Contact-Rich Collaborative Path Clearing via Physics-Informed Corridor Search

***Abstract*—Contact-rich path clearing in cluttered environments requires more than detecting blocked routes: it requires selecting interventions that can reliably enlarge a width-constrained corridor through executable multi-robot physical interaction. Existing Navigation Among Movable Obstacles(NAMO) methods typically reason about obstacle rearrangement at an abstract geometric level, while collaborative pushing methods focus on local contact execution without explicitly addressing how local interactions should be chosen to produce global corridor connectivity. We present a physics-informed corridor search framework, PICHS, for contact-feasible collaborative path clearing. PICHS uses a W-clearance connectivity graph (WCCG) to determine whether a traversable corridor already exists and, when it does not, to expose the reachable bottlenecks that currently limit progress. For each selected bottleneck, the method generates only gap-conditioned pushing candidates for adjacent movable obstacles and uses short-horizon physical simulation to validate whether the induced multi-robot interactions are executable and corridor-improving. This tightly couples scene-level bottleneck selection with contact-feasible action generation, reducing unnecessary branching while preserving consistency with real physical execution. Simulation and hardware results show that PICHS improves path-clearing reliability in dense clutter and provides a practical interface between global clearance reasoning and contact-rich multi-robot control.**

I. Introduction

Large transport vehicles often need to traverse cluttered environments in which direct passage is blocked by movable obstacles such as pallets, boxes, or equipment. In these settings, planning must go beyond collision-free routing. The environment itself must be reconfigured to create a corridor with sufficient clearance [1], [2]. This challenge arises in warehouse automation, disaster response, and other dense workspaces where obstacle interaction is unavoidable. The resulting problem is therefore not only to detect whether passage is blocked, but also to decide how the blockage can be removed through physical interaction.

As shown in Fig. 1, the difficulty becomes sharper when corridor construction is delegated to a team of smaller mobile robots acting on behalf of a larger transport platform [3], [4]. The large vehicle defines the required clearance but does not participate in pushing, so the robot team must identify and execute coordinated contacts that open a traversable corridor. Cooperative pushing is therefore a natural mechanism for path clearing, but feasibility is strongly contact-dependent. Pushing one obstacle can induce coupled motion of nearby objects, and whether a candidate action succeeds depends on local geometry, robot size, and obstacle arrangement [5], [6]. Thus, a geometrically plausible clearing sequence may still fail if the required contacts and motions are not executable.

Existing methods typically capture only part of this problem. NAMO-style approaches reason about obstacle rearrangement and corridor creation, but often abstract away the contact mechanics that determine whether an intervention is actually executable in tightly cluttered scenes [2], [7]. In contrast, collaborative pushing and multi-robot manipulation methods emphasize coordinated contact and force generation, but usually focus on single-object transport or more structured settings [8], [9]. Physics-informed planning adds simulation-based validation, yet this is often only loosely coupled to the upstream decision of which blockage to address first [10], [11]. For contact-rich collaborative path clearing, the missing link is therefore a search procedure that connects scene-level bottleneck selection to executable contact planning.

Fig. 1. Path clearing in simulation and hardware. The robot team selectively opens reachable bottlenecks until a W-clear corridor emerges.

This paper addresses that disconnect through a physics-informed corridor hybrid search, abbreviated PICHS. At a high level, PICHS integrates geometric reasoning about corridor connectivity with contact-feasibility evaluation and physical validation of candidate multi-robot interactions. It therefore couples bottleneck selection, action generation, and executability checking within a single search loop for collaborative corridor construction.

II. Method

PICHS links scene-level clearance reasoning to contact-feasible multi-robot execution. Given a start–goal query for a large vehicle in clutter, PICHS first tests whether a W-clear corridor already exists. If passage is blocked, it identifies reachable bottlenecks, generates local pushing modes only for those bottlenecks, and validates selected interventions through simulation. The overall pipeline is shown in Fig. 2.

