# A TRANSPORT-ROBOT NAVIGATION PLATFORM FOR PRE-COLLISION PATH DIAGNOSIS AND REPAIR

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Transport robots have been widely deployed in automated transportation scenarios such as warehouse logistics. When executing preplanned paths, transport robots may still collide with environmental obstacles, causing payload loss and subsequent task interruption. Existing methods mainly focus on post-collision recovery or replanning, without dedicated methods or data support for pre-collision path diagnosis and repair. To address this gap, we introduce **`RobotPath`**, a transport-robot navigation platform for pre-collision path diagnosis and repair in warehouse environments. **`RobotPath`** integrates three components: (1) **scene simulation and automatic path generation**, supporting path generation and execution across different robot configurations, shelf layouts, and execution regimes; (2) **repair generation and verification tools**, which automatically generate repairs and verify their geometric feasibility; and (3) **`LLM-RobotPath`**, a platform-native path diagnosis and repair model using numerically aware collision diagnosis and verifier-guided repair. We further release **`RobotPath-Bench`**, the first benchmark for this task, containing 20,399 path-execution records across 5 robot configurations, 7 shelf layouts, and 2 execution regimes. Extensive experiments show that **`RobotPath`** and **`RobotPath-Bench`** provide reliable support for modeling and evaluating pre-collision path diagnosis and repair. Our code and dataset are available at URL.

## 1 INTRODUCTION

In automated transportation scenarios (Usmani et al., 2023; Yu et al., 2024; Chang et al., 2024), transport robots typically follow planned paths composed of discrete waypoints to complete delivery tasks (Chen et al., 2025; Zhao et al., 2024; Yasuda et al., 2021). However, path execution is constrained by robot dimensions, motion characteristics, and local environmental geometry. Even when all waypoints lie in free space, the robot's continuous swept region during translation and turning between consecutive waypoints may still intersect environmental obstacles such as shelves, thereby introducing a risk of collision. For payload-carrying transport robots, a collision may cause payload loss and subsequently interrupt task execution, motivating collision diagnosis before path execution and local repair of path segments with potential collision risks.

Existing studies mainly focus on post-collision recovery (Lu et al., 2022; Abdullhak & Vardy, 2021; Boschetti et al., 2023; Bredenbeck et al., 2025), local replanning (Lu et al., 2021; Tonola et al., 2023; Li et al., 2021b; Ramesh et al., 2024; Lee & Song, 2023), or deviation correction during trajectory tracking (Dambly et al., 2025; Pan et al., 2025; Xia et al., 2026; Yu et al., 2026), and do not directly address pre-collision diagnosis and repair of planned paths. Correspondingly, existing navigation platforms and benchmarks primarily focus on path planning, navigation success, or trajectory tracking evaluation (Savva et al., 2019; Xiao et al., 2022; Chamzas et al., 2022; Aghzal et al., 2025), thereby providing limited data and evaluation support for pre-collision path diagnosis and repair (see Figure 1). Therefore, systematic simulation platforms, benchmarks, and models specifically designed for pre-collision path diagnosis and repair in transport robots remain to be developed.

In this work, we introduce **`RobotPath`**, a transport-**Robot** navigation platform for pre-collision **Path** diagnosis and repair. Taking warehouse transportation as a well-established scenario, **`RobotPath`** leverages physics-based simulation to provide a repeatable environment for data collection (Makoviychuk et al., 2021), thereby reducing the cost and risks associated with large-scale



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

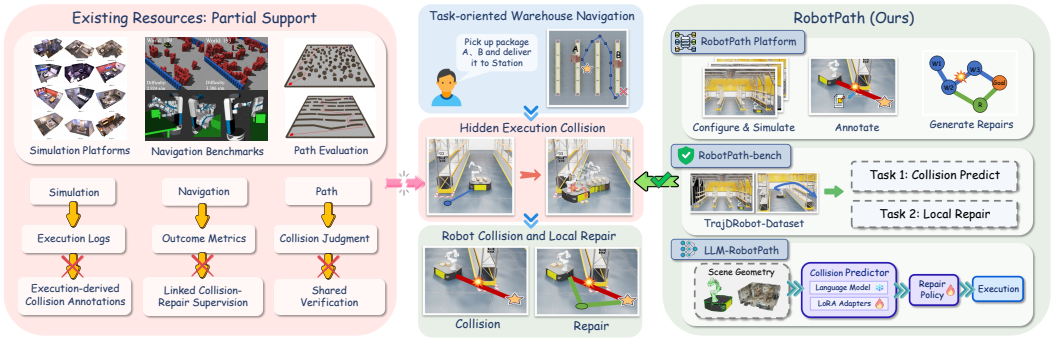



Figure 1: Overview of **RobotPath** vs. Existing Platforms, Benchmarks, and Models.

collection of collision and repair data on physical robots. **RobotPath** consists of three components: (1) *Scene simulation and automated path generation.* Built on Isaac Sim, **RobotPath** supports the automatic generation or import of planned paths across different robot assets and shelf layouts. The robots then execute these paths in simulation, while the platform records execution trajectories, contact information, and structured collision annotations. (2) *Repair generation and verification tools.* We develop an Expert Repair Generator to produce local repair candidates and a Geometric Verifier to automatically assess their feasibility. (3) *LLM-based pre-collision path diagnosis and repair model* (**LLM-RobotPath**). We introduce **LLM-RobotPath**, a platform-native model that leverages a multimodal large language model (LLM) to jointly reason over path information, robot geometry, and environmental constraints. The model adopts a two-stage framework for numerically aware collision diagnosis and verifier-guided local repair, providing an end-to-end evaluation of the platform for path diagnosis and repair.

