ADAPTIVE MANIFOLD-DRIVEN KNOWLEDGE TRANSFER VIA REINFORCEMENT LEARNING FOR MULTI-OBJECTIVE MULTI-TASK OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-objective multi-task optimization (MO-MTO) improves search performance by reusing cross-task manifold-structured knowledge. Existing manifold-driven MO-MTO algorithms adopt a fixed knowledge-transfer probability (KTP), which fails to adapt local transfer frequency according to runtime feedback including population evolution, manifold validity, and offspring survival outcomes. To address this limitation, we propose RL-EMT-MSKT, which formulates task-specific KTP adjustment as a Markov decision process and learns multiplicative KTP updates from runtime observations using proximal policy optimization. Specifically, we design a Cross-Task Relation Encoder with scaled dot-product attention to aggregate pairwise task states, and employ Proximal Policy Optimization to learn multiplicative KTP updates. We introduce an adaptive differential evolution (ADE) operator that adjusts its mutation and crossover parameters to strengthen intra-task search. For inter-task transfer, local structure learning (LSL) generates offspring using knowledge of task manifolds. The learned KTP then controls, for each individual, the probability of invoking LSL rather than ADE. Evaluated on the CEC21-MTMO-CPLX benchmark with IGD and normalized HV metrics alongside Wilcoxon statistical tests, our proposed method outperforms seven state-of-the-art MO-MTO competitors on most test instances. Ablation studies validate the individual contributions of the RL-based KTP controller and ADE component.The source code is available at:https://github.com/tomz18690-cell/RL-EMT-MSKT

1 INTRODUCTION

Multi-objective multi-task optimization (MO-MTO) improves search by transferring useful information across tasks, but transfer utility changes as populations evolve. Local manifold information reflects structural compatibility, offspring survival provides direct feedback on transfer effectiveness, and population progress or stagnation characterizes the current search state. Since these signals vary over time and may not always agree, fixed transfer probabilities cannot adapt transfer frequency to the evolving needs of individual tasks. This motivates task-specific transfer control that jointly considers structural information, search dynamics, and observed transfer outcomes.

Reinforcement learning provides a natural framework for sequential transfer control and has been applied to evolutionary parameter adaptation, operator selection, and knowledge-transfer control (Sharma et al., 2019; Sun et al., 2021; Tan & Li, 2021; Reijnen et al., 2022; Tian et al., 2023; Xue et al., 2022; Reijnen et al., 2025; Li et al., 2024; Wu et al., 2025). We formulate task-specific local manifold transfer control as a sequential decision problem, where selection outcomes provide feedback for subsequent transfer decisions.

We propose RL-EMT-MSKT to adaptively regulate the knowledge transfer probability (KTP) for each task during optimization. A Cross-Task Relation Encoder (CTRE) integrates the runtime representations of the paired tasks, including population statistics, manifold information, stagnation indicators, and offspring survival feedback, while a shared KTP Policy Actor maps the resulting representations to task-specific multiplicative KTP updates. The reward jointly considers population-quality improvement and the relative survival of offspring generated by local structure learning (LSL) and adaptive differential evolution (ADE), providing feedback on both optimization progress and

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

transfer effectiveness. For each parent, the current KTP determines whether the local offspring is generated by LSL or adaptive DE, with adaptive DE serving as the fallback when LSL is not selected or the required manifold models are invalid.

Our main contributions are as follows:

1. We formulate task-specific local transfer control as a sequential decision problem that integrates population progress, local manifold information, stagnation, and operator-specific survival feedback.
2. We design a Cross-Task Relation Encoder and a shared KTP Policy Actor to exchange and integrate paired-task representations, and generate task-specific multiplicative KTP updates under shared policy parameters.
3. We couple learned KTP control with offspring-level selection between LSL and adaptive DE, forming a closed-loop mechanism that adapts transfer decisions according to runtime feedback. Experiments on CEC21-MTMO-CPLX comprehensively evaluate solution quality, statistical significance, convergence behavior, KTP dynamics, and component contributions against seven representative MO-MTO methods.

2 Related Work

2.1 Multi-objective Multi-task Optimization

Multi-objective multi-task optimization (MO-MTO) seeks to improve search efficiency by exploiting complementary information across related tasks. Existing studies have investigated unified representations, task-relation modeling, transferable-solution selection, and search-space alignment (Gupta et al., 2017; Yuan et al., 2017; Bali et al., 2021; Chen et al., 2023; Liang et al., 2022). Collectively, these methods focus primarily on how cross-task knowledge is represented, identified, and transferred between tasks.

Manifold-based MO-MTO further exploits geometric structures learned from evolving populations. EMT-MSKT combines global and local structure learning with source selection to transfer manifold information across tasks (Zhang et al., 2026c). However, the effectiveness of local structural transfer changes with population states and search progress, motivating adaptive control of transfer frequency using structural cues and offspring feedback.

2.2 Adaptive Search and Reinforcement Learning for Evolutionary Control

Dynamic algorithm configuration adjusts parameters or operators according to runtime search states (Karafotias et al., 2015; Biedenkapp et al., 2020; Adriaensen et al., 2022). Reinforcement learning has been increasingly used for this purpose, including differential evolution parameter adaptation, mutation-operator selection, heterogeneous parameter control, and population-state representation (Reijnen et al., 2022; Tian et al., 2023; Xue et al., 2022; Reijnen et al., 2025). In evolutionary multitasking, RLMFEA adapts random-mating probabilities, while L2T learns transfer timing and transfer strategies (Li et al., 2024; Wu et al., 2025).

Our work focuses on task-specific control of local manifold transfer. The policy jointly exploits population states, local manifold information, and operator-specific survival feedback to adapt the probability of selecting LSL throughout the search. Adaptive differential evolution, inspired by MMDE (Chen et al., 2018), provides complementary within-task search and is employed whenever local manifold transfer is not selected. By integrating structural transfer with runtime feedback, the proposed design enables task-specific transfer decisions while retaining a shared policy across tasks.

3 Proposed Method

3.1 Overall Framework of RL-EMT-MSKT

Consider two multi-objective tasks T_1 and T_2, with populations P_1^g and P_2^g at generation g. As illustrated in Figure 1, RL-EMT-MSKT learns task-specific knowledge transfer probabilities (KTPs)

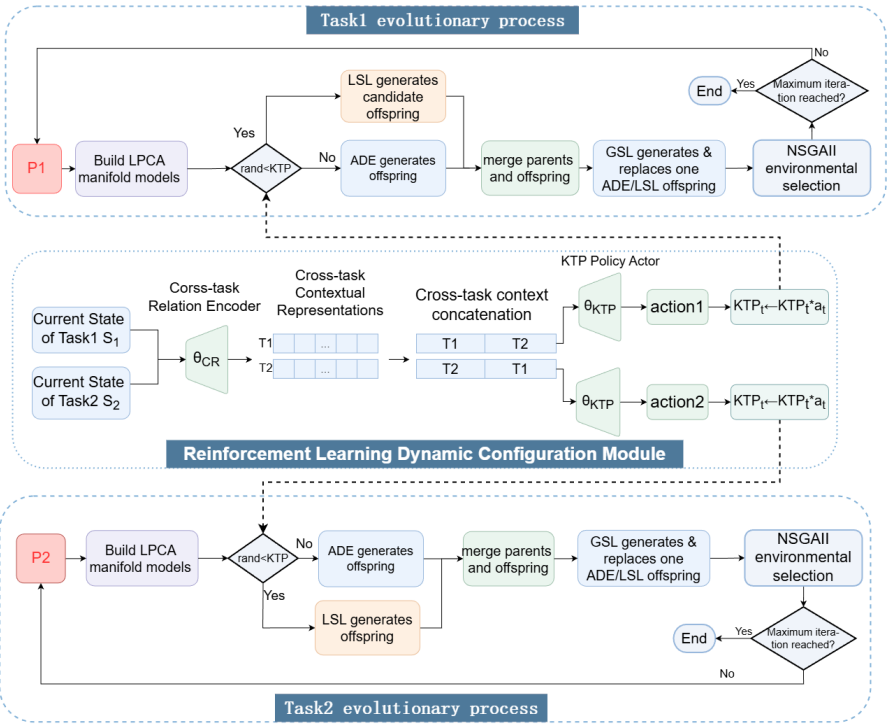


Figure 1: Overall framework of RL-EMT-MSKT.

by jointly encoding the evolutionary states of both tasks and producing multiplicative KTP adjustments. The updated KTP determines offspring-level selection between LSL and adaptive DE, and the resulting evolutionary outcomes are subsequently fed back to guide future KTP decisions.

For each parent, the current KTP determines whether LSL or ADE generates the local offspring. LSL is applied only when selected and when the required local principal component analysis (LPCA) models are valid; adaptive DE is used in all other cases. In addition, global structure learning (GSL) generates one offspring for each task at every generation. Environmental selection then constructs the next-generation populations and provides population-quality and operator-survival statistics to the KTP controller as feedback. Successful adaptive DE offspring are further used to update its parameter memories. The overall procedure is summarized in Algorithm 1.

3.2 Markov Decision Process for Transfer-Probability Control

We formulate task-specific KTP adaptation as a Markov decision process (MDP). At generation g, the state $\mathbf{s}_t^g$ describes the search status of target task t using population statistics, local manifold information, search progress, and recent operator outcomes. Based on this state and information from the counterpart task, the policy selects an action a_t^g that multiplicatively adjusts the KTP. The updated KTP affects offspring generation and environmental selection, whose outcomes determine the next state $\mathbf{s}_t^{g+1}$. The reward r_t^g evaluates this transition by combining population-quality improvement with the relative survival of LSL and adaptive DE offspring.

The resulting MDP is defined as $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, where $\mathcal{S}$ denotes the task-state space, $\mathcal{A}$ the KTP-adjustment action space, $\mathcal{P}$ the transition dynamics induced by evolutionary search, $\mathcal{R}$ the reward function, and γ the discount factor.

3.2.1 State Space

Each task uses an 11-dimensional runtime state $\mathbf{s}_t^g = [s_1, \ldots, s_{11}]$, whose components and control roles are summarized in Table 1. The state contains no problem indices or task labels; feature definitions and normalization are provided in Appendix A.

Table 1: Runtime state components for task-specific KTP control.

Components	Runtime evidence	Control role
(s_1, s_2)	Search progress and current KTP	Track optimization stage and transfer level
(s_3, s_4)	LSL/adaptive DE survival rates	Estimate recent operator effectiveness
(s_5, s_7)	SPEA2-based quality statistics	Characterize population quality and convergence
s_6	Decision-space diversity	Measure population contraction
(s_8, s_9, s_{10})	Manifold similarity and validity	Provide structural transfer cues
s_{11}	Stagnation	Detect lack of recent improvement

3.2.2 Action Space

For target task t, the KTP Policy Actor selects a multiplicative action $a_t^g \in \mathcal{A} = \{0.96, 0.98, 1.00, 1.02, 1.04\}$. The KTP is then updated as

$$\widehat{\mathrm{KTP}}_t^g = \mathrm{clip}\left(\mathrm{KTP}_t^g a_t^g,\, 0.05,\, 0.95\right), \qquad \mathrm{KTP}_t^{g+1} = \widehat{\mathrm{KTP}}_t^g. \tag{1}$$

Here, KTP_t^g denotes the KTP observed by the policy at generation g, whereas $\widehat{\mathrm{KTP}}_t^g$ denotes the updated probability of selecting LSL for local offspring generation. The updated value is applied within generation g and then propagated to generation $g+1$. Multiplicative actions induce gradual relative adjustments while accounting for the current KTP magnitude. The lower and upper bounds ensure that, whenever LSL is valid, both LSL and adaptive DE maintain nonzero selection probabilities. KTP governs only the choice between these two local offspring-generation operators; GSL and the parameter adaptation of adaptive DE operate independently of KTP control.

