



ARTICLE

# A Distributed Dual-Network Meta-Adaptive Framework for Scalable and Privacy-Aware Multi-Agent Coordination

Atef Gharbi<sup>1</sup>, Mohamed Ayari<sup>2</sup>, Nasser Albalawi<sup>3</sup>, Ahmad Alshammari<sup>3</sup>, Nadhir Ben Halima<sup>4,\*</sup> and Zeineb Klai<sup>3</sup>

<sup>1</sup>Department of Information Systems, Faculty of Computing and Information Technology, Northern Border University, Rafha, Saudi Arabia

<sup>2</sup>Department of Information Technology, Faculty of Computing and Information Technology, Northern Border University, Rafha, Saudi Arabia

<sup>3</sup>Department of Computer Sciences, Faculty of Computing and Information Technology, Northern Border University, Rafha, Saudi Arabia

<sup>4</sup>Department of Information Technology, Community College of Qatar, Doha, Qatar

\*Corresponding Author: Nadhir Ben Halima. Email: nadhir.benhalima@ccq.edu.qa

Received: 02 November 2025; Accepted: 08 January 2026; Published: 12 March 2026

**ABSTRACT:** This paper presents Dual Adaptive Neural Topology (Dual ANT), a distributed dual-network meta-adaptive framework that enhances ant-colony-based multi-agent coordination with online introspection, adaptive parameter control, and privacy-preserving interactions. This approach improves standard Ant Colony Optimization (ACO) with two lightweight neural components: a forward network that estimates swarm efficiency in real time and an inverse network that converts these descriptors into parameter adaptations. To preserve the privacy of individual trajectories in shared pheromone maps, we introduce a locally differentially private pheromone update mechanism that adds calibrated noise to each agent's pheromone deposit while preserving the efficacy of the global pheromone signal. The resulting system enables agents to dynamically and autonomously adapt their coordination strategies under challenging and dynamic conditions, including varying obstacle layouts, uncertain target locations, and time-varying disturbances. Extensive simulations of large grid-based search tasks demonstrated that Dual ANT achieved faster convergence, higher robustness, and improved scalability compared to advanced baselines such as Multi-Strategy ACO and Hierarchical ACO. The meta-adaptive feedback loop compensates for the performance degradation caused by privacy noise and prevents premature stagnation by triggering Lévy flight exploration only when necessary.

**KEYWORDS:** Ant colony optimization; multi-agent systems; deep neural networks; meta-adaptive learning; Lévy flight; differential privacy; swarm intelligence

## 1 Introduction

Coordinating large populations of distributed agents is a major challenge in swarm intelligence, multi-robot systems, and large-scale search and exploration tasks. ACO methods are a promising starting point, enabling groups to cooperatively solve complex tasks through simple decentralized pheromone-based interactions. Yet classical ACO relies on fixed control parameters and deterministic rules, limiting its use in uncertain, dynamic, or partially observable environments. Increasingly, multi-agent systems operate in privacy-sensitive contexts, such as environments where individual trajectories must be confidential. Here, standard pheromone-sharing mechanisms can leak agent-level information [1–3]. These issues highlight the



need for an adaptive, scalable, and privacy-preserving ACO framework that still offers ACO's simple, robust real-time coordination.

Among bio-inspired paradigms, ACO is one of the most successful models for decentralized coordination. It relies on pheromone-guided communication and simple decentralized interactions, which have been successfully applied to routing, scheduling, and swarm robotics [4–6]. However, ACO still has key limitations: its parameter settings are fixed or manually adjusted (no self-adaptation); agents may stagnate in suboptimal regions and fail to explore new areas; and it cannot learn or adapt to new environments. These issues undermine the scalability of ACO-based models in dynamic, data-rich scenarios [7].

To overcome these limitations, recent studies have proposed adaptive versions such as the Multi-Strategy ACO [8], which merges rule-based adaptation with chaotic initialization, and the Hierarchical ACO [9], which incorporates global planners for multi-tier task distribution. Despite their utility in improving convergence, these approaches still rely on predefined heuristics and centralized or hierarchical coordination layers, which limit their adaptability and scalability in dynamic environments. Moreover, the literature barely considers privacy and ethical implications, although Multi-Agent Systems (MAS) have become more embedded in connected and data-sensitive environments.

In this scenario, the current study presents the Dual ANT Framework, a distributed swarm system enhanced by learning mechanisms, expanding traditional ACO with dual deep-neural-network meta-adaptation and locally differentially private coordination. The dual ANT is fundamentally different from traditional ones based on static or hierarchical control, as it employs a dual-network topology: a forward Deep Neural Network (DNN) that evaluates swarm performance using multi-dimensional behavior descriptors, and an inverse DNN that performs real-time tuning of key ACO behavioral parameters. This feedback loop enables self-optimization and adaptability to environmental changes without the need for manual parameter tuning. Moreover, the system includes a Lévy flight-driven mobility strategy that increases global exploration and avoids stagnation by allowing occasional long-range exploratory jumps to escape stagnation regions. In conjunction, the use of differential privacy ensures that local pheromones obscure sensitive trajectory data, thereby facilitating reliable and privacy-preserving swarm behavior. This integration aligns with the emerging ethical and governance principles of artificial intelligence and multi-agent collaboration.

The primary contributions of this study are outlined as follows: (i) A neural-enhanced distributed swarm model that merges ACO principles with dual deep learning layers to enable real-time behavioral adaptation. (ii) A mathematical exploration of adaptive stability, which formally analyzes how pheromone entropy correlates with exploration efficiency. (iii) Incorporation of differentially private mechanisms into decentralized coordination to ensure data protection without affecting convergence. (iv) Comprehensive simulation-based validation showed superior performance compared to benchmark models (Multi-Strategy ACO and Hierarchical ACO) in terms of efficiency, scalability, and robustness.

Unlike existing ACO-based adaptive models, the Dual ANT framework integrates four components within a single decentralized architecture: (i) a forward neural network that provides a real-time introspective assessment of swarm efficiency based on multidimensional behavioral descriptors; (ii) an inverse neural network that translates this introspective signal into targeted online parameter adjustments; (iii) a Lévy-flight-driven mobility strategy that enables fast recovery from stagnation and enhances global exploration; and (iv) a locally differentially private pheromone update rule, ensuring agent-level trajectory privacy while preserving collective signal quality. The interaction of these four components yields a meta-adaptive and privacy-aware coordination mechanism that has not been explored in previous adaptive ACO or swarm intelligence studies.

The remainder of this paper is structured as follows: [Section 2](#) delves into the literature concerning decentralized ant colony optimization, swarm intelligence integrated with machine learning, and privacy-preserving multi-agent systems. [Section 3](#) outlines our methodology, which includes a decentralized ACO-based search planner enhanced with differential privacy, a dual neural network framework for introspective evaluation and adaptation, and a cohesive planning-evaluation-adaptation cycle. In [Section 4](#), we describe the experimental setup and parameter configurations, present the simulation outcomes, and provide an in-depth analysis of these results. Finally, [Section 5](#) concludes the paper and suggests intriguing directions for future research.

## 2 Related Work

Our research lies at the intersection of decentralized ant colony optimization, neural-enhanced swarm intelligence, and privacy-preserving multi-agent coordination. This section reviews the most recent advances in these areas (2023–present) to position our contributions within the contemporary state of the art.

### 2.1 Advanced Ant Colony Optimization Variants

Existing approaches have attempted to solve the known problems associated with standard ACO, most notably premature convergence and stagnation. Wang et al. [10] presented a multifactor adaptive ACO for robot path planning, in which chaotic initialization and dynamic evaporation rate are adopted to enhance global search performance. Liu et al. (2025) developed a hierarchical ACO system, with the help of global and local planners that identify better solutions than sub-swarms in large environments independently [11]. Although these methods improve ACO performance, they remain rule-based and do not incorporate true meta-adaptive learning mechanisms.

### 2.2 Machine Learning-Augmented Swarm Intelligence

Machine learning, particularly deep learning, has increasingly been combined with swarm intelligence to create adaptive multi-agent systems. Verma et al. (2024) incorporated neural networks into nature-inspired algorithms and reported significant performance improvements. However, most existing studies use neural networks primarily for surrogate modeling or state evaluation rather than for real-time parameter adaptation [12]. In a study closer to ours, Bany Salameh et al. (2024) proposed a Deep Reinforcement Learning (DRL) approach for deriving single-agent search policies that do not rely on pheromone communication [13]. Although effective, this approach requires extensive training and exhibits limited generalization across different environments. Conversely, our approach maintains the interpreted communication of the ACO model and leverages neural networks for a more intricate task: assessing the convergence of the swarm and adjusting the ACO parameters as needed. This strategy merges the sample efficiency of ACO with the adaptability of DNNs. The most closely related model is that of Liu et al. (2024), which is based on the work of Liu et al. [14]. In their study, a neural network surrogate model was applied to predict the performance of various ACO configurations that were pre-selected offline. We expanded on this concept by introducing an online adaptation method that allows continuous parameter evaluation and adjustment during use while providing immediate adaptation to changes in exposure conditions.

