2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Task Planning for Robotic Disinfection Using Generative-Adversary-Trimodel (GAT)

Jiajie Ye, Yongji Sheng, Tianyu Liu and Ning Xi*

Abstract— **Robotic disinfection can relieve human operators from repetitive, labor-intensive tasks while reducing the risk of pathogen transmission in public spaces. Recent advances in learning-based methods further enhance these systems by enabling robust dynamic task planning and the interpretation of ambiguous instructions. However, disinfection task planning remains a four-dimensional (interaction, logic, spatial and temporal) problem that requires expert knowledge. The robust task planning for autonomous disinfection in dynamic environment remains challenging. This paper proposes a novel framework that integrating the Generative Adversarial Trimodel (GAT) method with embodied framework to solve the four-dimensional problem in the dynamic environment. The GAT method injects expert knowledge and iteratively refines neural network-generated plans against analytical model (AM), driving dual convergence and reducing logic, spatial, and temporal errors. By combining embodied framework and the GAT method into a GAT-enhanced embodied framework, the robot system autonomously perceives objects of unknown shape and pose, long-horizon task sequence plans, and executes disinfection operations. Experimental results demonstrate an improvement in success rate and reduce the average task time and rule violation rates compared with non-GAT methods, demonstrating improved robustness and efficiency in dynamic environment.**

I. INTRODUCTION

Disinfection is a critical measure for mitigating pathogen transmission in public and clinical spaces, with documented demand surging during outbreaks and in high-traffic environments [1]. In addition, the disinfection task is a highly repetitive task that is performed with great frequency in public spaces and requires a certain level of expert knowledge. Translating these requirements to robotic systems is compelling because robots can reduce human exposure risk, alleviate repetitive labor, and improve efficiency. However, robotic disinfection task planning must satisfy the following requirements: (1) task-relevant reasoning ability under incomplete or ambiguous user instructions; (2) highly efficient long-horizon task sequence under logical, spatial, and temporal constraints; (3) safety and rule compliance (e.g., constraints on UVC lamp activation, bystander avoidance); and (4) adaptability to dynamic environments in which the number, types, and poses of objects change over time.

The crucial challenge of disinfection task planning in dynamic environments centers on four-dimensions: interaction, logic, spatial, and temporal. In addition, the task sequences produced by the disinfection task planner must satisfy logic, spatial, and temporal constraints. Specifically:

- Interaction Level: perform instruction grounding, i.e., translate high-level directives into concrete, executable task specifications;
- Logic Level: determine the interaction targets and establish the precedence relations among subtasks;
- Spatial Level: compute the optimal task sequence to minimize the overall travel;
- Temporal Level: ensure that execution meets timing and dosage requirements.

Addressing these dimensional problems requires the incorporation of domain-specific expertise.

Traditional task planning rooted in symbolic AI, for example, STRIPS-style operators [2] and HTNs [3] offers interpretability and discrete efficiency, but struggles to encode coupled constraints and adapt in dynamic environment with high cost of maintaining rules. It typically demands extensive manual programming and continual updates of operator preconditions and constraints. Integrated Task and Motion Planning (TAMP) [4] narrows the task–motion planning gap, yet is computationally heavy and sensitive to scene changes, while temporal planners for durative, resource-bounded actions [5] schedule time/resources but can produce logically valid yet physically infeasible plans when geometry and irradiation physics are weakly coupled. In disinfection, rule-based pipelines and coverage-based exposure optimization [6] improve structure under static environment but remain brittle to object/pose changes and incomplete maps, leading to plan–execution mismatches and dose/time violations.

Recently, LLMs/Vision-Language-Aciton (VLA) based planners have shown advanced task planning performance, generating reasonable logic output (task sequences). However, they struggle with strict spatio-temporal constraints and dynamic uncertainty. Voxposer [7] demonstrates how LLMs reasoning coupled with Vision-Language-Models (VLMs) perception can ground commands into actions for zero-shot execution. However, the LLM component primarily operates at the logic level, decomposing tasks into subtasks, while it relies on the VLM to handle the spatial level problems and leaves temporal constraints unaddressed. As a result, the generated subtask sequences provide limited guarantees for expert-dependent planning and are insufficient for disinfection, especially when long-horizon task sequencing with spatial and temporal requirements is critical. Large robotic datasets such as X-embodiment [8] and RoboNet [9] expand general skills through imitation, while learning-from-demonstrations approaches [10] promote human-like behaviors; however, these methods typically require retraining for specialized skills and lack mechanisms for constraint-

All authors are from the Department of Data and Systems Engineering, The University of Hong Kong, Pokfulam, Hong Kong, China (e-mail: xining@hku.hk).

