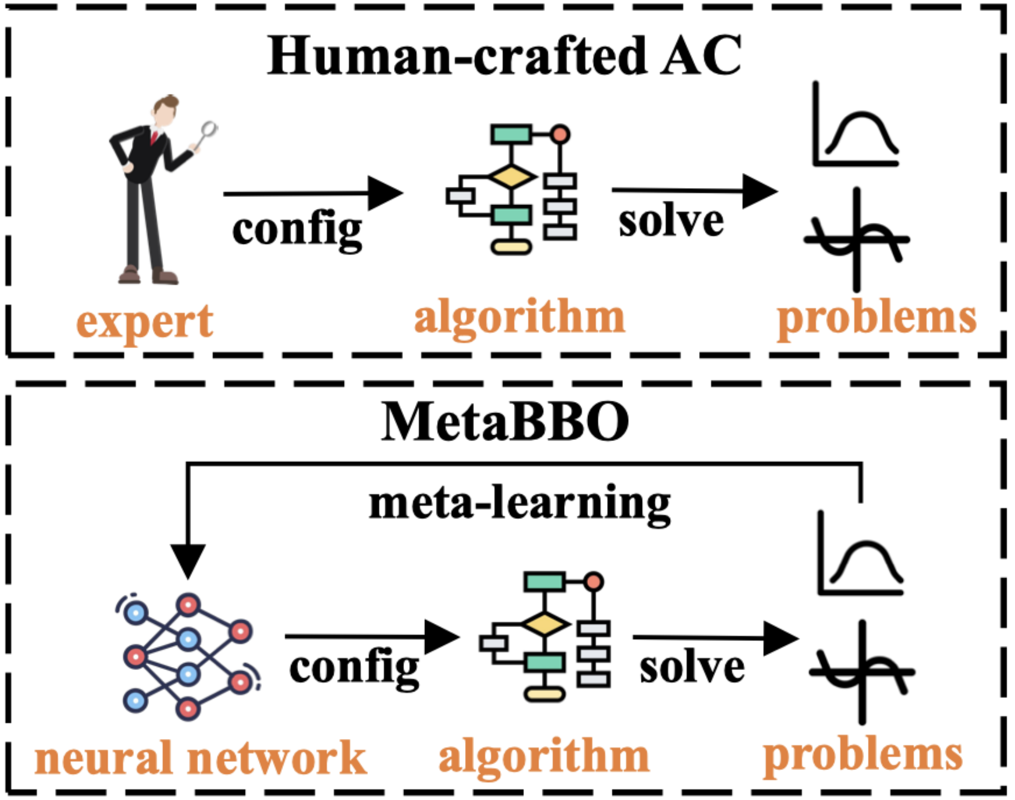
华南理工大学
South China University of Technology
SMU
Singapore Management University
Massachusetts Institute of Technology

ConfigX: Modular Configuration for Evolutionary Algorithms via Multitask Reinforcement Learning

by Yining Ma

Part I: Motivation of ConfigX



What is Meta-Black-Box-Optimization (MetaBBO) for Algorithm Configuration (AC)?

MetaBBO learns a neural network for the automatic configuration of a given Evolutionary Algorithm with meta learning, minimizing expertise costs [1] [2].

MetaBBO has adaptability to unseen problems!

But still require Retraining or even Redesign (e.g., new neural network) for different BBO algorithms.

[1] Ma Z et al. MetaBox: A Benchmark Platform for Meta-Black-Box Optimization with Reinforcement Learning. NeurIPS 2023.

[2] Ma Z et al. Toward Automated Algorithm Design: A Survey and Practical Guide to Meta-Black-Box-Optimization. arXiv preprint arXiv:2411.00625, 2024.

