

ConvNeXtV2-EMA for Multi-Robot Task Allocation on Dynamic Inspection Maps

Antoni Marciniak^{1,*}

¹ Faculty of Mechatronics and Automation, Koszalin University of Technology, Koszalin, 75-453, Poland

*Corresponding author: antoni.mar@tu.koszalin.pl

Abstract. Dynamic inspection maps change when aisles are blocked, hazards occur or new assets are identified; thus, static multi-robot assignment is no longer feasible. This paper develops a ConvNeXtV2-EMA allocation model that converts occupancy, risk, task urgency and robot-state layers into a compact spatial representation and couples it with an exponential-moving-average teacher for stable online decisions. A map-event gate limits inference to consequential changes, and a feasibility mask and a differentiable bipartite decoder enforce capability, battery and collision constraints. Tests in 2,400 simulated episodes and 180 hardware runs of 4-20 heterogeneous robots show a 94.6% task completion rate, a 12.8% reduction in mission time, and a 17.3% decrease in travelled distance compared with the best graph-attention baseline. With fewer than 30 map changes per hour, the method still reaches 91.8% completion and has a 73ms allocation latency; the 95th-percentile workload imbalance is reduced to 0.137. The ablation experiment results attribute a 3.9 percentage point improvement to the multi-scale ConvNeXtV2 encoding and a 2.7 percentage point improvement to EMA stabilization. Based on the above results, Spatially Grounded Representation Learning and Temporally Smoothed Supervision can support responsive, feasible and balanced task allocation without solving a full combinatorial problem after each map update.

Keywords: *Dynamic Inspection Map, Multi-Robot Task Allocation, ConvNeXtV2 Encoder, Exponential Moving Average, Online Replanning*

Received on 16 May 2022, Accepted on 30 September 2022, Published on 14 October 2022

Copyright © 2022 Author(s), licensed to DEA. This is an open access article distributed under the terms of the CC BY-NC-SA 4.0, which permits copying, redistributing, remixing, transformation, and building upon the material in any medium so long as the original work is properly cited.

Introduction

Inspection fleets are being deployed in substations, tunnels, warehouses and process plants increasingly frequently; here, the map is not a fixed geometric background but a living operational record. Temporary obstacles have reduced accessibility, abnormal areas have added new tasks, and the lack of charging stations has limited the effective operating area for all robots. Therefore, recent studies have treated multi-robot coordination as a joint perception, planning and scheduling problem rather than a one-shot assignment problem [1]. Map uncertainty is relatively higher in a narrow passage that is unavailable during the allocation cycle [2]. Heterogeneous sensing and mobility also prevent a nearest-robot rule from reflecting the actual execution cost [3]. Communication constraints may lead to a delay or local inconsistency in robot state estimation [4]. Given the above circumstances, the allocator is motivated to read the spatial pattern of change and respond at a speed suitable for field operations.

Traditional allocation methods explicitly set costs and optimize auctions, matching, mixed-integer programming, or metaheuristic algorithms. They are interpretable, but repeated optimization is costly under high-frequency task arrivals and map revisions [5]. Learning-based policies can amortize the search and absorb contextual features that are difficult to handle manually [6]. Graph models can represent robot-task relationships naturally, but they often simplify the environment into pairwise distances and ignore the shape of obstacles, risk areas and congestion corridors [7]. Deep reinforcement learning can learn long-term behaviors, but as the state distribution changes during training and deployment, the policy objectives will also change [8]. At present, the

engineering problem is to preserve the map geometry, ensure hard feasibility, and stabilize rapid allocation without having to convert every local update into a full fleet rescheduling.

ConvNeXtV2-EMA is used to address this problem in a work on event-driven multi-robot task allocation for dynamic inspection maps. Rasterize occupancy, change age, risk, urgency, robot pose, capability and battery information into aligned channels in the model; a lightweight ConvNeXtV2 encoder extracts multiscale context, and robot-task tokens query the spatial feature pyramid. An exponential-moving-average teacher provides temporally smoother allocation targets, and a feasibility mask and collision-aware decoder maintain operational constraints. The three main contributions are: A dynamic-map tensor that separates the permanent structure from transient variations; An event gate to prevent redundant inference; and A constrained decoder trained with assignment quality, balance and consistency objectives. Assessment includes workload scale, map volatility, sensing noise, communication loss and hardware transfer.

Related Work

Optimization-Based Multi-Robot Allocation

Most multi-robot task allocation problems can be formulated as matching or routing problems, aiming to minimize a combination of travel, service time, energy, and deadline penalties. The auction mechanism decentralizes the decision-making process and remains feasible even when communication is unreliable, while centralized integer programming provides a convenient upper bound for small-scale cases [9]. Correlation or clustering formulas can show the potential compatibility between robots and task groups [10]. Dynamic variants introduce reassignment penalties to avoid excessive route changes triggered by new optimal solutions [11]. The main limitation of the check graph is the cost model: the shortest path length is usually recalculated outside the allocator, and spatial bottlenecks and local map age information are only available as scalar summaries.

Therefore, mapping and scheduling are no longer coupled. Map updates first modify the planning graph, then the planner estimates many robot-task paths, and finally the allocator solves a discrete problem. This process is suitable for low event rate scenarios, but the delay will be longer when multiple obstacles, tasks, or robot failures occur simultaneously. If the path estimate is outdated at the time of optimization, it may also lead to a poor decision.

Learning-Based Coordination and Spatial Encoding

The learning-based allocation parameterized strategy has replaced repetitive search. Reinforcement Learning has been applied to cooperative assignment problems with combined throughput and route-cost rewards [12]. Attention and graph neural networks improve variable-size reasoning by exchanging information among robot and task nodes [13]. Modern convolutional backbones offer strong multi-scale spatial features and are required to handle corridors, exclusion zones and inspection targets with different receptive-field scales [14]. Teacher-or target-network averaging is often used to reduce training oscillation and improve the stability of pseudo-targets [15].

Most learned allocators still use a graph derived from a map, rather than the map itself. Abstraction is convenient, but it cannot account for the shape of obstacles or how recently they have changed spatially. Image-based navigation policies maintain geometry but frequently select motion actions instead of fleet-level assignments. At this time, the two are linked by a convolutional map encoder that conditions a constrained robot-task decoder, and EMA consistency is applied to the allocation logits rather than at the low-level control.

