Supplementary Materials

A Group-based Many-task Collaborative Optimization Framework for Evolutionary Robots Design

I. Symbols and Notations

This section provides a comprehensive list of symbols and notations used throughout the paper. Each symbol is accompanied by a short description to ensure that the symbol table is clear and can be easily found and referred to by the reader. Table S.I and Table S.II give notations and descriptions of the proposed framework and experiments, respectively.

TABLE S.I
NOTATIONS AND DESCRIPTIONS FOR PROPOSED FRAMEWORK.

Notations	Notation descriptions
n	Total number of tasks
$T(\tau_1, \tau_2, \ldots)$	Set of tasks
A	Task archive
b_i	Task descriptor for task τ_i
k	Number of task groups
$G(G_1, G_2, \ldots)$	Set of task groups
n_g	Expected number of tasks in each group
$cand_{opt}$	Most frequently selected crossover operator in a group
r	Probability for selecting crossover operators randomly
x_i^{new}, x_i^{last}	Current solution of τ_i, solution of τ_i in last evaluation
p_1, p_2	Solution of task τ_1, τ_2
OLT, PTs, UTs	Opinion leader task, progressive tasks, underperforming tasks
HI	Historical knowledge
δ	Small value used to identify underperforming tasks
x_u, x_s	Solutions set of UTs, PTs in the current group
x_{best}, x_{old}	Best solution in the current group, solution in HI
c_1, c_2, c_3	Hyperparameters that regulate the influences of OLT, PTs, and HI
r_1, r_2, r_3	Random generated values between 0 and 1
$inf_{OLT/PTs/HI}$	Knowledge from $OLT/PTs/HI$ for UTs
$S(s_1, s_2, \ldots)$	Set of knowledge combination strategies
B_{s_j}	Beta distribution parameters for each knowledge combination strategy
fit_{s_j}	Total fitness improvement of solutions
$times_{s_j}$	Number of times a strategy was launched after intra-group knowledge reunion stage
eff_{s_j}	Measurement index for strategy effectiveness
σ_G	Standard deviation of the Gaussian distribution
α, β	Beta distribution parameters for Thompson sampling
μ_{TS}, σ_{TS}	Mean and standard deviation of beta distribution for Thompson sampling

II. Illustrations for the Proposed Framework AGCO

In this section, we will supplement some algorithmic flow diagrams for the proposed framework AGCO.

A. Group Construction

Fig. S1 provides a concrete example to illustrate the process of group construction in AGCO. It is assumed that there are n manipulator tasks. Each task corresponds to a task

TABLE S.II
NOTATIONS AND DESCRIPTIONS FOR EXPERIMENTS.

Notations	Notation descriptions
d	Number of connecting joints of the manipulator
L	Length of the manipulator
$l_1, l_2, \ldots, l_d$	Lengths of each connecting joints of the manipulator
L_τ	Length of each connecting rod after normalization
γ	Vector of the angles of the joints
$\gamma_1, \gamma_2, \ldots, \gamma_d$	Angles of the joints of the manipulator
γ^{max}	Joint limit for the manipulator
P_d	End position of the manipulator
Tar	Target point for the manipulator
t	The total time hexapod moves in seconds
T	Total number of time steps during the evaluation
$\theta_i(j)$	Angle of the i-th joint at time step j
P_0	Initial position of the hexapod
P_f	Final position of the hexapod
TD	Transfer degree
SR	Success rate
$Reunion_{eva}$	Number of evaluations in intra-group knowledge reunion stage
$Total_{eva}$	Total number of evaluations in both inter-group knowledge separation and intra-group knowledge reunion stages
$Time_{suc}$	Number of times that the task in intra-group knowledge reunion stage successfully acquires the generated knowledge

descriptor, represented as $b_1, b_2, \ldots, b_n$. The task descriptor for a manipulator task is determined by a two-dimensional vector $[L, \gamma^{max}]$, which respectively represent the length of the manipulator and the maximum deflection angle limit of the joints. We utilize Euclidean distance between task descriptors to measure the similarity between tasks. Tasks are then grouped by using k-means based on the similarity of their task descriptors. Each group is a collection of manipulator tasks that share similar characteristics, aimed at optimizing control strategies efficiently by processing tasks that are alike.