3.2.3 Reward Function

The reward evaluates both population-level progress and the observed effectiveness of local transfer. Let $\bar{\phi}_t^g$ and $\bar{\phi}_t^{g+1}$ denote the mean SPEA2 fitness of task t before and after environmental selection, respectively. Since lower SPEA2 fitness indicates better population quality, we define the normalized improvement as

$$\Delta\phi_t^g = \frac{\bar{\phi}_t^g - \bar{\phi}_t^{g+1}}{|\bar{\phi}_t^g| + \epsilon}, \qquad R_{\mathrm{quality},t}^g = \tanh(10\Delta\phi_t^g)\,. \tag{2}$$

This term reflects the population-level outcome of the complete evolutionary step. SPEA2 fitness is used only as a runtime quality indicator and does not participate in environmental selection.

To provide operator-specific feedback, we compare the recent survival rates of LSL and adaptive DE offspring:

$$R_{\mathrm{transfer},t}^g = \tanh\Big(5\left(SR_{\mathrm{LSL},t}^g - SR_{\mathrm{aDE},t}^g\right)\Big)\,, \tag{3}$$

where the survival rates are computed over the recent feedback window. A positive value indicates that LSL offspring have recently survived more frequently than adaptive DE offspring, while a negative value indicates the opposite.

The two signals are bounded and combined with equal weights:

$$r_t^g = 0.5R_{\mathrm{quality},t}^g + 0.5R_{\mathrm{transfer},t}^g, \qquad R^g = \frac{r_1^g + r_2^g}{2}. \tag{4}$$

Here, r_t^g provides task-specific feedback, while R^g aggregates the two task rewards for training the shared policy. This design encourages KTP decisions that improve population quality while favoring local transfer when its offspring remain competitive under environmental selection.

3.3 Cross-Task Relation Encoding and the KTP Policy Network

KTP control relies on the evolving states of both the target task and its counterpart. Inspired by attention-based cross-task representation modeling in MetaMTO (Zhan et al., 2026), we adopt a Cross-Task Relation Encoder (CTRE) to exchange and fuse information from paired-task representations prior to selecting task-specific KTP actions. The controller comprises a shared state embedder, CTRE, a shared KTP Policy Actor, and a critic.

The embedder maps each 11-dimensional task state to a 64-dimensional representation. Given

$$\mathbf{E}^g = [\mathbf{e}_1^g; \mathbf{e}_2^g],$$

CTRE computes single-head scaled dot-product attention with self-connections masked:

$$\mathbf{A}^g = \frac{(\mathbf{E}^g\mathbf{W}_Q)(\mathbf{E}^g\mathbf{W}_K)^{\mathrm{T}}}{\sqrt{64}}, \qquad \boldsymbol{\alpha}^g = \mathrm{softmax}(\mathbf{A}^g + \mathbf{M}), \tag{5}$$

where $M_{t,t} = -\infty$ and $M_{t,\bar{t}} = 0$. The exchanged contextual representations are computed by aggregating the projected value features $\mathbf{V}^g$:

$$\widetilde{\mathbf{E}}^g = \boldsymbol{\alpha}^g\mathbf{V}^g. \tag{6}$$

In the paired-task setting considered here, self-masking leaves one admissible counterpart for each target. Consequently, $\alpha_{t,t}^g = 0$, $\alpha_{t,\bar{t}}^g = 1$, and $\widetilde{\mathbf{e}}_t^g = \mathbf{v}_{\bar{t}}^g$. CTRE therefore performs deterministic counterpart representation exchange. Residual connections, batch normalization, and a feed-forward transformation subsequently fuse the target and counterpart information into the final representations

$$\mathbf{O}^g = [\mathbf{o}_1^g; \mathbf{o}_2^g].$$

Additional architectural details are provided in Appendix B.

For each target task, the actor receives the target representation followed by that of its counterpart:

$$\mathbf{z}_1^g = [\mathbf{o}_1^g; \mathbf{o}_2^g], \qquad \mathbf{z}_2^g = [\mathbf{o}_2^g; \mathbf{o}_1^g]. \tag{7}$$

This ordering preserves the identity of the controlled task while allowing both tasks to share the same actor parameters. The shared actor maps each 128-dimensional input to a categorical distribution over the five KTP multipliers:

$$\pi_\theta(a_t^g \mid \mathbf{z}_t^g) = \mathrm{Categorical}(\mathrm{logits} = f_\pi(\mathbf{z}_t^g; \theta)). \tag{8}$$

For the joint action $\mathbf{a}^g = (a_1^g, a_2^g)$, the factorized policy and joint-state value are defined as

$$\log \pi_\theta(\mathbf{a}^g \mid \mathbf{S}^g) = \sum_{t=1}^{2} \log \pi_\theta(a_t^g \mid \mathbf{z}_t^g), \qquad V_\phi(\mathbf{S}^g) = \frac{1}{2}\sum_{t=1}^{2} V_\phi(\mathbf{z}_t^g). \tag{9}$$

Thus, the controller produces task-specific actions while sharing policy and value-network parameters across tasks.

Transitions $(\mathbf{S}^g, \mathbf{a}^g, R^g, \mathbf{S}^{g+1})$ are used to estimate advantages $\widehat{A}^g$. The actor is optimized with the PPO clipped objective:

$$L^{\mathrm{CLIP}}(\theta) = \mathbb{E}_g\left[\min\left(\rho^g\widehat{A}^g, \mathrm{clip}(\rho^g, 1-\epsilon_p, 1+\epsilon_p)\widehat{A}^g\right)\right], \tag{10}$$

where ρ^g denotes the joint-action probability ratio between the current and behavior policies. The critic minimizes the squared error between $V_\phi(\mathbf{S}^g)$ and the estimated return. The shared controller is trained using the mean two-task reward R^g defined above, while task-specific inputs and factorized actions preserve separate KTP decisions. Network parameters are fixed during testing. Appendix B provides the remaining network dimensions, equations, and training schedule.

3.4 Evolutionary Environment and Conditional Transfer

The evolutionary environment converts task-specific KTP decisions into offspring generation and returns population-level and operator-specific feedback after environmental selection. It combines manifold-based knowledge transfer with adaptive differential evolution (adaptive DE), allowing learned transfer control and within-task search to operate within the same evolutionary cycle.

3.4.1 Manifold-Based Knowledge Transfer

The environment employs three manifold-based transfer mechanisms: global structure learning (GSL), local structure learning (LSL), and local source selection (LSS) (Zhang et al., 2026c). GSL transfers global search directions estimated by principal component analysis (PCA), while LSS identifies source regions for local transfer and LSL generates offspring from the corresponding local manifold models. The learned KTP controls the use of LSL at the offspring level. GSL operates independently and contributes one additional offspring to each task per generation.

3.4.2 ADAPTIVE DE AND KTP-GUIDED OPERATOR SELECTION

Within-task search is performed by an adaptive DE operator inspired by MMDE (Chen et al., 2018). For each parent $\mathbf{x}_i$, the mutation follows current-to-best/1:

$$\mathbf{v}_i = \mathbf{x}_i + F_i(\mathbf{x}_{best} - \mathbf{x}_i) + F_i(\mathbf{x}_{r_1} - \mathbf{x}_{r_2}), \tag{11}$$

followed by binomial crossover. The attraction term promotes search toward high-quality regions, while the differential term maintains directional variation.

Each task maintains success-history parameters μ_F and μ_{CR}. For offspring i,

$$F_i \sim \mathrm{Cauchy}(\mu_F, 0.1), \qquad CR_i \sim \mathcal{N}(\mu_{CR}, 0.1). \tag{12}$$

After environmental selection, parameter values associated with surviving adaptive DE offspring form the success sets S_F and S_{CR}. The memories are updated by

$$\begin{aligned} \mu_F &\leftarrow (1-c)\mu_F + c\frac{\sum_{F_i \in S_F} F_i^2}{\sum_{F_i \in S_F} F_i + \epsilon}, \\ \mu_{CR} &\leftarrow (1-c)\mu_{CR} + c\,\mathrm{mean}(S_{CR}). \end{aligned} \tag{13}$$

The Lehmer mean assigns greater influence to larger successful scaling factors, while the arithmetic mean summarizes successful crossover rates. A memory remains unchanged when its corresponding success set is empty. Parameter repair, crossover, and detailed offspring accounting are provided in Appendix C.

For each parent, the environment first samples $\xi_i \sim U(0,1)$ and then uses the executed KTP to determine which local operator generates the offspring. For task t,

$$q_{t,i}^{g} = \begin{cases} q_{t,i}^{\mathrm{LSL}}, & \xi_i < \widehat{\mathrm{KTP}}_t^{\,g} \text{ and the required LPCA models are valid,} \\ q_{t,i}^{\mathrm{aDE}}, & \text{otherwise.} \end{cases} \tag{14}$$

Thus, each parent produces exactly one local offspring. LSL generates the offspring only when both the learned transfer decision and the manifold-validity condition are satisfied; adaptive DE is used in all other cases. Environmental selection records the surviving offspring generated by LSL and adaptive DE, yielding operator-specific survival feedback for subsequent KTP decisions. Only surviving adaptive DE offspring are used to update the success-history parameter memories.

KTP-guided operator selection directly determines the composition of local offspring submitted to environmental selection. Each parent generates one offspring through either LSL or adaptive DE, with operator identities retained for separate survival-rate estimation. The resulting operator outcomes, together with the updated task states, provide feedback for subsequent KTP adaptation, while successful adaptive DE offspring are used to update its parameter memories. The policy parameters are optimized only during training and remain fixed throughout testing.

4 EXPERIMENTS AND ANALYSIS

4.1 EXPERIMENTAL DESIGN

Training and evaluation protocol. We evaluate final solution quality, convergence behavior, learned transfer dynamics, and the contributions of KTP control and ADE. The policy was trained for 300 episodes using 300 augmented two-task instances generated from the 10 CEC21-MTMO-CPLX problems, with 30 perturbed variants per problem. Shift and rotation strengths were set to 0.10 and 0.08, while task types, objective templates, and Paret front(PF) structures were preserved. After training, all network parameters were frozen and evaluated on the 10 unperturbed official problems, corresponding to 20 task-level evaluations. Each algorithm was independently executed 30 times on every problem. Details of data augmentation and implementation are provided in Appendix D.

Comparators. RL-EMT-MSKT was compared with seven MO-MTO algorithms: EMT-MSKT, MMTO-ETA, MMTO-TGT, KR-MTEA, MTDE-MKTA, MTEA-DCK, and MTEA-D-DN. These methods represent diverse strategies for cross-task knowledge transfer and multitask evolutionary search. All algorithms were evaluated on the same official test problems using identical function-evaluation budgets, run counts, and task-specific reference data. Method descriptions, citations, and complete numerical results are provided in Appendix E.

Algorithm 1 RL-EMT-MSKT

Require: Tasks T_1, T_2, population size N, per-task evaluation budget $\mathrm{FE}_{\max}$, policy π_θ, PPO update interval H

Initialize and evaluate P_1^0, P_2^0; set $\mathrm{FE}_t \leftarrow N$
Initialize KTP_t^0, $\mu_{F,t}$, $\mu_{CR,t}$, feedback statistics, and trajectory buffer $\mathcal{B}$
while $\mathrm{FE}_t < \mathrm{FE}_{\max}$ for both tasks **do**
    Build PCA/LPCA models and construct states $\mathbf{s}_1^g, \mathbf{s}_2^g$
    Encode both task states with CTRE
    for each task $t \in \{1, 2\}$ **do**
        Sample $a_t^g \sim \pi_\theta(\cdot \mid \mathbf{z}_t^g)$ and obtain $\widehat{\mathrm{KTP}}_t^g$
        Initialize $Q_t^g \leftarrow \emptyset$
        for each parent $\mathbf{x}_{t,i} \in P_t^g$ **do**
            Sample $\xi_i \sim U(0, 1)$
            if $\xi_i < \widehat{\mathrm{KTP}}_t^g$ and the required LPCA models are valid **then**
                Generate $q_{t,i} \leftarrow q_{t,i}^{\mathrm{LSL}}$ using LSL
            else
                Sample F_i, CR_i and generate $q_{t,i} \leftarrow q_{t,i}^{\mathrm{aDE}}$
            end if
            Add $q_{t,i}$ to Q_t^g
        end for
        Generate one GSL offspring $q_{t,\mathrm{GSL}}^g$
        Evaluate Q_t^g and $q_{t,\mathrm{GSL}}^g$; update FE_t
        Apply NSGA-II environmental selection to $P_t^g \cup Q_t^g \cup \{q_{t,\mathrm{GSL}}^g\}$
        Record offspring survival ratio for each operator and update $\mu_{F,t}$ and $\mu_{CR,t}$
        Compute r_t^g and the next task state
        Set $\mathrm{KTP}_t^{g+1} \leftarrow \widehat{\mathrm{KTP}}_t^g$
    end for
    Compute $R^g = (r_1^g + r_2^g)/2$ and store $(\mathbf{S}^g, \mathbf{a}^g, R^g, \mathbf{S}^{g+1})$ in $\mathcal{B}$
    if training mode and $(g + 1) \bmod H = 0$ **then**
        Update the actor and critic using PPO; clear $\mathcal{B}$
    end if
end while
return final nondominated solution sets for both tasks

Performance metrics. Performance is reported separately for each task using inverted generational distance (IGD), where lower values are better, and normalized hypervolume (HV), where higher values are better. For task t, a common HV reference point is constructed from the reference PF and the final nondominated solutions produced by all quantitatively compared algorithms:

$$\mathbf{r}_t = 1.1 \max\left(PF_t^*, \bigcup_{a \in \mathcal{A}} \bigcup_{j=1}^{30} A_{t,a}^{(j)}\right), \tag{15}$$

where the maximum is taken componentwise. Here, $\mathcal{A}$ contains RL-EMT-MSKT and the seven quantitative comparators. MO-MTEA-PAE is included only in PF visualizations and is therefore excluded from the quantitative reference-point construction. Definitions of IGD, normalized HV, and supplementary problem-level aggregation are provided in Appendix D.