Dynamic-Map Representation and Spatial Encoding

Dynamic Inspection Map Representation

The Workspace is kept as a synchronous stack of raster layers with resolution r . Static occupancy, transient occupancy, risk, age variation, task urgency, robot accessibility, standardized battery reserve, and communication quality are all in the same coordinate system. An encoder differentiates between stationary and moving objects to distinguish a permanent wall from a recently parked car. Each task adds a Gaussian urgency

field at its position, and the amplitude is set by the deadline slack and inspection priority. Robot layers contain pose impulses broadened by a small kernel and multiplied by capability embeddings. The resulting tensor still has local geometry and fleet information.

$$X_t = [O_s, O_t, R_t, A_t, U_t, P_t, B_t, C_t] \in \mathbb{R}^{H \times W \times 8} \quad \text{Eq.(1)}$$

Here, O_s and O_t denote static and transient occupancy, R_t is risk, A_t is normalized change age, U_t is task urgency, P_t is robot presence, B_t is battery reserve, and C_t is communication quality. Change age decays smoothly rather than becoming a binary flag:

$$A_t(p) = \exp\left(-\frac{t - t_p}{\tau_a}\right) \mathbb{I}(p \text{ changed}) \quad \text{Eq.(2)}$$

The event gate aggregates geometric, task, and robot-state changes:

$$g_t = \mathbb{I}(\alpha_o \Delta O_t + \alpha_q \Delta Q_t + \alpha_r \Delta S_t > \theta_g) \quad \text{Eq.(3)}$$

The network is invoked only when $g_t = 1$ or a safety timeout expires. Figure 1 presents the resulting pipeline, from heterogeneous map updates and task arrivals to aligned layers, ConvNeXtV2 encoding, feasibility masking, assignment, and route dispatch.

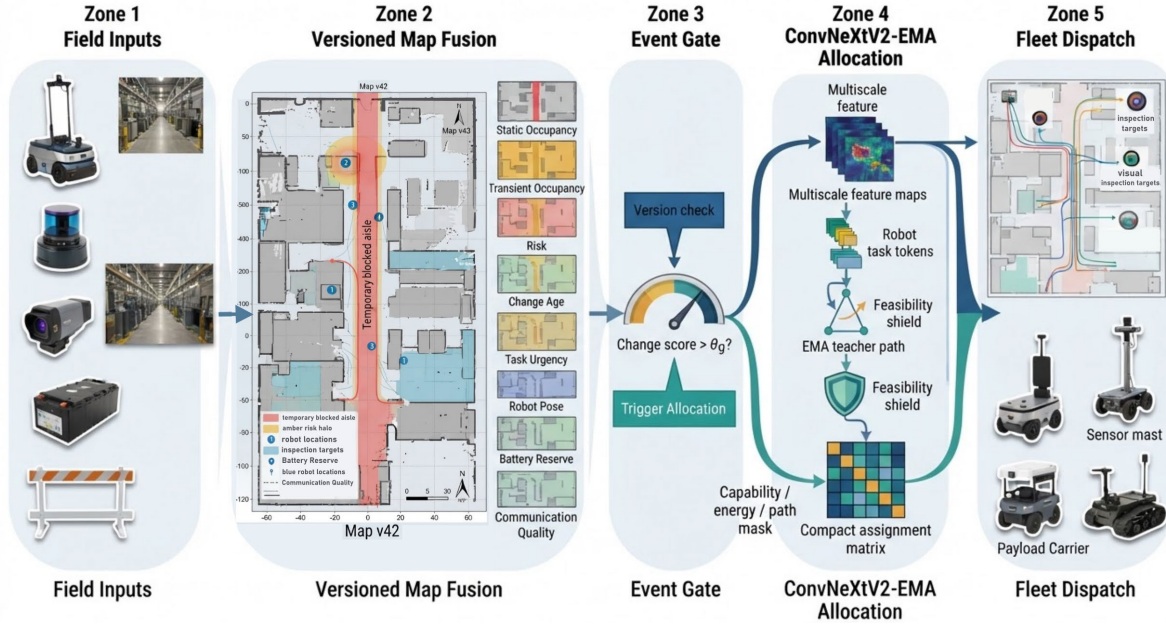


Figure 1. Dynamic inspection-map construction and event-driven multi-robot allocation pipeline.

ConvNeXtV2 Spatial Encoder

Four stages of the encoder have channel widths of 64, 128, 256 and 512. A 4×4 stem downsamples the map, and then a 7×7 convolution is used to learn corridor-scale features. Global response normalisation promotes channel specialisation in occupancy, risk, urgency and fleet-density patterns. The residual block for a stage input z is

$$z' = z + W_2 \text{GELU}(\text{GRN}(W_1 \text{DWConv}_{7 \times 7}(z))) \quad \text{Eq.(4)}$$

Global response normalization is computed as

$$\text{GRN}(x_i) = \gamma_i x_i \frac{\|x_i\|_2}{\sum_j \|x_j\|_2 + \varepsilon} + \beta_i + x_i \quad \text{Eq.(5)}$$

Feature maps at all stages are projected to a common dimension and sampled around robot and task coordinates. Maintain the fine obstacle boundary and grant each token access to a relatively large congestion structure.

A robot token is composed of the sampled map vector, battery, velocity, sensor suite, payload and current workload. A task token combines its sampled vector with service time, deadline slack, priority and capability requirements. Cross-attention outputs a compatibility logit

$$l_{ij} = \frac{(W_q h_i^r)^\top (W_k h_j^q)}{\sqrt{d}} + w_c^\top \phi_{ij} \quad \text{Eq.(6)}$$

where ϕ_{ij} contains path length, estimated energy, deadline slack, and route-overlap cues. Path quantities are computed on the latest traversability graph and are treated as auxiliary evidence rather than the sole environmental representation.

Event-Aware Candidate Construction

Candidate pairs are pruned before attention. A task is excluded from a robot if its capability requirements are not met, its minimum energy exceeds the available reserve, or the latest planner reports no traversable path. For feasible pairs, a local change score measures the fraction of recently modified cells in a corridor around the estimated route:

$$c_{ij} = \frac{\sum_{p \in \text{corridor}(i,j)} A_t(p)}{|\text{corridor}(i,j)|} \quad \text{Eq.(7)}$$

Candidate pruning keeps at most $k = 12$ tasks per robot using a lower-bound cost and always retains urgent tasks. This bounded candidate set makes latency predictable as the global task queue grows.

The model does not immediately revoke all previous assignments after a map event. A hysteresis term protects an active task unless the new decision improves the predicted mission cost by a margin of η , or the current route becomes infeasible. Retained assignment is encoded as a binary pair feature and added to elapsed service time. Change corridors are generated by dilating the provisional path with the robot footprint and a safety margin. When a task is outside the current crop, its token is given a pooled global feature and a direction-to-task embedding.