A. WCCG Representation and Reachable Bottlenecks

The scene is represented by a W-clearance connectivity graph, abbreviated WCCG, whose role is to expose the search-relevant clearance structure without recovering a detailed geometric path at every step. The graph is constructed directly

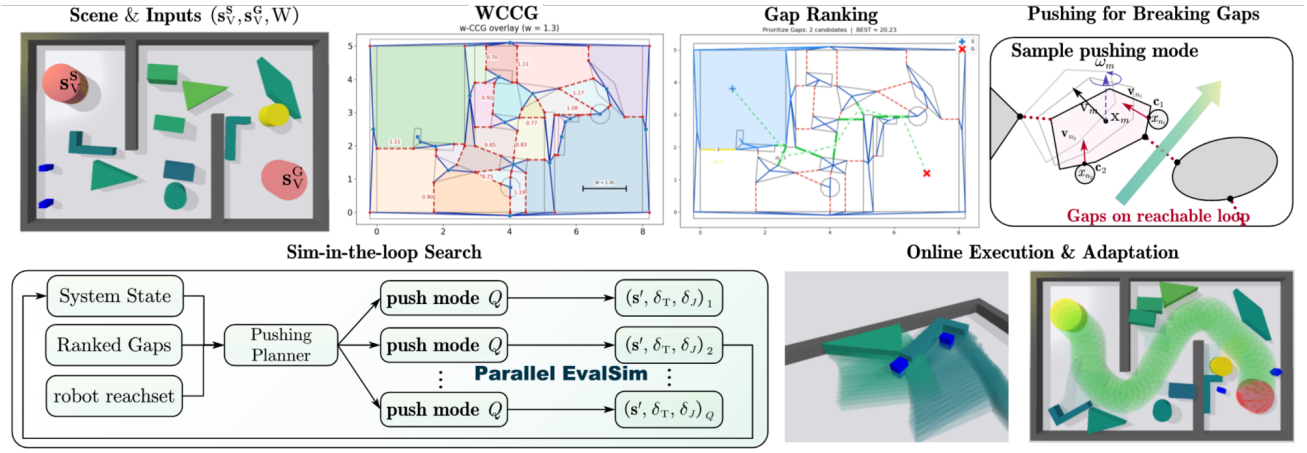

Fig. 2. Overview of PICHS. **Top:** The WCCG identifies reachable bottlenecks and induces gap-conditioned pushing candidates. **Bottom-left:** selective expansion in PICHS. **Bottom-right:** online execution.

from obstacle geometry in a width-aware manner. Centroid nodes encode obstacle components, and bridge relations encode local inter-obstacle passages. The resulting graph is defined as $\mathcal{G}_W \triangleq (\mathcal{V}, \mathcal{E}_c \cup \mathcal{E}_b)$, where $\mathcal{V}$ contains centroid and bridge nodes, $\mathcal{E}_c$ denotes centroid–bridge edges, and $\mathcal{E}_b$ denotes bridge–bridge edges. Each bridge edge is annotated with a local width w_{uv}, defined by the distance between the associated closest points on the two neighboring components. This makes $\mathcal{G}_W$ a compact representation of where the scene is already traversable and where narrow passages arise. In particular, the WCCG does not attempt to enumerate all possible obstacle rearrangements. Its role is to expose the small subset of bottlenecks that currently determines whether a W-clear corridor can be formed.

Given the vehicle start and goal, a width-constrained connectivity query is performed on $\mathcal{G}_W$. If the two lie in the same W-clear connected face, then a feasible corridor already exists and no clearing action is needed, i.e.,

$$\{\mathbf{s}_V^S,\ \mathbf{s}_V^G\} \textit{ belong to the same face of } \mathcal{G}_W, \tag{1}$$

which states that the start and goal of the large vehicle are already connected through the same width-feasible free-space region. If (1) does not hold, the query returns the current reachable frontier loop $\mathcal{L}$ that separates the two sides under the width constraint. The bridge–bridge edges visible on $\mathcal{L}$ define the first-hop set of reachable bottlenecks, $\Gamma_{\mathcal{L}} \triangleq \{g_1, \cdots, g_K\}$, where each g_k is a candidate gap exposed by the current clearance query. In PICHS, these frontier gaps are an internal search interface rather than the primary methodological contribution. Their role is to translate the global clearance query into a compact local decision set. More precisely, each $g \in \Gamma_{\mathcal{L}}$ identifies a reachable place where a small increase in local clearance may change the global connectivity outcome. Subsequent stages therefore do not generate contact actions over the full scene. Instead, they operate only on $\Gamma_{\mathcal{L}}$, which converts scene-level bottlenecks into compact contact-feasible queries for multi-robot pushing.

