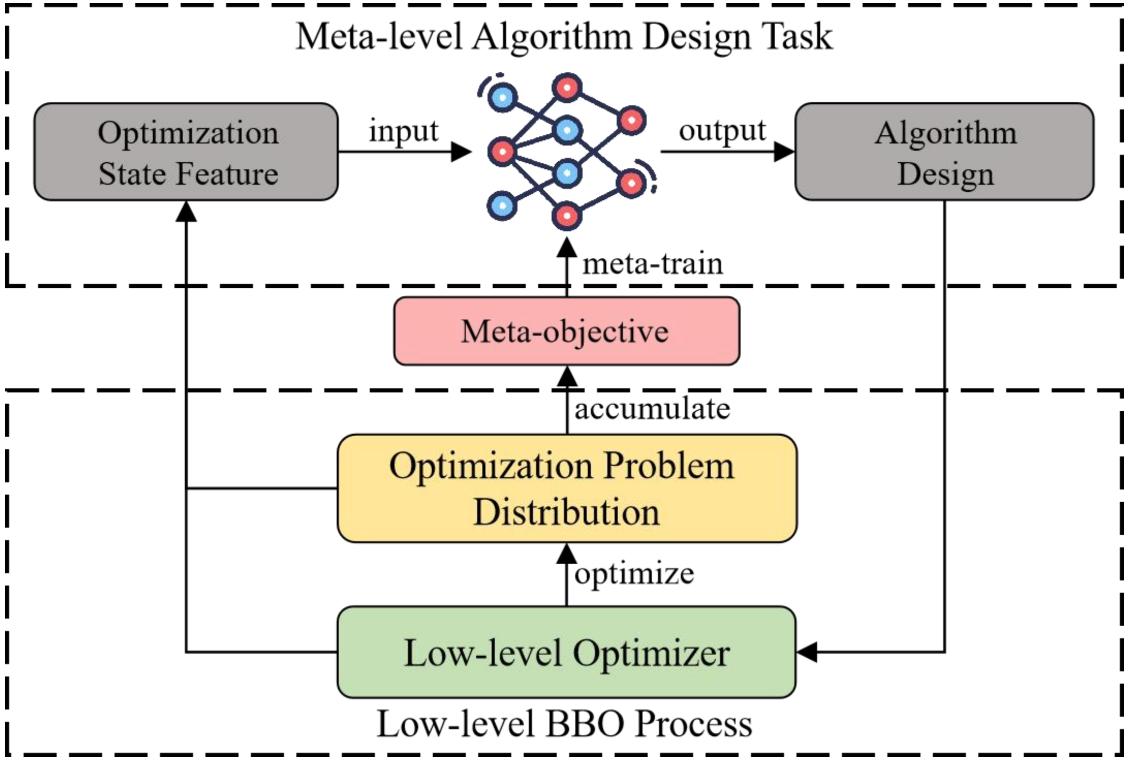
Meta-Black-Box-Optimization through Offline Q-function Learning

Zeyuan Ma, Zhiguang Cao, Zhou Jiang, Hongshu Guo, Yue-Jiao Gong

Part I: What is Meta-Black-Box-Optimization (MetaBBO)?



MetaBBO leverages the generalization strength of Meta-learning to enhance the optimization performance of BBO algorithms in the minimal expertise cost [1] [2].

Bi-level Paradigm:

$$\mathbb{J}(\theta) = \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{H} Perf(A, c_i^t, p_i)$$

$$c_i^t = \pi_\theta(s_i^t), \quad s_i^t = \mathrm{sf}(A, p_i, t)$$

Meta-level Algorithm Design Task:
The policy π_θ is trained to dictates desired algorithm design c by conditioning the optimization state feature s.

Low-level Optimization Task:
The algorithm A adopts the dictated design to optimize a distribution of problems $\mathcal{I}$, providing meta-performence $Perf(\cdot)$ for training the policy.