Experimental settings. Each task was assigned a budget of 200,000 function evaluations. RL-EMT-MSKT used a population size of 100 per task. The principal algorithmic and policy settings are summarized in Table 2, with the remaining implementation details reported in Appendix D.

Statistical analysis. For each task and metric, the final results from 30 independent runs were compared using two-sided Wilcoxon rank-sum tests at a significance level of $p < 0.05$. Task-level comparisons constitute the primary statistical analysis, while mean IGD over the two tasks of each problem is reported as supplementary problem-level evidence. In the main comparison table, $W/L/NS$ denotes a significant win, significant loss, or nonsignificant difference for RL-EMT-MSKT relative to the corresponding comparator. Supplementary tables follow the sign conventions stated in their captions.

Table 2: Parameter settings of RL-EMT-MSKT.

Setting	Value	Setting	Value
Official test problems	10 two-task CEC21-MTMO-CPLX problems		
Population size per task	100	Independent runs	30
Evaluation budget per task	200,000	Training episodes	300
Initial KTP	0.5	KTP range	$[0.05, 0.95]$
State / embedding dimensions	$11/64$	Adaptive DE update rate c	0.1
PPO learning rate	3×10^{-4}	PPO clipping coefficient	0.2
PPO update interval	10 generations	KTP action multipliers	$\{0.96, 0.98, 1.00, 1.02, 1.04\}$

Table 3: Task-level final IGD (mean $\pm$ standard deviation) on CEC21-MTMO-CPLX. Parenthetical markers report Holm-corrected pairwise comparisons with RL-EMT-MSKT.

Task	RL-EMT-MSKT	EMT-MSKT	MMTO-ETA	MMTO-TGT	KR-MTEA	MTDE-MKTA	MTEA-DCK	MTEA-D-DN
P1-T1	$3.42 \times 10^4 \pm 4.49 \times 10^4$	$1.83 \times 10^4 \pm 8410(=)$	$2.21 \times 10^4 \pm 1.04 \times 10^4(=)$	$\mathbf{1.54 \times 10^4 \pm 6375}(=)$	$9.35 \times 10^5 \pm 8.45 \times 10^5(-)$	$5.51 \times 10^4 \pm 3.19 \times 10^4(-)$	$2.67 \times 10^4 \pm 1.52 \times 10^4(=)$	$1.74 \times 10^5 \pm 9.97 \times 10^4(-)$
P1-T2	$\mathbf{5968 \pm 1948}$	$1.47 \times 10^4 \pm 5882(-)$	$2.92 \times 10^4 \pm 1.76 \times 10^4(-)$	$1.30 \times 10^4 \pm 5659(-)$	$2.55 \times 10^7 \pm 7.79 \times 10^6(-)$	$5.57 \times 10^4 \pm 3.58 \times 10^4(-)$	$2.85 \times 10^4 \pm 2.00 \times 10^4(-)$	$2.77 \times 10^5 \pm 1.25 \times 10^5(-)$
P2-T1	15.9 ± 5.54	$\mathbf{12.5 \pm 3.57}(=)$	$14.9 \pm 3.78(=)$	$21.2 \pm 6.02(-)$	$25.3 \pm 3.73(-)$	$27.3 \pm 2.59(-)$	$20.3 \pm 5.01(-)$	$34.1 \pm 23.3(-)$
P2-T2	15.7 ± 6.16	$\mathbf{11.8 \pm 3.03}(=)$	$15.4 \pm 5.13(=)$	$22.6 \pm 6.88(-)$	$36.0 \pm 7.35(-)$	$27.8 \pm 1.95(-)$	$19.2 \pm 4.71(-)$	$34.0 \pm 22.9(-)$
P3-T1	$\mathbf{132 \pm 150}$	$1917 \pm 239(-)$	$1856 \pm 189(-)$	$1750 \pm 433(-)$	$2253 \pm 238(-)$	$1922 \pm 337(-)$	$1835 \pm 219(-)$	$1312 \pm 521(-)$
P3-T2	$\mathbf{238 \pm 189}$	$2015 \pm 238(-)$	$1829 \pm 321(-)$	$1874 \pm 516(-)$	$3450 \pm 234(-)$	$1902 \pm 293(-)$	$1790 \pm 437(-)$	$1577 \pm 426(-)$
P4-T1	$\mathbf{8.25 \pm 2.17}$	$19.4 \pm 5.01(-)$	$20.4 \pm 6.57(-)$	$12.9 \pm 3.82(-)$	$9.67 \pm 3.82(=)$	$12.1 \pm 3.56(-)$	$23.7 \pm 9.28(-)$	$39.9 \pm 13.2(-)$
P4-T2	$\mathbf{10.1 \pm 3.36}$	$24.7 \pm 6.05(-)$	$25.1 \pm 7.46(-)$	$13.7 \pm 5.36(-)$	$27.6 \pm 1.65(-)$	$15.3 \pm 4.76(-)$	$27.3 \pm 8.95(-)$	$55.2 \pm 18.4(-)$
P5-T1	56.3 ± 50.6	$93.0 \pm 45.0(-)$	$112 \pm 63.9(-)$	$55.6 \pm 41.5(=)$	$\mathbf{29.4 \pm 33.8}(=)$	$77.9 \pm 43.4(=)$	$89.9 \pm 61.4(=)$	$90.4 \pm 55.0(-)$
P5-T2	18.3 ± 28.3	$34.6 \pm 34.6(-)$	$25.5 \pm 35.0(=)$	$28.0 \pm 32.6(=)$	$\mathbf{7.48 \pm 21.2}(+)$	$49.6 \pm 41.9(-)$	$35.8 \pm 41.5(=)$	$73.6 \pm 47.3(-)$
P6-T1	$\mathbf{53.3 \pm 11.1}$	$76.3 \pm 14.5(-)$	$93.9 \pm 18.8(-)$	$83.1 \pm 18.8(-)$	$60.3 \pm 13.0(=)$	$113 \pm 29.1(-)$	$89.3 \pm 19.9(-)$	$165 \pm 29.7(-)$
P6-T2	$\mathbf{60.0 \pm 11.7}$	$88.5 \pm 21.2(-)$	$99.2 \pm 12.7(-)$	$89.3 \pm 24.9(-)$	$366 \pm 10.7(-)$	$108 \pm 25.9(-)$	$93.9 \pm 22.0(-)$	$182 \pm 39.3(-)$
P7-T1	16.5 ± 4.36	$16.4 \pm 5.55(=)$	$6.31 \pm 2.52(+)$	$15.8 \pm 5.37(=)$	$\mathbf{1.53 \pm 1.82}(+)$	$19.8 \pm 5.10(-)$	$23.3 \pm 7.63(-)$	$7.73 \pm 9.01(+)$
P7-T2	18.7 ± 4.82	$14.9 \pm 5.16(+)$	$3.28 \pm 2.38(+)$	$19.0 \pm 6.45(=)$	$237 \pm 12.8(-)$	$22.4 \pm 6.15(-)$	$36.5 \pm 14.5(-)$	$\mathbf{0.0157 \pm 3.60 \times 10^{-3}}(+)$
P8-T1	1202 ± 1261	$1589 \pm 1330(=)$	$1498 \pm 1311(=)$	$1758 \pm 2009(=)$	$2440 \pm 2395(-)$	$3967 \pm 4194(-)$	$\mathbf{1070 \pm 772}(=)$	$2543 \pm 3100(-)$
P8-T2	$\mathbf{5200 \pm 5716}$	$1.79 \times 10^6 \pm 9.45 \times 10^5(-)$	$1.55 \times 10^6 \pm 5.54 \times 10^5(-)$	$8.78 \times 10^5 \pm 3.64 \times 10^5(-)$	$2.69 \times 10^6 \pm 2.42 \times 10^5(-)$	$2.93 \times 10^6 \pm 1.50 \times 10^6(-)$	$7.52 \times 10^5 \pm 2.46 \times 10^5(-)$	$1.11 \times 10^4 \pm 9347(-)$
P9-T1	$\mathbf{5032 \pm 793}$	$1.13 \times 10^4 \pm 882(-)$	$1.17 \times 10^4 \pm 1003(-)$	$1.04 \times 10^4 \pm 3102(-)$	$1.57 \times 10^4 \pm 1517(-)$	$1.31 \times 10^4 \pm 1291(-)$	$1.22 \times 10^4 \pm 1720(-)$	$6910 \pm 1061(-)$
P9-T2	506 ± 321	$1413 \pm 1016(-)$	$1200 \pm 892(-)$	$866 \pm 895(-)$	$3.60 \times 10^6 \pm 2.18 \times 10^6(-)$	$3634 \pm 3517(-)$	$\mathbf{465 \pm 143}(=)$	$2820 \pm 2320(-)$
P10-T1	$\mathbf{10.9 \pm 3.79}$	$25.3 \pm 7.27(-)$	$26.4 \pm 5.34(-)$	$13.4 \pm 3.17(-)$	$17.9 \pm 7.84(-)$	$12.0 \pm 2.85(=)$	$25.1 \pm 5.34(-)$	$29.1 \pm 8.15(-)$
P10-T2	$\mathbf{4.12 \times 10^4 \pm 2.33 \times 10^4}$	$1.48 \times 10^5 \pm 6.94 \times 10^4(-)$	$2.65 \times 10^5 \pm 1.78 \times 10^5(-)$	$4.54 \times 10^4 \pm 3.01 \times 10^4(=)$	$2.67 \times 10^7 \pm 6.12 \times 10^6(-)$	$1.43 \times 10^5 \pm 6.03 \times 10^4(-)$	$1.08 \times 10^5 \pm 7.89 \times 10^4(-)$	$4.83 \times 10^5 \pm 2.88 \times 10^5(-)$
IGD (+/ − / =)	–	1/14/5	2/13/5	0/13/7	2/15/3	0/18/2	0/15/5	2/18/0

Note: *Lower IGD is better. Bold numbers indicate the lowest mean IGD values. All algorithms were evaluated over 30 independent runs. Two-sided Mann–Whitney U (Wilcoxon rank-sum) tests were conducted at a significance level 0.05 with Holm correction. $(+/-/=)$ denotes the numbers of tasks on which the corresponding baseline algorithm performs significantly better than, significantly worse than, or is not significantly different from RL-EMT-MSKT, respectively.

4.2 Experimental Results and Analysis

4.2.1 Comparison with MO-MTO Algorithms

Table 3 reports task-level IGD statistics for all eight algorithms. RL-EMT-MSKT achieved the best mean IGD on 11 of the 20 tasks, including P1-T2, both tasks of P3, P4, P6, and P10, as well as P8-T2 and P9-T1. A particularly large difference was observed on P3, where RL-EMT-MSKT obtained mean IGD values of 132 and 238 on the two tasks, compared with 1312 and 1577 for the strongest comparator on the corresponding tasks.