B. Gap-Conditioned Pushing Mode Generation

For a frontier gap $g \in \Gamma_{\mathcal{L}}$ under consideration, the next question is whether this reachable bottleneck can be enlarged through executable multi-robot interaction. The role of this stage is therefore not to solve a fully general manipulation problem, but to convert a scene-level clearance bottleneck into a small set of contact-feasible intervention queries.

Let Ω_m denote one of the movable obstacles adjacent to g. For each such obstacle, a small set of short-horizon obstacle motion intents $\{\mathbf{v}_m^k\}_{k=1}^{K_m}$ is sampled, where $\mathbf{v}_m^k \triangleq (v_x, v_y, \omega) \in \mathbb{R}^3$. The sampled directions are biased toward local gap-opening motions while allowing some rotational components to account for local geometry. This bias is important. It steers the candidate set toward motions that are likely to widen the selected bottleneck, while still preserving enough directional variation to capture cases in which rotation or oblique contact produces a better opening effect than pure translation. These sampled directions are the candidate pushing intents for the obstacles adjacent to g.

For each sampled direction $\mathbf{v}_m^k$, an associated pushing mode $\boldsymbol{\xi} \triangleq (\mathcal{C}, \mathbf{u})$ is generated, where $\mathcal{C}$ denotes the robot–object contact configuration and $\mathbf{u}$ denotes the associated contact wrench parameters. Let $\mathbf{v}_m^{k,\mathrm{B}} \in \mathbb{R}^3$ denote $\mathbf{v}_m^k$ expressed in the body frame of Ω_m. This body-frame representation makes the feasibility score depend on the local contact geometry of the object rather than on the global scene orientation. A pushing mode is evaluated by the primary and multi-directional feasibility losses

$$\begin{aligned} \mathrm{J}_{\mathrm{F}}^m(\boldsymbol{\xi}, \mathbf{v}_m^{k,\mathrm{B}}) &\triangleq \min_{\mathbf{q}_{m,\boldsymbol{\xi}} \in \mathrm{Q}_{m,\boldsymbol{\xi}}} \left\| \mathbf{q}_{m,\boldsymbol{\xi}} + Q_\mu^m(\mathbf{v}_m^{k,\mathrm{B}}) \right\|_1, \\ \mathrm{J}_{\mathrm{MF}}^m(\boldsymbol{\xi}, \mathbf{v}_m^{k,\mathrm{B}}) &\triangleq \sum_{\mathbf{v}_d \in \mathcal{D}} w_d \cdot \mathrm{J}_{\mathrm{F}}^m(\boldsymbol{\xi}, \mathbf{v}_d), \end{aligned} \tag{2}$$

where $\mathbf{q}_{m,\boldsymbol{\xi}} \in \mathbb{R}^3$ is a feasible body wrench induced by mode $\boldsymbol{\xi}$, $\mathrm{Q}_{m,\boldsymbol{\xi}} \subseteq \mathbb{R}^3$ is the corresponding admissible wrench set under local contact and friction constraints, $Q_\mu^m(\mathbf{v}_m^{k,\mathrm{B}})$ is the quasi-static friction wrench of Ω_m along $\mathbf{v}_m^{k,\mathrm{B}}$, and $\mathcal{D}$ with weights w_d augments the desired direction by neighboring basis velocities. Similar measures are also adopted in [3], [12]. Small $\mathrm{J}_{\mathrm{F}}^m$ is necessary for force feasibility, whereas small $\mathrm{J}_{\mathrm{MF}}^m$ indicates a mode that remains effective under small directional perturbations. Accordingly, the associated mode is obtained by sparse constrained optimization over reachable contacts on $\partial\Omega_m$ and is defined as $\boldsymbol{\xi}_m^k \triangleq (\mathcal{C}_m^k, \mathbf{u}_m^k)$. The optimization minimizes $\mathrm{J}_{\mathrm{MF}}^m(\boldsymbol{\xi}, \mathbf{v}_m^{k,\mathrm{B}})$ under local friction