Based on the **RobotPath** platform, we construct **RobotPath-Bench**, comprising 20,399 path-execution records across 5 robot configurations, 7 shelf layouts, and 2 robot execution regimes. The benchmark provides execution-level collision annotations and geometrically verified local-repair supervision, supporting hierarchical evaluation of collision classification, attribute identification, and spatial localization for diagnosis, as well as repair feasibility, success, and quality. Among the 20,399 path-execution records, 13,924 contain collisions, with robot-center annotations, collision-object fields, and colliding path-segment indices passing validity checks. For 1,213 randomly sampled geometrically verified repairs, 99.09% safely reach the local target under independent execution in Isaac Sim, indicating strong consistency between geometric verification and actual execution. Additional experiments under configuration shifts and varying repair constraints further show that **RobotPath-Bench** captures performance variations across robot configurations, shelf layouts, execution regimes, and repair constraints, enabling systematic evaluation of path diagnosis and verifiable repair. Our contributions are summarized as follows:

- To our knowledge, **RobotPath** is the first integrated simulation platform for pre-collision path diagnosis and local repair in transport-robot navigation. It supports configurable path execution, fine-grained collision annotations, and automated local-repair verification.
- We establish **RobotPath-Bench**, a benchmark of 20,399 path-execution records spanning 5 robot configurations, 7 shelf layouts, and 2 execution regimes. It provides fine-grained collision and verified repair supervision, and supports hierarchical evaluation of both diagnosis and repair.
- To validate the platform and benchmark, we develop **LLM-RobotPath**, the first LLM-based model for pre-collision path diagnosis and repair, using a two-stage framework of numerically aware collision diagnosis and verifier-guided local repair. Extensive experiments validate the reliability of the platform's collision annotations and repair verification, while demonstrating the benchmark's effectiveness across diverse configurations.

## 2 ROBOTPATH PLATFORM

In this section, we introduce **RobotPath**, a platform for pre-collision path diagnosis and local repair in transport-robot navigation, as illustrated in Figure 2. We first present scene simulation and automatic path generation (Sec. 2.1), followed by expert repair generation and geometric verification

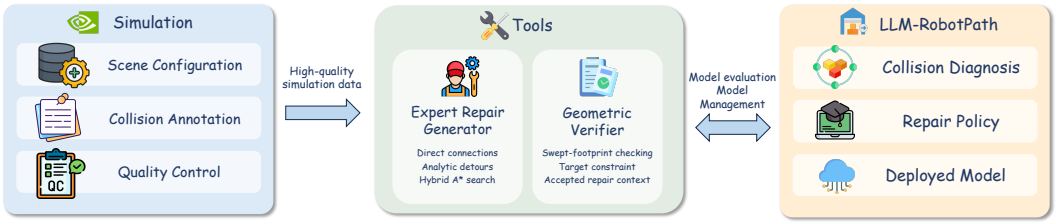



Figure 2: The **RobotPath** integrates Scene Simulation and Automated Path Generation, Repair-Generation and Verification Tools, and **LLM-RobotPath** model evaluation and deployment.

for constructing verified repair data (Sec. 2.2). Finally, we introduce **LLM-RobotPath**, a platform-native model for numerically aware collision diagnosis and verifier-guided local repair (Sec. 2.3).

### 2.1 Scene Simulation and Automated Path Generation

The **RobotPath** platform is built on Isaac Sim 5.0 (Gao et al., 2026) and provides a configurable simulation environment. Users can configure robot assets and footprint geometry, as well as shelf positions, dimensions, and orientations. The platform supports automatic generation or import of planned waypoint paths and executes them in simulation to collect trajectory and collision data. The data collection pipeline consists of three stages: path generation and execution, collision annotation, and quality control. Additional platform details are provided in Appendix A.

① **Path Generation and Execution.** **RobotPath** generates both colliding and non-colliding paths. Collision cases are introduced through lateral localization offsets, shelf-corner cutting, docking over-travel, and wall-adjacent drift. A physics-step controller executes each planned path while recording robot trajectories, contact events, and task status. Data are collected under two execution regimes. The Simple regime adopts turn-then-translate control, whereas the Hard regime allows simultaneous translation and rotation, producing coupled swept-footprint geometry between waypoints.

② **Collision Annotation.** Based on execution logs, **RobotPath** automatically constructs collision annotations. For each colliding path, the platform records the first colliding segment, collision object, and collision position, as well as the robot-center position at collision when available. These annotations characterize both where the collision occurs along the planned path and its local spatial context, providing fine-grained supervision for pre-collision path diagnosis.

③ **Quality Control.** **RobotPath** applies a unified filtering procedure to collected path-execution records, including checks on path length, zero-length segments, collision-annotation validity, and collision-segment indices. It also excludes ineligible or incomplete executions, such as non-colliding runs terminated by timeout or stuck states. Scene configurations, motion parameters, and contact-confirmation rules are retained as metadata to preserve the execution conditions of each record.

### 2.2 Repair-Generation and Verification Tools

For planned paths identified as colliding during simulated execution, **RobotPath** provides two tools: the **Expert Repair Generator** (ERG) for generating local repair candidates and the **Geometric Verifier** (GV) for evaluating their feasibility. Specifically, given a planned path $\mathcal{W} = \{W_1, \cdots, W_M\}$, where $M$ denotes the number of waypoints and $W_i$ the $i$-th waypoint, suppose a collision occurs on the segment $W_k \rightarrow W_{k+1}$. ERG first identifies a collision-free repair start pose preceding the collision and defines the segment endpoint $W_{k+1}$ as the original local target $g_0 = W_{k+1}$. The local repair then starts from this pose, bypasses the collision region, and reconnects to $g_0$. When target adjustment is allowed, a new local target $g$ may be selected within a constrained neighborhood of $g_0$ while keeping the target waypoint index fixed. Once the repair start pose and target are determined, ERG combines direct connections, analytic detour candidates, and Hybrid A$^*$ search to generate candidate repair paths that avoid obstacles and reach the local target. For each candidate, GV evaluates the repair against validity and target constraints, and performs geometric collision checking using the robot footprint and environment geometry. Candidates whose swept footprint intersects obstacles or environment boundaries are rejected, and only geometrically feasible repairs are retained.

