MetaBox: A Benchmark Platform for Meta-Black-Box Optimization with Reinforcement Learning

Zeyuan Ma[1] , Hongshu Guo[1] , Jiacheng Chen[1] , Zhenrui Li[1] , Guojun Peng[1],

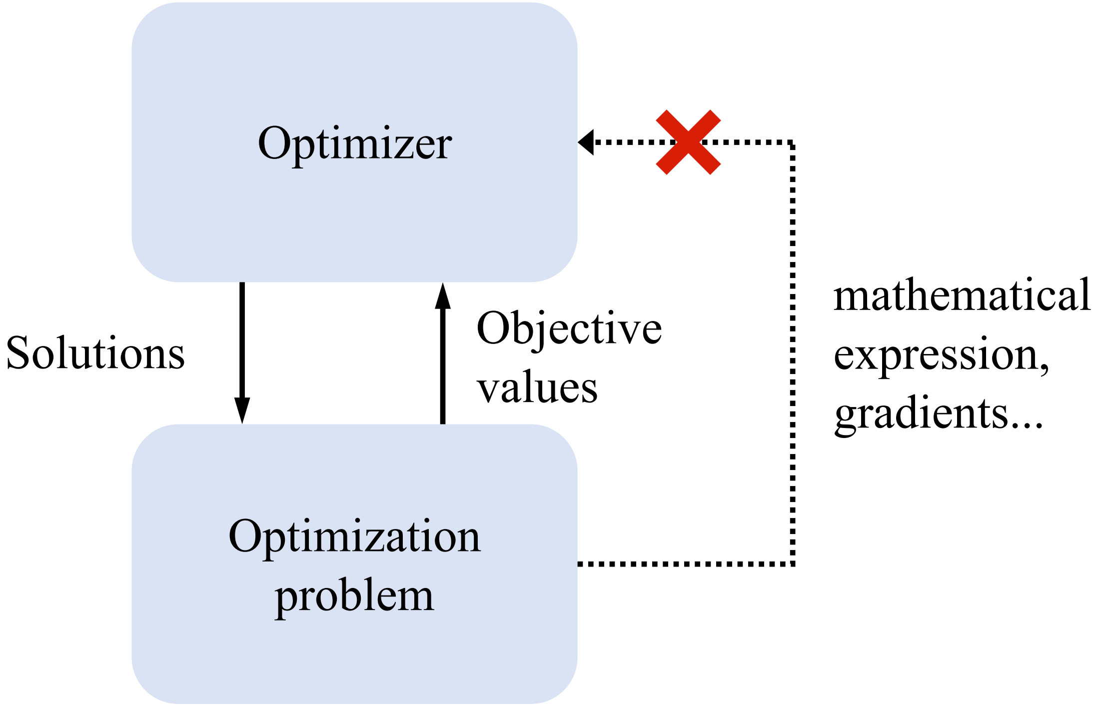
Yue-Jiao Gong[1, *], Yining Ma[2], Zhiguang Cao[3]

[1]South China University of Technology

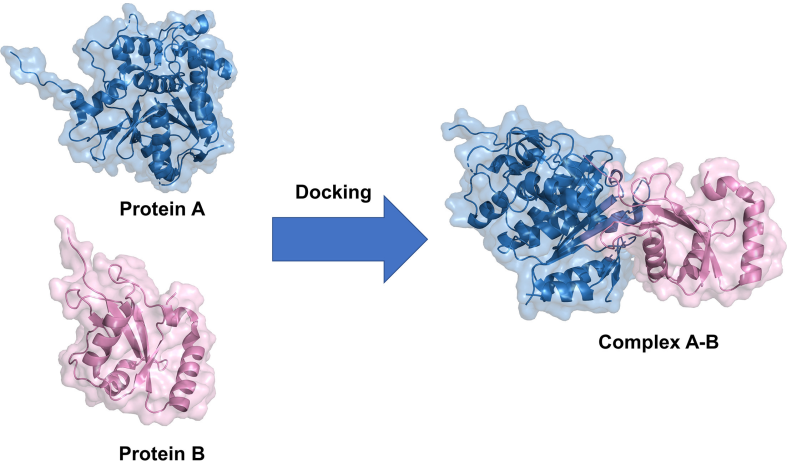
[2]National University of Singapore

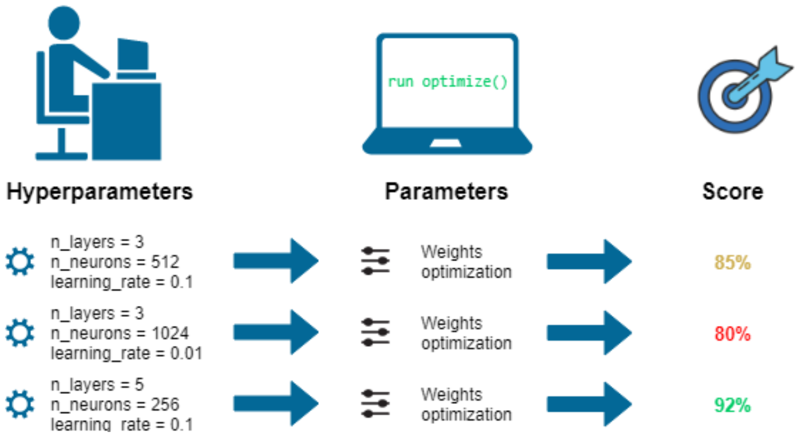
[3]Singapore Management University

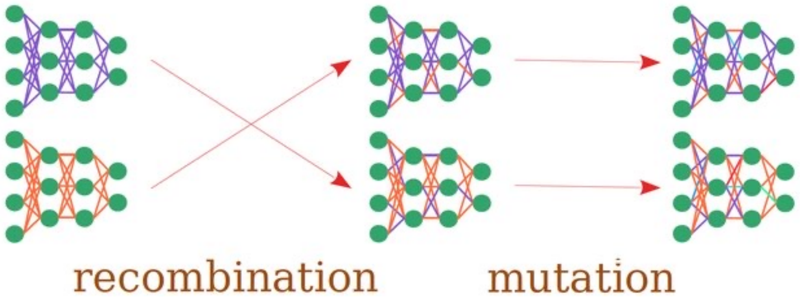
Black Box Optimization (BBO)



