Congestion-aware Multi-agent Reinforcement Learning for Wildfire Evacuation Routing

Bahareh Raei[1], Reza Safarzadeh[1], Xin Wang[1]

[1] Dept. of Geomatics Engineering, University of Calgary, 2500 University Dr NW, Calgary, AB T2N 1N4, Canada
(bahareh.mohammadraei, reza.safarzadeh, xcwang)@ucalgary.ca

Keywords: Wildfire evacuation, Multi-agent reinforcement learning, Adaptive vehicle routing, Disaster response

Abstract

Conventional navigation systems often cause severe bottlenecks during mass wildfire evacuations by routing vehicles onto the same capacity-limited corridors while ignoring advancing flame fronts. This paper introduces a congestion-aware multi-agent reinforcement learning (MARL) framework that models each road intersection as an independent Q-learning agent to balance route efficiency with strict hazard avoidance. During deployment, a batch-sequential mechanism dynamically adjusts these learned policies using real-time traffic, inherently dispersing vehicles away from overloaded roads. Evaluated on the real-world road network and parcel data of Lytton, British Columbia, the framework reduces peak edge congestion by 74% and achieves complete fire-zone avoidance compared to conventional fastest-path algorithms. With only a 7.4% increase in mean travel distance, these results demonstrate that distributed MARL policies yield significantly safer, more balanced, and highly scalable evacuation flows.

1. INTRODUCTION

Climate change is driving unprecedented increases in the frequency, intensity, and spatial extent of wildfires worldwide (Jain et al., 2022). Canada has experienced a particularly alarming trajectory: the 2023 season burned 18.5 million hectares and displaced over 232,000 residents (Jain et al., 2024). With wildfire risk projected to escalate through mid-century due to warming and drought (McEvoy et al., 2020), (Wotton et al., 2017), communities along the wildland-urban interface (WUI) face a mounting pressure. In this context, robust and efficient evacuation routing is essential for preserving human life.

Recent disasters starkly expose the vulnerabilities of evacuation routing. In 2021, a wildfire engulfed 90% of Lytton, British Columbia, within 15 minutes (Raei et al., 2025). The 2016 Fort McMurray wildfire forced 88,000 residents onto a single highway, causing 15-hour delays (Mamuji and Rozdilsky, 2019). Similarly, the 2018 Camp Fire in Paradise, California, left residents trapped in gridlocked traffic overtaking by flames, resulting in 85 fatalities and 23 documented burnover events (Link and Maranghides, 2023). These tragedies reveal a critical systemic failure: road networks designed for routine traffic are swiftly overwhelmed during mass evacuations.

This vulnerability is amplified by reliance on conventional navigation systems (e.g., Google Maps, Waze), which compute individually optimal shortest paths via Dijkstra's algorithm (Rahayuda and Santiari, 2021), (Safarzadeh et al., 2025). During mass evacuations, these applications inadvertently route thousands of users onto the same nominally optimal road segments, creating severe bottlenecks where capacity is most constrained (Raei et al., 2026), (Zhao et al., 2022). Furthermore, they lack real-time integration of advancing fire perimeters and emergent collective congestion. This dynamic creates a paradox of individual optimality leading to collective failure, akin to the Braess paradox (Roughgarden, 2005). As illustrated in Figure 1, while traditional algorithms force evacuees into a gridlocked corridor, a distributed multi-agent approach coordinates traffic at the intersection level, dynamically redistributing load across alternative routes to facilitate efficient network-wide egress.