The pairwise statistical results in the final row provide a complementary view of performance across all 20 tasks. Depending on the comparator, RL-EMT-MSKT achieved significant IGD improvements on 13 to 18 tasks, while significant degradations occurred on no more than two tasks. These results indicate that RL-EMT-MSKT generally provides more reliable reference-front approximation across the benchmark, although the magnitude of its advantage varies across tasks.

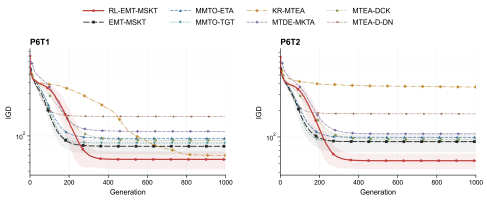
The performance gains are nevertheless problem dependent. RL-EMT-MSKT did not obtain the lowest mean IGD on P1-T1, either task of P2, either task of P5, either task of P7, or P8-T1, or P9-T2. The most notable case is P7-T2, where MTEA-D-DN achieved a substantially lower mean IGD. These cases indicate that the learned transfer policy does not universally dominate alternative transfer and search strategies, particularly when the underlying task characteristics favor the inductive biases of competing methods. Complete normalized HV results and additional statistical analyses are provided in Appendix E.

4.2.2 Convergence and Transfer Behavior

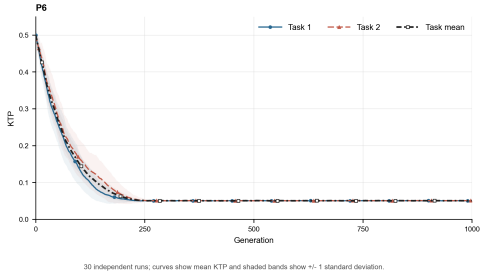
Figure 2 illustrates the search dynamics on Problem 6 through IGD convergence, learned KTP trajectories, and PF approximations.

In Figure 2(a), RL-EMT-MSKT attains lower IGD values than most comparators on both tasks and then converges toward a stable region, consistent with the final performance reported in Table 3. Figure 2(b) shows that the learned KTP gradually decreases from its initial value of 0.5 toward the lower bound of 0.05 for both tasks. The decline begins earlier on P6-T1, and both trajectories

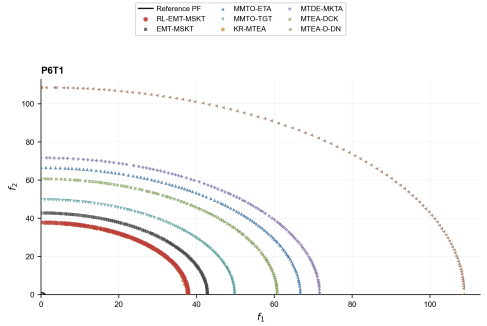
approach the lower bound within approximately 250 generations. This pattern suggests that the learned policy progressively favors reduced use of LSL as the search proceeds on Problem 6. However, the KTP trajectories alone are insufficient to establish a causal relationship between reduced LSL usage and the observed IGD improvement.



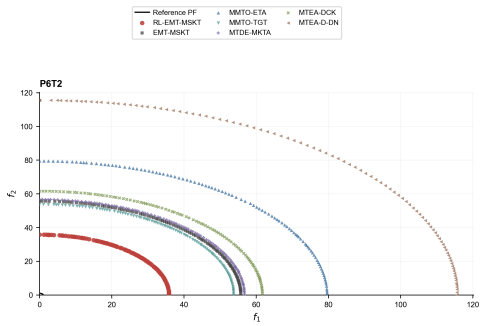
(a) IGD convergence curves



(b) KTP trajectories



(c) P6-T1's PF approximations



(d) P6-T2's PF approximations

Figure 2: Search behavior on Problem 6. (a) IGD convergence curves of all algorithms. (b) KTP trajectories of P6-T1 and P6-T2 learned by the RL-EMT-MSKT. (c) and (d) are PF approximations of the run with best IGD values among 30 independent runs. Shaded bands in (a) and (b) indicate variablity across runs. KR-MTEA is omitted from (d) because its final PF approximation is beyond the display range.

Figures 2(c) and (d) provide qualitative views of the best-run approximations obtained by RL-EMT-MSKT, with IGD values of 27.9 on P6-T1 and 33.8 on P6-T2. Because these panels show selected runs, the aggregate statistics in Table 3 remain the primary evidence for comparative performance. Additional visual results are provided in Appendix G.

4.3 Ablation Studies

Across the 20 tasks, the full method achieves the lowest average task-level IGD rank of 1.65, outperforming the variants without learned KTP control (1.90) and adaptive DE (2.45). These consistent performance advantages validate the efficacy of both core components. Appendix F presents the corresponding visualization results.

5 Conclusion

We proposed RL-EMT-MSKT for task-specific control of local manifold transfer. A shared policy leverages cross-task evolutionary states and operator-survival feedback to adapt KTP, which determines the selection between LSL and adaptive DE for local offspring generation. Across the 20 tasks of CEC21-MTMO-CPLX, RL-EMT-MSKT achieved the lowest mean IGD on 11 tasks among eight algorithms and obtained at least 13 significant IGD wins against each comparator. The ablation results further indicate that both learned KTP control and adaptive DE contribute to the overall performance.

Evaluation remains limited to two-task problems drawn from one benchmark family. Future work will examine disjoint problem families, more than two tasks, real-world settings, larger distribution shifts, and the overhead of policy inference and manifold modeling.

Reproducibility statement

The main paper and appendices specify the optimization setting, training-instance construction, evaluation budgets, metrics, parameter settings, statistical tests, network equations, reward function, and evolutionary procedure. Task-level summary statistics and component ablations document the reported results.

Ethics statement

This study evaluates optimization algorithms on synthetic benchmarks and involves no human participants, personal data, or safety-critical deployment. We identify no direct ethical risks specific to the reported experiments.

AI use statement

No generative AI tools were used in any stage of this work.All content, claims, equations, results, citations, and artifacts are produced and reviewed solely by the authors.

References

Steven Adriaensen, André Biedenkapp, Gresa Shala, Noor Awad, Theresa Eimer, Marius Lindauer, and Frank Hutter. Automated dynamic algorithm configuration. *Journal of Artificial Intelligence Research*, 75:1633–1699, 2022. doi: 10.1613/jair.1.13922.

Kavitesh Kumar Bali, Abhishek Gupta, Yew-Soon Ong, and Puay Siew Tan. Cognizant multitasking in multiobjective multifactorial evolution: MO-MFEA-II. *IEEE Transactions on Cybernetics*, 51 (4):1784–1796, 2021. doi: 10.1109/TCYB.2020.2981733.

André Biedenkapp, H. Furkan Bozkurt, Theresa Eimer, Frank Hutter, and Marius Lindauer. Dynamic algorithm configuration: Foundation of a new meta-algorithmic framework. In Giuseppe De Giacomo, Alejandro Catalá, Bistra Dilkina, Michela Milano, Senén Barro, Alberto Bugarín, and Jérôme Lang (eds.), *ECAI 2020*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, pp. 427–434. IOS Press, 2020. doi: 10.3233/FAIA200122.

Hongyan Chen, Hai-Lin Liu, Fangqing Gu, and Kay Chen Tan. A multiobjective multitask optimization algorithm using transfer rank. *IEEE Transactions on Evolutionary Computation*, 27(2): 237–250, 2023. doi: 10.1109/TEVC.2022.3147568.

Yongliang Chen, Jinghui Zhong, and Mingkui Tan. A fast memetic multi-objective differential evolution for multi-tasking optimization. In *2018 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1–8, Rio de Janeiro, 2018. IEEE. doi: 10.1109/CEC.2018.8477722.

Zhihua Cui, Ben Zhao, Tianhao Zhao, Xingjuan Cai, and Jinjun Chen. Adaptive multi-task evolutionary algorithm based on knowledge reuse. *Information Sciences*, 648:119568, August 2023. doi: 10.1016/j.ins.2023.119568.

Abhishek Gupta, Yew-Soon Ong, Liang Feng, and Kay Chen Tan. Multiobjective multifactorial optimization in evolutionary multitasking. *IEEE Transactions on Cybernetics*, 47(7):1652–1665, 2017. doi: 10.1109/TCYB.2016.2554622.

Giorgos Karafotias, Mark Hoogendoorn, and A. E. Eiben. Parameter control in evolutionary algorithms: Trends and challenges. *IEEE Transactions on Evolutionary Computation*, 19(2):167–187, 2015. doi: 10.1109/TEVC.2014.2308294.

Shuijia Li, Wenyin Gong, Ling Wang, and Qiong Gu. Evolutionary multitasking via reinforcement learning. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 8(1):762–775, 2024. doi: 10.1109/TETCI.2023.3281876.

Yanchi Li and Wenyin Gong. Multiobjective multitask optimization with multiple knowledge types and transfer adaptation. *IEEE Transactions on Evolutionary Computation*, 29(1):205–216, 2025. doi: 10.1109/TEVC.2024.3353319.

Yanchi Li, Dongcheng Li, Wenyin Gong, and Qiong Gu. Multiobjective multitask optimization via diversity- and convergence-oriented knowledge transfer. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 55(3):2367–2379, 2025. doi: 10.1109/TSMC.2024.3520526.

Yanchi Li, Wenyin Gong, Tingyu Zhang, Fei Ming, Shuijia Li, Qiong Gu, and Yew-Soon Ong. MToP: A MATLAB benchmarking platform for evolutionary multitasking. *ACM Transactions on Evolutionary Learning and Optimization*, 2026. doi: 10.1145/3812535.

Zhengping Liang, Hao Dong, Cheng Liu, Weiqi Liang, and Zexuan Zhu. Evolutionary multitasking for multiobjective optimization with subspace alignment and adaptive differential evolution. *IEEE Transactions on Cybernetics*, 52(4):2096–2109, 2022. doi: 10.1109/TCYB.2020.2980888.

Robbert Reijnen, Yingqian Zhang, Zaharah Bukhsh, and Mateusz Guzek. Deep reinforcement learning for adaptive parameter control in differential evolution for multi-objective optimization. In *2022 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 804–811, Singapore, Singapore, 2022. IEEE. doi: 10.1109/SSCI51031.2022.10022227.

Robbert Reijnen, Yaoxin Wu, Zaharah Bukhsh, and Yingqian Zhang. Graph-supported dynamic algorithm configuration for multi-objective combinatorial optimization. arXiv preprint arXiv:2505.16471, 2025.

Mudita Sharma, Alexandros Komninos, Manuel López-Ibáñez, and Dimitar Kazakov. Deep reinforcement learning based parameter control in differential evolution. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 709–717, Prague Czech Republic, 2019. ACM. doi: 10.1145/3321707.3321813.

Jianyong Sun, Xin Liu, Thomas Back, and Zongben Xu. Learning adaptive differential evolution algorithm from optimization experiences by policy gradient. *IEEE Transactions on Evolutionary Computation*, 25(4):666–680, 2021. doi: 10.1109/TEVC.2021.3060811.

Zhiping Tan and Kangshun Li. Differential evolution with mixed mutation strategy based on deep reinforcement learning. *Applied Soft Computing*, 111:107678, 2021. doi: 10.1016/j.asoc.2021.107678.

Ye Tian, Xiaopeng Li, Haiping Ma, Xingyi Zhang, Kay Chen Tan, and Yaochu Jin. Deep reinforcement learning based adaptive operator selection for evolutionary multi-objective optimization. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 7(4):1051–1064, 2023. doi: 10.1109/TETCI.2022.3146882.

Xianpeng Wang, Zhiming Dong, Lixin Tang, and Qingfu Zhang. Multiobjective multitask optimization - neighborhood as a bridge for knowledge transfer. *IEEE Transactions on Evolutionary Computation*, 27(1):155–169, 2023. doi: 10.1109/TEVC.2022.3154416.