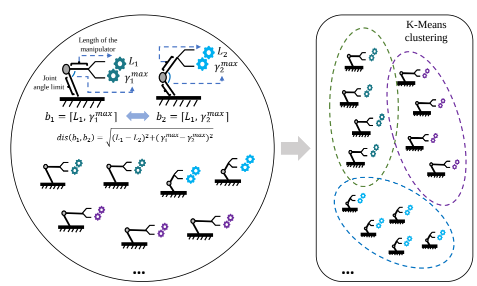


Fig. S1. A concrete example to illustrate the group construction process in AGCO.

B. Inter-group Knowledge Separation

The inter-group knowledge separation stage in AGCO facilitates knowledge transfer between tasks from different groups.

Fig. S2 provides an illustration of this process. The detailed steps described in the figure are as follows. Firstly, Two tasks τ_1 and τ_2 are randomly selected from all the tasks, where τ_1 belongs to group G_1 and τ_2 belongs to group G_2 (τ_1 and τ_2 may also belong to the same group). The solutions p_1 and p_2 corresponding to τ_1 and τ_2 are obtained. Next, a task similar to either τ_1 or τ_2 is selected as τ. Then, the task-group G_{sub} that task τ belongs to is identified (τ may also belong to G_1 or G_2), and the crossover operator most frequently selected by OLT and PTs from G_{sub} is recorded. Notably, there is also a certain probability to randomly select one crossover operator in order to increase randomness. Finally, a new solution x is generated by performing the crossover operation on p_1 and p_2 and x is evaluated on task τ. The above process will be repeated continuously until the current batch is complete.

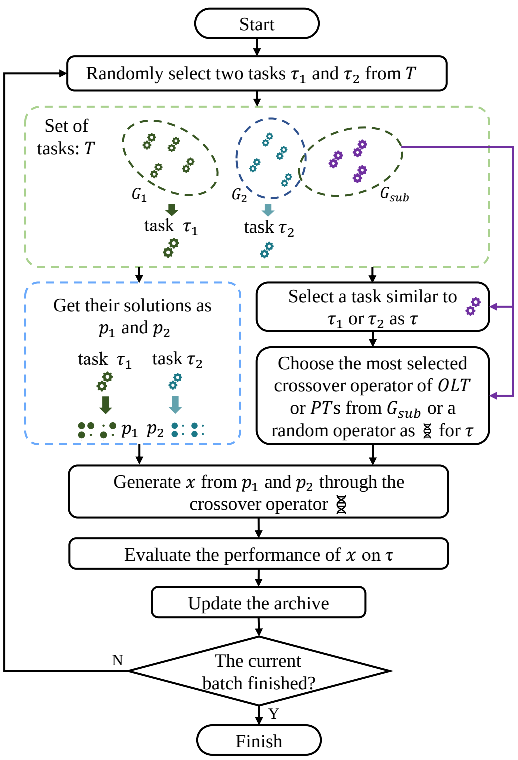


Fig. S2. An illustration of the inter-group knowledge separation in AGCO.

C. Intra-group Knowledge Reunion

The intra-group knowledge reunion process in AGCO aims to optimize task solutions through the selection of different strategies. Fig. S3 illustrates the steps of this process. The specific steps are as follows. First, traverse each task group and select underperforming tasks UTs from the current group G_{sub}. Next, select an appropriate strategy s_j through Thompson sampling for each task τ in UTs; Then, update the solution of task τ according to the selected strategy s_j. If s_j belongs to ***Strategy 1***, find the historical knowledge HI of the task τ, and select OLT and PTs from the current group G_{sub}, then the solution of the current underperforming task will acquire knowledge from OLT, PTs and HI. Otherwise, the solution will be updated through a Gaussian variation. Finally, the updated solution will be evaluated, and the parameters of Thompson sampling will be updated according to whether the newly obtained solution is placed in the archive. This stage will finish once all underperforming tasks UTs in all groups have undergone the above process.

