Supplementary Material for "Learning to Transfer for Evolutionary Multitasking"

Sheng-Hao Wu, *Member, IEEE*, Yuxiao Huang, Xingyu Wu, *Member, IEEE*, Liang Feng, *Senior Member, IEEE*, Zhi-Hui Zhan, *Fellow, IEEE*, Kay Chen Tan, *Fellow, IEEE*

Nomenclature

BBOB	Black-box Optimization Benchmark
DE	Differential Evolution
EMT	Evolutionary Multitasking
GA	Genetic Algorithm
KT	Knowledge Transfer
L2T	Learing to Transfer
MTOP	Multi-Task Optimization Problem

S.I. Literature Review

A. Knowledge Transfer for Evolutionary Multitasking

According to the literature review on KT [1], [2], there are two key concerns when designing the KT process for implicit EMT, namely when to transfer and how to transfer. Existing implicit EMT literature has paid research focus on addressing at least one of them and will be briefly reviewed in this subsection.

As the first issue of the implicit KT process, when to transfer refers to deciding whether to trigger a KT process when reproducing each offspring. The KT intensity is a concept that describes the frequency of performing KT operations over a time window of the implicit EMT process. The frequency of performing KT can be controlled by an intensity parameter in a deterministic or probabilistic manner. There are two branches of KT designs on this issue, including fixed-intensity KT and adaptive-intensity KT. The early KT processes are mainly fixed-intensity and the probability of performing KT is unchanged with time, where the intensity-related parameter is usually manually specified by the user before the algorithm running. A representative approach is the multifactorial evolutionary algorithm (MFEA) proposed by Gupta et al. [3]. In MFEA, a random mating probability rmp is defined to control the probability of performing crossover between solutions from different tasks and is fixed along the implicit EMT process. Later, researchers realized that the KT intensity may need to be adjusted according to the evolution status and the level of inter-task synergy. Then researchers began to design adaptive-intensity KT. Bali et al. [4] proposed a mixture sampling model coefficient to estimate the transferability of source tasks to dynamically control rmp online. Chen et al. [5] proposed a feedback-based strategy by reinforcing the transfer intensity when the KT between tasks brings positive results. Liang [6] et al. proposed to control the frequency of the KT according to the convergence state estimated based on the fitness difference at previous generations.

As the second issue of the implicit KT process, how to transfer refers to the process of extracting the truly useful materials from the source tasks that benefit the target task. In implicit EMT, this knowledge extraction process is realized by evolution operators. To this end, researchers have considered characteristics of the specific EC solvers such as DE [9], genetic algorithms (GA) [14], and evolution strategies [16] to design compatible evolution operators for implicit KT. For the implicit EMT algorithms using DE as the base solver, Feng et al. [8] proposed a mutation operator by transferring the differential vector of the source tasks to transfer knowledge. Jin et al. [11] explore the transfer of elite solutions from the source task as the base vector in the mutation process of DE to improve transfer quality. Besides, Chen et al. [5] proposed to use the binomial crossover to exchange dimensions between solutions from two tasks. For the implicit EMT algorithms using GA as the base solver, the simulated binary crossover [3] is a commonly used operator in the literature. Wang et al. [10] developed an implicit KT process that directly transfers the sampled solutions by the GA operator performed on the source population with an anomaly detection strategy. Besides, Zhou et al. [14] proposed to ensemble multiple genetic operators to perform KT, making use of the complementarity of different evolution operators. Due to the distribution discrepancy between source and target tasks, it is found that performing a domain transformation on the source solutions before the execution of the evolution operator of KT would be beneficial [15].

The similarities and differences between the proposed L2T and existing EMT methods are listed in Table I. The listed algorithms differ in the use of base solver, the class of implicit or explicit algorithms, and the designs on when to transfer and how to transfer. The definition of *online learning* is the process to explicitly using machine learning algorithms to estimate the cross-task knowledge or to transform the cross-task information for better utilization. The *self-adaptation* refers to the mechanism that dynamically selects proper knowledge for KT or adjusts the frequency of KT. The *transformation* refers to the process of translating and mapping the source solution from the source task for further processing in a KT action while the *evolution operator* in the KT process is used to generate solutions of the target task that contain cross-task knowledge.

Here, we give a formal definition on the *explicit* EMT as the algorithms that explicitly contain both an *online learning* process during the search process and a *solution transformation process* for further knowledge transfer action. Otherwise,

TABLE I
THE SIMILARITIES AND DIFFERENCES BETWEEN THE PROPOSED L2T AND COMPARED METHODS

Methods	Base solver	Implicit or Explicit	Online learning	Self-adaptation	How to transfer	
					Transformation	Evolution Operator(s)
AEMTO [7]	DE	Implicit	No	Yes	No	Crossover
MFDE [8]	DE	Implicit	No	No	No	Mutation
MKTDE [9]	DE	Implicit	No	No	Yes	Mutation
MTDE-AD [10]	DE	Implicit	No	Yes	No	Direct
MTDE-B [11]	DE	Implicit	No	No	No	Mutation
EMTEA [12]	DE/GA	Explicit	Yes	No	Yes	NA
MFEA [3]	GA	Implicit	No	No	No	Crossover
MFEA-II [4]	GA	Implicit	Yes	Yes	No	Crossover
GMFEA [13]	GA	Implicit	No	No	Yes	Crossover
MTEA-AD [10]	GA	Implicit	No	Yes	No	Direct
MFEA-AKT [14]	GA	Implicit	No	Yes	No	Crossover (6)
ATMFEA [15]	GA	Explicit	Yes	No	Yes	Crossover
L2T (Ours)	DE/GA	Implicit	No	Yes	Yes	Mutation

it will be categorized as *implicit*. For example, MFEA-II belongs to the implicit EMT algorithms because the KT is achieved by the evolution operator without any transformation, despite it contains an online learning process to learn the transfer coefficient rmp. Similarly, although the proposed L2T contains a solution transformation process, L2T does not contain any online learning process and the agent strategy for KT self-adaptation is *learned offline* from the optimization experience data.

S.II. PSEUDOCODE OF IMPLEMENTATION OF LEARNING TO TRANSFER

The rollout process by the agent with base solver Genetic Algorithm (GA) is given in **Algorithm S.1**. The pseudocode of the Multi-Task Differential Evolution with Learning to Transfer (MTDE-L2T) equipped with the learned agent is given in **Algorithm S.2**. The pseudocode of MTGA-L2T equipped with the learned agent is given in **Algorithm S.3**.

S.III. PROBLEM SETUP AND PARAMETER CONFIGURATION

The research objective in experimental study is to evaluate the adaptability of EMT algorithms, which is measured by the average optimization performance over many unseen MTOP instances. Existing benchmarks such as CEC17MTOP [17] with 9 MTOP instances and CEC19MTOP [18] with 10 MTOP instances can not satisfy our experimental requirements due to the very limited size of the MTOP instance set. Therefore, we propose a new test suite to evaluate the adaptability of EMT algorithms.