Sheng-Hao Wu, Yuxiao Huang, Xingyu Wu, Liang Feng, Zhi-Hui Zhan, and Kay Chen Tan. Learning to transfer for evolutionary multitasking. *IEEE Transactions on Cybernetics*, 55(7):3342–3355, 2025. doi: 10.1109/TCYB.2025.3561518.

Ke Xue, Jiacheng Xu, Lei Yuan, Miqing Li, Chao Qian, Zongzhang Zhang, and Yang Yu. Multi-agent dynamic algorithm configuration. In *Advances in Neural Information Processing Systems 35*, 2022.

Yuan Yuan, Yew-Soon Ong, Liang Feng, A. K. Qin, Abhishek Gupta, Bingshui Da, Qingfu Zhang, Kay Chen Tan, Yaochu Jin, and Hisao Ishibuchi. Evolutionary multitasking for multiobjective continuous optimization: Benchmark problems, performance metrics and baseline results. arXiv preprint arXiv:1706.02766, 2017.

Jiajun Zhan, Zeyuan Ma, Yue-Jiao Gong, and Kay Chen Tan. Learning where, what and how to transfer: A multi-role reinforcement learning approach for evolutionary multitasking. *IEEE Transactions on Evolutionary Computation*, pp. 1–1, 2026. doi: 10.1109/TEVC.2026.3707341.

Tingyu Zhang, Kuihua Huang, Xinyi Wu, Yanchi Li, Jianfeng Liu, Shuijia Li, and Wenyin Gong. Ensemble of transfer techniques via adaptive resource allocation for multiobjective multitask optimization. *IEEE Transactions on Artificial Intelligence*, pp. 1–13, 2026a. doi: 10.1109/TAI.2026.3689513.

Tingyu Zhang, Xinyi Wu, Wenyin Gong, Shuijia Li, Yanchi Li, and A. K. Qin. Topology-guided knowledge transfer for multiobjective multitask optimization. *IEEE Transactions on Evolutionary Computation*, 2026b. doi: 10.1109/TEVC.2026.3696355.

Tingyu Zhang, Xinyi Wu, Yanchi Li, Shuijia Li, and Wenyin Gong. Multiobjective multitask optimization with manifold structure-driven knowledge transfer. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 56(4):2426–2438, 2026c. doi: 10.1109/TSMC.2025.3649528.

A Detailed MDP State Definition

Table 1 groups the state components by their roles in KTP control. This appendix defines the components and their associated statistics.

At generation g, task t is represented by the 11-dimensional state

$$\mathbf{s}_t^g = [s_1, s_2, \ldots, s_{11}].$$

Table 4 defines each component. Here, G is the maximum number of generations, W is the feedback-window length, and $\mathcal{W}_g = \{\max(1, g - W + 1), \ldots, g\}$ for $g \geq 1$. The constant $\epsilon > 0$ stabilizes divisions. Feedback-dependent components are initialized to zero at $g = 0$. For task t at generation τ, $N^{\tau}_{\mathrm{LSL,in},t}$ and $N^{\tau}_{\mathrm{LSL,survive},t}$ count LSL offspring entering and surviving environmental selection, respectively. The corresponding counts for adaptive DE offspring are $N^{\tau}_{\mathrm{aDE,in},t}$ and $N^{\tau}_{\mathrm{aDE,survive},t}$. Finally, $\bar{\phi}_t^{\tau}$ denotes mean population SPEA2 fitness, with lower values indicating better quality.

Table 4: MDP state components in RL-EMT-MSKT.

State component	Definition	Interpretation
s_1	$\dfrac{\mathrm{FE}_t}{\mathrm{FE}_{\max}}$	Normalized function-evaluation progress
s_2	KTP_t^g	Current LSL selection probability
s_3	$\dfrac{\sum_{\tau\in\mathcal{W}_g} N^{\tau}_{\mathrm{LSL,survive},t}}{\sum_{\tau\in\mathcal{W}_g} N^{\tau}_{\mathrm{LSL,in},t} + \epsilon}$	LSL-offspring survival rate over the recent window
s_4	$\dfrac{\sum_{\tau\in\mathcal{W}_g} N^{\tau}_{\mathrm{aDE,survive},t}}{\sum_{\tau\in\mathcal{W}_g} N^{\tau}_{\mathrm{aDE,in},t} + \epsilon}$	Adaptive DE offspring survival rate over the recent window
s_5	$\dfrac{1}{\lvert\mathcal{W}_g\rvert} \sum_{\tau\in\mathcal{W}_g} \dfrac{\bar{\phi}_t^{\tau-1} - \bar{\phi}_t^{\tau}}{\lvert\bar{\phi}_t^{\tau-1}\rvert + \epsilon}$	Mean SPEA2 fitness improvement over the recent window
s_6	$\dfrac{1}{D} \sum_{d=1}^{D} \mathrm{std}(X_{:,d})$	Mean decision-space diversity
s_7	$\dfrac{\bar{\phi}_t^g}{\bar{\phi}_t^0 + \epsilon}$	SPEA2 convergence state relative to the initial population
s_8	$\begin{cases} \frac{1}{\lvert\Omega_t^g\rvert} \sum_{(i,j)\in\Omega_t^g} \exp(-d_{ij}^g), & \lvert\Omega_t^g\rvert > 0, \\ s_{\mathrm{empty}}, & \lvert\Omega_t^g\rvert = 0 \end{cases}$	Mean inter-task local-manifold similarity
s_9	$\begin{cases} \max_{(i,j)\in\Omega_t^g} \exp(-d_{ij}^g), & \lvert\Omega_t^g\rvert > 0, \\ s_{\mathrm{empty}}, & \lvert\Omega_t^g\rvert = 0 \end{cases}$	Maximum inter-task local-manifold similarity
s_{10}	$\dfrac{N^g_{\text{valid cluster},t}}{N^g_{\text{cluster},t} + \epsilon}$	Proportion of valid LPCA clusters
s_{11}	$\dfrac{\mathrm{stagnation}_t^g}{G}$	Normalized stagnation duration

In Table 4, $X \in \mathbb{R}^{N\times D}$ is the normalized decision matrix, where N is the population size and D is the decision-space dimension. The column $X_{:,d}$ contains the values of the d-th decision variable across individuals. Normalization limits the influence of differences in variable ranges on the diversity feature.

Inter-task manifold similarity is computed after LPCA clustering, excluding clusters with insufficient samples or unstable local principal component models. The quantities $N^g_{\text{cluster},t}$ and $N^g_{\text{valid cluster},t}$ denote the total and valid cluster counts, respectively. For a valid target cluster i and source cluster j, d_{ij}^g measures the distance between their local manifold models, and $\exp(-d_{ij}^g)$ is the corresponding similarity. The set Ω_t^g contains all valid cross-task cluster pairs. If it is empty, both similarity features are set to s_{empty}.

B Network Architecture Details

B.1 State Embedding

Let the task states at generation g be

$$\mathbf{S}^g = \{\mathbf{s}_t^g\}_{t=1}^2,$$

where $\mathbf{s}_t^g \in \mathbb{R}^{11}$ contains progress, current KTP, LSL and adaptive DE offspring survival, SPEA2 fitness change, decision-space diversity, inter-task manifold similarity, and stagnation.

A shared state embedder $f_{\text{emb}}(\cdot;\theta_{\text{emb}})$ maps each task state to a common latent space:

$$\mathbf{e}_t^g = f_{\text{emb}}\left(\mathbf{s}_t^g \mid \theta_{\text{emb}}\right), \quad \mathbf{e}_t^g \in \mathbb{R}^{64}.$$

Stacking the two embeddings gives

$$\mathbf{E}^g = f_{\text{emb}}\left(\mathbf{S}^g \mid \theta_{\text{emb}}\right) = [\mathbf{e}_1^g; \mathbf{e}_2^g] \in \mathbb{R}^{2\times 64}.$$

Parameter sharing ensures that both task states use the same feature map. CTRE is permutation equivariant with respect to task order, and the subsequent ordered concatenation identifies the target task.

B.2 Cross-Task Relation Encoder

The Cross-Task Relation Encoder (CTRE) adapts MetaMTO's attention-based relation modeling to two-task KTP control (Zhan et al., 2026). CTRE exchanges context between the paired task representations before KTP actions are selected. Its self-connections are masked so that each task receives contextual information from its counterpart.

Given $\mathbf{E}^g \in \mathbb{R}^{2\times d_{\text{emb}}}$, with $d_{\text{emb}} = 64$, CTRE computes single-head scaled dot-product scores between the two task embeddings.

The query, key, and value matrices are computed as

$$\mathbf{Q}^g = \mathbf{E}^g\mathbf{W}_Q, \qquad \mathbf{K}^g = \mathbf{E}^g\mathbf{W}_K, \qquad \mathbf{V}^g = \mathbf{E}^g\mathbf{W}_V, \tag{16}$$

where $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{d_{\text{emb}}\times d_{\text{emb}}}$ are independent trainable projection matrices. The scaled attention-score matrix is

$$\mathbf{A}^g = \frac{\mathbf{Q}^g(\mathbf{K}^g)^{\mathrm{T}}}{\sqrt{d_{\text{emb}}}} \in \mathbb{R}^{2\times 2}. \tag{17}$$

Let $\mathbf{M} \in \mathbb{R}^{2\times 2}$ be the self-attention mask, where $M_{t,t} = -\infty$ and $M_{t,\bar{t}} = 0$. Applying a row-wise softmax after masking yields

$$\boldsymbol{\alpha}^g = \mathrm{softmax}\left(\mathbf{A}^g + \mathbf{M}\right). \tag{18}$$

For task t, let $\bar{t}$ denote its counterpart, where $\bar{1} = 2$ and $\bar{2} = 1$, and let $\mathbf{v}_t^g$ denote the t-th row of $\mathbf{V}^g$. The attention-based contextual representation of task t is

$$\widetilde{\mathbf{e}}_t^g = \alpha_{t,t}^g\mathbf{v}_t^g + \alpha_{t,\bar{t}}^g\mathbf{v}_{\bar{t}}^g, \tag{19}$$

Because the diagonal entry is masked by $-\infty$ and one non-self counterpart is available for each target in the present two-task setting,

$$\alpha^g_{t,t} = 0, \qquad \alpha^g_{t,\bar{t}} = 1. \tag{20}$$

Therefore,

$$\widetilde{\mathbf{e}}^g_t = \mathbf{v}^g_{\bar{t}}. \tag{21}$$

In matrix form,

$$\widetilde{\mathbf{E}}^g = \boldsymbol{\alpha}^g \mathbf{V}^g. \tag{22}$$

A residual connection reincorporates the original task embeddings, followed by batch normalization:

$$\mathbf{H}^g = \mathrm{LayerNorm}\left(\mathbf{E}^g + \widetilde{\mathbf{E}}^g\right). \tag{23}$$

A feed-forward network applies a nonlinear transformation:

$$\mathrm{FFN}(\mathbf{x}) = \mathbf{W}_2\sigma\left(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1\right) + \mathbf{b}_2, \tag{24}$$

where $\sigma(\cdot)$ is the ReLU activation function. The final relation-aware task representations are given by

$$\mathbf{O}^g = \mathrm{LayerNorm}\left(\mathbf{H}^g + \mathrm{FFN}\left(\mathbf{H}^g\right)\right), \tag{25}$$

where

$$\mathbf{O}^g = \begin{bmatrix} \mathbf{o}^g_1 \\ \mathbf{o}^g_2 \end{bmatrix} \in \mathbb{R}^{2\times 64}. \tag{26}$$

Thus, each $\mathbf{o}^g_t$ combines its residual target embedding with the exchanged, key-projected counterpart embedding and subsequent nonlinear transformation. It is placed first in the ordered input used to select the KTP adjustment for target task t.

B.3 KTP Policy Actor

The KTP Policy Actor controls the probability of selecting LSL rather than adaptive DE for local offspring generation. EMT-MSKT provides LSS-based source-region selection and GSL/LSL offspring generation, while adaptive DE adapts F and CR separately.