### 2.3 Privacy-Preserving Multi-Agent Systems

As MAS is used in sensitive areas, privacy protection has become a major concern. Akbari and Zeng (2025) proposed a unified learning method for multiple agent systems that allows agents to train models together without exchanging raw data, thus effectively protecting privacy [15]. However, this approach does not protect the space paths of the agent during the task but creates a common model. Baharisangari et al.

(2024) used differential privacy (DP) in a centralized multi-agent path planning algorithm, injecting noise into a global plan to obscure the goals of individual agents [16]. Shi et al. introduced a decentralized strategy. (2024), introduced noise into the transmission velocity vector and developed a local differential privacy model (LDP) for agent communication in a flock task.

## 2.4 Summary and Novelty Positioning

Recent adaptive ACO variants incorporate metaheuristic learning, chaotic initialization, or dynamic evaporation rates to address premature convergence. However, these approaches rely primarily on predefined rule-based adjustments and operate without feedback mechanisms. In contrast, the Dual ANT employs a dual-network meta-adaptive controller that continuously evaluates the swarm's behavioral descriptors and updates parameters online using learned mappings rather than manually designed rules. Furthermore, none of the surveyed adaptive ACO variants addressed privacy preservation, whereas Dual ANT integrated a locally differentially private pheromone mechanism. Table 1 summarizes these differences and highlights the unique role of real-time learning and privacy awareness in our framework.

**Table 1:** Key differences between existing adaptive ACO approaches and the proposed dual ANT meta-adaptive framework

| Aspect              | Recent adaptive ACO variants                       | Dual ANT (Proposed)                            |
|---------------------|--|--|
| Adaptation type     | Rule-based/heuristic                               | Learned, neural meta-adaptation                |
| Timing              | Mostly offline or static during run                | Online, continuous parameter tuning            |
| Surrogate models    | Some use offline surrogates to rank configurations | Forward network estimates efficiency online    |
| Parameter control   | Manual thresholds or predefined rules              | Inverse network predicts optimal adjustments   |
| Stagnation handling | Reactive evaporation or local heuristics           | Lévy-flight escape triggered via introspection |
| Privacy             | Not addressed                                      | Built-in local differential privacy            |

## 3 Methodology

This section introduces the main contribution of this study, a methodology that combines an ACO planner for search trajectory generation with a dual neural network structure to evaluate and improve real-time search efficiency. The objective is to achieve a self-conscious system capable of generating plans for action and monitoring, evaluating, and improving its performance on the fly.

### 3.1 ACO-Based Multi-Agent Search Planner

The decentralized search planner is motivated by the foraging behavior of ants. Every swarm agent behaves like an ant, exploring the environment and depositing digital pheromones marking the way for a common search.

#### 1. Agent Movement and State Transition:

At each iteration  $t$ , an agent  $a_i$  located in its current cell  $(x, y)$  selects its next position  $(x', y')$  from the collection of neighboring passable cells  $\mathcal{N}(x, y)$  using a probabilistic transition rule  $P$ :

$$P_{(x,y) \rightarrow (x',y')} = \frac{[\tau_{(x,y)}]^\alpha [\eta_{(x',y')}]^\beta}{\sum_{(x',y') \in \mathcal{N}(x,y)} [\tau_{(x,y)}]^\alpha [\eta_{(x',y')}]^\beta} \quad (1)$$

where  $\tau_{(x',y')}$  denotes the pheromone intensity,  $\eta_{(x',y')}$  represents the heuristic desirability (which is inversely proportional to the distance to the closest undiscovered target), and  $\alpha$  and  $\beta$  are parameters that determine their relative impact.

## 2. Pheromone Update with Differential Privacy:

To protect the privacy of individual agent paths, we used the Laplace mechanism for DP. The global pheromone map is updated with noise deposits as follows:

$$\Delta\tau_{xy} = \sum_{i=1}^n \Delta\tau_{xy}^i + \text{Lap}\left(\frac{\Delta f}{\epsilon_{\text{iter}}}\right) \quad (2)$$

$$\tau_{(x,y)} \leftarrow (1 - \rho) \cdot \tau_{(x,y)} + \Delta\tau_{xy} \quad (3)$$

where  $\Delta f = Q$  is the sensitivity (maximum contribution of one agent), and  $\epsilon_{\text{iter}}$  is the privacy budget per iteration.

**Privacy Budget ( $\epsilon$ ) and Composition:** The parameter  $\epsilon$  represents the privacy budget that calibrates the magnitude of the Laplace noise added to each agent's pheromone deposit. Because pheromone updates occur at every time step, privacy loss accumulates under the standard sequential composition rule. If  $\epsilon_t$  is the per-step budget, then after  $T$  iterations, the total privacy guarantee becomes  $\epsilon_{\text{total}} = \sum_{t=1}^T \epsilon_t$ . In our implementation, we used small  $\epsilon_t$  values to keep the total privacy leakage bounded while still preserving sufficient gradient information for effective coordination.

**Sensitivity ( $\Delta f$ ):** The sensitivity  $\Delta f$  denotes the maximum influence a single agent can exert on pheromone entry during one update. Because each agent deposits at most a fixed quantity  $Q$  at its current location, the sensitivity is tightly bounded by  $\Delta f = Q$ . This ensured the proper calibration of the Laplace noise injected into the pheromone map.

**Impact of Noise on Convergence:** Injecting Laplace noise flattens the pheromone distribution and reduces the gradient sharpness, which can slow convergence during the early exploration. However, the dual-network controller compensates for this by increasing the exploration parameters whenever the forward model detects a performance drop attributable to noise. Empirically, moderate privacy settings preserve convergence while yielding only a slight increase in the exploration steps.

**Local Differential Privacy (LDP):** Because each agent independently perturbs its own pheromone deposit before sending it to the shared map, the mechanism satisfies local differential privacy. This protects each agent's trajectory, even if the global pheromone map is fully observable by external parties. Our study introduces a decentralized LDP-based method for updating pheromones. However, we perform a novel study of the interplay between this privacy noise and a neural network adaptation system, showing how the latter can remain robust even under performance loss owing to the privacy constraints. The proposed pheromone update mechanism satisfies LDP because each agent independently perturbs its pheromone deposit before contributing to the global map. Under the Laplace mechanism with sensitivity  $\Delta f = Q$  and per-step privacy budget  $\epsilon_t$ , the probability of distinguishing an agent's true movement at any step is bounded by  $e^{\epsilon_t}$ . With the sequential composition property, the total privacy leakage over  $T$  steps remains bounded by  $\sum_{t=1}^T \epsilon_t$ . In practice, moderate noise levels (e.g.,  $\epsilon_t \in [0.1, 0.3]$ ) substantially obscure individual trajectories without disrupting swarm-level coordination.

### 3.2 Dual Neural Network Architecture for Introspective Evaluation and Adaptation

The ACO planner provides a robust mechanism for decentralized coordination. However, its performance is highly sensitive to the parameters  $\alpha$ ,  $\beta$ , and the Lévy flight scale. To overcome this limitation and create a truly adaptive system, we introduced a dual-neural network architecture. This architecture allows

the system to introspectively evaluate its current performance and autonomously adapt its search parameters in real-time.

### 3.2.1 Feature Extraction

At each time step, a feature vector  $\mathbf{F}_t$  is generated from the global system state. This vector records high-level behaviors that are crucial for assessing the effectiveness of the search:

$$\mathbf{F}_t = [\phi_{\text{dispersion}}, \phi_{\text{discovery}}, \phi_{\text{entropy}}, \phi_{\text{coverage}}, \phi_{\text{stagnation}}] \quad (4)$$

- Agent Dispersion ( $\phi_{\text{dispersion}}$ ): the standard deviation of the agent's position normalized by the grid size. This measured the efficiency of agent dispersion.
- Target Discovery Rate ( $\phi_{\text{discovery}}$ ): the proportion of targets located in the time steps that have elapsed. This indicates the extent to which the agents experience stagnation.
- Pheromone Entropy ( $\phi_{\text{entropy}}$ ): normalized Shannon entropy of the pheromone map. It measures the diversity of exploration; low entropy reveals highly concentrated trails, and high entropy indicates that exploration is uniformly balanced.
- Percentage of Coverage ( $\phi_{\text{coverage}}$ ): percentage of grid cells visited at least once. It is a measure of the completeness of the search.
- Stagnation Level ( $\phi_{\text{stagnation}}$ ): the average stagnation counter of agents, normalized by its threshold. This represents the degree of immobilization of the agent.