A. Basic Concepts

Task instance. In this study, an optimization task instance $g(x)$ is in the form of

$$g(x; f, M, x_O) = f(M(x - x_O)) \tag{1}$$

where f denotes a function class represented by mathematical formula, M denotes the rotation matrix, and x_O denotes the shift vector. Here, M rotates the search landscape, introducing variable correlations, while x_O shifts the optimum. Note that we assume the optimization task to be black-box, meaning the structure information of the function like first-order gradient is inaccessible to the algorithms. Herein, we denote the set containing different function classes f as $\mathcal{F}$ and the set containing different parametric configurations $s : (M, x_O)$ as the configuration set $\mathcal{S}$.

Task instance set. The task instance set Θ is used to construct MTOP instances. A task instance set Θ is defined by the product of a function class set $\mathcal{F}$ and a configuration set $\mathcal{S}$, i.e., $\Theta = \mathcal{F} \times \mathcal{S}$. Hence, the task instance set's size is $|\Theta| = |\mathcal{F}| \times |\mathcal{S}|$. There are two ways to configure a task instance set either by (1) parameterization or (2) specification. In the parameterization, the task instance set Θ is configured by parameterizing the task e.g., by (M, x_O) in a continuous space and defining a distribution on this space. In the specification, the task instance set Θ can be constructed by manually specifying a number of task instances. With the definition of Θ, we have explicitly or implicitly defined the distribution of g as a random variable G, i.e.,

$$g(x) \sim p(G; \Theta). \tag{2}$$

MTOP instance. An MTOP instance, denoted as $\mathcal{T}$, for the EMT algorithm to solve is a pair or a set of task instances, which is defined as

$$\mathcal{T} = \{g_k(x)\}, \mathcal{X}_k \subseteq \mathbb{R}^{D_k}, k = 1, 2, \ldots, K, \tag{3}$$

where K is the number of tasks, $\mathcal{X}_k$ and D_k are the solution space and the dimensionality of the solution for task $g_k(x)$, respectively. Without loss of generality, only the box constraints will be studied, i.e., $\mathcal{X}_k = [L_k, U_k]^{D_k}$, where L_k and U_k are the lower and upper boundaries of the solution space respectively. The objective of an MTOP instance is to find the optimal solution for each task, and in the case of minimization can be represented as

$$x_k^* = \arg\min_{x \in \mathcal{X}_k} g_k(x), k = 1, 2, \ldots, K. \tag{4}$$

The solution spaces of tasks may not be identical and have different lower and upper boundaries. To allow KT between different tasks, a common strategy [3] is to perform an linear transformation $(x_k - L_k)/(U_k - L_k)$ on x_k to encode the solutions to a unified search space $\mathcal{U}_k = [0, 1]^{D_{\mathcal{U}}}$, where $D_{\mathcal{U}} = \max\{D_k\}$. An MTOP instance $\mathcal{T}$ with K tasks is constructed

Algorithm S.1: Agent rollout with base solver GA

Input: Task pair $\mathcal{T}$=$\{f_1,\ f_2\}$, base solver $\mathcal{A} = GA$, maximum generations G_{roll}, parameterized agent $\pi(s|\theta)$, initial population set $\mathcal{P} = \{P_1, ..., P_{N_P}\}$, population size per task N
Output: Experience data buffer $\mathcal{D}$

$\mathcal{D} = \varnothing$; `// Empty rollout data buffer`
Initialize population X of size N for each task by randomly selecting initial population from $\mathcal{P}$;
Evaluate fitness of X to obtain Y for each task;
Calculate initial state s by concatenating O_c and O_t;
$g = 0$;
while $g < G_{\max}$ **do**
  $a = \pi(s|\theta)$; `// predict action by actor network`
  $P_m = X_1 \cup X_2$;`// merge populations of two tasks`
  Empty the offspring set U_1, U_2 for two tasks;
  Retrieve KT action parameters $a_{k,1}, a_{k,2}, a_{k,3}$ for task f_k from a;
  while *number of offspring for each task* $< N$ **do**
    Sample two individuals p_a, p_b from P_m as parents;
    $P_m = P_m - \{p_a, p_b\}$;
    Get the associated task indices k_a, k_b of p_a, p_b;
    if $k_a == k_b$ **then**
      `// belong to the same task`
      Perform crossover and mutation on p_a, p_b to obtain two offsprings u_a, u_b for task f_k;
    else if $rand < a_{k_a,1}$ **then**
      `// perform KT between tasks`
      Sample u_a by the proposed action formulation with parameters $a_{k_a,2}, a_{k_a,3}$;
      Perform mutation on p_b to obtain u_b;
    else
      Perform mutation on p_a, p_b to obtain u_a, u_b;
    end
    Add u_a, u_b to their corresponding offspring set U_1, U_2;
  end
  Evaluate fitness of U_1, U_2 on $f_1,\ f_2$ respectively;
  foreach task f_k **do**
    Calculate task-specific features $O_{t,k}$ of task f_k;
    Calculate reward r_k;
    Update population POP_k by selection;
  end
  $r = r_1 + r_2$;`// Sum up rewards of the tasks`
  Calculate common features O_c;
  Update state s by concatenating O_c and O_t;
  $\mathcal{D} = \mathcal{D} \cup (s, a, r)$;
  $g = g + 1$;
end

by selecting or sampling K pairs (f, s) from a task instance set Θ. In this study, we assume all tasks of an MTOP instance are drawn independently from the same distribution, i.e.,

$$p(\mathcal{T}; \Theta) = p(g_1, ..., g_K; \Theta) = \prod_{k=1}^{K} p(G = g_k; \Theta) \qquad (5)$$

MTOP instance set. With the definition of MTOP instance $\mathcal{T}$, we can now define an MTOP instance set $\Gamma = \{\mathcal{T}_j\}_{j=1}^{N_\Gamma}$ containing N_Γ different MTOP instances that are randomly sampled based on the predefined task instance set Θ. Without loss of generality, we assume all MTOP instances are drawn from the same distribution $\mathcal{T}_j \sim p(\mathcal{T}; \Theta), j = 1, ..., N_\Gamma$. Next, we can define multiple task instance sets $\Theta_1, \Theta_2, ...$ by

Algorithm S.2: MTDE-L2T

Input: Task pair $\mathcal{T}$=$\{f_1,\ f_2\}$, maximum generations $G_{\max}$, learned agent $\pi(s|\theta^*)$, initial population set $\mathcal{P} = \{P_1, ..., P_{N_P}\}$, population size per task N
Output: Best-found solutions for two tasks x_1^*, x_2^*

Initialize population X of size N for each task by randomly selecting initial population from $\mathcal{P}$;
Evaluate fitness of X to obtain Y for each task;
Calculate initial state s by concatenating O_c and O_t;
$g = 0$;
while $g < G_{\max}$ **do**
  $a = \pi(s|\theta^*)$;`// predict action by actor network`
  foreach task f_k **do**
    Retrieve KT action parameters $a_{k,1}, a_{k,2}, a_{k,3}$ for task f_k from a;
    Sample offspring population U by base solver DE;
    $N_{KT} = \lceil 0.5 \cdot a_{k,1} \rceil$;
    Randomly select N_{KT} indices from $\{1, ..., N\}$ to construct a index set $\mathcal{I}_{KT} = \{j_1, ..., j_{N_{KT}}\}$;
    foreach index j in $\mathcal{I}_{KT}$ **do**
      Sample $v_{k,j}$ by the proposed action formulation with KT action parameters $a_{k,2}, a_{k,3}$;
      Perform binomial crossover to obtain $u_{k,j}$;
      Replace j-th solution in U with $u_{k,j}$;
    end
    Evaluate fitness Y_k of U on f_k;
    Calculate task-specific features $O_{t,k}$ of task f_k;
    Update population POP_k by selection;
  end
  Calculate common features O_c;
  Update state s by concatenating O_c and O_t;
  Update best-found solutions x_1^*, x_2^*;
  $g = g + 1$;
end

configuring $\mathcal{F}_1, \mathcal{F}_2, ...$ and $\mathcal{S}_1, \mathcal{S}_2, ...$ respectively to construct multiple MTOP instance sets $\Gamma_1, \Gamma_2, \ldots$. Different MTOP instance sets represent different distributions for learning the agent and testing the performance of the EMT algorithms. In the experiment, we train an agent on a specific MTOP set and test the performance on multiple MTOP sets.