Part I: Motivation of ConfigX

➢ Two different DE algorithms induce different configuration space

Vanilla DE

Algorithm 1: basic DE algorithm

Generate a uniformly distributed random initial population including NP solutions that contain D variables according to $X_{i,j}^{0} = X_{j}^{\min} + rand(0,1)\cdot\left(X_{j}^{\max} - X_{j}^{\min}\right) \quad (i \in [1, NP], j \in [1, D])$
while termination condition is not satisfied
  for i=1 to NP
    Generate three random indexes $r1$, $r2$ and $r3$ with $r1 \neq r2 \neq r3 \neq i$ **//mutation**
    $V_{i}^{G} = X_{r1}^{G} + F\cdot\left(X_{r2}^{G} - X_{r3}^{G}\right)$ **//end mutation**
    $j_{rand} = randind(1, D)$ **//crossover**
    for i=1 to D
      if $rand(0,1) \leq CR$ or $j == j_{rand}$
        $U_{i,j}^{G} = V_{i,j}^{G}$
      else
        $U_{i,j}^{G} = X_{i,j}^{G}$
      end if
    end for **//end crossover**
    if $f(U_{i}^{G}) \leq f(X_{i}^{G})$ **//selection**
      $X_{i}^{G+1} = U_{i}^{G}$
    else
      $X_{i}^{G+1} = X_{i}^{G}$
    end if **//end selection**
  end for
end while