Constrained ConvNeXtV2-EMA Allocation

Feasibility-Masked Assignment Decoder

The decoder forms a bipartite score matrix and adds a learned idle token for each robot. A Boolean mask M removes incompatible pairs before normalization:

$$p_{ij} = \frac{\exp(l_{ij}/T) M_{ij}}{\sum_k \exp(l_{ik}/T) M_{ik} + \exp(l_{i0}/T)} \quad \text{Eq.(8)}$$

Here, T is the training temperature and index 0 denotes idling. A differentiable Sinkhorn layer encourages near oneto-one assignments during training:

$$P^{(s+1)} = \text{ColNorm} \left(\text{RowNorm} \left(P^{(s)} \right) \right) \quad \text{Eq.(9)}$$

Six alternating normalization steps are used. A sparse linear assignment of the retained candidates yields discrete actions for deployment. Already executing non-preemptive tasks are locked, and preemptive tasks have a switching penalty.

Collision coupling is shown as an overlap matrix of the tentative paths. If the two selected routes are close and unsafe during a certain period, then the lower-priority pair will be delayed or given to another task.

$$\mathcal{L}_{col} = \sum_{(i,j) \neq (u,v)} P_{ij} P_{uv} \omega_{ijuv} \quad \text{Eq.(10)}$$

The coefficient ω_{ijuv} is high only for temporal-spatial conflicts. Figure 2 details the complete architecture, including the four-stage ConvNeXtV2 pyramid, coordinate sampling, robot-task cross-attention, EMA branch, feasibility mask, and collision-aware Sinkhorn decoder.

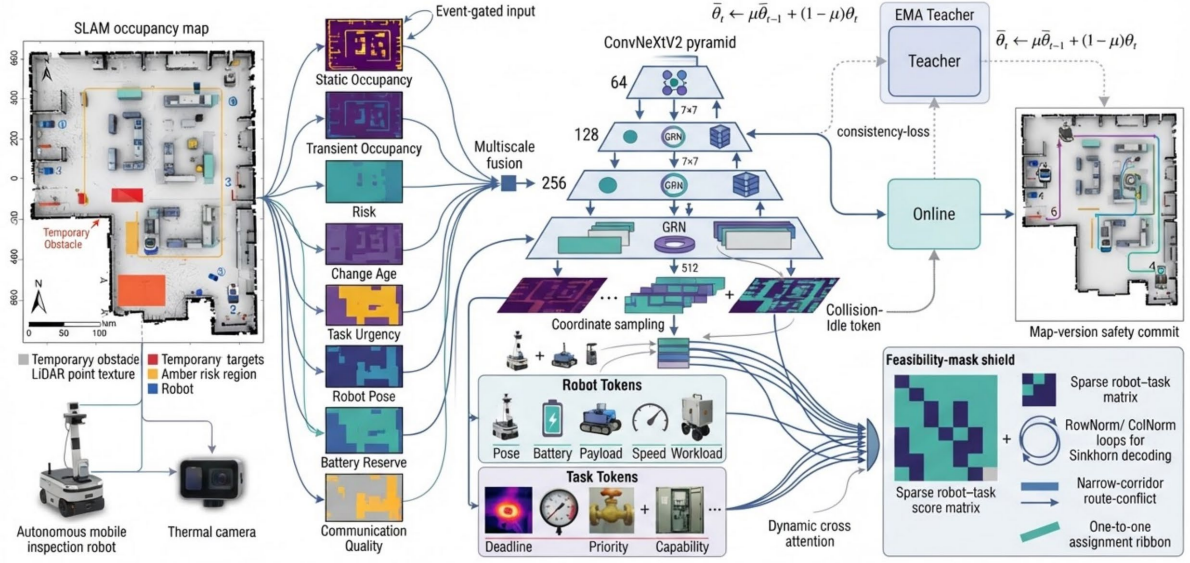


Figure 2. ConvNeXtV2-EMA architecture and constrained assignment decoder.

EMA Teacher and Training Objective

The online network is optimized by gradient descent, whereas the teacher parameters are updated without gradients:

$$\bar{\theta}_t = \mu \bar{\theta}_{t-1} + (1 - \mu) \theta_t \quad \text{Eq.(11)}$$

The coefficient μ increases from 0.99 to 0.9997. The teacher observes a weakly perturbed map, while the student receives stronger cell dropout, localization jitter, and channel noise. Consistency is evaluated only on pairs feasible in both views:

$$\mathcal{L}_{cons} = - \sum_{ij} M_{ij}^* \bar{p}_{ij} \log p_{ij} \quad \text{Eq.(12)}$$

The slowly moving teacher suppresses target flips caused by small map perturbations while remaining responsive to sustained environmental changes.

Supervised targets are generated by a rolling-horizon solver for small and medium training instances. Policy-improvement trajectories augment large-scale cases. Mission cost includes normalized travel distance, energy, lateness and reassignment:

$$J = \lambda_d D + \lambda_e E + \lambda_l L + \lambda_s S \quad \text{Eq.(13)}$$

Workload balance is measured by

$$V = \frac{\text{std}(w_1, \dots, w_n)}{\text{mean}(w_1, \dots, w_n) + \varepsilon} \quad \text{Eq.(14)}$$

The complete objective is

$$\mathcal{L} = \mathcal{L}_{assign} + 0.35 \mathcal{L}_{cons} + 0.20 \mathcal{L}_{col} + 0.15 V + 10^{-4} \|\theta\|_2^2 \quad \text{Eq.(15)}$$

Optimization uses AdamW, cosine decay, stochastic depth, and gradient clipping. Training batches mix robot counts and event rates so that padding patterns cannot reveal instance difficulty.

Online Allocation and Safety Handling

At runtime, the mapping server publishes a versioned delta. The allocator updates affected tensor tiles and invalidates path features whose corridors intersect those tiles. If the event gate fires, inference and sparse matching run asynchronously while robots continue their current safe motion. A decision is committed only if its map version still matches the latest safety-critical occupancy state.

Emergency stops, localization loss, and low-battery states bypass learned scores. A deterministic safety supervisor parks or redirects the affected robot, marks its assigned task as pending, and triggers reallocation. Learning is therefore used to rank feasible alternatives, not to override certified motion and energy constraints.