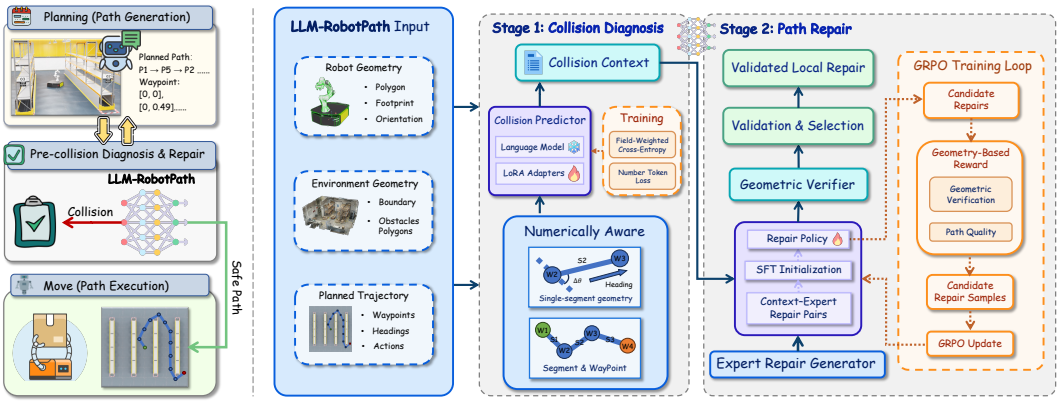



Figure 3: The overall framework of **LLM-RobotPath** for pre-collision path diagnosis and repair.

### 2.3 LLM-RobotPath Model

In this section, we develop **LLM-RobotPath** on the platform for pre-collision path diagnosis and local repair. As shown in Figure 3, **LLM-RobotPath** follows a two-stage framework consisting of a numerically aware collision diagnosis module and a verifier-guided local repair module.

#### 2.3.1 Numerically Aware Collision Diagnosis Module

The collision diagnosis module predicts the occurrence of collisions during planned-path execution and localizes the first collision. The input comprises the robot geometry, environment geometry, and a planned path $\mathcal{W} = (W_1, \cdots, W_M)$. The robot geometry is represented by its footprint polygon in the local frame, whereas the environment geometry is represented by obstacle polygons and warehouse boundaries in the world frame. To explicitly encode the numerical geometry of the planned path, adjacent waypoints are organized into indexed path segments, with endpoint coordinates, segment lengths, planned headings, and turning angles included as geometric descriptors. Let the $k$-th waypoint be $W_k = (x_k, y_k)$ and define path segment as $S_k = (W_k \rightarrow W_{k+1})$. Its length $L_k$, planned heading $\theta_k$, and turning angle $\Delta\theta_k$ relative to the preceding segment are defined as:

$$L_k = \|W_{k+1} - W_k\|_2, \quad \theta_k = \operatorname{atan2}(y_{k+1} - y_k,\ x_{k+1} - x_k), \quad \Delta\theta_k = \operatorname{wrap}(\theta_k - \theta_{k-1}) \tag{1}$$

where $\operatorname{wrap}(\cdot)$ normalizes the angular difference to $(-\pi, \pi]$. The robot, environment, and path representations are combined with task instructions and converted into the LLM input sequence using the prompt template (see Appendix D) and tokenizer. The LLM autoregressively generates a structured collision diagnosis. It first outputs a binary collision indicator $b \in \{0, 1\}$. When $b = 0$, no additional collision attributes are generated. When $b = 1$, the model predicts the index $k$ of the first colliding path segment, the collision object identifier $o$, and four continuous spatial variables $(\tau, \ell, \delta_\parallel, \delta_\perp)$. Here, $\tau$ denotes the relative progress of the robot center along the colliding path segment, and $\ell$ is its lateral offset from the segment. The variables $\delta_\parallel$ and $\delta_\perp$ represent the collision position relative to the robot center along and perpendicular to the segment direction, respectively.

For a colliding path, we define a local coordinate frame relative to the first colliding segment $W_k \rightarrow W_{k+1}$. The unit direction vector $u_k$ and its perpendicular unit vector $n_k$ are defined as:

$$u_k = \frac{W_{k+1} - W_k}{\|W_{k+1} - W_k\|_2}, \quad n_k = (-u_{k,y}, u_{k,x}). \tag{2}$$

Let $c \in \mathbb{R}^2$ be the robot center at collision and $p \in \mathbb{R}^2$ the contact position. They are represented as:

$$c = W_k + \tau(W_{k+1} - W_k) + \ell n_k, \quad p = c + \delta_\parallel u_k + \delta_\perp n_k. \tag{3}$$

Here, $\tau$ is a signed dimensionless quantity, while $\ell$, $\delta_\parallel$, and $\delta_\perp$ are measured in meters. Rather than predicting absolute world-frame coordinates, this representation describes the robot center and collision position with respect to the colliding path segment.

**Collision Diagnosis Training.** To improve the numerical accuracy of the four continuous spatial variables $(\tau, \ell, \delta_\parallel, \delta_\perp)$, we retain two decimal places and use a digit-level representation. For example, $-0.25$ is serialized as $[-, 0, ., 2, 5]$. The training loss is defined as:

$$\mathcal{L}_{\text{pred}} = \mathcal{L}_{\text{weighted-CE}} + 0.3\mathcal{L}_{\text{NTL}}. \tag{4}$$

where $\mathcal{L}\text{weighted-CE}$ is a weighted cross-entropy (CE) loss that provides token-level supervision over the structured collision diagnosis. $\mathcal{L}_{\text{NTL}}$ is applied only to tokens corresponding to the four continuous spatial variables and uses the digit-level WAS formulation (Zausinger et al., 2025) to provide numerical-distance supervision. We use Llama-3.2-3B (Grattafiori et al., 2024) as the backbone and apply LoRA (Hu et al., 2021) for parameter-efficient fine-tuning. Training follows a two-stage curriculum (Wang et al., 2021): the first stage uses synthetic single-segment geometry samples to learn basic spatial relations, while the second stage trains on path-execution data while replaying a subset of the synthetic samples.