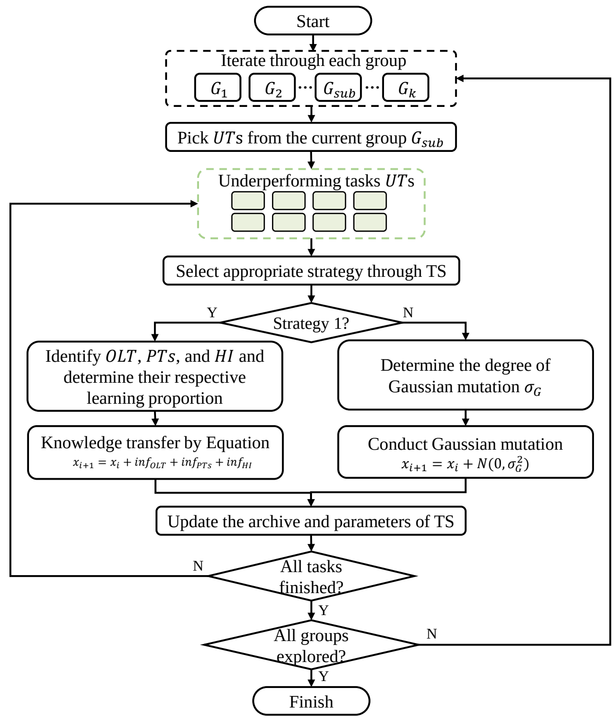


Fig. S3. An illustration of the intra-group knowledge reunion in AGCO.

III. Experimental Design Supplement

Table S.III shows the dimensions and task descriptors of the manipulator and the hexapod experiment and summarizes the differences in task characteristics between the two types of robot experiments. In the manipulator experiment, the parameter dimension is 30 or 50, and the task descriptor is the two-dimensional $[L, \gamma^{max}]$, representing the length of the manipulator and the maximum deflection angle. In the hexapod experiment, the parameter dimension is 36, and the task descriptor is the 12-dimensional variation in the length of each link. Notably, each dimension in the task descriptor vector is generated as a random number between 0 and 1.

TABLE S.III
The dimensions and task descriptors of the manipulator and the hexapod experiment.

	Manipulators experiment	Hexapods experiment
Dimension	d = 30 or 50	d = 36
Task descriptor	$[L, \gamma^{max}]$, 2-dimensional	Length variation of each link, 12-dimensional

IV. Result and Analysis

In this section, more experiments are conducted over more metrics to validate the effectiveness and generality of the proposed AGCO framework, while also confirming the rationality of the parameters involved.

A. Performance Comparisons on Quality Diversity Metrics

In this section, we report quality diversity metrics in our comparison with multitasking quality diversity algorithms (e.g., MME, MMKT). These metrics are common in MAP-Elites and have the following implications.

1) QD-score (QDS): The total sum of fitness across all solutions in the archive. The higher the QD-score, the better the quality and diversity of the solutions searched by the algorithm.
2) Max fitness (MAF): The overall fittest solution in the archive, which represents the ability of the algorithm to capture the global optimal solution.
3) Coverage (COV): The total number of solutions in the archive. Full coverage is achieved when a solution is found for each task in the archive.

1) The Manipulator Experiment: The results of AGCO, MMKT, and MME over quality diversity metrics (QDS, MAF, and COV) on manipulator experiments after 1 million evaluations are shown in Table S.IV. We can see that all algorithms achieved full coverage of the archive, by reporting a COV rate of 100%. Among all methods, AGCO obtained the highest QD-score, hence demonstrating its significance in searching for solutions with higher quality and diversity. Further, compared to MMKT, AGCO obtained a slightly worse performance in terms of MAF in solving a smaller number of tasks. However, as the number of tasks increases, AGCO gradually shows an advantage over MMKT, obtaining better MAF values. Overall, AGCO performs excellently in terms of QD-score and MAF as the number of tasks increases, demonstrating its effectiveness in achieving high-quality and diverse solutions when dealing with a large number of complex tasks.

