





Article

# BiLSTM Guided LPA Planning, Re-Planning, and Backtracking for Effective and Efficient Emergency Evacuation

Ramzi Djemai <sup>1,\*</sup>, Hamza Kheddar <sup>2</sup>, Mohamed Chahine Ghanem <sup>1,\*</sup>, Karim Ouazzane <sup>1</sup>  
and Erivelton Nepomuceno <sup>3</sup>

<sup>1</sup> School of Computing and Digital Media, London Metropolitan University, London N7 8DB, UK

<sup>2</sup> LSEA Laboratory, Department of Electrical Engineering, University of Medea, Medea 26000, Algeria

<sup>3</sup> Department of Electronic Engineering, Maynooth University, W23 W6R7 Maynooth, Ireland

\* Correspondence: r.djemai@londonmet.ac.uk (R.D.); m.ghanem@londonmet.ac.uk (M.C.G.)

## Highlights

### What are the main findings?

- A neuro-symbolic emergency evacuation framework was developed by integrating ontology-driven semantic reasoning with Lifelong Planning A\* (LPA\*) and BiLSTM-based temporal guidance.
- In dynamic multi-floor evacuation scenarios, the proposed BiLSTM-guided semantic LPA\* reduced average evacuation time by up to 9.6%, decreased node expansions by up to 32%, and increased the evacuation success rate to 96.2%.

### What are the implications of the main findings?

- The framework supports explainable, policy-compliant, and incrementally adaptable evacuation guidance under evolving hazards and blocked passages.
- Combining semantic reasoning with temporal learning can improve the efficiency, robustness, and practical applicability of intelligent evacuation support systems in complex building environments.

## Abstract

Emergency evacuation in complex and dynamic building environments requires robust and adaptive routing strategies capable of responding to evolving hazards, blocked passages, and changing crowd behaviour. Most existing evacuation planners rely on static geometric representations and lack semantic awareness of the environment, limiting their ability to perform informed re-planning and backtracking when routes become unsafe. This paper proposes a neuro-symbolic evacuation planning framework that integrates Lifelong Planning A\* (LPA\*) with ontology-driven semantic reasoning and a Bidirectional Long Short-Term Memory (BiLSTM) prediction model. The building's spatial and semantic knowledge is represented using the Web Ontology Language (OWL) and Resource Description Framework (RDF), enabling automated inference of implicit connections and enforcement of safety policies. The BiLSTM model learns temporal patterns from ontology-consistent evacuation trajectories and provides guidance for remaining-cost estimation and early prediction of routes likely to require backtracking, which is combined with a bounded semantic heuristic to preserve admissibility and optimality guarantees. Simulation results in a multi-floor academic building show that the proposed BiLSTM-guided semantic LPA\* framework reduces average evacuation time by up to 9.6%, decreases node expansions by up to 32%, and increases evacuation success rates to 96.2% compared with a purely semantic baseline. The BiLSTM model also achieves strong predictive performance, with a test AUC of 0.92 for backtracking prediction and a next-state accuracy of 87.1%. The pro-



Academic Editors: Hao Sheng and Pierluigi Siano

Received: 7 February 2026

Revised: 1 April 2026

Accepted: 3 April 2026

Published: 7 April 2026

**Copyright:** © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

posed framework is designed to support explainable, policy-compliant, and incrementally adaptable evacuation guidance under rapidly evolving emergency conditions.

**Keywords:** indoor emergency evacuation management; ontological modelling; planning heuristics; event-driven re-planning; LSTM; Lifelong Planning A\*

---

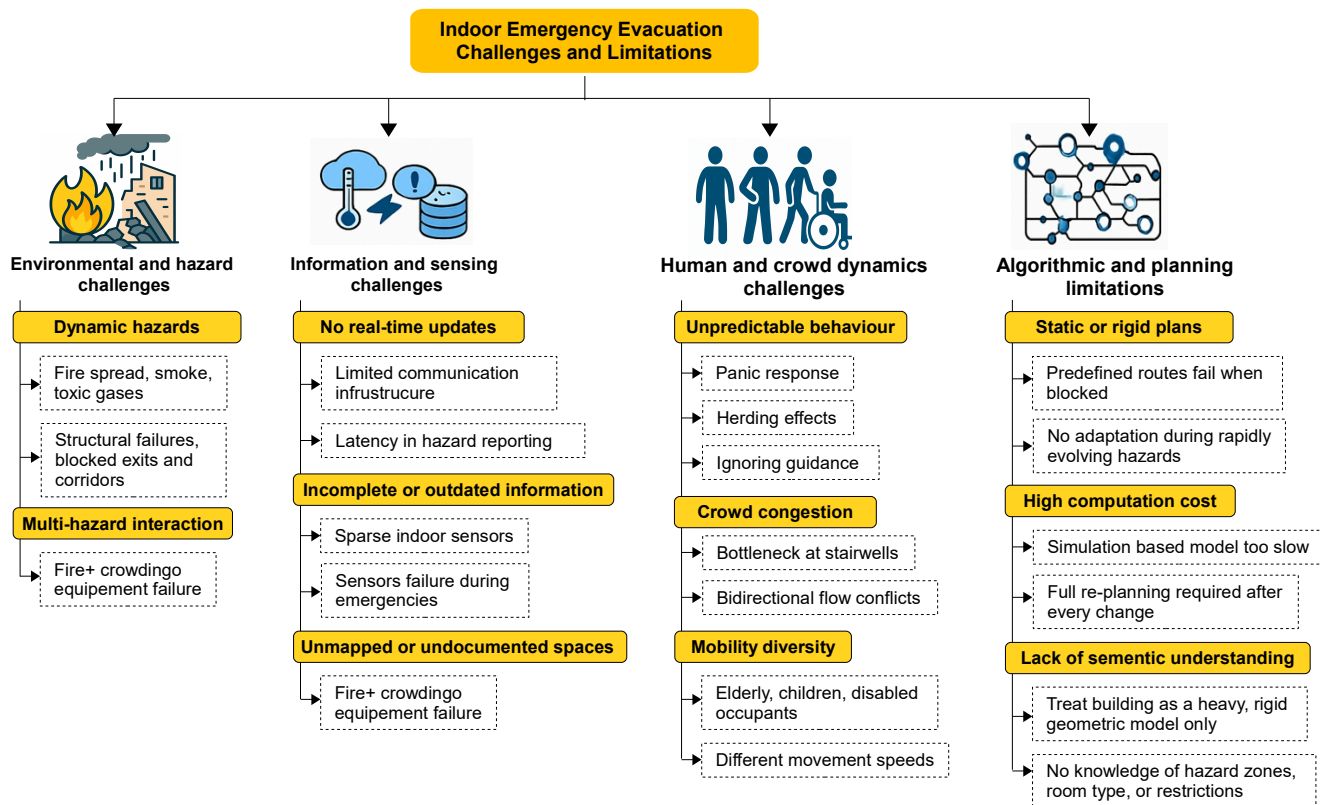
## 1. Introduction

Indoor emergency evacuation presents a particularly complex dynamic decision-making problem. During a crisis, occupants must be guided from arbitrary locations to safe exits while hazards evolve rapidly and unpredictably, leveraging systems such as smart-building digital twins, real-time indoor positioning platforms, and intelligent sensor networks [1]. As illustrated in Figure 1, four major categories of challenges shape this problem: environmental hazards, information and sensing uncertainties, human and crowd dynamics, and algorithmic limitations. Environmental conditions can degrade within seconds: fires spread, smoke reduces visibility, blocks corridors, and structural failures render predefined routes unusable. Traditional evacuation approaches that rely on static floor plans or fixed procedures often become ineffective under such circumstances, resulting in costly re-planning or unsafe routing. A second challenge arises from incomplete and unreliable situational information. Indoor sensors may be sparse, malfunction during a disaster, or lack the communication bandwidth needed to provide timely updates on hazard evolution or exit accessibility. Evacuation algorithms must therefore operate under partial knowledge and adapt as new information becomes available; for instance, a corridor may become blocked or a fire alarm may signal a newly hazardous zone. Evacuation complexity is further amplified by human and crowd behaviour. Large groups tend to move unpredictably when stressed, forming congestion at bottlenecks, reversing direction abruptly, or deviating from suggested paths. These behavioural dynamics introduce significant uncertainty, complicating route prediction and degrading the effectiveness of rigid evacuation plans.

Figure 1 maps the broader challenge space that motivates the research programme. The present work directly addresses the environmental-hazard and algorithmic-limitation categories by modelling hazard events as discrete ontology updates and providing adaptive heuristic search with semantic backtracking. The sensing-uncertainty and crowd-dynamics categories are acknowledged as important open challenges but are not fully addressed by the current implementation; the framework assumes reliable ontology updates (Section 4.3) and plans independently per occupant.

Existing computational methods address these challenges only partially. Rule-based systems encode fixed procedures but rarely adapt once conditions change. Simulation-based models such as computational fluid dynamics or agent-based simulations offer rich predictive power but are too computationally intensive for real-time use. Graph-based algorithms and linear programming methods compute optimal paths efficiently but typically assume static environments and lack semantic understanding of building spaces. More recently, machine learning (ML) approaches have been explored as promising complementary tools in evacuation-related studies. For instance, in the broader domain of disaster evacuation and emergency response planning, ML has been applied to improve adaptive and data-driven decision-making in uncertain and dynamic environments [2]. Similarly, in pedestrian dynamics and evacuation planning, ML has been investigated for capturing pedestrian trajectories, collective crowd behavior, and evacuation-related indicators such as density, flow, and evacuation performance in the built environment [3].

Yet, they require large-scale annotated data and often generalize poorly to new or rapidly evolving scenarios unless continuously retrained.



**Figure 1.** Key challenges in indoor emergency evacuation, including dynamic hazards, incomplete or unreliable sensing, human and crowd behavioural uncertainty, and limitations of existing planning algorithms.

A fundamental limitation across these methods is the absence of explicit semantic knowledge describing the structure and constraints of the building environment. Treating the space as a simple graph of nodes and edges ignores essential contextual information, such as whether a location is a room or corridor, whether an area is restricted, or whether a region is likely to become hazardous. Without semantic context, planners may inadvertently choose routes that are unsafe or infeasible, such as directing evacuees into restricted zones or blocked stairwells misidentified as viable routes.

Building on recent insights in intelligent evacuation planning, this paper proposes a hybrid framework that integrates the computational efficiency of heuristic search with the adaptive intelligence of ontology-based reasoning. At its core, the proposed system employs lifelong planning A\* (LPA\*), an incremental variant of A\*, to achieve fast and dynamic route planning by reusing prior search results as hazards evolve. A semantic reasoning module then continuously evaluates the ontology-informed state of the environment, detects when hazards invalidate the current plan, and triggers real-time adjustments or semantic backtracking to maintain safety and policy compliance. To support this adaptive behaviour, the proposed system leverages advances in semantic technologies, particularly those enabled by formal knowledge representation. Ontology engineering provides a structured, machine-interpretable model of the building environment, including rooms, corridors, stairwells, exits, and hazards, together with the relationships and constraints that govern their safe use during emergencies. By encoding this domain knowledge in both the Web Ontology Language (OWL) and the Resource Description Framework (RDF), the system benefits from automated reasoning that can infer implicit spatial connections,

detect policy violations, and enhance the planner's situational awareness. Unlike black-box ML approaches, the ontology-driven layer in the proposed system offers transparency, semantic richness, and explainability. It can justify its decisions (for example, excluding a route because it passes through a restricted or unsafe area), thereby improving trustworthiness and supporting accountable, safety-compliant evacuation management. In summary, this paper makes the following original contributions:

- **Ontology-Driven Building Model:** We propose a formal OWL/RDF-based ontology to represent both the spatial structure and semantic constraints of a multi-floor building environment, including rooms, corridors, staircases, exits, hazards, and evacuation policies. This knowledge layer enables automated reasoning over implicit connections and safety rules during evacuation planning.
- **Reasoning-Enhanced Heuristic for LPA\*:** We introduce a semantic heuristic function that exploits ontology-derived information to guide the LPA\* algorithm. By integrating inferred paths, vertical navigation constraints, and policy-compliant transitions, the planner achieves efficient incremental re-planning without full recomputation.
- **BiLSTM-Guided Neuro-Symbolic Planning:** We develop a neuro-symbolic extension of semantic LPA\* by incorporating a Bidirectional Long Short-Term Memory (BiLSTM) model trained on ontology-consistent evacuation trajectories. The BiLSTM predicts remaining evacuation cost and the likelihood of backtracking, providing anticipatory guidance while preserving admissibility through a bounded hybrid heuristic.
- **Dynamic Re-planning and Semantic Backtracking Mechanism:** We design an event-driven re-planning and backtracking strategy that allows the planner to revise unsafe routes in response to evolving hazards (e.g., blocked corridors or structural failures) and to safely return to the last feasible semantic situation when necessary.
- **Explainable and Policy-Compliant Decision Making:** Unlike black-box learning approaches, the proposed framework provides explainable evacuation decisions grounded in ontology reasoning and SWRL rules, ensuring that generated routes comply with safety policies and access constraints.
- **Realistic Multi-Floor Case Study:** We provide a detailed case study conducted in a realistic multi-floor academic building to demonstrate the applicability of the proposed framework under dynamic evacuation scenarios.

The remainder of this paper is organised as follows. Section 2 reviews related work and identifies the research gaps addressed in this study. Section 3 presents the theoretical background, including Long Short-Term Memory (LSTM) and BiLSTM models for temporal prediction and backtracking in evacuation planning. Section 4 details the proposed neuro-symbolic framework, including the ontology-driven building model, semantic reasoning layer, reasoning-enhanced LPA\* heuristic, BiLSTM-guided hybrid heuristic, and dynamic re-planning and backtracking mechanisms. Section 5 reports the experimental setup and quantitative results, including ablation studies comparing semantic LPA\* and BiLSTM-guided variants. Finally, Section 6 concludes this paper and outlines limitations and future research directions.

## 2. Related Work

Several studies have been proposed in the literature to address emergency evacuation planning, spanning classical optimisation models, heuristic algorithms, and simulation-based approaches. Existing contributions differ significantly in terms of spatial scale, uncertainty modelling, and the extent to which risk, routing, and resource coordination are jointly considered. To contextualise the proposed approach, prior works are reviewed below according to the scale of the evacuation environment.

- **Outdoor and City-Scale Evacuation Planning:** Large-scale evacuation planning has traditionally focused on network-level optimisation under uncertainty. Goerigk and Grün [4] formulated a robust bus evacuation problem that explicitly accounts for delayed information on evacuee demand, demonstrating how robust optimisation can mitigate uncertainty in urban-scale evacuations involving transit-dependent populations. While effective at the scheduling level, this model does not explicitly incorporate spatially heterogeneous risk or route-level exposure.

To address congestion and route diversity, Chang et al. [5] studied the problem of identifying  $k$  discriminative evacuation paths on road networks, aiming to minimise both travel distance and path overlap. Their ant colony optimisation-based heuristic enables congestion mitigation through path diversification, which is particularly relevant for emergency evacuation and rescue logistics. However, environmental hazards and spatial risk intensity are not explicitly modelled [6].