#### 2.3.2 VERIFIER-GUIDED PATH REPAIR

After the collision diagnosis module localizes the collision, the local repair module aims to bypass the collision region and reconnect to the original local target or its allowable adjustment. For a collision on segment $W_k \to W_{k+1}$, we follow Sec. 2.2 and define the segment endpoint $g_0 = W_{k+1}$ as the original local target. A collision-free repair start pose preceding the collision is first identified as $s = (x_s, y_s, \psi_s)$, where $(x_s, y_s)$ is the repair start position and $\psi_s$ is the robot yaw angle in the world frame. Let $D \geq 0$ denote the maximum allowed target displacement. The repaired local target $g$ must satisfy $\|g - g_0\|_2 \leq D$. The repair policy conditions on the robot and environment geometry, diagnosed collision information, repair start pose $s$, and target constraint to generate local repair candidates. If a valid repair context cannot be constructed, the procedure returns failure. Let $p \in \mathbb{R}^2$ denote the first collision position provided by the preceding collision diagnosis module. Each repair candidate contains $m \in \{1, 2, 3\}$ planar waypoints represented as offsets from $p$:

$$Q = (\Delta_1, \cdots, \Delta_m), \quad q_i = p + \Delta_i, \quad g = q_m. \tag{5}$$

Here, $\Delta_i \in \mathbb{R}^2$ is the offset of the $i$-th repair waypoint from $p$, and $q_i$ is the corresponding waypoint. The final waypoint $g = q_m$ is the repaired local target and must satisfy $\|g - g_0\|_2 \leq D$. The first $m - 1$ waypoints bypass the collision region, while the final waypoint reconnects to the local target of the current path segment without skipping to a later waypoint. When $D = 0$, the original local target is preserved, i.e., $g = g_0$. During training, $p$ is obtained from the collision annotation; at inference, it is provided by the diagnosis module. The collision position $p$ serves as the coordinate reference for the repair waypoints, while repair execution starts from the repair start pose $s$.

For each candidate, the GV checks output validity and the target-displacement constraint, and verifies that the robot footprint remains collision-free with respect to obstacles and warehouse boundaries along the local repair path. Only candidates that pass these checks are retained for selection.

**Repair Policy Training.** We first perform supervised fine-tuning (SFT) on expert repairs generated by ERG and verified by GV. We then further optimize the repair policy using Group Relative Policy Optimization (GRPO) (Shao et al., 2024). For each repair context, the policy samples repair candidates and constructs the reward $R(Q)$ from geometric verification and path quality. At inference, the repair policy samples $N$ repair candidates under the same repair context, denoted by $Q^{(1)}, \cdots, Q^{(N)}$, where $N$ is the candidate sampling budget and $Q^{(j)}$ denotes the $j$-th repair candidate, following the representation above with one to three local repair waypoints. GV independently verifies all candidates, and the verified candidate set and final selection are defined as:

$$\mathcal{A} = \{Q^{(j)} \mid \text{GV}(Q^{(j)}) = \text{true},\ 1 \leq j \leq N\}, \quad Q^* = \arg\max_{Q \in \mathcal{A}} R(Q). \tag{6}$$

Here, $\mathcal{A}$ is the set of verified candidates and $Q^*$ is the selected repair. If no model-generated candidate passes verification, the platform invokes a deterministic planner under the same repair start pose and target constraints to generate a fallback repair, which is retained only if it passes GV. If no valid repair is found, the procedure reports failure. Training details are provided in Appendix B.4.

## 3 ROBOTPATH-BENCH

In this section, we introduce **RobotPath-Bench**, a benchmark for pre-collision path diagnosis and repair in transport-robot navigation, as shown in Figure 4. It includes large-scale path-execution data across multiple robot configurations and warehouse layouts (Sec. 3.1), and a hierarchical evaluation protocol for pre-collision path diagnosis and repair (Sec. 3.2).

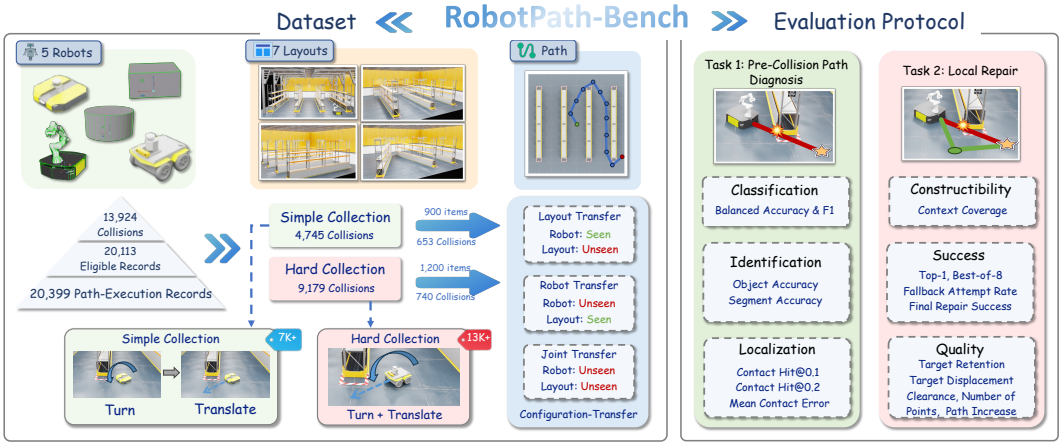



Figure 4: Composition of **RobotPath-Bench**.

### 3.1 Data Collection and Scale

**RobotPath-Bench** contains 20,399 path-execution records spanning 5 robot configurations and 7 shelf layouts. Each record includes robot geometry, environment geometry, planned waypoints, and the corresponding execution annotations. After filtering, 20,113 eligible records remain, including 13,924 records containing collisions. All retained collision records contain the robot center, collision positions, and colliding path-segment indices. To support evaluation under different execution regimes and configuration shifts, the benchmark is organized into Simple and Hard collections corresponding to the two execution regimes. Figure 4 summarizes the two collections.

**Simple collection.** The Simple collection comprises 7,200 records in total, consisting of 6,300 records collected under non-transfer configurations and 900 records reserved for configuration-transfer evaluation. Overall, 4,745 records contain collisions. The non-transfer subset is collected using seen layouts and robot configurations and constitutes the primary data split, including 4,092 collision-containing records. The configuration-transfer subset contains 900 records, of which 653 involve collisions, and is evenly partitioned into three evaluation settings: unseen layout (300 records), unseen robot (300 records), and unseen robot & layout (300 records).

