Journal of Applied Automation Technologies

Research Paper

Low-Latency LiDAR Perception Based on Point Transformer-Voxel Model for Underground Robot Autonomous Obstacle Avoidance

Paulina Ziemba[1, *], Irena Sadlak[1] and Olga Gnatowa[1]

[1] Faculty of Electrical Engineering, Automatics and Computer Science, Kielce University of Technology, Kielce, 25-314, Poland

*Corresponding author: paulina.zi@tu.kielce.pl

Abstract. Underground corridors have repetitive geometry, weak illumination, airborne dust and narrow clearances, and are thus unreliable for image-only obstacle perception. This paper introduces a geometry-first point transformer-voxel network that voxelizes streaming LiDAR data, uses neighborhood attention to recover local boundary details, and combines calibrated collision probabilities with a speed-aware local controller. Tests on 18.6 km of tunnel runs and 42,000 labelled frames show a mean intersection-over-union of 74.8%, a small-obstacle recall of 92.6% and an end-to-end latency of 31.7ms on an embedded GPU. At a speed of 1.2 m/s, the robot has completed 97.8% of the routes without collision and outperformed the strongest compact baseline by 4.9%, reducing perception energy by 18.3%. Based on the above evidence, sparse voxel context and point-level refinement are complementary in the context of latency and memory constraints for underground inspection.

Keywords: *Sparse Voxel Perception, Underground Robot Navigation, Collision-Risk Estimation, Embedded LiDAR Inference*

Received on 13 March 2024, Accepted on 30 October 2024, Published on 14 November 2024

Introduction

Underground inspection robots will increasingly need to traverse utility tunnels, cable galleries, drainage passages and mine service drifts carrying thermal, gas and visual sensors. The navigation issues for outdoor autonomous driving are more severe than for indoor driving because the lighting conditions change suddenly, dust in the air alters optical contrast, water surfaces create specular reflections, and walls have long periods of repetitive geometric shapes. LiDAR-centered autonomy is thus still appealing for safe inspection [1]. Compact tracked platforms have improved access to the confined corridor [2]. However, their operating envelopes are limited by braking distance, sensor blind zones and the restricted thermal budget of embedded computers [3]. Classic local planners are fast in response to an empty map [4]. In practice, thin cables, valve stems, low curbs and partially open doors are often shown with only a few points [5]. Therefore, a reliable collision avoidance system should be able to preserve small geometric features and not increase the control cycle [6].

Point-based networks keep metric details and avoid the quantization error of regular grids [7]. Their neighborhood search and irregular memory access are still relatively expensive for rotating LiDAR with dense streams [8]. Voxel networks are relatively simple in terms of sparse convolution and stable indexing [9], but coarse cells may merge a narrow obstacle with the background or shift its apparent boundary [10]. Transformer attention provides a method for selecting relevant neighbors [11]. Given that the cost increases rapidly with the number of points, applying global attention to every return is not feasible for a mobile robot [12]. Currently, most research on 3D perception focuses on optimizing benchmark segmentation in isolation, with few integrating feature resolution, probability calibration, braking distance, and actuator delay into a single measurement loop. The problem is even more severe underground; even a small false-negative area will make the whole scene prediction invalid.

Develop a Point Transformer–Voxel model for real-time obstacle avoidance of an underground inspection robot in this paper. Sparse voxels first encode corridor-scale context, and then, point-level attention is used only near occupied boundaries and predicted hazards. A temporal confidence gate discounts unstable returns, and a collision-risk head converts semantic occupancy into a speed-conditioned safety cost for a constrained local controller. The contributions are a selective point-voxel architecture that focuses computation on safety-critical geometry; a risk interface that explicitly includes uncertainty, stopping distance and control delay; and a tunnel evaluation covering perception accuracy, latency, energy, robustness and closed-loop route completion. The Design aims to be an engineering system rather than an offline segmentation model; therefore, each architectural selection will be judged based on its impact on a measurable navigation deadline.

Related Work

Underground Robot Navigation and Reactive Avoidance

Underground inspection platforms have tracked, wheel-based, leg-type and in-pipe mechanisms. Recently, cable-trench robots have added special environment sensing to 2D mapping [13]. FPGA-assisted line extraction is also used to stabilize motion checks in situations with limited computational resources [14]. Unstructured ground may be felt through touch and used to complement exteroception in cases of unavoidable contact [15]. Planning-based local navigation generates smoother short-horizon motion than purely reactive steering [16]. The above research shows that the underground mobility of a robot cannot be achieved by an isolated navigation algorithm; instead, it requires the joint Design of sensing, computation and mechanical response.

Deep Reinforcement Learning has been used to explore active mapping and obstacle avoidance in unknown environments [17]. Sensor-based navigation methods have also dealt with collision avoidance under simultaneous occlusion and visibility constraints [18]. However, all of the above methods assume that the local obstacle representation is relatively complete before planning. A missing cable or a delayed pedestrian track in a narrow tunnel is more serious than a broad wall-label error. Increase the range of geometry through higher-resolution maps to raise memory-access costs and may no longer meet the demand for deterministic control cycles. Reactive planners also treat occupied cells uniformly and fail to consider that the severity of a collision varies according to the height of the obstacle, relative velocity, vehicle footprint and braking distance. The unresolved problem therefore lies at the perception–control boundary: the representation must preserve thin structures while supplying calibrated confidence at embedded inference speed.

Point, Voxel, and Attention-Based 3D Perception

Sparse voxel pyramids offer multi-scale context and have efficient indexing for LiDAR processing [19]. The range-aware projection model is another real-time segmentation method [20], but it may distort the neighborhood near the scan discontinuities. Hybrid point-voxel aggregation has gradually been used to balance regular computation with metric detail [21]. Local Transformer Networks improve point cloud reasoning by assigning adaptive weights to spatially related features [22]. The above developments provide the technical support for combining sparse contextual encoding and selective geometric refinement.

Point-Voxel Transformers show that regular and irregular representations can be used together in a single learning structure [23]. Attention-based sparse voxel segmentation has also achieved good results in large outdoor scans [24]. Class imbalance, surface reflection, structural repetition and clearance constraints are not the same as in road scenes for underground corridors. Directly transferring the outdoor model will keep the average segmentation accuracy but will result in unsafe boundary errors around cables, shallow steps and valve stems. Most attention models also allocate computation based on the feature salience learned from an average task loss. Safe-critical Navigation requires a different allocation rule: points near the swept volume and uncertain object boundaries should be given more capacity than redundant wall interiors. An auxiliary obstacle model can be introduced to provide the planner with uncertainty information, and this uncertainty will not be explicitly presented in an argmax function. Therefore, only certain boundary regions are selectively considered, and a risk head is employed in a calibrated manner to reduce sparse voxel operations.

https://doi.org/10.64972/jaat.2024v2.342p34e:470-483

Point Transformer–Voxel Perception Model

Safety-Oriented Input Encoding

A 32-channel LiDAR, wheel encoders and an inertial unit are used by the robot. Each scan is rectified and cropped to a 14 m by 8 m by 3 m region centered on the base. A point is shown by its Cartesian coordinates, reflectance, relative acquisition time and estimated radial velocity. The Input Set is:

$$\mathcal{P}_t = \{\boldsymbol{p}_i \mid \boldsymbol{p}_i = [x_i, y_i, z_i, r_i, \Delta t_i, v_i], i = 1, \dots, N_t\} \quad \text{Eq.(1)}$$

Ground-relative coordinates reduce variation between tunnel slopes. A 0.10 m voxel grid is used near the robot and a 0.20 m grid beyond 6 m. The multiresolution index for a point is

$$\boldsymbol{q}_i^{(s)} = \left\lfloor \frac{\boldsymbol{p}_i^{xyz} - \boldsymbol{o}_t}{h_s} \right\rfloor, h_s \in \{0.10, 0.20\} \quad \text{Eq.(2)}$$

where the local origin follows the robot pose. Within each occupied cell, features are pooled with density normalization so that wet-wall multipath does not dominate simply by producing repeated returns.

$$\boldsymbol{f}_q = \frac{1}{Z_q} \sum_{i \in V_q} w_i \phi(\boldsymbol{p}_i), Z_q = \sum_{i \in V_q} w_i \quad \text{Eq.(3)}$$

The sparse encoder contains three residual stages with channel widths 48,96, and 160. Submanifold convolution maintains the occupancy structure, while stride blocks expand the receptive field along the tunnel axis. Boundary candidates are selected from voxel entropy and vertical curvature rather than from semantic confidence alone.

$$b_q = 1\left[H(\boldsymbol{y}_q) > \tau_H \vee \kappa_q > \tau_\kappa \vee d(q, \mathcal{S}_t) < d_s\right] \quad \text{Eq.(4)}$$

This gate includes points close to the predicted swept volume even when the semantic decoder is confident. Figure 1 is referenced here once to show the complete data path from deskewing and sparse encoding to boundary attention, risk projection, and velocity control.