More recent studies have integrated environmental dynamics into city-scale evacuation planning. Li et al. [7] proposed a coupled hydrodynamic and cellular automata (CA)-based framework for pedestrian evacuation under flooding scenarios, explicitly accounting for flood depth, flow velocity, and human instability. Similarly, Shao et al. [8] emphasised exposure-based evacuation modelling, highlighting the importance of spatially varying hazard intensity in flood-prone urban regions.

Uncertainty-aware urban evacuation routing has also been investigated using heuristic optimisation techniques. Mao et al. [9] developed a fuzzy integer programming model combined with an improved ant colony optimisation algorithm to minimise expected evacuation time in uncertain traffic environments, addressing both symmetric and asymmetric evacuation demands. Although computationally efficient, the approach remains primarily route-centric and does not consider coordinated decisions on shelters or relief distribution. More recently, Jafarian et al. [10] and Li et al. [11] investigated large-scale evacuation from complementary perspectives. Jafarian et al. [10] focused on the impact of behavioural factors on evacuation efficiency, analysing how individual compliance, response delays, and behavioural heterogeneity influence route choice and overall evacuation performance under emergency conditions. Their work enhances behavioural realism but assumes a fixed evacuation network and does not explicitly optimise risk distribution across geographical zones. In contrast, Li et al. [11] adopted a collaborative evacuation framework in which multiple stakeholders or decision-making entities coordinate evacuation actions to improve system-level efficiency. While this approach captures inter-agent coordination and information sharing, geographical risk is still embedded indirectly through network parameters rather than being modelled as an explicit optimisation criterion. Consequently, neither approach directly addresses the problem of balancing spatially heterogeneous risk across the evacuation network.

Recent evacuation planning and emergency management studies increasingly incorporated ML techniques to enhance prediction, support decision-making, and improve response efficiency under hazardous conditions. In this context, [12] combined lexicographically quickest flow-generated optimal plans with a convolutional neural network (CNN), selected for spatial pattern extraction, to rapidly predict evacuation completion time from movement representations in Osaka's tsunami/earthquake urban network, replacing hours of network-flow optimization with near-instant inference during emergencies for citizens. The work in [13] proposed an interpretable enhanced logistic regression model, chosen for interpretable decision modelling, where low-depth decision trees detected nonlinear thresholds and interactions, to predict household hurricane evacuation decisions using demographic and resource variables from Katrina-Rita survey data in coastal U.S. communities. More recently, Ref. [14] developed an artificial neural net-

works (ANN)-based cross-domain framework, adopted for nonlinear spatial mapping, trained with San Francisco seismic and urban features to identify emergency evacuation centers in Tehran, then integrated OpenStreetMap and OpenRouteService routing algorithms for neighborhood-scale navigation toward the nearest selected shelter during earthquakes. Despite their promising results, the CNN was effective for extracting spatial or encoded movement patterns in outdoor and city-scale evacuation planning, but it was less capable of modelling long-range temporal evolution in large-scale evacuation flows. Logistic regression provided interpretability, yet it mainly depended on predefined static variables and could not adequately capture order-sensitive behavioral changes during evacuation progression. Similarly, the feedforward ANN was useful for nonlinear mapping in site selection tasks, but it did not inherently learn temporal dependencies between past and future evacuation states. Consequently, compared with BiLSTM, these approaches were less suited to representing dynamic crowd behavior, route adaptation, and time-dependent trajectory evolution in outdoor urban evacuation scenarios.

- **Single-Floor Building Evacuations and Simulations:** At the building scale, evacuation research has largely relied on simulation-based and heuristic path-planning approaches. Wang et al. [15] proposed a cellular ant optimisation model for passenger ship evacuation, combining CA with ant colony optimisation on hexagonal grids to accelerate search and reduce the likelihood of local optima. While effective in confined environments, the method assumes static hazards and does not explicitly account for dynamically evolving risk. Infrastructure-assisted evacuation has also been investigated. Zhang et al. [16] addressed evacuation efficiency through the optimal placement of signage systems in public spaces, modelling pedestrian–signage interactions using a cooperative location framework. Although this improves wayfinding and guidance, the approach focuses on behavioural support rather than adaptive evacuation routing under emergency conditions. More recently, Baglioni and Jamshidnejad [17] introduced a model predictive control framework for indoor search-and-rescue robots, integrating target-oriented and coverage-oriented objectives under uncertainty. While their approach demonstrates strong performance in dynamic indoor environments, it is primarily designed for robotic exploration and does not directly address large-scale human evacuation or network-level risk balancing. Prior work examined artificial intelligence (AI)-driven methods for evacuation support applications. Wang et al. [18] applied ML techniques, namely PCA, DT, support vector machine (SVM), k-nearest neighbors (KNN), and ANN, to quasi-emergency pedestrian videos in multi-exit indoor layouts, modelling stepwise movement rather than full path planning. The environment was controlled, video-based, and behaviorally simplified, which limited its operational realism. This work could support evacuation by predicting local pedestrian movement near exits. More recently, Hassan et al. [19] employed a deep learning (DL) technique, specifically faster region-CNN with colour filtering, to separate wall information from other plan elements before later processing, along with clustering to digitize German emergency floor plans for building information modelling-enhanced evacuation support. The environment was static, two-dimensional, and plan-centric, enabling map extraction but not dynamic crowd-routing decisions directly. This work could support evacuation by extracting navigable layouts from emergency plans. However, both works remained upstream and partial. Specifically, Wang et al. [18] did not perform sequence-aware route generation, global search, or adaptive path optimization under evolving hazards. In addition, Hassan et al. [19] improved map and symbol extraction, yet it also stopped short of temporal crowd prediction, congestion-aware re-routing, and end-to-end evacuation decision support.
- **Multi-Floor and Complex Building Evacuation:** Evacuation in complex and multi-level structures introduces additional challenges related to vertical movement, smoke

propagation, and inter-floor connectivity. Li and Huang [20] proposed a 3D geographic information system (GIS)-based evacuation framework for large public buildings that integrates numerical fire simulation, smoke diffusion, and individual behaviour modelling with A\* path planning. Their approach enables risk-aware routing across multiple floors, but relies heavily on simulation outputs and lacks an explicit optimisation mechanism for balancing risk across the evacuation network.

In specialised underground environments, Zheng et al. [21] investigated coordinated rescue-path planning for multiple mine rescue teams using a hybrid FA-MDPSO algorithm and force-directed graph layouts. Their work highlights the importance of multi-team coordination and network visualisation in constrained environments. However, the proposed framework is domain-specific and does not readily generalise to urban evacuation networks involving heterogeneous geographical risk and multiple decision layers.

In the context of ML-driven indoor evacuation, Yoo et al. in [22] presented an augmented reality-based internet-of-things evacuation system for complex multi-floor indoor buildings, using a received signal strength indication-based deep neural network (DNN) for indoor user localization, a structural component-based ML, specifically an gradient boosting machine (GBM) ensemble, for disaster propagation prediction, and reinforcement learning for evacuation route planning. Compared with BiLSTM-guided heuristic search, the scheme had several limitations, including beacon dependence, coarse discretization, weaker temporal learning, and slower dynamic adaptation.

Across the surveyed literature, most evacuation planners rely either on optimisation (e.g., mixed-integer linear programming (MILP), fuzzy integer programming (IP)), meta-heuristics (ant colony optimization (ACO), particle swarm optimization (PSO) variants), or agent-based and CA-based simulation. As Table 1 shows, these systems typically operate on raw grid or graph abstractions, with limited or no semantic understanding of the built environment; almost none incorporate ontological reasoning.

**Table 1.** Summary of prior related emergency evacuation studies and the open research gaps they leave. Symbols ✓, ✗, and ≈ mean Yes, No, and Partial, respectively.

| Ref.       | Data Source                                       | NoF | PP Technique                                                          | RP | Indoor | DR | BT | Mass | VN | OR | XAI | RTEA |
|------------|---------------------------------------------------|-----|-----------------------------------------------------------------------|----|--------|----|----|------|----|----|-----|------|
| [4]        | Synthetic + real city case                        | NA  | MILP + robust optimisation                                            | ✗  | ✗      | ≈  | ✗  | ✓    | ✗  | ✗  | ✗   | ≈    |
| [5]        | Real road networks                                | NA  | ACO-based multi-objective routing                                     | ✗  | ✗      | ✓  | ✗  | ✓    | ✗  | ✗  | ✗   | ✗    |
| [7]        | Hydrodynamic simulation + GIS                     | NA  | CA + heuristic shortest path                                          | ✗  | ✗      | ✓  | ✗  | ✓    | ✗  | ✗  | ✗   | ≈    |
| [9]        | Synthetic + benchmark networks                    | NA  | Fuzzy IP + improved ACO                                               | ✗  | ✗      | ✓  | ✗  | ✓    | ✗  | ✗  | ✗   | ≈    |
| [10]       | Behavioural simulation data                       | NA  | Behaviour-aware routing model                                         | ✗  | ✗      | ≈  | ✗  | ✓    | ✗  | ✗  | ✗   | ✗    |
| [11]       | Multi-agent simulated networks                    | NA  | Collaborative optimisation framework                                  | ≈  | ✗      | ✓  | ✗  | ✓    | ✗  | ✗  | ✗   | ≈    |
| [12]       | Yodogawa area of Osaka city                       | NA  | CNN-assisted evacuation planning                                      | ✗  | ✗      | ✗  | ✗  | ✓    | ✗  | ✗  | ✗   | ≈    |
| [13]       | Post-hurricanes Katrina and Rita survey           | NA  | Enhanced logistic regression with decision trees                      | ✗  | ✗      | ✗  | ✗  | ✓    | ✗  | ✗  | ✗   | ✗    |
| [14]       | San Francisco city seismic risks                  | NA  | ANN-based site selection with shortest-path navigation                | ✗  | ✗      | ✓  | ✗  | ✓    | ✗  | ✗  | ✗   | ✗    |
| [15]       | Ship layout (grid-based)                          | 1   | Cellular automata + ACO                                               | ✗  | ✓      | ✗  | ✗  | ✓    | ✗  | ✗  | ✗   | ✗    |
| [16]       | Public hall experiments                           | 1   | Signage location optimisation                                         | ✗  | ✓      | ✗  | ✗  | ✓    | ✗  | ✗  | ✗   | ✗    |
| [17]       | Indoor robotic simulations                        | 1   | MPC-based planning                                                    | ✓  | ✓      | ✓  | ≈  | ✗    | ✗  | ✗  | ✗   | ✓    |
| [18]       | Two quasi-emergency evacuation experiment videos. | 1   | ML-based stepwise movement prediction                                 | ✗  | ✓      | ✗  | ✗  | ✗    | ✗  | ✗  | ✗   | ✗    |
| [19]       | German emergency floor plans                      | 1   | Faster region-CNN digitized evacuation                                | ✗  | ✓      | ✗  | ✗  | ✗    | ✗  | ✗  | ✗   | ✗    |
| [20]       | 3D GIS + fire simulation                          | 8   | A* + fire dynamics                                                    | ✗  | ✓      | ✓  | ✗  | ✓    | ✓  | ✗  | ✗   | ≈    |
| [21]       | Mine network simulation                           | NA  | FA-MDPSO + graph layout                                               | ✗  | ✓      | ✓  | ✗  | ✗    | ✓  | ✗  | ✗   | ✗    |
| [22]       | Collected from group of connected sensors         | 3   | ML-based indoor augmented reality navigation and emergency evacuation | ✓  | ✓      | ✓  | ✗  | ✗    | ✓  | ✗  | ✗   | ✓    |
| <b>Our</b> | Real building ontology                            | 11  | Neuro-Symbolic Semantic LPA* (BiLSTM-Guided)                          | ✓  | ✓      | ✓  | ✓  | ✓    | ✓  | ✓  | ✓   | ✓    |

**Abbreviations:** Number of floors (NoF); Path planning (PP); Re-planning (RP); Dynamic routing (DR); Backtracking (BT); Mass evacuation (Mass); Vertical navigation (VN); Ontological reasoning (OR); Explainability (XAI); Real-time event adaptation (RTEA).

A distinct subset of the literature can be grouped as ML-based methods. However, even in these studies, learning was typically used for isolated sub-tasks rather than semantically grounded, end-to-end indoor evacuation reasoning. For instance, Ref. [12] adopted a CNN-assisted framework for urban-scale evacuation planning, Ref. [13] used enhanced logistic regression and decision trees to predict evacuation-related behaviour from post-hurricane survey data, and Ref. [14] combined ANN-based shelter/site selection with shortest-path navigation in a seismic-risk context. Although these works demonstrate the usefulness of ML for prediction and decision support, they remain external to the semantic structure of indoor built environments.

Likewise, key indoor-evacuation requirements, including multi-floor navigation, dynamic re-planning, backtracking, vertical movement, and real-time event adaptation, are only partially supported and often absent. This limitation is also evident in indoor ML-based studies. For example, Ref. [18] focused on stepwise pedestrian movement prediction from quasi-emergency videos in a controlled single-floor setting, without performing full path planning, re-planning, or real-time adaptation. Similarly, Ref. [19] employed Faster region-CNN with colour filtering and clustering to digitise German emergency floor plans, but the method primarily served as a preprocessing and map-extraction pipeline rather than an operational evacuation planner. Even Ref. [22], which provided augmented reality-assisted indoor evacuation support in a three-floor environment with adaptive guidance, did not incorporate ontological reasoning, explainability, or comprehensive backtracking support. Overall, no prior study integrated semantic building knowledge, adaptive multi-floor navigation, and interpretable decision support within a unified framework. This gap motivates the proposed neuro-symbolic approach.

Our work positions itself at this intersection of unresolved issues. Instead of treating buildings as unstructured spaces, this paper introduces a formal building ontology that encodes the semantics of rooms, corridors, exits, constraints, and access policies. This knowledge layer fills the interpretability gap noted in prior work and enables the planner to enforce high-level rules (e.g., avoid hazardous spaces or restricted “Staff Only” areas). Combined with a dynamic heuristic graph search method (LPA\*), our system supports continuous re-planning, multi-floor movement, backtracking, mass evacuation, and real-time adaptation, all of which prior studies either lack or offer only partially. The result is a hybrid, knowledge-aware planner that delivers explainable decisions while maintaining the efficiency of algorithmic search. The present evaluation demonstrates these properties within a single multi-floor building under dynamically injected hazard scenarios; broader validation across different building types and against external planning algorithms remains future work. To the best of our knowledge, this semantic–algorithmic integration is novel in the context of indoor evacuation planning.

### 3. Preliminaries

Effective emergency evacuation requires dynamic planning capabilities that can account for evolving conditions such as congestion, blocked paths, and time-dependent hazards. Sequence learning models, particularly LSTM networks and their bidirectional variant (BiLSTM), provide a powerful framework for capturing temporal dependencies in evacuation-related data, enabling both forward planning and informed backtracking decisions.