F, Cr, pop_size

SHADE

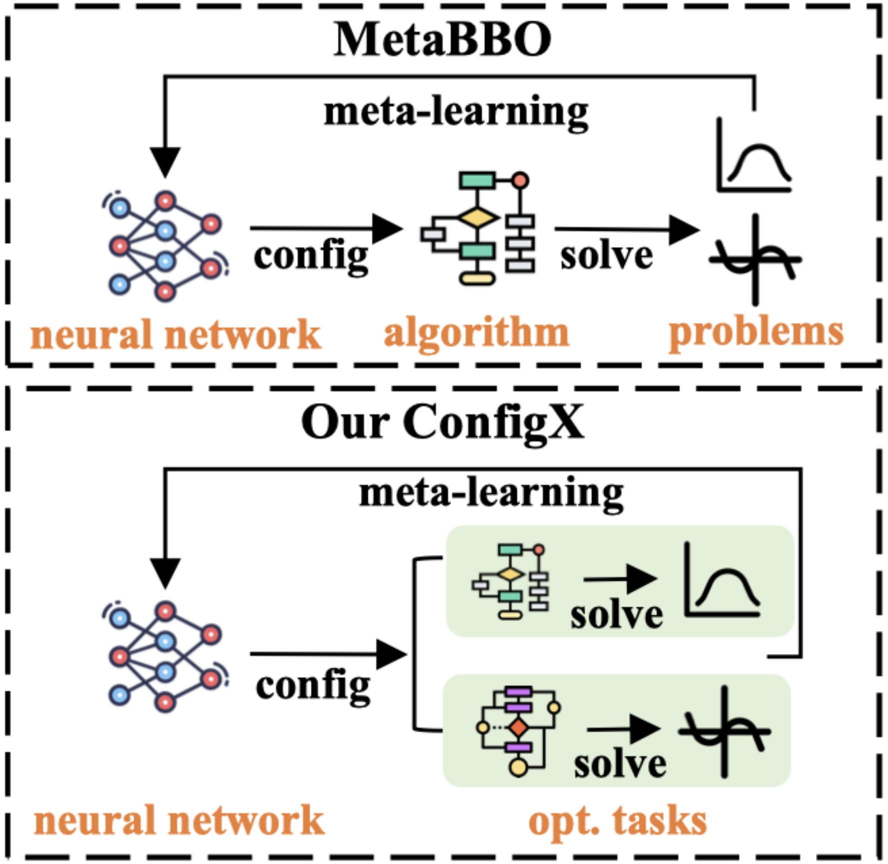
Algorithm 1: SHADE

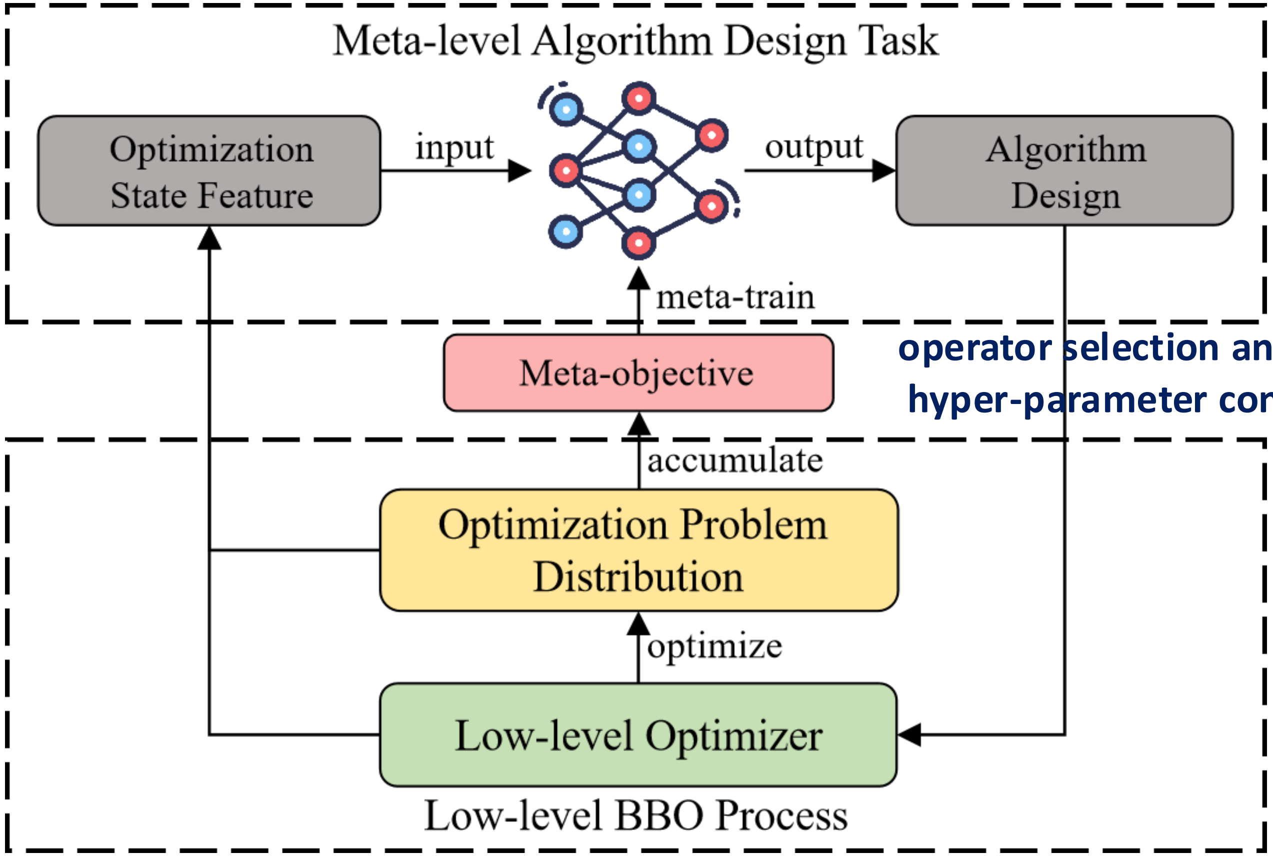
// Initialization phase
$G = 0$;
Initialize population $\boldsymbol{P}_0 = (\boldsymbol{x}_{1,0}, ..., \boldsymbol{x}_{N,0})$ randomly;
Set all values in M_{CR}, M_F to 0.5;
Archive $\boldsymbol{A} = \emptyset$;
Index counter $k = 1$;
// Main loop
while *The termination criteria are not met* **do**
  $S_{CR} = \emptyset, S_F = \emptyset$;
  for $i = 1$ **to** N **do**
    r_i = Select from $[1, H]$ randomly;
    $CR_{i,G} = \mathrm{randn}_i(M_{CR,r_i}, 0.1)$;
    $F_{i,G} = \mathrm{randc}_i(M_{F,r_i}, 0.1)$;
    $p_{i,G} = \mathrm{rand}[p_{min}, 0.2]$;
    Generate trial vector $\boldsymbol{u}_{i,G}$ by current-to-$p$best/1/bin;
  end
  for $i = 1$ **to** N **do**
    if $f(\boldsymbol{u}_{i,G}) \leq f(\boldsymbol{x}_{i,G})$ **then**
      $\boldsymbol{x}_{i,G+1} = \boldsymbol{u}_{i,G}$;
    else
      $\boldsymbol{x}_{i,G+1} = \boldsymbol{x}_{i,G}$;
    end
    if $f(\boldsymbol{u}_{i,G}) < f(\boldsymbol{x}_{i,G})$ **then**
      $\boldsymbol{x}_{i,G} \to \boldsymbol{A}$;
      $CR_{i,G} \to S_{CR}$, $F_{i,G} \to S_F$;
    end
  end
  Whenever the size of the archive exceeds $|\boldsymbol{A}|$, randomly selected individuals are deleted so that $|\boldsymbol{A}| \leq |\boldsymbol{P}|$;
  if $S_{CR} \neq \emptyset$ **and** $S_F \neq \emptyset$ **then**
    Update $M_{CR,k}$, $M_{F,k}$ based on S_{CR}, S_F;
    $k++$;
    If $k > H$, k is set to 1;
  end
end