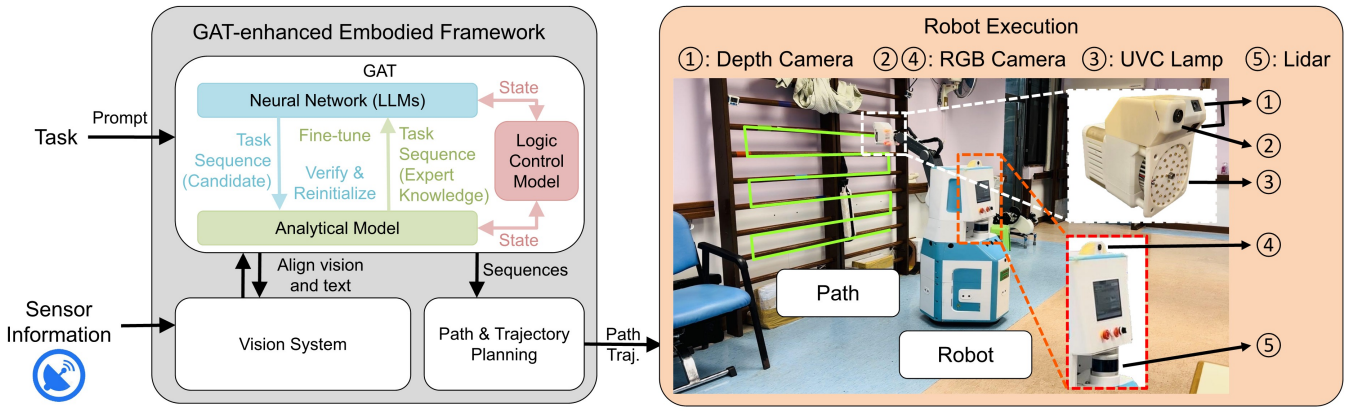

Fig. 1: The system (gray block) converts task requirements into a prompt and feeds it to the GAT. Assisted by vision system, which provide target-related information, the components within the GAT—LLMs and the analytical model—iteratively converge to generate task sequences. Path and trajectory planning are then performed to produce the corresponding path and trajectory. Finally, the robot (orange block) executes the disinfection operations following these plans.

feasible long-horizon task planning. The VLA method, RT-2 [11], adopts visual–language pretraining to directly predict actions from observations. It issues step-by-step actions, each requiring a new observation and a network forward pass, which is computationally and temporally costly. Moreover, the resulting action sequences are not continuous, making them unsuitable for disinfection tasks.

Traditional methods can address logic, spatial, and temporal dimensions in static environment by relying on prespecified rules. However, in dynamic environments their performance degrades: uncertainty necessitates time-consuming replanning and frequent rule updates or reconstructions to accommodate environmental changes. In contrast, learning-based approaches exhibit strong logical reasoning and can decompose tasks into coherent subtasks at the logic level, but they typically require embodiment-based framework to resolve spatial and temporal aspects (e.g., waypoint/subtask optimization, timing/dosage allocation) and to remain robust under real-world dynamics.

To address these limitations, we introduce a GAT method and GAT-enhanced embodiment framework to meet the disinfection task requirements and solve the four-dimensional problem. In summary, the main contributions of this paper are summarized as follows:

- A GAT method that iteratively refines LLMs plans against logic/spatial/temporal analytical constraints, achieving dual convergence and fewer logic, spatial, and temporal violations.
- Using the GAT method that injects expert knowledge to enhance the efficiency of disinfection task plan, especially in long-horizon task sequence planning.
- Proposes a GAT-enhanced Embodied Framework for planning and execution of disinfection tasks in dynamic environments, addressing the four-dimensional problems — interaction, logic, spatial, and temporal.

II. METHODOLOGY

The pipeline comprises a GAT-enhanced embodied framework (gray block) and a robot system (orange block), as shown in Fig. 1. Given a task prompt and multimodal sensor inputs (RGB image, depth information, LiDAR), visison system align visual and textual cues to form a scene representation. Within GAT, a neural network (LLMs) generates candidate task sequences, while an AM verifies and refines them, transfers expert knowledge, and triggers fine-tuning of the LLMs during the iteration. The validated task sequences are then provided to a path and trajectory planner to generate executable motions. The robot subsequently follows the planned paths and trajectories to perform surface disinfection. Details of the GAT method and the GAT-enhanced embodied framework are presented below.

A. GAT Method

The GAT method is defined as an integrated tri-model architecture that combines neural network (NN) with symbolic models. The symbolic models consists of an analytical model and a logic control model. The core idea of the GAT method is to pits a data-driven generator against a model-driven generator in an adversarial iteration of collaboration.

Data-driven neural network:

- Leveraging large-scale data, the data-driven NN learns complex task patterns and implicit strategies to rapidly generate diverse candidate plans (e.g., for navigation, perception, and manipulation subtasks).

Model-driven analytical model:

- Explainable constraints: Explicitly encodes domain rules and physical limits (logic, spatial, temporal).
- Verification & optimization: Uses optimization and validation algorithms to evaluate candidate plans, providing actionable feedback or higher-quality alternatives.

Logic control model:

- State transitions between discrete states are event-driven and govern the iteration loop.

Unlike zero-sum Generative Adversarial Networks (GANs) [12], GAT method achieves dual convergence. The NN and AM enhance each other and converge together. This is significantly different from the game theory background of GANs. After GANs training, we obtain a generator that can generate realistic data (the generator wins), while the discriminator fails because it cannot distinguish between real and fake data (the discriminator loses). The value function proposed in the original GAN paper perfectly embodies this. In the GAT method, however, both trained models can be used independently, competing with each other to achieve the best output, creating a positive-sum game otherwise zero-sum game.

The interaction between NN and AM is shown in Algorithm 1. In a given weighted, connected graph (G), a randomly permuted task sequence is used as the initial solution. Iteration then proceeds between the NN and the AM: the NN's output initializes the AM's parameters, and the AM's solution subsequently fine-tunes the NN.