Applications of BBO





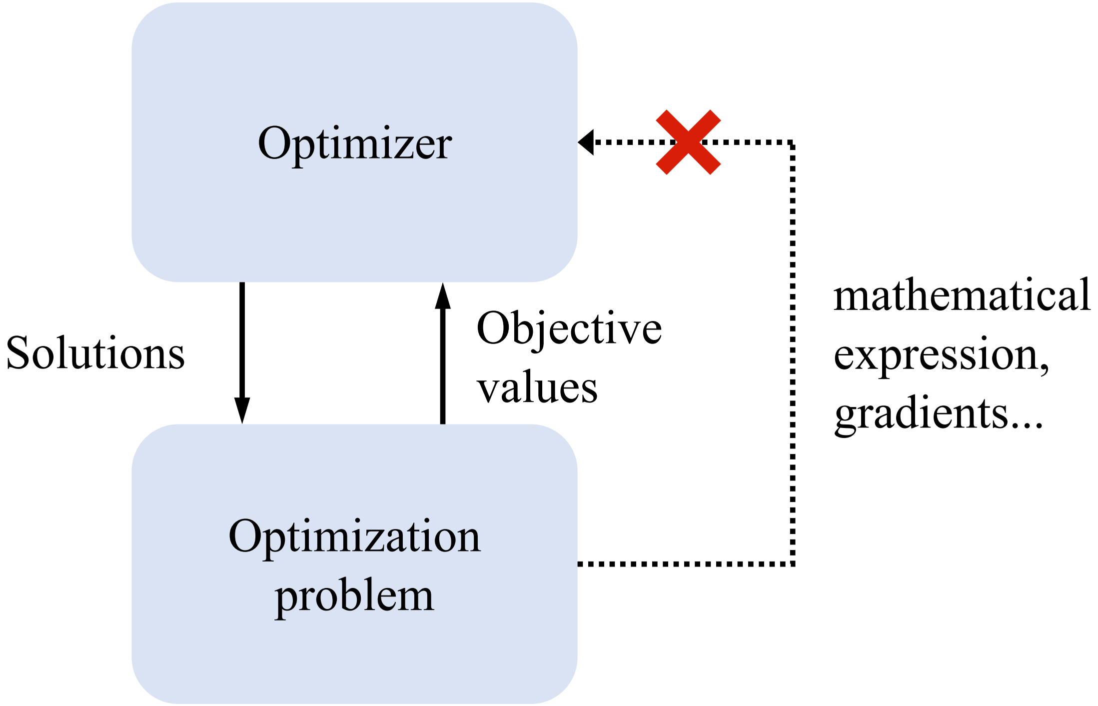


Protein Docking

Hyper Parameter Optimization

Neural Evolution

Black Box Optimization (BBO)



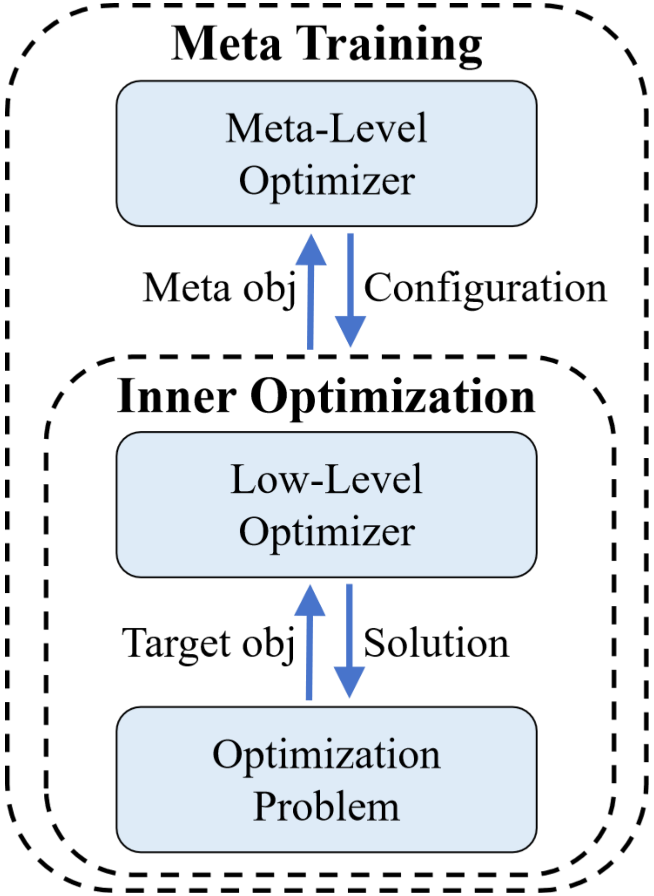
Traditional optimizer:

- Genetic algorithm
- evolutionary strategies
- particle swarm optimization
- differential evolution

Drawbacks:
lean on carefully hand-crafted designs to strike a balance between exploration and exploitation