F, Cr, pop_size, archive_size, memory_size, p_best...

Part II: Our ConfigX Workflow

Can we develop a MetaBBO paradigm that can train an automatic, general-purpose configuration agent (deep model) for diverse EAs?



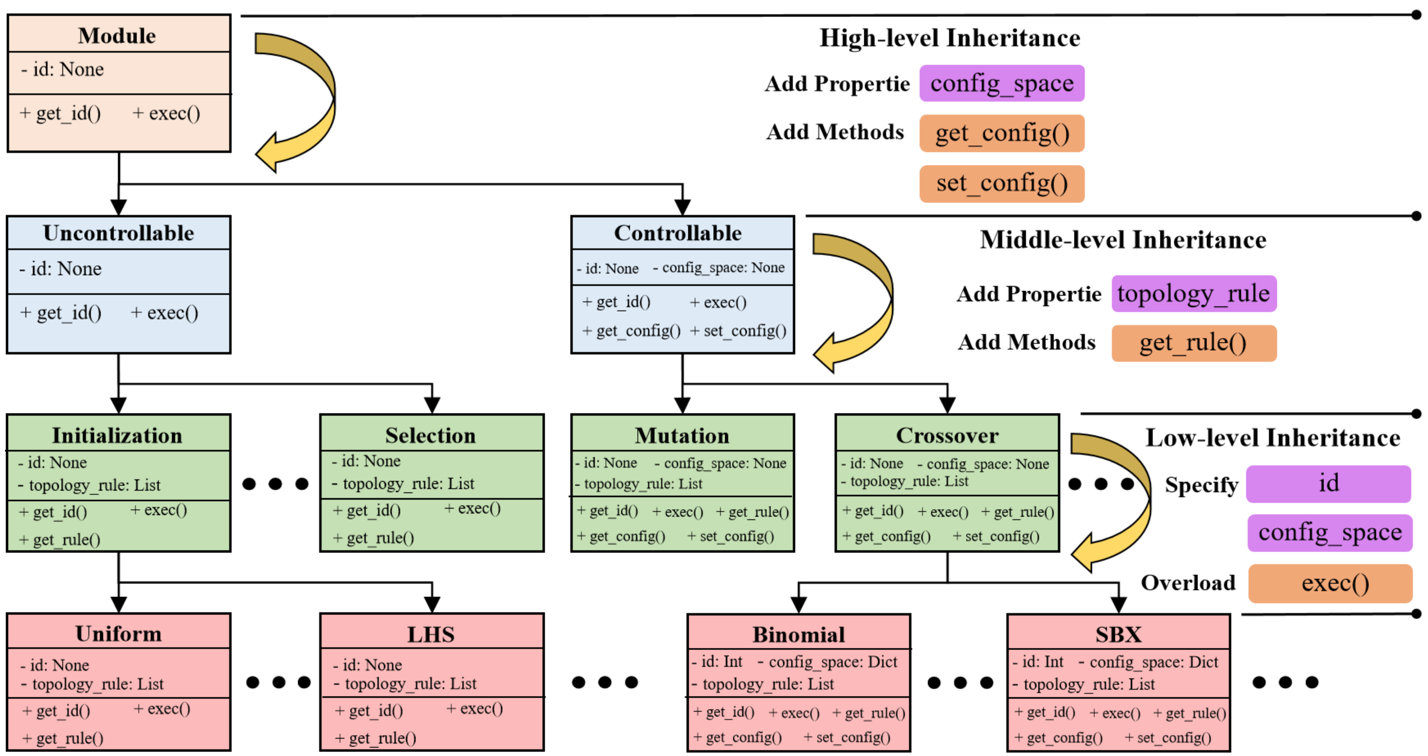


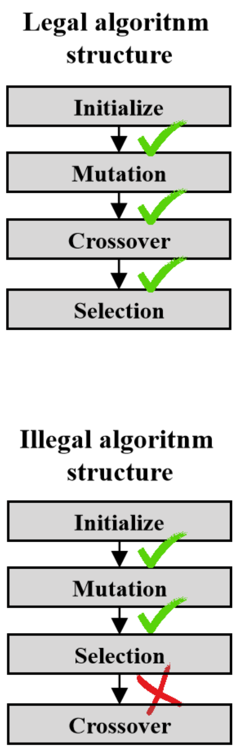
operator selection and/or hyper-parameter control

Part III: Contribution 1 - Modular-BBO

➢ A comprehensive Algorithm Space by Modular-BBO

We propose **Modular-BBO** as a novel system for EA modularization that leverages hierarchical polymorphism in Python to efficiently encapsulate various algorithmic submodules within the EAs.





Part III: Contribution 1 - Modular-BBO

➢ A comprehensive Algorithm Space by Modular-BBO

❖**Modular-BBO has 11 sub-module categories:**

INITIALIZATION, MUTATION, CROSSOVER, PSO_UPDATE, BOUNDARY_CONTROL, SELECTION, MULTI_STRATEGY, NICHING, INFORMATION_SHARING, RESTRT_STRATEGY, POPULATION_REUCTION.

❖**Modular-BBO contains a collection of 100+ variants**

❖**Modular-BBO supports the implementation of many well-known EAs**

JADE, MadDE, LSHADE, L-SHADE-LBC, FDR-PSO, CLPSO, GA, etc

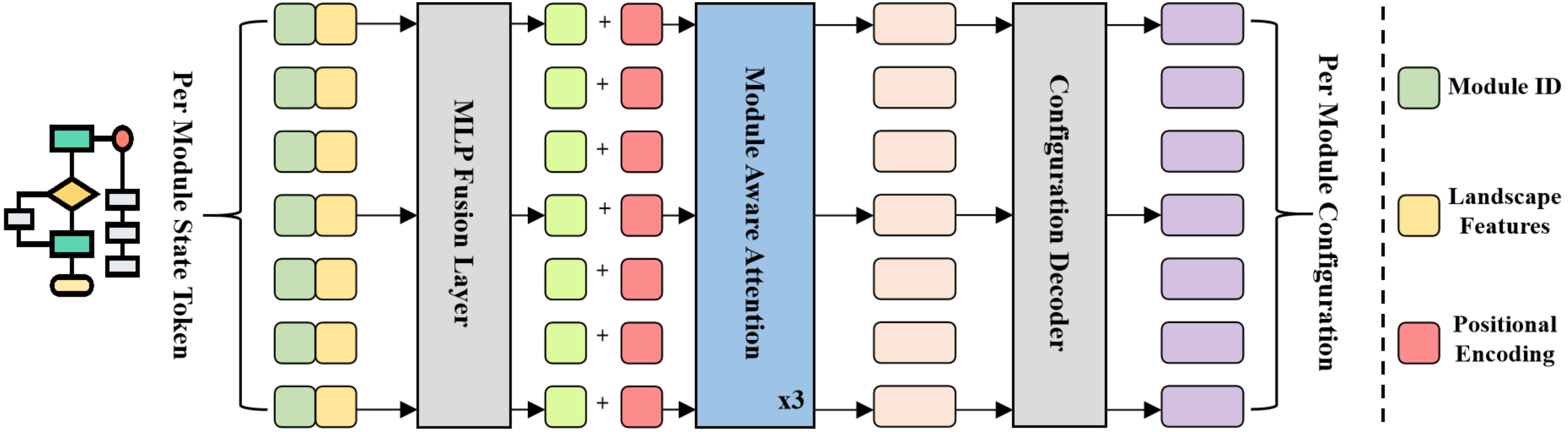
❖**Modular-BBO induces a algorithm space with millions of algorithm structures**