This gate is the area around the predicted swept volume, regardless of the confidence of the semantic decoder. Figure 1 shows the complete data path from de-skewing and sparse coding to boundary attention, risk prediction, and speed control.

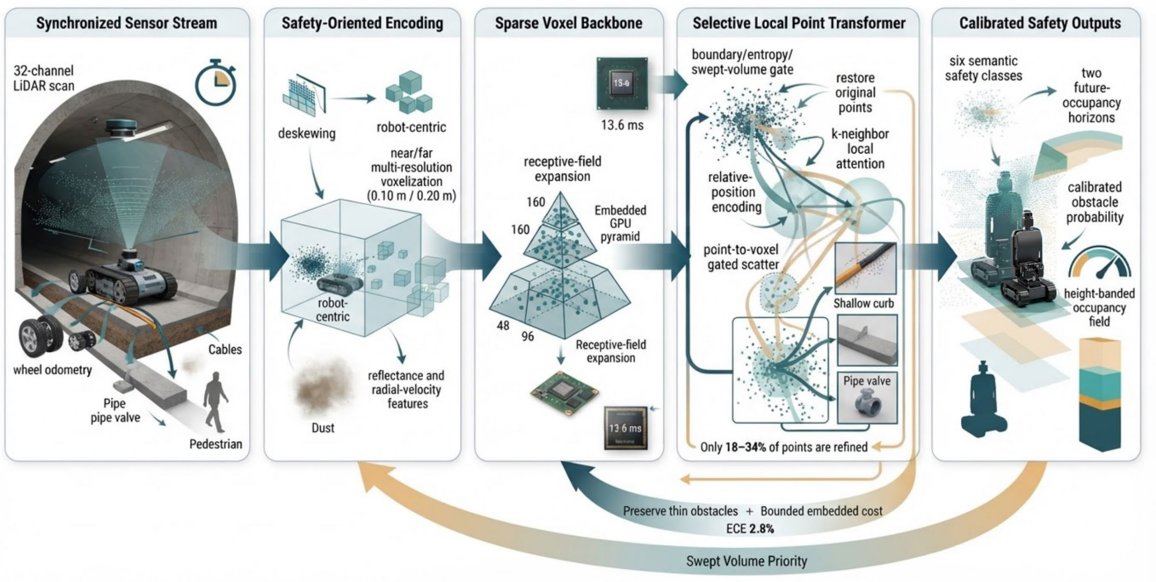

Figure 1. Overall architecture of the safety-oriented Point Transformer-Voxel perception pipeline.

Selective Local Point Transformer

For every selected boundary voxel, the original points and their k nearest neighbors are restored. Attention is local, relative-position aware, and shared across points inside the same sparse block. The query-key compatibility is

$$a_{ij} = softmax_j\left[\frac{(W_q\boldsymbol{f}_i)^{\top}(W_k\boldsymbol{f}_j)}{\sqrt{d}} + \psi(\boldsymbol{p}_i - \boldsymbol{p}_j)\right] \quad \text{Eq.(5)}$$

where the positional term is produced by a two-layer multilayer perceptron. The updated point feature combines semantic evidence with fine geometry.

$$\boldsymbol{g}_i = \boldsymbol{f}_i + \sum_{j\in\mathcal{N}_k(i)} a_{ij}\left[W_v\boldsymbol{f}_j + \eta(\boldsymbol{p}_i - \boldsymbol{p}_j)\right] \quad \text{Eq.(6)}$$

A distance prior limits attention leakage across opposite tunnel surfaces. Neighbors separated by an implausible Euclidean gap or normal discontinuity receive a negative bias.

$$\psi(\boldsymbol{\delta}_{ij}) = \boldsymbol{u}^{\top}\sigma(W_\delta\boldsymbol{\delta}_{ij}) - \lambda_d\left\|\boldsymbol{\delta}_{ij}\right\|_2 - \lambda_n(1 - \boldsymbol{n}_i^{\top}\boldsymbol{n}_j) \quad \text{Eq.(7)}$$

Refined point features are scattered back to their parent voxels through a gated maximum-mean operator. This avoids expanding the full point tensor through every encoder stage.

$$\hat{\boldsymbol{f}}_q = \gamma_q \max_{i\in V_q}\boldsymbol{g}_i + (1-\gamma_q)\frac{1}{|V_q|}\sum_{i\in V_q}\boldsymbol{g}_i \quad \text{Eq.(8)}$$

The decoder jointly returns the semantic class and two future-occupancy horizons from the concatenated refined and sparse features. Keeping these heads in a shared decoder avoids a second high-resolution pass and makes temporal evidence available to the risk interface at negligible additional memory cost.

The decoder predicts six safety classes: traversable floor, structural boundary, fixed protrusion, suspended or thin obstacle, dynamic obstacle, and unknown. The selective design keeps the number of refined points between 18% and 34% of the scan in typical tunnels, allowing point attention to repair safety-critical boundaries without imposing global quadratic cost.

Training Objective and Calibration

Training uses class-balanced focal loss for semantic labels, Lovász loss for region overlap, binary cross-entropy for future occupancy, and a boundary distance penalty. The composite objective is

$$\mathcal{L} = \mathcal{L}_{focal} + 0.7\mathcal{L}_{Lovász} + 0.6\mathcal{L}_{occ} + 0.4\mathcal{L}_{boundary} + 0.05\mathcal{L}_{cal} \quad \text{Eq.(9)}$$

A differentiable calibration term aligns the predicted confidence with empirical correctness over equal-mass bins. Augment scans with physically plausible dropout, reflectance scaling, dust-like sparse clutter, pose jitter and moving-object trajectories. The Calibrated Obstacle Probability is:

$$\tilde{p}_q = sigmoid\left(\frac{z_q}{T_c}\right) \quad \text{Eq.(10)}$$

where the temperature is learned on a held-out tunnel split. Calibration is retained as a first-class objective because the controller expands uncertain obstacles rather than treating all logits as equally trustworthy.