### 3.2.2 Forward Model: Real-Time Efficiency Evaluation

The first component is a feedforward Deep Neural Network (DNN) that acts as a forward model. Its purpose is to map the feature vector  $\mathbf{F}_t$  to a scalar efficiency score  $\hat{E}_t \in [0, 1]$ .

- Architecture: It contains an input layer (five neurons), two hidden layers (64 neurons per layer, ReLU activation), and an output layer (one neuron, Sigmoid activation).

- Training Data: Networks are trained offline using simulation data. The data point  $(\mathbf{F}_i, E_i)$  comprises historical state descriptors and their corresponding ground truth efficiency  $E_i$  computed using the parameters given by Eq. (5).

$$E_i = w_1 \cdot \phi_{\text{dispersion}} + w_2 \cdot \phi_{\text{discovery}} + w_3 \cdot \phi_{\text{entropy}} + w_4 \cdot \phi_{\text{coverage}} - w_5 \cdot \phi_{\text{stagnation}} \quad (5)$$

where  $w_1, w_2, \dots, w_5$  are the model weights obtained during offline training.

- Training Process: The model was learned using the Adam optimizer to minimize the average square error (MSE) between prediction  $\hat{E}_i$  and true efficiency score  $E_i$ . The model can be further adapted online by training.

The forward network is trained to approximate the true efficiency score  $E_t$  derived from Eq. (5). Given a dataset of state descriptors  $\phi_t$  and corresponding ground-truth efficiencies, the parameters  $\theta$  of the forward model  $F_\theta(\phi_t)$  are optimized using the mean squared error (MSE) as follows:

$$\mathcal{L}_{\text{forward}} = \frac{1}{N} \sum_{t=1}^N (F_\theta(\phi_t) - E_t)^2$$

This objective encourages the model to provide accurate, real-time estimates of swarm performance.

### 3.2.3 Inverse Model: Adaptive Parameter Control

The second component is a second DNN that acts as the inverse model of the first. This is the core of the system's adaptability. When the forward model predicts an efficiency score below the threshold  $\theta$ , the inverse model is activated. It takes the current feature vector  $\mathbf{F}_t$  and maps it to optimal adjustments for the key behavioral parameters  $(\Delta\alpha, \Delta\beta, \Delta\lambda)$  (Levy scale).

- Architecture: Same as the forward model but with three output neurons (Tanh activation) emitting adjustments in  $[-1, 1]$ .

- Objective: The model was trained to predict parameter changes that would maximize the efficiency score at a certain system state. The goal of training is to decrease the gap between actual efficiency and optimal performance.

- Action: scale the changes through the output using the following global parameters:

$$\begin{cases} \alpha_{t+1} = \alpha_t * (1 + \gamma_\alpha \Delta\alpha) \\ \beta_{t+1} = \beta_t * (1 + \gamma_\beta \Delta\beta) \\ \lambda_{t+1} = \lambda_t * (1 + \gamma_\lambda \Delta\lambda) \end{cases} \quad (6)$$

where  $\gamma$  is the scaling factor that regulates the speed of adaptation.

The inverse network learns to predict the parameter adjustments  $\Delta\mathbf{p}_t = [\Delta\alpha, \Delta\beta, \Delta\lambda]^\top$  that improve efficiency. Given the optimal adjustments  $\Delta\mathbf{p}_t^*$  obtained from offline simulation-based optimization, the parameters  $\psi$  of the inverse model  $G_\psi(\phi_t)$  were trained using:

$$\mathcal{L}_{\text{inverse}} = \frac{1}{N} \sum_{t=1}^N \| G_\psi(\phi_t) - \Delta\mathbf{p}_t^* \|_2^2 \quad (7)$$

This supervised objective enables the network to approximate the best parameter updates for each observed swarm state.

We applied the following three constraints to avoid instability:

- (1) Parameter Bounds. Each behavioral parameter  $(\alpha, \beta, \lambda)$  is clipped to be within a predefined safe area  $[\alpha_{\min}, \alpha_{\max}]$ ,  $[\beta_{\min}, \beta_{\max}]$ , and  $[\lambda_{\min}, \lambda_{\max}]$ . This way, it disallows the optimization system from going through states that would lead to a destabilization of the search dynamics.

- (2) Adaptation Scaling. The raw outputs of the inverse model are multiplied by a small global scaling coefficient  $\eta$  to maintain soft parameter updates. This also prevents sudden instabilities that might lead to oscillatory behavior.

- (3) Hysteresis Triggering. The inverse model is photographed only when the predicted output efficiency continues below a certain threshold  $\theta$  for  $N$  consecutive iterations. This requirement ensures that parameter adaptation reacts only to sustained problems with performance and not to transient blips.

These mechanisms work together to maintain stable behavior during online learning and avoid oscillations owing to over-aggressive adaptation.

The inverse network is mainly trained offline on supervised targets obtained from batched high-efficiency simulation episodes. This offline phase prepares a stable base model that can learn the general adaptation of multiple stages of the environment.

During execution, the network may optionally undergo lightweight online fine-tuning using a small replay buffer of recent state-adjustment pairs. This step does not involve trial-and-error or reward-driven

optimization (as in the case of reinforcement learning). Instead, it refines the model using the same supervised loss to improve the performance under previously unseen conditions while preserving stability.

### 3.3 The Integrated Planning-Evaluation-Adaptation Loop

The primary strength of the framework lies in the closed-loop interaction between the ACO planner and dual neural network architecture, which together establish a feedback cycle for autonomous optimization.

Operational Workflow:

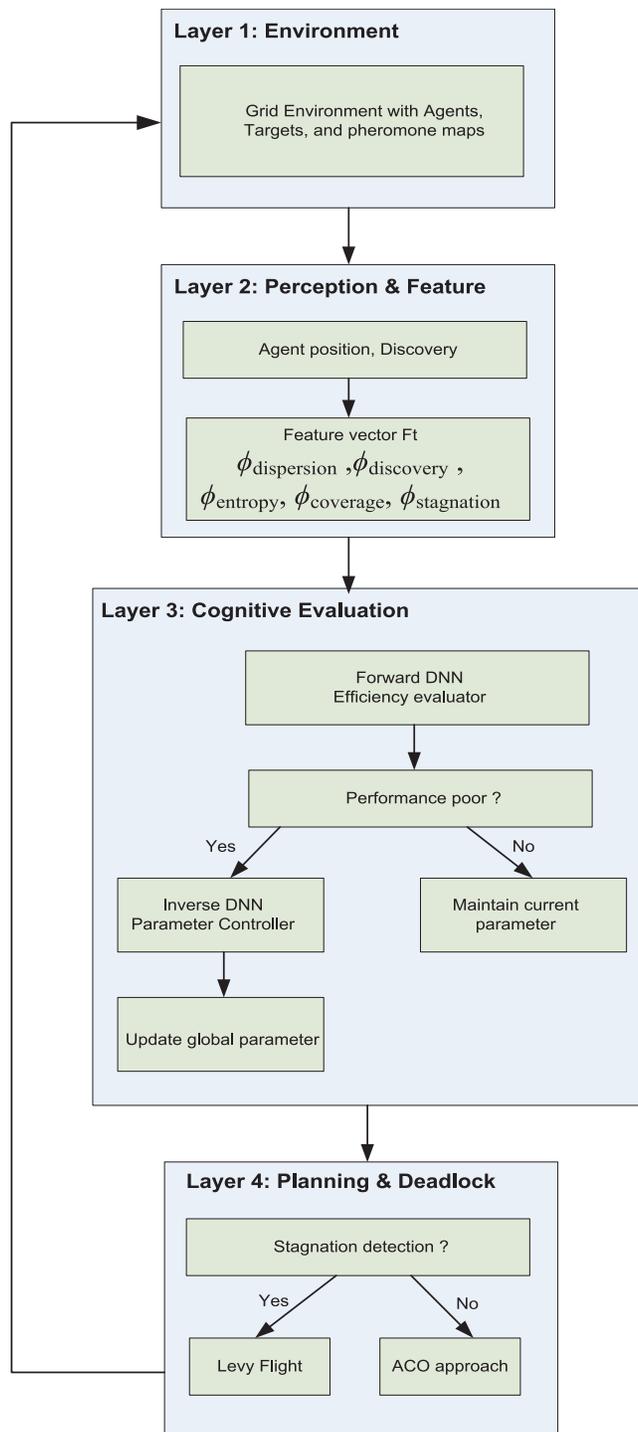
1. Plan & Execute: Agents execute the ACO algorithm (Section 3.1) until a target is located or a predetermined number of iterations is reached.
2. Extract Features: The feature vector  $\mathbf{F}_t$  is computed while the system state (agent positions, pheromone map, and discovery status) is monitored.
3. Evaluate (Forward Model): The forward model DNN receives a feature vector  $\mathbf{F}_t$  and produces an efficiency score  $\hat{E}_t$ .
4. Adapt (Inverse Model): The inverse model is used if  $\hat{E}_t$  is below a predetermined threshold  $\theta$  for  $N$  consecutive iterations. Using  $\mathbf{F}_t$ , it forecasts the best parameter changes  $(\Delta\alpha, \Delta\beta, \Delta\lambda)$ , which are then applied.
5. Loop: Agents work with new, updated parameters.