For each target task t, an ordered target-counterpart representation is constructed by concatenating its encoded representation with that of its counterpart:

$$\mathbf{z}^g_t = \mathrm{Concat}\left(\mathbf{o}^g_t, \mathbf{o}^g_{\bar{t}}\right), \tag{27}$$

where $\bar{1} = 2$ and $\bar{2} = 1$. Accordingly, the ordered representations of the two tasks are

$$\begin{aligned} \mathbf{z}^g_1 &= \mathrm{Concat}\left(\mathbf{o}^g_1, \mathbf{o}^g_2\right), \\ \mathbf{z}^g_2 &= \mathrm{Concat}\left(\mathbf{o}^g_2, \mathbf{o}^g_1\right). \end{aligned} \tag{28}$$

The ordering identifies the target and counterpart roles. CTRE and the KTP Policy Actor each share parameters across tasks but use separate parameter sets. Reversing the concatenation order allows the shared actor to produce task-specific actions. Concatenating two 64-dimensional representations gives

$$\mathbf{z}^g_t \in \mathbb{R}^{128}. \tag{29}$$

The KTP Policy Actor $f_\pi(\cdot;\theta)$ is a two-layer MLP with a 64-dimensional ReLU hidden layer and five output logits. Given $\mathbf{z}_t^g$, it defines

$$\pi_\theta\left(a_t^g \mid \mathbf{z}_t^g\right) = \text{Categorical}(\text{logits} = f_\pi(\mathbf{z}_t^g;\theta))\,.$$

The sampled action is a KTP multiplier from Section 3.2.2. Parameter sharing allows one actor to serve both tasks, and the ordered inputs enable task-specific decisions about the LSL selection probability.

The critic uses the same ordered task-pair representations as the actor, with a separate parameter set ϕ. It produces a task-conditioned value estimate $V_\phi(\mathbf{z}_t^g)$ for each task, and their mean defines the joint-state value:

$$V_\phi(\mathbf{S}^g) = \frac{1}{2}\sum_{t=1}^{2} V_\phi\left(\mathbf{z}_t^g\right). \tag{30}$$

The shared critic and mean aggregation preserve task symmetry and provide a scalar baseline for the joint PPO transition.

B.4 Policy Optimization

Each generation produces a joint action $\mathbf{a}^g = (a_1^g, a_2^g)$ and mean reward $R^g = (r_1^g + r_2^g)/2$, yielding a transition $(\mathbf{S}^g, \mathbf{a}^g, R^g, \mathbf{S}^{g+1})$. Transitions are collected over $H = 10$ generations. Let $\widehat{G}^g$ and $\widehat{A}^g$ denote the estimated returns and advantages, respectively. The actor uses the clipped PPO objective defined in the main paper, and the critic minimizes

$$L_V(\phi) = \mathbb{E}_g\left[\left(V_\phi(\mathbf{S}^g) - \widehat{G}^g\right)^2\right]. \tag{31}$$

The actor and critic are updated at this interval during training and remain fixed during testing.

C Evolutionary-Environment Details

C.1 Adaptive Differential Evolution

Equation 13 defines parameter sampling and success-history memory updates, and Section 3.4.2 specifies current-to-best/1/bin mutation. This appendix describes crossover and offspring accounting.

After generating mutant vector $\mathbf{v}_i$, binomial crossover produces trial vector $\mathbf{u}_i$:

$$u_{i,d} = \begin{cases} v_{i,d}, & \zeta_{i,d} \le CR_i \text{ or } d = j_{\text{rand}}, \\ x_{i,d}, & \text{otherwise} \end{cases},$$

where $\zeta_{i,d} \sim U(0,1)$, CR_i is the crossover rate, and j_{rand} is a uniformly sampled dimension that ensures the trial vector inherits at least one component from the mutant vector.

The sampled F_i and CR_i are repaired to satisfy their bounds before offspring generation. Both values are stored as offspring metadata, and only those associated with surviving adaptive DE offspring enter S_F and S_{CR}.

C.2 KTP-Guided Local Offspring Generation

For each parent i of task t, the environment first samples

$$\xi_i \sim U(0,1).$$

KTP then selects the operator that generates the local offspring. LSL is selected if $\xi_i < \widehat{\text{KTP}}_t^g$ and both required LPCA models are valid; adaptive DE is selected otherwise:

$$q_{t,i}^{g} = \begin{cases} q_{t,i}^{\mathrm{LSL}}, & \xi_i < \widehat{\mathrm{KTP}}_t^{\,g} \text{ and LSL is valid}, \\ q_{t,i}^{\mathrm{aDE}}, & \text{otherwise} \end{cases},$$

where $q_{t,i}^{\mathrm{LSL}}$ and $q_{t,i}^{\mathrm{aDE}}$ denote offspring generated by LSL and adaptive DE, respectively. Only the selected operator is executed, so each parent produces exactly one local offspring. An offspring generated by adaptive DE retains its associated F_i and CR_i values as metadata.

For task t, the resulting local offspring set is

$$Q_t^g = \{q_{t,1}^g, q_{t,2}^g, \ldots, q_{t,N}^g\}.$$

The executed probability $\widehat{\mathrm{KTP}}_t^{\,g}$ governs LSL selection, while GSL generates one additional offspring per task independently of this probability.

C.3 Global Transfer, Environmental Selection, and Feedback Statistics

For each task and generation, GSL generates one offspring $q_{t,\mathrm{GSL}}^g$ in addition to the local offspring set Q_t^g. Parents and offspring are combined as

$$R_t^g = P_t^g \cup Q_t^g \cup \{q_{t,\mathrm{GSL}}^g\}.$$

NSGA-II environmental selection retains N individuals to form the next population:

$$P_t^{g+1} = \mathrm{Select}(R_t^g, N).$$

After environmental selection, survival statistics are recorded separately for LSL, GSL, and adaptive DE offspring. LSL survival determines state component s_3 and the reward term $\mathrm{SR}_{\mathrm{LSL},t}^g$; adaptive DE survival determines s_4 and $\mathrm{SR}_{\mathrm{aDE},t}^g$. The F_i and CR_i values associated with surviving adaptive DE offspring update $\mu_{F,t}$ and $\mu_{CR,t}$.

Population and survival statistics determine the next task state and training reward. During testing, the frozen policy continues to use updated states, and adaptive DE continues to adapt its parameters using surviving offspring.

D Experimental Details

D.1 Construction of the Training Set

Training instances are derived from the 10 two-task CEC21-MTMO-CPLX problems, which differ in Pareto-set geometry, variable transformations, and inter-task relatedness. Using the benchmark organization in MToP (Li et al., 2026), we perturb each official shift vector and rotation matrix to generate multiple instances per problem template:

$$\mathrm{shift}_{\mathrm{new}} = \mathrm{shift}_{\mathrm{base}} + \mathrm{noise}, \tag{32}$$

$$\mathrm{rotation}_{\mathrm{new}} = \mathrm{small_perturb}\left(\mathrm{rotation}_{\mathrm{base}}\right). \tag{33}$$

Here, noise denotes a proportional shift perturbation, and $\mathrm{small_perturb}(\cdot)$ applies a small perturbation to the base rotation matrix. We generate 30 instances per problem, yielding 300 training instances, each used for one episode. The augmentation preserves the task types, objective templates, and Pareto-front shapes of the base problems. The perturbation strengths are

$$\begin{aligned} \texttt{shift_noise_ratio} &= 0.10, \\ \texttt{rotation_strength} &= 0.08. \end{aligned} \tag{34}$$

Each episode comprises one complete two-task evolutionary run. The actor and critic are trained using rewards that combine population quality improvement with the relative survival advantage of LSL over adaptive DE offspring.

D.2 Performance Metrics

Inverted generational distance (IGD) evaluates the quality of the final Pareto-front approximation. Let PF_t^* denote the reference Pareto-front samples for task t, and let A_t denote the final nondominated solution set. IGD is defined as

$$\mathrm{IGD}(A_t, PF_t^*) = \frac{1}{|PF_t^*|} \sum_{\mathbf{y} \in PF_t^*} \min_{\mathbf{x} \in A_t} \|\mathbf{F}_t(\mathbf{x}) - \mathbf{y}\|_2 \,. \tag{35}$$

Lower IGD indicates better approximation of the reference front in terms of proximity and coverage. Task-level IGD is the primary statistic. For the supplementary problem-level analysis, we average the IGD values of the two tasks:

$$\mathrm{IGD}_p = \frac{1}{2} \left(\mathrm{IGD}_{p,1} + \mathrm{IGD}_{p,2} \right), \tag{36}$$

where $\mathrm{IGD}_{p,1}$ and $\mathrm{IGD}_{p,2}$ denote the final IGDs of the two tasks in problem p, respectively.

Hypervolume (HV) complements IGD by measuring the dominated objective-space area relative to a reference point, with higher values indicating better performance. We report normalized HV using a common reference point for all algorithms and runs on each task. For task t, define

$$\mathbf{r}_t = 1.1 \times \max \left(PF_t^*, \bigcup_{a \in \mathcal{A}} \bigcup_{j=1}^{30} A_{t,a}^{(j)} \right), \tag{37}$$

where $\mathcal{A}$ contains RL-EMT-MSKT, EMT-MSKT, MMTO-ETA, MMTO-TGT, KR-MTEA, MTDE-MKTA, MTEA-DCK, and MTEA-D-DN, and $A_{t,a}^{(j)}$ denotes the final nondominated objective-vector set obtained by algorithm a for task t in the j-th independent run. Visualization-only methods are excluded from $\mathcal{A}$. The maximum is taken componentwise over the reference-front samples and the final nondominated objective vectors from all included algorithms and runs. The resulting reference point $\mathbf{r}_t$ is shared across algorithms for task t. Denoting the dominated hypervolume relative to $\mathbf{r}_t$ by $\mathrm{HV}(A_t, \mathbf{r}_t)$, we compute normalized HV as

$$\mathrm{HV}_{\mathrm{norm}}(A_t) = \frac{\mathrm{HV}(A_t, \mathbf{r}_t)}{r_{t,1} r_{t,2}}. \tag{38}$$

Task-level IGD and normalized HV across the 20 tasks provide the primary quantitative evidence; problem-level IGD provides a supplementary analysis.

D.3 Supplementary Implementation Settings

The key experimental settings, including the evaluation budget, population size, KTP policy configuration, and adaptive DE update rate, are reported in Table 2. Table 5 lists the additional augmentation and controller-implementation settings required to reproduce RL-EMT-MSKT.

E Complete Task-Level Results

E.1 Comparison with the Original EMT-MSKT

We compare RL-EMT-MSKT and EMT-MSKT across 30 independent runs on all 20 tasks. Task-level IGD comparisons are conducted via two-sided Mann–Whitney U tests with Holm correction, as summarized in Table 6. The corresponding IGD mean and standard deviation values are provided in

Table 5: Supplementary implementation and augmentation settings for RL-EMT-MSKT.

Parameter	Value
Augmented instances per official problem	30
Shift perturbation ratio	0.10
Rotation perturbation strength	0.08
GSL offspring per task and generation	1
Actor input dimension	128
Discount factor γ	0.99

Table 6: IGD comparisons between RL-EMT-MSKT and EMT-MSKT.

	Task 1		Task 2	
Problem	p_{Holm}	Result	p_{Holm}	Result
P1	0.7581	$=$	< 0.001	$+$
P2	0.1213	$=$	0.0692	$=$
P3	< 0.001	$+$	< 0.001	$+$
P4	< 0.001	$+$	< 0.001	$+$
P5	0.0383	$+$	0.0373	$+$
P6	< 0.001	$+$	< 0.001	$+$
P7	0.8187	$=$	0.0338	$-$
P8	0.3164	$=$	< 0.001	$+$
P9	< 0.001	$+$	< 0.001	$+$
P10	< 0.001	$+$	< 0.001	$+$
Overall $+/-/=$	**14/1/5**			

Note: Each algorithm was evaluated over 30 independent runs. The reported values are Holm-adjusted two-sided p-values. The significance level is also set to 0.05. $+/-/=$ indicates that the RL-EMT-MSKT performs better, worse, and equally to the EMT-MSKT.

Table 3. Meanwhile, normalized HV is statistically evaluated through two-sided Mann–Whitney U tests at a significance level of $\alpha = 0.05$. The mean values, standard deviations, and overall statistical results of normalized HV are presented in Table 7.