**Hard collection.** The Hard collection comprises 13,199 records in total, consisting of 11,999 records collected under non-transfer configurations and 1,200 records reserved for configuration-transfer evaluation. Overall, 9,179 records contain collisions. The non-transfer subset is collected using seen layouts and robot configurations and constitutes the primary data split. After excluding 116 non-colliding executions terminated by timeout or stuck states, 11,883 eligible records remain, including 8,439 collision-containing records. The configuration-transfer subset contains 1,200 records, of which 740 involve collisions, and is partitioned into three evaluation settings: unseen layout (300 records), unseen robot (300 records), and unseen robot & layout (600 records). Further details on the data collection settings are provided in Appendix C.

### 3.2 Task Setup and Evaluation

**RobotPath-Bench** comprises two related tasks: pre-collision path diagnosis and repair. Pre-collision path diagnosis evaluates whether a model can identify the occurrence of a collision and localize the first collision along a planned path, while path repair evaluates whether the model can generate a geometrically feasible local repair conditioned on the collision context.

**Evaluation Protocol for Collision Diagnosis.** Collision diagnosis is evaluated in three aspects: *collision classification*, *collision attribute identification*, and *collision localization*. We use **Balanced Accuracy (BAcc)** and **F1 score (F1)** to determine whether a collision occurs. For collision attributes, **Object Accuracy (ObjAcc)** and **Segment Accuracy (SegAcc)** measure identification of the collision object and first colliding path segment, respectively. For collisions with valid coordinates, spatial localization is evaluated by the Euclidean distance between predicted and ground-truth

Table 1: Collision prediction variants on the Simple and Hard collections. All metrics are percentages except MCE (m). Best values within each collection are **bold**.

| | Classification | | Attribute ID | | Localization | | |
|---|---|---|---|---|---|---|---|
| **Model** | BAcc↑ | F1↑ | ObjAcc↑ | SegAcc↑ | Hit@0.1↑ | Hit@0.2↑ | MCE↓ |
| *Simple collection: in-distribution test set (617 paths, 411 collisions)* | | | | | | | |
| `LLM-RobotPath_Plain` | 97.33 | 98.04 | 96.59 | 83.21 | 73.3 | 85.8 | 0.1270 |
| `LLM-RobotPath_Digit` | 97.45 | 98.04 | 96.59 | 88.32 | **79.8** | **93.0** | 0.0875 |
| `LLM-RobotPath_DigitNTL` | **98.42** | **99.03** | **98.78** | **90.51** | 79.7 | 91.4 | **0.0820** |
| *Hard collection: in-distribution test set (929 paths, 638 collisions)* | | | | | | | |
| `LLM-RobotPath_Plain` | 91.01 | **95.33** | 86.17 | 74.92 | 59.00 | 64.15 | 1.2013 |
| `LLM-RobotPath_Digit` | 90.28 | 95.30 | 87.10 | 75.64 | 61.46 | 66.88 | 1.0892 |
| `LLM-RobotPath_DigitNTL` | **93.66** | 94.93 | **88.14** | **76.10** | **64.24** | **68.47** | **1.0073** |

collision positions. We report **Mean Collision Error (MCE)**, **Hit@0.1**, and **Hit@0.2**, where Hit@$r$ denotes the fraction of samples whose collision-position error is no greater than $r$ meters.

**Evaluation Protocol for Path Repair.** Path repair is evaluated along three aspects: *repair-context constructibility*, *repair success*, and *repair quality*. For repair-context constructibility, **Context Coverage (CC)** measures the fraction of collisions for which a valid repair context can be constructed. For repair success, **Top-1** and **Best-of-8 (Bo8)** evaluate the model's ability to generate geometrically feasible repairs on valid repair contexts. Top-1 uses greedy decoding, while Bo8 evaluates eight stochastic samples. If all model-generated candidates fail verification, a planner fallback is invoked; **Fallback Attempt Rate (FAR)** and **Final Repair Success (FRS)** are then reported to characterize the performance of the complete repair pipeline. For repair quality, Target Retention and Target Displacement measure preservation of the original local target, while Clearance, Number of Points, and Path Increase characterize the geometric quality of the repaired path.

# 4 Experiments

We evaluate **`RobotPath`** from three complementary perspectives. Sec. 4.1 examines the evaluation dimensions provided by **`RobotPath-Bench`** for pre-collision path diagnosis and repair. Sec. 4.2 validates the reliability of the platform's repair generation and verification tools, as well as the resulting repair annotations. Sec. 4.3 further examines whether the benchmark provides informative evaluation conditions under configuration shifts and varying repair constraints.

## 4.1 Overall Results on RobotPath-Bench

### 4.1.1 Collision Prediction

**Setting.** We evaluate collision diagnosis on the Simple and Hard collections, each split into training, validation, and test sets at an 8:1:1 ratio. For each execution regime, we consider four evaluation settings: an in-distribution test set and three configuration-transfer settings (unseen layout, unseen robot, and unseen robot & layout) to assess generalization across configurations. We compare three numerical representation and supervision variants of **`LLM-RobotPath`**: **`LLM-RobotPath_Plain`**, which generates continuous values in standard numeric form with cross-entropy supervision; **`LLM-RobotPath_Digit`**, which represents numerical values as digit-level tokens; and **`LLM-RobotPath_DigitNTL`**, which further incorporates Number Token Loss for numerical-distance supervision. Appendix B provides training settings and evaluation details.

**Results.** Tables 1 and 2 show that **`RobotPath-Bench`** reveals performance differences in collision diagnosis across multiple evaluation dimensions. The three variants show only minor differences in collision classification, but exhibit clearer differences in colliding segment identification and collision localization. On the Simple test set, **`LLM-RobotPath_DigitNTL`** improves SegAcc from 83.21% to 90.51% and reduces MCE from 0.1270 m to 0.0820 m compared with **`LLM-RobotPath_Plain`**. Meanwhile, Hit@0.2 and MCE do not vary consistently, indicating that threshold-based localization accuracy and average localization error capture different error characteristics. Spatial localization performance also degrades substantially under the Hard execu-

Table 2: Collision prediction under configuration-transfer settings.