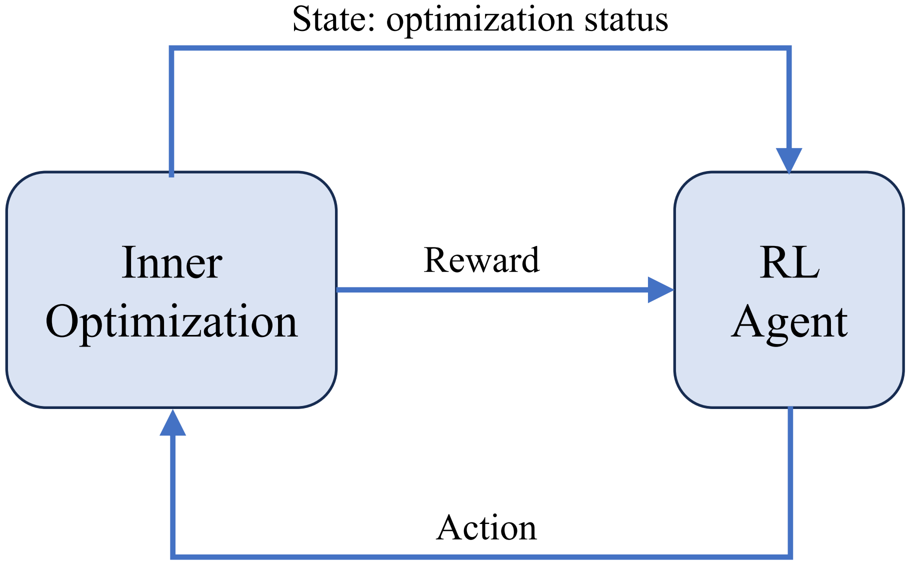
MetaBBO Formulation

Bi-level optimization framework



MetaBBO-RL

MetaBBO-RL model this bi-level optimization procedure as a Markov Decision Process (MDP):

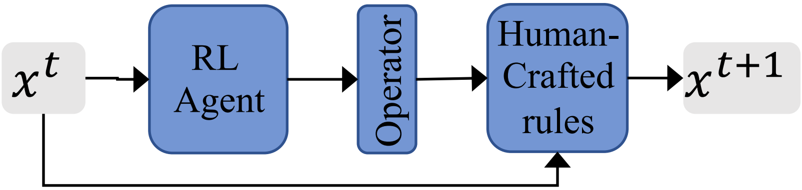


The objective of MetaBBO-RL is to learn an policy(RL agent) that maximizes the expectation of optimization performance on a task distribution:

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

Current attempts of MetaBBO-RL

Approach1: selecting operators

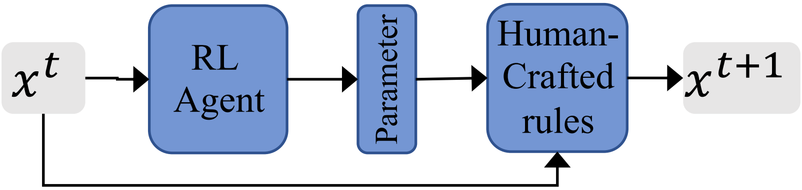


e.g. Adaptively selecting mutation operators in DE

$$x^{t+1} = x^t + F * rand1(x^t)$$

rand1 is selected from [rand1, rand2, cur2rand, cur2gbest…]

Approach2: selecting hyper parameters



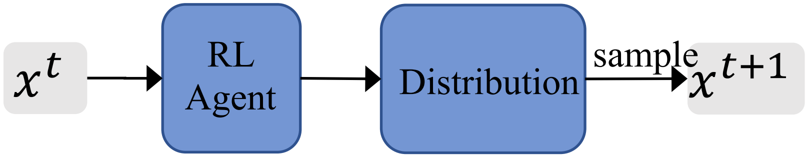
e.g. Adaptively tuning parameters in PSO update rule

$$x^{t+1} = \omega v^t + c_1 r_1 (pbest^t - x^t) + c_2 r_2 (gbest^t - x^t)$$

Approach3: Approach1 + Approach2

e.g. $x^{t+1} = x^t + F * rand1(x^t)$

Approach4: directly determine the sample distribution



e.g.