This iterative loop enables the system to self-correct and ideally maintain high performance. For example, if low efficiency is due to clustering of agents (low dispersion), the inverse model may increase  $\beta$  to encourage more exploration, or it will change the Levy scale  $\lambda$  to explore a larger area that has not been previously visited.

Fig. 1 presents a closed-loop adaptive multi-agent system with targets and a pheromone map. The system works in four layers: (i) The Environment & Execution layer where agents interact with the grid environment and pheromone map; (ii) The Perception & Feature Extraction layer that computes high-level metrics (e.g., dispersion, discovery); (iii) The Cognitive Evaluation & Adaptation layer consisting of a Forward DNN to evaluate efficiency, and an Inverse DNN to provoke adaptation when performance drops below a threshold  $\theta$ ; and (iv) The Planning & Reallocation layer that uses Levy Flight and ACO rules to overcome prolonged stagnation. This forms a feedback loop for independent optimization.

### 3.4 Overall Algorithm Flow

Algorithm 1 provides a step-by-step description of the Dual ANT framework, illustrating how decentralized ACO planning is coupled with dual neural network evaluation and adaptive parameter control. The Dual ANT Framework is a sophisticated multi-agent search algorithm that integrates a decentralized ACO with a dual neural network setup for introspective evaluation and adaptive management. The process begins by initializing the environment, agent, and pretrained neural network. During execution, each agent assesses its immediate environment and navigates using the standard ACO probability rule, following pheromone trails. If stagnation occurs, the agent uses the Lévy flight jump to escape the local optima. Pheromone deposits are infused with noise to maintain differential privacy and protect the data of the agent's path. Periodically, high-level features such as agent dispersion, target discovery rate, and pheromone entropy are extracted and fed into a forward deep neural network (DNN), which generates efficiency scores. If these scores consistently fall below a specific threshold, an inverse DNN is activated to compute dynamic adjustments to key behavioral parameters, such as pheromone and heuristic weight, allowing the real-time adaptation of the agent's search strategy. This closed-loop process continues until all targets are found or the step limits are reached, resulting in a robust, self-optimizing system that effectively balances efficiency, adaptability, and privacy.



**Figure 1:** The closed-loop architecture of the Dual ANT framework

---

**Algorithm 1:** Dual ANT framework for multi-agent collaborative search
 

---

## 1. Initialize

- Grid environment, targets, and pheromone map.
- Neural networks (forward DNN and inverse DNN) with pre-trained weights.
- Agent parameters:  $\alpha, \beta, \lambda, \rho$ , etc.
- Set thresholds: efficiency threshold  $\theta$ , stagnation threshold  $T_s$ , step limit.

## 2. For each time step t:

- For each agent

Perceive local environment (neighboring cells, pheromone levels, targets).

If (stagnation counter >  $T_s$ )

Trigger Levy flight to jump to a random target point.

Else

Select the next cell using the ACO probabilistic rule (Eq. (1)).

Move to the selected cell.

Deposit pheromone with differential privacy (Eqs. (2) and (3)).

## 3. Extract Global Features (Eq. (4)):

- Compute:  $\phi_{\text{dispersion}}, \phi_{\text{discovery}}, \phi_{\text{entropy}}, \phi_{\text{coverage}}, \phi_{\text{stagnation}}$

## 4. Evaluate Efficiency via Forward DNN:

- Input: Feature vector  $\mathbf{F}_t$
- Output: Efficiency score  $\hat{E}_t \in [0, 1]$

5. If ( $\hat{E}_t < \theta$ ) for N consecutive steps:

- Activate Inverse DNN:
- Input:  $\mathbf{F}_t$
- Output: Parameter adjustments ( $\Delta\alpha, \Delta\beta, \Delta\lambda$ )
- Update parameters (Eq. (6)):

$$\begin{cases} \alpha_{t+1} = \alpha_t * (1 + \gamma_\alpha \Delta\alpha) \\ \beta_{t+1} = \beta_t * (1 + \gamma_\beta \Delta\beta) \\ \lambda_{t+1} = \lambda_t * (1 + \gamma_\lambda \Delta\lambda) \end{cases}$$

## 6. Check Termination Condition:

- If all targets found OR max steps reached  $\rightarrow$  End.
  - Else  $\rightarrow$  Repeat from Step 2.
- 

The computational complexity of a single simulation step is a function of the number of agents, grid size  $G = M \times N$ , and neural network operations.

- Agent movement: For each agent, the movement probability is calculated by checking eight neighbors. The complexity is  $O(n)$ .
- Pheromone update: Evaporation and deposit are  $O(G)$  operations.
- Feature extraction: the calculation of functions such as dispersion and entropy is  $O(n + G)$ .
- Neural network inference: because the network architecture is fixed, forward transmission through the two DNNs is constant time,  $O(1)$ , compared to the problem size.

The complexity of the entire step is  $O(n + G)$ , which is linear and efficient. This allows the framework to be scaled to larger networks without incurring high computational costs.

## 4 Experimental Setup and Simulation Results

We demonstrated the effectiveness of the proposed model through simulations in a grid-based, discrete world. All simulations were performed on a discrete 2D grid of  $80 \times 80$  cells. Unless otherwise specified, the swarm was composed of 40 independent agents that were placed uniformly and randomly in the work area. Each task contained 10 randomly positioned targets and a field of static obstacles with an obstacle density of 8% so that different levels of navigation difficulty could be experienced. An additional random seed, which generated the environment layouts, was changed from episode to episode to allow different configurations, such as varying obstacle clusters, initial agent positions, and goal locations.

All experiments were bootstrapped based on a static base seed of 42 (implemented in NumPy, Python's random module, and within the machine learning framework) to guarantee reproducibility. For the multi-run evaluations, we adopted deterministic seed offsets ( $42 + i$ ) for run  $i$  to achieve both diversity and reproducibility of results.

Environment generation (obstacles, target placement, agent initialization) and stochastic elements of agent behavior used the same seed values, guaranteeing that the experiments could be replicated exactly by reusing the recorded seeds.

On the specified hardware, a single simulation episode of 1000 iterations with 40 agents on an  $80 \times 80$  grid required approximately 1.8 s of computation time. Larger grids ( $100 \times 100$ ) increase the runtime to approximately 2.3 s, whereas heavier agent populations (e.g., 60–80 agents) increase the runtime to 2.5–3.1 s due to additional movement and pheromone update computations. These runtimes confirm that the Dual ANT operates efficiently on a scale and is suitable for real-time or near-real-time decentralized coordination tasks.

### 4.1 Comparative Analysis: Dual ANT Framework vs. Advanced ACO Variants

#### 4.1.1 Multi-Strategy Adaptive ACO

In 2025, Wang et al. presented a multi-strategy adaptive ACO approach that incorporates several key characteristics to improve both exploration and convergence performance [10]. This method uses chaotic initialization to obtain the initial population, reducing the likelihood of premature convergence. Furthermore, it includes the dynamic evaporation rates that vary according to the stage of the search process promoting exploration in the early stages and exploitation in the later stages. In addition, the algorithm adopts multiple pheromone update schemes to efficiently capture the local and global search information. Finally, rule-based adaptation schemes are used to dynamically adjust the parameters and behavior of the algorithm to adapt intelligently to different search conditions and problem difficulties. To clearly contrast rule-based adaptive mechanisms with our learning-driven meta-adaptation, Table 2 compares the multi-strategy ACO of Wang et al. with the proposed Dual ANT framework.