| Setting | Classification | | Attribute ID | | Localization | | |
|---|---|---|---|---|---|---|---|
| | BAcc↑ | F1↑ | ObjAcc↑ | SegAcc↑ | Hit@0.1↑ | Hit@0.2↑ | MCE↓ |
| *Simple collection: configuration-transfer settings* | | | | | | | |
| Unseen layout | **99.77** | **99.77** | **82.87** | **83.80** | **63.26** | **81.40** | **0.1484** |
| Unseen robot | 95.17 | 94.92 | 56.52 | 66.18 | 16.04 | 31.55 | 0.5617 |
| Unseen robot & layout | 87.76 | 86.76 | 59.13 | 60.43 | 14.12 | 27.68 | 0.5254 |
| All unseen (900 paths) | 94.13 | 93.92 | 66.16 | 69.98 | 32.99 | 48.88 | 0.3971 |
| *Hard collection: configuration-transfer settings* | | | | | | | |
| Unseen layout (300) | 69.85 | 65.05 | **48.15** | **19.44** | 0.93 | **6.54** | 5.0656 |
| Unseen robot (300) | **89.18** | 87.87 | 35.10 | 6.53 | 0.00 | 1.04 | 4.3307 |
| Unseen robot & layout (600) | 88.02 | **93.07** | 43.15 | 9.82 | **1.60** | 3.46 | **4.2137** |
| All unseen (1,200 paths) | 78.33 | 85.77 | 41.22 | 10.14 | 1.04 | 3.26 | 4.3820 |

tion regime and configuration-transfer settings, even when collision classification remains relatively strong. These results demonstrate that the benchmark evaluates not only collision classification, but also object, segment, and localization for collision, while further characterizing robustness and generalization across execution regimes and configuration shifts.

#### 4.1.2 Path Repair

**Setting.** We evaluate path repair on 400 valid repair contexts constructed from 411 test collisions in the primary records of the Simple collection. Under the same repair constraints, we compare two policies: **`LLM-RobotPath_SFT`**, trained with SFT only, and **`LLM-RobotPath_SFT_GRPO`**, further optimized with GRPO after SFT, with a maximum target displacement of $D = 3$ m.

**Results.** Table 3 reports the results of path repair. If all model-generated candidates fail geometric verification, the platform invokes a deterministic planner as fallback. Here, Top-1 and Bo8 measure direct model repair capability, while FAR and FRS characterize fallback usage and final repair success. Compared with **`LLM-RobotPath_SFT`**, **`LLM-RobotPath_SFT_GRPO`** increases Top-1 from 68.00% to 92.50% and Bo8 from 94.25% to 96.00%, while reducing FAR from 5.75% to 4.00%. Both variants achieve over 99% final repair success after fallback. Overall, **`RobotPath-Bench`** distinguishes direct model repair capability from fallback-assisted system performance, revealing performance differences that final repair success alone would obscure.

Table 3: Path repair on 400 valid contexts with $D = 3$ m. $\star$ is a separately selected GRPO result.

| Method | CC (%) ↑ | Top-1(%) ↑ | Bo8(%) ↑ | FAR (%) ↓ | FRS (%) ↑ |
|---|---|---|---|---|---|
| `LLM-RobotPath_SFT` | **97.32** | 68.00 | 94.25 | 5.75 | 99.25 |
| `LLM-RobotPath_SFT_GRPO` | **97.32** | **92.50** | **96.00** | **4.00** | **99.50**$^\star$ |

### 4.2 Reliability of Platform Tools

**Setting.** To evaluate the execution reliability of the platform's repair-generation and verification tools, 1,263 collision records are randomly sampled, of which 1,213 yield valid repairs after `GV`. We independently execute the verified repair paths in Isaac Sim. A repair is considered successful if the robot reaches the specified local target without collision.

**Results.** Among the 1,213 verified repairs, 1,202 safely reach the local target, while 10 result in collisions and one times out, yielding a 99.09% safe-arrival rate. An additional 50 cases are marked as `safe_stop_required`, indicating that no valid repair is generated by the platform, and are excluded from the above statistics. The consistency between geometric verification and simulator execution supports the reliability of the platform's local repair generation and verification pipeline.

### 4.3 Benchmark Analysis and Findings

**Evaluation under Configuration Shifts.** Table 2 evaluates three configuration-transfer settings: unseen layout, unseen robot, and unseen robot & layout. The results show that different types of

configuration shifts lead to different performance changes. For example, BAcc and Hit@0.2 are 99.77% and 81.40% under the unseen-layout setting in the Simple collection, respectively. Hit@0.2 drops to 31.55% under the unseen robot setting and further to 27.68% when both the robot and layout are unseen. This stratified evaluation reveals distinct performance variations associated with layout changes, robot changes, and their combination, enabling a more fine-grained assessment of generalization under configuration shifts.

**Different Repair Constraints.** To examine the sensitivity of repair performance to target-displacement constraints, we evaluate **LLM-RobotPath_SFT** and **LLM-RobotPath_SFT_GRPO** under three maximum target displacement constraints: $D = 3$ m, $D = 0.25$ m, and $D = 0$ m. As shown in Figure 5, Context Coverage (CC) decreases substantially as the repair constraint becomes stricter, from 97.32% at $D = 3$ m to 58.15% at $D = 0.25$ m and further to 1.95% at $D = 0$ m. The relative performance of the two policies also varies with the constraint. At $D = 3$ m, **LLM-RobotPath_SFT_GRPO** improves Top-1 from 68.00% to 92.50%. However, at $D = 0.25$ m, it underperforms **LLM-RobotPath_SFT** in both Top-1 and Bo8. These results show that **RobotPath-Bench** captures both the reduction in valid repair contexts and the performance variations of repair policies under different repair constraints.

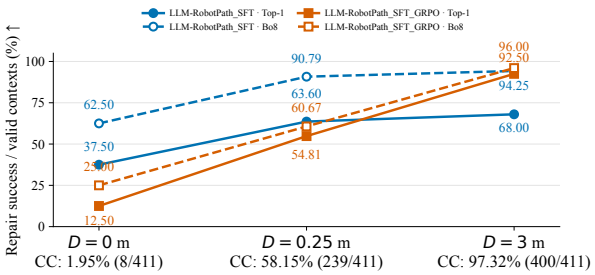



Figure 5: Repair performance under different target-displacement bounds.