$$\mu, \sigma = \theta(x^t)$$
$$x^{t+1} = N(\mu, \sigma)$$

Limitations of Existing Benchmarks

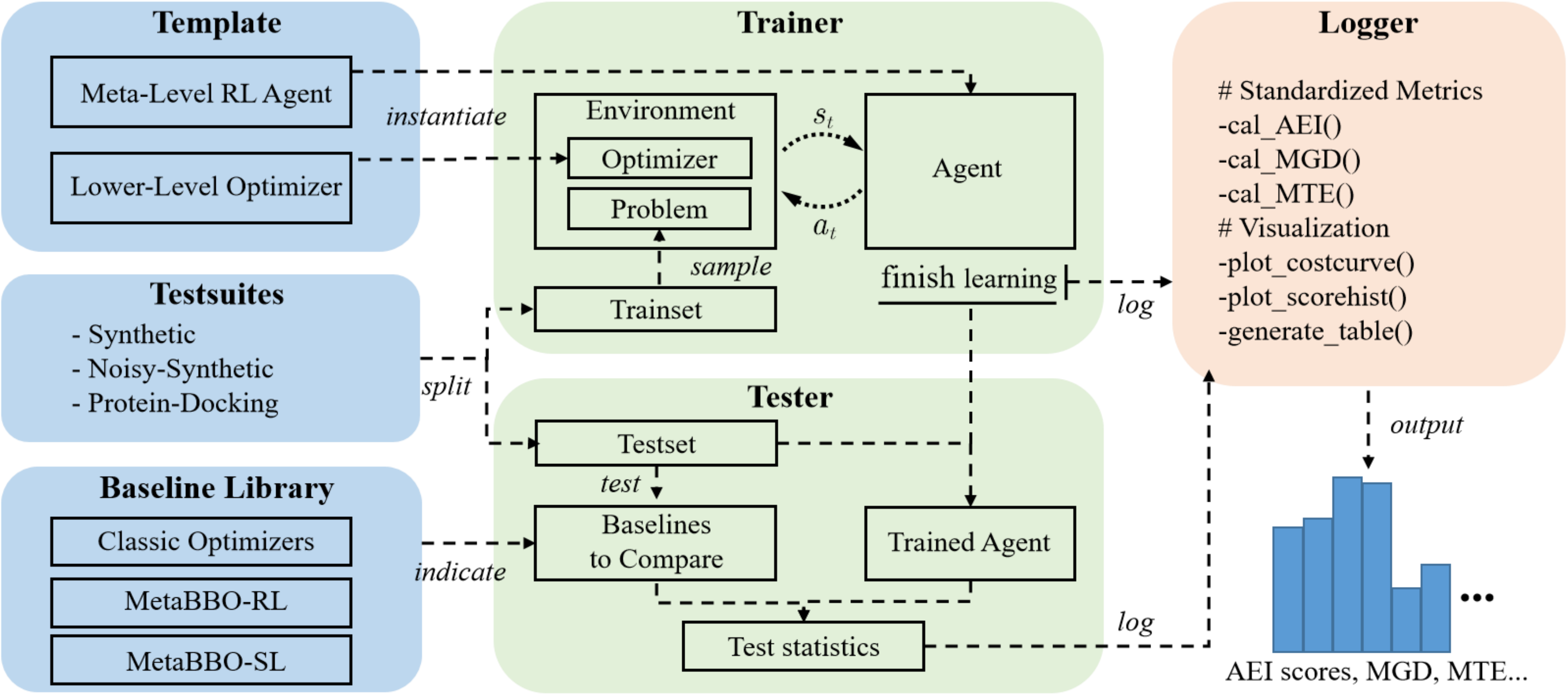
- Lack of template coding and automation for MetaBBO with RL algorithm
- Limited benchmark task instances and up-to-date baseline implementations
- Designed for only evaluating optimization performance, while the evaluation of learning effectiveness is omitted

Table 1: Comparison to BBO benchmarks. We report *#Problem*: the number of problems (*#synthetic* + *#realistic*); *#Baseline*: the number of baselines; *Template*: Template coding support; *Automation*: automated train/test workflow support; *Customization*: configurable settings; *Visualization*: visualization tools support; and *RLSupport*: Gym-style [33] RL benchmark.

	#Problem	*#Baseline*	*Template*	*Automation*	*Customization*	*Visualization*	*RLSupport*
COCO [28]	54+0	2	✓	✓	×	✓	×
CEC [34, 36, 37]	28+0	0	×	×	×	×	×
IOHprofiler [38, 39]	24+0	0	✓	×	✓	✓	×
Bayesmark [40, 41]	0+228	10	✓	✓	×	×	×
Zigzag [42, 43]	4+0	0	×	×	✓	×	×
MetaBox	54+280	19	✓	✓	✓	✓	✓

Overview of MetaBox

- Template scripts
- Workflow automation
- Broad Testsuites
- Baseline Library
- Visualization
- Perfromance Metrics



Blueprint of our MetaBox platform.

Our Proposed MetaBox Platform

To simplify the development of MetaBBO-RL and ensure an automated workflow

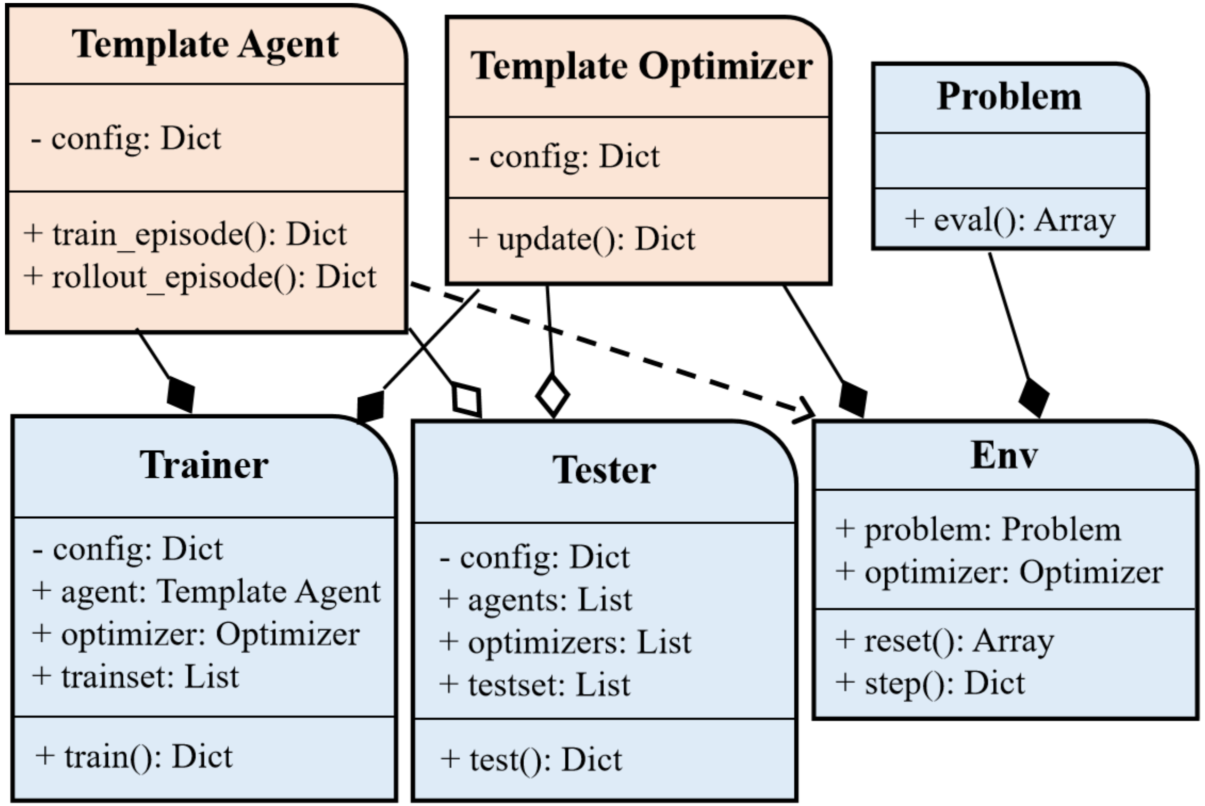
1) *MetaBBO-RL Template:*
Two components: the meta-level RL agent and the low-level optimizer; unified interface protocol;