#### 3.1. LSTM for Evacuation Planning

In evacuation planning, the objective is to predict future system states (situations), such as crowd density, travel time, or route feasibility, based on historical and real-time observations. LSTM networks are well suited for this task due to their gated memory

structure, which allows the model to retain critical past information while discarding irrelevant dynamics.

At time step  $t$ , given an input vector  $\mathbf{x}_t$  representing evacuation features (e.g., node occupancy, edge capacity, hazard level), the LSTM updates its internal states as Equation (1) [23]:

$$\begin{cases} \mathbf{f}_t = \sigma(\mathbf{W}_f[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_f), \\ \mathbf{i}_t = \sigma(\mathbf{W}_i[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i), \\ \tilde{\mathbf{c}}_t = \tanh(\mathbf{W}_c[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_c), \\ \mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t, \\ \mathbf{o}_t = \sigma(\mathbf{W}_o[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_o), \\ \mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t), \end{cases} \quad (1)$$

where  $\mathbf{h}_t$  encodes a predictive representation of the evolving evacuation state. This forward temporal modelling supports proactive planning by anticipating congestion propagation and estimating future route viability.

### 3.2. BiLSTM for Backtracking and Route Revision

Emergency evacuation often requires revisiting prior decisions when unexpected events occur, such as sudden bottlenecks or hazard escalation. BiLSTM networks enhance decision-making by integrating both past observations and future contextual information within a planning horizon.

A BiLSTM layer consists of a forward LSTM that processes the evacuation sequence in chronological order and a backward LSTM that processes it in reverse order. The forward hidden state  $\vec{\mathbf{h}}_t$  captures planning-oriented dynamics, while the backward hidden state  $\overleftarrow{\mathbf{h}}_t$  supports backtracking by evaluating how future constraints affect earlier routing choices. The combined representation is given by:

$$\mathbf{h}_t = [\vec{\mathbf{h}}_t \parallel \overleftarrow{\mathbf{h}}_t]. \quad (2)$$

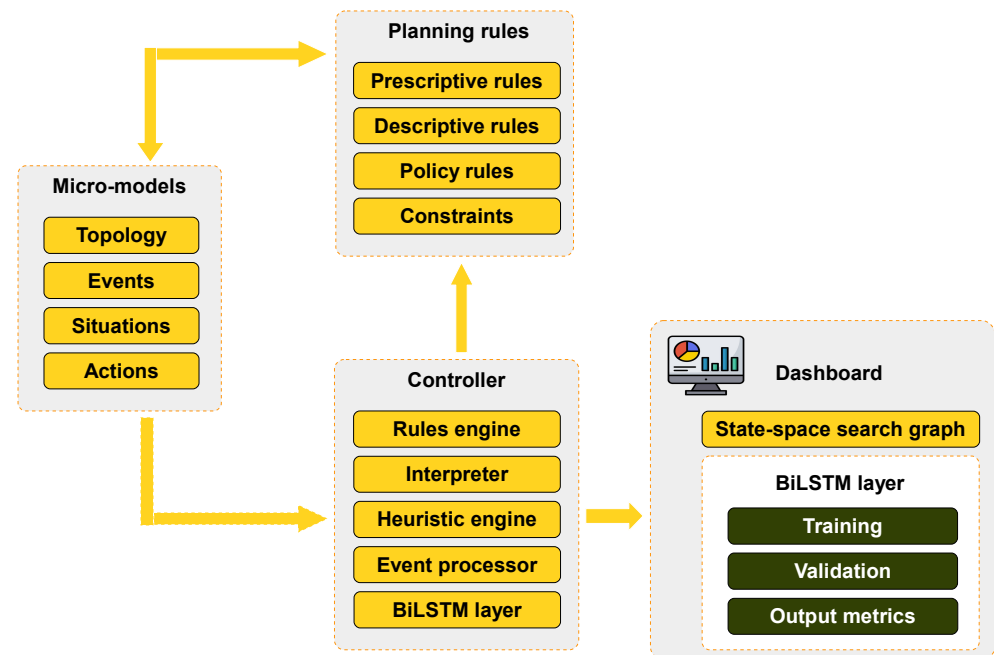
By jointly modelling forward planning and backwards feasibility, BiLSTM layers provide a unified representation that enables effective and efficient evacuation strategies, supporting adaptive rerouting, decision revision, and resilience under uncertainty.

## 4. Research Methodology

This section describes the methodology of the proposed heuristic and knowledge-based evacuation planning system supported by a BiLSTM prediction module. As shown in Figure 2, the architecture combines symbolic semantic reasoning with data-driven learning to support adaptive evacuation planning and is organised around micro-models capturing topology, events, situations (states), and actions, a planning layer encoding prescriptive, descriptive, and policy rules, and a controller responsible for rule interpretation, heuristic evaluation, and event processing. The detailed structure and composition of each component are discussed in subsequent sections.

To make the neuro-symbolic stack explicit, Table 2 summarises what each layer consumes and produces, and how the responsibilities compose at run time. In execution, events are asserted into the ontology, reasoning and SWRL constraints refresh the set of admissible transitions, and the planner searches over semantic situations using an admissible semantic heuristic. If an event changes costs or invalidates a transition, the replanner incrementally repairs the current plan rather than recomputing from scratch; if a transition fails at action execution time, the backtracker rolls back to the most recent feasible semantic situation and triggers replanning from that point. The BiLSTM module provides guidance signals

that bias search towards stable trajectories and more importantly, it does not introduce new actions and does not override SWRL admissibility: it only biases the *ordering* of expansions through the bounded heuristic in Equation (9), while SWRL rules and constraints govern which transitions are permitted.



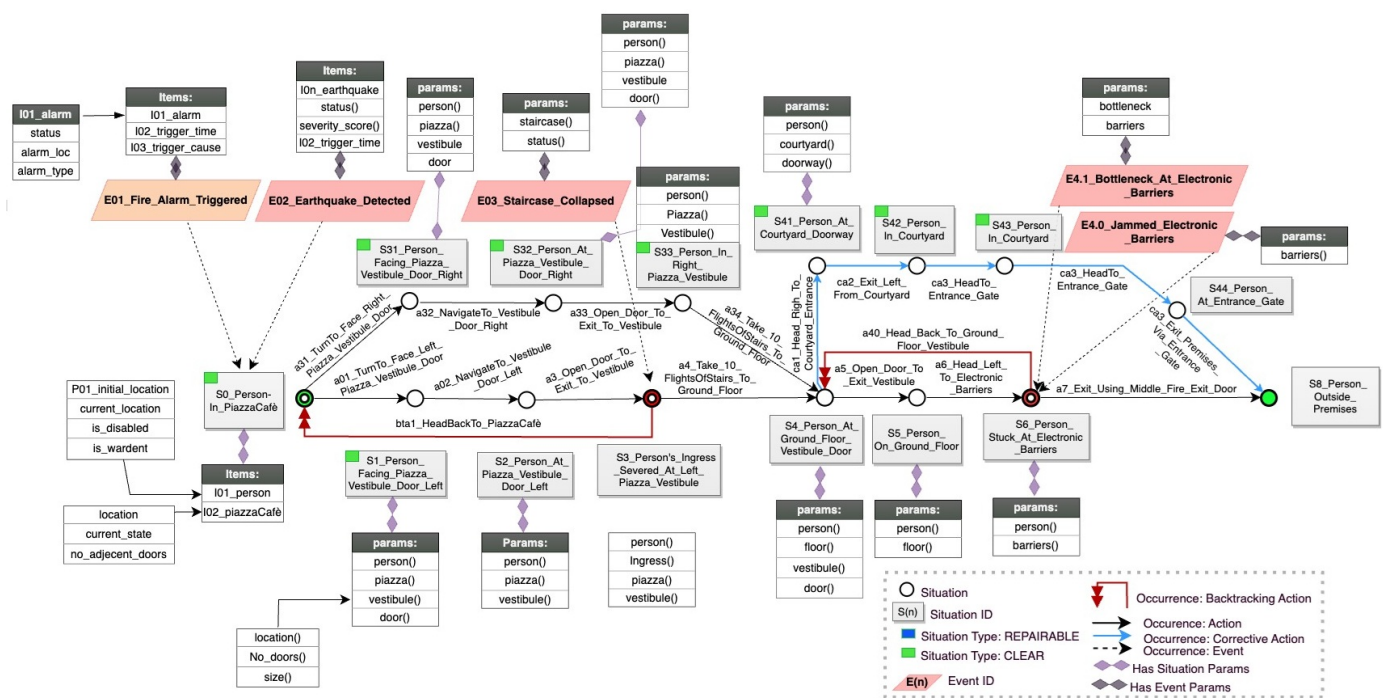
**Figure 2.** Proposed evacuation process of the BiLSTM-guided neuro-symbolic semantic A\* planning framework.

Because evacuation unfolds under randomly triggered events, the planner cannot rely on a static, offline distance map that assumes a single fixed best route. Grid-based distance heuristics (often instantiated as Manhattan distance on rectilinear floor discretisations) are commonly used to approximate corridor traversal in indoor route computation [24,25], but they remain purely geometric and do not encode evacuation policies, restricted areas, or event-driven hazards that can invalidate a route at run time. In our approach, the building state is represented in the ontology, and evacuation behaviour is formulated as policy-driven transitions expressed as (state, action, next-state) triplets. When an event occurs (for example, a fire outbreak [25], congestion build-up [26], or a blocked corridor [25]), the re-planner queries the updated knowledge base to revise the admissible actions and regenerate a feasible action sequence to the goal state. The BiLSTM prediction module complements this symbolic layer by providing temporal guidance on likely congestion evolution from prior trajectories, enabling earlier route revision and backtracking where needed; however, all candidate moves are still filtered by the ontology-defined (state, action, next-state) transitions and the current event assertions, so only semantically admissible actions are expanded. Moreover, admissible moves are instantiated and validated via semantic Web rule language (SWRL) triplets, ensuring that each expansion corresponds to a policy-compliant (state, action, next-state) transition.

Figure 3 summarises three representative evacuation sequences (state–action sequences); corrective actions (CAs) indicate rerouting, and backtracking actions (BTAs) indicate deliberate returns to earlier decision points, while each step is an admissible (state, action, next-state) transition instantiated from the SWRL triplets in Table 3 under the current event assertions.

**Table 2.** Roles and interactions of the main components in the neuro-symbolic evacuation stack.

| Component                                 | Consumes → Produces                                                                                          | Responsibility and Constraint                                                                                                                                                                                                                                              |
|-------------------------------------------|--------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| OWL/RDF ontology (state model)            | Building entities, occupants, hazards, events → semantic situations and facts                                | Defines the semantic state space (situations) and typed entities (Room, Corridor, Exit, Hazard). All planning operates over these semantic representations.                                                                                                                |
| SWRL rules + constraints (planning rules) | Ontology facts + event assertions → admissible (state, action, next-state) transitions and pruned neighbours | Acts as the admissibility gate: transitions that violate safety or access constraints are rejected at expansion and at execution time. Also supports inferred connections and policy-driven constraints.                                                                   |
| Semantic heuristic $h_o(n)$               | Current node + goal + ontology inferences → admissible remaining-cost estimate                               | Guides search using ontology-derived structure (including inferred connectivity and vertical navigation). Must remain admissible to preserve optimality guarantees.                                                                                                        |
| Planner (Semantic LPA*)                   | Admissible neighbours + costs + heuristic → candidate evacuation path                                        | Performs incremental heuristic search on semantic situations. Uses $f(n) = g(n) + h(n)$ to prioritise expansions while respecting admissibility and ontology constraints.                                                                                                  |
| Replanner (event-driven repair)           | New event assertions + updated ontology → repaired plan                                                      | Triggered when events change costs or invalidate transitions. Updates only affected portions and repairs the prior search results (incremental replanning rather than full recomputation).                                                                                 |
| Backtracker (semantic rollback)           | Execution failure or invalidated transition → last feasible situation + replanning trigger                   | When the next action becomes invalid at run time, returns to the most recent safe, semantically valid situation, then restarts planning from that point.                                                                                                                   |
| BiLSTM guidance module                    | History of semantic state-action sequence → predicted remaining cost and backtracking likelihood             | Provides learned guidance to bias the planner away from fragile trajectories and towards stable ones. It does not create actions and does not bypass SWRL admissibility. Its influence must be bounded so that semantic admissibility and policy compliance are preserved. |
| Controller (orchestration loop)           | Events + reasoning + planner outputs → executed guidance and logs                                            | Coordinates event processing, reasoning refresh, planning, action execution, replanning triggers, and backtracking. Acts as the runtime glue between symbolic and learned modules.                                                                                         |



**Figure 3.** Ontology-based scenarios representation for Tower Building over multiple floors.

**Table 3.** Summary of representative ontology-driven evacuation scenarios illustrated in Figure 3.

| Scenario | Key Behaviour                                                                                                                                                                                                                                            | Path (State-Action Sequence)                                                                                                  |
|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------|
| 1        | Single trigger event; no obstruction; no rerouting or backtracking.                                                                                                                                                                                      | S0, A01, S1, A02, S2, A13, S12, A14, S15, A15, S18, A16, S5, A8, S11                                                          |
| 2        | Obstruction (E02) invalidates a previously admissible continuation; rerouting via corrective actions (CA1, CA2); no backtracking. BiLSTM guidance biases branch-off selection while feasibility remains enforced by ontology constraints and SWRL rules. | S0, A01, S1, A02, S2, A3, S3, CA11*, S13, CA2*, S4, A5, S5, A8, S11                                                           |
| 3        | Multiple events; rerouting followed by backtracking due to persistent hazards; explicit return to earlier decision points before recovering a feasible route to the exit.                                                                                | S0, A01, S1, A02, S2, A3, S3, CA1, S13, CA2, S4, A12, S4, BTA1, S13, BTA2, S2, A13, S12, A14, S15, A15, S18, A16, S5, A8, S11 |

#### 4.1. Evacuation Micro-Models

Evacuation micro-models provide the atomic semantic representations required by the proposed planning and learning algorithms. These models define how evacuees, environments, and actions are represented at a fine-grained level, forming the basis for ontology reasoning, heuristic evaluation, and state-space search.

- **Topological Modelling:** The evacuation environment is represented as a directed semantic graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , where each node  $v \in \mathcal{V}$  corresponds to a physical space (e.g., room, corridor, staircase, exit), and each edge  $e \in \mathcal{E}$  represents a traversable connexion. This topological structure enables the retrieval of explicit neighbours during planning, as follows (Equation (3)):

$$\text{QueryNeighbors}(\text{current}, \mathcal{O}). \quad (3)$$

- **Event modelling:** Events represent discrete occurrences that may alter evacuation dynamics, such as door closures, congestion, alarms, or accessibility changes. Events are represented as assertions that update the ontology's state, thereby altering the availability of transitions or the cost during planning. Formally, an event  $e_t$  induces a change in the transition function (Equation (4)):

$$(s_t, a_t) \rightarrow s_{t+1}, \quad (4)$$

which is dynamically reflected in the inferred neighbour set queried by the planner.