Algorithm 1: GAT Models

Input: Set of n points $P = \{(x_1, y_1, z_1), \ldots, (x_n, y_n, z_n)\}$
Output: Sequence S_c and path length L_c

Initialize:
$G \leftarrow P$ ▷ Weighted connected graph
$L_{\text{approx}} \leftarrow \text{Christofides}(G), \lambda_K \leftarrow 1$ ▷ Approximate solution
$S_0 \leftarrow \text{RandomPermutation}(G)$
$L_0 \leftarrow \text{PathLength}(S_0)$
$S_c \leftarrow S_0,\ L_c \leftarrow L_0$ ▷ Current solution
$K \leftarrow 0$ ▷ Iteration counter
while true **do**
  $S_{\text{nn}} \leftarrow \text{NN}(G, S_c, L_c)$ ▷ Neural network
  $S_{\text{am}}, \nabla \leftarrow \text{AM}(G, S_{\text{nn}}, L_{nn})$ ▷ Algorithm 2, 3
  $\nabla L_{\text{logic}}, \nabla L_{\text{temporal}} \leftarrow \nabla$ ▷ Penalty gradient
  $L_{\text{am}} \leftarrow \text{PathLength}(S_{\text{am}})$
  $S_c \leftarrow S_{\text{am}},\ L_c \leftarrow L_{\text{am}}$
  $K \leftarrow K + 1$
  $\epsilon_K \leftarrow L_{nn} - \lambda_K L_{\text{approx}}$ ▷ Error
  Update λ_K : ▷ Scaling factor
  $\lambda_K = \min\{\lambda_K - \theta_K,\ 1\},\ \theta_K \in [0, 1)$
  $\nabla L_{\text{spatial}} \leftarrow \epsilon_K$
  $\nabla L_{GAT} = \alpha \nabla L_{\text{logic}} + \beta \nabla L_{\text{spatial}} + \gamma \nabla L_{\text{temporal}}$
  if $(\epsilon_K \leq \epsilon_{min}\ and\ \nabla L_{\text{GAT}} \leq \nabla L_{min})\ or\ K > K_{max}$ **then**
    break ▷ Convergence achieved
  end if
end while
Return S_c, L_c

The iterative process is regulated by an error bound ϵ_K and the Pareto optimality condition, which serves as the stopping criterion to control convergence and terminate the loop.

The Pareto optimality condition is met:

$$\nabla L_{GAT} = \alpha \nabla L_{\text{logic}} + \beta \nabla L_{\text{spatial}} + \gamma \nabla L_{\text{temporal}} \to 0,$$

where ∇L_{logic}, $\nabla L_{\text{spatial}}$ and $\nabla L_{\text{temporal}}$ denote the penalty gradients for the candidate task sequence of generator violating logic, spatial, and temporal, respectively. The resultant task sequences maximize task completion fidelity (↑ Success Rate), task time cost (↓ Average Task Time), and physical feasibility (↓ Rule Violation Count).

The AM integrates logical, spatial, and temporal constraints distributed across the respective levels.

1) Logic Constraints: Task sequence transitions are verified against task-specific preconditions and post-conditions. Typical constraints include (i) spatial reachability: The navigation subtask must precede the disinfection subtask, because the robot must first move to the target vicinity before disinfection can begin (e.g., $a_1 = NAVIGATE(x, y)$ before $a_2 = DISINFECT(obj)$); and (ii) safety requirements: The UVC lamp must be activated prior to disinfection subtask and deactivated at the end of the task to prevent harm to bystanders. If the current sequence violates the logic constraints, the logic violation count increase ($\nabla L_{\text{logic}}\uparrow$).

2) Spatial Rulesets: Objective function is to minimize total path length with constraints. Initial task sequences is $S_c = \{a_0, a_1, \ldots, a_N\}$. The spatial level algorithm is shown in Algorithm 2. Given a weighted, connected graph (G), a high-quality initial solution is first obtained using the classical Christofides algorithm [13]. During the iterative loop, whenever a superior solution L_c is identified, L_c is adopted as the new initial sequence and a local search procedure is applied. Specifically, the Lin–Kernighan heuristic is employed rather than more time-consuming metaheuristics such as ant colony optimization or particle swarm optimization, prioritizing a superior quality–time trade-off. This choice is motivated by the need to enable rapid iterations between the AM and NN. The procedure returns the local optimal task sequence under the given initialization.

Algorithm 2: Analytical Model — Spatial Constraints

Input: Current sequence G, S_c, L_c
Output: New sequence S_{new}

procedure MAIN
  $S_{chris} \leftarrow \text{Christofides}(G)$ ▷ Approximate solution
  if $L_{chris} < L_c$ **then**
    $S_c \leftarrow S_{chris},\ L_c \leftarrow L_{chris}$
  end if
  $S_c \leftarrow \text{LOCALSEARCH}(S_c,\ start\_idx,\ G)$
  $L_c \leftarrow \text{Length}(S_c)$
  if $L_c < L_{new}$ **then**
    $S_{new} \leftarrow S_c$
    $L_{new} \leftarrow L_c$
  end if
  return $S_{\text{new}}, L_{\text{new}}$
end procedure

3) Temporal Constraints: Verify the total time cost not out of the limitation when task having time requirement. The temporal constrains algorithm is shown in Algorithm 3 .

Algorithm 3: Analytical Model — Temporal Constraints

Input: Task sequences S_c, Maximum time limit $T_{\max}$, Max spatial iterations $I_{\max}$, Current spatial iteration count i
Output: Adjusted task sequences S_{adj}

procedure TEMPORALCONSTRAINTS($S_c, T_{\max}, I_{\max}, i$)
  total_time ← calculate_total_time(S_c)
  Time constraints satisfied:
  if total_time $\leq T_{\max}$ **then**
    return S_c
  else
    $\nabla L_{\text{temporal}} = total\_time - T_{max}$
    Request rescheduling:
    if $i < I_{\max}$ **then**
      return to spatial level
    else
      Find the least used object in history data:
      $obj_{Least} \leftarrow$ Find_object($data_{history}$)
      Remove the object from sequence:
      $S_{\text{adj}} \leftarrow$ remove_obj_seq(S_c, obj_{Least})
      Check logical consistency:
      return to logic level
      Re-optimize sequence:
      re-enter spatial level
    end if
  end if
end procedure

The total execution time of each candidate sequence is evaluated, and if it exceeds the time budget, re-planning is triggered at the spatial layer. Once the spatial layer reaches its iteration limit, the lowest-usage target, determined by a Poisson prior from historical data, is removed, after which the logic layer verifies consistency and the spatial layer performs re-optimization.