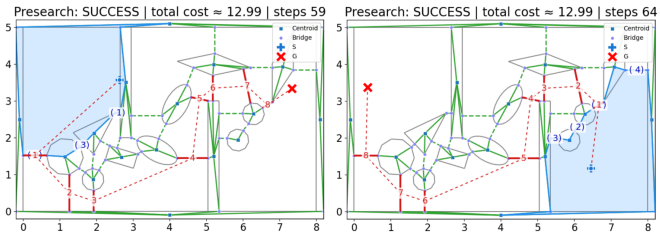

Fig. 3. Ranked reachable bottlenecks under swapped start–goal queries in the same scene. Each panel shows the WCCG with the current reachable face in light blue. Candidate gaps on the frontier loop are ranked by local priority in blue and refined by short presearch in red.

and wrench constraints. Only the best few pairs $(\mathbf{v}_m^k, \boldsymbol{\xi}_m^k)$ are retained, while a persistent *ModeTable* reuses effective entries across similar local geometries. This reuse reduces repeated contact optimization for geometrically similar local arrangements and keeps the candidate set compact. Each retained pushing candidate therefore takes the form $(g, \mathbf{v}, \boldsymbol{\xi})$, and the gap-induced candidate set is denoted by Ξ_g.

A candidate pushing mode is retained only if it passes short-horizon physical validation. The robots must be able to reach the proposed pre-contact poses, avoid collision during approach and pushing, maintain stable contact, and induce obstacle motion that actually enlarges the target gap. This validation is performed by the short-horizon rollout

$$(\mathbf{s}', \delta_T, \delta_J) \triangleq \mathsf{EvalSim}\big(\mathbf{s}, (g, \mathbf{v}, \boldsymbol{\xi})\big), \tag{3}$$

which returns the successor state, time cost, and control effort. The validation step is important because low analytical feasibility loss alone does not guarantee executable multi-robot behavior in dense clutter. This stage therefore does not enumerate contact actions over the full scene. Instead, it converts each reachable bottleneck into a compact set of physically testable multi-robot interventions for the search.

C. Selective Expansion in PICHS

At node ν, PICHS first checks whether the current state satisfies the connectivity criterion in (1). If not, the updated WCCG induces a reachable frontier loop $\mathcal{L}_\nu$ and the associated candidate gaps $\Gamma_{\mathcal{L}_\nu}$. As illustrated in Fig. 3, these gaps are ranked by the predicted cost-to-connect

$$\widehat{\mathsf{Cost}}(g) \triangleq \mathsf{C}(g \mid \mathcal{L}_\nu, \mathbf{s}_\mathcal{R}) + \mathsf{C}_{\mathrm{pre}}(g), \tag{4}$$

where $\mathsf{C}(g \mid \mathcal{L}_\nu, \mathbf{s}_\mathcal{R})$ combines the robots' transition cost to the gap and the estimated effort required to widen it, and $\mathsf{C}_{\mathrm{pre}}(g)$ is a short presearch estimate of downstream gap-crossing cost toward the goal. The first term therefore captures how expensive the candidate is to activate from the current state, whereas the second term estimates whether opening that gap is likely to improve progress toward a full corridor. The ranked list is defined by $\mathsf{Rank}(\nu) \triangleq \mathrm{argsort}_{g \in \Gamma_{\mathcal{L}_\nu}} \widehat{\mathsf{Cost}}(g)$, and the node priority is defined by $f(\nu) \triangleq \chi(\nu) + \min_{g \in \mathsf{Rank}(\nu)} \widehat{\mathsf{Cost}}(g)$, where $\chi(\nu)$ is the realized path cost accumulated from the root to node ν. Equation (II-C) therefore balances what has already been spent against the best currently predicted remaining cost. This balance encourages PICHS to prefer nodes that are both physically promising and globally efficient. Alg. 1 summarizes the resulting selective expansion loop.