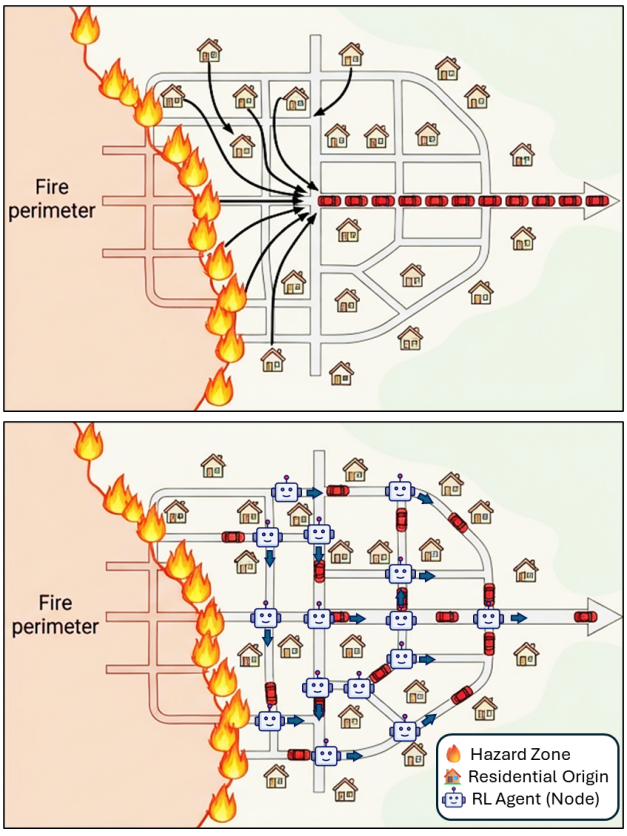

Figure 1. Comparison of evacuation routing strategies. (Top) Traditional navigation systems converge on nominally optimal routes, creating severe congestion and a central bottleneck. (Bottom) The proposed node-based multi-agent architecture positions independent RL agents at intersections to distribute vehicles across available road segments, reducing gridlock and accelerating network evacuation.

Efforts to address evacuation routing span several domains, each with limitations. Classical network flow optimisation provides provably optimal solutions under static conditions (Hamacher

and Tjandra, 2001), but these centralised methods scale poorly and struggle to adapt to dynamic hazards. Agent-based simulations like MATSim and SUMO offer high-fidelity traffic modelling (W Axhausen et al., 2016), (Lopez et al., 2018), yet primarily evaluate pre-computed plans rather than generating adaptive strategies. GIS-based approaches capture spatial realism but typically rely on static algorithms that disregard evolving congestion (Cova and Johnson, 2003). Recently, reinforcement learning (RL) has demonstrated success in traffic signal control (Wei et al., 2021) and vehicle routing (Safarzadeh and Wang, 2024). However, its application to wildfire evacuation remains nascent. Early efforts include the PyroRL simulator (Guman et al., 2024) and indoor evacuation models (Zhou et al., 2024). Crucially, existing RL approaches predominantly operate in simplified grid or indoor environments, lacking integration with real-world road typologies and parcel-level geospatial data.

Wildfire evacuation routing is fundamentally a coupled, multi-actor problem: a single overloaded corridor can easily cascade into network-wide gridlock. A centralised controller computing system-optimal assignments is impractical given the infrastructure damage and timing constraints of active emergencies (Cova and Johnson, 2003). Instead, the problem demands *distributed, multi-agent* decision-making, where individual nodes autonomously coordinate traffic to balance load across alternative paths.

Furthermore, accurate origin modelling is critical. Routing models that distribute demand uniformly across a network overlook the reality that evacuations begin at specific residential parcels. This spatial distribution profoundly shapes bottleneck formation. Without authoritative cadastral data, models risk misrepresenting the magnitude and geography of congestion. To date, no prior work unifies distributed multi-agent learning, real-world road networks, parcel-level demand modelling, and coupled hazard-congestion awareness within a single architecture.

To address this gap, this paper presents a congestion-aware multi-agent reinforcement learning (MARL) framework for wildfire evacuation routing on real road networks. The principal contributions of this work are:

1. Distributed node-based MARL architecture. Each road intersection operates as an independent Q-learning agent that learns destination-directed navigation policies, enabling decentralised, scalable decision-making without a central controller.

2. Coupled congestion-hazard reward formulation. A composite reward function jointly penalises traffic congestion through tiered edge-occupancy penalties and fire-zone proximity through severe hazard penalties, training agents to balance route efficiency against both safety risks.

3. Parcel-integrated geospatial modelling. Government cadastral parcel data is integrated with OpenStreetMap road networks to generate realistic vehicle origins by snapping residential parcel centroids to their nearest drivable nodes, providing demand distributions that reflect actual community structure.

4. Batch-sequential routing with dynamic feedback. Vehicles are routed in sequential batches with edge-usage statistics updated between batches, enabling the trained agents to dynamically redistribute traffic and prevent the bottleneck convergence characteristic of static shortest-path approaches.

5. The framework is evaluated on the village of Lytton, British Columbia, a small and constrained highway-corridor community that suffered catastrophic wildfire damage in 2021, demonstrating substantial reductions in peak congestion and fire-zone incursions compared with conventional shortest-path routing.

2. RELATED WORK

This section reviews two bodies of literature most relevant to the proposed framework: evacuation route optimisation methods and reinforcement learning approaches for transportation and emergency management.

2.1 Evacuation Route Optimisation

Routing algorithms are central to evacuation planning, as the quality of assigned routes directly governs both clearance time and occupant safety. However, the inherent unpredictability of wildfire advancement, combined with rapidly evolving traffic conditions, renders static routing formulations insufficient for practical deployment.