## 5 Related Work

Transport robots executing preplanned paths can still collide with environmental obstacles, potentially causing payload loss and interrupting task execution. This motivates collision diagnosis and local repair before path execution. Existing studies on robot collision handling mainly focus on post-collision recovery, local replanning, and trajectory correction during execution. Post-collision recovery methods aim to restore robot motion after a collision has occurred (Lu et al., 2022; Abdullhak & Vardy, 2021; Boschetti et al., 2023; Bredenbeck et al., 2025). Local replanning methods generate alternative paths in response to environmental changes, model deviations, or newly detected obstacles (Lu et al., 2021; Tonola et al., 2023; Li et al., 2021b; Ramesh et al., 2024; Lee & Song, 2023; Guo et al., 2022; Tonola et al., 2025). Other studies correct deviations arising during trajectory tracking or execution, including those caused by dynamics, perception, human feedback, or policy execution (Dambly et al., 2025; Pan et al., 2025; Xia et al., 2026; Yu et al., 2026). Meanwhile, simulation platforms and benchmarks have been developed to support environment construction, navigation research, motion planning, and systematic evaluation. Platforms such as AI2-THOR (Kolve et al., 2022), Habitat (Savva et al., 2019), Habitat 2.0 (Szot et al., 2022), Habitat 3.0 (Puig et al., 2023), and iGibson (Li et al., 2021a) provide simulated environments, sensing interfaces, physics or object-state simulation, and task protocols for embodied-agent research. However, existing collision-handling methods primarily intervene during or after execution, while existing platforms and benchmarks mainly support environment simulation, navigation, motion planning, or trajectory-tracking evaluation. Overall, existing work still lacks a unified platform, dedicated data, and models for pre-collision path diagnosis and repair.

## 6 Conclusion

In this paper, we introduce **RobotPath**, a platform for pre-collision path diagnosis and local repair in transport-robot navigation. The platform supports flexible configuration across robots, shelf layouts, planned paths, and execution regimes, with integrated functions for path generation and execution, collision annotation, repair generation and verification, and training and evaluation of **LLM-RobotPath**. Based on this platform, we construct **RobotPath-Bench**, the first benchmark for this setting, enabling systematic evaluation of collision diagnosis and local repair. Future work will explore broader configurations, dynamic environments, and physical-robot validation.

## AI USE STATEMENT

We used AI-assisted tools to improve the clarity and grammar of the manuscript and to assist with LaTeX formatting. All suggested changes were reviewed and edited by the authors, who take full responsibility for the final content of the paper.

## REFERENCES

Mohammed Abdullhak and Andrew Vardy. Deadlock prediction and recovery for distributed collision avoidance with buffered voronoi cells. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 429–436. IEEE, 2021.

Mohamed Aghzal, Xiang Yue, Erion Plaku, and Ziyu Yao. Evaluating Vision-Language Models as Evaluators in Path Planning. arXiv preprint arXiv:2411.18711, 2025. URL https://arxiv.org/abs/2411.18711.

Giovanni Boschetti, Maurizio Faccio, Irene Granata, and Riccardo Minto. 3d collision avoidance strategy and performance evaluation for human–robot collaborative systems. *Computers & Industrial Engineering*, 179:109225, 2023.

Anton Bredenbeck, Teaya Yang, Salua Hamaza, and Mark W Mueller. A tactile feedback approach to path recovery after high-speed impacts for collision-resilient drones. *Drones*, 9(11):758, 2025.

Constantinos Chamzas, Carlos Quintero-Peña, Zachary Kingston, Andreas Orthey, Daniel Rakita, Michael Gleicher, Marc Toussaint, and Lydia E. Kavraki. MotionBenchMaker: A Tool to Generate and Benchmark Motion Planning Datasets. arXiv preprint arXiv:2112.06402, 2022. URL https://arxiv.org/abs/2112.06402.

Cheng Chang, Jiawei Zhang, Jingwei Ge, Zuo Zhang, Junqing Wei, Li Li, and Fei-Yue Wang. Vistascenario: Interaction scenario engineering for vehicles with intelligent systems for transport automation. *IEEE Transactions on Intelligent Vehicles*, 2024.

Peng Chen, Jing Liang, Kang-Jia Qiao, Hui Song, Ponnuthurai Nagaratnam Suganthan, Lou-Lei Dai, and Xuan-Xuan Ban. A reinforced neighborhood search method combined with genetic algorithm for multi-objective multi-robot transportation system. *IEEE Transactions on Intelligent Transportation Systems*, 26(8):12586–12599, 2025.

Valentin Dambly, Bryan Olivier, Edouard Rivière-Lorphèvre, François Ducobu, and Olivier Verlinden. Iterative offline trajectory correction based on dynamic model for compensating robot-dependent errors in robotic machining. *Robotics and Computer-Integrated Manufacturing*, 94: 102960, 2025.

Sicong Gao, Maurice Pagnucco, Tomasz Bednarz, and Yang Song. Nvidia isaac sim: Enabling scalable, gpu-accelerated simulation for robotics. *arXiv preprint arXiv:2606.03551*, 2026.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Jiuming Guo, Jiwen Zhang, Dan Wu, Yuhang Gai, and Ken Chen. A local manipulation path replanning algorithm on deformable linear objects for collisions resulted from model deviation. *Journal of Manufacturing Systems*, 65:362–377, 2022.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, Aniruddha Kembhavi, Abhinav Gupta, and Ali Farhadi. AI2-THOR: An Interactive 3D Environment for Visual AI. arXiv preprint arXiv:1712.05474, 2022. URL https://arxiv.org/abs/1712.05474.

Chuan-Che Lee and Kai-Tai Song. Path re-planning design of a cobot in a dynamic environment based on current obstacle configuration. *IEEE Robotics and Automation Letters*, 8(3):1183–1190, 2023.

Chengshu Li, Fei Xia, Roberto Martín-Martín, Michael Lingelbach, Sanjana Srivastava, Bokui Shen, Kent Vainio, Cem Gokmen, Gokul Dharan, Tanish Jain, Andrey Kurenkov, C. Karen Liu, Hyowon Gweon, Jiajun Wu, Li Fei-Fei, and Silvio Savarese. iGibson 2.0: Object-Centric Simulation for Robot Learning of Everyday Household Tasks. arXiv preprint arXiv:2108.03272, 2021a. URL https://arxiv.org/abs/2108.03272.