Algorithm 1: Selective Expansion in PICHS

Input: $\mathbf{s}_0$, α, $\mathsf{EvalSim}(\cdot)$
Output: $\pi^\star$

$\nu_0 \leftarrow (\mathbf{s}_0, \varnothing)$, $\chi(\nu_0) = 0$;
Initialize $\mathcal{Q}$ by (II-C);
while $\mathcal{Q} \neq \varnothing$ **do**
  $\nu = (\mathbf{s}, \pi) \leftarrow \mathcal{Q}.\mathtt{pop\_min}()$;
  if $\mathbf{s}$ *satisfies* (1) **then**
    $\pi^\star \leftarrow \pi$, **break**;
  Update the WCCG and construct $\mathsf{Rank}(\nu)$;
  foreach $g \in \mathsf{Rank}(\nu)$ **do**
    Generate Ξ_g;
    foreach $(\mathbf{v}, \boldsymbol{\xi}) \in \Xi_g$ **do**
      $\tau \triangleq (g, \mathbf{v}, \boldsymbol{\xi})$;
      $(\mathbf{s}', \delta_T, \delta_J) \leftarrow \mathsf{EvalSim}(\mathbf{s}, \tau)$;
      if *success* **then**
        $\nu' \leftarrow (\mathbf{s}', \pi \cup \{\tau\})$;
        $\chi(\nu') \leftarrow \chi(\nu) + \delta_T + \alpha\, \delta_J$;
        $\mathcal{Q}.\mathtt{push}(\nu')$;
return $\pi^\star$;

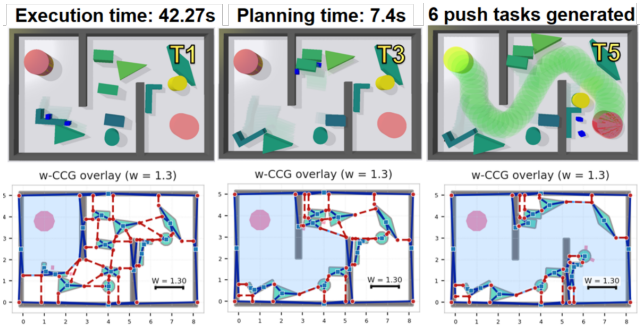

Fig. 4. Representative simulation run with planning–execution alignment. **Top:** PyBullet snapshots of corridor opening by two robots. **Bottom:** corresponding WCCG overlays with the start face shown in blue.

For each ranked gap $g \in \mathsf{Rank}(\nu)$, PICHS calls the gap-conditioned mode generator in Sec. II-B. There, constrained mode generation guided by (2) produces a compact set Ξ_g, and each candidate $\tau \triangleq (g, \mathbf{v}, \boldsymbol{\xi})$ is evaluated by the short-horizon rollout in (3). PICHS is selective in three senses. The WCCG restricts to reachable bottlenecks. Gap ranking prioritizes which bottlenecks to test first. Candidate generation together with short-horizon simulation validates only those interventions worth expanding. This selectivity is the main reason PICHS remains tractable in dense clutter. It reduces the branching factor before simulation, and it reserves physics evaluation for actions that are relevant to corridor construction. Repeating this loop yields a clearing schedule that specifies which gaps to open and how to open them.

III. Numerical Evaluation

The proposed formulation is evaluated in cluttered multi-obstacle clearing tasks with three objectives. The first is comparison against representative baselines. The second is assessment of the contributions of search-level prioritization and structured contact reuse through ablations. The third is verification on hardware.

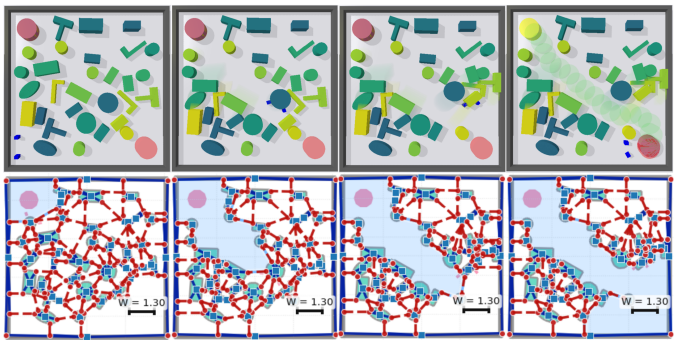

Fig. 5. Scalability in dense clutter with 30 movable objects. **Top:** snapshots during path clearing over 24.5 s. **Bottom:** WCCG overlays with the active face shown in blue and the ranked reachable bottlenecks.