B. GAT-enhanced Embodied Framework

The GAT-enhanced Embodied Framework is designed as a hierarchical architecture, as illustrated in Fig. 2. Robot tasks can be organized into five levels: interaction, logic, spatial, temporal, and control. The disinfection robot system follows this five-level structure:

Interaction Level: This level processes user instructions and environmental information into prompts. In addition, ensure UVC dosage compliance using $t_{uvc,i} \geq \frac{\Phi_{\text{target}}}{E_{\text{eff}}(obj_i)}$, where t_{uvc} represents the UVC exposure time required for the robot to effectively disinfect the target object; Φ_{target} represents the UVC dosage required on the surface of the target object, as defined by disinfection standards [14]; $E_{\text{eff}}(obj_i)$ represents the effective irradiance intensity, which depends on the distance between the UVC and object's surface. Then, UVC dosage can be evaluated by $D_{\text{UV}} = \int_0^t E_{\text{eff}}(obj_i)dt$.

Logic Level: This level is responsible for all logical operations and comprises components such as LLMs and a finite state machine (FSM) module, where the LLMs serve as the NN component of the GAT method and the FSM implements its logic control model.The LLMs perform logical reasoning, decision-making, and disinfection task planning, while the FSM maintains the robot's discrete states, coordinates state transitions, and manages the execution of task sequences and operational phases.

Spatial Level: This level processes spatial information and extracts key data to enhance the performance of other levels. The vision system incorporates multiple vision processing algorithms and models.

Temporal Level: The trajectory planner at this level generates the disinfection trajectory and designs temporal constraints within the analytical model.

Control Level: This level processes real-time sensory data and publishes physical control commands to the robot.

In dynamic environments, disinfection often constitutes a zero-shot setting: the robot lacks prior knowledge of target types, shapes, and visually similar variants, making task and motion planning difficult and ambiguous instruction grounding. Our GAT-enhanced embodied framework mitigates these issues by jointly leveraging perception, expert knowledge transfer, and planning (Fig. 3). Given a user instruction, GAT performs iterative task decomposition: it first acquires scene information from images and point clouds processed by the vision system, then selects the target object and refines the plan to extract its surface geometry for disinfection planning. Finally, it fuses instruction semantics with the estimated surface to identify planes to disinfect and outputs the execution task sequence.

III. EXPERIMENTS SETUP AND RESULTS

A. System Implementation

1) Hardware System: The disinfection system in Fig. 1 is a 6-DOF mobile manipulator with a three-wheeled 3-DOF base and a 3-DOF arm, whose end-effector carries a UVC lamp (270 nm peak, 11.2 mW/cm²). Sensing includes a 16-line Velodyne 3D LiDAR at the midsection and three cameras: one top-mounted RGB (1920×1080, 2.7 mm) plus a co-located RGB (same specs) and a pmd flexx2 depth camera on the UVC lamp house. Computation is split between an industrial PC (Advantech UNO-238; i7-8665UE; 32 GB; Ubuntu 20.04) for control/perception and a NUC (ASUS; i7-1165G7; 32 GB) for the vision system and cloud-based LLMs interfacing.

2) Software System: The software system operates within the Robot Operating System (ROS) framework and interfaces with GPT-4/4o via their APIs for interactions with the LLMs. The prompting strategy is inspired by [15]. The vision system comprising multiple vision modules, including perception and geometry modules, such as object detection algorithm (YOLOv10 [16]), 3D object segmentation pipeline (SAM [17] with point cloud segmentation), viewpoint selection algorithm [18], and plane-fitting module (RANSAC), to meet diverse disinfection task requirements. The motion planning and navigation components are inspired by [19], [20].