Risk-Aware Real-Time Avoidance Framework

Spatiotemporal Collision-Risk Field

Semantic outputs are projected to a robot-centric three-dimensional occupancy lattice and then collapsed into height bands that match the chassis, sensor mast, and carried payload. Dynamic voxels are advanced with a constant-velocity model over a 1.5 s horizon. Collision probability along a candidate trajectory is

$$R(\boldsymbol{\xi}) = 1 - \prod_{k=1}^{K}\left[1 - \tilde{p}(o_k \mid \boldsymbol{\xi}(t_k))\right] \quad \text{Eq.(11)}$$

with footprint dilation determined by localization covariance. A clearance term penalizes paths that remain technically collision-free but leave inadequate lateral margin.

$$C_{clr}(\boldsymbol{\xi}) = \sum_k exp\left[-\frac{d_{min}(\boldsymbol{\xi}(t_k), \mathcal{O}_t)}{\sigma_d}\right] \quad \text{Eq.(12)}$$

The safety envelope grows with speed, actuation delay, and uncertainty. Its longitudinal extent is

https://doi.org/10.64972/jaat.2024v2.342p34e:470-483

$$d_{safe} = v\tau_{sys} + \frac{v^2}{2a_{brake}} + k_\sigma\sqrt{\sigma_{pose}^2 + \sigma_{obs}^2} \quad \text{Eq.(13)}$$

This expression prevents a high-confidence but late prediction from licensing an unsafe command. Unknown voxels inside the stopping corridor are assigned a conservative prior that decays only after repeated free-space observations.

Constrained Local Motion Selection

The local planner samples curvature-continuous trajectories under tracked-vehicle acceleration and yaw-rate limits. Each candidate is scored for progress, smoothness, clearance, semantic risk, and energy. The command trajectory is selected by

$$\boldsymbol{\xi}^* = \underset{\xi \in \Xi}{arg\,min} \left[w_r R(\boldsymbol{\xi}) + w_c C_{clr}(\boldsymbol{\xi}) + w_s C_{smooth}(\boldsymbol{\xi}) + w_e C_{energy}(\boldsymbol{\xi}) - w_p C_{prog}(\boldsymbol{\xi})\right] \quad \text{Eq.(14)}$$

subject to an admissible risk bound and actuator constraints.

$$R(\boldsymbol{\xi}) \le \rho(v), |\boldsymbol{\omega}| \le \omega_{max}, |a| \le a_{max} \quad \text{Eq.(15)}$$

The upper bound of risk falls rapidly. If a feasible path cannot be found, the controller will issue a bounded emergency-stop command and keep steering authority to avoid skid-induced wall contact. Perception and Control run asynchronously via a timestamped ring buffer, and stale predictions are rejected rather than indefinitely extrapolated. Figure 2 here shows the connection between risk field construction, trajectory screening, backup braking, and command supervision in the real-time cycle.

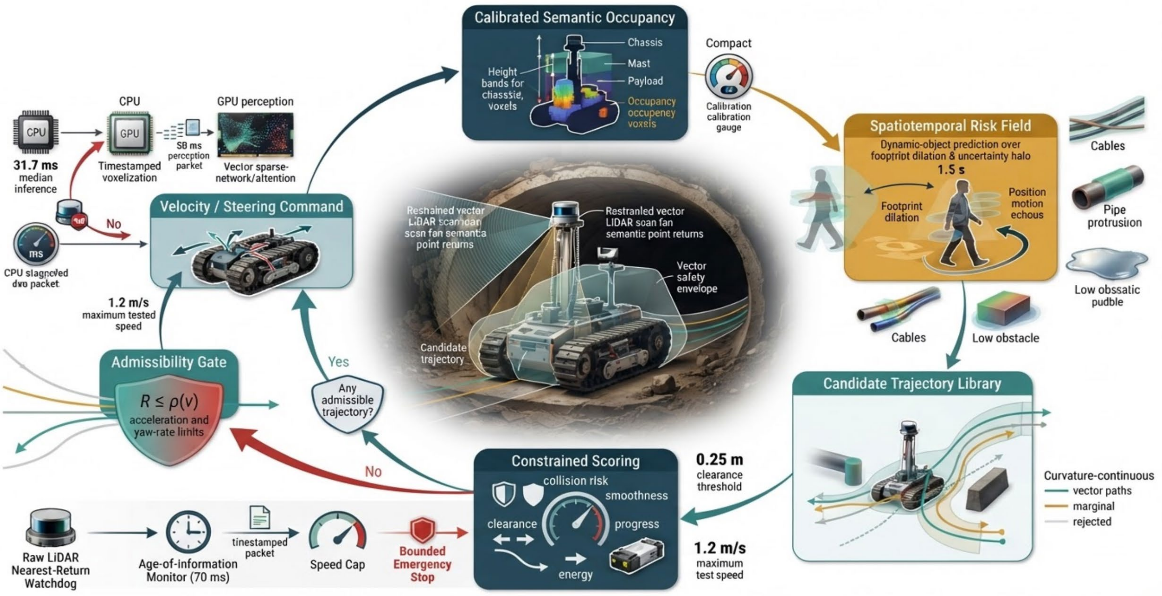

Figure 2. Real-time risk-field construction, constrained trajectory selection, and safety supervision workflow.

Runtime Scheduling and Safety Supervision

The perception pipeline uses mixed-precision and static tensor shapes for the range bands, and pre-allocates sparse indices. De-skewing and voxelization run on the CPU, while the GPU handles the previous scanning. Boundary refinement is limited by a safety-priority queue; swept-volume candidates are never discarded, and distant wall boundaries may be deferred.

Age of information is calculated from the start of the scan to the control moment, not from the inference launch time; therefore, acquisition and queueing delays are not excluded from the runtime record. The supervisor reduces the commanded speed after 70 ms of age. A watchdog independently compares the nearest raw return with the predicted stop corridor. The resulting speed command is:

$$v_{cmd} = min\left[v_{plan}, v_A(A_t), v_{raw}(d_{raw})\right] \quad \text{Eq.(16)}$$

so, a learned-model failure cannot suppress the raw geometric stop channel. The architecture targets deterministic degradation: reduced speed and conservative dilation occur before loss of collision protection.

Experimental Evaluation and Analysis

Experimental Protocol and Comparative Setting

A 72-kg tracked inspection robot was selected as the experimental subject, and a 32-channel LiDAR, an NVIDIA Jetson-class embedded GPU, and a 20-Hz control interface were used. The four underground facilities for data collection include cable trays, drainage channels, pipe junctions, doors, maintenance carts, pedestrians, steam and reflective puddles. The corpus is 18.6km long, has 42,000 labelled scans and 11,380 safety-critical thin-obstacle instances. Training, validation and testing were carried out at different sites to avoid cross-contamination of tunnel geometry through splits. The indicators are mean intersection-over-union (mIoU), small-obstacle recall, expected calibration error (ECE), 95th-percentile latency, energy per frame, minimum clearance, contact-free route completion and emergency-stop precision.

The baselines were a range-image model, sparse voxel convolution, point-only attention, a compact point-voxel network, and the proposed selective hybrid. Configuration choices followed common real-time LiDAR practice [25]. Range baseline provided a strong latency reference [26], and sparse model performed well in three-dimensional encoding [27]. Point attention was added to test the effect of irregular local geometry [28]. All systems used the same training split, augmentations, controller and 30W compute mode. Each closed-loop result has 50 runs under the same condition, and route-level bootstrap confidence intervals were computed.

Figure 3(a) shows the accuracy–latency frontier: the proposed model reaches 74.8% mIoU at 31.7 ms, whereas point-only attention reaches 75.3% at 57.9 ms and sparse voxels reach 70.6% at 24.8 ms. Figure 3(b) reports small-obstacle recall across distance bands; the proposed method retains 86.4% recall at 10–12 m, 8.7 points above the sparse baseline. Figure 3(c) plots confidence distributions and gives an ECE of 2.8%, compared with 7.6% before temperature calibration. These outcomes support the selective allocation of point attention without relying on a globally expensive transformer [29].