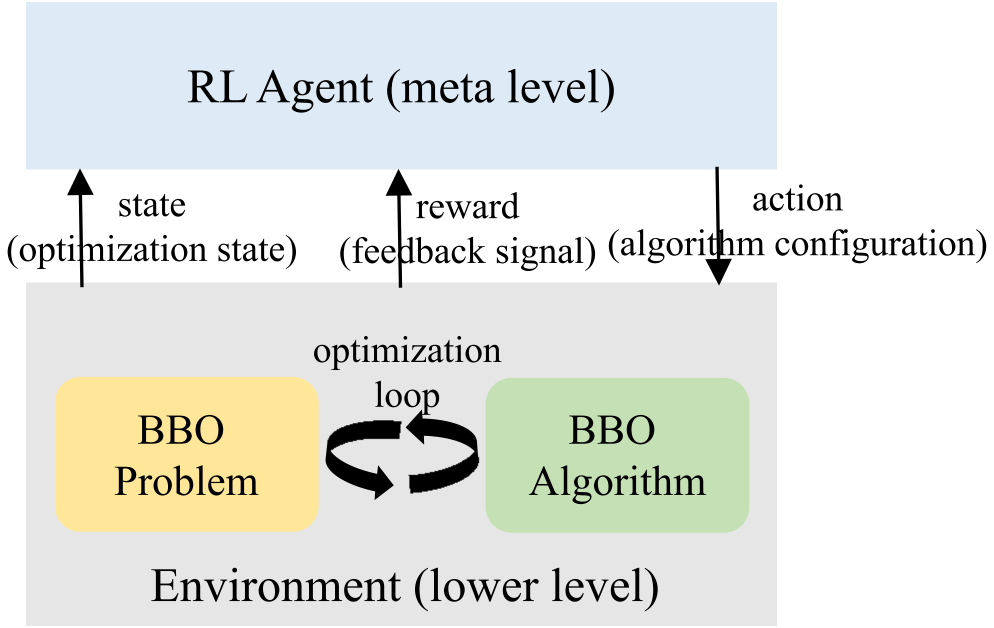
[1] Ma Zeyuan, et al. MetaBox: A Benchmark Platform for Meta-Black-Box Optimization with Reinforcement Learning. NeurIPS 2023.
[2] Ma, Zeyuan, et al. "Toward automated algorithm design: A survey and practical guide to meta-black-box-optimization." IEEE TEVC (2025).

Part I: What is Meta-Black-Box-Optimization?

In this work, we focus on a particular MetaBBO algorithm design task: Dynamic Algorithm Configuration

Meanwhile, we focus on a particular learning methodology: Reinforcement Learning

A general workflow of using RL for DAC can be instantiated from MetaBBO as below [1] [2]:



[1] Xue K et al. Multi-Agent Dynamic Algorithm Configuration. NeurIPS 2022.

[2] Ma Z et al. Auto-Configuring Exploration-Exploitation Tradeoff in Evolutionary Computation via Deep Reinforcement Learning. GECCO 2024.

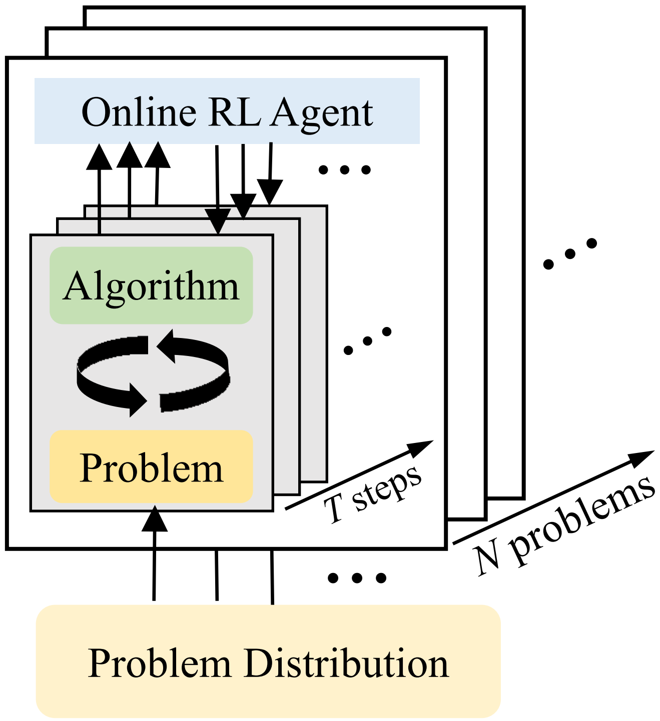
South China University of Technology

SINGAPORE MANAGEMENT UNIVERSITY

Part II: Motivation?

