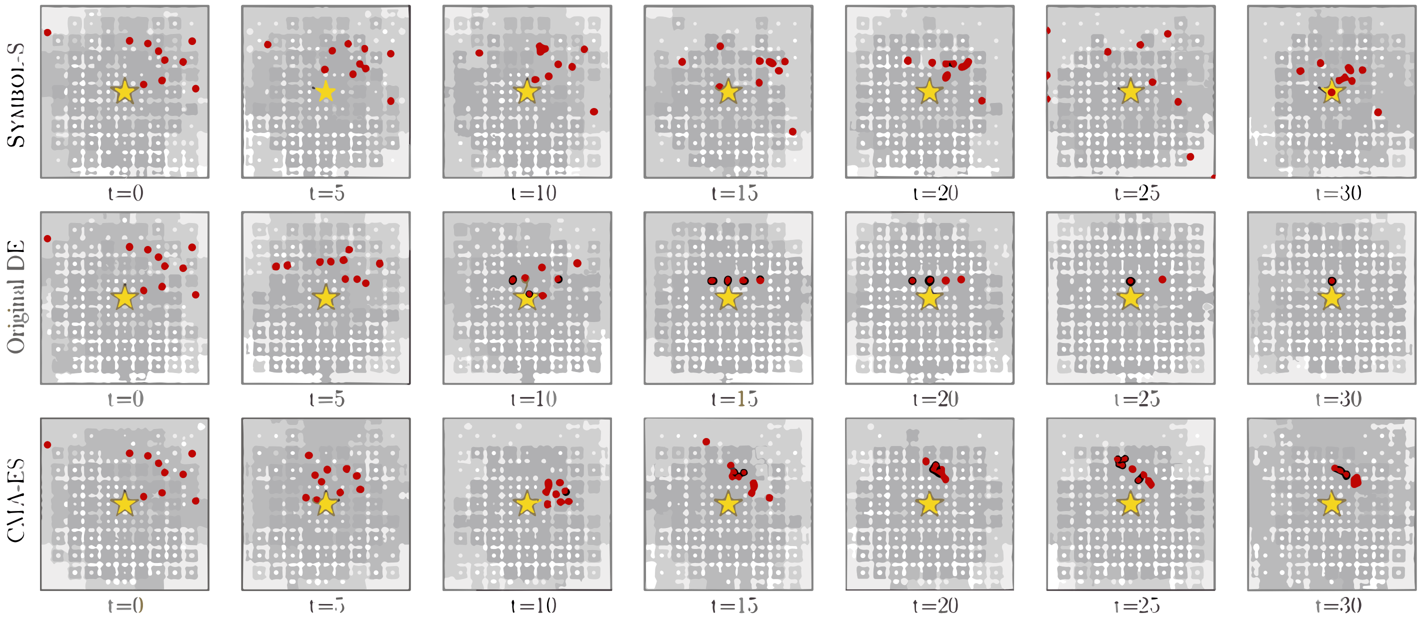


Figure 4: Evolution visulization of the optimizers, showing the position of population (red dots) and the global optimal (yellow star). **Top**: SYMBOL-S; **Middle**: original DE; **Bottom**: CMA-ES.

We would like to list several promising future works of our Symbol:

I. auto-extraction of optimization status features
II. futher extension of basis symbol set with careful design
III. the use of Large Language Models (LLMs)
IV. Can we learn from multiple teachers?

project page

We also welcome any questions about our work, feel free for asking!!!

our team page