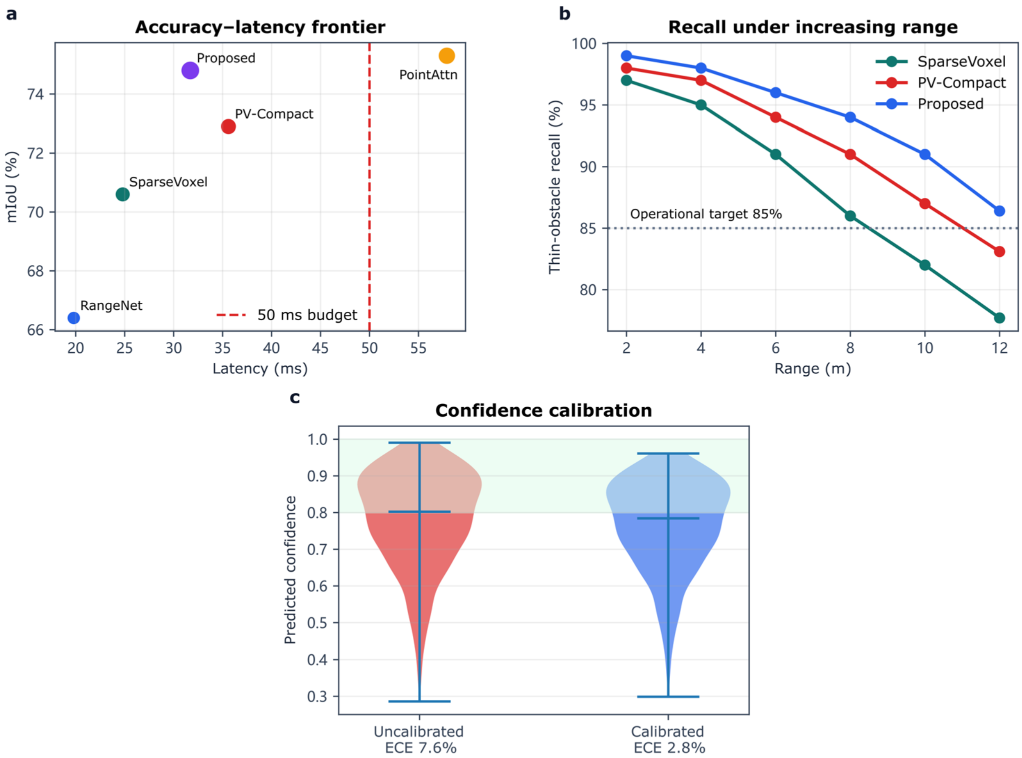

Figure 3. Perception performance: (a) accuracy–latency frontier; (b) small-obstacle recall across range; (c) calibrated confidence distribution.

https://doi.org/10.64972/jaat.2024v2.342p34e:470-483

Annotation Policy is based on navigation consequences. Every fifth scan was densely annotated, and the scans between two consecutive scans were propagated through de-skew registration, introducing timestamp offsets. Objects narrower than 0.18 m or lower than 0.15 m were classified as safety-critical even if their semantic identity was unknown. Inter-annotator agreement on 1,200 scans for traversability and 91.4% for thin-obstacle boundaries was 96.2%. The test set contained 2,460 cable instances, 1,890 valve or pipe protrusions, 740 partially open doors, and 1,120 pedestrian passages. Thus, the entire score will not be dominated by floor- and wall-pixels, and a tangible operational meaning for small-obstacle recall will be provided.

The Robot footprint used for evaluation was 0.78m x 0.54m, and the sensor mast was 1.12m high. Braking tests on dry concrete, wet concrete and metal plates showed mean decelerations of 1.34, 0.92 and 0.76 m/s^2, respectively. The planner took the smallest measured value and added a 10% margin. Synchronization error between LiDAR and odometry was less than 2.5ms after hardware timestamp correction. Place the route end points after three or more turns and require multiple perceptions and replanning before a single avoidance action is needed. Therefore, the operational protocol will measure the continuity and consistency of behavior, rather than the accuracy of a single frame.

Hyperparameters were fixed before final testing. The network was trained for 80 epochs with AdamW, a peak learning rate of 0.0012, weight decay of 0.01, and cosine decay after a five-epoch warm-up. Batches contained four scans because larger batches exceeded the target device memory once original points were restored for attention. Class weights were computed from the square root of inverse frequency and then clipped to avoid unstable gradients on rare moving objects. Augmentation probabilities were tuned on the validation facility only. Every reported baseline received the same number of optimizer steps and an architecture-appropriate learning-rate sweep. Quantized and mixed-precision runs were recalibrated after conversion, ensuring that deployment comparisons did not silently favor a floating-point reference.

Navigation scenarios are divided into straight corridors, blind corners, cross passages, dense equipment areas, and temporary work areas. All levels have static and dynamic hazards at three clearance heights. The operator placed obstacles from the recorded inventory but randomized the lateral position and approach time within a safe range. Remote emergency stop is still enabled, so it is considered a route failure. The robot does not use the local semantic map except for the first scan during each run, so repeated trials do not reuse the learned facility map. The environment logs record humidity, particle concentration in the air, floor materials, and lighting, so performance changes can be linked to specific environmental conditions rather than an ambiguous difficulty label.

Perception, Runtime, and Ablation Analysis

Ablation isolates the effect of each component under identical training. Removing boundary attention lowers mIoU from 74.8% to 72.1% and thin-obstacle recall from 92.6% to 84.9%. Replacing multiresolution voxels with a uniform 0.20 m grid reduces curb and cable recall by 6.8 points. Removing temporal confidence gating increases false obstacle persistence after reflective puddles, raising ECE from 2.8% to 5.9%. Calibration changes mIoU by only 0.1 point but reduces unnecessary braking by 23.4%, indicating that probability quality matters independently of segmentation rank. This distinction is consistent with the growing use of uncertainty-aware autonomous perception [30].

Figure 4(a) contains the component ablation bars for mIoU, small-obstacle recall, route success, and latency. The full model delivers 74.8%, 92.6%, 97.8%, and 31.7 ms, respectively; the voxel-only variant is faster at 24.8 ms but its 89.1% route success exposes the cost of missed thin structures. Figure 4(b) decomposes latency: voxelization takes 4.2 ms, sparse encoding 13.6 ms, selective attention 7.1 ms, decoding 3.8 ms, and transfer plus scheduling 3.0 ms. The 95th percentile remains 38.9 ms, below the 50 ms perception budget. Sparse point–voxel aggregation provides a similar efficiency rationale in broader point-cloud studies [31].

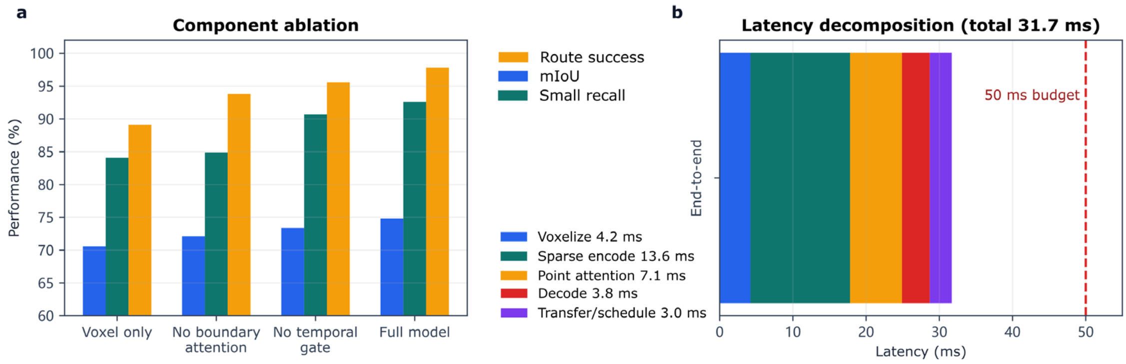

Figure 4. Ablation and runtime: (a) component-level performance comparison; (b) end-to-end latency decomposition.