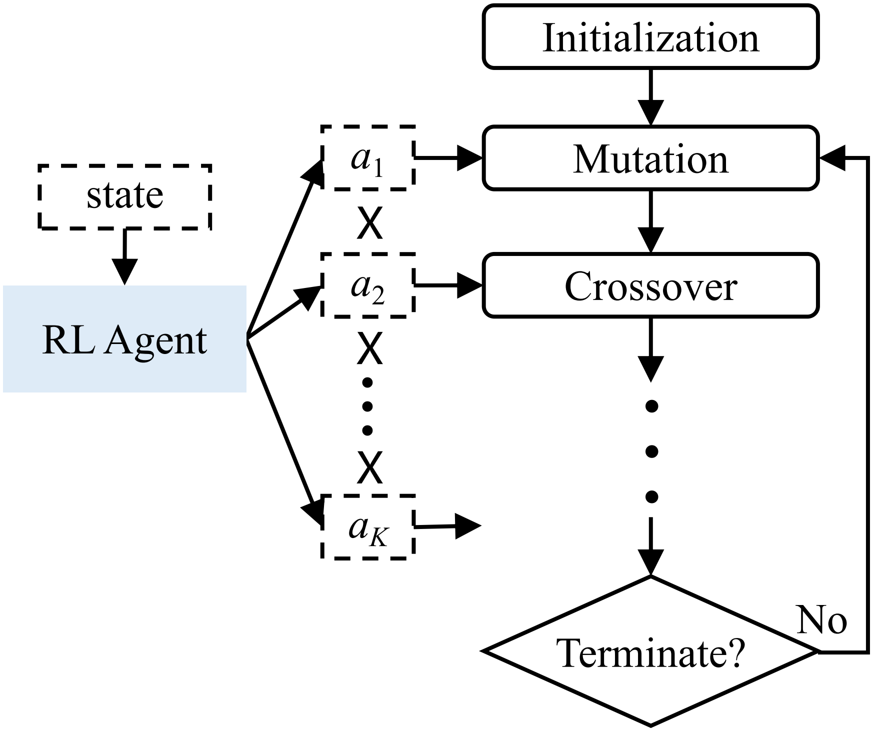
➢ Problematic Efficiency

In BBO scenarios, the collection of trajectories is expensive or time-consuming, making the efficiency of the online learning paradigms in existing MetaBBO works problematic.



➢ Massive Configuration Space

Existing BBO Algorithms usually contain many controllable hyper-parameters, making it difficult to search for the optimal algorithm configuration policy and slowing down the training.