One research direction incorporates empirical travel behaviour into evacuation models. de Oliveira e Silva et al. (de Oliveira e Silva et al., 2022) showed that historical GPS trajectories reveal systematic deviations from shortest-path assumptions and proposed a trajectory-driven recommendation model accounting for individual habits. Rahimi et al. (Rahimi et al., 2020) clustered spatiotemporal movement patterns to enable real-time rerouting that jointly considers fire progression and traffic states. These studies underscore that realistic modelling of evacuee decision-making is essential for actionable routing guidance.

Graph-based shortest-path methods remain the most widely studied family of approaches. Zhu et al. (Zhu et al., 2022) extended Dijkstra's algorithm into three dimensions for urban flood evacuations, reducing both hazard exposure and travel time. Gai et al. (Gai et al., 2014) combined Dijkstra and A* search in a bi-objective framework that simultaneously minimises travel duration and maximises route safety. Saadatseresht et al. (Saadatseresht et al., 2009) employed a multi-objective evolutionary strategy that prioritises corridors according to household-level characteristics, while Udhan et al. (Udhan et al., 2022) augmented Dijkstra's algorithm with dynamic traffic predictions. More recently, machine-learning-driven routing has gained traction: Jain et al. (Jain, 2025) and Ekaputra et al. (Ekaputra et al., 2024) fused GPS traces, fire-spread models, and road-closure feeds to produce adaptive recommendations that anticipate collective evacuation patterns.

Despite these advances, a common limitation persists: all approaches reviewed above treat routing as an *individual-vehicle* problem, computing each path independently without accounting for the feedback loop between simultaneous routing decisions and emergent congestion. None incorporates a mechanism through which vehicles can implicitly coordinate their choices to spread traffic across the network and steer away from both congestion hotspots and advancing fire zones.

2.2 Reinforcement Learning for Evacuation

Reinforcement learning (RL) has attracted growing interest as a means of generating adaptive routing policies under dynamic

hazard conditions. Early work focused on single-agent formulations for natural-disaster evacuation. Takabatake and others (Takabatake et al., 2025) integrated Q-learning with agent-based tsunami evacuation simulations to optimise individual paths across road and pedestrian networks in two Japanese coastal communities, demonstrating both effectiveness and robustness. Mas et al. (Mas et al., 2024) proposed an RL-based tsunami guidance system for vehicle routing, motivated by casualties attributed to traffic congestion during the 2011 Great East Japan Earthquake, and validated RL performance against shortest-path baselines. Li et al. (Li et al., 2023) introduced the ReinforceRouting model, which optimises evacuation routes on large street networks under dynamic traffic and hazard conditions, outperforming both traditional RL and Dijkstra-based methods in safety scores and planning speed.

Deep RL architectures have extended these ideas to more complex scenarios. Yang et al. (Yang et al., 2025) applied the Asynchronous Advantage Actor-Critic (A3C) algorithm to real-time airport evacuation, generating routes reactively but without multi-agent coordination. Gu et al. (Gu et al., 2023) demonstrated that DRL-based dynamic guidance in building evacuations outperforms static signage by processing real-time sensor data to avoid crowding. Zhang et al. (Zhang et al., 2023) combined deep RL with 3D physical environments for crowd evacuation in congested scenarios, advancing the modelling of complex pedestrian dynamics.

Multi-agent RL (MARL) frameworks have begun to address the coordination challenge inherent in large-scale evacuations. Zhong et al. (Zhong and Ren, 2026) proposed a MARL approach for urban earthquake evacuation that generates adaptive routes on road networks with GIS-based decision support, demonstrating the potential for dynamic multi-agent coordination under evolving hazard conditions. Tang et al. (Tang et al., 2024) developed a data-driven RL framework for equitable bus-based evacuations, modelling the problem as a Markov Decision Process and routing buses on the San Francisco Bay Area network using OpenStreetMap data.

Notwithstanding this substantial progress, several gaps remain. First, most RL evacuation studies validate only in simplified or simulated environments, grid worlds, single buildings, or synthetic networks, without grounding in real geospatial road topologies and authoritative land-parcel data. Second, existing MARL formulations for evacuation have not jointly modelled *traffic congestion feedback* and *fire-hazard avoidance* within a unified reward structure. Third, none of the reviewed methods incorporates parcel-level origin modelling to ensure that evacuation demand reflects actual community layouts. The framework proposed in this paper addresses all three gaps by deploying distributed Q-learning agents on real OpenStreetMap road networks, training them with a coupled congestion, hazard reward, and generating vehicle origins from government cadastral parcels.