The average-case computational cost is

$$\mathcal{C} = O\left(HW \sum_s d_s k_s^2 + nkd\right) \quad \text{Eq.(16)}$$

The final sparse matching remains small because candidate pruning bounds k . Cached feature tiles reduce repeated work when a change occupies only a limited map region. If inference exceeds its deadline or returns non-finite values, the service falls back to the auction baseline using the current feasibility mask.

Experimental Evaluation

Experimental Protocol

Experiments used 2,400 simulated episodes distributed in warehouse, utility-tunnel and substation layouts, as well as 180 runs of six differential-drive inspection robots. Fleets contained 4, 8, 12 or 20 robots and 20-160 tasks. Dynamic events include temporary aisle closures, moving exclusion zones, urgent task insertions, robot dropouts and unavailable charging stations. 20 Random Seeds were used for each configuration. Baselines included greedy nearest-feasible assignment, a market auction, rolling mixed-integer programming, graph attention with reinforcement learning, and a CNN allocator without EMA. Hyperparameters were chosen by separate maps. Recent evaluation practice recommends reporting both solution quality and replanning burden for dynamic allocation [16]. The hardware protocol also performed a map-version consistency check as in field robotics [17].

The main indicators were task completion, mission time, distance travelled, deadline violation, energy per completed task, workload coefficient of variation, reassignment count, inference latency and safety interventions. Completion is recorded only when the required sensor payload has reached the destination and uploaded a valid inspection record. Statistical intervals were obtained by bootstrap resampling of episodes; paired tests used the same map-event streams. Robustness tests added map-cell corruption, pose noise, message loss and event rates from 0 to 30 per hour. These disturbances reflect the sensing and networking conditions assumed in the current multi-robot system [18].

The simulator has a resolution of 0.10m for occupancy and runs at 25Hz. Allocation was judged at the time of the event and every two seconds. Robots varied in maximum speed, endurance, turning radius, sensor payload and access rights. One of the four sensing packages and a service duration of 20-180s were required for the task. Deadline slack is 3-20 minutes, and 15% of the tasks are overdue. A generated episode started with a feasible nominal map, and then dynamic events were sampled from a marked temporal process, but the spatial locations were biased towards corridors and work cells to avoid unrealistically harmless changes. A blocked cell remained for 20-240 s. Robot failures were infrequent but severe; the task had to be re-queued, and the failed robot became a dynamic obstruction until it could be recovered.

All baselines received the same capability masks, shortest-path service and map versions. The mixed-integer method used a 500 ms time limit and returned the best incumbent; the auction executed three bid rounds; and learned baselines used the same token dimension and training episodes where applicable. Normalization prevents the improvement from being due to a faster planner or an advantageous state. The proposed encoder had 8.4 million trainable parameters. Training for 120 epochs, using batches of 16 variable-size instances, an initial learning rate of $3e-4$, weight decay of 0.05, and stochastic depth of 0.15. The EMA schedule reached 0.9997 after 70% of the training. Validation stopped after 12 epochs without improving the composite mission score.

Confidence intervals were computed for episode means instead of individual tasks because the tasks within an episode share map changes and are not independent. The Paired Bootstrap was performed 10,000 times. A difference was considered operationally meaningful only if its 95% interval did not include zero and its relative size exceeded 2%. The six blocks of hardware runs were divided over three days, and map layouts and event streams were rotated among methods to reduce time-of-day and battery-aging effects. Balance the battery to

within 3% of the state of charge (SOC) range, and reinitialize the localization map using the same measurement anchor point before each block.

Figure 3(a) shows completion rising from 86.1% with four robots to 94.6% with twelve before saturating at 95.0% with twenty; Figure 3(b) shows mission time remaining 8.7–12.8% below the graph-attention baseline across task counts; and Figure 3(c) shows traveled distance decreasing by 17.3% at 120 tasks while the interquartile range narrows. These scale results indicate that the spatial encoder benefits most when route interaction is nontrivial. The comparison is consistent with findings that learned coordination gains become clearer as pairwise conflicts increase [19].

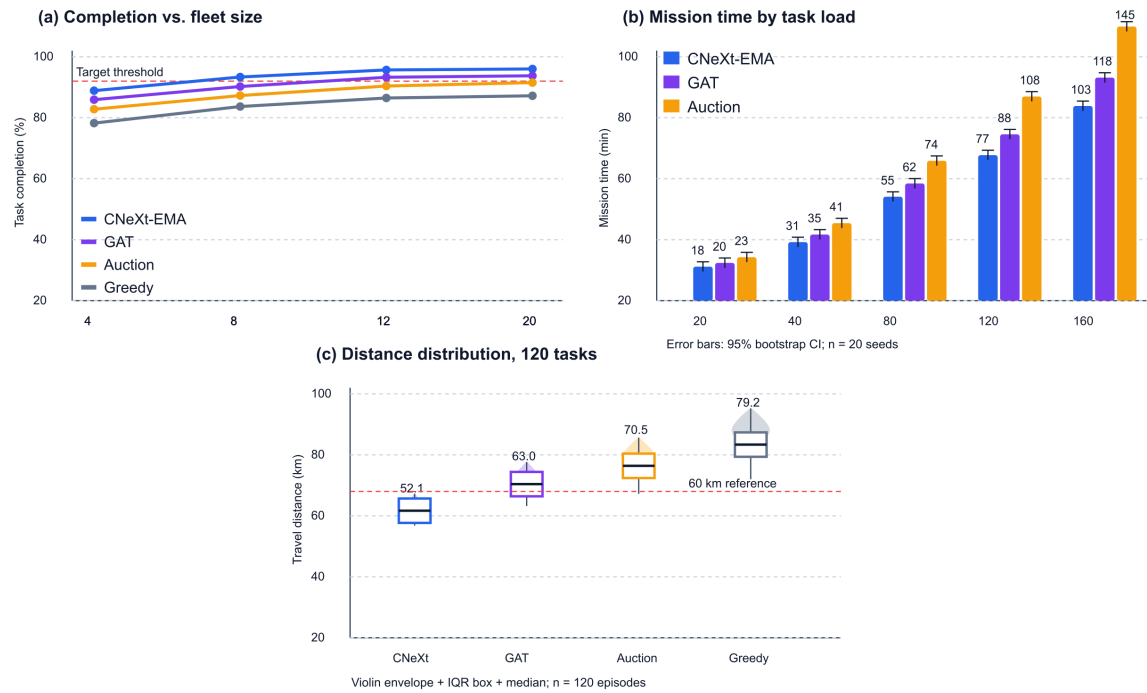


Figure 3. Scalability results: (a) completion rate versus fleet size; (b) mission time versus task count; (c) traveled-distance distributions at 120 tasks.