**Table 2:** Comparative analysis of adaptation and learning mechanisms: multi-strategy ACO vs. dual ANT framework [10]

| Aspect               | Wang et al. (Multi-strategy ACO) | Our dual ANT framework                        |
|----------------------|----------------------------------|---|
| Adaptation mechanism | Predefined rule-based strategies | Real-time neural network-driven adaptation    |
| Learning capability  | Static rules, no learning        | Online learning through dual DNN architecture |

(Continued)

**Table 2 (continued)**

| Aspect                 | Wang et al. (Multi-strategy ACO)      | Our dual ANT framework                                       |
|------------------------|---------------------------------------|--|
| Parameter optimization | Manual tuning of rule thresholds      | Automatic parameter adjustment via inverse DNN               |
| Stagnation handling    | Reactive evaporation rate changes     | Proactive deadlock detection + Levy flight escape            |
| Scalability            | Limited by predefined rule complexity | Linear complexity $O(n+G)$ with neural inference             |
| Generalization         | Domain-specific rule design           | Learned adaptation policies transferable across environments |

Key Difference: Unlike Wang et al. [10], who employed several predetermined strategies, our framework developed optimal adaptation policies on the fly, removing the necessity for manually designed rules.

#### 4.1.2 Hierarchical ACO

Liu et al. (2025) proposed a hierarchical framework for ant colony optimization (ACO) that incorporates a structured, multi-layered coordination system [11]. This model includes global planners that oversee several sub-swarms, thereby enabling the efficient distribution of exploration efforts across different regions of the search space. The hierarchical coordination system enables clear communication between the global and local levels, improving scalability and decision-making efficiency. A top-down approach in which agents assign tasks to subordinate swarms based on global priority and environmental feedback. The system also uses multi-level optimization with global exploration and local refinement to ensure better convergence and result quality in complex dynamic environments. A detailed comparison of Liu et al.'s hierarchical ACO and our Dual ANT framework is provided in Table 3.

**Table 3:** Architectural and operational comparison between hierarchical ACO and dual ANT framework [11]

| Aspect                  | Liu et al. (Hierarchical ACO)     | Our dual ANT framework                     |
|-------------------------|-----------------------------------|--|
| Architecture            | Centralized hierarchical control  | Fully decentralized, emergent coordination |
| Communication overhead  | High (inter-level communication)  | Low (local pheromone trails only)          |
| Single point of failure | Yes (global planner)              | No (fully distributed)                     |
| Adaptation speed        | Slower (top-down decision making) | Faster (local, real-time adjustments)      |
| Privacy protection      | Not addressed                     | Built-in differential privacy              |
| Computational load      | Concentrated at hierarchy levels  | Distributed across all agents              |

Distinctive Feature: Our framework achieves coordinated actions through emergent intelligence while preserving full decentralization, in contrast to the centralized hierarchical control found in Liu et al. [11].

### 4.2 Simulation

The simulation framework was implemented in Python 3.8 and included the necessary libraries for numerical algorithms, machine learning, and plotting. The parameters were chosen using an iterative sensitivity analysis and grid search that provides robust performance across contexts. The parameters used belong to three types (ACO, Levy Flight, and Neural Network), which are listed in Table 3. Sensitivity analysis and grid search were performed for each parameter over a pre-designed range according to previous ACO literature and preliminary experiments. For instance, the pheromone influence factor  $\alpha$  was experimented within the spectrum [0.5, 2.0] because a larger value indicates an overweight of pheromone trails and thus premature convergence for this parameter. The heuristic guidance  $\beta$  was examined in the range of [1.0, 5.0] to achieve a trade-off between the local heuristic guidance and global exploration. The evaporation rate  $\rho$  was in the range of [0.1, 0.5] to prevent persistent or sudden disappearance of the trail. The Lévy flight scale was between 0.1 and 1.0, allowing occasional long jumps without disrupting navigation. The best values in this table correspond to the evaluation of these new parameters, considering those that provided better performance and stability in all test environments.

To avoid bias and ensure a fair performance evaluation, the following measures were implemented:

(1) Multiple Random Seeds: All experiments were repeated on 20 different seeds to model the variance in the initial agent positioning, obstacle configuration, and goal distribution.

(2) Variety of Environment Settings: We tested all the algorithms on randomly chosen environments with different obstacle densities and target distributions, such that no algorithm was fine-tuned for specific environment settings.

(3) Baseline Parameter Tuning: Competing algorithms (Multi-Strategy ACO and Hierarchical ACO) were tuned to their best parameter ranges to obtain strong and fair baselines.

(4) Statistical Reporting: 95% confidence intervals and standard deviation bars are provided in the results to avoid undue embarrassing single-run anomalies.

These measures ensure that the reported improvements of the Dual ANT are not due to experimental artifacts but are indeed reliable and reproducible.

Each parameter in Table 4 has a different effect on the swarm coordination dynamics.

**Table 4:** Optimized parameter settings for the proposed framework

| Category | Parameter           | Symbol   | Value | Description & Rationale  |
|----------|---------------------|----------|-------|--|
| ACO core | Evaporation rate    | $\rho$   | 0.15  | It balances the decay of old information with the integration of new knowledge. Higher rates promote exploration.                  |
|          | Pheromone influence | $\alpha$ | 1.2   | It controls the tendency of an agent to follow established trails. Adjusted to prevent premature convergence to suboptimal routes. |
|          | Heuristic influence | $\beta$  | 2.5   | It controls the attraction to unexplored areas and targets. Set the maximum value of (alpha) to encourage exploration.             |

(Continued)

**Table 4 (continued)**

| Category       | Parameter            | Symbol    | Value  | Description & Rationale  |
|----------------|----------------------|-----------|--------|--|
| Levy flight    | Deposit amount       | Q         | 2.0    | The pheromone base amount deposited at each stage. Increase the intensity of the pheromone map.  |
|                | Scale parameter      | $\lambda$ | 7.0    | It controls the average size of the jump steps. Designed to be large enough to escape local regions, but not to overtake the grid.                                 |
|                | Stagnation threshold | Ts        | 8      | Number of consecutive steps without exploration before triggering a deadlock recovery. Balance between reactivity and stability.                                   |
| Neural network | DNN learning rate    | $\eta$    | 0.001  | The learning rate for Adam optimizer. Standard values for stable training of medium-sized networks.  |
|                | Batch size           | B         | 32     | The number of experiences sampled from the replay memory for each training step.   |
|                | Memory capacity      | M         | 10,000 | The size of the replay buffer. Large enough to ensure training samples.  |
| Evaluation     | Adaptation scaling   | $\gamma$  | 0.2    | Factors controlling the size of parameter adjustments from reverse models ( $\gamma_\alpha, \gamma_\beta, \gamma_\lambda$ ). Avoid excessively volatile behaviour. |
|                | Efficiency threshold | $\theta$  | 0.4    | The threshold below which the reverse model is activated to adapt the parameters.  |

(1) Pheromone Influence ( $\alpha$ ): The amount of pheromones used by an agent to make moves. Higher values lead to overexploitation.

(2) Heuristic Influence ( $\beta$ ): Weight of local heuristic information (e.g., distance to the nearest target). Larger values of  $\beta$  favor directed exploration, whereas smaller values enforce random motion.

(3) Evaporation Rate ( $\rho$ ): Specifies the speed at which pheromone trails fade. Low  $\rho$  enhances long-term memory; whereas high  $\rho$  fosters adaptability but possibly undermines coordination.

(4) Lévy Flight Scale ( $\lambda$ ): Determines the probability and distance of long-range exploratory jumps. A larger  $\lambda$  leads to larger amounts of global exploration and better escape from stagnation.

(5) Dual-Network Adaptation Threshold ( $\theta$ ): The inverse network is activated only if the forward model triggers a level of efficiency below  $\theta$ , being self-corrective only when necessary.

(6) Adaptation Scaling Factor ( $\eta$ ): This constrains the scale of online parameter updates to ensure safe and stable updates.

This detailed examination elucidates how the Dual ANT model is designed to trade off these properties during the self-organization of swarm coordination.

### Tuning Process:

The fundamental concept: The initial settings were established following standard practices in the ACO literature.

Grid Search: A grid search was conducted for key parameters ( $\alpha$ ,  $\beta$ ,  $\lambda$ ,  $\rho$ ) over a specified range. Each setup was assessed through 10 simulations, with performance measured by the average coverage rate and the number of targets located.

System sensitivity analysis: The system sensitivity in response to each parameter was analyzed by varying one of the parameters at a time while keeping the others constant. This strategy contributed to the identification of the critical parameters influencing performance.