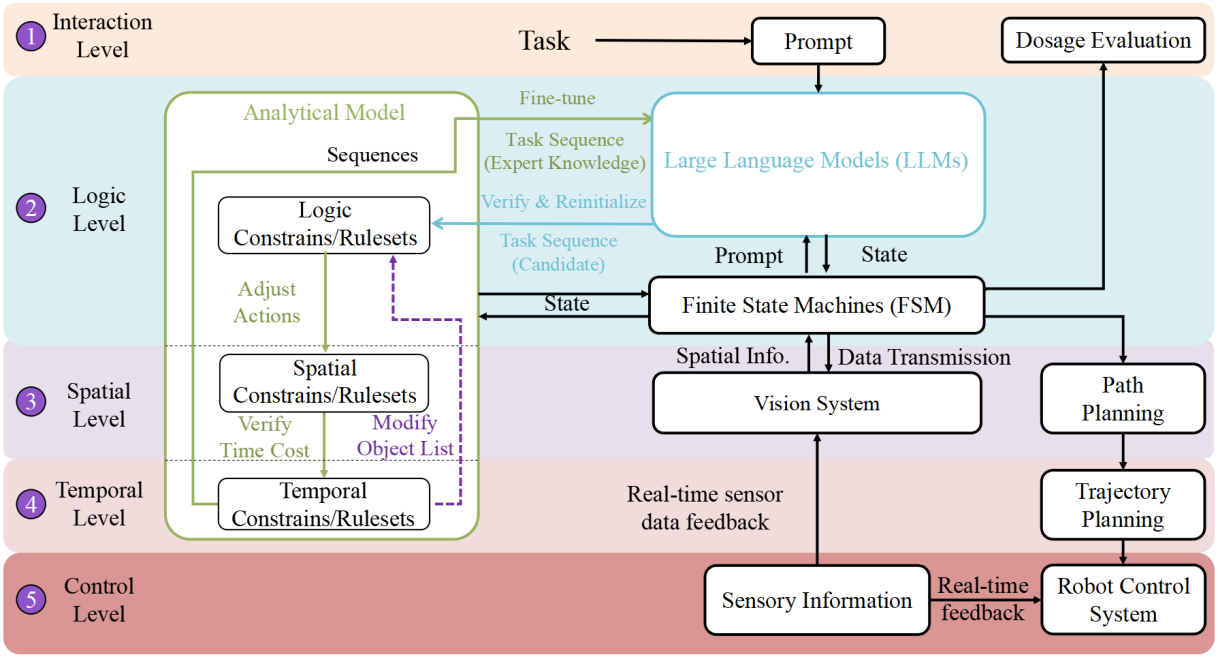

Fig. 2: GAT-enhanced Embodied Framework for Scheduling Disinfection Task. (1) Interaction Level: transfers task into prompts; the dosage evaluator estimates UVC exposure after each disinfection operation; (2) Logic Level: LLMs generate candidate task sequence into AM to for verification and possible re-initialization; expert-knowledge transfer fine-tunes the LLMs; a FSM manages the robot states; (3) Spatial Level: vision system provide perception modules, and a path planner generates disinfection paths; (4) Temporal Level: the trajectory planner generates trajectories that satisfy the dosage requirements; (5) Control Level: real-time sensor data are fed upward for processing and used for low-level robot control.

B. Experimental Protocol

1) Implementation Details: The experiments are conducted on the mobile manipulator shown in Fig. 1. The experimental venue is a training room in an elderly care center with randomly distributed objects, including hand/leg training devices, tables, chairs, and racks. Dynamic factors include an unknown number of objects, varying poses, and shape variations within the same object category.

Two task classes are considered: patrolling and disinfection. In patrolling, waypoint orderings are generated by either an LLM or the GAT method; no time constraints are imposed, and all waypoints are reachable navigation goals. Disinfection is performed in a dynamic environment with a variable number of target objects of diverse types and shapes. Prior knowledge is available for a subset of targets, either instance specific or type level across shape variations, and is represented in text, image, and point cloud modalities. The robot is assumed to have sufficient battery capacity to complete all tasks.

2) Experiments Setup: To evaluate the proposed GAT method and the GAT-enhanced embodied framework, two experiments were designed. Details are as follows:

- *E1 — Measurement the efficiency of the GAT method for task planning:* In the public areas of an elderly care center, the robot is assigned substantial daily patrolling tasks. Two patrolling scenarios with different numbers of waypoints (51 and 125) are constructed. Task planning is performed using an LLM only approach and the GAT method for comparison, in order to quantify planning efficiency. The evaluation metric is:
  - **Path Length (PL)**: the total distance traveled to visit all patrolling points.
- *E2 — Assessment of GAT-enhanced Embodied Framework in dynamic environment:* Fifteen instructions were tested across three scenarios: with prior knowledge, without prior knowledge, and zero-shot learning. These instructions pertained to at least 10 randomly chosen objects within a venue, including chairs, training devices, and racks. Some instructions targeted specific components of these objects, such as the back of a chair. Five different methods were compared to evaluate task planning performance. The first method utilizes LLMs with a static objects process module, which is suitable for generally immobile objects. The second method integrates LLMs with mechanisms for operating in dynamic environments [21]. The third method utilizes an embodied assistant with multi-level task feedback mechanism [22]. The fourth approach is rule-based, relying on pre-set rules to complete tasks step-by-step, with rules remaining unchanged during the task. The last one is the proposed method.
  To evaluate the performance of different approaches for disinfection task planning, three metrics are selected:
  - **Success Rate (SR):** The completion of the full task without human intervention, while autonomously