Shiqi Li, Ke Han, Xiao Li, Shuai Zhang, Youjun Xiong, and Zheng Xie. Hybrid trajectory replanning-based dynamic obstacle avoidance for physical human-robot interaction. *Journal of Intelligent & Robotic Systems*, 103(3):41, 2021b.

Zhouyu Lu, Zhichao Liu, and Konstantinos Karydis. Deformation recovery control and post-impact trajectory replanning for collision-resilient mobile robots. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2030–2037. IEEE, 2021.

Zhouyu Lu, Zhichao Liu, Merrick Campbell, and Konstantinos Karydis. Online search-based collision-inclusive motion planning and control for impact-resilient mobile robots. *IEEE Transactions on Robotics*, 39(2):1029–1049, 2022.

Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. Isaac Gym: High Performance GPU-Based Physics Simulation For Robot Learning. arXiv preprint arXiv:2108.10470, 2021. URL https://arxiv.org/abs/2108.10470.

Shaohui Pan, Yong Xu, Ruotao Xu, Zihan Zhou, Si Wu, and Zhuliang Yu. Self-correcting robot manipulation via gaussian-splatted foresight. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 26642–26650, 2025.

Xavier Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Tsung-Yen Yang, Ruslan Partsey, Ruta Desai, Alexander William Clegg, Michal Hlavac, So Yeon Min, Vladimír Vondruš, Theophile Gervet, Vincent-Pierre Berges, John M. Turner, Oleksandr Maksymets, Zsolt Kira, Mrinal Kalakrishnan, Jitendra Malik, Devendra Singh Chaplot, Unnat Jain, Dhruv Batra, Akshara Rai, and Roozbeh Mottaghi. Habitat 3.0: A Co-Habitat for Humans, Avatars and Robots. arXiv preprint arXiv:2310.13724, 2023. URL https://arxiv.org/abs/2310.13724.

Megnath Ramesh, Frank Imeson, Baris Fidan, and Stephen L Smith. Anytime replanning of robot coverage paths for partially unknown environments. *IEEE Transactions on Robotics*, 40:4190–4206, 2024.

Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A Platform for Embodied AI Research. arXiv preprint arXiv:1904.01201, 2019. URL https://arxiv.org/abs/1904.01201.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. arXiv preprint arXiv:2402.03300, 2024. URL https://arxiv.org/abs/2402.03300.

Andrew Szot, Alex Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Chaplot, Oleksandr Maksymets, Aaron Gokaslan, Vladimir Vondrus, Sameer Dharur, Franziska Meier, Wojciech Galuba, Angel Chang, Zsolt Kira, Vladlen Koltun, Jitendra Malik, Manolis Savva, and Dhruv Batra. Habitat 2.0: Training Home Assistants to Rearrange their Habitat. arXiv preprint arXiv:2106.14405, 2022. URL https://arxiv.org/abs/2106.14405.

Cesare Tonola, Marco Faroni, Manuel Beschi, and Nicola Pedrocchi. Anytime informed multi-path replanning strategy for complex environments. *IEEE Access*, 11:4105–4116, 2023.

Cesare Tonola, Marco Faroni, Saeed Abdolshah, Mazin Hamad, Sami Haddadin, Nicola Pedrocchi, and Manuel Beschi. Reactive and safety-aware path replanning for collaborative applications. *IEEE Transactions on Automation Science and Engineering*, 2025.

Usman Ahmad Usmani, Ari Happonen, and Junzo Watada. Revolutionizing transportation: Advancements in robot-assisted mobility systems. In *International conference on ICT for sustainable development*, pp. 603–619. Springer, 2023.

Xin Wang, Yudong Chen, and Wenwu Zhu. A survey on curriculum learning. *IEEE transactions on pattern analysis and machine intelligence*, 44(9):4555–4576, 2021.

Wenke Xia, Yichu Yang, Hongtao Wu, Xiao Ma, Tao Kong, and Di Hu. Human-assisted robotic policy refinement via action preference optimization. *Advances in Neural Information Processing Systems*, 38:36746–36768, 2026.

Xuesu Xiao, Zifan Xu, Zizhao Wang, Yunlong Song, Garrett Warnell, Peter Stone, Tingnan Zhang, Shravan Ravi, Gary Wang, Haresh Karnan, Joydeep Biswas, Nicholas Mohammad, Lauren Bramblett, Rahul Peddi, Nicola Bezzo, Zhanteng Xie, and Philip Dames. Autonomous Ground Navigation in Highly Constrained Spaces: Lessons learned from The BARN Challenge at ICRA 2022. arXiv preprint arXiv:2208.10473, 2022. URL https://arxiv.org/abs/2208.10473.

Shinya Yasuda, Taichi Kumagai, and Hiroshi Yoshida. Cooperative transportation robot system using risk-sensitive stochastic control. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5981–5988. IEEE, 2021.

Xinbo Yu, Xiong Guo, Wei He, Muhammad Arif Mughal, and Dawei Zhang. Real-time trajectory planning and obstacle avoidance for human–robot co-transporting. *IEEE Transactions on Automation Science and Engineering*, 22:2969–2985, 2024.

Zhuoyuan Yu, Yuxing Long, Zihan Yang, Chengyan Zeng, Hongwei Fan, Jiyao Zhang, and Hao Dong. Correctnav: Self-correction flywheel empowers vision-language-action navigation model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 18737–18745, 2026.

Jonas Zausinger, Lars Pennig, Anamarija Kozina, Sean Sdahl, Julian Sikora, Adrian Dendorfer, Timofey Kuznetsov, Mohamad Hagog, Nina Wiedemann, Kacper Chlodny, Vincent Limbach, Anna Ketteler, Thorben Prein, Vishwa Mohan Singh, Michael Morris Danziger, and Jannis Born. Regress, Don't Guess – A Regression-like Loss on Number Tokens for Language Models. arXiv preprint arXiv:2411.02083, 2025. URL https://arxiv.org/abs/2411.02083.

Ziyan Zhao, Xingyang Li, Shixin Liu, MengChu Zhou, and Xiaochun Yang. Multi-mobile-robot transport and production integrated system optimization. *IEEE Transactions on Automation Science and Engineering*, 22:7480–7491, 2024.