TABLE S.IV
THE RESULTS OF AGCO, MMKT, AND MME OVER QUALITY DIVERSITY METRICS (QDS, MAF, AND COV) ON MANIPULATOR EXPERIMENTS AFTER 1 MILLION EVALUATIONS.

Tasks		AGCO			MMKT			MME		
		QDS	MAF	COV	QDS	MAF	COV	QDS	MAF	COV
2000	30	**-676**	-8.51×10^{-5}	100%	-682	$\mathbf{-6.43\times10^{-5}}$	100%	-686	-7.68×10^{-5}	100%
	50	**-680**	-9.82×10^{-5}	100%	-692	$\mathbf{-7.72\times10^{-5}}$	100%	-696	-6.81×10^{-5}	100%
5000	30	**-1695**	-9.56×10^{-5}	100%	-1715	$\mathbf{-9.42\times10^{-5}}$	100%	-1735	-1.04×10^{-4}	100%
	50	**-1700**	-1.14×10^{-4}	100%	-1760	$\mathbf{-9.36\times10^{-5}}$	100%	-1785	-1.06×10^{-4}	100%
10000	30	**-3380**	$\mathbf{-1.21\times10^{-4}}$	100%	-3450	-1.22×10^{-4}	100%	-3570	-1.44×10^{-4}	100%
	50	**-3400**	$\mathbf{-1.24\times10^{-4}}$	100%	-3590	-1.32×10^{-4}	100%	-3780	-2.75×10^{-4}	100%

TABLE S.V
THE RESULTS OF AGCO, MMKT, AND MME OVER MEAN FITNESS AND QUALITY DIVERSITY METRICS (QDS, MAF, AND COV) ON HEXPODS EXPERIMENTS AFTER 1 MILLION EVALUATIONS.

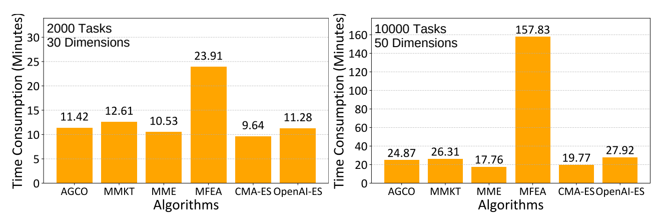
	Mean fitness	QD-Score (QDS)	Max fitness (MAF)	Coverage (COV)
AGCO	**0.810**	**1620**	**1.304**	100%
MMKT	0.722 (-)	1444	1.259	100%
MME	0.723 (-)	1446	1.198	100%

'≈', '+', and '-' indicate that the effect of the comparison algorithm is similar to, significantly better than, and significantly worse than AGCO, respectively, which are derived based on the Wilcoxon ranksum tests. The significance level is $\alpha = 0.05$.

2) The Hexapod Robot Experiment: The results of AGCO, MMKT, and MME over mean fitness and quality diversity metrics (QDS, MAF, and COV) on hexpods experiments are shown in Table S.V. It can be observed that AGCO significantly outperforms other algorithms in the metric of mean fitness, achieving a score of 0.810 compared to 0.722 and 0.723 for MMKT and MME, respectively. Additionally, it achieves the best results in quality diversity metrics (with QDS of 1620 and MAF of 1.304), confirming advantages of AGCO in complex problem-solving. Furthermore, all algorithms achieved a 100% coverage rate, demonstrating their capacity to comprehensively explore the solution space. Overall, AGCO is the most competitive in comparison to other algorithms.