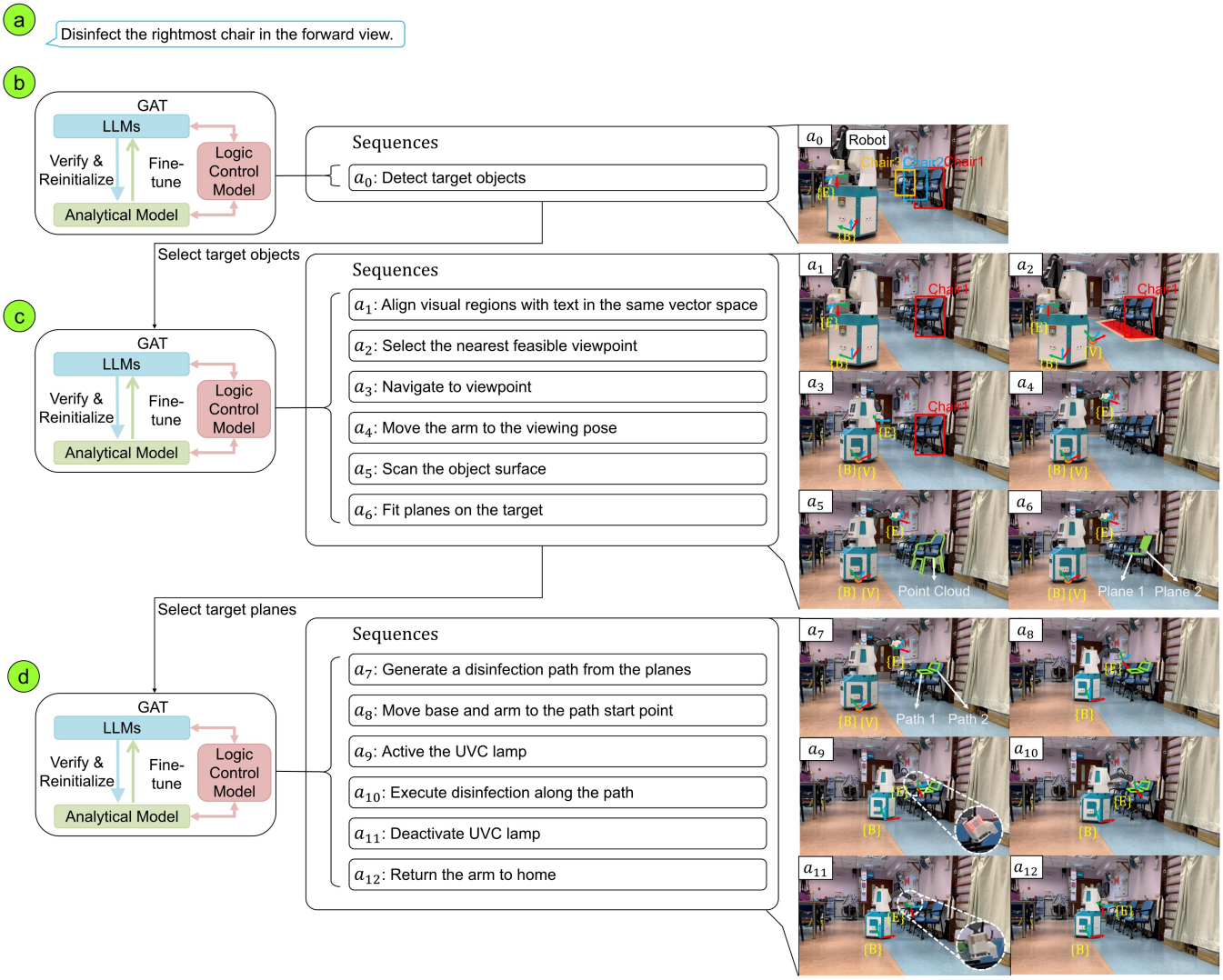

Fig. 3: (a) The task requirements are translated into a prompt format; (b) The GAT generates an initial task sequence to acquire environmental information; (c) Leveraging objects' spatial information and the relative positional relationships to the target, the GAT selects the target objects and generates a new task sequence to acquire their surface information; (d) Based on the planes information and task requirements, the GAT selects target planes and generates a new task sequence to disinfect the target surfaces.

adjusting actions to meet task requirements under the influence of dynamic environmental changes.

- **Average Task Time (ATT):** The average time cost taken from command input to the completion of the last subtask, calculated over multiple task executions.
- **Rule Violation Count (RVC):** The total number of constraint violation per trial.

C. Experimental Results and Discussion

1) Measurement the efficiency of GAT Method: A comparison between a LLMs only approach and the GAT method (Fig. 4) under two subtask scales shows that the LLMs only approach fails to converge to improved sequences, whereas GAT consistently yields shorter paths. For 51 subtasks, the GAT achieves 77.50 m versus 261.978 m for the LLMs only approach; for 125 subtasks, it achieves 95.489 m versus 733.072 m. The LLMs only approach tends to stall in poor permutations as task size increases, while GAT method coupling of LLMs generation with AM verification steadily suppresses long inter-cluster jumps and accumulates reusable ordering. This yields stable convergence and large distance savings at both scales (77.50 m vs. 261.978 m for 51 subtasks; 95.489 m vs. 733.072 m for 125), implying shorter cycles, lower energy use, and time-saving.

2) Assessment of GAT-enhanced Embodied Framework: As shown in Tab. I, the proposed GAT-enhanced framework outperforms all baselines across SR, ATT, and RVC.

SR: The GAT-enhanced embodied framework matches or exceeds embodiment-based methods and substantially outperforms LLM-only and rule-based baselines. With prior knowledge, SR reaches 93.3% (on par with Embodiment), and under no prior and zero-shot settings it attains 86.7%