B. Problem Settings

To define task instances, we employ two sets of synthetic functions. The first function set $\mathcal{F} = \{Ackley, Griewank, Rastrigin, Sphere, Weierstrass\}$ includes the functions from the CEC17MTOP benchmark [17] with highly configurable global optimum by varying x_O. The functions' solution space is normalized to $\mathcal{X} = [0, 1]^D$ based on the lower bound and upper bound of each function [19]. Based on this function set $\mathcal{F}$, we build different task instance sets Θ by defining different task optimum distributions $p(x_O)$, thereby obtaining multiple MTOP instance sets. The configurations of MTOP instance sets are given in Table S.I. The shift x_O lies in the search space $[0, 1]^D$, C is the number of clusters, $x_{c,i}$ is the center of the i-th cluster, and Δ_i is the radius of the i-th cluster. Specifically, we define 10 MTOP sets with different characteristics in the task optimum distribution range (i.e., VS, S, M, L, and VL) and the number of distribution clusters (i.e., C1-C5). Note

Algorithm S.3: MTGA-L2T

Input: Task pair $\mathcal{T}$={f_1, f_2}, maximum generations $G_{\max}$, learned agent $\pi(s|\theta^*)$, initial population set $\mathcal{P} = \{P_1, ..., P_{N_P}\}$, population size per task N
Output: Best-found solutions for two tasks x_1^*, x_2^*

Initialize population X of size N for each task by randomly selecting initial population from $\mathcal{P}$;
Evaluate fitness of X to obtain Y for each task;
Calculate initial state s by concatenating O_c and O_t;
$g = 0$;
while $g < G_{\max}$ **do**
  $a = \pi(s|\theta^*)$;`// predict action by actor network`
  $P_m = X_1 \cup X_2$;`// merge populations of two tasks`
  Empty the offspring set U_1, U_2 for two tasks;
  Retrieve KT action parameters $a_{k,1}, a_{k,2}, a_{k,3}$ for task f_k from a;
  while *number of offspring for each task* $< N$ **do**
    Sample two individuals p_a, p_b from P_m as parents;
    $P_m = P_m - \{p_a, p_b\}$;
    Get the associated task indices k_a, k_b of p_a, p_b;
    if $k_a == k_b$ **then**
      `// belong to the same task`
      Perform crossover and mutation on p_a, p_b to obtain two offsprings u_a, u_b for task f_k;
    else if $rand < a_{k_a,1}$ **then**
      `// perform KT between tasks`
      Sample u_a by the proposed action formulation with parameters $a_{k_a,2}, a_{k_a,3}$;
      Perform mutation on p_b to obtain u_b;
    else
      Perform mutation on p_a, p_b to obtain u_a, u_b;
    end
    Add u_a, u_b to their corresponding offspring set U_1, U_2;
  end
  Evaluate fitness of U_1, U_2 on f_1, f_2 respectively;
  foreach task f_k **do**
    Calculate task-specific features $O_{t,k}$ of task f_k;
    Calculate reward r_k;
    Update population POP_k by selection;
  end
  Calculate common features O_c;
  Update state s by concatenating O_c and O_t;
  Update best-found solutions x_1^*, x_2^*;
  $g = g + 1$;
end

that since we directly define the distribution of $p(x_O)$ in a continuous space, the cardinality of the configuration set $\mathcal{S}$ and produced task instance set Θ is infinite. For each MTOP set in Table S.I, we learn an agent and test the learned agent on the problem set independently. That is, we obtain 10 learned agents for 10 problem sets, respectively. Therefore, the training MTOP instances and testing MTOP instances are independent and identically distributed (i.i.d.), allowing us to assess the effectiveness of L2T in adapting to different MTOP distributions of interest.

Different from the first function set, the second function set includes a broader spectrum of functions with diverse characteristics from the black-box optimization benchmark (BBOB) [20]. BBOB is a widely used benchmark for evaluating and comparing black-box optimizers. The BBOB contains a total of 24 classes of functions to serve as optimization tasks, denoted as $\{f_1, ..., f_{24}\}$, with different landscape properties which can be categorized into five groups, namely separable functions $\{f_1, ..., f_5\}$, functions with low or moderate conditioning $\{f_6, ..., f_9\}$, functions with high conditioning and unimodal $\{f_{10}, ..., f_{14}\}$, multi-modal functions with adequate global structure $\{f_{15}, ..., f_{19}\}$, and multi-modal functions with weak global structure $\{f_{20}, ..., f_{24}\}$. The function ID is denoted as $fid \in \{1, ..., 24\}$ and (M, x_O) of the task instances in BBOB is generated using a random number generator a with seed ID denoted as sid. Hence, each task instance in BBOB is defined as a tuple (fid, sid). This setup enables BBOB to create numerous task instances by varying the function ID fid and the seed sid. The functions' search space is set to $\mathcal{X} = [-5, 5]^D$. The optimums of most functions are uniformly distributed in a wide range $[-4, 4]^D$, posing challenges to EMT algorithms' adaptability. For clarity, we define the function ID set as $\mathcal{F}$ containing different fid and the seed set as $\mathcal{S}$ containing different sid.

For the BBOB, we formulate 16 MTOP instance sets with their corresponding configurations on $\mathcal{F}$ and $\mathcal{S}$ shown in Table S.II. The 16 MTOP instance sets contain one set for learning (i.e., BBOB$_{learn}$) and 15 sets (i.e., BBOB1-BBOB15) with unseen MTOP instances for testing. That is, we obtain one learned agent and test it on the remaining 15 MTOP sets. Regarding the MTOP set for learning, we want to cover different kinds of complex functions for the agent to learn versatile KT skills for handling diverse problems. Therefore, we select functions $\{f_1, f_3, f_8, f_{10}, f_{16}, f_{20}\}$ from each of the five function groups in BBOB. We categorize the test MTOP instance sets into two groups to assess adaptability: one for evaluating nearly i.i.d. scenarios and another for potential out-of-distribution (o.o.d.) cases. The nearly i.i.d. group contains MTOP instances with the same functions as the learning phase but varies in real distribution coverage (BBOB1 and BBOB2) and function-type weighting (BBOB3-BBOB8), aligning closely with the training set BBOB$_{learn}$. Conversely, the o.o.d. group comprises entirely new functions not encountered during learning (BBOB9-BBOB15), with BBOB9 presenting a substantial challenge by including all functions omitted in the learning stage. To evaluate adaptability to unseen yet similar functions, we create six MTOP sets (BBOB10-BBOB15) using selected functions $f_2, f_6, f_{12}, f_{15}, f_{21}$ from BBOB's five function groups. This problem setting allows us to evaluate the generalization ability of the learned agent. A notable difference between BBOB-based MTOP sets and the ones based on CEC19MTOP is that the number of training instances in BBOB-based MTOP sets is limited, which is a more practical situation faced in the real world.