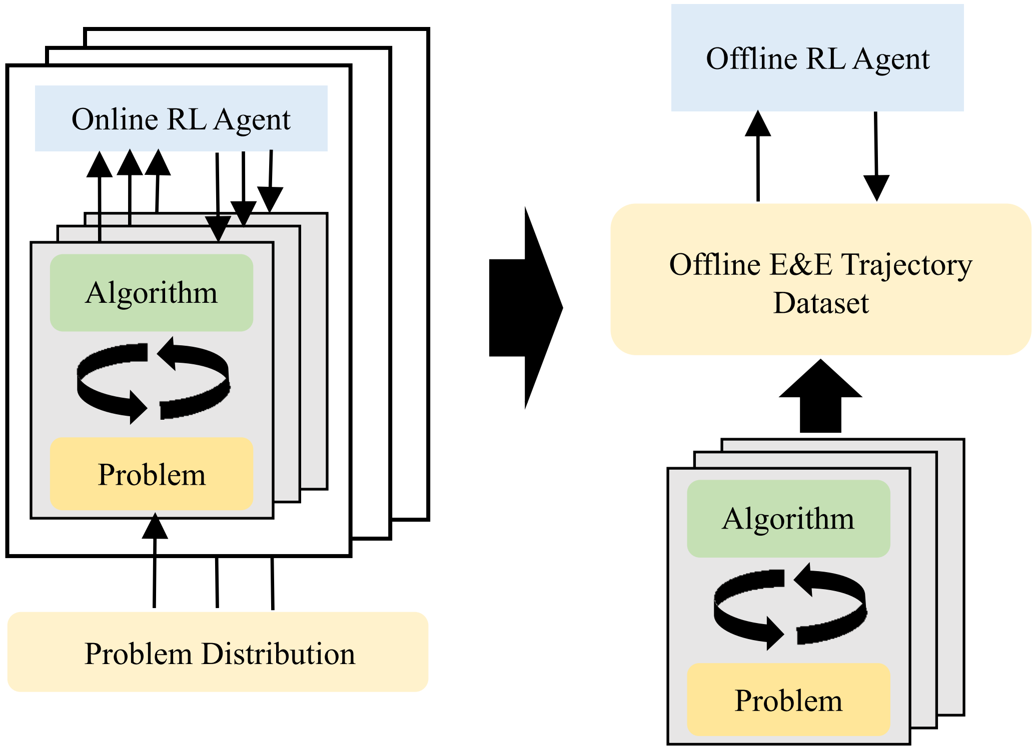


South China University of Technology

SMU SINGAPORE MANAGEMENT UNIVERSITY

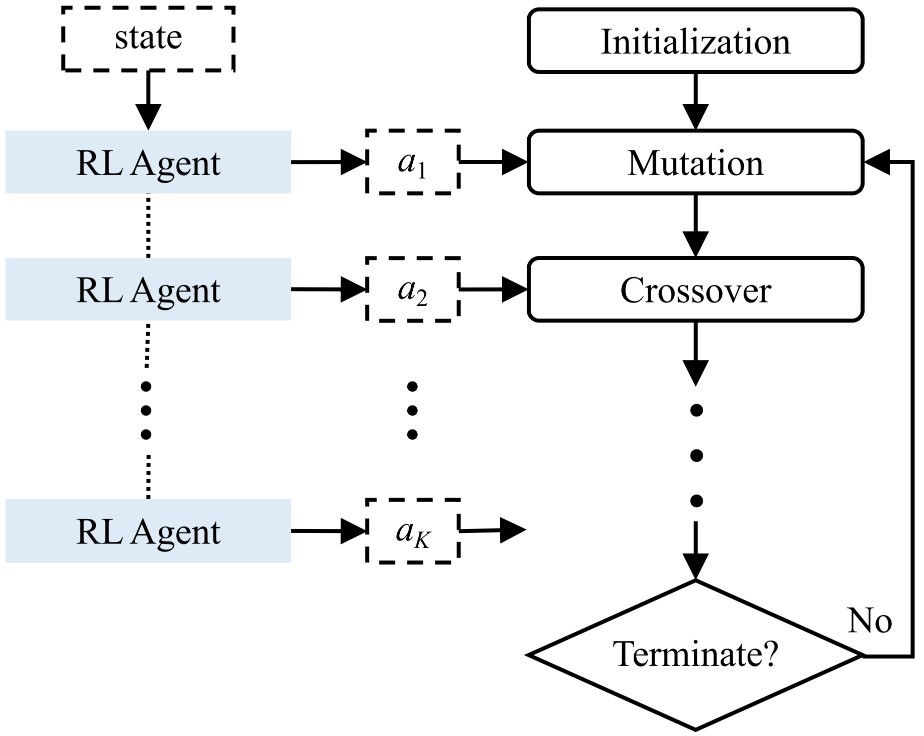
Part II: Motivation?

➢ Problematic Efficiency



We collect offline DAC experience trajectories from both strong MetaBBO baselines and a random policy to provide exploitation and exploration data used for robust training.