Relative to EMT-MSKT, RL-EMT-MSKT obtains a lower mean IGD on 15 out of the 20 tasks. The IGD comparisons yield 14 significant wins, one significant loss, and five nonsignificant differences. In terms of normalized HV, RL-EMT-MSKT achieves a higher mean on 14 tasks, with the corresponding comparisons returning 14 significant wins, no significant losses, and six nonsignificant differences.

Taken together, the IGD and normalized-HV results show that RL-EMT-MSKT improved both reference-front approximation and dominated objective-space coverage on many evaluated tasks.

E.2 Statistical Comparison with Other Multiobjective Multitask Algorithms

We compare RL-EMT-MSKT with six MO-MTO methods: MMTO-ETA, which combines ensemble transfer with adaptive resource allocation (Zhang et al., 2026a); MMTO-TGT, which uses topology-guided knowledge transfer (Zhang et al., 2026b); KR-MTEA, which uses knowledge reuse (Cui et al., 2023); MTDE-MKTA, which combines multiple knowledge types with transfer adaptation (Li & Gong, 2025); MTEA-DCK, which uses diversity- and convergence-oriented transfer (Li et al., 2025); and MTEA-D-DN, which uses neighborhoods to support knowledge transfer (Wang et al., 2023).

All methods were evaluated on the same official test problems with identical evaluation budgets and task-specific reference data. Each algorithm, including RL-EMT-MSKT, EMT-MSKT and the six baseline methods, was independently executed for 30 runs on all test tasks.

Table 3 in the main paper presents the task-level IGD results and pairwise statistical outcomes. Table 7 lists the normalized-HV means, standard deviations, and aggregate statistical results. All pairwise comparisons adopt two-sided Mann–Whitney U tests at $\alpha = 0.05$.

From the perspective of RL-EMT-MSKT, the counts of significant wins, significant losses, and comparisons with no significant difference against MMTO-ETA, MMTO-TGT, KR-MTEA, MTDE-MKTA, MTEA-DCK, and MTEA-D-DN are $13/2/5$, $10/0/10$, $18/2/0$, $13/4/3$, $13/3/4$, and $10/1/9$ for IGD, respectively. The corresponding normalized HV counts are $13/2/5$, $12/0/8$,

Table 7: HV comparisons (mean $\pm$ standard deviation) on CEC21-MTMO-CPLX.

Task	RL-EMT-MSKT	EMT-MSKT	MMTO-ETA	MMTO-TGT	KR-MTEA	MTDE-MKTA	MTEA-DCK	MTEA-D-DN
P1-T1	$0.9991 \pm 2.62 \times 10^{-3}$	$\mathbf{0.9999 \pm 8.02 \times 10^{-5}}$	$\mathbf{0.9999 \pm 1.64 \times 10^{-4}}$	$\mathbf{0.9999 \pm 1.03 \times 10^{-4}}$	0.8106 ± 0.2151	$0.9996 \pm 1.90 \times 10^{-4}$	$0.9998 \pm 1.78 \times 10^{-4}$	0.9911 ± 0.0142
P1-T2	$\mathbf{0.9999 \pm 2.46 \times 10^{-5}}$	$0.9998 \pm 6.36 \times 10^{-5}$	$0.9997 \pm 1.64 \times 10^{-4}$	$\mathbf{0.9999 \pm 6.63 \times 10^{-5}}$	0.6135 ± 0.1068	$0.9994 \pm 2.91 \times 10^{-4}$	$0.9996 \pm 2.13 \times 10^{-4}$	$0.9961 \pm 4.39 \times 10^{-3}$
P2-T1	0.9705 ± 0.0206	$\mathbf{0.9829 \pm 5.04 \times 10^{-3}}$	0.9744 ± 0.0216	0.9613 ± 0.0183	$0.9424 \pm 7.88 \times 10^{-3}$	0.9234 ± 0.0104	0.9527 ± 0.0218	0.8832 ± 0.1898
P2-T2	0.9746 ± 0.0214	$\mathbf{0.9850 \pm 3.60 \times 10^{-3}}$	0.9714 ± 0.0226	0.9594 ± 0.0156	0.8725 ± 0.0516	0.9207 ± 0.0144	0.9592 ± 0.0189	0.8826 ± 0.1901
P3-T1	$\mathbf{0.9989 \pm 2.02 \times 10^{-3}}$	0.7869 ± 0.0527	0.7733 ± 0.0560	0.7983 ± 0.1108	0.6261 ± 0.0858	0.7292 ± 0.0611	0.7720 ± 0.0773	0.8691 ± 0.0557
P3-T2	$\mathbf{0.9591 \pm 0.0457}$	0.6028 ± 0.0509	0.5931 ± 0.0412	0.6229 ± 0.0796	0.2802 ± 0.0339	0.6222 ± 0.0628	0.6557 ± 0.0649	0.6701 ± 0.0644
P4-T1	$\mathbf{0.9033 \pm 0.0255}$	0.7845 ± 0.0558	0.7845 ± 0.0499	0.8812 ± 0.0252	0.8946 ± 0.0308	0.8930 ± 0.0273	0.7708 ± 0.0642	0.4796 ± 0.1581
P4-T2	$\mathbf{0.8666 \pm 0.0381}$	0.7508 ± 0.0650	0.7253 ± 0.0792	0.8244 ± 0.0458	0.7067 ± 0.0106	0.8387 ± 0.0295	0.7019 ± 0.0592	0.4599 ± 0.1409
P5-T1	0.9435 ± 0.0329	0.8623 ± 0.1050	0.7343 ± 0.1390	0.8259 ± 0.1932	$\mathbf{0.9830 \pm 0.0307}$	0.9141 ± 0.0901	0.8769 ± 0.1131	0.8176 ± 0.1124
P5-T2	0.8749 ± 0.1462	$\mathbf{0.9730 \pm 0.0313}$	0.8550 ± 0.1587	0.9618 ± 0.0921	0.9644 ± 0.0938	0.7336 ± 0.2082	0.8687 ± 0.1496	0.5727 ± 0.2540
P6-T1	$\mathbf{0.9725 \pm 0.0106}$	0.9387 ± 0.0308	0.9177 ± 0.0207	0.9280 ± 0.0373	0.9649 ± 0.0129	0.8635 ± 0.0679	0.9286 ± 0.0283	0.7022 ± 0.1280
P6-T2	$\mathbf{0.9894 \pm 5.53 \times 10^{-3}}$	$0.9636 \pm 7.15 \times 10^{-3}$	0.9653 ± 0.0134	$0.9757 \pm 6.95 \times 10^{-3}$	0.4702 ± 0.0202	0.9507 ± 0.0298	$0.9768 \pm 6.40 \times 10^{-3}$	0.8618 ± 0.0886
P7-T1	0.8780 ± 0.0846	0.8817 ± 0.0499	0.9839 ± 0.0107	0.9232 ± 0.0616	$\mathbf{0.9952 \pm 6.22 \times 10^{-3}}$	0.8370 ± 0.1055	0.8249 ± 0.1769	0.9841 ± 0.0378
P7-T2	0.9454 ± 0.0126	0.9498 ± 0.0206	$0.9880 \pm 7.30 \times 10^{-3}$	0.9514 ± 0.0152	0.3239 ± 0.0625	0.9330 ± 0.0122	0.8882 ± 0.0294	$\mathbf{0.9991 \pm 1.49 \times 10^{-5}}$
P8-T1	$\mathbf{0.9985 \pm 1.41 \times 10^{-3}}$	0.9730 ± 0.0317	0.9941 ± 0.0142	$0.9944 \pm 8.08 \times 10^{-3}$	0.9901 ± 0.0165	0.8649 ± 0.2247	0.9802 ± 0.0352	0.9711 ± 0.0291
P8-T2	$\mathbf{0.9990 \pm 1.06 \times 10^{-3}}$	0.7005 ± 0.0832	0.6274 ± 0.0940	0.7908 ± 0.0686	0.5045 ± 0.0427	0.5355 ± 0.1969	0.8443 ± 0.0746	$0.9981 \pm 1.42 \times 10^{-3}$
P9-T1	$\mathbf{0.9551 \pm 9.92 \times 10^{-3}}$	0.7545 ± 0.0457	0.7312 ± 0.0604	0.7628 ± 0.1392	0.4302 ± 0.0483	0.6915 ± 0.0411	0.6669 ± 0.0362	0.9096 ± 0.0223
P9-T2	$\mathbf{1.0000 \pm 9.85 \times 10^{-7}}$	$\mathbf{1.0000 \pm 9.16 \times 10^{-7}}$	$\mathbf{1.0000 \pm 6.46 \times 10^{-7}}$	$\mathbf{1.0000 \pm 5.31 \times 10^{-8}}$	0.5720 ± 0.1374	$\mathbf{1.0000 \pm 3.24 \times 10^{-6}}$	$\mathbf{1.0000 \pm 6.99 \times 10^{-8}}$	$\mathbf{1.0000 \pm 3.28 \times 10^{-6}}$
P10-T1	$\mathbf{0.9703 \pm 0.0101}$	0.8718 ± 0.0837	0.8348 ± 0.0679	0.9416 ± 0.0250	0.8949 ± 0.0983	0.9569 ± 0.0211	0.8552 ± 0.0958	0.6951 ± 0.1393
P10-T2	$\mathbf{0.9997 \pm 1.75 \times 10^{-4}}$	$0.9977 \pm 1.23 \times 10^{-3}$	$0.9960 \pm 1.81 \times 10^{-3}$	$0.9995 \pm 3.11 \times 10^{-4}$	0.6664 ± 0.0691	$0.9986 \pm 9.24 \times 10^{-4}$	$0.9989 \pm 4.51 \times 10^{-4}$	$0.9923 \pm 4.36 \times 10^{-3}$
Compared algorithm vs. RL (+/ − / =)	–	0/14/6	2/13/5	0/12/8	3/14/3	0/14/6	0/14/6	2/17/1

Note: Higher normalized HV is better. Bold indicates the highest reported mean for each task, including ties at the displayed precision. In the final row, "+", "−", and "=" count tasks on which the column algorithm performs significantly better, significantly worse, or with no significant difference relative to RL-EMT-MSKT, respectively.

$14/3/3$, $14/0/6$, $14/0/6$, and $17/2/1$. RL-EMT-MSKT therefore achieves at least 10 significant IGD wins and 12 significant normalized HV wins against each comparator.

In the comparison restricted to RL-EMT-MSKT and the six additional methods, RL-EMT-MSKT attains the best mean on 13 of 20 tasks for each metric. Including EMT-MSKT reduces the number of lowest-mean IGD results to 11, as reported in the unified main-paper comparison.

F Additional Ablation Visualization

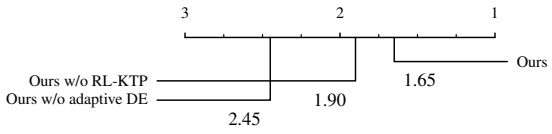


Figure 3: Average IGD ranks of the ablation variants on CEC21-MTMO-CPLX. Lower ranks indicate better performance.

G Additional Experimental Visualizations

The following figures visualize Pareto-front approximations, convergence, and learned KTP trajectories across all CEC21-MTMO-CPLX tasks.

G.1 Interpretation of Pareto-Front Quality

Figure 4 shows the best observed Pareto-front approximations over 30 runs for each task. The visualizations highlight RL-EMT-MSKT on P1-T2, both tasks of P3, P4, P6, and P10, as well as P8-T2 and P9-T1, together with the competitive performance of other methods on P2, P5, and P7. These best-run examples provide qualitative context for the results across runs in Table 3 and Appendix E.

G.2 Interpretation of Convergence and Transfer Behavior

The logarithmic IGD trajectories in Figure 5 show rapid early improvement on some tasks and continued later improvement on others. On P1-T2, P9-T1, and P10-T2, RL-EMT-MSKT continues to improve after progress slows for several baselines. Other methods achieve comparable or better performance during parts of the search on P2, P5, and P7, indicating task-dependent benefits.