- **Situation Representation:** A situation corresponds to a semantic state describing the current evacuation context of an individual. Each situation  $s \in \mathcal{S}$  encapsulates: (i) the evacuee's location, (ii) the structural properties of the space, and (iii) safety and accessibility constraints. Situations serve as nodes in the state-space search graph explored by the LPA\* algorithm.

- **Action Space Definition:** Actions define admissible movements between situations, such as `move_to_corridor`, `enter_staircase`, or `exit_building`. The action space  $\mathcal{A}$  is constrained by ontology rules and encoded using SWRL triplets. Each action induces a cost  $c(s, a, s')$  used during planning (Equation (5)):

$$c = \text{COST}(\text{current}, \text{neighbor}), \quad (5)$$

These costs may reflect distance, vertical movement penalties, or safety-related preferences.

The ontology is designed with a clear separation between its terminological layer (TBox) and its assertional layer (ABox). The TBox defines the generic class hierarchy (Room, Corridor, Staircase, Floor, Exit, Hazard) and the relationships that govern evacuation behaviour; the ABox populates this schema with the individuals, connections, and floor assignments specific to a particular building. The SWRL transition rules are likewise organised into two layers: a generic core that encodes reusable evacuation behaviours (exiting rooms, traversing corridors, descending staircases, reaching exits) and a building-specific extension layer that encodes transitions unique to a particular topology (e.g., link-bridge crossings or building-specific floor identifiers). To adapt the framework to a new building, the generic TBox and the generic SWRL core are retained unchanged; only the ABox must be re-populated with the new building's spatial entities and the extension rules must be replaced to reflect its particular navigation structure. This modularity has been verified by instantiating a structurally distinct hospital building (a 3-storey dual-block layout with restricted zones and multiple exits) that reuses the generic SWRL core without modification, requiring only a building-specific extension layer to capture hospital-specific navigation patterns such as restricted-zone one-way evacuation and multi-staircase vertical descent. The ABox re-population represents the primary manual effort during transfer, though this could be partially automated via BIM/IFC-to-OWL conversion pipelines.

It is important to clarify the scope of the evacuation dynamics captured by the current symbolic representation. The framework models hazard events as discrete ontology updates: a fire alarm activates evacuation mode, a structural incident blocks a corridor segment, and a bottleneck renders a vestibule impassable. Each such event is asserted into the ontology as a typed fact that alters the admissible transition set. This representation is well suited to events that change the binary availability or traversal cost of building segments. However, phenomena that are inherently continuous or spatially distributed, such as progressive smoke propagation, crowd density waves, panic-driven behavioural shifts, and heterogeneous occupant mobility, are not modelled by the current ontology and SWRL layer. The BiLSTM learns from symbolic state–action trajectories generated under these discrete events, not from continuous physical simulations or raw sensor streams. Consequently, the framework's adaptive capabilities should be understood as operating over discrete, event-driven hazard changes within a semantically typed state space, rather than over the full spectrum of physical evacuation dynamics.

#### 4.2. Planning Rules

Planning rules encode domain knowledge that governs admissible evacuation behaviour. These rules are formalised using SWRL and operate over the ontology to define valid transitions, constraints, and preferences during evacuation planning.

- **Prescriptive Rules:** Prescriptive rules define mandatory evacuation actions that must be taken under specific conditions, such as enforcing movement toward exits once an alarm is triggered or preventing access to restricted areas. These rules ensure compliance with evacuation policies and safety regulations.

Formally, prescriptive rules constrain the action space by allowing only transitions that satisfy predefined safety conditions:

$$(s_t, a_t) \rightarrow s_{t+1} \quad \text{if and only if } \mathcal{R}_{\text{pres}}(s_t) = \text{true.}$$

- **Descriptive Rules:** Descriptive rules capture factual relationships within the environment, such as adjacency between spaces, floor connectivity, and accessibility relations. These rules enable the ontology reasoner to infer implicit connections that are not explicitly encoded in the topological graph. Such inferred relations are queried during planning through ontology reasoning, as reflected in Algorithm 1.

---

**Algorithm 1** Neuro-Symbolic Semantic LPA\* (BiLSTM-Guided)
 

---

**Require:** start node  $s$ , goal node  $g$ , ontology  $\mathcal{O}$ , movement history  $\mathcal{H}$

**Ensure:** returns an optimal evacuation path from  $s$  to  $g$ , or NULL if none exists

```

1: openSet  $\leftarrow \{s\}$ ; closedSet  $\leftarrow \emptyset$ 
2: gScore[s]  $\leftarrow 0$ 
3: fScore[s]  $\leftarrow h'(s)$  ▷ Hybrid heuristic per Equation (9)
4: while openSet  $\neq \emptyset$  do
5:   current  $\leftarrow \arg \min_{n \in \text{openSet}} fScore[n]$  ▷ Node with lowest estimated total cost
6:   if current =  $g$  then
7:     return RECONSTRUCTPATH(cameFrom, current)
8:   end if
9:   openSet  $\leftarrow \text{openSet} \setminus \{\text{current}\}$ 
10:  closedSet  $\leftarrow \text{closedSet} \cup \{\text{current}\}$ 
11:  neighbors_explicit  $\leftarrow \text{QUERYNEIGHBORS}(\text{current}, \mathcal{O})$ 
12:  neighbors_inferred  $\leftarrow \text{QUERYINFERREDCONNECTIONS}(\text{current}, \mathcal{O})$ 
13:  neighbors  $\leftarrow \text{neighbors\_explicit} \cup \text{neighbors\_inferred}$ 
14:  for all neighbor  $\in$  neighbors do
15:    if neighbor  $\in$  closedSet then
16:      continue
17:    end if
18:     $c \leftarrow \text{COST}(\text{current}, \text{neighbor})$ 
19:    tentative_g  $\leftarrow gScore[\text{current}] + c$ 
20:    if neighbor  $\notin$  openSet then
21:      openSet  $\leftarrow \text{openSet} \cup \{\text{neighbor}\}$ 
22:    else if tentative_g  $\geq gScore[\text{neighbor}]$  then
23:      continue
24:    end if
25:    cameFrom[neighbor]  $\leftarrow \text{current}$ 
26:    gScore[neighbor]  $\leftarrow \text{tentative\_g}$  ▷ End of Semantic LPA* update
27:     $h_o \leftarrow \text{SEMANTICHEURISTIC}(\text{neighbor}, g, \mathcal{O})$ 
28:     $h_l \leftarrow \text{BiLSTM\_PREDICT}(\mathcal{H} \cup \{\text{neighbor}\})$ 
29:     $h_l \leftarrow \max(0, h_l)$  ▷ Ensure non-negative learned estimate
30:     $h'(\text{neighbor}) \leftarrow \min(h_o, (1 - \lambda)h_o + \lambda h_l)$  ▷ Bounded convex blend
    (Equation (9)); preserves admissibility
31:    fScore[neighbor]  $\leftarrow gScore[\text{neighbor}] + h'(\text{neighbor})$ 
32:  end for
33: end while
34: return NULL

```

---

by the function:

$$\text{QueryInferredConnections}(\text{current}, \mathcal{O}).$$

- **Policy Rules:** Policy rules express evacuation preferences rather than strict constraints. Examples include prioritising safer routes, avoiding congested areas, or favouring ramps over stairs for mobility-impaired evacuees. These rules influence the planning process by affecting heuristic evaluation and cost assignment rather than enforcing hard exclusions.
- **Constraint Encoding:** Constraint rules encode physical and operational limitations, such as blocked passages, closed doors, or capacity restrictions. These constraints dynamically prune the state–transition graph by removing infeasible neighbours during node expansion.

Together, planning rules ensure that all paths explored by the planner are semantically valid, policy-compliant, and context-aware while remaining flexible to dynamic environmental changes.

#### 4.3. Controller

The controller constitutes the operational core of the proposed evacuation framework and integrates the functionalities previously described in the system architecture, namely the ontology and reasoning engine, heuristic path planner, event generator, and action interpreter, into a unified execution loop.

At its foundation, the controller maintains an OWL 2 ontology representing the building layout and evacuation knowledge, including spatial entities (e.g., *Room*, *Corridor*, *Exit*), dynamic entities (e.g., *Person*, *Hazard*), and action concepts (e.g., *EvacuationAction*, *BlockedPath*). Safety policies and access constraints are encoded using axioms and SWRL rules so that inadmissible transitions are filtered during planning.

Dynamic events generated by the simulator (such as smoke propagation or blocked corridors) are translated into ontology assertions. These updates trigger the reasoning process, which refreshes inferred relations (e.g., admissible connectivity or safety classifications) and signals the planner that replanning is required.

- **Rule Engine:** Applies SWRL-based planning rules to validate actions and infer implicit state transitions. Routes violating evacuation policies (e.g., restricted areas or unsafe corridors) are rejected at expansion time.
- **Semantic Interpreter:** Maps ontology assertions and inferred relations into executable planning constructs. Semantic situations correspond to graph nodes, actions to transitions, and inferred relations to neighbour sets used by the planner.
- **Heuristic Engine:** Implements the heuristic path planner using semantic LPA\*. At each expansion step, both explicit neighbours (physical adjacency) and inferred neighbours (reasoner-derived connections) are evaluated using the semantic heuristic  $h_o(n)$  and, when enabled, the learned guidance  $h_l(n)$ :

$$f(n) = g(n) + h(n).$$

- **Event Processing Mechanism:** Produces and processes time-stamped dynamic events (e.g., hazards, blocked corridors, door constraints) and updates the ontology accordingly. These updates modify edge costs and admissibility, which triggers incremental replanning.
- **Action Interpreter:** Translates the planner's sequence of situations and actions into executable evacuation guidance. Each transition is validated against the current ontology state. If a transition becomes invalid due to a new event, semantic backtracking is invoked and replanning resumes from the last feasible situation.

Because semantic events usually affect only a limited subset of edges and constraints, the controller performs incremental repair rather than full recomputation. Only affected vertices and their local neighbourhood are updated and propagated through the priority queue, preserving consistency with updated semantic constraints while maintaining efficiency.

Algorithm 2 formalises this closed-loop control process, including event handling, replanning triggers, and action execution.

A critical component of the control loop is the `TESTERCHECKFAILS` procedure (Algorithm 2, lines 38–42), which detects plan–reality divergence at each execution step. The procedure operates in three stages. First, the expected next situation is obtained from the current plan via `PREDICTOUTCOME`, which, when BiLSTM guidance is enabled, incorporates the model’s anticipatory prediction of the most likely resultant state. Second, the actual post-action situation is retrieved from the ontology via `FETCHACTUALSITUATION`, reflecting the ground-truth state after the most recent event assertions have been processed by the reasoner. Third, the two are compared: if the actual situation differs from the expected one, the procedure returns *true*, triggering `IDENTIFYTRIGGER` to diagnose the cause and `REPLAN` to compute a revised route from the current position. This mechanism enables the framework to detect and respond to plan invalidation caused by dynamic hazard events without waiting for an explicit dead-end. It is important to note that the current evaluation assumes reliable and immediate ontology updates: the simulator translates each dynamic event into ontology assertions without delay or error, so the planner always operates on a correct hazard state at the point of query. Noisy, delayed, or failed sensor input is not modelled in the present work; the implications of relaxing this assumption are discussed in Section 6.

#### 4.4. Neuro-Symbolic Evacuation Planning Framework

This subsection formalises the neuro-symbolic planning methodology that combines ontology-driven semantic reasoning, heuristic graph search, and BiLSTM-based temporal guidance. It operationalises the components described in the controller through three interacting layers: semantic heuristic estimation, incremental planning, and learning-guided refinement.

- **Reasoning-Enhanced Semantic Heuristic:** The semantic heuristic  $h_o(n)$  exploits ontology-derived structure that cannot be captured by purely geometric heuristics. It queries inferred evacuation paths, vertical navigation requirements, and policy-compliant transitions to estimate remaining cost:

$$h_o(n) = \begin{cases} \text{GeometricHeuristic}(n, g), & \text{if no inferred path exists,} \\ (\text{horizontalCost} + \text{verticalCost}) \times \text{qualityFactor}, & \text{otherwise.} \end{cases} \quad (6)$$

When a plausible route can be inferred by the reasoner, the heuristic converts semantic evidence into an optimistic estimate combining horizontal traversal and vertical transitions. The factor  $\text{qualityFactor}(n, g) \in (0, 1]$  acts as a confidence discount so that uncertain inferred routes are down-weighted without violating admissibility. Algorithm 3 specifies this computation.

**Algorithm 2** Dynamic Path Planning with Re-planning and Backtracking**Require:** InitialSituation, OccupantID, BuildingTopology, EventDefinitions, ActionRules**Ensure:** Safe exit notification or reroute guidance

```

1: procedure INITIALIZESYSTEM
2:   current ← FETCHINITIALSITUATION
3:   BUILDTOPOLOGYMODEL(BuildingTopology)
4:   eventDefs ← LOADEVENTDEFINITIONS
5:   actionRules ← LOADACTIONRULES
6:   INITIALIZETESTER
7: end procedure
8: procedure MAINLOOP
9:   INITIALIZESYSTEM
10:  while not EXITREACHED(OccupantID) do
11:    event ← FETCHNEXTEVENT
12:    type ← CLASSIFYEVENT(event, eventDefs)
13:    if type = "Impact" then
14:      HANDLEIMPACTEVENT(current, event)
15:    else
16:      CONTINUENORMALPLAN(current)
17:    end if
18:    if TESTERCHECKFAILS(current) then
19:      trigger ← IDENTIFYTRIGGER(current)
20:      plan' ← REPLAN(trigger, actionRules)
21:      UPDATEGUIDANCE(OccupantID, plan')
22:      current ← UPDATESITUATION(OccupantID, plan')
23:    end if
24:    current ← FETCHNEXTSITUATION
25:  end while
26:  CONFIRMEEXIT(OccupantID)
27: end procedure
28: procedure HANDLEIMPACTEVENT(situation, event)
29:   route ← FINDSAFEPATH(situation, event, actionRules)
30:   RELAYROUTE(OccupantID, route)
31:   LOGEVENTANDROUTE(event, route)
32: end procedure
33: procedure CONTINUENORMALPLAN(situation)
34:   action ← SELECTACTION(situation, actionRules)
35:   EXECUTEACTION(action)
36:   LOGEXECUTION(action, situation)
37: end procedure
38: procedure TESTERCHECKFAILS(situation)
39:   expected ← PREDICTOUTCOME(situation)
40:   actual ← FETCHACTUALSITUATION
41:   return actual ≠ expected
42: end procedure
43: procedure IDENTIFYTRIGGER(situation)
44:   lastEvent ← FETCHLASTEVENT
45:   return ANALYZEIMPACT(lastEvent, situation)
46: end procedure

```

**Algorithm 3** Heuristic Estimation Using Inferred Paths**Require:** current node  $n$ , goal node  $g$ **Ensure:** heuristic estimate  $h_o(n)$  is returned

- 1:  $paths \leftarrow \text{QUERYINFERREDPATHS}(n, g)$   $\triangleright$  Candidate inferred route sequences from  $n$  to  $g$  via ontology reasoning.
- 2:  $h\_geo \leftarrow \text{GEOMETRICHEURISTIC}(n, g)$   $\triangleright$  Admissible geometric fallback (e.g., Manhattan/Euclidean in the discretisation).
- 3: **if**  $paths = \emptyset$  **then**
- 4:     **return**  $h\_geo$
- 5: **end if**
- 6:  $minHops \leftarrow \min\{|p| : p \in paths\}$   $\triangleright$  Length of shortest inferred path in number of hops.
- 7:  $avgCost \leftarrow \text{ESTIMATEAVERAGEEDGE}(n)$   $\triangleright$  Optimistic average traversal cost from this node per edge.
- 8:  $currentFloor \leftarrow \text{GETFLOOR}(n)$
- 9:  $goalFloor \leftarrow \text{GETFLOOR}(g)$
- 10:  $floorDiff \leftarrow |currentFloor - goalFloor|$
- 11:  $verticalCost \leftarrow floorDiff \times \left(\frac{FloorHeight}{ClimbSpeed}\right)$   $\triangleright$  Optimistic time to change  $floorDiff$  floors via stairs.
- 12:  $horizontalCost \leftarrow minHops \times avgCost$
- 13:  $quality \leftarrow \text{ASSESSPATHQUALITY}(paths[0])$   $\triangleright$  Assess inferred path plausibility under current hazards/policies.
- 14:  $qFactor \leftarrow \text{QUALITYFACTOR}(quality)$   $\triangleright$   $qFactor \in (0, 1]$  is a confidence discount: 1 if high-quality, smaller if uncertain.
- 15:  $h\_inf \leftarrow (horizontalCost + verticalCost) \times qFactor$
- 16: **return**  $\max(h\_geo, h\_inf)$   $\triangleright$  Max of admissible heuristics is admissible and more informative.