South China University of Technology
SMU Singapore Management University
Massachusetts Institute of Technology

Part IV: Contribution 2 - ConfigX

➢ Train a general-purpose configuration policy

- ❖ **Multi-task Reinforcement Learning**
- ❖ **Network Architecture**



Part V: Experiments

Experiment Setup

Training: 32 algorithms sampled from Modular-BBO (DE); 8 BBO problems from CoCo-BBOB (5D-50D)

Test: 32 unseen algorithms sampled from Modular-BBO (DE); the rest 16 problems in CoCo-BBOB (5D - 50D)

Out-Of-Distribution Test Sets:

1. **Same algorithms** yet **unseen realistic BBO problems** from Protein-Docking and HPO-B
2. **Same optimization problems**, yet **unseen algorithms** sampled from Modular-BBO (PSO/GA)

Training Settings: PPO, 50 epoch, learning rate 1e-3, learn 3 times every 10 sample steps

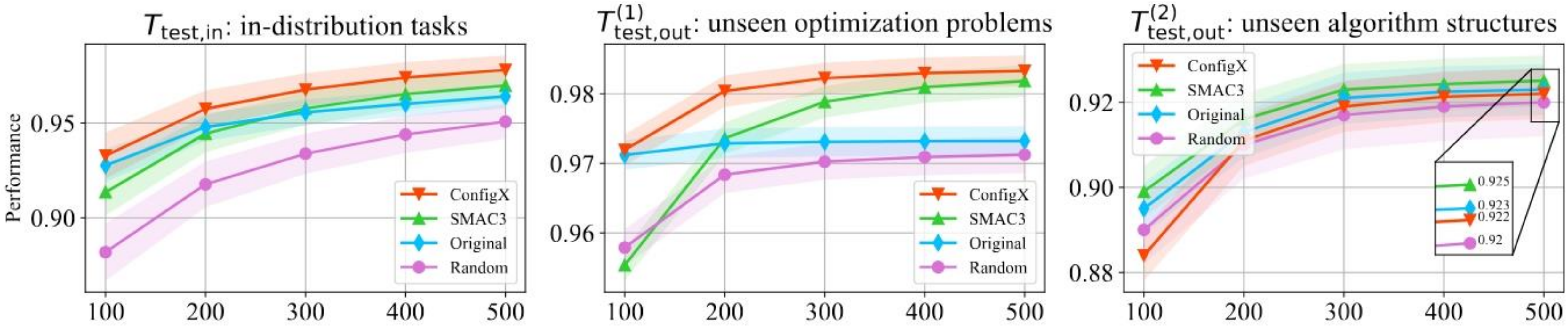
Baselines:

- **Original**: all sub-modules follows the suggested setting in original paper.
- **Random**: hyper-parameters are randomly configured during the optimization process.
- **SMAC3**: state-of-the-are algorithm configuration software based on Bayesian Optimization.