C. Parameter Configuration

The parameters of the proposed L2T framework are given in Table S.I. The number of tasks of the MTOP instance is set to $K = 2$. For the learning stage, the maximum number of rollout generations is set as $G_{\text{roll}} = 100$ and the target accuracy is $\xi = 1e-8$. For the testing stage, we set the maximum number of generations as $G_{\max} = 250$ which is larger than the rollout generations $G_{\mathrm{roll}} = 100$. The parameters β_1, β_2, and β_3 in the reward function are set to $\beta_1 = 1, \beta_2 = 10$ and $\beta_3 = G_{\mathrm{roll}} = 100$, respectively. Each problem set for the testing stage contains $N_\Gamma = 100$ randomly sampled MTOP instances.

REFERENCES

[1] K. C. Tan, L. Feng, and M. Jiang, "Evolutionary transfer optimization-a new frontier in evolutionary computation research," *IEEE Computational Intelligence Magazine*, vol. 16, no. 1, pp. 22–33, 2021.

[2] T. Wei, S. Wang, J. Zhong, D. Liu, and J. Zhang, "A review on evolutionary multitask optimization: Trends and challenges," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 5, pp. 941–960, 2021.

[3] A. Gupta, Y.-S. Ong, and L. Feng, "Multifactorial evolution: toward evolutionary multitasking," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 3, pp. 343–357, 2015.

[4] K. K. Bali, Y.-S. Ong, A. Gupta, and P. S. Tan, "Multifactorial evolutionary algorithm with online transfer parameter estimation: Mfea-ii," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 1, pp. 69–83, 2019.

[5] Y. Chen, J. Zhong, L. Feng, and J. Zhang, "An adaptive archive-based evolutionary framework for many-task optimization," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 3, pp. 369–384, 2019.

[6] Z. Liang, X. Xu, L. Liu, Y. Tu, and Z. Zhu, "Evolutionary many-task optimization based on multisource knowledge transfer," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 2, pp. 319–333, 2021.

[7] H. Xu, A. K. Qin, and S. Xia, "Evolutionary multitask optimization with adaptive knowledge transfer," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 2, pp. 290–303, 2021.

[8] L. Feng, W. Zhou, L. Zhou, S. Jiang, J. Zhong, B. Da, Z. Zhu, and Y. Wang, "An empirical study of multifactorial pso and multifactorial de," in *Proc. IEEE Congress on evolutionary computation (CEC)*, 2017, pp. 921–928.

[9] J.-Y. Li, Z.-H. Zhan, K. C. Tan, and J. Zhang, "A meta-knowledge transfer-based differential evolution for multitask optimization," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 4, pp. 719–734, 2021.

[10] C. Wang, J. Liu, K. Wu, and Z. Wu, "Solving multitask optimization problems with adaptive knowledge transfer via anomaly detection," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 2, pp. 304–318, 2021.

[11] C. Jin, P.-W. Tsai, and A. K. Qin, "A study on knowledge reuse strategies in multitasking differential evolution," in *Proc. IEEE Congress on Evolutionary Computation (CEC)*, 2019, pp. 1564–1571.

[12] L. Feng, L. Zhou, J. Zhong, A. Gupta, Y.-S. Ong, K.-C. Tan, and A. K. Qin, "Evolutionary multitasking via explicit autoencoding," *IEEE Transactions on Cybernetics*, vol. 49, no. 9, pp. 3457–3470, 2018.

[13] J. Ding, C. Yang, Y. Jin, and T. Chai, "Generalized multitasking for evolutionary optimization of expensive problems," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 1, pp. 44–58, 2017.

[14] L. Zhou, L. Feng, K. C. Tan, J. Zhong, Z. Zhu, K. Liu, and C. Chen, "Toward adaptive knowledge transfer in multifactorial evolutionary computation," *IEEE Transactions on Cybernetics*, vol. 51, no. 5, pp. 2563–2576, 2020.

[15] X. Xue, K. Zhang, K. C. Tan, L. Feng, J. Wang, G. Chen, X. Zhao, L. Zhang, and J. Yao, "Affine transformation-enhanced multifactorial optimization for heterogeneous problems," *IEEE Transactions on Cybernetics*, vol. 52, no. 7, pp. 6217–6231, 2020.

[16] Y. Li, W. Gong, and S. Li, "Multitask evolution strategy with knowledge-guided external sampling," *IEEE Transactions on Evolutionary Computation*, 2023.

[17] B. Da, Y.-S. Ong, L. Feng, A. K. Qin, A. Gupta, Z. Zhu, C.-K. Ting, K. Tang, and X. Yao, "Evolutionary multitasking for single-objective continuous optimization: Benchmark problems, performance metric, and baseline results," *arXiv preprint arXiv:1706.03470*, 2017.

[18] L. Feng, K. Qin, A. Gupta, Y. Yuan, Y. Ong, and X. Chi, "Ieee cec 2019 competition on evolutionary multi-task optimization," *MTO_Competition_CEC_2019.html*, 2019.

[19] S.-H. Wu, Z.-H. Zhan, K. C. Tan, and J. Zhang, "Orthogonal transfer for multitask optimization," *IEEE Transactions on Evolutionary Computation*, vol. 27, no. 1, pp. 185–200, 2022.

[20] N. Hansen, A. Auger, R. Ros, O. Mersmann, T. Tušar, and D. Brockhoff, "Coco: A platform for comparing continuous optimizers in a black-box setting," *Optimization Methods and Software*, vol. 36, no. 1, pp. 114–144, 2021.

TABLE S.I

CONFIGURATIONS OF MTOP INSTANCE SET WITH DIFFERENT TASK OPTIMUM DISTRIBUTION BASED ON CEC17MTOP

MTOP set	Task optimum distribution configuration ($\mathcal{S}$)
VS	$x_o \sim U[0.5-\Delta, 0.5+\Delta]^D, \Delta=0.025$
S	$x_o \sim U[0.5-\Delta, 0.5+\Delta]^D, \Delta=0.05$
M	$x_o \sim U[0.5-\Delta, 0.5+\Delta]^D, \Delta=0.1$
L	$x_o \sim U[0.5-\Delta, 0.5+\Delta]^D, \Delta=0.2$
VL	$x_o \sim U[0.5-\Delta, 0.5+\Delta]^D, \Delta=0.4$
C1	$x_o \sim x_{c,1}+L(x_{c,2}-x_{c,1}), L \sim U[0,1], x_{c,1}, x_{c,2} \in [0,1]^D$
C2	$x_o \sim U[x_{c,i}-\Delta_i, x_{c,i}+\Delta_i \vert Z=z_i]^D, Z \sim p(Z)=1/C, C=2$
C3	$x_o \sim U[x_{c,i}-\Delta_i, x_{c,i}+\Delta_i \vert Z=z_i]^D, Z \sim p(Z)=1/C, C=3$
C4	$x_o \sim U[x_{c,i}-\Delta_i, x_{c,i}+\Delta_i \vert Z=z_i]^D, Z \sim p(Z)=1/C, C=4$
C5	$x_o \sim U[x_{c,i}-\Delta_i, x_{c,i}+\Delta_i \vert Z=z_i]^D, Z \sim p(Z)=1/C, C=5$