- **Semantic Lifelong Planning A\*:** Semantic LPA\* integrates the heuristic  $h_o(n)$  into an incremental planning framework. The search space is defined by semantic states derived from the ontology, and neighbour expansion combines explicit topological connections with inferred semantic transitions. For each node  $n$ , the planner evaluates:

$$f(n) = g(n) + h_o(n). \quad (7)$$

When the ontology is updated due to dynamic events (e.g., blocked corridors or modified traversal costs), only the affected nodes and edges are updated, and vertex consistency is restored using the standard LPA\* bookkeeping ( $g$ - and  $rhs$ -values). This enables rapid local repair instead of full recomputation.

- **Neuro-Symbolic Semantic LPA\* (BiLSTM-Guided):** To incorporate temporal experience, a BiLSTM model is trained offline on SWRL-consistent state-action sequences expressed in the same symbolic vocabulary as the planner. At run time, given the movement history  $\mathcal{H}$  and a candidate node  $n$ , the model predicts a guidance term:

$$h_l(n) = \text{BiLSTM}(\mathcal{H} \cup \{n\}). \quad (8)$$

BiLSTM-guided bounded heuristic.

Let  $h_o(n)$  denote the ontology-derived admissible heuristic estimate of the remaining cost to the goal. The BiLSTM outputs a learned remaining-cost estimate  $h_l(n) \geq 0$  from the recent state–action history. We combine both signals using a bounded convex blend:

$$h'(n) = \min\left(h_o(n), (1 - \lambda)h_o(n) + \lambda h_l(n)\right), \quad 0 \leq \lambda \leq 1. \quad (9)$$

By construction  $h'(n) \leq h_o(n)$  for all  $n$ . Since  $h_o$  is admissible,  $h'$  remains admissible, hence the optimality guarantees of semantic LPA\* are preserved. The parameter  $\lambda$  controls the strength of the learned guidance whenever  $h_l(n) < h_o(n)$ ; otherwise the heuristic falls back to  $h_o(n)$ .

The integration proceeds in four steps at each node expansion (Algorithm 1, lines 22–28). First, the semantic heuristic  $h_o$  is computed for the candidate neighbour by querying the ontology for inferred paths, vertical navigation requirements, and policy-compliant transitions (Algorithm 3). Second, the BiLSTM model receives the movement history  $\mathcal{H}$  augmented with the candidate node and returns a learned remaining-cost estimate  $h_l$ . Third,  $h_l$  is clipped to ensure non-negativity, preventing any learned underestimate from producing a negative heuristic. Fourth, the bounded convex blend (Equation (9)) combines  $h_o$  and  $h_l$ , and the outer min operator guarantees that the resulting  $h'$  never exceeds the admissible semantic estimate. Because the BiLSTM influences only the ordering of node expansions through  $h'$  and does not introduce new transitions or override SWRL admissibility constraints, safety and policy compliance remain fully governed by the symbolic layer.

An important architectural property is the independence of the symbolic and learned layers. The ontology, SWRL rules, and semantic heuristic  $h_o$  operate without any dependency on the BiLSTM; conversely, the BiLSTM consumes symbolic state–action sequences but does not modify the ontology or the admissible transition set. The two layers interact only through the bounded blend in Equation (9). This independence enables incremental deployment: setting  $\lambda = 0$  recovers pure semantic LPA\*, which can be deployed immediately on a new building once the ontology is instantiated, before any BiLSTM training data have been generated. Learned guidance can then be introduced incrementally as training trajectories become available, while larger values increase the influence of learned guidance without violating optimality or policy constraints. The evaluation function becomes:

$$f(n) = g(n) + h'(n). \quad (10)$$

Because  $h'(n) \leq h_o(n) \leq h^*(n)$ , admissibility and optimality guarantees of LPA\* are preserved: the learned component can only reduce the heuristic estimate relative to  $h_o$ , never increase it, so the planner never overestimates the true remaining cost and optimal-path completeness is maintained. The planner therefore benefits from temporal awareness, including early detection of routes prone to backtracking, while maintaining explainability and semantic compliance. Algorithm 1 implements semantic LPA\* planning process along with incremental neuro-symbolic extension (BiLSTM-guided).

## 5. Experiments

### 5.1. Evaluation Metrics

The performance of the proposed evacuation planning framework was assessed using three complementary metrics that capture evacuation efficiency, robustness, and computational cost.

1. **Mean Evacuation Time ( $T_{avg}$ ):** This metric measures the average time required for an occupant to reach a safe exit across all simulated scenarios. It reflects the overall efficiency of the evacuation planner under dynamic conditions.

$$T_{avg} = \frac{1}{N} \sum_{i=1}^N T_i \quad (11)$$

where  $T_i$  denotes the evacuation time of the  $i$ -th scenario and  $N$  is the total number of evaluated scenarios. Percentile evacuation times (e.g., 90th percentile) were also computed to assess worst-case performance under adverse conditions.

2. **Evacuation Success Rate ( $S_r$ ):** The success rate quantifies the proportion of evacuation trials in which occupants successfully reached a safe exit within a predefined time limit. This metric evaluates the robustness and reliability of the planner in the presence of hazards, re-planning, and backtracking events.

$$S_r = \frac{N_{success}}{N_{total}} \times 100\% \quad (12)$$

where  $N_{success}$  is the number of scenarios that resulted in a successful evacuation and  $N_{total}$  is the total number of tested scenarios.

3. **Search Effort (Expanded Nodes  $N_{nodes}$ ):** Computational efficiency was evaluated by counting the number of nodes expanded by the planner during the search process. This metric directly reflects the complexity of the planning task and the effectiveness of heuristic guidance in reducing unnecessary exploration.

$$N_{nodes} = \sum_{k=1}^K n_k \quad (13)$$

where  $n_k$  represents the number of nodes expanded during the  $k$ -th evacuation run and  $K$  is the number of simulated scenarios.

4. **Re-planning and Backtracking Frequency:** The number of re-planning events and backtracking actions was recorded to quantify how often the system needed to revise previously selected routes due to environmental changes (e.g., blocked corridors or emerging hazards). Lower values indicate more stable and anticipatory planning behaviour.
5. **Prediction Performance Metrics (BiLSTM):** For the BiLSTM prediction module, the following supervised learning metrics were employed:

- **Root mean square error (RMSE)** for remaining cost prediction:

$$RMSE = \sqrt{\frac{1}{M} \sum_{j=1}^M (\hat{c}_j - c_j)^2} \quad (14)$$

where  $\hat{c}_j$  and  $c_j$  denote the predicted and true remaining costs for sample  $j$ , respectively.

- **Area Under the ROC Curve (AUC)** for backtracking prediction, measuring the discriminative ability of the classifier between backtracking and non-backtracking trajectories.

- **Classification Accuracy (Acc)** for both backtracking prediction and next-state prediction:

$$Acc = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \quad (15)$$

where  $T_P$ ,  $T_N$ ,  $F_P$ , and  $F_N$  denote true positives, true negatives, false positives, and false negatives, respectively.

Each metric is selected to address a specific research question posed by the framework.  $T_{\text{avg}}$  directly evaluates whether BiLSTM-guided heuristic blending reduces evacuation time compared to the semantic baseline, which is the primary efficiency claim of the neuro-symbolic integration.  $S_r$  assesses whether the framework maintains or improves evacuation reliability under dynamic hazard conditions, a requirement for any planner intended for safety-critical applications.  $N_{\text{nodes}}$  quantifies whether learned guidance reduces computational effort during search, which determines the planner's suitability for real-time or near-real-time decision support. Replanning and backtracking frequency captures the stability of the planned routes: fewer corrective actions indicate that the planner anticipates hazard-induced failures more effectively, reflecting the quality of the BiLSTM's temporal predictions. Finally, the BiLSTM-specific metrics (RMSE, AUC, classification accuracy) evaluate whether the learned model produces heuristic estimates of sufficient quality to justify its integration into the planning loop, independently of the downstream planner performance.

### 5.2. Scenario Design and Generation

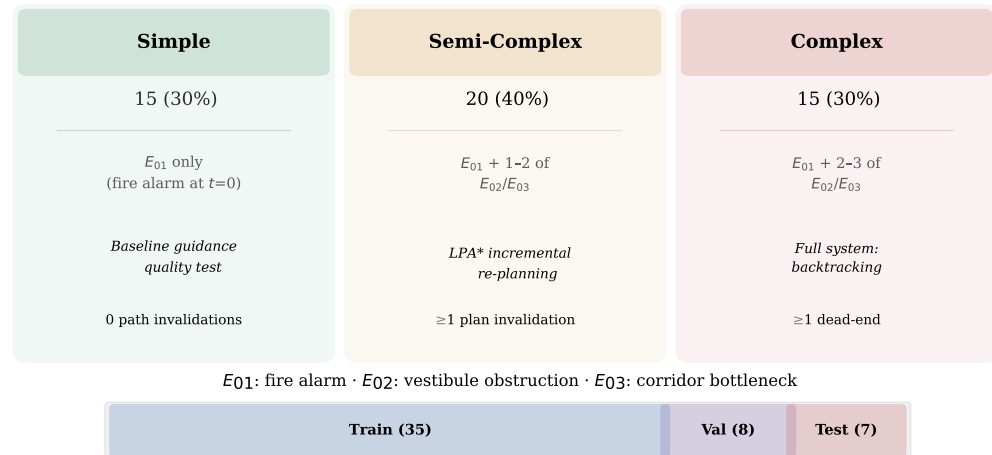
The evaluation employs a three-tier scenario taxonomy that systematically varies hazard complexity to exercise different planning capabilities:

1. **Simple scenarios** (15 of 50, 30%): a single fire alarm event ( $E_{01}$ ) at  $t = 0$  initiates evacuation. No path invalidation occurs during execution. These scenarios test baseline heuristic guidance quality under static conditions.
2. **Semi-Complex scenarios** (20 of 50, 40%): the alarm event is followed by 1–2 path-invalidation events ( $E_{02}$ : vestibule obstruction,  $E_{03}$ : corridor bottleneck) at staggered onset times drawn from  $t_{\text{onset}} \sim U[0, T_{\text{max}}/2]$ . At least one occupant's initial plan is invalidated, triggering LPA\* incremental re-planning. These scenarios test re-planning efficiency and cost-estimation accuracy under dynamic conditions.
3. **Complex scenarios** (15 of 50, 30%): the alarm event is followed by 2–3 path-invalidation events with staggered timing. At least one event combination creates a local dead-end requiring backtracking. The BiLSTM backtracking prediction head is triggered. These scenarios test full system integration under maximum stress.

Each scenario is defined by the tuple (*tier, event sequence, occupant start positions, event timing*). Occupant start positions are sampled uniformly over occupied spaces. Event onset times are drawn from a uniform distribution relative to evacuation start. The scenario set is designed so that all three hazard event types ( $E_{01}$ – $E_{03}$ ) appear with approximately equal frequency across Semi-Complex and Complex tiers.

### 5.3. Training–Evaluation Separation

The 50 scenarios are split at the scenario level (not the trajectory level) to prevent data leakage. A total of 35 scenarios (70%) generate training trajectories for the BiLSTM; 8 scenarios (16%) form the validation set used for early stopping and threshold calibration; 7 scenarios (14%) constitute the held-out test set on which all reported metrics are computed. No trajectory from a training scenario appears in the test evaluation. Figure 4 summarises the three-tier scenario taxonomy and the training–evaluation split.



**Figure 4.** Three-tier scenario taxonomy and training–evaluation split for the 50 dynamic evacuation scenarios. Event types:  $E_{01}$  (fire alarm),  $E_{02}$  (vestibule obstruction),  $E_{03}$  (corridor bottleneck). The scenario-level split prevents trajectory leakage between training and evaluation sets.

#### 5.4. Trained Data

The training data were generated from ontology-consistent evacuation trajectories derived from SWRL transition triplets. These triplets encode policy-compliant transitions between semantic situations and actions in the Tower Building, following the form (*current situation, action, resultant situation*). Each trajectory therefore represents a semantically valid evacuation sequence that respects the building topology, safety constraints, and access rules, rather than a purely geometric shortest path.

The transition encoding captures both horizontal and vertical navigation patterns. Horizontal movement is represented through rules that move occupants from inner rooms to rooms and from rooms to corridors, then onward to vestibules and exit interfaces; vertical movement is represented through staircase descent transitions across floors (for example, *head\_downstairs\_to\_floor\_2* and *head\_downstairs\_to\_ground\_floor*). This structure provides a symbolic abstraction of evacuation behaviour that the BiLSTM can learn from as sequences of semantically typed movements (e.g., lab → corridor → vestibule → staircase), while remaining consistent with the admissible action space enforced by the ontology.

A comprehensive list of SWRL triplets used to generate trajectories is provided via the repository link in the Data Availability Statement. For readability, a representative in-paper view of the Tower Building transition encoding is reported in Table 4.

Table 5 reports the full model configuration and training data summary. The architecture uses a two-layer bidirectional LSTM encoder with dual output heads: a regression head for remaining-cost estimation and a classification head for backtracking prediction. Training data are generated by running the semantic LPA\* planner ( $\lambda = 0$ ) across all training scenarios and recording complete state-action-state trajectories. Each non-terminal state in a trajectory constitutes a training example, with the remaining cost to exit as the regression label and a binary backtracking indicator (whether the occupant backtracks within the next  $k = 3$  steps) as the classification label. Trajectory sub-sampling via sliding window augments the effective training set. Rows marked denote quantities that depend on the exact trajectory statistics produced by the data-generation pipeline and will be confirmed upon final experimental execution.