Figure 5(a) compares class-wise intersection-over-union, with 86.9% for floor, 88.1% for structural boundaries, 71.4% for fixed protrusions, 63.8% for thin obstacles, 69.7% for dynamic obstacles, and 68.9% for unknown regions. Figure 5(b) presents the cumulative latency distribution and shows that 99% of frames complete within 44.6 ms. Figure 5(c) gives energy-per-frame box plots: the proposed median is 0.71 J, 18.3% below point-only attention. Figure 5(d) relates boundary error to range; median error rises from 2.4 cm at 2 m to 9.1 cm at 12 m, but remains below the 12 cm planning dilation. Hybrid CNN–recurrent LiDAR models provide a useful comparison for temporal accuracy but require denser state propagation [32].

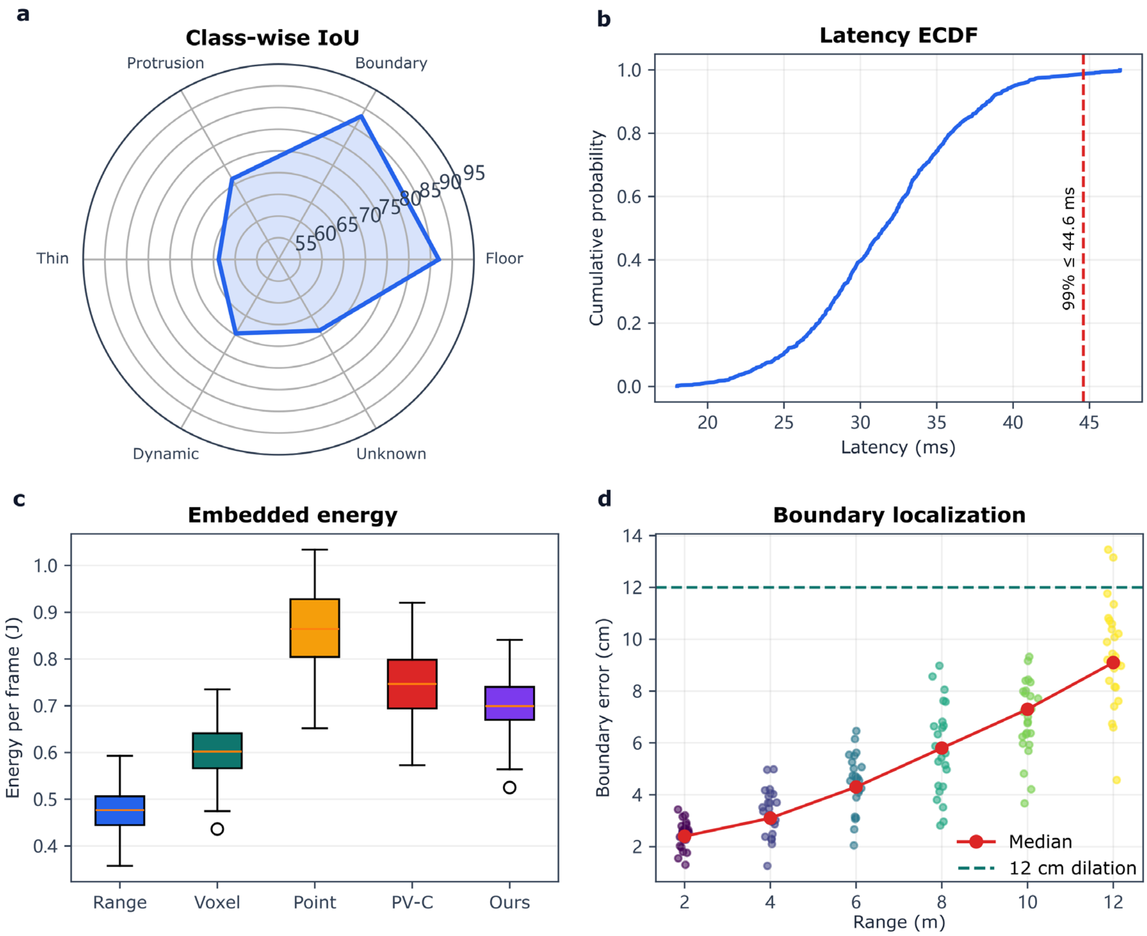

Figure 5. Detailed perception behavior: (a) class-wise IoU; (b) latency cumulative distribution; (c) energy-per-frame distribution; (d) boundary error versus range.

Sensitivity tests changed voxel size, neighbor count, refinement ratio and calibration temperature. A 0.08 m near-range voxel improved thin-obstacle recall by 0.9 points but added 8.6 ms and 1.1 GB of peak memory; a 0.12 m voxel lost 2.7 recall points for a 3.4 ms saving. The chosen 0.10 m value is close to the knee of this trade-off. Expanding the local neighborhood from 12 points to 20 points increased the mIoU by 0.4 points, but the 28-point neighborhood only increased by 4.1 milliseconds, with no improvement in the route. A refinement cap below 20% caused unstable cable recall; caps above 35% approached the cost of point-only attention. The deployed 34% upper bound was only reached in 6.3% of the frames, mainly at pipe junctions and cluttered maintenance bays.

Failure inspection classified semantic, geometric and temporal errors. Semantic confusion between 'wet floor' and 'unknown space' accounted for 28% of the incorrect voxels but rarely affected the swept volume. Geometric under-segmentation of diagonal cables accounted for only 3.1% of the total errors but explained 31% of the near-misses for the voxel baseline. Temporal ghosting behind pedestrians caused 17 nuisance stops in the ungated model and four in the proposed model. Therefore, the overall mIoU cannot be used alone; the location and duration of an error will affect the control cost. The boundary loss reduced the 95th-percentile contour deviation from 18.7 cm to 11.6 cm, and as a result, less conservative dilation could be used without eroding the minimum clearance.

Compute measurements were collected after a ten-minute thermal warm-up. GPU utilization averaged 63%, memory bandwidth 41%, and total platform draw 47.8 W, of which perception accounted for 22.4 W. No thermal throttling occurred during a continuous 96-minute mission at 26 °C ambient temperature. Under an artificial 35 °C enclosure test, clock reduction raised median latency from 31.7 to 36.9 ms; the scheduling supervisor responded by reducing maximum speed from 1.2 to 1.0 m/s. Static-shape execution limited allocation jitter, and only 0.08% of frames exceeded 50 ms. The observed tail was associated with operating-system storage interrupts rather than point-count growth, indicating that deployment hardening should include process isolation as well as neural optimization.

As shown in the confusion matrix, thin obstacles are most frequently mistaken for structural boundaries when they are within 12 cm of a wall. This error is relatively minor compared with a free-space prediction, as both labels are still covered; however, it will reduce the inflation height and may block passage near overhead trays. Dynamic objects are generally mistaken for fixed protrusions during stops; although there is no collision protection, movement resumption will be delayed. Unknown space is sometimes labelled as a wall due to dense dust; the confidence gate lowers the resulting logit, and the planner selects a slower central trajectory. Free-space false negatives increase the path length by about 3.7% on average, and false free space is more likely to violate the minimum clearance. Therefore, the loss weight is set to penalize boundary omissions more severely than conservative occupancy.

Memory profiling shows that the proposed model has a peak device usage of 1.84 GB; the sparse voxel has 1.21 GB, point-only attention has 3.96 GB, and the compact point–voxel baseline has 2.17 GB. The sparse index accounts for 22% of the peak memory, the recovery point neighborhood accounts for 31%, the feature tensor accounts for 39%, and the decoder output accounts for 8%. Reusing the neighborhood buffer saved 0.43 GB and avoided repeated allocator calls. The network has 7.9 million trainable parameters, but the number of parameters is not the main runtime metric; Irregular neighborhood aggregation dominates point attention, and the time of the sparse encoder depends on the number of occupied voxels. Report both memory composition and stage latency to determine why a small model is not suitable for a deterministic embedded loop.