B. Running Time Comparison of Algorithms

In this section, we conducted algorithm comparisons on time efficiency. Fig. S4 compares the time consumption (in minutes) of several comparison methods in the manipulator experiment (2000 tasks, 30 joints, and 10000 tasks, 50 joints). For instance, in the manipulator experiment with 2000 tasks, 30 joints, the execution time of AGCO (i.e., 11.42 minutes) is similar to MME (i.e., 10.53 minutes) and MMKT (i.e., 12.61 minutes), while its optimization performance is significantly better than the latter two. Similarly, the time consumption of these three algorithms is similar (i.e., 24.87, 17.76 and 26.31 minutes) in the manipulator experiment with 10,000 tasks and 50 joints. Besides, MFEA takes several times (i.e., 23.91 and 157.83 minutes, respectively, in the two manipulator experiments) longer than AGCO to solve the same problem, suggesting that traditional multitasking optimization algorithms like MFEA may not be very effective in solving a large number of tasks in this study. Overall, AGCO obtained more competitive performance in a relatively short period of time.



(a) (b)

Fig. S4. Comparison of the time consumption (in minutes) of AGCO, MMKT, MME, MFEA, CMA-ES, and OpenAI-ES. (a) 2000 tasks, 30 joints; (b) 10000 tasks, 50 joints.

Table S.VI compares the time consumption (in minutes) of several methods in the experiments of hexapod robots. As can be seen from the table, the time consumption of AGCO (i.e., 550.23 minutes) is similar to that of MME (i.e., 571.72 minutes) and MMKT (i.e., 538.40 minutes), with little difference in calculation complexity. Furthermore, for tasks with heavy computational demands, traditional single-task algorithms (e.g., CMA-ES and OpenAI-ES) exhibit disadvantages in terms of computation time which are less efficient compared to multitask algorithms.

C. Knowledge Transfer Analysis

In the stage of inter-group knowledge separation, AGCO adopts the same task exploration strategy as MME and

TABLE S.VI
COMPARISON OF THE TIME CONSUMPTION (IN MINUTES) OF AGCO, MMKT, MME, CMA-ES, AND OPENAI-ES ON HEXPOD ROBOT EXPERIMENTS.

	AGCO	MMKT	MME	CMA-ES	OpenAI-ES
Computation Time (Minutes)	550.23	538.40	571.72	779.37	841.18

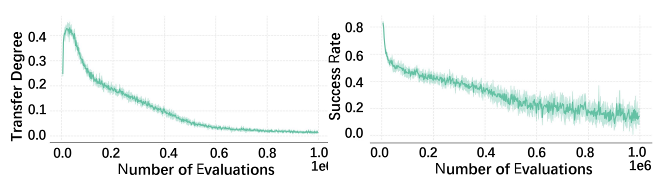
MMKT. Here, we explore why AGCO makes a difference in yielding promising results. We take the planar manipulator experiment with 50 joints and 5000 tasks as an example to analyze the knowledge generation and absorption between tasks in the same group. First of all, we define transfer degree (TD) and success rate (SR) as measures reflecting the effect of knowledge transfer, which are expressed as follows:

$$TD = \frac{Reunion_{eva}}{Total_{eva}}. \tag{1}$$

$$SR = \frac{Time_{suc}}{Reunion_{eva}}. \tag{2}$$

Specifically, $Total_{eva}$ refers to the sum of the number of evaluations in inter-group knowledge separation and intra-group knowledge reunion stages; $Reunion_{eva}$ refers to the evaluations in intra-group knowledge reunion; $Time_{suc}$ represents the number of times that the task in the intra-group knowledge reunion stage successfully acquires the generated knowledge. In particular, TD reflects the active degree of intra-group knowledge exchange, and SR reflects the probability of successful absorption of knowledge generated in the process of intra-group knowledge exchange. Meanwhile, it represents the efficiency of intra-group knowledge absorption. As shown in Fig. S5, we record TD and SR in each round of intra-group knowledge reunion and inter-group knowledge separation.