➢ Massive Configuration Space

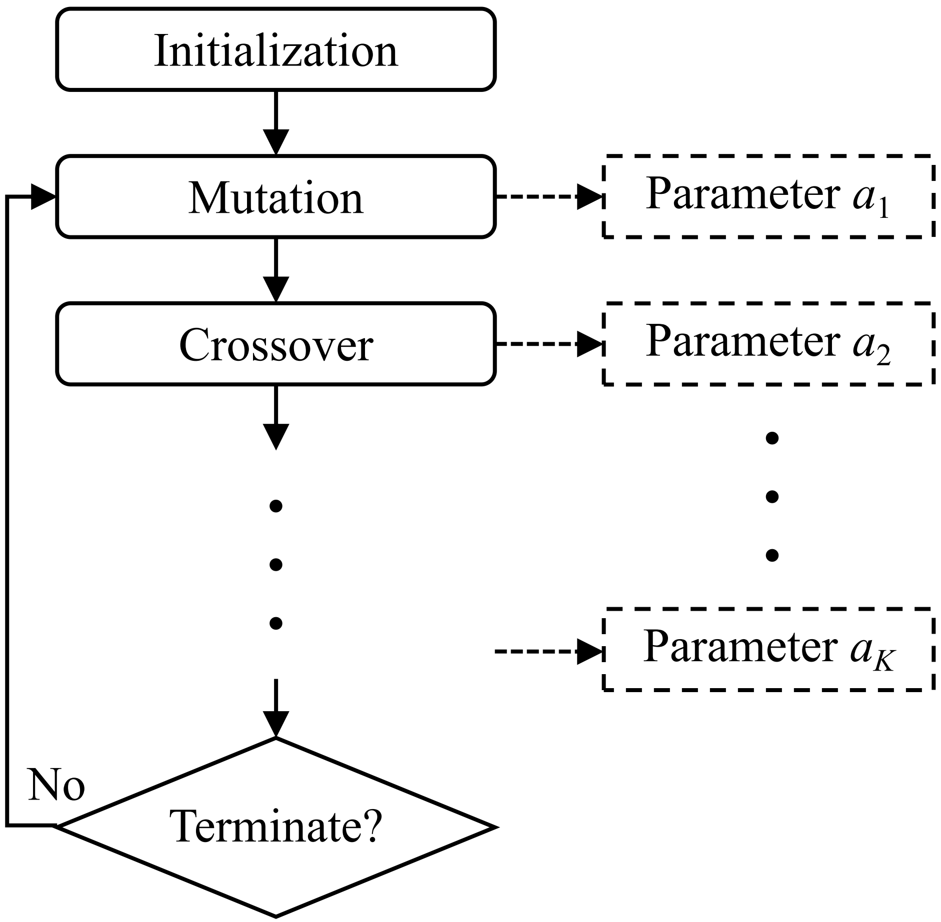


We transform DAC task into a long-sequence decision process and introduce a Q-function decomposition scheme to represent each hyper-parameter as a single action step.

华南理工大学 South China University of Technology

SMU SINGAPORE MANAGEMENT UNIVERSITY

Part III: Problem Formulation



- Transform dynamic algorithm configuration (DAC) task into a long-sequence decision process and introduce a Q-function decomposition scheme to represent each hyper-parameter as a single action step.

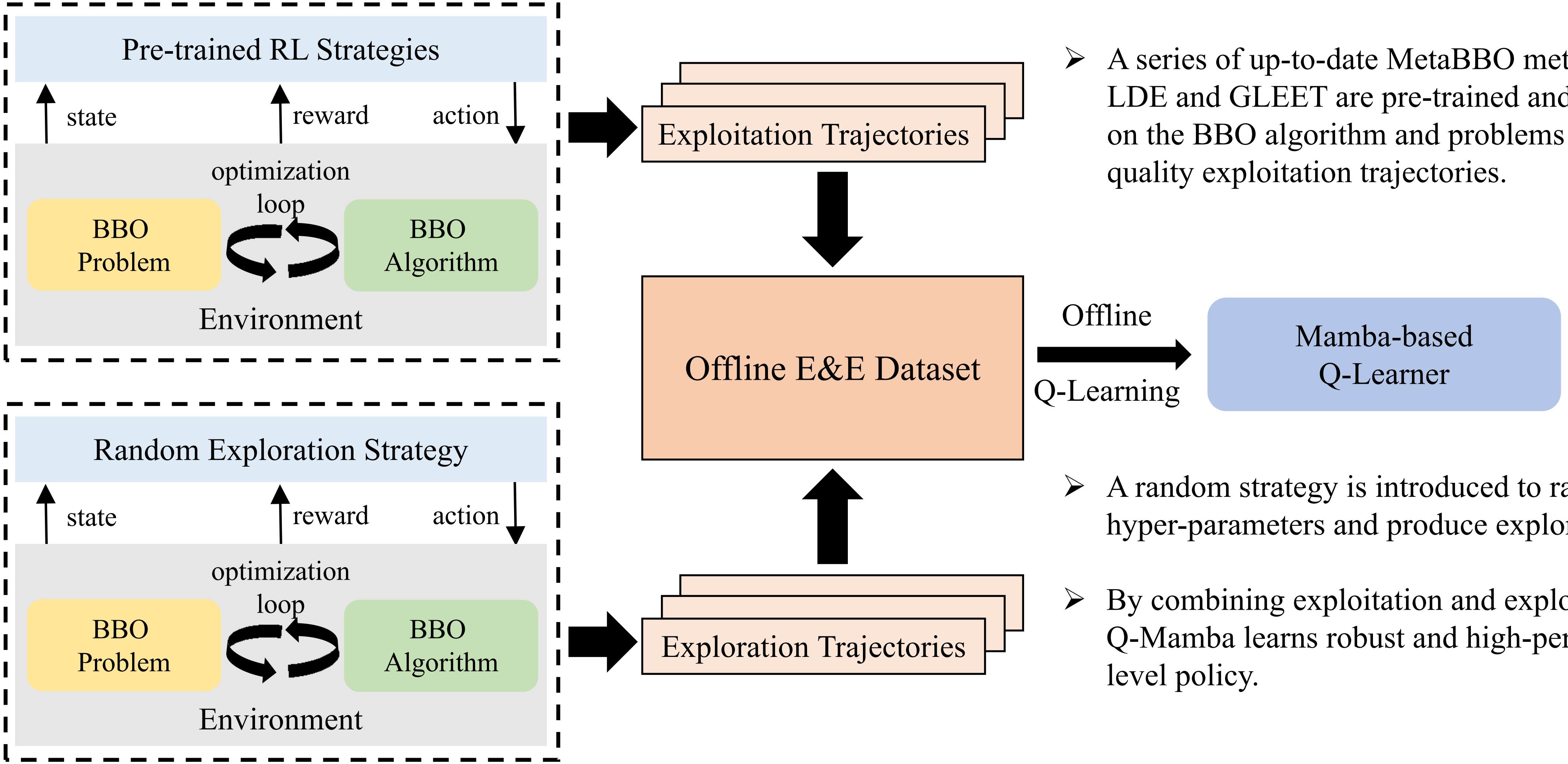
$$Q(a_{1:K}^t|s^t) \leftarrow R(s^t, a_{1:K}^t) + \gamma \max_{a_{1:K}^{t+1}} Q(a_{1:K}^{t+1}|s^{t+1})$$