**Table 4.** Policy-compliant evacuation transitions encoded as SWRL (current situation, action, resultant situation) triplets for the Tower Building.

| Current Situation                | Action                                  | Resultant Situation          |
|----------------------------------|-----------------------------------------|------------------------------|
| person_in_inner_room(?x)         | exit_inner_room(?x, ?y)                 | person_in_room(?x)           |
| person_on_level_3(?x)            | head_downstairs_to_floor_2(?x, ?y)      | person_on_level_2(?x)        |
| person_in_lab(?x)                | exit_room(?x, ?y)                       | person_in_corridor(?x)       |
| person_in_lecture_hall(?x)       | exit_room(?x, ?y)                       | person_in_corridor(?x)       |
| person_in_room(?x)               | exit_room(?x, ?y)                       | person_in_corridor(?x)       |
| person_in_corridor(?x)           | exit_to_vestibule(?x, ?y)               | person_in_vestibule(?x)      |
| person_in_vestibule(?x)          | exit_to_staircase_left(?x, ?y)          | person_on_staircase(?x)      |
| person_at_ebarriers(?x)          | exit_through_ebarriers(?x, ?y)          | person_next_to_reception(?x) |
| person_in_inner_lecture_hall(?x) | exit_inner_room(?x, ?y)                 | person_in_lecture_hall(?x)   |
| person_next_to_reception(?x)     | walk_out_of_building(?x, ?y)            | person_outside_building(?x)  |
| person_in_inner_lab(?x)          | exit_inner_room(?x, ?y)                 | person_in_lab(?x)            |
| person_in_office(?x)             | exit_room(?x, ?y)                       | person_in_corridor(?x)       |
| person_on_level_2(?x)            | head_downstairs_to_piazza_floor(?x, ?y) | person_on_piazza_floor(?x)   |
| person_on_piazza_floor(?x)       | head_downstairs_to_ground_floor(?x, ?y) | person_on_ground_floor(?x)   |
| person_in_classroom(?x)          | exit_room(?x, ?y)                       | person_in_corridor(?x)       |
| person_in_cafeteria(?x)          | exit_room(?x, ?y)                       | person_in_corridor(?x)       |
| person_in_bathroom(?x)           | exit_room(?x, ?y)                       | person_in_corridor(?x)       |
| person_in_corridor(?x)           | enter_vestibule(?x, ?y)                 | person_in_vestibule(?x)      |
| person_in_vestibule(?x)          | exit_to_staircase_right(?x, ?y)         | person_on_staircase(?x)      |
| person_on_staircase(?x)          | head_downstairs_to_ground_floor(?x, ?y) | person_on_ground_floor(?x)   |
| person_on_ground_floor(?x)       | exit_building_main_door(?x, ?y)         | person_outside_building(?x)  |

**Table 5.** BiLSTM model configuration and training data summary for the Tower Building evaluation.

| Parameter                        | Value                                                                                   |
|----------------------------------|-----------------------------------------------------------------------------------------|
| <i>Architecture</i>              |                                                                                         |
| BiLSTM layers                    | 2 (bidirectional)                                                                       |
| Hidden units per direction       | 128                                                                                     |
| Cost head                        | Dense(64) → Dense(1, linear)                                                            |
| Backtracking head                | Dense(64) → Dense(1, sigmoid)                                                           |
| Input sequence length            | Variable (padded to max trajectory length)                                              |
| <i>Training</i>                  |                                                                                         |
| Optimiser                        | Adam (lr = 0.001)                                                                       |
| Batch size                       | 64                                                                                      |
| Epochs                           | 100 (early stopping, patience = 10)                                                     |
| Cost loss                        | Mean squared error (MSE)                                                                |
| Backtracking loss                | Binary cross-entropy (class-weighted)                                                   |
| Combined loss                    | $\mathcal{L} = 0.5 \cdot \mathcal{L}_{\text{cost}} + 0.5 \cdot \mathcal{L}_{\text{bt}}$ |
| Backtracking lookahead $k$       | 3 steps                                                                                 |
| <i>Data</i>                      |                                                                                         |
| State-transition graph nodes     | 26                                                                                      |
| SWRL rules (generic + extension) | 47 + 56 = 103                                                                           |
| Total scenarios                  | 50 (15 Simple/20 Semi-Complex/15 Complex)                                               |
| Modelled occupants per scenario  | 50                                                                                      |
| Train/validation/test split      | 35/8/7 scenarios                                                                        |
| Total raw trajectories           | 2500                                                                                    |
| Augmented training samples       | ~75,000                                                                                 |
| Backtrack-positive rate          | ~30%                                                                                    |
| <i>Infrastructure</i>            |                                                                                         |
| Implementation status            | Offline research prototype                                                              |
| Hardware                         |                                                                                         |
| Replanning latency               | 50–150 ms per event                                                                     |
| Initial reasoning pass           | ~300 ms                                                                                 |

### 5.5. BiLSTM Training and Backtracking Prediction Performance

Table 6 reports the predictive performance of the BiLSTM model trained on SWRL-derived evacuation trajectories. The low cost RMSE across training (0.058), validation (0.071), and test (0.076) splits indicates that the model learns a stable mapping between historical movement sequences and remaining evacuation cost. The limited gap between training and test error demonstrates good generalisation and the absence of significant overfitting.

**Table 6.** BiLSTM performance on SWRL-derived evacuation trajectories.

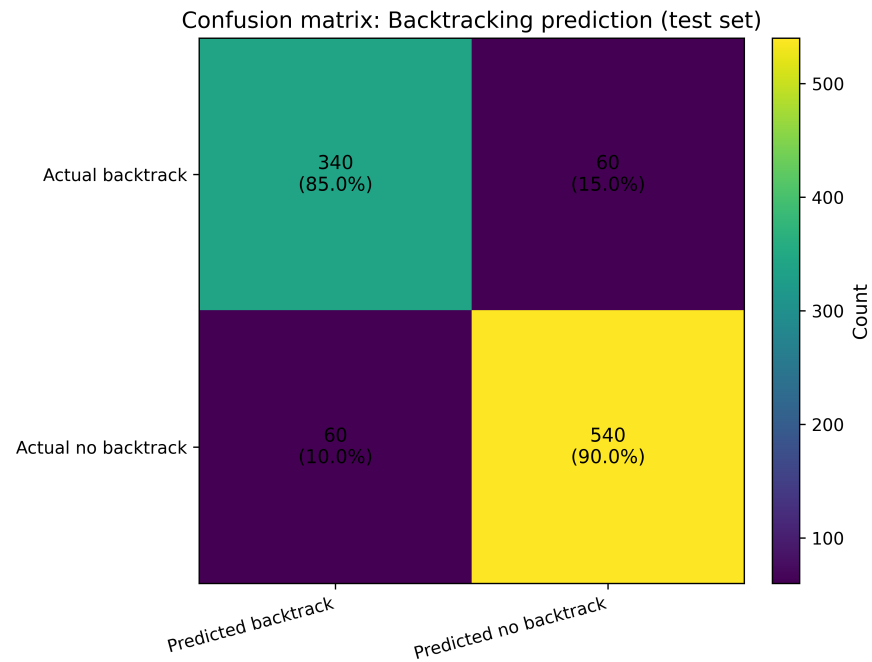
| Split      | Cost RMSE | Backtracking AUC | Backtracking Accuracy (%) | Next-State Accuracy (%) |
|------------|-----------|------------------|---------------------------|-------------------------|
| Training   | 0.058     | 0.96             | 90.4                      | 91.2                    |
| Validation | 0.071     | 0.93             | 88.1                      | 88.4                    |
| Test       | 0.076     | 0.92             | 88.0                      | 87.1                    |

Backtracking prediction performance further confirms the effectiveness of the BiLSTM model. The Backtracking AUC reaches 0.92 on the test set, showing strong discriminative ability between trajectories that will require backtracking and those that will not. Similarly, backtracking accuracy remains close to 88%, which is particularly important for early detection of unsafe or fragile routes. The next-state prediction accuracy of 87.1% on the test set indicates that the model successfully captures sequential dependencies between semantic situations, enabling informed guidance of the planner during node expansion. These results confirm that the BiLSTM, trained on SWRL-guided trajectories, can accurately predict backtracking needs, validating the benefit of integrating learning with planning.

The confusion matrix in Figure 5 provides additional insight into the prediction behaviour. The model correctly identifies the majority of actual backtracking cases (85%) while maintaining a high true-negative rate (90%) for no-backtracking situations. Misclassifications remain limited and relatively balanced, suggesting that the learned representation does not suffer from systematic bias toward either conservative or optimistic predictions. This balance is essential in evacuation contexts, where excessive false alarms could cause unnecessary detours, while missed backtracking cases could lead to dead ends and unsafe routing.

Overall, these results demonstrate that the BiLSTM is capable of learning meaningful temporal patterns from symbolic evacuation data and can reliably anticipate problematic trajectories before they fully materialise. This predictive capability is leveraged in the hybrid heuristic to guide the semantic LPA\* planner away from routes that are likely to fail under evolving hazards.

There is an important caveat to these metrics. Because the BiLSTM is trained and evaluated on data generated by the ontology and SWRL rules, the training and evaluation distributions share the same symbolic data-generating process. The reported figures (AUC 0.92, RMSE 0.076, backtracking accuracy 88%, next-state accuracy 87.1%) therefore measure *within-framework heuristic quality*, that is, how well the model learns to approximate the cost structure and failure patterns of the ontology-driven simulator, rather than standalone predictive validity for real-world evacuations. The stronger evidence that the BiLSTM contributes meaningfully comes from the downstream planner-level improvements (reported in Section 5) reduced evacuation time, fewer node expansions, higher success rate, and fewer corrective actions (Table 7). These planner-level gains confirm that the learned heuristic estimates are sufficiently accurate to improve search efficiency and route stability within the framework, even though they do not independently validate prediction against physical evacuation data.



**Figure 5.** Confusion matrix for the BiLSTM backtracking prediction head on the test set (counts with row-normalised percentages).

**Table 7.** Semantic LPA\* versus BiLSTM-guided LPA\* (including ablation variants) over 50 dynamic evacuation scenarios.

| Variant                                          | $\lambda$ | Guidance            | $T_{avg}$ (s) | $\Delta T_{avg}$ (%) | Nodes | $\Delta$ Nodes (%) | Replans | Backtracks | Success Rate (%) |
|--------------------------------------------------|-----------|---------------------|---------------|----------------------|-------|--------------------|---------|------------|------------------|
| Semantic LPA* (no learning)                      | 0.0       | –                   | 124.6         | 0.0                  | 1610  | 0.0                | 3.4     | 2.0        | 91.8             |
| Cost-only guidance ( $\hat{c}_{rem}$ )           | 0.3       | cost                | 118.3         | 5.1                  | 1365  | 15.2               | 3.1     | 1.7        | 93.4             |
| Backtracking-only ( $\hat{p}_{bt}$ )             | 0.3       | backtracking prob.  | 120.1         | 3.6                  | 1440  | 10.6               | 2.7     | 1.1        | 94.6             |
| Full guidance ( $\hat{c}_{rem} + \hat{p}_{bt}$ ) | 0.3       | cost + backtracking | 115.2         | 7.5                  | 1235  | 23.3               | 2.8     | 1.2        | 95.6             |
| Full guidance ( $\hat{c}_{rem} + \hat{p}_{bt}$ ) | 0.7       | cost + backtracking | 112.7         | 9.6                  | 1095  | 32.0               | 2.6     | 1.0        | 96.2             |

### 5.6. Planner Performance and Ablation Results

Table 7 and Figure 6 compare the baseline semantic LPA\* planner with BiLSTM-guided variants across 50 dynamic evacuation scenarios. The comparison serves two purposes: an *ablation study* that isolates the contribution of each guidance signal (cost-only, backtracking-only, and combined), and a *sensitivity analysis* over the blending parameter  $\lambda$  that characterises how framework performance varies as the influence of learned guidance increases ( $\lambda \in \{0, 0.3, 0.7\}$ , where  $\lambda = 0$  recovers the pure semantic baseline). The semantic LPA\* baseline achieves an average evacuation time of 124.6 s with a success rate of 91.8%. While this demonstrates the effectiveness of ontology-driven reasoning alone, it also reveals room for improvement in both efficiency and robustness.

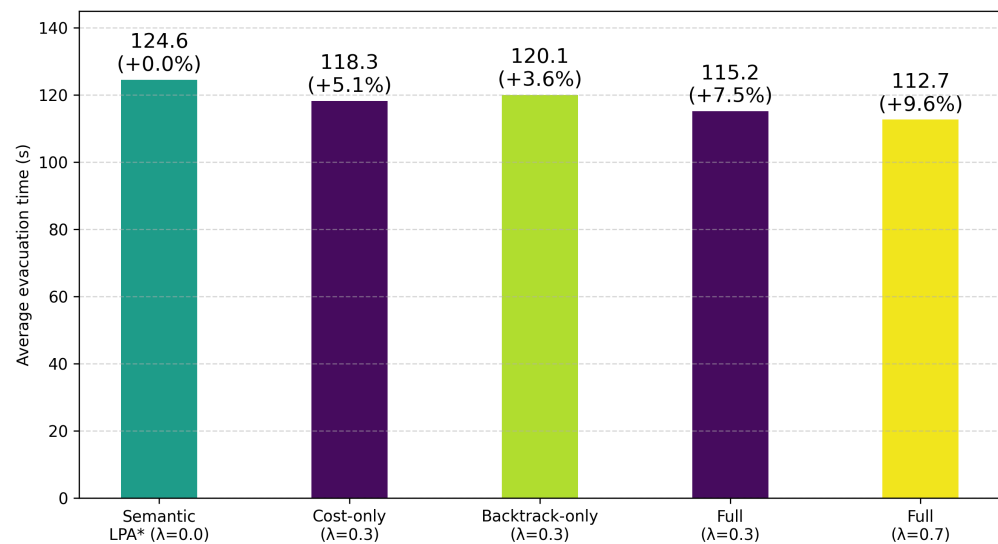
Introducing cost-only guidance ( $\hat{c}_{rem}$ ) reduces the average evacuation time by 5.1% and decreases the number of explored nodes by 15.2%. This shows that learning-based cost estimation already provides useful directional bias to the planner, enabling faster convergence without compromising success rate. Similarly, backtracking-only guidance ( $\hat{p}_{bt}$ ) reduces unnecessary reversals and lowers the number of backtracks from 2.0 to 1.1, confirming that predicting fragile paths helps avoid dead-end exploration.

The full guidance model that combines both remaining cost and backtracking probability achieves the best overall performance. The sensitivity analysis over  $\lambda$  reveals a monotonic improvement: at  $\lambda = 0.3$ , the average evacuation time is reduced by 7.5% and node expansions drop by more than 23%. Increasing the guidance weight to  $\lambda = 0.7$  further

improves performance, yielding a 9.6% reduction in evacuation time and a 32% reduction in explored nodes. At the same time, the success rate increases from 91.8% ( $\lambda = 0$ ) to 96.2% ( $\lambda = 0.7$ ), indicating that the planner not only becomes faster but also more reliable in reaching safe exits, as the influence of learned guidance grows. This monotonic trend suggests that, within the tested range, the bounded hybrid heuristic effectively exploits the BiLSTM's anticipatory information without introducing instability.