2) *Automated Train-Test-Log* procedure:
run_experiment() command

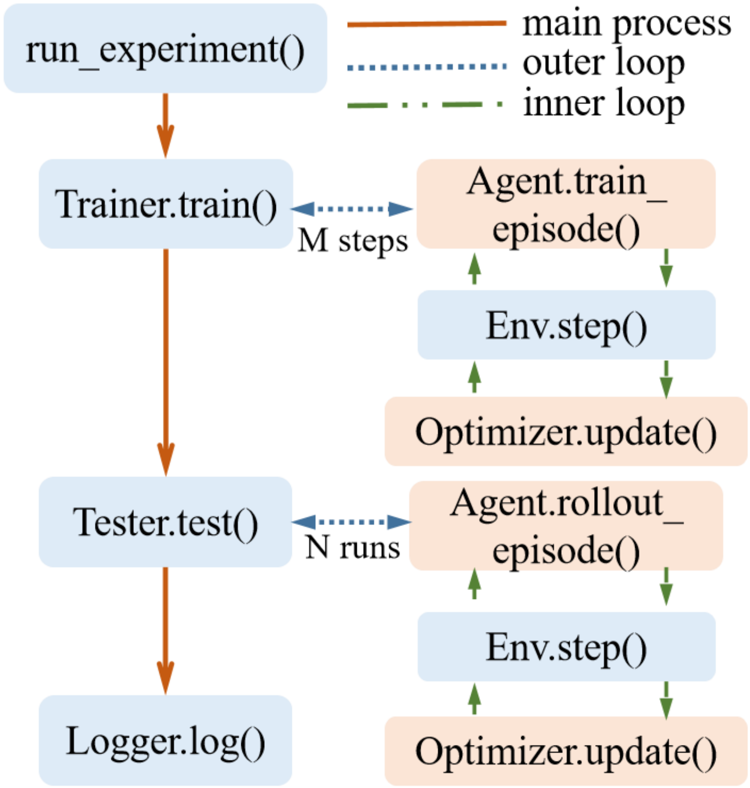
Template coding & workflow automation

To simplify the development of MetaBBO-RL and ensure an automated workflow

MetaBBO-RL Template



Workflow Automation



Our Proposed MetaBox Platform

To facilitate broad and standardized comparison studies

1) *large-scale MetaBox testsuite:*
over 300 benchmark problems with diverse landscape characteristics; inherits problem definitions from the well-known COCO platform and the Protein-Docking benchmark (version 4.0).

2) *Baseline Library*:
a wide range of classic optimizers, up-to-date MetaBBO-RL approaches, a MetaBBO-SL approach.

Large-scale Testsuites

To facilitate broad and standardized comparison studies

MetaBox Testsuites

Synthetic

24 #instance
COCO:bbob

Noisy-Synthetic

30 #instance
COCO:bbob-noisy

Protein-Docking

280 #instance
Protein-Docking V4

Adjustable difficulty: proportion of training and testing sets (easy: 75% : 25%, difficult: 25% : 75%)
Customized problem dimension

Baseline Library

To facilitate broad and standardized comparison studies

Baseline Library

Classic

11 #baseline
Classical black box optimizer

MetaBBO-RL

7 #baseline
Up-to-date MetaBBO-RLs

MetaBBO-SL

1 #baseline
One MetaBBO-SL method

Implemented following the template
Compatible with existing libraries

Our Proposed MetaBox Platform

To comprehensively evaluate the effectiveness of MetaBBO-RL approaches

1) *three Standardized Metrics:*
Aggregated Evaluation Indicator (AEI)
Meta Generalization Decay (MGD)
Meta Transfer Efficiency (MTE)

2) *a tutorial large-scale comparison study*:
using *Baseline Library*, evaluate them on *MetaBox testsuite* by the proposed *Standardized Metrics*.