Performance under Dynamic Map Changes

Figure 4(a) plots completion against map-event rate: ConvNeXtV2-EMA declines from 95.2% at a static map to 91.8% at 30 changes per hour, whereas graph attention falls from 92.4% to 84.7%; Figure 4(b) reports allocation latency distributions with a 73 ms median and 119 ms 95th percentile under the highest event rate. The event gate skips 41% of inconsequential updates and avoids repeated decoding after small free-space corrections. Dynamic navigation research similarly identifies update selectivity as a key control on end-to-end response time [20].

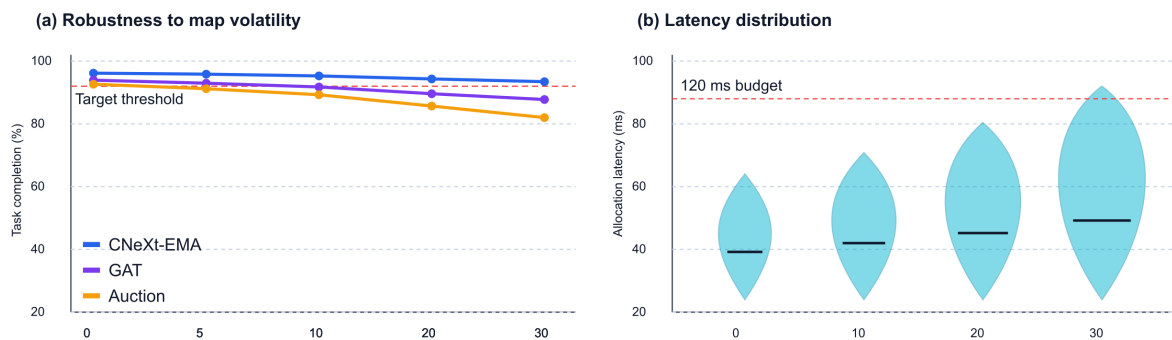


Figure 4. Dynamic-map responsiveness: (a) completion under increasing event rates; (b) allocation-latency violin distributions with median and 95th-percentile markers.

The model reassigned 1.42 tasks per episode at 20 events per hour, which was lower than 2.31 for the CNN without EMA and 2.68 for graph attention. Reduce churn because of the communication and route-reset costs of handoffs; a purely geometric objective does not include these costs. EMA did not delay the required response: after a corridor closure, the median time to a feasible replacement was 0.41s and only 0.03s slower than for the non-EMA CNN. Target-network stabilization has been associated with reduced policy variance in related learning systems [21].

Mission-time improvements were concentrated in medium- and large-sized queues. At 20 tasks, the proposed model was 2.1 minutes faster than graph attention, and at 160 tasks, it was 15.0 minutes faster. The reason is shown in the route-overlap record: the number of predicted narrow-corridor conflicts at 120 tasks dropped from 7.3 to 4.1 per episode. Collision avoidance in the motion-planning layer solved all the outstanding conflicts, but each resolution added an average wait of 11.6s. Therefore, the distribution model extends the mission time by not modifying the robot's path directly. Energy per completed task decreased from 0.91 to 0.84 kWh in the warehouse layout due to fewer robot crossings of the central aisle without productive service.

The Event Gate was analyzed independently of the Encoder. Gating was disabled, and the completion change was less than 0.2 points; However, the median compute utilization increased from 31% to 54%, while the 95th percentile latency for burst updates increased from 119 milliseconds to 164 milliseconds. The threshold for benign location spots is set below 0.12, while the threshold for delayed corridor closure responses is set above 0.28. The selected value of 0.20 resulted in the smallest product of latency and missed-event cost. Watchdog allocation accounted for 7% of all invocations, and recovered cases were those where several individually small changes accumulated into a significant route difference.

Figure 5(a) presents a radar profile in which the proposed model leads on completion, distance, energy, balance, and deadline adherence; Figure 5(b) shows a single heat map of relative improvement, ranging from 6.4% in energy to 24.9% in deadline violations; Figure 5(c) gives the Pareto front between mission time and reassignment count, where EMA solutions occupy the lower-left region; and Figure 5(d) displays deadline-violation densities with a median of 2.8% against 6.9% for graph attention. These results show that the gain is not produced by trading stability for missed deadlines. Multi-objective reporting is important because a faster fleet can still be operationally poor if workload or energy is concentrated [22].

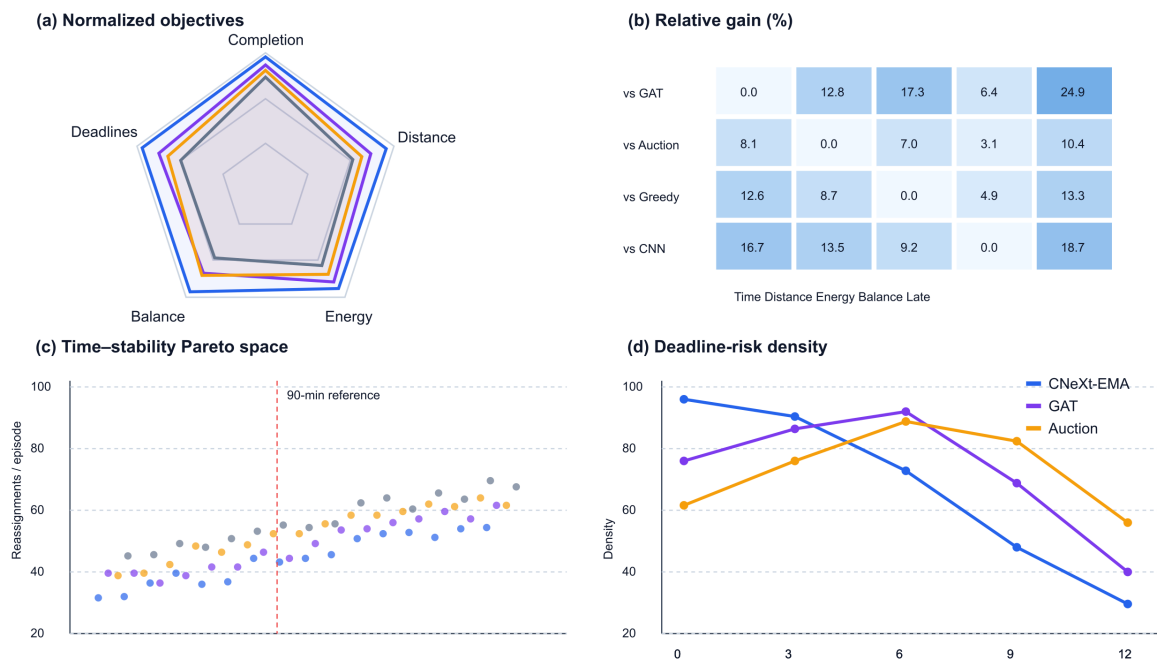


Figure 5. Multi-objective comparison: (a) normalized radar profile; (b) relative-improvement heat map; (c) mission-time/reassignment Pareto front; (d) deadline-violation density.