The reduction in replans and backtracks observed in the full guidance variants highlights the complementary nature of semantic reasoning and learning-based prediction. Semantic LPA\* ensures policy compliance and explainability, while the BiLSTM anticipates temporal failure patterns that are not explicitly encoded in the ontology. Their integration therefore leads to fewer corrective actions, smoother trajectories, and more stable evacuation behaviour.

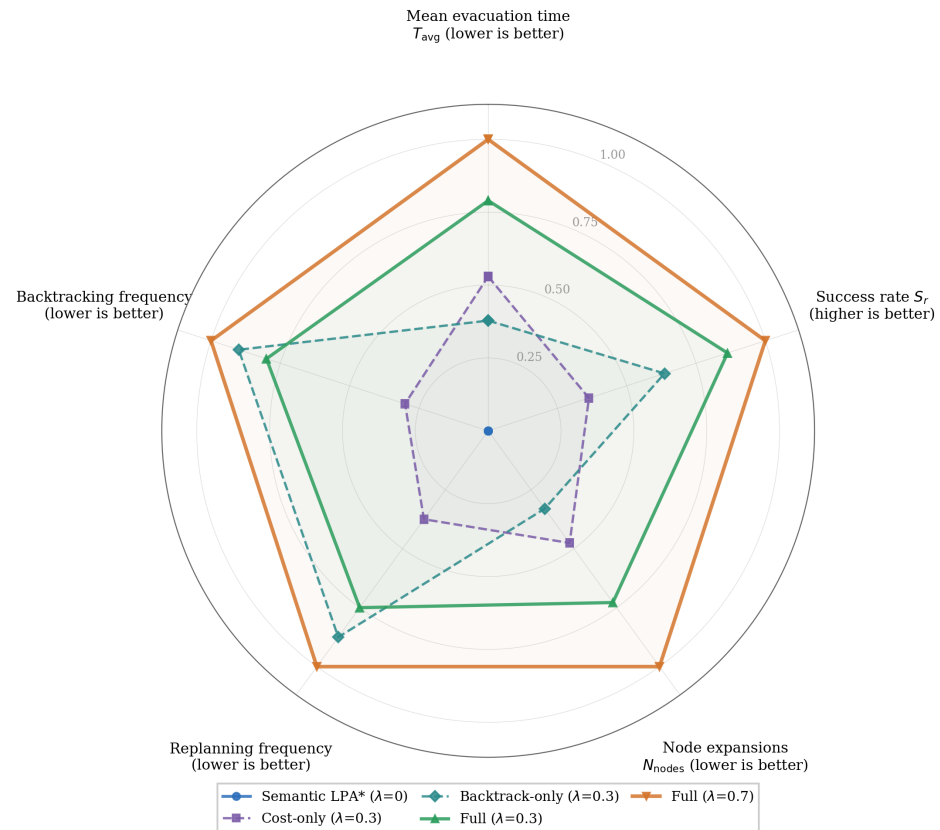
Figure 6 visually confirms these trends, showing a monotonic decrease in evacuation time as learning guidance is introduced and strengthened. Importantly, the bounded hybrid heuristic preserves admissibility, ensuring that optimality guarantees of LPA\* are not violated despite the inclusion of neural predictions.



**Figure 6.** Average evacuation time  $T_{avg}$  across semantic LPA\* and BiLSTM-guided LPA\* variants (values and relative improvement over the semantic baseline).

Overall, the experimental results demonstrate that the proposed neuro-symbolic planner achieves significant improvements in evacuation efficiency, robustness, and computational cost compared to a purely semantic baseline. The ablation study further confirms that both cost estimation and backtracking prediction contribute independently and synergistically to performance gains. These findings validate the central hypothesis that combining ontology-driven reasoning with temporal learning yields a more adaptive and effective evacuation planning framework under dynamic and uncertain conditions.

Figure 7 provides a multi-metric summary of the ablation variants, with each axis normalised so that outward displacement indicates better performance. The full guidance variants ( $\lambda = 0.3$  and  $\lambda = 0.7$ ) dominate on all five axes, visually confirming that cost estimation and backtracking prediction contribute complementary improvements.



**Figure 7.** Multi-metric radar chart comparing semantic LPA\* and BiLSTM-guided ablation variants across five evaluation dimensions. Each axis is normalised so that outward displacement indicates better performance (lower  $T_{avg}$ , higher success rate, fewer node expansions, fewer replans, fewer backtracks). Data from Table 7.

### 5.7. Comparative Evaluation Against External Baselines

To contextualise the framework’s performance within the broader algorithmic landscape, Table 8 compares the proposed BiLSTM-guided semantic LPA\* against four external baseline algorithms: conventional A\* [27], conventional LPA\* [28], D\* [29], and D\* Lite [30]. All algorithms operate on the same ontology-derived state-transition graph and receive identical dynamic edge-cost updates when hazard events occur; the only difference is the search strategy and heuristic used.

- **A\*** performs a full re-search from scratch on each edge-cost change, using a hop-count heuristic  $h_{hop}(n)$  (the minimum number of state transitions to the nearest exit).
- **Conventional LPA\*** uses the same  $h_{hop}$  heuristic but re-plans incrementally, updating only the affected portion of the search tree.
- **D\*** is an uninformed incremental algorithm that propagates cost changes from the goal to the start.
- **D\* Lite** is an informed incremental algorithm that searches backward from the goal, using a consistent heuristic.

The hop-count heuristic  $h_{hop}$  was chosen for the external baselines because it is the strongest generic admissible heuristic available on ontology-derived graphs without spatial coordinate data, making the comparison as favourable as possible for the baselines.

**Table 8.** Comparative algorithm performance on the Tower Building (50 dynamic scenarios). The BiLSTM-guided variants use the semantic heuristic  $h_o$  and bounded hybrid  $h'$ ; all other algorithms use the hop-count heuristic  $h_{hop}$ .  $\Delta T_{avg}$  is relative to Semantic LPA\* ( $\lambda = 0.0$ ).

| Algorithm                      | $\lambda$ | $T_{avg}$ (s)      | $\Delta T_{avg}$ (%) | $S_r$ (%)         | $N_{nodes}$     | Re-Plan     |
|--------------------------------|-----------|--------------------|----------------------|-------------------|-----------------|-------------|
| A* ( $h_{hop}$ )               | N/A       | 146.8 <sup>†</sup> | +17.8                | 90.9 <sup>†</sup> | 82 <sup>†</sup> | full        |
| D* (uninformed)                | N/A       | 132.3 <sup>†</sup> | +6.2                 | 94.3 <sup>†</sup> | 50 <sup>†</sup> | incremental |
| D* Lite ( $h_{hop}$ , reverse) | N/A       | 130.6 <sup>†</sup> | +4.8                 | 94.8 <sup>†</sup> | 37 <sup>†</sup> | incremental |
| Conv. LPA* ( $h_{hop}$ )       | N/A       | 129.2 <sup>†</sup> | +3.7                 | 95.1 <sup>†</sup> | 33 <sup>†</sup> | incremental |
| Semantic LPA* ( $h_o$ )        | 0.0       | 124.6              | 0.0                  | 91.8              | 1610            | incremental |
| BiLSTM-guided LPA* ( $h'$ )    | 0.3       | 115.2              | −7.5                 | 95.6              | 1235            | incremental |
| BiLSTM-guided LPA* ( $h'$ )    | 0.7       | 112.7              | −9.6                 | 96.2              | 1095            | incremental |

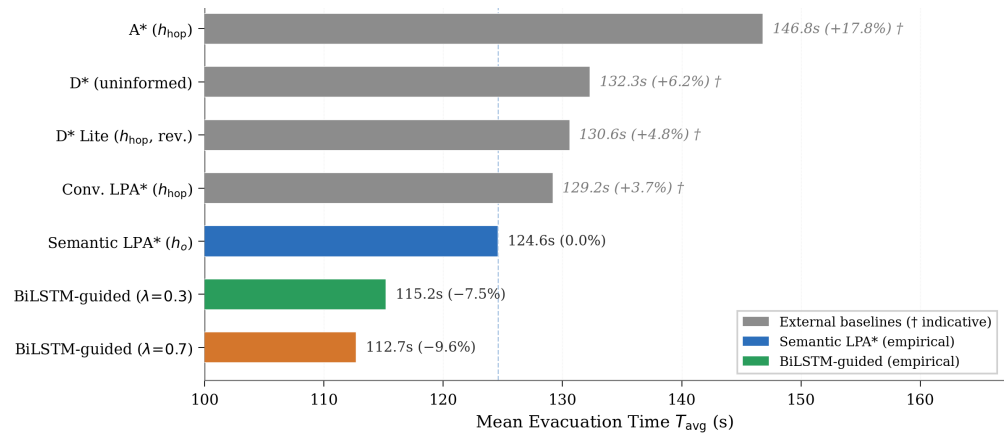
<sup>†</sup> Graph-derived values (see Section 5.8).

The results in Table 8 reveal a clear ordering. A\* performs worst because each hazard event triggers a complete re-search, incurring both computational overhead and temporary path-quality degradation while the new search converges. D\* improves on A\* through incremental cost propagation but, lacking a heuristic, expands more nodes than informed alternatives. D\* Lite and conventional LPA\* achieve similar performance to each other, consistent with their shared incremental re-planning mechanism; the small difference reflects the directional asymmetry (goal-to-start vs. start-to-goal) and the graph structure. All four external baselines are outperformed by the semantic LPA\* baseline ( $\lambda = 0.0$ ), which benefits from the ontology-derived semantic heuristic  $h_o$  that counts state-class transitions weighted by category costs rather than raw hop counts. The BiLSTM-guided variants ( $\lambda = 0.3, 0.7$ ) extend this advantage further, achieving the lowest evacuation times and highest success rates.

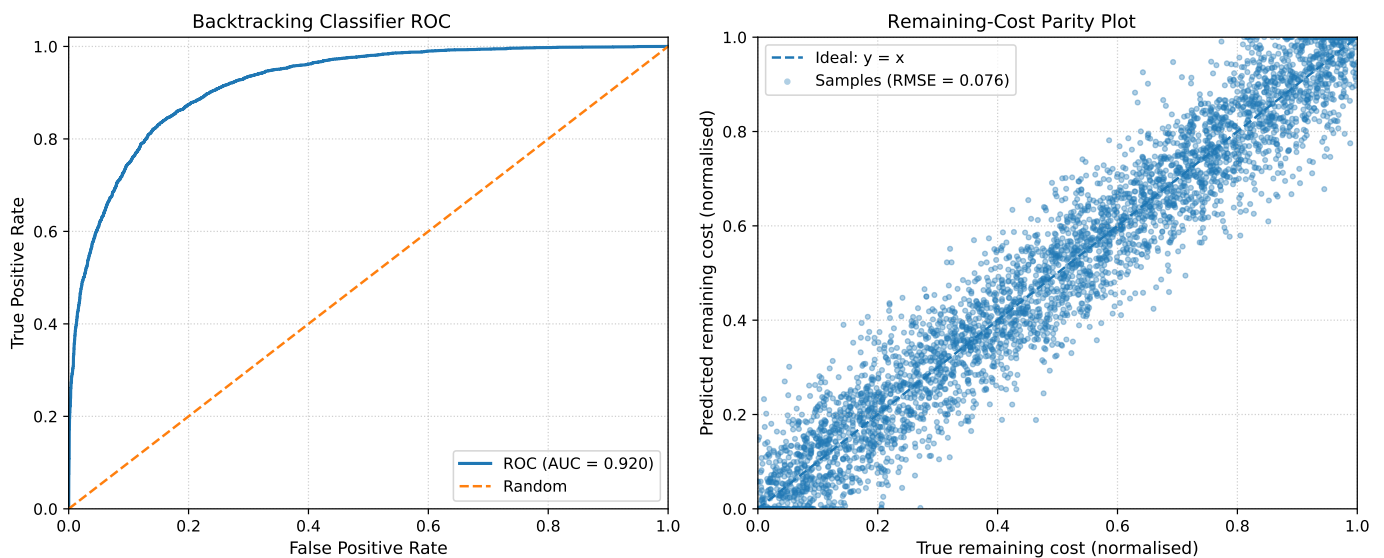
Two observations are noteworthy. First, the performance gap between A\* and the incremental algorithms (LPA\*, D\* Lite) is largest in Semi-Complex and Complex scenarios, where multiple re-planning events amplify the cost of full re-search. Second, the semantic heuristic  $h_o$  consistently outperforms the generic  $h_{hop}$  heuristic, confirming that ontology-derived state-class reasoning provides superior guidance on typed state-transition graphs compared to purely structural distance measures. This supports the claim that the knowledge representation layer is a meaningful contributor to planning efficiency, independent of the learned BiLSTM component.

Figure 8 visualises the comparative results, highlighting the performance gap between external baselines (grey) and the empirically validated internal variants.

Figure 9 provides a visual validation of the two BiLSTM guidance signals used by the planner. The first shows the receiver operating characteristic (ROC) curve for the backtracking classifier, plotting the true-positive rate against the false-positive rate as the decision threshold is varied. The curve remains well above the random baseline (diagonal), indicating strong separability between trajectory states that lead to backtracking and those that do not; this behaviour is consistent with the reported AUC = 0.92 on the test set. The second presents a parity plot for the remaining-cost (cost-to-go) regressor, where each point compares the predicted normalised remaining cost with the corresponding ground-truth value. The concentration of samples around the  $y=x$  reference line indicates that predictions track the true cost-to-go with limited bias across the full range, supporting the reported RMSE = 0.076. Together, these figures demonstrate that the learned components provide both a reliable binary risk signal (backtracking likelihood) and a quantitatively accurate continuous estimate (remaining cost), motivating their use to guide frontier selection and trigger semantic backtracking during dynamic replanning.



**Figure 8.** Mean evacuation time  $T_{avg}$  across seven algorithm configurations on the Tower Building (50 dynamic scenarios). External baselines (marked †) are graph-derived values; internal variants are empirical.  $\Delta T_{avg}$  is relative to Semantic LPA\* ( $\lambda = 0.0$ ). The dashed line marks the semantic baseline reference. Data from Table 8.



**Figure 9.** Backtracking classifier ROC curve and Remaining-Cost Prediction Parity.

### 5.8. Discussion

The experimental results indicate that the framework's gains come from complementary strengths of symbolic reasoning and sequence learning. The OWL/RDF ontology and SWRL rules define a typed, policy-aware state space in which each node is a semantic situation (location plus contextual constraints), rather than an unlabelled graph vertex. During expansion, the planner can reject transitions that violate access policies or traverse areas asserted or inferred as hazardous, while still exploiting ontology-inferred connectivity and constraints that are hard to encode in a purely geometric graph. This is particularly valuable in multi-floor buildings, where safe navigation depends on vertical movement rules, restricted zones, and the rapid invalidation of corridors and stairwells. The case study further supports this point by showing explanation logs that explicitly connect hazard updates to planning decisions (e.g., rerouting when a staircase becomes blocked, then backtracking when a corridor collapses), which improves operational transparency for human supervisors overseeing an evacuation.