3. METHODOLOGY

This section presents the proposed framework for congestion-aware multi-agent reinforcement learning (MARL) for wildfire evacuation routing. As illustrated in Figure 2, the pipeline comprises three stages: (i) *data integration*, in which a transportation graph is constructed from OpenStreetMap (OSM) road data and vehicle origins are derived from cadastral parcel records; (ii) *distributed multi-agent Q-learning*, in which an independent learning agent is placed at every intersection to collaboratively discover congestion- and fire-aware routing policies; and (iii) *evacuation and evaluation*, in which trained agents are deployed to route vehicles toward designated exit points and the resulting routes are benchmarked against baseline methods. Each stage is detailed in the subsections that follow.

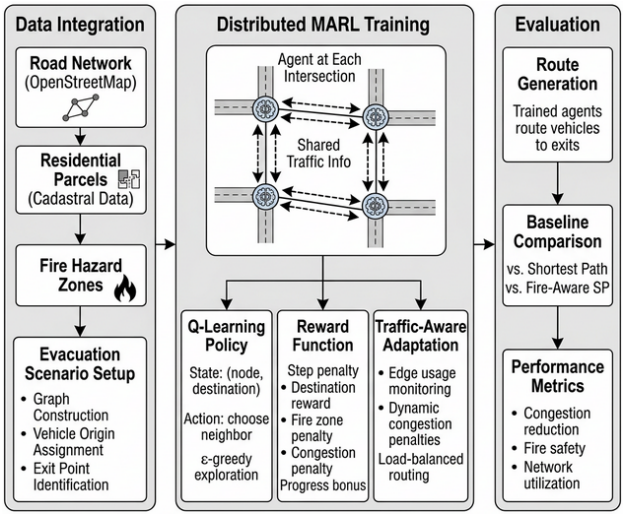

Figure 2. Overview of the proposed congestion-aware MARL framework for wildfire evacuation routing. The pipeline flows from geospatial data integration through distributed multi-agent Q-learning to evacuation deployment and evaluation.

3.1 Geospatial Data Integration

The first stage transforms heterogeneous geospatial data sources into a unified graph representation suitable for reinforcement learning.

3.1.1 Road Network Graph Construction. The drivable road network for a study area is retrieved from OpenStreetMap via the OSMnx library (Boeing, 2017). Let G_{osm} denote the raw directed multigraph returned by OSMnx, whose node identifiers are arbitrary OSM identifiers. Because reinforcement-learning agents require compact, sequential state indices, we construct a mapping

$$\phi : V_{\text{osm}} \to \{0,\, 1,\, \ldots,\, |V| - 1\}, \tag{1}$$

and define the undirected, weighted road graph $G = (V, E, w)$ as follows. Each node $v \in V$ corresponds to a road intersection or dead end, and is annotated with its geographic coordinates (x_v, y_v) in the WGS 84 coordinate reference system. Each edge $e = (u, v) \in E$ represents a road segment, with weight $w(u, v)$ equal to the segment length in metres as reported by OSMnx. The resulting graph preserves the full topology of the local road network while providing a compact state space for agent indexing.

3.1.2 Vehicle Origin Assignment from Cadastral Parcels. Evacuation demand must reflect the actual spatial distribution of residents rather than artificial assumptions. To this end, we obtain building footprints from OSM using the building tag, filtering for residential categories (residential, house, detached, apartments, etc.). For each residential polygon, its centroid is computed and projected to the road network by snapping it to the nearest graph node using a nearest-neighbour search on the Haversine distance:

$$v_i^* = \arg\min_{v \in V}\; d_{\text{haversine}}\big((x_i^{\text{home}},\, y_i^{\text{home}}),\, (x_v,\, y_v)\big), \tag{2}$$

where $(x_i^{\text{home}}, y_i^{\text{home}})$ denotes the centroid of the i-th residential parcel. Each snapped node v_i^* becomes the origin of one evacuating vehicle, thus directly coupling evacuation demand with the community's built form. In the current formulation, each residential parcel generates a single vehicle; extending the model to variable vehicle counts per dwelling (e.g., based on census data) is straightforward and does not alter the underlying framework.

3.1.3 Evacuation Scenario Definition. With the road graph G and vehicle origins $\{v_i^*\}$ established, the evacuation scenario is completed by specifying (1) a set of *exit nodes* $D \subset V$, typically located on highway junctions or boundary roads that lead away from the fire, and (2) a set of *fire hazard nodes* $V_f \subset V$, which represent intersections within or immediately adjacent to the active fire zone. These four components, G, $\{v_i^*\}$, D, and V_f, constitute the input to the MARL training module.