Fig. S5(a) shows how TD varies with the number of evaluations. At the beginning of optimization, the degree of knowledge transfer is relatively low. This is because AGCO may not initially generate solutions for most tasks, so knowledge could not be smoothly generated or acquired within groups. As the number of evaluations increases, a growing number of tasks are resolved, thereby amplifying TD quickly. As the optimization process progresses, tasks gradually reach a better value and stop converging, and fewer tasks are classified as underperforming tasks (UTs). Therefore, the transfer degree gradually decreases.



(a) (b)

Fig. S5. Knowledge generation and acquisition of AGCO. (a): Transfer degree (TD); (b): Success rate (SR). The shaded area shows the range of standard deviations for TD and SR in 30 independent repetitions of experiments.

Fig. S5(b) shows how SR varies with the number of evaluations. It is observed that knowledge from other tasks is more likely to be successfully acquired at the beginning. As shown in Fig. S5(a) and Fig. S5(b), the transfer degree remains above 20% and the success rate remains above 50% until the number of evaluations reaches 1.0E + 05. This indicates that in the early stages of optimization, high-quality knowledge can be actively exchanged between tasks within a group. As the optimization process progresses, the solutions of different tasks tend to converge and will not change significantly across generations. Therefore, despite the transfer degree, the success rate of knowledge transfer will also decrease.

D. Analysis on Adaptive Methods

In this section, we consider the validity of adaptive methods designed in AGCO. The effectiveness of the adaptive crossover operator selection method and the adaptive knowledge combination strategy are evaluated through experiments on manipulators and hexapods.

1) Adaptive Selection of Crossover Operator: We take the manipulator and hexapod robot experiments as examples to verify the effectiveness of the adaptive crossover operator selection method in the inter-group knowledge separation stage. The number of joints in the manipulator experiment is 50 and the number of tasks is 5000. Specifically, we construct the following three algorithms, each of which uses a solely fixed crossover operator, and the other components of the algorithm remain exactly the same as AGCO. The three algorithms are described as follows:

1) AGCO-D: it only uses the directional crossover operator in the inter-group knowledge separation stage.
2) AGCO-U: it only uses the uniform crossover operator in the inter-group knowledge separation stage.
3) AGCO-S: it only uses the SBX crossover operator in the inter-group knowledge separation stage.

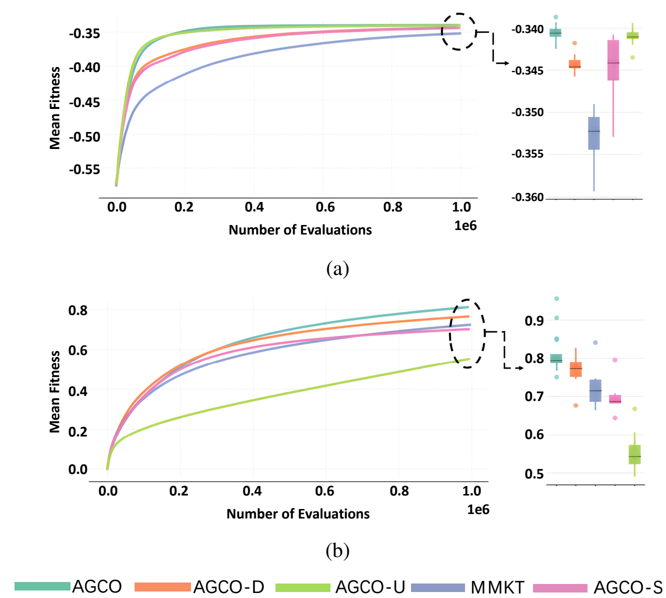


Fig. S6. Mean fitness curves obtained by AGCO, MMKT, AGCO-D, AGCO-U, and AGCO-S. (a): The manipulator experiment; (b): The hexapod robot experiment.

Fig. S6 plots the results of AGCO compared with AGCO-D, AGCO-U, AGCO-S, and MMKT. In the manipulator experiment, the performance of AGCO-U is comparable to AGCO and better than AGCO-D and AGCO-S. In the hexapod robot experiment, AGCO-D performs better than AGCO-S which performs similarly to MMKT and is far better than AGCO-U. The results show that the crossover operators suitable for different scenarios are different. It implies that adaptively selecting the most appropriate crossover operator for each task is critical and instrumental. This has been confirmed by the best results achieved by AGCO in both experiments.