Figure 6 shows that KTP generally decreases from 0.5 toward 0.05, assigning higher LSL selection probabilities early in the search and lower probabilities later. The decrease is rapid on P2, P4, P6, P7, and P8, and is more gradual or differs between tasks on P1, P3, P5, P9, and P10. These

trajectories demonstrate task-specific KTP schedules, while the roles of individual state features remain unresolved. The fixed-KTP ablation provides complementary evidence of the performance contribution of learned KTP control.

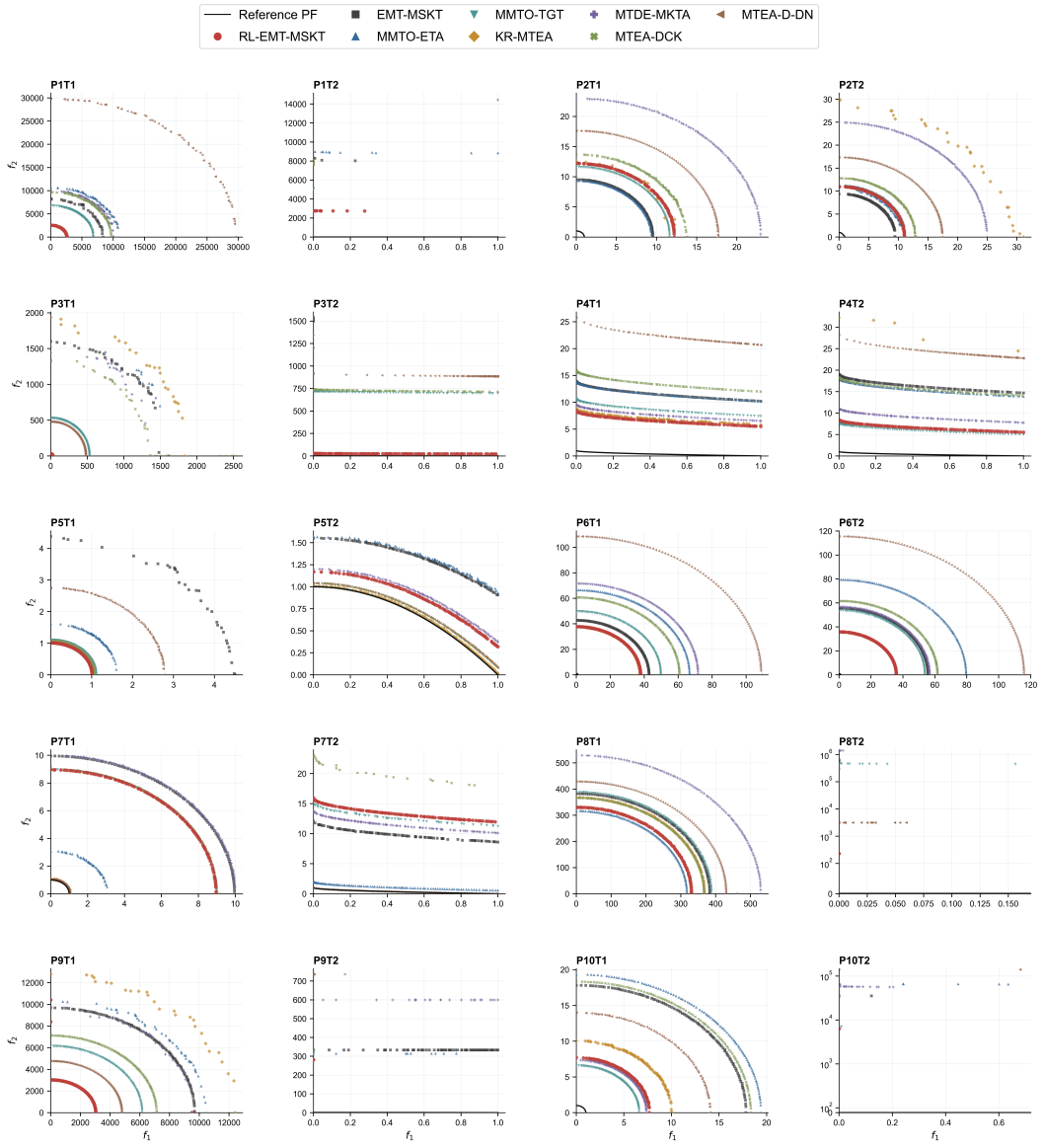


Figure 4: PF approximations obtained from the best-IGD run for all CEC21-MTMO-CPLX instances. Curves of off-scale comparison algorithms are omitted in some panels for readability. MO-MTEA-PAE is included only for visualization.

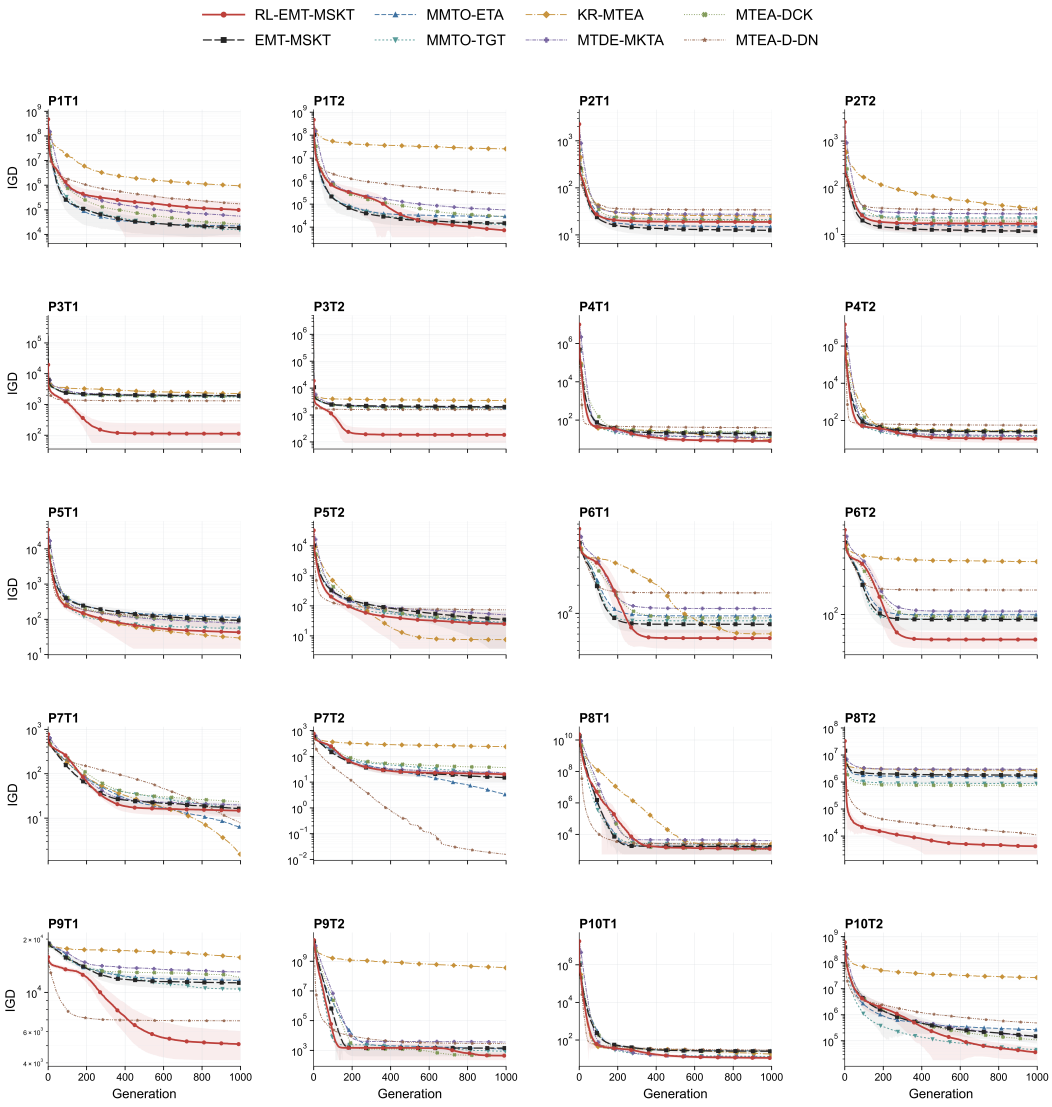


Figure 5: IGD convergence curves of all CEC21-MTMO-CPLX instances. IGD values have undergone the logarithmic transformation. Shaded bands indicate variability across runs.

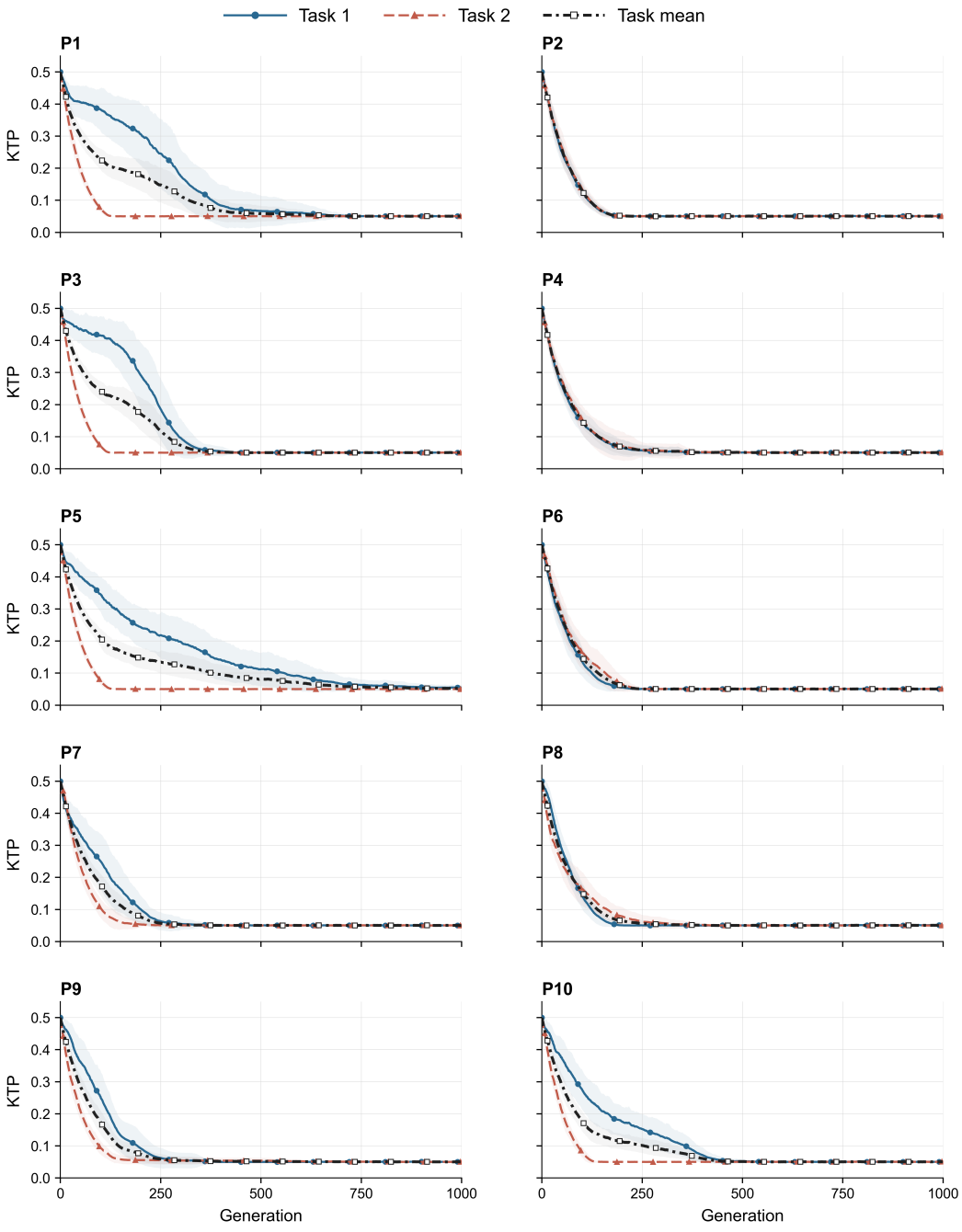


Figure 6: KTP trajectories of all CEC21-MTMO-CPLX instances. The black curves are the mean KTP of two tasks. Shaded bands indicate variability across runs.