A cost decomposition supports the same interpretation. Travel accounted for 61.4% of the normalized objective for greedy assignment but 52.7% for ConvNeXtV2-EMA; lateness dropped from 18.6% to 10.2%, and switching

was less than 4%. The auction distributed the workload reasonably well and selected a route through a frequently blocked transfer aisle because its bid used the latest path length without a change in age. The graph model showed that aisles were edges with discontinuous cost functions, and thus, subsequent updates resulted in alternating assignments. Age of spatial change predicted that the model would issue a graded warning before the path became entirely infeasible.

Ablation, Robustness, and Hardware Transfer

Figure 6(a) reports the ablation waterfall: removing EMA lowers completion by 2.7 points, removing multiscale fusion lowers it by 2.4 points, replacing ConvNeXtV2 with a shallow CNN costs 3.9 points, and removing route-overlap loss costs 1.6 points. Figure 6(b) shows calibration curves; expected calibration error is 0.031 for the full model and 0.084 without EMA. Figure 6(c) plots workload Lorenz curves, with the 95th-percentile imbalance reduced from 0.214 to 0.137. Calibration is relevant because confidence margins control hysteresis and reassignment [23].

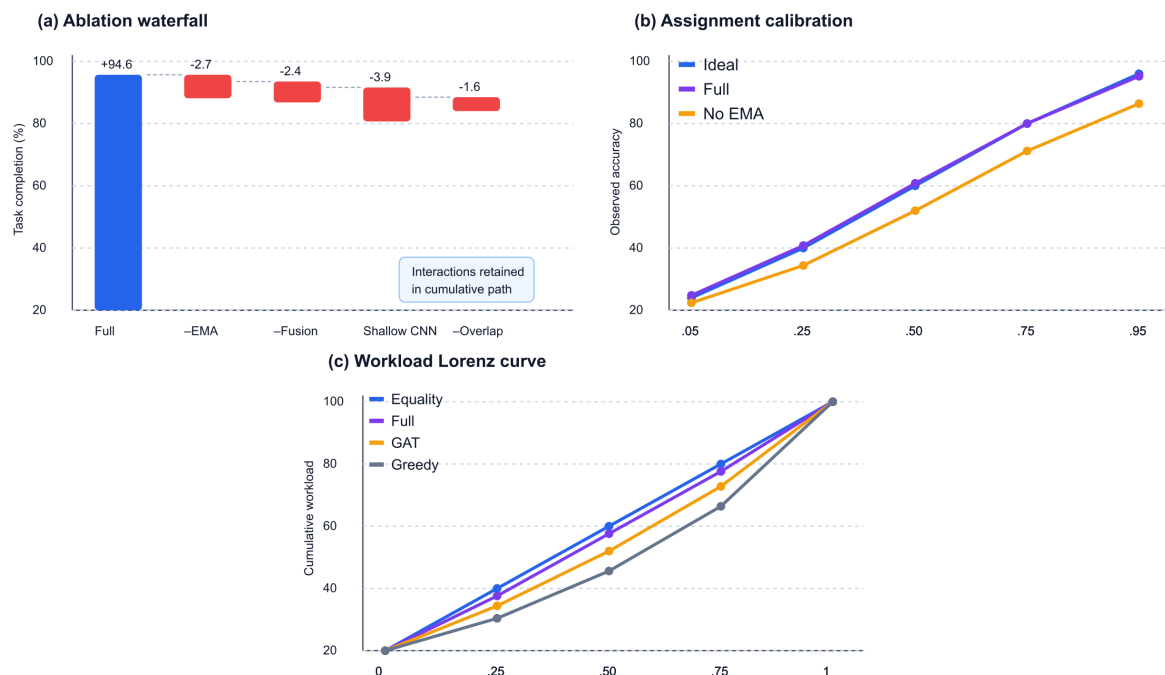


Figure 6. Ablation and allocation quality: (a) completion waterfall; (b) confidence calibration; (c) workload Lorenz curves.

Robustness is relatively low-impact, not severe. With 10% random occupancy-cell corruption, the completion rate is 92.7%; with 0.20m pose noise, it is 92.1%; and with 15% message loss, it is 91.5%. Mask refresh avoids corrupt scores due to capability or battery constraints. Research on distributed robot teams has also found that old communication adversely affects performance before posing a feasibility problem under local safety constraints [24]. The maximum sensitivity occurs when the map corruption is spatially correlated along a narrow corridor, indicating that topology-aware uncertainty should be modelled explicitly.

Component Interactions were not cumulative. Both EMA and multiscale fusion lowered the completion rate by 6.4 percentage points; the sum of their individual effects was 5.1 percentage points. Without multi-scale features, the student logits were more sensitive to cell-level noise and required more temporal averaging. EMA is not used; therefore, the entire feature pyramid will show reduced variation after map augmentation. Pruning candidates from twelve to six tasks reduced the median latency by 8ms but lost 1.5 completion points; increasing it to twenty gained only 0.2 points and added 14ms. Six Sinkhorn iterations were sufficient: the mean row/column residual was less than 0.006, and further iterations did not change discrete matches in more than 0.3% of cases.

Under correlated corridor corruption, the completion rate was 88.9%, and for uniformly random corruption at the same number of cells, it was 92.7%. Uncertainty dilation of two cells recovered 1.7 points but increased the travelled distance by 3.4%, showing a conservative-routing trade-off. The impact of pose noise on coordinate

sampling is greater than that on path features, while bilinear sampling and coordinate jitter training recover about half of the loss of the non-augmented model. Message loss resulted in outdated battery data. An 8% reserve margin avoided energy infeasibility but led to an additional six idle decisions among the 400 stress episodes.