2) Adaptive Knowledge Combination Strategy: We also adopt the manipulator and hexapod robot experiments to verify the effectiveness of the adaptive knowledge combination strategy selection method. In this study, AGCO is decomposed into five algorithms using a specified knowledge combination strategy, which is described as follows:

1) AGCO-H: the knowledge combination strategy is realized based on ***Strategy 1***. The parameters c_1 and c_2 are set to 0.5 and c_3 is set to 1.5, to make the knowledge mainly acquired from HI.
2) AGCO-E: the knowledge combination strategy is realized based on ***Strategy 1***. The parameters c_1 and c_2 are set to 0.9, and c_3 is set to 1.1. Knowledge from HI or other tasks in the same group has a similar weight.
3) AGCO-OLT&PTs: the knowledge combination strategy is realized based on ***Strategy 1***. The parameters c_1 and c_2 are set to 1.5, and c_3 is set to 0.5. The acquired knowledge mainly comes from other tasks in the same group.
4) AGCO-V1: the knowledge combination strategy is realized based on ***Strategy 2***, and only a small variation is made based on the current solution ($\mu = 0$, $\sigma_G = 0.04$).
5) AGCO-V2: the knowledge combination strategy is realized based on ***Strategy 2***, and a large variation is made based on the current solution ($\mu = 0$, $\sigma_G = 0.08$).

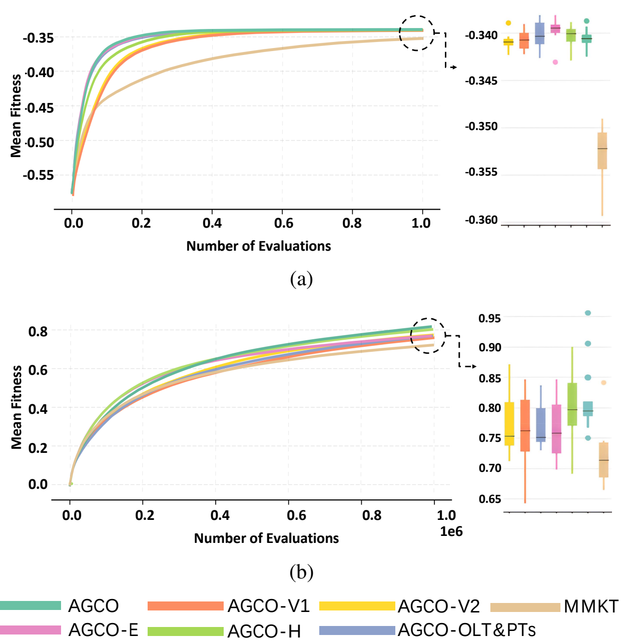


Fig. S7. Mean fitness curves obtained by AGCO, MMKT, AGCO-H, AGCO-E, AGCO-OLT&PTs, AGCO-V1, and AGCO-V2. (a): The manipulator experiment; (b): The hexapod robot experiment.

Fig. S7 plots the results of AGCO compared with AGCO-H, AGCO-E, AGCO-OLT&PTs, AGCO-V1, AGCO-V2, and MMKT. It can be seen from Fig. 7(a) that in the manipulator experiment, AGCO, AGCO-E, and AGCO-OLT&PTs obtained competitive convergence speeds that are superior to all other variants. However, in Fig. 7(b), AGCO and AGCO-H achieved better convergence performance after 1 million evaluations, significantly outperforming the other variants (e.g., AGCO-E, AGCO-OLT&PTs, AGCO-V1, and AGCO-V2). These results suggest that a specific knowledge combination strategy (i.e., AGCO-E, AGCO-OLT&PTs, and AGCO-H) effective in one scenario may not be effective in other scenarios. Therefore, adaptive knowledge combination strategies are crucial to enhance the generality of the algorithm in different task scenarios. Nevertheless, our proposed AGCO still obtained a competitive convergence speed against all of its variant algorithms in both manipulator and hexapod experiments. These results hence verify the efficacy of the adaptive knowledge combination strategy selection method in our proposed AGCO framework.