TABLE S.II

CONFIGURATIONS OF MTOP INSTANCE SET BASED ON BBOB

Use purpose	MTOP set	Task function ID set ($\mathcal{F}$)	Task seed set ($\mathcal{S}$)
For learning	BBOB$_{learn}$	$\{1,3,8,10,16,20\}$	[1,100]
For testing nearly i.i.d. adaptability	BBOB1	$\{1,3,8,10,16,20\}$	[500,1500]
	BBOB2	$\{1,3,8,10,16,20\}$	[1000,1005]
	BBOB3	$\{1\}$	[500,1500]
	BBOB4	$\{3\}$	[500,1500]
	BBOB5	$\{8\}$	[500,1500]
	BBOB6	$\{10\}$	[500,1500]
	BBOB7	$\{16\}$	[500,1500]
	BBOB8	$\{20\}$	[500,1500]
For testing o.o.d. adaptability	BBOB9	$\{1,...,24\}-\{1,3,8,10,16,20\}$	[500,1500]
	BBOB10	$\{2,6,12,15,21\}$	[500,1500]
	BBOB11	$\{2\}$	[500,1500]
	BBOB12	$\{6\}$	[500,1500]
	BBOB13	$\{12\}$	[500,1500]
	BBOB14	$\{15\}$	[500,1500]
	BBOB15	$\{21\}$	[500,1500]

TABLE S.III

PARAMETER CONFIGURATION OF THE PROPOSED L2T FRAMEWORK IN THE EXPERIMENTAL STUDIES

Parameter	Value
Maximum rollout generations G_{roll}	$G_{\mathrm{roll}}=100$
Maximum generations for testing $G_{\max}$	$G_{\max}=250$
Target accuracy ξ	$\xi=1e-8$
Actor-critic network structure	Two hidden layers and one linear layer, # hidden neurons=64, activation function is $\tanh(\cdot)$
Population size N	$N=50$
Evolution operators for DE	DE/rand/1 mutation and binomial crossover
DE-related parameters	$F=0.5, CR=0.5$
Evolution operators for GA	Simulated binary crossover (SBX) and polynomial mutation (PM)
GA-related parameters	$\eta_c=2, \eta_m=5$
Reward function coefficients $\beta_1, \beta_2, \beta_3$	$\beta_1=1, \beta_2=10, \beta_3=G_{\mathrm{roll}}$
PPO-related parameters	$\gamma=0.99, \lambda=0.95, \epsilon=0.2$
Initial population set size N_P	$N_P=10$
Number of parallel environments N_{env}	$N_{env}=20$
Rollout data buffer size N_{buff}	$N_{buff}=2048*N_{env}=40960$
Maximum number of time steps for learning T	$T=5e6$ for BBOB$_{learn}$ and $T=2e6$ for VS, S, M, L, VL, C1, C2, C3, C4, and C5

TABLE S.IV

COMPARATIVE RESULTS BETWEEN MTDE-L2T AND OTHER IMPLICIT EMT ALGORITHMS WITH AGENTS LEARNED SEPARATELY ON DIFFERENT PROBLEM SETS AT GENERATION=G_{roll}

Problem	**AEMTO**	**MFDE**	**MKTDE**	**MTDE-AD**	**MTDE-B**
VS	100/0/0(+)	100/0/0(+)	86/7/7(+)	99/1/0(+)	97/3/0(+)
S	94/6/0(+)	88/8/4(+)	66/15/19(+)	88/11/1(+)	93/7/0(+)
M	87/13/0(+)	72/15/13(+)	50/17/33(+)	84/14/2(+)	82/16/2(+)
L	79/19/2(+)	51/29/20(+)	46/24/30(+)	71/26/3(+)	46/49/5(+)
VL	50/48/2(+)	31/47/22(+)	35/34/31(+)	45/54/1(+)	2/92/6(−)
C1	74/26/0(+)	61/29/10(+)	44/30/26(+)	63/36/1(+)	40/58/2(+)
C2	47/48/5(+)	51/33/16(+)	32/26/42(−)	34/59/7(+)	16/75/9(+)
C3	35/56/9(+)	37/42/21(+)	34/30/36(−)	34/51/15(+)	9/87/4(+)
C4	42/44/14(+)	43/39/18(+)	46/21/33(+)	35/51/14(+)	9/83/8(+)
C5	45/48/7(+)	37/46/17(+)	45/27/28(+)	44/46/10(+)	7/83/10(−)
BBOB9	29/69/2(+)	44/40/16(+)	55/26/19(+)	36/61/3(+)	3/88/9(−)
BBOB10	55/45/0(+)	50/34/16(+)	53/29/18(+)	61/36/3(+)	3/87/10(−)

TABLE S.V

Comparative Results between the Proposed L2T-based and Other Implicit EMT Algorithms at Generation=$G_{\max}$