Figure 7(a) compares simulation and hardware completion, measuring 94.6% and 92.8%, respectively; Figure 7(b) gives per-stage latency, with map update, encoding, attention/decoding, and safety commit requiring 11, 38, 19, and 9 ms; and Figure 7(c) shows energy per completed task across three layouts, with reductions of 8.1%, 6.7%, and 9.4% over graph attention. The 1.8-point transfer gap is mainly associated with wheel slip near temporary barriers. Contemporary sim-to-real evaluations stress the need to retain deterministic supervision at deployment [25]. No learned decision caused a safety intervention; 14 interventions were initiated by the independent supervisor because of localization or proximity events. This separation accords with engineering recommendations that learning components remain bounded by explicit runtime assurance [26].

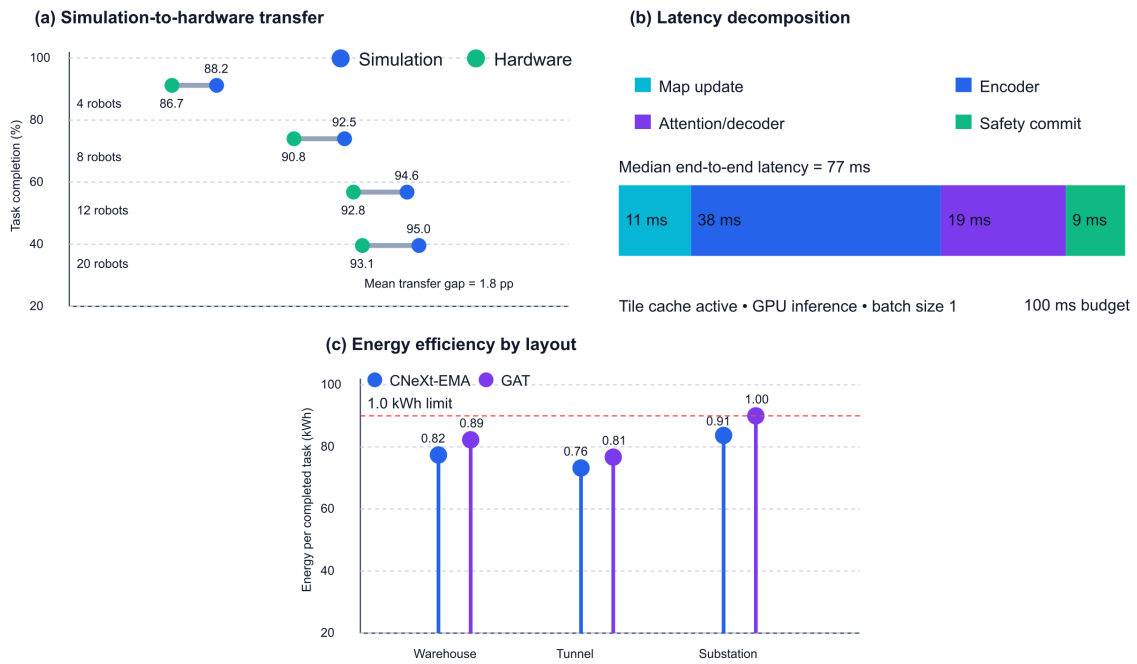


Figure 7. Hardware transfer and efficiency: (a) simulation-to-hardware completion; (b) latency decomposition; (c) energy per completed task across layouts.

A profile of the hardware in an industrial computer with an 8-core processor and a mid-range embedded GPU was taken. The full 77 ms median pipeline was still under the 100 ms operating goal. Tile caching reduced the encoder time to 38ms when fewer than 15% of the map cells changed. CPU-only inference required 214ms and was still suitable for the two-second watchdog rate, but burst response dropped to 30 events per hour. The total memory consumption of the online and EMA networks was 1.2 GB; after disabling teacher-assisted calibration, only the online weights were needed, and the deployment memory dropped to 0.71 GB.

Inspection Validity has also been verified. Images and thermal records obtained after the learned tasks met the same quality gate as the baseline records; thus, improved completion was not due to reduced service time. Operators randomly sampled 60 task handoffs and found that 56 were easily explained from the distance, deadline or corridor change; only four required fleet-level congestion analysis that could not be observed by a single robot. The above cases show that an operator-facing explanation layer is needed to inform them why the allocator chose a relatively far robot to protect a constrained corridor.

Conclusion

A ConvNeXtV2-EMA model for multi-robot task allocation on dynamic inspection maps was proposed in this paper. This approach preserves the spatial variation patterns in the hierarchical map tensor, uses modern

convolutional encoders to extract multi-scale context, and combines robot task attention with feasibility masking and collision perception decoding. An event gate reduces unnecessary inference, and an exponential-moving-average teacher stabilizes assignments in the presence of noise or short-lived map changes. Based on the above experiments, the new allocator can enhance completion rate, mission efficiency, workload distribution and decision-making speed in both simulation and hardware inspection environments.

The current form is not ideal. Rasterization is a fixed-resolution spatial method that may not be able to display small 3D structures. The EMA teacher reduces target variation but adds a second parameter set during training, and the rolling-horizon solver employed for supervision does not provide globally optimal labels for large instances. Hardware evaluation is limited to ground robots and facilities with reliable centralized map fusion. The safety supervisor will resolve route conflicts after allocation and will not jointly optimize continuous trajectories.

Future work will explore sparse three-dimensional map encoders, uncertainty fields that distinguish correlated mapping errors from transient obstacles, and decentralized feature exchange for communication-limited fleets. A hierarchical decoder can jointly allocate inspection bundles and reserve charging resources over a longer period. Set a formal upper bound on the learned ranking, conduct extensive trials of aerial and legged robots, and clarify the transfer characteristics of this method to mixed-mobility teams and safety-critical infrastructure.

Author Contributions

Antoni Marciniak contributes to conceptualization, methodology, software, validation, analysis, investigation, data collection, draft preparation, manuscript editing, visualization, supervision. All authors have read and agreed with the manuscript before its submission and publication.

Funding

This research received no specific financial support from any funding agency.

Institutional Review Board Statement

Not applicable.