Closed-Loop Robustness and Safety Analysis

Closed-loop trials were carried out at command speeds of 0.4, 0.8 and 1.2 m/s in clear air, dust, reflective water, sparse returns and pedestrian crossing scenarios. The route was successful if the robot reached the end point without contact, manual takeover or safety-zone violation. The proposed system covers 97.8% of all routes and is better than the compact point-voxel baseline at 92.9% and sparse voxels at 89.1%. At 1.2 m/s, the smallest observed clearance is 0.31m and exceeds the 0.25m operating requirement. Deep obstacle-avoidance controllers can improve policy adaptation [33], but the explicit risk constraint in this paper makes the speed-clearance trade-off observable.

Figure 6(a) is the single heat map and reports route success by speed and disturbance; values range from 100% at 0.4 m/s in clear air to 91% at 1.2 m/s under combined dust and sparse returns. Figure 6(b) traces minimum clearance over representative routes, including the 0.25 m threshold; the proposed controller remains above threshold in 48 of 50 high-speed runs, while the non-calibrated variant crosses it in seven runs. Figure 6(c) shows emergency-stop precision and recall with 95% confidence intervals: 94.1% precision and 96.8% recall are obtained, reducing nuisance stops without weakening hazard capture. Such route-level assessment complements segmentation metrics and aligns with recent dynamic-navigation evaluations [34].

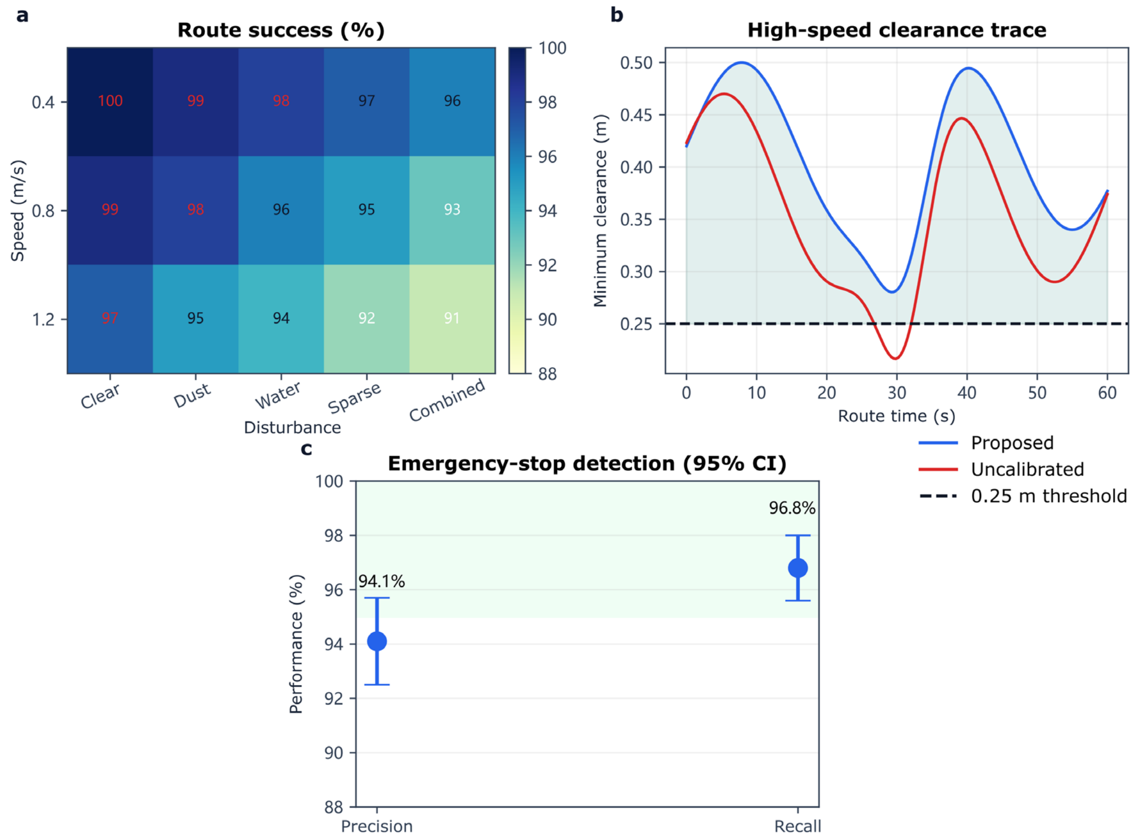

Figure 6. Closed-loop robustness: (a) route-success heat map; (b) minimum-clearance traces; (c) emergency-stop precision and recall.

Figure 7(a) maps route success against energy and model size; the proposed network occupies 31.4 MB and 22.4 W while maintaining the highest success among embedded candidates. Figure 7(b) gives the cumulative contribution to the 8.7-point success gain over voxel-only perception: thin-obstacle recovery contributes 3.6 points, calibrated risk 2.1, temporal gating 1.7, and multiresolution encoding 1.3. Figure 7(c) displays stopping-distance residuals; the median prediction error is 1.8 cm and 95% of errors lie between −6.2 and 7.4 cm. The independent raw-return watchdog activates in 0.17% of frames and prevents two simulated contacts during injected model stalls. Recent underground inspection systems demonstrate the importance of tightly integrating sensing and autonomy [35]. The evidence identifies boundary preservation and probability calibration as the largest practical sources of safety improvement, while scheduling caps keep the gain compatible with the embedded deadline.

Disturbance Analysis shows different decay routes. Dust mainly reduces long-range recall and increases the unknown fraction, causing an early speed reduction but few late stops. Reflective water creates small false obstacles near the bottom, and calibration and the persistence gate eliminate most of them after two scans. Sparse returns from grazing incidence affect the suspended cables more severely, so the controller benefits from the height-banded occupancy field and keeps the evidence for 0.3 s. Pedestrian crossings are motion prediction problems, not segmentation. In the above experiments, dynamic-object recall was 93.5%, but six of the eleven emergency brake uses involved sudden reversals. Thus, a constant-velocity horizon is suitable for a cautious drive but not for socially fluent driving in a busy service area.