Problem	MTDE-L2T vs					MTGA-L2T vs				
	AEMTO	MFDE	MKTDE	MTDE-AD	MTDE-B	GMFEA	MFEA	MFEA-AKT	MFEA2	MTEA-AD
BBOB1	35/55/10(+)	44/52/4(+)	59/26/15(+)	40/50/10(+)	29/61/10(+)	66/26/8(+)	64/30/6(+)	66/27/7(+)	62/35/3(+)	54/23/23(+)
BBOB2	41/54/5(+)	44/50/6(+)	66/28/6(+)	40/53/7(+)	26/60/14(+)	63/30/7(+)	65/30/5(+)	65/31/4(+)	67/31/2(+)	63/14/23(+)
BBOB3	0/100/0(=)	0/100/0(=)	0/100/0(=)	7/93/0(+)	0/100/0(=)	100/0/0(+)	100/0/0(+)	100/0/0(+)	100/0/0(+)	100/0/0(+)
BBOB4	67/32/1(+)	10/85/5(+)	100/0/0(+)	57/43/0(+)	37/62/1(+)	100/0/0(+)	100/0/0(+)	100/0/0(+)	100/0/0(+)	100/0/0(+)
BBOB5	1/88/11(−)	3/80/17(−)	3/79/18(−)	2/92/6(−)	4/88/8(−)	51/44/5(+)	62/33/5(+)	58/36/6(+)	39/59/2(+)	46/49/5(+)
BBOB6	30/70/0(+)	38/62/0(+)	81/19/0(+)	30/69/1(+)	1/81/18(−)	8/71/21(−)	7/72/21(−)	9/77/14(−)	28/68/4(+)	6/64/30(−)
BBOB7	4/93/3(+)	3/92/5(−)	2/95/3(−)	2/92/6(−)	4/89/7(−)	11/82/7(+)	13/78/9(+)	14/83/3(+)	23/74/3(+)	0/29/71(−)
BBOB8	8/86/6(+)	3/93/4(−)	73/27/0(+)	10/87/3(+)	6/88/6(=)	29/70/1(+)	29/71/0(+)	16/80/4(+)	16/81/3(+)	58/42/0(+)
BBOB9	41/53/6(+)	36/45/19(+)	54/28/18(+)	41/53/6(+)	26/59/15(+)	51/37/12(+)	59/32/9(+)	56/29/15(+)	65/25/10(+)	54/20/26(+)
BBOB10	60/32/8(+)	36/41/23(+)	40/30/30(+)	68/25/7(+)	36/41/23(+)	77/13/10(+)	80/16/4(+)	79/16/5(+)	80/14/6(+)	58/13/29(+)
BBOB11	96/0/4(+)	12/88/0(+)	2/93/5(−)	96/0/4(+)	9/91/0(+)	99/0/1(+)	99/0/1(+)	99/0/1(+)	99/0/1(+)	97/0/3(+)
BBOB12	33/61/6(+)	0/4/96(−)	0/9/91(−)	40/58/2(+)	30/65/5(+)	97/0/3(+)	99/0/1(+)	100/0/0(+)	100/0/0(+)	98/1/1(+)
BBOB13	6/80/14(−)	10/86/4(+)	15/81/4(+)	6/90/4(+)	11/87/2(+)	97/0/3(+)	96/0/4(+)	96/0/4(+)	96/0/4(+)	90/0/10(+)
BBOB14	11/88/1(+)	9/88/3(+)	8/87/5(+)	3/93/4(−)	6/90/4(+)	5/83/12(−)	3/87/10(−)	8/87/5(+)	18/82/0(+)	0/31/69(−)
BBOB15	7/88/5(+)	12/81/7(+)	19/78/3(+)	14/82/4(+)	3/85/12(−)	26/49/25(+)	25/51/24(+)	21/55/24(−)	25/55/20(+)	22/54/24(−)

TABLE S.VI

Comparative Results between the Proposed L2T-based and Explicit EMT Algorithms at Different Generation G

Problem	MTDE-L2T vs MTDE-EA		MTGA-L2T vs ATMFEA	
	$G = G_{\text{roll}}$	$G = G_{\max}$	$G = G_{\text{roll}}$	$G = G_{\max}$
BBOB1	61/39/0(+)	39/52/9(+)	77/15/8(+)	59/31/10(+)
BBOB2	62/37/1(+)	40/53/7(+)	76/18/6(+)	65/30/5(+)
BBOB3	98/2/0(+)	0/100/0(=)	100/0/0(+)	100/0/0(+)
BBOB4	22/77/1(+)	61/39/0(+)	100/0/0(+)	100/0/0(+)
BBOB5	12/84/4(+)	4/91/5(−)	74/15/11(+)	58/32/10(+)
BBOB6	16/80/4(+)	30/68/2(+)	8/63/29(−)	6/70/24(−)
BBOB7	5/94/1(+)	1/89/10(−)	29/69/2(+)	16/78/6(+)
BBOB8	34/66/0(+)	6/90/4(+)	100/0/0(+)	28/72/0(+)
BBOB9	33/61/6(+)	41/49/10(+)	51/30/19(+)	50/22/28(+)
BBOB10	42/41/17(+)	50/38/12(+)	70/22/8(+)	73/14/13(+)
BBOB11	94/1/5(+)	96/1/3(+)	100/0/0(+)	99/0/1(+)
BBOB12	37/62/1(+)	36/61/3(+)	57/37/6(+)	99/0/1(+)
BBOB13	59/38/3(+)	10/82/8(+)	98/0/2(+)	94/0/6(+)
BBOB14	7/93/0(+)	8/87/5(+)	5/85/10(−)	2/55/43(−)
BBOB15	16/78/6(+)	12/77/11(+)	21/55/24(−)	22/51/27(−)

TABLE S.VII

Comparative Results of Trained-from-scratch and Fine-tuned MTDE-L2T and Other Implicit EMT Algorithms at Generation=$G_{\max}$

Problem	MTDE-L2T-w/o-FT vs					
	AEMTO	MFDE	MKTDE	MTDE-AD	MTDE-B	MTDE-EA
VS	97/2/1(+)	77/3/20(+)	56/13/31(+)	93/5/2(+)	92/7/1(+)	96/3/1(+)
S	91/9/0(+)	70/8/22(+)	53/11/36(+)	85/13/2(+)	87/12/1(+)	88/10/2(+)
M	84/15/1(+)	67/9/24(+)	47/15/38(+)	84/13/3(+)	77/21/2(+)	88/11/1(+)
L	79/15/6(+)	59/14/27(+)	49/12/39(+)	79/16/5(+)	45/50/5(+)	82/15/3(+)
VL	49/45/6(+)	27/39/34(−)	28/29/43(−)	43/43/14(+)	3/93/4(−)	48/47/5(+)
C1	69/21/10(+)	53/13/34(+)	35/16/49(−)	68/26/6(+)	37/60/3(+)	69/29/2(+)
C2	44/46/10(+)	41/20/39(+)	26/21/53(−)	45/44/11(+)	22/76/2(+)	56/38/6(+)
C3	41/50/9(+)	28/31/41(−)	31/22/47(−)	42/48/10(+)	10/87/3(+)	46/48/6(+)
C4	41/43/16(+)	35/27/38(−)	37/13/50(−)	38/39/23(+)	7/83/10(−)	52/43/5(+)
C5	43/48/9(+)	31/34/35(−)	34/22/44(−)	46/37/17(+)	6/86/8(−)	53/44/3(+)
BBOB9	21/74/5(+)	36/31/33(+)	49/27/24(+)	29/63/8(+)	4/88/8(−)	16/80/4(+)
BBOB10	53/46/1(+)	34/23/43(−)	33/25/42(−)	48/51/1(+)	10/83/7(+)	52/45/3(+)
Problem	**MTDE-L2T-FT vs**					
	AEMTO	**MFDE**	**MKTDE**	**MTDE-AD**	**MTDE-B**	**MTDE-EA**
VS	87/8/5(+)	82/14/4(+)	41/21/38(+)	91/6/3(+)	78/18/4(+)	91/7/2(+)
S	88/10/2(+)	80/17/3(+)	45/21/34(+)	88/10/2(+)	72/24/4(+)	91/8/1(+)
M	87/9/4(+)	81/15/4(+)	52/17/31(+)	89/9/2(+)	76/23/1(+)	88/9/3(+)
L	89/9/2(+)	75/20/5(+)	52/22/26(+)	87/7/6(+)	69/26/5(+)	89/8/3(+)
VL	72/13/15(+)	55/32/13(+)	51/25/24(+)	67/19/14(+)	47/42/11(+)	80/14/6(+)
C1	68/15/17(+)	62/26/12(+)	44/25/31(+)	69/17/14(+)	56/35/9(+)	83/10/7(+)
C2	53/24/23(+)	49/42/9(+)	28/31/41(−)	49/27/24(+)	34/50/16(+)	61/36/3(+)
C3	55/28/17(+)	39/44/17(+)	43/27/30(+)	61/24/15(+)	35/51/14(+)	69/26/5(+)
C4	65/12/23(+)	50/35/15(+)	38/22/40(−)	59/16/25(+)	50/37/13(+)	75/13/12(+)
C5	63/24/13(+)	48/42/10(+)	53/24/23(+)	65/21/14(+)	40/48/12(+)	65/27/8(+)
BBOB9	40/49/11(+)	33/45/22(+)	54/29/17(+)	40/51/9(+)	29/63/8(+)	39/53/8(+)
BBOB10	67/24/9(+)	35/41/24(+)	36/29/35(+)	63/26/11(+)	39/58/3(+)	66/26/8(+)