References

- [1] Dutta, A., Czarnecki, E., Ufimtsev, V., & Asaithambi, A. (2020). Correlation clustering-based multi-robot task allocation: A tale of two graphs. *ACM SIGAPP Applied Computing Review*, 19(4), 5–16. <https://doi.org/10.1145/3381307.3381308>
- [2] Otte, M. W., Kuhlman, M. J., & Sofge, D. (2020). Auctions for multi-robot task allocation in communication limited environments. *Autonomous Robots*, 44(3–4), 547–584. <https://doi.org/10.1007/s10514-019-09828-5>
- [3] Lujak, M., Giordani, S., Omicini, A., & Ossowski, S. (2020). Decentralizing coordination in open vehicle fleets for scalable and dynamic task allocation. *Complexity*, 2020, 1–21. <https://doi.org/10.1155/2020/1047369>
- [4] Zitouni, F., Harous, S., & Maamri, R. (2020). Towards a distributed solution to multi-robot task allocation problem with energetic and spatiotemporal constraints. *Computer Science*, 21(1), 3–24. <https://doi.org/10.7494/csci.2020.21.1.3427>
- [5] Jin, K., Tang, P., Chen, S., & Peng, J. (2021). Dynamic task allocation in multi-robot system based on a team competition model. *Frontiers in Neurorobotics*, 15, 674949. <https://doi.org/10.3389/fnbot.2021.674949>
- [6] Martin, J. G., Frejo, J. R. D., García, R. A., & Camacho, E. F. (2021). Multi-robot task allocation problem with multiple nonlinear criteria using branch and bound and genetic algorithms. *Intelligent Service Robotics*, 14(5), 707–727. <https://doi.org/10.1007/s11370-021-00393-4>
- [7] Mazdin, P., & Rinner, B. (2021). Distributed and communication-aware coalition formation and task assignment in multi-robot systems. *IEEE Access*, 9, 35088–35100. <https://doi.org/10.1109/ACCESS.2021.3061149>
- [8] Cechinel, A. K., De Pieri, E. R., Perez, A. L. F., & Plentz, P. D. M. (2021). Multi-robot task allocation using island model genetic algorithm. *IFAC-PapersOnLine*, 54(1), 558–563. <https://doi.org/10.1016/j.ifacol.2021.08.063>

- [9] Park, B., Kang, C., & Choi, J. (2022). Cooperative multi-robot task allocation with reinforcement learning. *Applied Sciences*, 12(1), 272. <https://doi.org/10.3390/app12010272>
- [10] Fazal, N., Khan, M. T., Anwar, S., Iqbal, J., & Khan, S. (2022). Task allocation in multi-robot system using resource sharing with dynamic threshold approach. *PLOS ONE*, 17(5), e0267982. <https://doi.org/10.1371/journal.pone.0267982>
- [11] Kyprianou, G., Doitsidis, L., & Chatzichristofis, S. A. (2022). Towards the achievement of path planning with multi-robot systems in dynamic environments. *Journal of Intelligent & Robotic Systems*, 104, 15. <https://doi.org/10.1007/s10846-021-01555-3>
- [12] Aziz, H., Pal, A., Pourmiri, A., Ramezani, F., & Sims, B. (2022). Task allocation using a team of robots. *Current Robotics Reports*, 3, 227–238. <https://doi.org/10.1007/s43154-022-00087-4>
- [13] Harman, H., & Sklar, E. I. (2022). Multi-agent task allocation for harvest management. *Frontiers in Robotics and AI*, 9, 864745. <https://doi.org/10.3389/frobt.2022.864745>
- [14] Hu, J., Niu, H., Carrasco, J., Lennox, B., & Arvin, F. (2020). Voronoi-based multi-robot autonomous exploration in unknown environments via deep reinforcement learning. *IEEE Transactions on Vehicular Technology*, 69(12), 14413–14423. <https://doi.org/10.1109/TVT.2020.3034800>
- [15] Kolvenbach, H., Wisth, D., Buchanan, R., Valsecchi, G., Grandia, R., Fallon, M., & Hutter, M. (2020). Towards autonomous inspection of concrete deterioration in sewers with legged robots. *Journal of Field Robotics*, 37(8), 1314–1327. <https://doi.org/10.1002/rob.21964>
- [16] Cui, X., Lu, C., & Wang, J. (2020). 3D semantic map construction using improved ORB-SLAM2 for mobile robot in edge computing environment. *IEEE Access*, 8, 67179–67191. <https://doi.org/10.1109/ACCESS.2020.2983488>
- [17] Bolu, A., & Korcak, O. (2021). Adaptive task planning for multi-robot smart warehouse. *IEEE Access*, 9, 27346–27358. <https://doi.org/10.1109/ACCESS.2021.3058190>
- [18] Yao, C., Liu, C., Liu, M., & Chen, Q. (2021). Navigation in multi-agent system with side preference path optimizer. *IEEE Access*, 9, 113944–113953. <https://doi.org/10.1109/ACCESS.2021.3104470>
- [19] Chang, L., Shan, L., Jiang, C., & Dai, Y. (2021). Reinforcement based mobile robot path planning with improved dynamic window approach in unknown environment. *Autonomous Robots*, 45(1), 51–76. <https://doi.org/10.1007/s10514-020-09947-4>
- [20] Jin, X., & Wang, Z. (2022). Proximal policy optimization based dynamic path planning algorithm for mobile robots. *Electronics Letters*, 58(1), 13–15. <https://doi.org/10.1049/ell2.12342>
- [21] Han, X., Li, S., Wang, X., & Zhou, W. (2021). Semantic mapping for mobile robots in indoor scenes: A survey. *Information*, 12(2), 92. <https://doi.org/10.3390/info12020092>
- [22] Ghassemi, P., & Chowdhury, S. (2022). Multi-robot task allocation in disaster response: Addressing dynamic tasks with deadlines and robots with range and payload constraints. *Robotics and Autonomous Systems*, 147, 103905. <https://doi.org/10.1016/j.robot.2021.103905>
- [23] Szominski, S., Turek, W., & Byrski, A. (2022). Socially-inspired fully decentralized robot coordination. *Simulation Modelling Practice and Theory*, 119, 102528. <https://doi.org/10.1016/j.simpat.2022.102528>
- [24] Charitidou, M., & Keviczky, T. (2022). An MILP approach for persistent coverage tasks with multiple robots and performance guarantees. *European Journal of Control*, 64, 100610. <https://doi.org/10.1016/j.ejcon.2021.12.005>
- [25] Morilla-Cabello, D., Bartolomei, L., Teixeira, L., Montijano, E., & Chli, M. (2022). Sweep-Your-Map: Efficient coverage planning for aerial teams in large-scale environments. *IEEE Robotics and Automation Letters*, 7(4), 10810–10817. <https://doi.org/10.1109/LRA.2022.3194686>
- [26] Huang, J. Y., Zhou, S. Y., Tu, H., Yao, Y. H., & Liu, Q. S. (2022). Distributed optimization algorithm for multi-robot formation with virtual reference center. *IEEE/CAA Journal of Automatica Sinica*, 9(4), 732–734. <https://doi.org/10.1109/JAS.2022.105473>