E. Sensitivity Analysis on Hyperparameter n_g

The hyperparameter n_g configures the number of tasks per group in K-means clustering which needs to be pre-tuned in our experiment. Generally, the choice of n_g is mainly influenced by the number of tasks and the similarity between tasks. When the number of tasks increases or the similarity between tasks is high, the value of n_g can be increased appropriately. We tested the effect of n_g values of 3, 5, 10, and 20 in the manipulator and hexapod experiments, respectively.

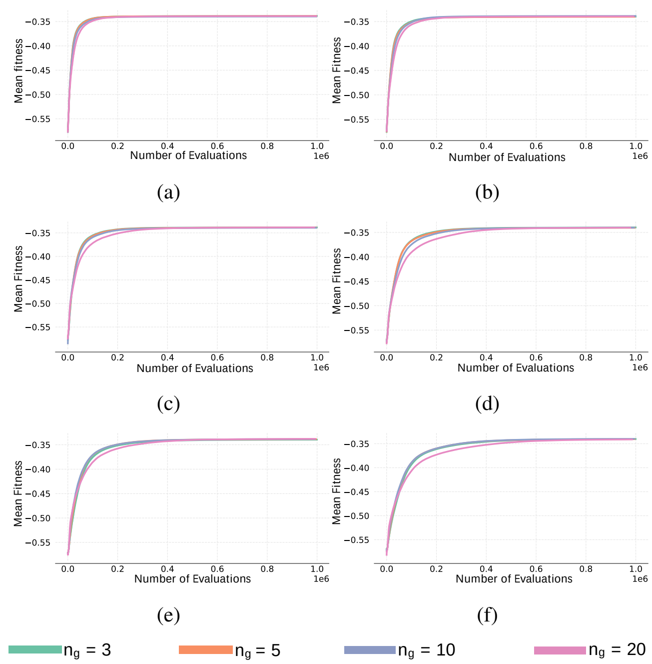


Fig. S8. The experimental results of manipulators under different n_g settings. (a), (b): Manipulators, 2000 tasks, 30 and 50 joints; (c), (d): Manipulators, 5000 tasks, 30 and 50 joints; (e), (f): Manipulators, 10000 tasks, 30 and 50 joints.

Fig. S8 and Fig. S9 show the experimental results of manipulators and hexapods under different n_g settings respectively. From the experimental results, we can observe from Fig. S8 that when n_g is too large (e.g., n_g = 20), the convergence speed

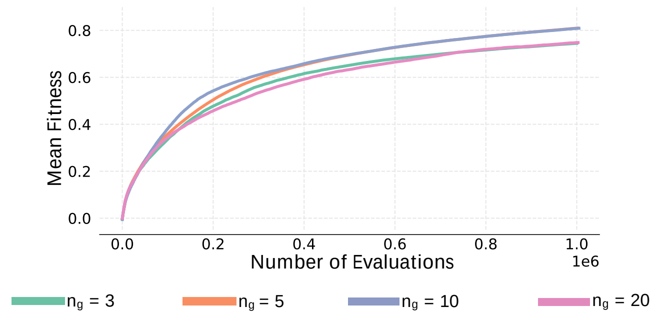


Fig. S9. The experimental results of hexapods under different n_g settings.

tends to decrease, which may be caused by the reduction of task similarity in the same group, which may lead to negative knowledge transfer. Additionally, as illustrated in Fig. S9, setting n_g too small (e.g., n_g = 3) results in a diminished optimization performance. This result can be attributed to the excessive number of groups, while the number of excellent tasks that each group can learn is reduced, thus compromising the effectiveness of knowledge transfer. Therefore, considering the above experimental results of manipulators and hexapods, the value of n_g is set to 5.