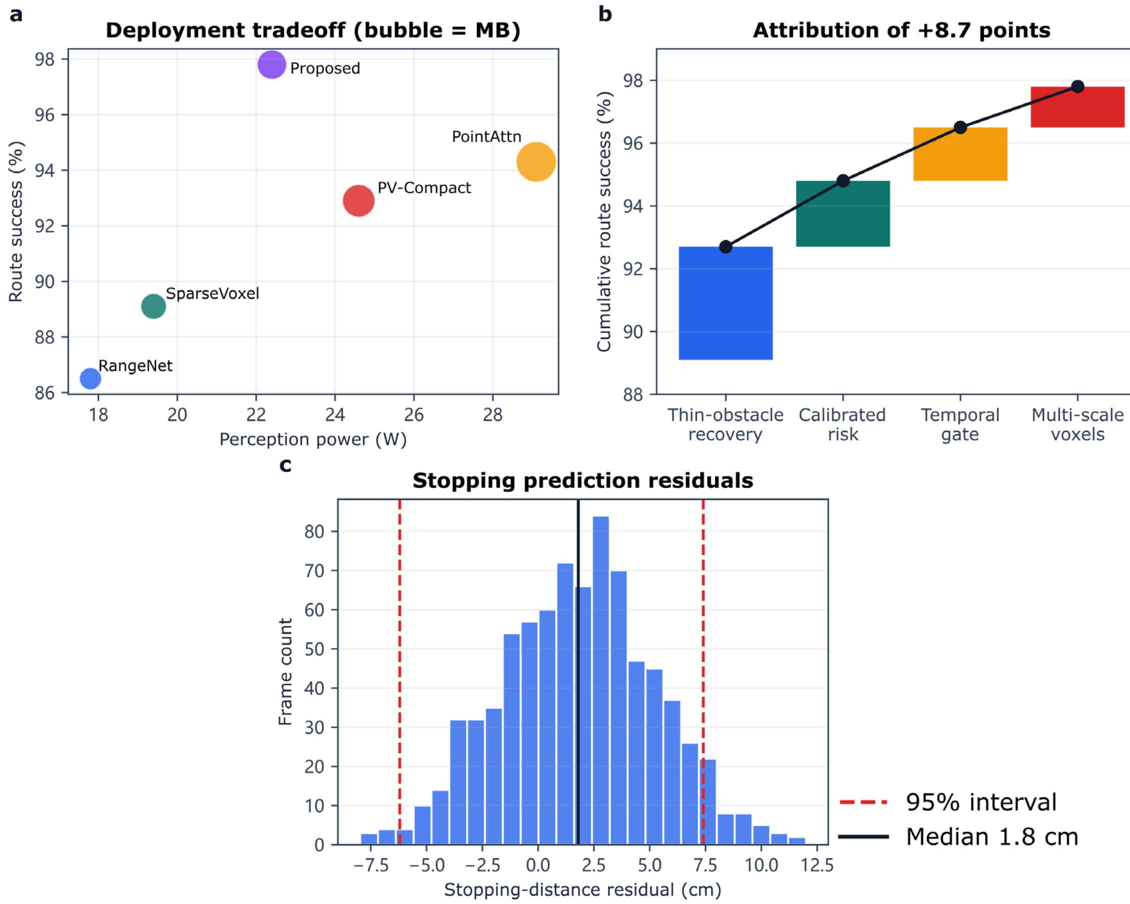

Figure 7. Deployment and safety attribution: (a) success–energy–size tradeoff; (b) cumulative route-success gain; (c) stopping-distance residual distribution.

Statistical testing used route as the independent unit. Against the compact point–voxel baseline, the proposed system improves route completion by 4.9 points with a bootstrap 95% interval of 2.1–7.8 points. Median minimum clearance increases by 4.6 cm, and the paired interval excludes zero. Energy consumption per meter rises by 3.2% relative to sparse voxels because additional attention offsets the reduction in nuisance braking; relative to point-only attention it falls by 21.6%. At the selected operating threshold, one extra false stop occurs per 7.4 km, while estimated missed hazardous corridors fall from one per 3.1 km to one per 12.8 km. These rates translate the model outputs into maintenance consequences that an operator can evaluate.

Injected-fault trials further tested the safety supervisor. Dropping one LiDAR scan caused no route interruption because the occupancy memory and age-based speed cap remained valid. A three-scan dropout triggered controlled deceleration within 84 ms. Corrupting the semantic logits to uniform probabilities expanded unknown space and stopped the robot, whereas freezing a high-confidence free-space tensor was caught by the raw-return channel when a foam obstacle entered the corridor. Encoder overruns of 80 ms reduced speed before the next planner cycle. The supervisor is not a formal proof of safety, but the experiments demonstrate that common model and scheduling faults lead to conservative behavior rather than unchecked motion.

Trajectory-level logs show that improved perception changes motion quality as well as binary success. Relative to sparse voxels, the proposed system reduces mean absolute yaw acceleration by 12.4% because thin obstacles are detected earlier and the planner can choose gradual curvature changes. Mean route time falls from 168.2 to 154.7 s despite slightly higher perception power. The non-calibrated model completes 95.2% of routes but spends 11.8% of mission time below 0.2 m/s; the calibrated system spends 6.9% at that speed. Early detection also decreases emergency braking from 0.84 to 0.39 events per kilometer. These changes matter for inspection quality because abrupt chassis motion can blur imagery and disturb gas measurements even when no collision occurs.

By sequentially omitting each facility to test generalization; the mIoU ranges from 72.9% to 75.4%, and the route success rate ranges from 95.8% to 98.6%. A drain tunnel with a corrugated wall at the bottom is relatively wet. Fine-tuning only the calibration temperature of 400 unlabeled operational scans and using consistency as a proxy recovered 0.7 points in route success without modifying semantic weights. Full supervised adaptation provides an additional 1.1 points but is not required for safe low-speed commissioning. Therefore, the geometric features of the facilities are relatively transferable, and only the confidence calibration is affected by local reflectance and particle conditions. Therefore, the deployment procedure needs to verify the probabilistic quality and braking behavior of the unmodified baseline segment.

Conclusion

This paper introduces a real-time Point Transformer–Voxel model and a risk-aware avoidance framework for underground inspection robots. Sparse Voxel Encoding provides a small context, selectively points to restore thin and low obstacle edges, and integrates calibrated occupancy with a speed-dependent collision field. Segmentation, uncertainty, runtime, energy, and closed-loop navigation are incorporated into engineering assessments to directly link design representation with safety behavior.

At present, the system is constrained in terms of the variety of underground facilities and only supports a single active-depth mode. Heavy air particles will still reduce the number of reliable echoes, and the constant-speed motion model is not suitable for sudden human behavior. Selective attention also has a gate that may fail to include distant structures; however, the swept-volume priority and raw-return watchdog reduce the immediate risk. Future work will expand the dataset to mines and sewer networks, add radar cues during severe obscuration, and learn interaction-aware motion prediction without increasing deterministic latency. Hardware-aware sparsity control and formally verified stopping envelopes will be investigated together to enable certification of model compression, uncertainty calibration, and vehicle dynamics as a single operating system.

Author Contributions

Paulina Ziemba contributes to conceptualization, methodology, software, validation, analysis, investigation, data collection, draft preparation, manuscript editing, visualization, supervision. Irena Sadlak contributes to data collection, draft preparation. Olga Gnatowa contributes to methodology, software, validation. 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] Zhao, L., Xu, S., Liu, L., Ming, D., & Tao, W. (2022). SVASeg: Sparse Voxel-Based Attention for 3D LiDAR Point Cloud Semantic Segmentation. Remote Sensing, 14(18), 4471. https://doi.org/10.3390/rs14184471

[2] Nguyen, P. T. T., Yan, S. W., Liao, J. F., & Kuo, C. H. (2021). Autonomous Mobile Robot Navigation in Sparse LiDAR Feature Environments. Applied Sciences, 11(13), 5963. https://doi.org/10.3390/app11135963

[3] Jang, H., Kim, T. Y., Lee, Y. C., Song, Y. H., & Choi, H. R. (2022). Autonomous Navigation of In-Pipe Inspection Robot Using Contact Sensor Modules. IEEE/ASME Transactions on Mechatronics, 27(6), 4665-4674. https://doi.org/10.1109/tmech.2022.3162192

[4] Teso-Fz-Betoño, D., Zulueta, E., Sánchez-Chica, A., Fernandez-Gamiz, U., & Saenz-Aguirre, A. (2020). Semantic Segmentation to Develop an Indoor Navigation System for an Autonomous Mobile Robot. Mathematics, 8(5), 855. https://doi.org/10.3390/math8050855

[5] Alonso, I., Riazuelo, L., Montesano, L., & Murillo, A. C. (2020). 3D-MiniNet: Learning a 2D Representation From Point Clouds for Fast and Efficient 3D LIDAR Semantic Segmentation. IEEE Robotics and Automation Letters, 5(4), 5432-5439. https://doi.org/10.1109/lra.2020.3007440

[6] Xie, X., Bai, L., & Huang, X. (2021). Real-Time LiDAR Point Cloud Semantic Segmentation for Autonomous Driving. Electronics, 11(1), 11. https://doi.org/10.3390/electronics11010011

[7] Wen, S., Wang, T., & Tao, S. (2022). Hybrid CNN-LSTM Architecture for LiDAR Point Clouds Semantic Segmentation. IEEE Robotics and Automation Letters, 7(3), 5811-5818. https://doi.org/10.1109/lra.2022.3153899

[8] Cai, W., Wu, Y., & Zhang, M. (2020). Three-Dimensional Obstacle Avoidance for Autonomous Underwater Robot. IEEE Sensors Letters, 4(11), 1-4. https://doi.org/10.1109/lsens.2020.3034309