$$Q(a_i^t|s^t) \leftarrow \begin{cases} \max_{a_{i+1}^t} Q(a_{i+1}^t|s^t, a_{1:i}^t), & if \quad i < K \\ R(s^t, a_{1:K}^t) + \gamma \max_{a_1^{t+1}} Q(a_1^{t+1}|s^{t+1}). & \\ & if \quad i = K \end{cases}$$

- The meta-objective of MetaBBO is to search the optimal policy $\pi_{\theta*}$ that maximizes the expectation of accumulated performance improvement over all problem instances in the training set:

$$\theta^* = \arg\max_\theta \frac{1}{N} \sum_{j=1}^{N} \sum_{t=1}^{T} R(s^t, a_{1:K}^t|\pi_\theta)$$

Part IV: Offline E&E Dataset Collection



- A series of up-to-date MetaBBO methods: RLPSO, LDE and GLEET are pre-trained and then rollouted on the BBO algorithm and problems to collect high quality exploitation trajectories.

- A random strategy is introduced to randomly control the hyper-parameters and produce exploration trajectories.

- By combining exploitation and exploration experience, Q-Mamba learns robust and high-performance meta-level policy.

Part V: Conservative Q-learning Loss

We represent each hyper-parameter as a single action step in the decision process and learn the decomposed sequential Q-function through offline RL to improve the training efficiency of MetaBBO. A compositional Q-loss which enhances the offline learning by removing distributional shift is proposed.