TABLE S.VIII
COMPARATIVE RESULTS BETWEEN THE LEARNED AGENT AND PREDEFINED AGENTS AT GENERATION=$G_{\max}$

Problem	MTDE-f(.5,0,1)	MTDE-f(.5,1,0)	MTDE-f(.5,1,1)	MTDE-f(1,0,1)	MTDE-f(1,1,0)	MTDE-f(1,1,1)	MTDE-r	STDE
VS	76/6/18(+)	97/1/2(+)	99/0/1(+)	74/5/21(+)	100/0/0(+)	98/0/2(+)	92/6/2(+)	95/5/0(+)
S	69/9/22(+)	98/2/0(+)	97/3/0(+)	68/8/24(+)	100/0/0(+)	100/0/0(+)	91/8/1(+)	88/11/1(+)
M	67/9/24(+)	92/7/1(+)	90/9/1(+)	67/9/24(+)	100/0/0(+)	100/0/0(+)	90/8/2(+)	72/25/3(+)
L	57/16/27(+)	84/10/6(+)	80/11/9(+)	57/15/28(+)	91/1/8(+)	92/0/8(+)	75/13/12(+)	44/52/4(+)
VL	28/38/34(−)	57/26/17(+)	52/28/20(+)	26/37/37(−)	81/4/15(+)	78/4/18(+)	43/23/34(+)	7/91/2(+)
C1	51/14/35(+)	75/16/9(+)	80/11/9(+)	51/14/35(+)	93/2/5(+)	89/1/10(+)	66/19/15(+)	46/46/8(+)
C2	32/26/42(−)	52/31/17(+)	51/29/20(+)	37/20/43(−)	70/13/17(+)	66/14/20(+)	53/29/18(+)	24/70/6(+)
C3	31/30/39(−)	63/24/13(+)	62/29/9(+)	35/23/42(−)	84/7/9(+)	83/7/10(+)	58/31/11(+)	19/78/3(+)
C4	36/29/35(+)	55/25/20(+)	58/24/18(+)	35/20/45(−)	78/2/20(+)	74/6/20(+)	44/29/27(+)	11/84/5(+)
C5	26/35/39(−)	60/25/15(+)	52/30/18(+)	31/26/43(−)	80/8/12(+)	79/9/12(+)	51/32/17(+)	5/93/2(+)
BBOB1	37/57/6(+)	80/19/1(+)	74/24/2(+)	54/42/4(+)	92/7/1(+)	86/11/3(+)	76/21/3(+)	31/58/11(+)
BBOB2	46/51/3(+)	80/16/4(+)	77/21/2(+)	50/42/8(+)	92/6/2(+)	87/11/2(+)	68/27/5(+)	42/52/6(+)
BBOB9	36/49/15(+)	63/25/12(+)	63/25/12(+)	47/35/18(+)	72/21/7(+)	72/18/10(+)	51/29/20(+)	29/59/12(+)
BBOB10	35/42/23(+)	74/18/8(+)	72/18/10(+)	37/36/27(+)	72/15/13(+)	73/16/11(+)	62/18/20(+)	36/34/30(+)

TABLE S.IX
COMPARATIVE RESULTS OF THE ABLATION STUDY AT GENERATION=$G_{\max}$

Problem	L2T-w/o-a_1	L2T-w/o-a_2	L2T-w/o-a_3	L2T-w/o-O_c	L2T-w/o-O_t	L2T-w/o-FE	L2T-w/o-r_{conv}	L2T-w/o-r_{KT}
VS	67/26/7(+)	90/7/3(+)	76/22/2(+)	23/50/27(−)	51/47/2(+)	95/4/1(+)	14/58/28(−)	90/7/3(+)
S	47/28/25(+)	86/12/2(+)	76/22/2(+)	26/71/3(+)	39/52/9(+)	85/14/1(+)	19/76/5(+)	84/16/0(+)
M	24/64/12(+)	74/23/3(+)	62/35/3(+)	12/86/2(+)	78/19/3(+)	76/21/3(+)	8/79/13(−)	59/36/5(+)
L	27/59/14(+)	47/47/6(+)	52/43/5(+)	19/74/7(+)	48/48/4(+)	46/49/5(+)	29/59/12(+)	48/37/15(+)
VL	2/83/15(−)	5/88/7(−)	8/89/3(+)	5/89/6(−)	6/88/6(=)	4/91/5(−)	5/92/3(+)	11/84/5(+)
C1	27/64/9(+)	38/55/7(+)	43/52/5(+)	41/55/4(+)	50/48/2(+)	32/59/9(+)	33/63/4(+)	24/63/13(+)
C2	21/71/8(+)	28/66/6(+)	23/70/7(+)	20/78/2(+)	22/72/6(+)	20/72/8(+)	19/69/12(+)	35/54/11(+)
C3	14/72/14(=)	15/77/8(+)	15/81/4(+)	15/82/3(+)	15/84/1(+)	14/79/7(+)	16/81/3(+)	23/70/7(+)
C4	8/88/4(+)	9/87/4(+)	10/88/2(+)	7/91/2(+)	10/87/3(+)	6/93/1(+)	7/90/3(+)	17/79/4(+)
C5	3/90/7(−)	6/88/6(=)	4/92/4(=)	6/86/8(−)	6/86/8(−)	10/86/4(+)	4/92/4(=)	10/84/6(+)
BBOB1	17/78/5(+)	31/59/10(+)	29/58/13(+)	10/87/3(+)	35/51/14(+)	30/60/10(+)	8/87/5(+)	33/57/10(+)
BBOB2	17/74/9(+)	31/58/11(+)	29/65/6(+)	6/91/3(+)	36/55/9(+)	40/52/8(+)	13/85/2(+)	30/58/12(+)
BBOB9	23/72/5(+)	23/60/17(+)	31/61/8(+)	17/75/8(+)	26/63/11(+)	31/55/14(+)	9/79/12(−)	26/63/11(+)
BBOB10	11/68/21(−)	36/38/26(+)	55/36/9(+)	11/65/24(−)	38/39/23(+)	38/36/26(+)	13/68/19(−)	38/40/22(+)