Neural Network Training: We trained the forward model using a pre-registered dataset of state-efficiency pairs derived from simulations of random parameter choices. The data were split into 80% for training and 20% for validation.

### 4.3 Results and Analysis

The data depicted in Fig. 2 provides a comparative evaluation of the overall efficiency scores between three optimization approaches: Multi-Strategy ACO [10], Hierarchical ACO [11], and the novel Dual ANT Framework. In this analysis, efficiency refers to the capacity of the algorithm to balance exploration, speed of convergence and goal achievement in a simulated environment. The results show that the Dual ANT Framework provides better efficiency with a maximum score of 61.6 compared to both reference models. Multi-strategy ACO has a medium improvement (47.2 score), and the Hierarchical ACO has the lowest performance level (23.7). The Dual ANT Framework is more stable and has less variance during repeated runs. This improvement can be attributed to the dual neural adaptation mechanism and closed-loop self-optimization, which enable agents to learn dynamic optimal exploration strategies and adjust parameters in time during the run time. In contrast, Multi-Strategy ACO and Hierarchical ACO have predefined rules and static hierarchies of evolution which make them less adaptive to dynamic or uncertain issues.

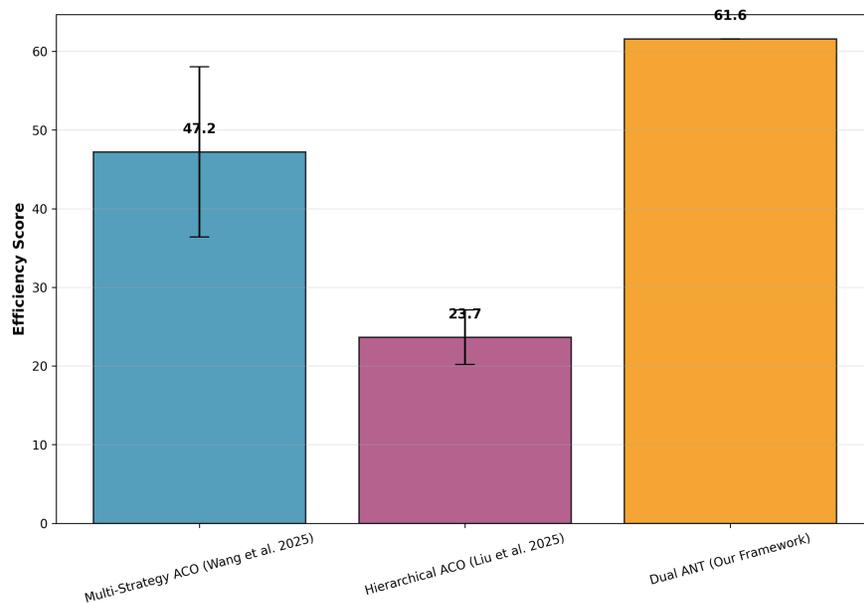
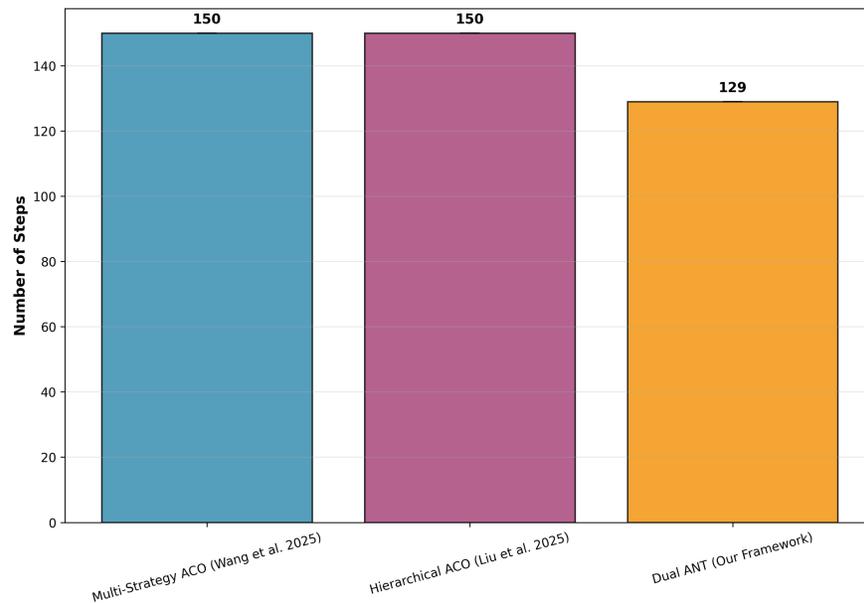


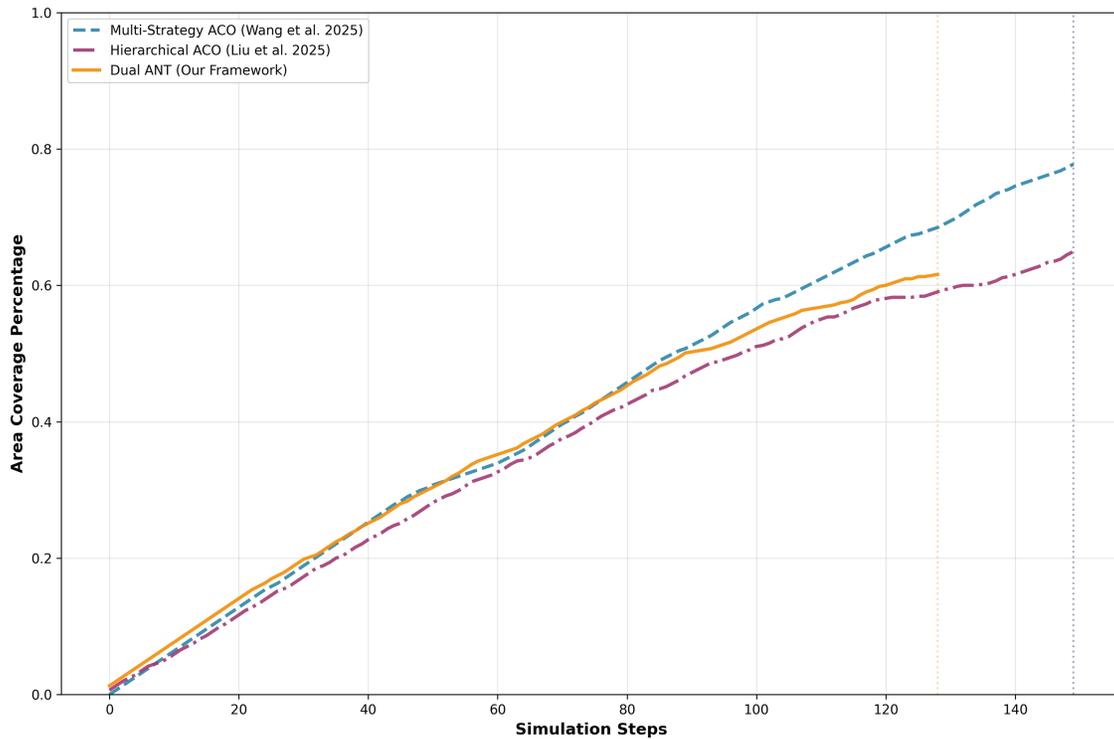
Figure 2: Overall efficiency core [10,11]

Fig. 3 compares the number of necessary steps to achieve all the goals between these three strategies: Multi-Strategy ACO [10], Hierarchical ACO [11], and our Dual ANT Framework. The results show that the Dual ANT Framework performs better than the others by solving all targets in an average of 129 steps, which is much lower than the Hierarchical ACO and Multi-Strategy ACO at 150 steps. We consider such an enhancement a testament to the ability of Dual ANT to dynamically balance exploration and exploitation, allowing agents to reach their goals faster with minimal delays. The reduction in steps comes from the two-level adaptive learning strategy, such that the system can adjust its navigation policy adaptively in real-time and avoid revisiting previously explored areas. In addition, agents use Lévy flight to escape local traps, enabling more efficient exploration of the environment. In contrast, hierarchical and multi-strategy ACO models depend on fixed rules, which limit adaptation and lead to longer convergence times.