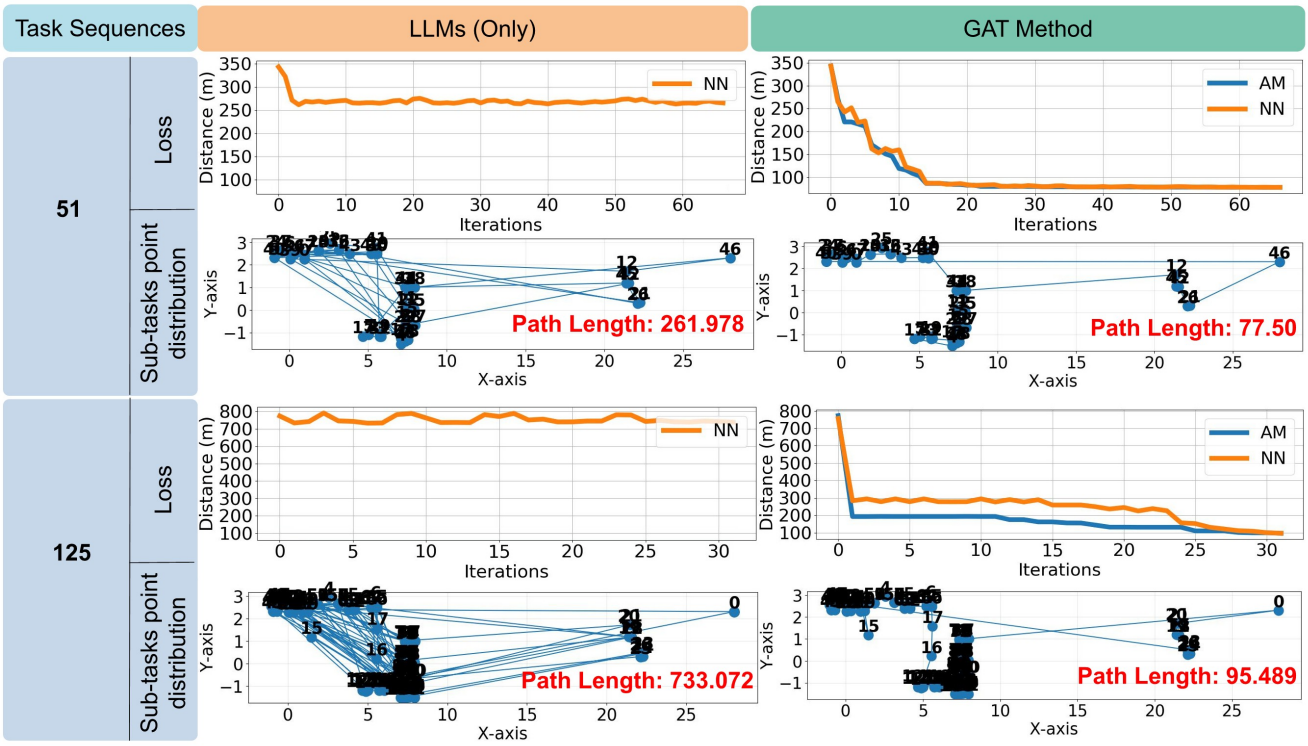

Fig. 4: Compare the proposed GAT method with the LLM-only baseline on two task (51 and 125 sub-tasks). Observe that GAT reduces loss much faster and further: AM and NN curves drop sharply within 10–15 iterations and keep converging, while the baseline plateaus high. Translate this into execution efficiency: with 51 sub-tasks, path length falls from 261.98 m (LLMs only) to 77.50 m (GAT); with 125 sub-tasks, from 733.07 m to 95.49 m. Note that GAT also yields more coherent sub-task visitation, avoiding the long cross-scene traversals seen in the baseline, leading to more efficient planning as task complexity grows.

and 80.0%, respectively, well above LLM+Stat. (13.3% with prior) and Rule-based (6.67% with prior).

ATT: Relative to Embodiment, the proposed method shortens ATT to 37.4 min with prior knowledge from 45.3 min, 46.8 min with no prior from 59.7 min, and 51.7 min in zero-shot from 63.4 min, i.e., reductions of 17.4%, 21.6%, and 18.45%. It is also faster than LLM+Dyn. with prior knowledge (37.4 vs. 49.3 min).

RVC: The proposed method achieves the lowest RVC across all settings: 4 (with prior), 8 (no prior), and 8 (zero-shot). This improves substantially over Embodiment (7/13/13) and dramatically over Rule-based (28/40/40). Relative to Rule-based, RVC drops by about 85.7% (with prior) and 80.0% (zero-shot).

Although ATT increases when prior knowledge is absent due to iterative LLM refinement and 5 to 12 s API latency, the GAT enhanced embodied framework still achieves lower ATT than all non GAT baselines and consistently yields the fewest rule violations. Overall, the proposed integration preserves high SR, shortens completion time, and markedly improves rule adherence relative to alternative methods.

During the experiment, the statistical results of error rates at different layers are shown in Fig. 5. Compared with rule-based methods, data-driven methods (LLMs) perform better

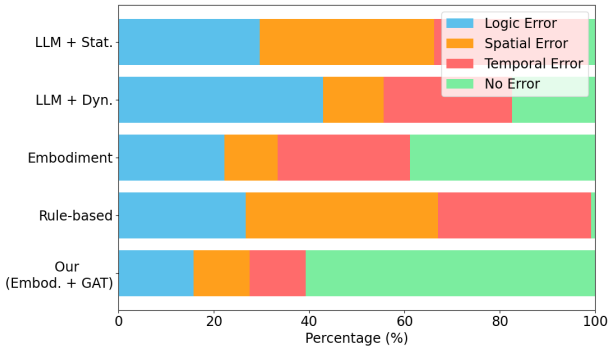

Fig. 5: Error rate of GAT-enhanced embodied framework and baselines. Compared to baselines, although the proposed method, it effectively reduces the error rate in different levels.

in dynamic environments, reducing errors at multiple levels and improving robot adaptability. The embodiment-based framework also achieves higher no-error rates, indicating that LLMs combined with robot-derived feedback can effectively mitigate logical, spatial, and temporal errors. Our method attains the lowest overall error rate, with the most significant source of error occurring at the logic level.

TABLE I: Comparison Results with different methods

	Randomly Distributed Objects								
	SR. (%)			ATT. (min)			RVC.		
Method	W/ Prior	No Prior	Zero-shot	W/ Prior	No Prior	Zero-shot	W/ Prior	No Prior	Zero-shot
LLM+Stat.	13.3	-	-	44.4	-	-	22	22	22
LLM+Dyn. [23]	40.0	20.0	-	49.3	51.1	-	7	17	28
Embodiment [22]	**93.3**	80.0	66.7	45.3	59.7	63.4	7	13	13
Rule-based [24]	6.67	-	-	53.0	-	-	28	40	40
Ours	**93.3**	**86.7**	**80.0**	**37.4**	**46.8**	**51.7**	**4**	**8**	**8**

IV. CONCLUSIONS

This work proposes a GAT method and a GAT enhanced embodied framework for disinfection planning on a mobile manipulator. The GAT generated task sequences satisfy disinfection requirements, while the framework addresses four dimensional challenges to enable autonomous disinfection in dynamic environments. In contrast, embodiment only approaches can yield logically incorrect outputs or inefficient long horizon plans when specialized domain knowledge is required. A key component of GAT method is the AM, which encodes multi-level constraints and rules to support knowledge implementation and transfer. Experiments show that this design effectively transfers knowledge to LLMs, significantly reducing errors and yielding more reasonable and reliable logical outputs. However, GAT method has limitations. Formulating physical models into the rules or constraints of AM can be challenging in certain cases. For example, transferring manipulation skills for non-rigid objects remains challenging due to their complex properties. Such knowledge may instead be acquired through trial, evaluation, and experience summarization to better understand the physical model.