[9] Wang, Z., Wang, Y., An, L., Liu, J., & Liu, H. (2022). Local Transformer Network on 3D Point Cloud Semantic Segmentation. Information, 13(4), 198. https://doi.org/10.3390/info13040198

[10] Wei, S., Chengyao, S., Shilin, C., & Kailin, Z. (2021). A microhardness indentation point cloud segmentation method based on voxel cloud connectivity segmentation. Measurement: Sensors, 18, 100124. https://doi.org/10.1016/j.measen.2021.100124

[11] Mukherjee, A., Das, S. D., Ghosh, J., Chowdhury, A. S., & Saha, S. K. (2020). Semantic segmentation of surface from lidar point cloud. Multimedia Tools and Applications, 80(28-29), 35171-35191. https://doi.org/10.1007/s11042-020-09841-2

[12] Geng, X., Ji, S., Lu, M., & Zhao, L. (2021). Multi-Scale Attentive Aggregation for LiDAR Point Cloud Segmentation. Remote Sensing, 13(4), 691. https://doi.org/10.3390/rs13040691

[13] Luo, N., Jiang, Y., & Wang, Q. (2021). Supervoxel-Based Region Growing Segmentation for Point Cloud Data. International Journal of Pattern Recognition and Artificial Intelligence, 35(03), 2154007. https://doi.org/10.1142/s0218001421540070

[14] Zhang, J., Hu, X., & Dai, H. (2020). A Graph-Voxel Joint Convolution Neural Network for ALS Point Cloud Segmentation. IEEE Access, 8, 139781-139791. https://doi.org/10.1109/access.2020.3013293

[15] Kuang, H., Wang, B., An, J., Zhang, M., & Zhang, Z. (2020). Voxel-FPN: Multi-Scale Voxel Feature Aggregation for 3D Object Detection from LIDAR Point Clouds. Sensors, 20(3), 704. https://doi.org/10.3390/s20030704

[16] Jalali, S. M. J., Ahmadian, S., Khosravi, A., Mirjalili, S., Mahmoudi, M. R., & Nahavandi, S. (2020). Neuroevolution-based autonomous robot navigation: A comparative study. Cognitive Systems Research, 62, 35-43. https://doi.org/10.1016/j.cogsys.2020.04.001

[17] Nizar, I., & Mestari, M. (2022). Mobile Robot Autonomous Navigation: A Path Planning Approach. IFAC-PapersOnLine, 55(12), 610-615. https://doi.org/10.1016/j.ifacol.2022.07.379

[18] Dulce-Galindo, J. A., Santos, M. A., Raffo, G. V., & Pena, P. N. (2022). Distributed supervisory control for multiple robot autonomous navigation performing single-robot tasks. Mechatronics, 86, 102848. https://doi.org/10.1016/j.mechatronics.2022.102848

[19] Krichen, N., Masmoudi, M. S., & Derbel, N. (2020). Autonomous omnidirectional mobile robot navigation based on hierarchical fuzzy systems. Engineering Computations, 38(2), 989-1023. https://doi.org/10.1108/ec-08-2019-0380

[20] Abbasi, A., MahmoudZadeh, S., Yazdani, A., & Moshayedi, A. J. (2021). Feasibility assessment of Kian-I mobile robot for autonomous navigation. Neural Computing and Applications, 34(2), 1199-1218. https://doi.org/10.1007/s00521-021-06428-2

[21] Mackay, A. K., Riazuelo, L., & Montano, L. (2022). RL-DOVS: Reinforcement Learning for Autonomous Robot Navigation in Dynamic Environments. Sensors, 22(10), 3847. https://doi.org/10.3390/s22103847

[22] Xiao, X., Xu, Z., Wang, Z., Song, Y., Warnell, G., Stone, P., Zhang, T., Ravi, S., Wang, G., Karnan, H., Biswas, J., Mohammad, N., Bramblett, L., Peddi, R., Bezzo, N., Xie, Z., & Dames, P. (2022). Autonomous Ground Navigation in Highly Constrained Spaces: Lessons Learned From the Benchmark Autonomous Robot Navigation Challenge at ICRA 2022 [Competitions]. IEEE Robotics & Automation Magazine, 29(4), 148-156. https://doi.org/10.1109/mra.2022.3213466

[23] Malviya, V., Reddy, A. K., & Kala, R. (2020). Autonomous Social Robot Navigation using a Behavioral Finite State Social Machine. Robotica, 38(12), 2266-2289. https://doi.org/10.1017/s0263574720000259

[24] Han, J. H., & Kim, H. W. (2021). Lane Detection Algorithm Using LRF for Autonomous Navigation of Mobile Robot. Applied Sciences, 11(13), 6229. https://doi.org/10.3390/app11136229

[25] Nadour, M., & Cherroun, L. (2022). Using flood-fill algorithms for an autonomous mobile robot maze navigation. International Journal of System Assurance Engineering and Management, 13(1), 546-555. https://doi.org/10.1007/s13198-022-01630-4

[26] Doukhi, O., & Lee, D. J. (2022). Deep Reinforcement Learning for Autonomous Map-Less Navigation of a Flying Robot. IEEE Access, 10, 82964-82976. https://doi.org/10.1109/access.2022.3162702

[27] Ejaz, M. M., Tang, T. B., & Lu, C. K. (2020). Vision-Based Autonomous Navigation Approach for a Tracked Robot Using Deep Reinforcement Learning. IEEE Sensors Journal, 21(2), 2230-2240. https://doi.org/10.1109/jsen.2020.3016299

[28] Chen, M. Y., Wu, Y. J., & He, H. (2021). A novel navigation system for an autonomous mobile robot in an uncertain environment. Robotica, 40(3), 421-446. https://doi.org/10.1017/s0263574721000497

[29] Jia, Z., Liu, H., Zheng, H., Fan, S., & Liu, Z. (2022). An Intelligent Inspection Robot for Underground Cable Trenches Based on Adaptive 2D-SLAM. Machines, 10(11), 1011. https://doi.org/10.3390/machines10111011

[30] Lin, H., Wu, S., Chen, Y., Li, W., Luo, Z., Guo, Y., Wang, C., & Li, J. (2021). Semantic segmentation of 3D indoor LiDAR point clouds through feature pyramid architecture search. ISPRS Journal of Photogrammetry and Remote Sensing, 177, 279-290. https://doi.org/10.1016/j.isprsjprs.2021.05.009

[31] Wang, S., Zhu, J., & Zhang, R. (2022). Meta-RangeSeg: LiDAR Sequence Semantic Segmentation Using Multiple Feature Aggregation. IEEE Robotics and Automation Letters, 7(4), 9739-9746. https://doi.org/10.1109/lra.2022.3191040

[32] Chen, T. H., & Chang, T. S. (2022). RangeSeg: Range-Aware Real Time Segmentation of 3D LiDAR Point Clouds. IEEE Transactions on Intelligent Vehicles, 7(1), 93-101. https://doi.org/10.1109/tiv.2021.3085827

[33] Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Gall, J., & Stachniss, C. (2021). Towards 3D LiDAR-based semantic scene understanding of 3D point cloud sequences: The SemanticKITTI Dataset. The International Journal of Robotics Research, 40(8-9), 959-967. https://doi.org/10.1177/02783649211006735

[34] Guo, M. H., Cai, J. X., Liu, Z. N., Mu, T. J., Martin, R. R., & Hu, S. M. (2021). PCT: Point cloud transformer. Computational Visual Media, 7(2), 187-199. https://doi.org/10.1007/s41095-021-0229-5

[35] Engel, N., Belagiannis, V., & Dietmayer, K. (2021). Point transformer. IEEE Access, 9, 134826-134840. https://doi.org/10.1109/ACCESS.2021.3116304