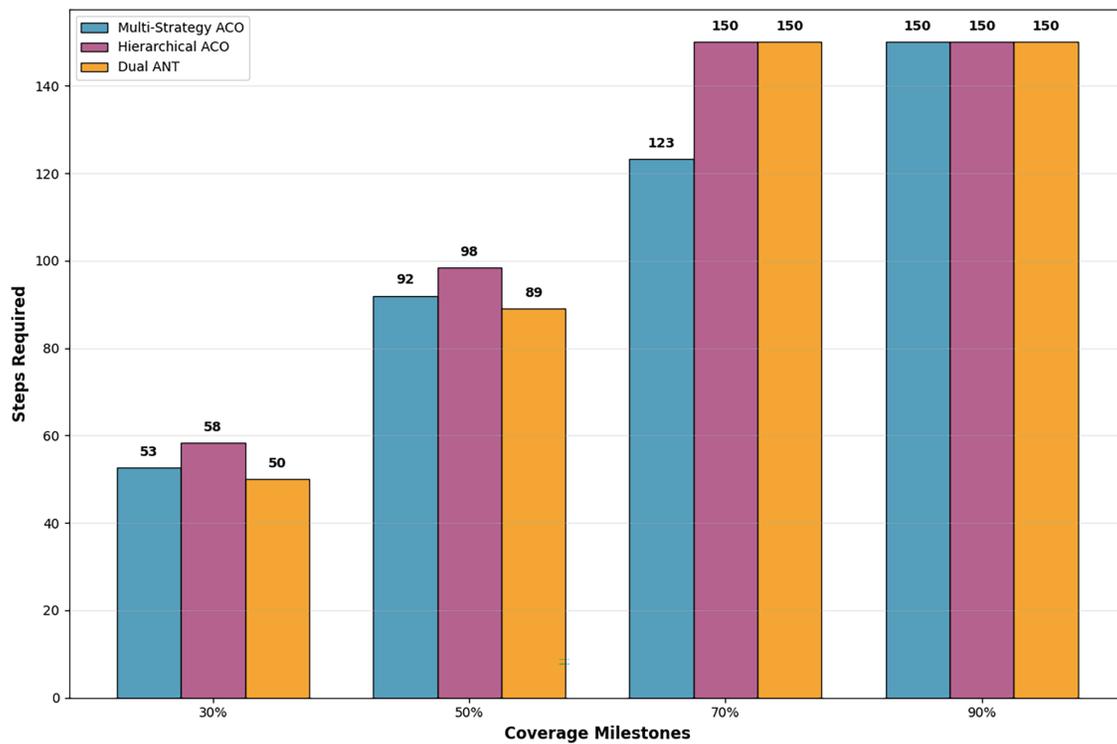
**Figure 3:** Steps to complete all targets [10,11]

Fig. 4 shows a comparison of coverage between different optimization strategies (Multi-strategy ACO [10], Hierarchical ACO [11], and Proposed Dual ANT Framework). This shows how well each algorithm explores and covers the entire environment over the simulation steps. The percentage of the area covered is represented on the  $y$ -axis and number of simulation steps is represented on the  $x$ -axis. The steeper the gradient, the faster the exploration and better the search capability. From Fig. 4, we observe that the Dual ANT Framework is faster than the two benchmark models. The Dual ANT mechanism allows agents to adjust their exploration intensity and direction in response to feedback from the environment. This leads to an optimized scan of the area and effective target identification. As the simulation progressed, all the algorithms encountered a diminishing coverage gain curve, constrained by the increasingly explored environment. Furthermore, Dual ANT maintains a competitive performance throughout and achieves a stable plateau at  $\sim 60\%$  coverage before converging. In contrast, Multi-Strategy ACO explores more aggressively initially with higher variance and slower convergence, as well as stabilizes slowly. Hierarchical ACO searches for solutions more steadily with a slower overall search.



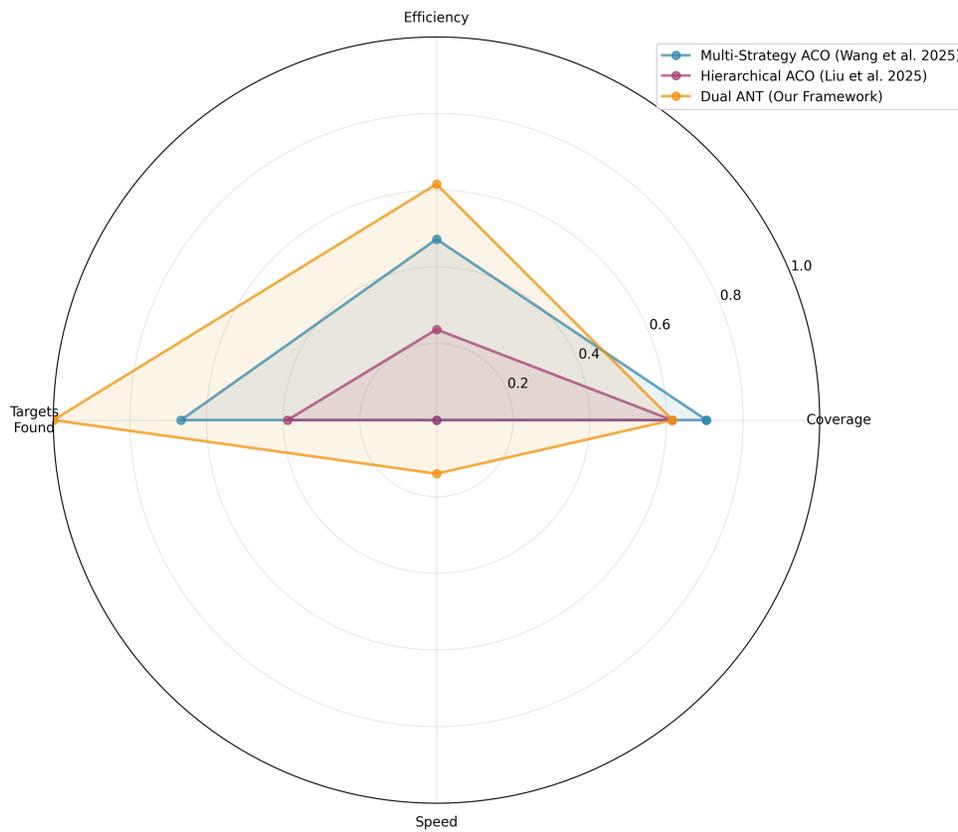
**Figure 4:** Coverage progression over time [10,11]

Fig. 5 illustrates the convergence speed of three algorithms—Multi-Strategy ACO [10], Hierarchical ACO [11], and Dual ANT (our framework) by comparing the number of steps required to achieve various coverage milestones (30%, 50%, 70%, and 90%). Because this metric measures the steps required, lower numbers indicate quicker convergence and thus better performance. The results show that the Dual ANT Framework consistently converged more rapidly at all coverage milestones. It takes only 50 steps to reach 30% coverage and 89 steps to achieve 50%, surpassing both Multi-Strategy ACO (53 and 92 steps, respectively) and Hierarchical ACO (58 and 98 steps, respectively). This demonstrates the ability of the framework to make continuous decisions on exploration and exploitation at the start of the search. Furthermore, at higher milestones (70% and 90%), all the algorithms tended to converge around the same number of steps (~150), reflecting that the exploration was saturated after covering most parts of the environment. However, Dual ANT can still get a little bit ahead on hitting intermediate coverage more efficiently and displays its early stage adaptation and faster decision making. In general, the results confirm that Dual ANT can converge faster with better scalability, especially in the early and middle exploration stages, which is an important characteristic for space-time-aware multi-agent coordination tasks.



**Figure 5:** Convergence speed: steps to reach coverage milestones

Fig. 6 illustrates a comprehensive performance comparison between the Multi-Strategy ACO [10], Hierarchical ACO [11], and the newly proposed Dual ANT Framework, assessed using four criteria: Efficiency, Coverage, Speed, and Targets Found. The greater the area on the radar plot, the better the overall algorithm performance. In Fig. 6, we can observe that the Dual ANT Framework covers the maximum area and thus performs better according to most metrics. Dual ANT is efficient in reaching all targets, demonstrating its success in reconciling exploration and exploitation. This improvement is attributed to the dual neural adaptation mechanism which adjusts the agent's behavior and the influence of pheromones, leading to more intelligent search paths. Dual ANT also produces a similar coverage performance to Multi-Strategy ACO, which demonstrates that they can perform a good exploration of the search space. In contrast, Dual ANT achieves it with fewer redundant moves and a more balanced space distribution of agents, as indicated by the efficiency and step count which are higher for parallel than for other shapes in all previous figures. In terms of speed, although Dual ANT is somewhat slower than Multi-Strategy ACO, the reason is its adaptive action section, which sacrifices minor delay time but provides more intelligent and coordinated exploration. Conversely, Hierarchical ACO has the smallest enclosed area, indicating that it restricts flexibility. In general, the radar chart clearly illustrates that Dual ANT Framework achieves the most balance and firmness in terms of efficiency, task completion, and adaptability. It exhibits great potential for scalable intelligent swarm coordination and outperforms conventional ACO-based approaches in terms of accuracy and operational robustness.