3.2 Distributed Multi-Agent Q-Learning

The core contribution of this work is a distributed reinforcement-learning architecture in which every intersection node in the road network hosts an independent Q-learning agent. During training, each agent explores the graph under fire constraints and a multi-objective reward signal, learning Q-values for *all* neighbour–destination pairs. This process encodes not only the optimal forwarding direction but also the relative quality of every alternative next-hop, producing a rich Q-value landscape that captures the network's structural diversity, including parallel corridors and detour options that a single shortest-path computation would discard. At deployment time, agents leverage this learned route repertoire in conjunction with real-time traffic feedback: a shared edge-usage mechanism penalises congested links, causing agents to shift from nominally optimal forwarding decisions toward the alternative paths discovered during training, thereby distributing traffic across the network.

3.2.1 Agent Architecture and State-Action Space. For every node $v \in V$, an agent $\mathcal{A}_v$ maintains a Q-table

$$Q_v : D \times \mathcal{N}(v) \to \mathbb{R}, \tag{3}$$

where D is the set of exit destinations and $\mathcal{N}(v)$ denotes the set of neighbours of v in G. The *state* observed by agent $\mathcal{A}_v$ when a vehicle arrives is the tuple (v, d), where $d \in D$ is the vehicle's assigned destination. The *action* is the choice of a neighbouring node $u \in \mathcal{N}(v)$ to which the vehicle should be forwarded. An action is deemed *valid* only if u is neither a known fire node nor a previously visited node along the current trajectory, i.e., $u \notin V_f \cup H$, where H is the vehicle's visit history. This constraint prevents the agent from routing vehicles into fire zones or creating routing loops. We note that including H in the action mask introduces a mild non-Markovian element, since the visit history is not encoded in the state tuple (v, d). In practice, this approximation acts as a regulariser that prevents short cycles during training; urban road networks typically provide sufficient alternative paths at each intersection, so the effect on Q-value convergence is limited.

3.2.2 Coupled Reward Function. The reward function is designed to jointly optimise route length, fire avoidance, and traffic distribution. Upon transitioning from node v to neighbour u toward destination d, the agent receives a composite reward:

$$\begin{aligned} R(v, u, d, H) = {} & R_{\text{step}} + R_{\text{dest}}(u, d) + R_{\text{fire}}(u) \\ & + R_{\text{revisit}}(u, H) + R_{\text{traffic}}(v, u) + R_{\text{progress}}(v, u, d), \end{aligned} \tag{4}$$

where the individual components are defined as follows.

Step Cost. A constant penalty $R_{\text{step}} = -1$ is applied at every hop to incentivise shorter paths.

Destination Reward. $R_{\text{dest}}(u, d) = +100$ if the vehicle reaches its assigned exit node ($u = d$), and 0 otherwise. Upon reaching the destination, the episode terminates successfully.

Fire Penalty. $R_{\text{fire}}(u) = -200$ if the next node lies in the fire zone ($u \in V_f$), and 0 otherwise. Entering a fire node also terminates the episode, teaching agents to avoid fire-affected regions entirely.

Revisit Penalty. $R_{\text{revisit}}(u, H) = -10$ if the next node has already been visited ($u \in H$), and 0 otherwise. This discourages routing loops.

Congestion Penalty. Traffic load on the edge (v, u) is monitored via shared edge-usage counters. The penalty follows a tiered scheme:

$$R_{\text{traffic}}(v, u) = \begin{cases} -80 & \text{if } c(v, u) \geq \tau_{\text{high}}, \\ -40 & \text{if } \tau_{\text{med}} \leq c(v, u) < \tau_{\text{high}}, \\ 0 & \text{otherwise}, \end{cases} \tag{5}$$

where $c(v, u)$ is the current vehicle count on edge (v, u), τ_{high} and τ_{med} are configurable congestion thresholds. This tiered design ensures that lightly loaded edges remain penalty-free while heavily loaded edges become progressively less attractive, thereby encouraging natural load balancing.

Progress Bonus. A bonus $R_{\text{progress}} = +2$ is awarded whenever a step reduces the shortest-path distance to the destination, i.e., $\delta_G(u, d) < \delta_G(v, d)$, where $\delta_G(\cdot, \cdot)$ denotes the weighted shortest-path distance in G; the bonus is 0 otherwise. This component provides a curriculum signal that accelerates convergence in large networks where unguided exploration alone would be prohibitively slow. Importantly, the progress bonus is only one component of the composite reward; the fire, congestion, and revisit penalties ensure that agents still discover and retain routes that deviate from the shortest path when safety or traffic conditions demand it.

3.2.3 Training Procedure. Training proceeds over N episodes using an ε-greedy exploration strategy. At the start of each episode, a starting node and a destination are selected uniformly at random from the set of vehicle origins and exit nodes, respectively. Algorithm 1 outlines the procedure.