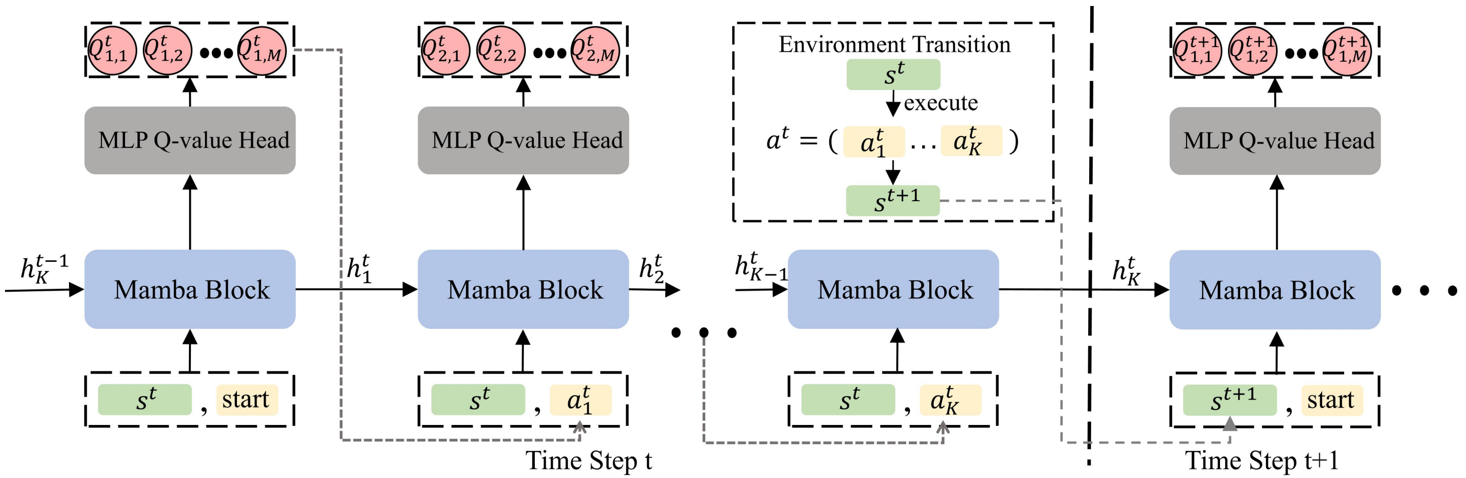
$$J(\tau|\theta) = \sum_{t=1}^{T}\sum_{i=1}^{K}\sum_{j=1}^{M} J(Q^t_{i,j}|\theta) = \begin{cases} \frac{1}{2}(Q^t_{i,j} - \max\limits_{j} Q^t_{i+1,j})^2, & if \quad i < K, j = a^t_i \\ \frac{\beta}{2}\left[Q^t_{i,j} - (r^t + \gamma \max\limits_{j} Q^{t+1}_{1,j})\right]^2, & \\ & if \quad i = K, j = a^t_i \\ \frac{\lambda}{2}(Q^t_{i,j} - 0)^2, & if \quad j \neq a^t_i \end{cases}$$

- The first two branches are TD errors following the Bellman backup for decomposed Q-function, with weight β=10 on the last action dimension to reinforce the learning on this dimension.
- The conservative regularization introduced in offline RL method CQL, which is used to relieve the over-estimation due to the distribution shift. We set λ=1 in this paper to strike a good balance.

South China University of Technology

SMU SINGAPORE MANAGEMENT UNIVERSITY

Part VI: Mamba-based Q-Learner



- MetaBBO task features long-sequence process that involves thousands of decision steps since there are hundreds of optimization steps and K hyper-parameters to be decided per optimization step. Mamba is adopted since it parameterizes the dynamic parameters as functions of input state token, which facilitate flexible learning of long-term and short-term dependencies from historical state sequence.

- Mamba equips itself with hardware-aware I/O computation and a fast parallel scan algorithm: PrefixSum, which allows Mamba has the same memory efficiency as highly optimized FlashAttention

Part VII: Some Empirical Obervation

- **Experiment Setup**:

Training dataset: Three low-level BBO algorithms with 3, 10 and 16 hyper-parameters sampled from the algorithm space in ConfigX. The problem distribution includes 16 problems (5D-50D) from CoCo BBOB Testsuites which contains 24 synthetic functions.

In-Dsitribution Test Set: The three low-level BBO algorithms on the 8 problems (5D-50D) from CoCo BBOB Testsuites which have not been used for training.

Out-Of-Distribution Test Sets: A continuous control neuroevolution task on a 2-layer MLP policy for Mujoco.

Training Settings: Decomposed Offline Q-function Learning, 300 epoch, learning rate 5e-3.

Baselines:

- **Online**: **RLPSO** that uses simple MLP , **LDE** that facilitates LSTM and **GLEET** that uses Transformer architecture.
- **Offline**: **DT**, **DeMa**, **QDT** and **QT** that apply conditional imitation learning on the E&E dataset, and **Q-Transformer** that uses similar Q-value decomposition scheme as Q-Mamba, while the neural network architecture is Transformer.

Part VII: Some Empirical Obervation

- **<u>In-distribution Generalization</u>**:

Table 1. Performance comparison between Q-Mamba and other online/offline baselines. All baselines are tested on unseen problem instances within the training distribution P_{bbob}. We additionally present the averaged training/inferring time of all baselines in the last row.