ACKNOWLEDGMENT

The work described in this paper was partially supported by a grant from the University Grants Committee, Hong Kong, China (C7100-22GF, C7174-20GF, 17203024).

REFERENCES

[1] M. Biasin, A. Bianco, G. Pareschi, A. Cavalleri, C. Cavatorta, C. Fenizia, P. Galli, L. Lessio, M. Lualdi, E. Tombetti *et al.*, "Uv-c irradiation is highly effective in inactivating sars-cov-2 replication," *Scientific reports*, vol. 11, no. 1, p. 6260, 2021.

[2] M. Held and R. M. Karp, "A dynamic programming approach to sequencing problems," *Journal of the Society for Industrial and Applied mathematics*, vol. 10, no. 1, pp. 196–210, 1962.

[3] D. S. Nau, T.-C. Au, O. Ilghami, U. Kuter, J. W. Murdock, D. Wu, and F. Yaman, "Shop2: An htn planning system," *Journal of artificial intelligence research*, vol. 20, pp. 379–404, 2003.

[4] C. R. Garrett, R. Chitnis, R. Holladay, B. Kim, T. Silver, L. P. Kaelbling, and T. Lozano-Pérez, "Integrated task and motion planning," *Annual review of control, robotics, and autonomous systems*, vol. 4, no. 1, pp. 265–293, 2021.

[5] M. Cashmore, D. Magazzeni, and P. Zehtabi, "Planning for hybrid systems via satisfiability modulo theories," *Journal of Artificial Intelligence Research*, vol. 67, pp. 235–283, 2020.

[6] S. Dogru and L. Marques, "Path and trajectory planning for uv-c disinfection robots," *IEEE Robotics and Automation Letters*, vol. 8, no. 7, pp. 4099–4106, 2023.

[7] W. Huang, C. Wang, R. Zhang, Y. Li, J. Wu, and L. Fei-Fei, "Voxposer: Composable 3d value maps for robotic manipulation with language models," *arXiv preprint arXiv:2307.05973*, 2023.

[8] A. O'Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain *et al.*, "Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 6892–6903.

[9] S. Dasari, F. Ebert, S. Tian, S. Nair, B. Bucher, K. Schmeckpeper, S. Singh, S. Levine, and C. Finn, "Robonet: Large-scale multi-robot learning," *arXiv preprint arXiv:1910.11215*, 2019.

[10] A. Zhan, R. Zhao, L. Pinto, P. Abbeel, and M. Laskin, "Learning visual robotic control efficiently with contrastive pre-training and data augmentation," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 4040–4047.

[11] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," in *Conference on Robot Learning*. PMLR, 2023, pp. 2165–2183.

[12] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[13] N. Christofides, "Worst-case analysis of a new heuristic for the travelling salesman problem," in *Operations Research Forum*, vol. 3, no. 1. Springer, 2022, p. 20.

[14] E. S. Moyer, W. E. Miller, M. A. Commodore, C. C. Coffey, J. L. Hayes, S. A. Fotta, and G. Sims, "Aerosol and biological sampling of a ventilation fan-bank modified with ultraviolet germicidal irradiation and improved filter holders," *Indoor and Built Environment*, vol. 19, no. 2, pp. 230–238, 2010.

[15] N. Wake, A. Kanehira, K. Sasabuchi, J. Takamatsu, and K. Ikeuchi, "Chatgpt empowered long-step robot control in various environments: A case application," *IEEE Access*, 2023.

[16] A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han *et al.*, "Yolov10: Real-time end-to-end object detection," *Advances in Neural Information Processing Systems*, vol. 37, pp. 107 984–108 011, 2024.

[17] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment anything," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 4015–4026.

[18] J. Ye, Y. Sheng, S. Wang, Y. Ma, Z. Yang, X. Liu, and N. Xi, "Embodied generative method for scheduling disinfection tasks using robot," in *2024 IEEE 14th International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*. IEEE, 2024, pp. 422–427.

[19] M. Song, T.-J. Tarn, and N. Xi, "Integration of task scheduling, action planning, and control in robotic manufacturing systems," *Proceedings of the IEEE*, vol. 88, no. 7, pp. 1097–1107, 2002.

[20] N. Xi, *Event-based motion planning and control for robotic systems*. Washington University in St. Louis, 1993.

[21] I. Chio, K. Ruan, Z. Wu, K. I. Wong, L. M. Tam, and Q. Xu, "Design and autonomous navigation of a new indoor disinfection robot based on disinfection modeling," *IEEE Transactions on Automation Science and Engineering*, vol. 20, no. 1, pp. 649–661, 2023.

[22] Y. Guo, Y. Ni, J. Jin, Y. Jiang, D. Li, H. Zhao, and Y. Shen, "Embodied assistant: Robot mobility operations guided by open vocabulary in open environments utilizing llm," *IEEE Internet of Things Journal*, pp. 1–1, 2025.

[23] J. Ye, Y. Sheng, S. Wang, Y. Ma, C. Kwok, Q. Wang, and N. Xi, "Integrated task scheduling, action planning and control for autonomous disinfection by a mobile robotic manipulator," in *2023 WRC Symposium on Advanced Robotics and Automation (WRC SARA)*. IEEE, 2023, pp. 60–65.

[24] Z. Liu, Y. Xu, M. Jin, and S. Li, "Disinfection robots scheduling and routing problem for healthy buildings," *Journal of Building Engineering*, vol. 87, p. 108894, 2024.