Standardized evaluation metrics

To comprehensively evaluate the effectiveness of MetaBBO-RL approaches

Aggregated Evaluation Indicator (AEI)

AEI aggregate three traditional BBO performance metrics:
1. the best objective value,
2. the budget to achieve a predefined accuracy (convergence rate)
3. the runtime complexity

$$AEI = \frac{1}{K}\sum_{k=1}^{K} e^{Z^k_{obj} + Z^k_{com} + Z^k_{res}}$$

Meta Generalization Decay (MGD)

MGD assess the generalization performance of MetaBBO-RL for unseen tasks.

$$MGD(A, B) = 100 \times (1 - \frac{AEI_A}{AEI_B})\%$$

Where A, B are two different problem set

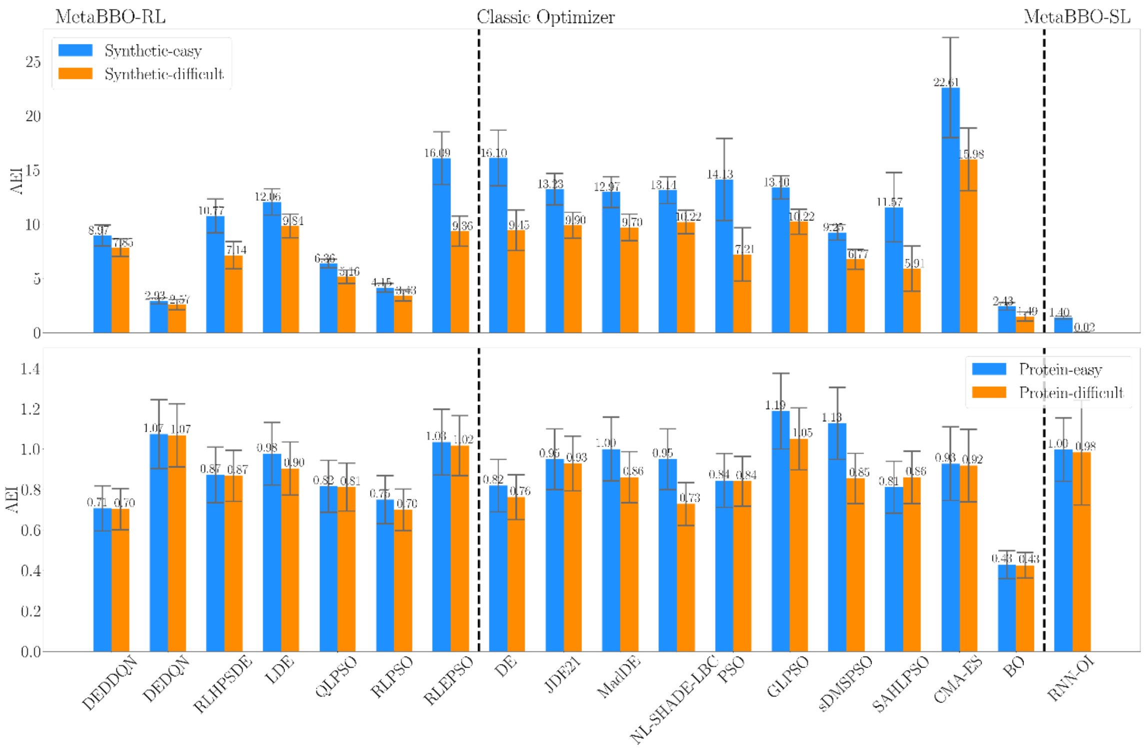
Meta Transfer Efficiency (MTE)

MTE evaluate the transfer learning capacity of a MetaBBO RL approach.

$$MTE(A, B) = 100 \times (1 - \frac{T_{\text{finetune}}}{T_{\text{scratch}}})\%$$

Experiment results

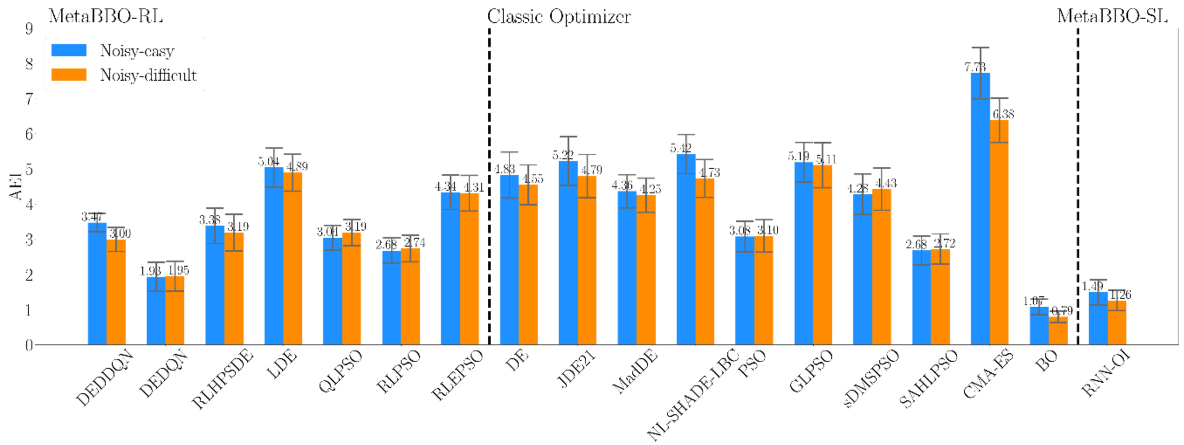
To comprehensively evaluate the effectiveness of MetaBBO-RL approaches



observations:

classic optimizer vs MetaBBO-RL

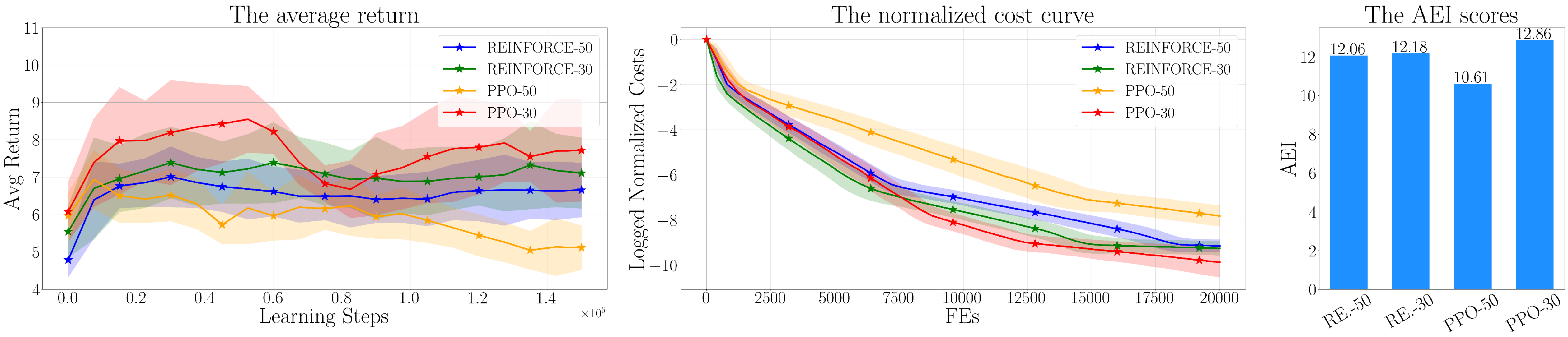
Robustness among different test suits



Comparison of different baseline (Meta)BBO methods (AEI)

Experiment results

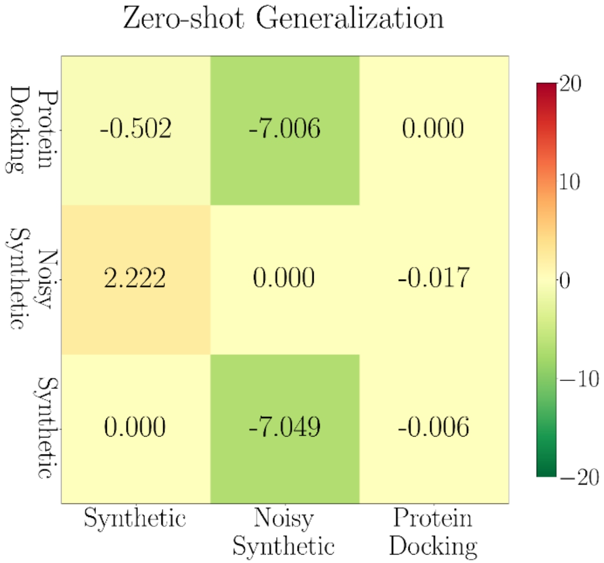
To comprehensively evaluate the effectiveness of MetaBBO-RL approaches

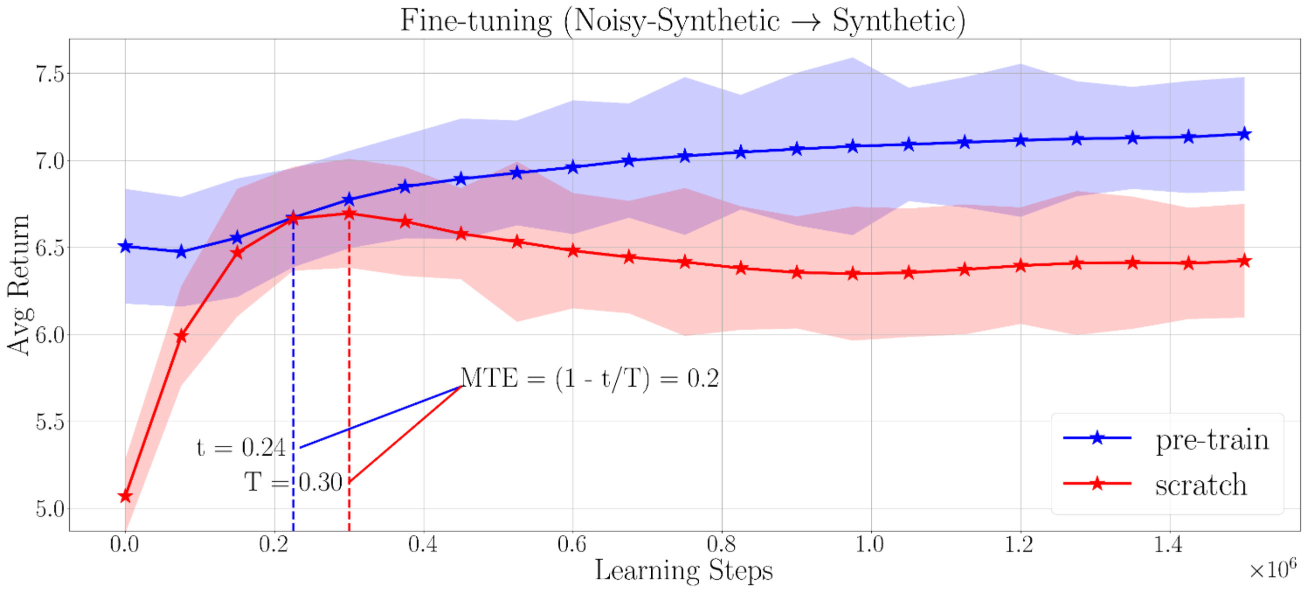


Hyper-tuning a MetaBBO-RL approach (LDE)