	Online			Offline					
	RLPSO (MLP)	LDE (LSTM)	GLEET (Transformer)	DT	DeMa	QDT	QT	Q-Transformer	Q-Mamba
$Alg0$ $K=3$	9.855E-01 ±9.038E-03	9.563E-01 ±1.830E-02	9.616E-01 ±3.110E-03	9.325E-01 ±2.680E-02	9.492E-01 ±2.467E-02	9.683E-01 ±2.131E-02	9.729E-01 ±1.934E-02	9.666E-01 ±1.975E-02	**9.889E-01 ±7.779E-03**
$Alg1$ $K=10$	9.953E-01 ±3.322E-03	9.877E-01 ±1.118E-02	9.938E-01 ±2.834E-03	6.764E-01 ±1.193E-01	9.015E-01 ±1.688E-02	9.917E-01 ±5.454E-03	9.955E-01 ±3.115E-03	9.951E-01 ±3.487E-03	**9.973E-01 ±2.441E-03**
$Alg2$ $K=16$	9.914E-01 ±4.497E-03	9.904E-01 ±6.306E-03	9.910E-01 ±5.846E-03	8.706E-01 ±3.951E-02	9.159E-01 ±2.015E-02	9.919E-01 ±7.456E-03	9.926E-01 ±6.874E-03	9.895E-01 ±6.754E-03	**9.950E-01 ±9.981E-03**
Avg Time	28h / 11s	28h / 12s	25h / 13s	13h / 10s	12h / 10s	20h / 12s	20h / 12s	16h / 11s	13h / 10s

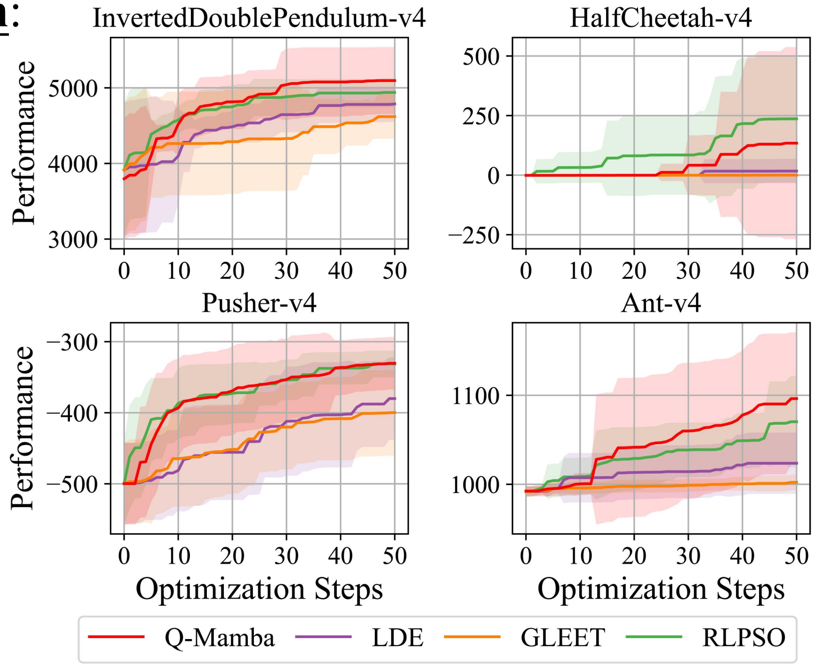
- Q-Mamba effectively achieves competitive or even superior optimization performance to prior online/offline baselines.

- Q-Mamba v.s. Online MetaBBO: By learning from the offline E&E dataset, Q-Mamba reduces the training budget which is especially appealing for BBO scenarios where the simulation is expensive or time-consuming.

- Q-Mamba v.s. Offline MetaBBO: The weighted Q-loss function accelerates the learning of the TD error and the Mamba architecture allows selectively remember or forget historical states which avoids the linear time invariance of Transformer.

Part VII: Some Empirical Obervation

- **Out-of-distribution Generalization**:



- While only trained on low-dimensional synthetic problems, Q-Mamba is capable of optimizing the MLP polices which hold thousands of parameters in neuroevolution tasks.

- Compared to online baselines, Q-Mamba is capable of learning effective policy with comparable generalization performance, with only consuming less than half training resources.

South China University of Technology

SMU SINGAPORE MANAGEMENT UNIVERSITY

Thanks for Listening!

Our team page