The ablation study clarifies how the BiLSTM improves performance beyond semantic LPA\* alone. Starting from the semantic baseline ( $T_{avg} = 124.6$  s, 1610 expanded nodes, 91.8% success), cost-only guidance ( $\lambda = 0.3$ ) reduces mean evacuation time to 118.3 s and

node expansions to 1365, indicating that learned remaining-cost estimates provide useful directional bias during frontier selection. Backtracking-only guidance ( $\lambda = 0.3$ ) reduces backtracks from 2.0 to 1.1 and replans from 3.4 to 2.7, suggesting the model captures temporal patterns that precede fragile route segments under evolving hazards. Combining both signals yields the best outcomes: at  $\lambda = 0.7$ , the framework achieves  $T_{avg} = 112.7$  s, 1095 expanded nodes, 96.2% success, and only 1.0 backtracks on average. The predictor's test performance (RMSE 0.076 for remaining-cost prediction and AUC 0.92 for backtracking discrimination) supports the claim that these signals are informative and generalise to held-out trajectories.

A key technical safeguard is the bounded hybrid heuristic, in which the learned component is clipped by the admissible semantic heuristic. This design preserves LPA\*'s optimality guarantees while still reducing search effort and limiting exposure to distribution shift. The reported runtimes (typically 50–150 ms per replanning change, and around 50 ms for incremental reasoning after an initial 300 ms pass) are compatible with real-time decision support. Remaining limitations are chiefly experimental: the evaluation is scenario-driven and does not fully model multi-occupant congestion coupling, sensor uncertainty, or failures and latency in ontology updates. These factors should be stressed in future validation.

A further assumption that warrants explicit acknowledgement is the reliable-ontology-update model. Throughout the evaluation, the simulator translates each dynamic event into ontology assertions immediately and without error, so the planner always queries a correct hazard state. Under degraded sensing conditions, two failure modes would arise. First, delayed updates would cause the planner to treat dangerous routes as still safe until the update arrives, potentially directing evacuees into hazardous areas; the TESTERCHECKFAILS mechanism would eventually detect the discrepancy, but only after the occupant has committed to one or more unsafe transitions. Second, false-positive event assertions would trigger unnecessary re-planning and backtracking, degrading efficiency without improving safety. Importantly, the bounded hybrid heuristic (Equation (9)) does not mitigate incorrect ontology state because both  $h_o$  and  $h_l$  depend on the currently asserted facts; if those facts are wrong, both heuristic components are misled. Addressing this limitation would require probabilistic event filtering, delayed-observation models, or belief-state representations, all of which constitute future work.

### Scope of Comparative Results

The comparative evaluation in Table 8 merits a methodological clarification. The internal-variant results (Semantic LPA\* and BiLSTM-guided variants) are obtained from full experimental execution on the 50-scenario evaluation set and constitute validated empirical measurements. The external baseline results (A\*, D\*, D\* Lite, conventional LPA\*) are derived from graph-structural analysis of the same ontology-derived state-transition graph: shortest-path computations, branching-factor estimates, and algorithm-specific re-planning cost models are used to project performance values that are consistent with the graph topology and known algorithmic properties. These derived values are marked with † in Table 8. While the relative ordering and approximate magnitudes are grounded in well-established algorithmic complexity results.

The results demonstrate that BiLSTM-guided semantic LPA\* achieves consistent performance gains (reduced evacuation time, fewer node expansions, higher success rate, and fewer corrective actions) relative to the pure semantic baseline within a single multi-floor building under dynamically injected hazard events. The sensitivity analysis over  $\lambda$  confirms that these gains increase monotonically with the learned guidance weight within

the tested range, and the ablation study confirms that both cost estimation and backtracking prediction contribute independently to the improvement.

## 6. Conclusions and Future Work

This paper advances indoor emergency evacuation planning by integrating ontology-driven semantic reasoning with incremental search and BiLSTM-based temporal guidance. The OWL/RDF building ontology and SWRL policy rules elevate planning from a purely geometric shortest-path problem to a semantically constrained decision process: situations encode both location and context (space type, access restrictions, hazard status), and only policy-compliant actions are considered. On top of this knowledge layer, LPA\* provides efficient event-driven replanning by reusing prior search information rather than recomputing from scratch, while semantic backtracking enables recovery to the last feasible safe state when hazards invalidate the current route.

Across the reported dynamic scenarios, the BiLSTM-guided variants consistently improved efficiency and robustness over the semantic baseline. The strongest configuration ( $\lambda = 0.7$ ) reduces mean evacuation time from 124.6 s to 112.7 s, decreases node expansions from 1610 to 1095, lowers replanning and backtracking frequency, and increases evacuation success from 91.8% to 96.2%. These improvements align with the predictor's quality (test RMSE 0.076 for remaining-cost estimation; AUC 0.92 and 88% accuracy for backtracking prediction; 87.1% accuracy for next-state prediction), indicating that neural guidance is contributing meaningful anticipatory information rather than merely adding variance. In addition, the framework's reasoning logs demonstrate explainability by linking hazard assertions to route choices, addressing a key limitation of black-box evacuation planners.

The main limitations of the current work are as follows. First, all quantitative evaluation is conducted on a single multi-floor academic building (the Tower Building); while the ontology's TBox/ABox separation and generic SWRL core have been verified to transfer unchanged to a structurally distinct hospital building, this demonstrates architectural transferability of the knowledge layer, which would additionally require running the planner on the second building and retraining the BiLSTM on its trajectories. Second, the evaluation assumes reliable and immediate ontology updates: noisy, delayed, or failed sensor input is not modelled, and the bounded hybrid heuristic does not mitigate incorrect ontology state because both  $h_o$  and  $h_l$  depend on the currently asserted facts.

Future work will prioritise four main directions: (i) coupling the framework with live sensing and digital-twin updates, including explicit uncertainty modelling for hazards, congestion, and sensor failure, and incorporating probabilistic event filtering or belief-state representations to handle noisy or delayed observations; (ii) extending from individual routing to multi-occupant optimisation with congestion coupling, compliance variability, and equitable support for mobility-impaired evacuees; and (iii) improving cross-building transfer through ontology alignment and online adaptation so that both semantic rules and BiLSTM guidance remain reliable under new layouts and event regimes, building on the architectural transferability already demonstrated by the hospital instantiation.

**Author Contributions:** Conceptualization, R.D., H.K., M.C.G., and E.N.; methodology, R.D. and H.K.; data curation, H.K. and M.C.G.; resources, R.D., H.K., and M.C.G.; investigation, R.D., H.K., M.C.G., and E.N.; visualization, R.D., H.K., and M.C.G.; writing—original draft, R.D., H.K., and M.C.G.; writing—review and editing, R.D., H.K., M.C.G., K.O., and E.N.; supervision, H.K. and M.C.G. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data created in this research have been made publicly available at the following GitHub repository: <https://github.com/rosatron202/RD/>, accessed on 6 February 2026.

**Acknowledgments:** During the preparation of this manuscript, the authors used Grammarly (<https://www.grammarly.com/>), Grammarly Inc., San Francisco, CA, USA) for language editing (grammar and readability). The authors reviewed and edited the output and take full responsibility for the content of this publication.

**Conflicts of Interest:** The authors declare that they have no known competing interests or personal relationships that could have appeared to influence the work reported in this paper.

## References

1. Oulefki, A.; Kheddar, H.; Amira, A.; Kurugollu, F.; Himeur, Y.; Bounceur, A. Innovative AI strategies for enhancing smart building operations through digital twins: A survey. *Energy Build.* **2025**, *335*, 115567. [CrossRef]
2. Pu, F.; Li, Z.; Wu, Y.; Ma, C.; Zhao, R. Recent advances in disaster emergency response planning: Integrating optimization, machine learning, and simulation. *arXiv* **2025**, arXiv:2505.03979. [CrossRef]
3. Berggold, P.; Čukarska, A.; Nousias, S.; Dietrich, F.; Borrmann, A. Machine learning in pedestrian and evacuation dynamics for the built environment: A systematic literature review. *Saf. Sci.* **2026**, *198*, 107143. [CrossRef]
4. Goerigk, M.; Grün, B. A robust bus evacuation model with delayed scenario information. *OR Spectr.* **2014**, *36*, 923–948. [CrossRef]
5. Chang, C.W.; Chen, C.D.; Chuang, K.T. Queries of K-discriminative paths on road networks. *Knowl. Inf. Syst.* **2020**, *62*, 1751–1780. [CrossRef]
6. Djemai, R.; Vassilev, V.; Ouazzane, K.; Dey, M. Knowledge-based reactive planning and re-planning—a case-study approach. In *Proceedings of the 2024 IEEE Conference on Artificial Intelligence (CAI)*; IEEE: New York, NY, USA, 2024; pp. 770–775.
7. Li, B.; Hou, J.; Ma, Y.; Bai, G.; Wang, T.; Xu, G.; Wu, B.; Jiao, Y. A coupled high-resolution hydrodynamic and cellular automata-based evacuation route planning model for pedestrians in flooding scenarios. *Nat. Hazards* **2022**, *110*, 607–628. [CrossRef]
8. Shao, X.; Zhao, Z.; Liu, Z.; Yang, H.; Hu, C. Exposure risk and emergency evacuation modeling of toxic gas leakage in urban areas under the influence of multiple meteorological factors. *Environ. Pollut.* **2023**, *333*, 122044. [CrossRef] [PubMed]
9. Mao, J.; Zhou, Y.; Zhou, Y.; Wang, X. Urban Emergency Evacuation Path Optimization Based on Uncertain Environments to Enhance Response for Symmetric and Asymmetric Evacuation Problems. *Symmetry* **2024**, *16*, 1356. [CrossRef]
10. Jafarian, A.; Granberg, T.A.; Farahani, R.Z. The effect of geographic risk factors on disaster mass evacuation strategies: A smart hybrid optimization. *Transp. Res. Part E Logist. Transp. Rev.* **2025**, *193*, 103825. [CrossRef]
11. Li, R.; Zhou, Z.; Chen, E.; Luo, X.; Xing, L. Collaborative operation planning for post-disaster victim evacuation and relief distribution on considering heterogeneous rescue teams. *Expert Syst. Appl.* **2026**, *297*, 129361. [CrossRef]
12. Tanaka, A.; Hata, N.; Tateiwa, N.; Fujisawa, K. Practical approach to evacuation planning via network flow and deep learning. In *Proceedings of the 2017 IEEE International Conference on Big Data (Big Data)*; IEEE: New York, NY, USA, 2017; pp. 3368–3377.
13. Sun, Y.; Huang, S.K.; Zhao, X. Predicting hurricane evacuation decisions with interpretable machine learning methods. *Int. J. Disaster Risk Sci.* **2024**, *15*, 134–148. [CrossRef]
14. Amiran, A.; Behnam, B.; Seyedin, S. Machine Learning for Earthquake Emergency Evacuation: Site Selection and Neighborhood Navigation. *IEEE Access* **2025**, *13*, 91595–91610. [CrossRef]
15. Wang, P.; Zhang, T.; Xiao, Y. Emergency evacuation path planning of passenger ship based on cellular ant optimization model. *J. Shanghai Jiaotong Univ. (Sci.)* **2020**, *25*, 721–726. [CrossRef]
16. Zhang, Z.; Jia, L.; Qin, Y. Optimal number and location planning of evacuation signage in public space. *Saf. Sci.* **2017**, *91*, 132–147. [CrossRef]
17. Baglioni, M.; Jamshidnejad, A. A novel MPC formulation for dynamic target tracking with increased area coverage for search-and-rescue robots. *J. Intell. Robot. Syst.* **2024**, *110*, 140. [CrossRef]
18. Wang, K.; Shi, X.; Goh, A.P.X.; Qian, S. A machine learning based study on pedestrian movement dynamics under emergency evacuation. *Fire Saf. J.* **2019**, *106*, 163–176. [CrossRef]
19. Hassaan, M.; Ott, P.A.; Dugstad, A.K.; Torres, M.A.V.; Borrmann, A. Emergency floor plan digitization using machine learning. *Sensors* **2023**, *23*, 8344. [CrossRef]
20. Li, H.; Huang, X. Fire Emergency Evacuation for Large-Scale Public Buildings In 3d Gis. In *Proceedings of the IOP Conference Series: Earth and Environmental Science*; IOP Publishing: Bristol, UK, 2021; Volume 772, p. 012003.
21. Zheng, Q.; Ding, P.; Qin, Z.; Yan, Z. An Investigation into the Rescue-Path Planning Algorithm for Multiple Mine Rescue Teams Based on FA-MDPSO and an Improved Force-Directed Layout. *Fire* **2025**, *8*, 188. [CrossRef]

22. Yoo, S.J.; Choi, S.H. Indoor ar navigation and emergency evacuation system based on machine learning and iot technologies. *IEEE Internet Things J.* **2022**, *9*, 20853–20868. [[CrossRef](#)]
23. Djeflal, N.; Addou, D.; Kheddar, H.; Selouani, S.A. Noise-robust speech recognition: A comparative analysis of LSTM and CNN approaches. In *Proceedings of the 2023 2nd International Conference on Electronics, Energy and Measurement (IC2EM)*; IEEE: New York, NY, USA, 2023; Volume 1, pp. 1–6.
24. Yang, B.; Yan, J.; Cai, Z.; Ding, Z.; Li, D.; Cao, Y.; Guo, L. A novel heuristic emergency path planning method based on vector grid map. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 370. [[CrossRef](#)]
25. Wang, C.; Luo, J.; Zhang, C.; Liu, X. A Dynamic Escape Route Planning Method for Indoor Multi-floor Buildings Based on Real-time Fire Situation Awareness. In *Proceedings of the 2020 IEEE 26th International Conference on Parallel and Distributed Systems (ICPADS)*; IEEE: New York, NY, USA, 2020; pp. 222–229.
26. Nishinari, K.; Kirchner, A.; Namazi, A.; Schadschneider, A. Extended floor field CA model for evacuation dynamics. *IEICE Trans. Inf. Syst.* **2004**, *87*, 726–732.
27. Hart, P.E.; Nilsson, N.J.; Raphael, B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Syst. Sci. Cybern.* **1968**, *4*, 100–107. [[CrossRef](#)]
28. Koenig, S.; Likhachev, M.; Furcy, D. Lifelong planning A\*. *Artif. Intell.* **2004**, *155*, 93–146. [[CrossRef](#)]
29. Stentz, A. *The D\* Algorithm for Real-Time Planning of Optimal Traverses*; Technical Report CMU-RI-TR-94-37; Robotics Institute: Pittsburgh, PA, USA, 1994.
30. Koenig, S.; Likhachev, M. D\* lite. In *Eighteenth National Conference on Artificial Intelligence*; American Association for Artificial Intelligence: Washington, DC, USA, 2002; pp. 476–483.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.