Experiment results

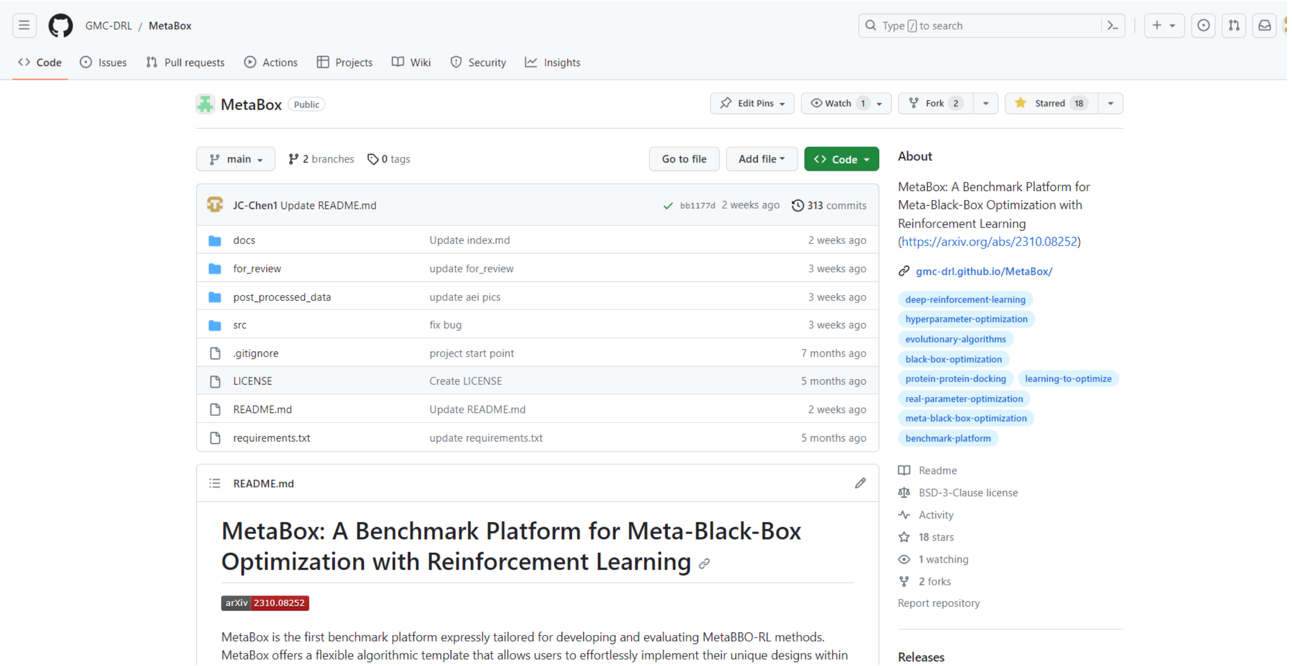
To comprehensively evaluate the effectiveness of MetaBBO-RL approaches

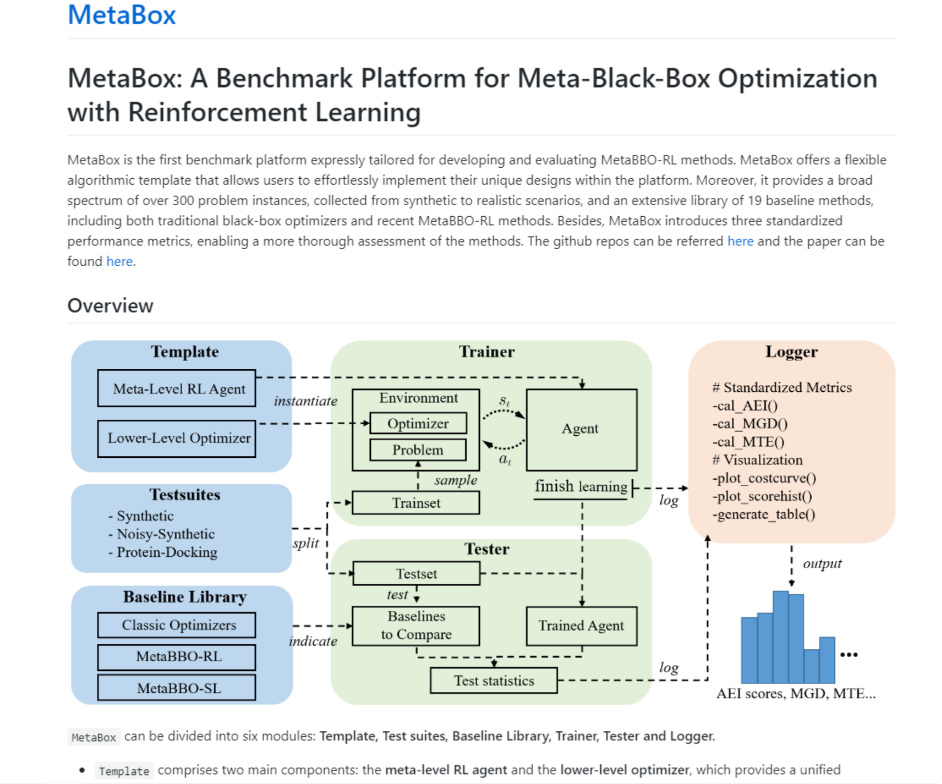




Investigating generalization and transfer learning performance (LDE)

Open source





MetaBox can be accessed on https://github.com/GMC-DRL/MetaBox

Github code

User guides can be accessed on https://gmc-drl.github.io/MetaBox/

Discussion And Conclusion

Contributions:

1. provide the first unified benchmark platform,
2. Simplify coding towards efficient researching,
3. provide broad testsuites and baselines for comprehensive comparison
4. provide novel evaluation metrics for in-depth analysis.

Key observation:

1. MetaBBO-RL vs hand-crafted optimizer (performance, robustness)
2. room for improvement: to discover more effective designs in both meta-level agents and low-level optimizers.
3. interpreting the generalization and transfer effects in MetaBBO-RL can be challenging

Future improvement:

Parallel technique; Testsuite; Baseline library…