The exploration rate ε decays exponentially from an initial value ε_0 toward a minimum $\varepsilon_{\min}$ with decay factor ρ, following the schedule $\varepsilon_k = \max(\varepsilon_{\min},\ \varepsilon_0 \cdot \rho^k)$. Early episodes are dominated by exploration, which allows agents to discover diverse routes, while later episodes exploit the learned Q-values to refine policies.

At each step within an episode, the agent at the current node v selects a next-hop u via ε-greedy action selection. With probability ε, a valid neighbour is chosen uniformly at random; otherwise, the agent selects $u^* = \arg\max_{u \in \mathcal{N}(v) \setminus (V_f \cup H)} Q_v(d, u)$.

Algorithm 1 Multi-Agent Q-Learning Training

Require: Road graph $G = (V, E, w)$; origin set O; destination set D; fire set V_f
Ensure: Trained Q-tables $\{Q_v\}_{v \in V}$
Initialise $Q_v(d, u) \leftarrow 0 \ \ \forall\, v \in V,\ d \in D,\ u \in \mathcal{N}(v)$
Broadcast V_f to all agents
for episode $k = 1, \ldots, N$ **do**
  $\varepsilon \leftarrow \max\big(\varepsilon_{\min},\ \varepsilon_0 \cdot \rho^k\big)$ {Decay exploration rate}
  $s \leftarrow \textsc{RandomChoice}(O)$;
  $d \leftarrow \textsc{RandomChoice}(D)$;
  $H \leftarrow \{s\};\ v \leftarrow s$
  while $v \neq d$ **and** steps $< T_{\max}$ **do**
    $u \leftarrow \mathcal{A}_v.\textsc{ChooseAction}(d,\ \varepsilon,\ V_f \cup H)$ {ε-greedy}
    if $u =$ `null` **then**
      break {No valid neighbours}
    end if
    $r \leftarrow R(v,\ u,\ d,\ H)$ {Equation (4)}
    $Q_v(d, u) \leftarrow Q_v(d, u) + \alpha\big[r + \gamma \max_{u'} Q_u(d, u') - Q_v(d, u)\big]$
    $H \leftarrow H \cup \{u\};\ v \leftarrow u$
  end while
end for
return $\{Q_v\}_{v \in V}$

Line 13 of Algorithm 1 applies the standard Q-learning (off-policy temporal-difference) update, where $\alpha \in (0, 1]$ is the learning rate and $\gamma \in [0, 1]$ is the discount factor. Although formal convergence guarantees for tabular Q-learning with dynamic action masks remain an open theoretical question, we empirically observe stable convergence of both the cumulative reward and the routing success rate within the specified training budget (see Section 4).

3.2.4 Traffic-Aware Adaptation. While the tiered congestion penalty (Equation 5) shapes the agents' Q-values during training, real-world deployment must also handle the *emergent* congestion caused by simultaneously routing many vehicles. At deployment time, the framework therefore applies a continuous, parameterised adjustment to the learned Q-values, enabling fine-grained control over the trade-off between route quality and congestion avoidance. This is achieved through two complementary mechanisms.

Shared Edge-Usage Counters. A global edge-traffic table $c : E \to \mathbb{Z}_{\geq 0}$ records the number of vehicles currently assigned to each edge. After a batch of vehicles is routed, the coordinator updates c by iterating over all active paths and incrementing the counter for every edge traversed. Each agent $\mathcal{A}_v$ has read access to the counters for its incident edges, enabling it to factor real-time congestion into its decisions. Edges are treated as undirected for counting purposes, $c(u, v) = c(v, u)$ for all $(u, v) \in E$, reflecting the assumption that vehicles travelling in either direction along a road segment contribute equally to congestion.

Traffic-Adjusted Action Selection. During the deployment (inference) phase, when a vehicle requests a route from node v to destination d, the agent computes *adjusted* Q-values that incorporate both real-time edge usage and fire-zone proximity:

$$\tilde{Q}_v(d, u) = Q_v(d, u) - \lambda_{\text{traffic}}\, c(v, u) - \lambda_{\text{fire}}\, \text{risk}(u), \qquad (6)$$

where $c(v, u)$ is the number of vehicles already routed through edge (v, u), $\text{risk}(u) \in \{0, 1\}$ is a binary indicator equal to 1 if $u \in V_f$ and 0 otherwise, and λ_{traffic}, λ_{fire} are weighting coefficients that control the sensitivity to congestion and hazard, respectively. The agent selects the action $u^* = \arg\max_u \tilde{Q}_v(d, u)$.