**Figure 6:** Multi dimensionnel performance radar chart [10,11]

Although our empirical results demonstrate the stable and reliable convergence of the Dual ANT in a variety of challenging environments, we cannot provide formal convergence guarantees. Rather, what the framework shows is empirical robustness—performance that works and holds in different scenarios and with different random seeds. Establishing theoretical convergence proofs under differential privacy noise in the case of dual-network adaptation is an interesting open problem for future research.

The Dual ANT framework has several advantages over existing methods.

(i) **Autonomous Adaptation:** The tuning process is performed continuously rather than being carried out by humans, leading to a fast adaptation to different structures of the environments, distribution of the agents, or density for targets.

(ii) **Robustness to Dynamic Disturbances:** By implementing introspective efficiency evaluation and Lévy flight escape behavior, the scheme can quickly recognize stagnation and adapt the behavior of exploration, leading to persistently high efficiency (even in the face of suddenly changing environments).

(iii) **Scalability:** The computational complexity of  $O(n+G)$  per step is scalable with large agent populations, and grid environments are significantly larger than other hierarchical or centralized systems.

(iv) **Privacy preservation:** With local differential privacy, agent-level trajectory information is preserved without affecting the convergence of the swarm.

(v) **Superior Empirical Performance:** Experiments demonstrate that Dual ANT achieves better coverage efficiency during the entire process, a stronger early good convergence rate, and is more stable than Multi-Strategy ACO and Hierarchical ACO.

Together, these benefits make the Dual ANT a promising candidate for practical deployment with adaptive, scalable, and privacy-respecting swarm coordination.

## 5 Conclusion

This study introduced Dual ANT, a meta-adaptive extension of ACO that integrates dual neural networks, online parameter tuning, and locally differentially private pheromone updates to enhance performance. By combining a forward network for real-time efficiency estimation with an inverse network for adaptive parameter tuning, the framework establishes a closed feedback loop that continuously refines the exploration–exploitation balance during the execution. The introduction of the Lévy flight escape mechanism strengthens the robustness against stagnation, and the local DP scheme avoids centralized trust to protect the agent trajectory information. On various simulated tasks, Dual ANT achieved significantly faster convergence, better scalability on both the number of agents and grid size, and stronger resistance to time-varying disturbances than ACO-based baselines. The approach was robust under a high rate of adaptation by utilizing bounded parameter updates, hysteresis conditions, and conservative rates of adaptation. However, the introduction of noise in the pheromone landscape by the privacy mechanism is compensated for by the meta-adaptive controller, and the collective behavior becomes robust despite a moderate level of privacy.

The performance is promising in practice, although formal convergence analysis and a deeper understanding of the privacy–performance relationship remain unresolved issues. Future efforts will involve applying the Dual ANT to continuous spaces, physical multi-robot systems, and heterogeneous-agent scenarios, and investigating other introspection metrics and online learning approaches. In summary, the Dual ANT framework is a flexible, scalable, and privacy-preserving basis for next-generation distributed coordination algorithms.

**Acknowledgement:** The authors extend their appreciation to the Deanship of Scientific Research at Northern Border University, Arar, KSA for funding this research work through the project number “NBU-FFR-2026-2441-02”.

**Funding Statement:** This research was funded by the Deanship of Scientific Research at Northern Border University, Arar, Saudi Arabia, under project number NBU-FFR-2026-2441-02.

**Author Contributions:** Conceptualization: Atef Gharbi; methodology: Atef Gharbi and Mohamed Ayari; software: Nadhir Ben Halima; validation: Ahmad Alshammari and Zeineb Klai; formal analysis: Atef Gharbi; investigation: Mohamed Ayari; resources: Nasser Albalawi; writing—original draft preparation: Atef Gharbi; writing—review and editing: all authors; visualization: Nadhir Ben Halima; funding acquisition: Nasser Albalawi and Ahmad Alshammari. All authors reviewed and approved the final version of the manuscript.

**Availability of Data and Materials:** The data that support the findings of this study are available from the Corresponding Author, Nadhir Ben Halima, upon reasonable request.

**Ethics Approval:** Not applicable.

**Conflicts of Interest:** The authors declare no conflicts of interest to report regarding the present study.

## References

1. Mancy H, Ghannam NE, Abozeid A, Taloba AI. Decentralized multi-agent federated and reinforcement learning for smart water management and disaster response. *Alex Eng J.* 2025;126(7):8–29. doi:10.1016/j.aej.2025.04.033.

2. Zhang K, Kong L, Zhang K, Huang C, Kuang Y, Huang X, et al. Privacy-aware multi-agent deep reinforcement learning for ethical coordinated control in smart regional grids. *Array*. 2026;29(2):100585. doi:10.1016/j.array.2025.100585.
3. Baccour E, Erbad A, Mohamed A, Hamdi M, Guizani M. Multi-agent reinforcement learning for privacy-aware distributed CNN in heterogeneous IoT surveillance systems. *J Netw Comput Appl*. 2024;230(3):103933. doi:10.1016/j.jnca.2024.103933.
4. Awadallah MA, Makhadmeh SN, Al-Betar MA, Dalbah LM, Al-Redhaei A, Kouka S, et al. Multi-objective ant colony optimization: review. *Arch Computat Methods Eng*. 2025;32(2):995–1037. doi:10.1007/s11831-024-10178-4.
5. Jiang C, Wang K, Zou Y, Gou R. A particle swarm-ant colony fusion algorithm for path planning based on chaotic perturbation and adaptive pheromone adjustment. *J Comput Meth Sci Eng*. 2025;57(18):14727978251366518. doi:10.1177/14727978251366518.
6. Raja V. Exploring challenges and solutions in cloud computing: a review of data security and privacy concerns. *J Artif Intell Gen Sci JAIGS ISSN 3006 4023*. 2024;4(1):121–44. doi:10.60087/jaigs.v4i1.86.
7. Zhao Y, Du JT, Chen J. Scenario-based adaptations of differential privacy: a technical survey. *ACM Comput Surv*. 2024;56(8):1–39. doi:10.1145/3651153.
8. Deng W, Zhang L, Zhou X, Zhou Y, Sun Y, Zhu W, et al. Multi-strategy particle swarm and ant colony hybrid optimization for airport taxiway planning problem. *Inf Sci*. 2022;612(5):576–93. doi:10.1016/j.ins.2022.08.115.
9. Yang X, Zhu D, Yang C, Zhou C. H-ACO with consecutive bases pairing constraint for designing DNA sequences. *Interdiscip Sci*. 2024;16(3):593–607. doi:10.1007/s12539-024-00614-1.
10. Wang X, Wang J, Cao J, Sun R. Research on mobile robot path planning based on multi-strategy improved ant colony algorithm. *Robotica*. 2025;43(10):3594–614. doi:10.1017/s0263574725102610.
11. Liu H, Wang Y, Pei S, Sun G, Yao W. An efficient hierarchical planner for autonomous exploration based on ant colony path optimization. *IEEE Sens J*. 2025;25(19):36607–16. doi:10.1109/JSEN.2025.3599173.
12. Verma AK, Dubey AK. Enhancing frequent itemset mining through machine learning and nature-inspired algorithms: a comprehensive review. *Int J Adv Comput Res*. 2024;14(68):97. doi:10.1109/icmlc.2007.4370245.
13. Bany Salameh H, Hussienat A, Alhafnawi M, Al-Ajlouni A. Autonomous UAV-based surveillance system for multi-target detection using reinforcement learning. *Clust Comput*. 2024;27(7):9381–94. doi:10.1007/s10586-024-04452-0.
14. Liu B, Chua PC, Lee J, Moon SK, Lopez M. A digital twin emulator for production performance prediction and optimization using multi-scale IDCNN ensemble and surrogate models. *J Intell Manuf*. 2024:1–19. doi:10.1007/s10845-024-02545-6.
15. Akbari E, Zeng K. Federated deep reinforcement learning for privacy-preserving spectrum sharing among MNOs. In: 2025 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN); 2025 May 12–15; London, UK. p. 1–8. doi:10.1109/DySPAN64764.2025.11115932.
16. Baharisangari N, Saravanane N, Xu Z. Distributed differentially private control synthesis for multi-agent systems with metric temporal logic specifications. In: 2024 American Control Conference (ACC); 2024 Jul 10–12; Toronto, ON, Canada. p. 4289–95.