TABLE S.X
INVESTIGATION RESULTS ON PARAMETER b_2 AT GENERATION=G_{roll}

Problem	$b_2 = 0.1$	$b_2 = 0.5$	$b_2 = 1$	$b_2 = 5$	$b_2 = 10$	$b_2 = 50$	$b_2 = 100$
BBOB1	6.19	6.25	6.02	**4.17**	4.32	4.62	4.43
BBOB2	6.16	5.98	5.45	**3.93**	4.39	4.53	4.94
BBOB3	5.90	5.49	5.00	5.82	**3.92**	4.06	5.30
BBOB4	**4.39**	5.48	5.44	4.93	5.44	4.91	4.65
BBOB5	**4.47**	4.81	5.50	5.31	5.02	4.92	4.92
BBOB6	5.21	5.70	**4.81**	5.05	5.38	5.20	5.15
BBOB7	5.00	4.86	5.22	4.86	5.24	5.12	**4.56**
BBOB8	4.94	5.32	5.11	5.05	5.06	**4.59**	5.78
BBOB9	5.50	5.34	5.24	5.06	**4.55**	4.81	5.20
BBOB10	5.19	5.71	5.26	4.86	**4.26**	5.02	4.67
BBOB11	6.02	5.62	5.69	4.78	4.40	**3.67**	5.01
BBOB12	6.14	6.67	5.60	5.32	**3.07**	4.79	5.03
BBOB13	5.09	5.56	5.85	5.48	**4.00**	4.46	5.31
BBOB14	4.99	**4.90**	5.27	4.96	5.37	5.08	5.18
BBOB15	4.92	5.15	5.07	5.72	4.95	4.98	**4.90**
# best	2	1	1	2	5	2	2

TABLE S.XI
INVESTIGATION RESULTS ON PARAMETER b_2 AT GENERATION=$G_{\max}$

Problem	$b_2 = 0.1$	$b_2 = 0.5$	$b_2 = 1$	$b_2 = 5$	$b_2 = 10$	$b_2 = 50$	$b_2 = 100$
BBOB1	5.59	5.71	5.98	4.67	4.39	4.50	**4.32**
BBOB2	6.16	5.72	5.54	4.36	4.19	**4.02**	5.05
BBOB3	**2.98**	3.98	5.02	6.00	6.96	7.96	8.96
BBOB4	6.77	7.00	7.07	**3.32**	3.43	3.59	3.58
BBOB5	4.97	4.82	5.47	**4.66**	5.27	4.78	4.94
BBOB6	5.55	6.13	4.95	**4.45**	5.25	4.88	5.14
BBOB7	5.09	4.87	5.27	4.77	5.10	5.60	**4.55**
BBOB8	5.54	5.24	5.10	5.31	4.76	4.54	**4.31**
BBOB9	5.66	5.94	5.29	4.61	**4.27**	4.38	4.74
BBOB10	6.16	6.23	6.26	**4.04**	4.22	4.08	4.41
BBOB11	5.82	5.66	5.57	4.44	4.78	**3.28**	4.87
BBOB12	6.62	7.31	6.35	5.79	**2.72**	4.16	3.70
BBOB13	5.13	5.06	4.95	**4.87**	5.19	5.04	5.39
BBOB14	4.72	4.91	5.36	5.06	5.34	**4.71**	5.11
BBOB15	**4.85**	5.17	4.94	5.72	5.04	5.00	4.94
# best	2	0	0	5	2	3	3

TABLE S.XII
THE RESULTS OF MTDE-L2T WITH AGENT TRAINED BY DIFFERENT RL ALGROTIHMS

Problem	PPO vs SAC		PPO vs TD3	
	$G = G_{\text{roll}}$	$G = G_{\max}$	$G = G_{\text{roll}}$	$G = G_{\max}$
BBOB1	63/37/0(+)	49/42/9(+)	23/65/12(+)	12/74/14(−)
BBOB2	56/43/1(+)	62/20/18(+)	23/63/14(+)	10/76/14(−)
BBOB3	80/20/0(+)	9/91/0(+)	0/0/100(−)	0/100/0(=)
BBOB4	34/65/1(+)	100/0/0(+)	30/67/3(+)	15/84/1(+)
BBOB5	7/88/5(+)	2/86/12(−)	3/86/11(−)	4/86/10(−)
BBOB6	44/55/1(+)	58/42/0(+)	2/59/39(−)	1/61/38(−)
BBOB7	6/83/11(−)	8/86/6(+)	7/88/5(+)	4/89/7(−)
BBOB8	40/60/0(+)	49/51/0(+)	4/84/12(−)	7/92/1(+)
BBOB9	36/57/7(+)	37/46/17(+)	17/57/26(−)	24/52/24(=)
BBOB10	39/33/28(+)	52/24/24(+)	34/35/31(+)	42/33/25(+)
BBOB11	95/1/4(+)	96/0/4(+)	6/47/47(−)	1/48/51(−)
BBOB12	31/67/2(+)	21/49/30(−)	7/51/42(−)	10/46/44(−)
BBOB13	52/47/1(+)	7/80/13(−)	1/11/88(−)	11/69/20(−)
BBOB14	9/87/4(+)	35/64/1(+)	2/87/11(−)	1/93/6(−)
BBOB15	12/75/13(−)	10/72/18(−)	5/67/28(−)	4/79/17(−)
# W/T/L	13/0/2	11/0/4	5/0/10	3/2/10

TABLE S.XIII
COMPARATIVE RESULTS BETWEEN THE PROPOSED MTJADE-L2T AND SINGLE-TASK STJADE AT DIFFERENT GENERATION G

Problem	MTJADE-L2T vs STJADE	
	$G = G_{\text{roll}}$	$G = G_{\max}$
VS	57/40/3(+)	60/38/2(+)
S	57/41/2(+)	45/50/5(+)
M	50/50/0(+)	37/59/4(+)
L	5/90/5(=)	4/90/6(−)
VL	2/91/7(−)	2/93/5(−)
C1	17/79/4(+)	22/75/3(+)
C2	7/90/3(+)	14/83/3(+)
C3	6/89/5(+)	2/93/5(−)
C4	6/90/4(+)	6/90/4(+)
C5	4/87/9(−)	6/89/5(+)
BBOB9	4/92/4(=)	8/88/4(+)
BBOB10	4/90/6(−)	5/88/7(−)

TABLE S.XIV
THE RESULTS OF MTDE-L2T AND MTGA-L2T COMPARING WITH PEER EMT ALGORITHMS ON HPO PROBLEMS

Problem	MTDE-L2T vs					
	AEMTO	MFDE	MKTDE	MTDE-AD	MTDE-B	MTDE-EA
SVM	16/79/5(+)	11/77/12(−)	11/68/21(−)	12/85/3(+)	8/85/7(+)	44/50/6(+)
XGBoost	8/89/3(+)	9/90/1(+)	72/18/10(+)	8/91/1(+)	4/93/3(+)	17/78/5(+)
FCNet	10/88/2(+)	8/91/1(+)	33/53/14(+)	6/93/1(+)	1/94/5(−)	7/91/2(+)
Problem	**MTGA-L2T vs**					
	ATMFEA	**GMFEA**	**MFEA**	**MFEA-AKT**	**MFEA2**	**MTEA-AD**
SVM	56/19/25(+)	52/25/23(+)	52/20/28(+)	69/12/19(+)	68/12/20(+)	46/38/16(+)
XGBoost	67/17/16(+)	69/14/17(+)	69/14/17(+)	72/14/14(+)	81/9/10(+)	70/17/13(+)
FCNet	15/61/24(−)	18/73/9(+)	18/71/11(+)	22/66/12(+)	26/68/6(+)	7/78/15(−)