Because agents have learned Q-values for all neighbour, destination pairs during training, the Q-value landscape already encodes a ranked set of alternative routes; the deployment-time adjustment simply re-ranks these pre-learned alternatives based on current traffic and fire conditions. This two-phase design, RL-based policy learning followed by parameterised congestion correction, can be understood as computing a nominal routing policy that is then refined via penalty-based adjustments at deployment time.

To illustrate, suppose $Q_v(d, a_1) = 70$ and $Q_v(d, a_2) = 65$, so that the nominal policy favours a_1. If edge (v, a_1) carries three vehicles and lies near the fire zone while (v, a_2) carries one vehicle and is safe, then with $\lambda_{\text{traffic}} = 10$ and $\lambda_{\text{fire}} = 20$:

$$\tilde{Q}(a_1) = 70 - 10{\cdot}3 - 20 = 20, \quad \tilde{Q}(a_2) = 65 - 10{\cdot}1 = 55.$$

The agent thus shifts from the nominal choice (a_1) to the safer, less congested alternative (a_2), demonstrating how the adjustment mechanism enables adaptive rerouting.

Batch-Sequential Routing. To further mitigate collective congestion, vehicles are not routed all at once. Instead, they are processed in batches of size B, assigned in the order they appear in the vehicle list. After each batch is routed, the edge-usage counters are updated and the traffic penalties are recalculated. Subsequent batches therefore "see" the congestion created by earlier batches and are steered away from overloaded corridors. Because the adjustment operates on aggregated edge counts rather than individual vehicle identities, the routing outcome is insensitive to the ordering of vehicles within each batch. This batch-sequential strategy approximates the temporal staggering of real evacuations and is formalised in Algorithm 2.

Algorithm 2 Batch-Sequential Traffic-Aware Routing

Require: Trained Q-tables $\{Q_v\}$; vehicle list $\mathcal{V} = \{(o_i, d_i)\}_{i=1}^{M}$; batch size B
Ensure: Route set $\mathcal{P} = \{P_1, \ldots, P_M\}$
Clear all edge-usage counters: $c(e) \leftarrow 0 \ \ \forall\, e \in E$
for $j = 0,\ B,\ 2B,\ \ldots$ **do**
  for each vehicle (o_i, d_i) in batch j **do**
    $P_i \leftarrow \textsc{GreedyRoute}(o_i, d_i, \{Q_v\}, V_f)$ using Eq. (6)
  end for
  Update c using all successfully routed paths $\{P_i\}$
  Push updated c to all agents
end for
return $\mathcal{P}$

3.3 Evacuation Deployment and Evaluation

3.3.1 Route Generation. Once the Q-tables are trained and the edge-usage counters are initialised, the framework routes all vehicles from their parcel-derived origins to their assigned exit nodes. Each vehicle is forwarded hop by hop: at every intermediate intersection, the resident agent selects the best next-hop according to the traffic-adjusted Q-values (Equation (6)). Fire nodes and previously visited nodes are excluded from the action space to guarantee safety and loop prevention.

3.3.2 Baseline Comparison. To quantify the benefit of multi-agent coordination and traffic awareness, the proposed MARL framework is compared against two conventional routing strategies:

- **Shortest Path (SP):** Dijkstra's algorithm on the weighted graph G, ignoring fire zones and congestion. This represents the default behaviour of most navigation systems.

- **Fire-Aware Shortest Path (FASP):** Dijkstra's algorithm with all fire nodes and their incident edges removed from G, yielding the shortest safe path. This isolates the effect of hazard avoidance without congestion management.

3.3.3 Performance Metrics. To evaluate the effectiveness of the routing strategies objectively, we formally define the following performance metrics. Let $\mathcal{P} = \{P_1, P_2, \ldots, P_M\}$ be the set of generated routes for all M vehicles, where each route P_i is a sequence of edges. Let $c(e)$ denote the final number of vehicles assigned to edge $e \in E$, and let $E_{\text{active}} = \{e \in E \mid c(e) > 0\}$ be the subset of edges carrying at least one vehicle.

- **Peak Congestion (C_{peak}):** The maximum vehicle load on any single edge in the network, $C_{\text{peak}} = \max_{e \in E} c(e)$. Lower peak congestion indicates successfully mitigated bottlenecks.

- **Mean Active Congestion (C_{mean}):** The average vehicle load across all utilised edges, $C_{\text{mean}} = \frac{1}{|E_{\text{active}}|} \sum_{e \in E_{\text{active}}} c(e)$. This measures the typical traffic density on active evacuation corridors.