A. Simulation Results

All numerical experiments are implemented in Python3 and PyBullet [13]. As shown in Fig. 4, the nominal setting uses an $8 \times 5\,\mathrm{m}$ bounded workspace in which two robots must clear a passage for an external vehicle through scenes containing 10–15 randomly placed movable obstacles. A trial is counted as successful when a W-clear path exists from start to goal and the target reaches its goal disk. All baselines are evaluated under the same W-clearance criterion and contact model.

For comparisons, three representative baselines are adopted: **DFS-WCCG**, a simulation-in-the-loop depth-first search with fixed pushes; **SL-Push**, which clears obstacles along a route and validates the induced pushes with physics; and **Rec-NAMO**, a traditional NAMO-style baseline that improves local traversability incrementally. As summarized in Table I, PICHS achieves the highest success rate at 92.5% across the evaluated scenes, while also giving the lowest combined planning and execution time among the successful planners. **DFS-WCCG** suffers from rapid branching growth, **SL-Push** requires longer action sequences and more validation, and **Rec-NAMO** often makes only partial progress without establishing a complete W-clear corridor.

Furthermore, the ablation study clarifies which parts of the formulation drive this performance. Removing `presearch` reduces success from 92.5% to 85.0% while increasing planning time and simulation calls. This indicates that the search-level bottleneck ranking reduces unnecessary validation and focuses computation on useful interventions. Removing `ModeTable` further reduces success to 80.0%, again with higher planning cost and more simulation calls. This suggests that structured reuse of contact-conditioned action templates improves search efficiency in the same task setting. Together, these ablations support that PICHS benefits from both bottleneck-aware prioritization before simulation and structured reuse when converting selected gaps into physically-feasible candidates of collaborative pushing modes.

Lastly, to analyze scalability, Fig. 5 shows a cluttered $8 \times 5\,\mathrm{m}$ workspace with 30 movable obstacles. Despite the increased density, PICHS maintains a compact branching factor. It completes the task with 10 node expansions out of 115 visited nodes, 312 short simulation calls, and 7 executed push tasks, yielding a connected W-clear corridor. The planning and execution times are 21.2 s and 24.5 s.

TABLE I
PERFORMANCE COMPARISON AND ABLATIONS.

Method / Variant	Succ.(%)	PT (s)	ET (s)	#Sims	#Pushes
DFS-WCCG	25.0	>100.0	51.2	>500	8.0
SL-Push	75.0	18.2	64.6	20.0	10.0
Rec-NAMO	37.5	13.3	42.4	0.0	7.0
PICHS (ours)	**92.5**	**10.3**	**28.6**	**121.5**	**6.0**
w/o presearch	85.0	16.9	35.2	178	7.0
w/o ModeTable	80.0	15.1	33.4	164	6.0

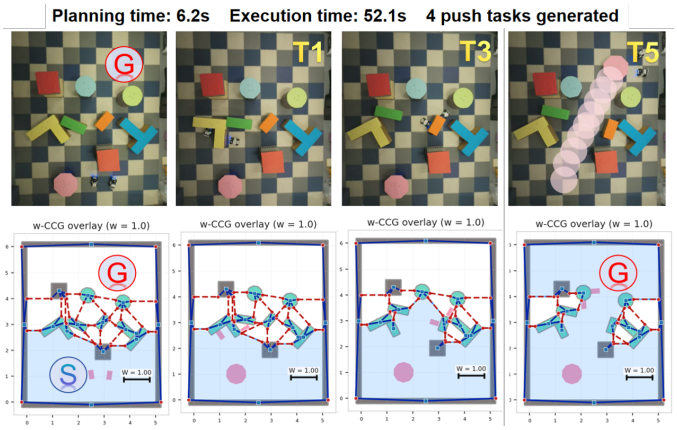

Fig. 6. Representative hardware execution with planning–execution alignment. **Top:** snapshots of two robots sequentially opening frontier gaps in a cluttered workspace. **Bottom:** corresponding WCCG overlays with the current reachable face shown in blue.