South China University of Technology
SINGAPORE MANAGEMENT UNIVERSITY
Massachusetts Institute of Technology

Part V: Experiments

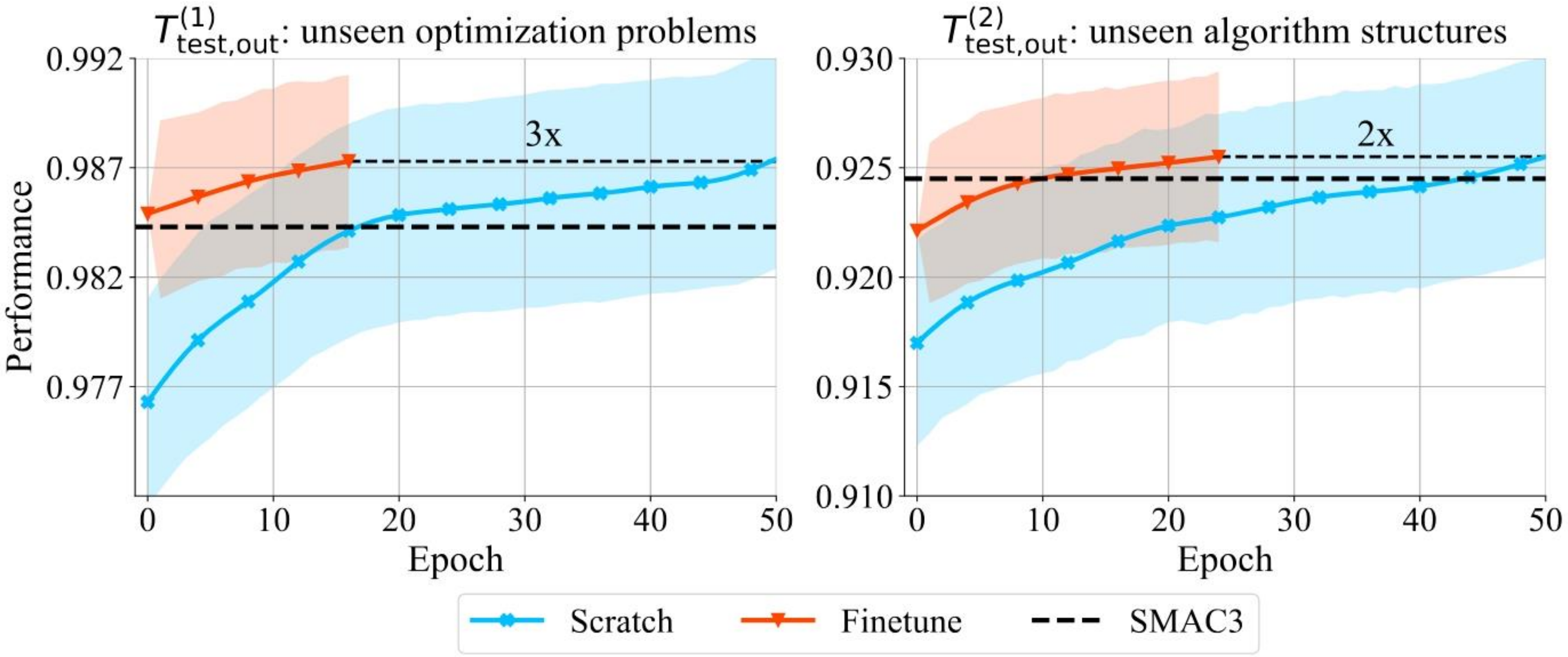
➢ Results



- ConfigX v.s. Random: the multi-task RL training is effective.
- ConfigX v.s. SMAC3: pre-training a model on a small number of task samples could yield superior performance
- Unseen Algorithm Structure: potential generalization failure is observed when algorithm structure is totally different

Part V: Experiments

➢ **Results**



Fine-tuning reduces learning steps by 3x and 2x compared to re-training.

華南理工大學 South China University of Technology

SMU SINGAPORE MANAGEMENT UNIVERSITY

Massachusetts Institute of Technology

paper page

code page

team page