B. Hardware Experiments

To verify that PICHS remains valid beyond simulation, the method is deployed on a real two-robot platform in cluttered corridor-construction tasks. Each trial is conducted in a $5 \times 6\,\mathrm{m}$ workspace with 2 UGVs and 6 movable obstacles, whose observed poses are streamed online for execution and visualization. As shown in Fig. 6, PICHS computes a plan in 6.2 s and completes execution in 52.1 s while generating only 4 push tasks. Thus, execution is about $8.4\times$ longer than planning, and the average execution cost is about 13.0 s per push task. The figure also shows that the corridor is cleared without moving all 6 obstacles. Instead, PICHS selectively opens only the bottlenecks required to create a W-clear passage from start to goal. From T1 to T5, the reachable free region in the WCCG overlay expands from the start side toward the goal, indicating steady corridor growth under real execution. Throughout the run, the WCCG is rebuilt from observations, so the planner continues reasoning over the currently reachable bottlenecks rather than following a fixed open-loop removal sequence.

IV. Conclusion and Outlook

PICHS combines width-aware reasoning with executable multi-robot pushing for cluttered path clearing. Frontier gaps focus validation on a small set of promising actions, supporting reliable corridor construction in simulation and hardware. Future work will focus on uncertainty in state estimation of both the objects and the robots during execution.

REFERENCES

[1] L. Liu, X. Wang, X. Yang, H. Liu, J. Li, and P. Wang, "Path planning techniques for mobile robots: Review and prospect," *Expert Systems with Applications*, vol. 227, p. 120254, 2023.

[2] M. Stilman and J. J. Kuffner, "Navigation among movable obstacles: Real-time reasoning in complex environments," *International Journal of Humanoid Robotics*, vol. 2, no. 04, pp. 479–503, 2005.

[3] Z. Tang, Y. Feng, and M. Guo, "Collaborative planar pushing of polytopic objects with multiple robots in complex scenes," in *Robotics: Science and Systems (RSS)*, 2024.

[4] Z. Ren, B. Suvonov, G. Chen, B. He, Y. Liao, C. Fermuller, and J. Zhang, "Search-based path planning in interactive environments among movable obstacles," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 533–539.

[5] R. Ni and A. H. Qureshi, "Progressive learning for physics-informed neural motion planning," *arXiv preprint arXiv:2306.00616*, 2023.

[6] Y. Liu, R. Ni, and A. H. Qureshi, "Physics-informed neural mapping and motion planning in unknown environments," *IEEE Transactions on Robotics*, 2025.

[7] L. Yao, V. Modugno, A. M. Delfaki, Y. Liu, D. Stoyanov, and D. Kanoulas, "Local path planning among pushable objects based on reinforcement learning," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 3062–3068.

[8] R. Ni and A. H. Qureshi, "Physics-informed neural motion planning on constraint manifolds," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 12 179–12 185.

[9] Y. Feng, C. Hong, Y. Niu, S. Liu, Y. Yang, and D. Zhao, "Learning multi-agent loco-manipulation for long-horizon quadrupedal pushing," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 14 441–14 448.

[10] Y.-C. Lin, B. Ponton, L. Righetti, and D. Berenson, "Efficient humanoid contact planning using learned centroidal dynamics prediction," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 5280–5286.

[11] Q. Rouxel, S. Ivaldi, and J.-B. Mouret, "Multi-contact whole-body force control for position-controlled robots," *IEEE Robotics and Automation Letters*, vol. 9, no. 6, pp. 5639–5646, 2024.

[12] Z. Tang, Y. Zhang, and M. Guo, "Pushingbots: Collaborative pushing via neural accelerated combinatorial hybrid optimization," *IEEE Transactions on Robotics*, 2025.

[13] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," http://pybullet.org, 2016–2019.