- **Network Utilisation (U):** The proportion of the road network leveraged during the evacuation, $U = \frac{|E_{\text{active}}|}{|E|} \times 100\%$. Higher utilisation demonstrates better spatial distribution of traffic.

- **Safety Rate (S):** The proportion of routes that successfully avoid the fire zone entirely. A route P_i is safe if it contains no nodes from V_f. The safety rate is $S = \frac{1}{M} \sum_{i=1}^{M} \mathbb{I}(P_i \cap V_f = \emptyset)$, where $\mathbb{I}(\cdot)$ is the indicator function.

- **Mean Route Length (L_{mean}):** The average travel distance per vehicle, $L_{\text{mean}} = \frac{1}{M} \sum_{i=1}^{M} \sum_{e \in P_i} w(e)$, where $w(e)$ is the edge length in metres. Shortest-path baselines minimise this metric at the expense of congestion.

Table 1 summarises the default hyperparameter settings used throughout the experiments.

Table 1. Default hyperparameter settings for the MARL framework.

Parameter	Symbol	Value
Learning rate	α	0.1
Discount factor	γ	0.9
Initial exploration rate	ε_0	1.0
Minimum exploration rate	$\varepsilon_{\min}$	0.01
Exploration decay factor	λ	0.995
Training episodes	N	3 000
Max steps per episode	$T_{\max}$	100
Destination reward	R_{dest}	$+100$
Step penalty	R_{step}	-1
Fire penalty	R_{fire}	-200
High-traffic penalty	–	-80
Medium-traffic penalty	–	-40
Revisit penalty	R_{revisit}	-10
Progress bonus	R_{progress}	$+2$
High-traffic threshold	τ_{high}	2 vehicles
Medium-traffic threshold	τ_{med}	1 vehicle
Traffic weight (deployment)	λ_{traffic}	10
Fire-risk weight (deployment)	λ_{fire}	20
Routing batch size	B	10

4. EXPERIMENTS AND RESULTS

This section presents a systematic empirical evaluation of the proposed congestion-aware MARL framework applied to a real-world wildfire evacuation scenario. The experiments are organised into five analyses: (i) characterisation of the study area and dataset (Section 4.1); (ii) verification that Q-learning agents converge to stable, hazard-avoiding policies (Section 4.2); (iii) quantitative comparison of routing performance against the SP and FASP baselines using the metrics defined in Section 3.3 (Section 4.3); (iv) scalability assessment under increasing fleet demand (Section 4.4); and (v) sensitivity analysis of the deployment time hyperparameters λ_{traffic} and B (Section 4.5). All experiments use the hyperparameter settings listed in Table 1 unless explicitly stated otherwise.

4.1 Study Area and Dataset

The framework is evaluated on the village of Lytton, British Columbia, a small rural community situated at the confluence of the Fraser and Thompson rivers. Lytton is an operationally relevant and well-documented test case: in June 2021, the village recorded a national temperature record of 49.6°C before a wildfire engulfed approximately 90% of the community within 15 minutes, displacing the entire population. The road network follows a constrained, semi-linear highway corridor with two primary egress routes, Highway 1 northbound and Highway 12 southbound, creating a natural bottleneck topology that is representative of many single-corridor communities in mountainous terrain.

The road graph $G = (V, E, w)$ was constructed from OpenStreetMap via OSMnx as described in Section 3.1. Residential building footprints were extracted and snapped to the nearest graph nodes to generate the vehicle origin set $\{v_i^*\}$. A set of fire-hazard nodes V_f was defined based on post-event burn perimeters, and the two highway junctions serve as the exit node set D. Table 2 summarises the key structural properties of the resulting network and Figure 3 shows the the map of study area.

Table 2. Structural characteristics of the Lytton road network.

Property	Value
Nodes ($\lvert V\rvert$)	492
Edges ($\lvert E\rvert$)	681
Residential parcels (M)	42
Exit nodes ($\lvert D\rvert$)	2
Fire-hazard nodes ($\lvert V_f\rvert$)	8
Network topology	Semi-linear corridor

4.2 Training Convergence

Before evaluating deployment-phase routing performance, it is necessary to confirm that the distributed Q-learning agents converge to stable, hazard-avoiding policies. Figure 4 plots the mean cumulative reward per episode and the routing success rate (proportion of episodes in which the vehicle reaches an exit without entering the fire zone) over $N = 3{,}000$ training episodes.

Two convergence phases are visible in Figure 4. During the first $\sim$700 episodes, the total reward rises steeply from near zero to approximately 100, indicating that agents rapidly learn to reach exit nodes and collect the terminal reward ($R_{\text{dest}} = +100$). The reward then stabilises, with the light-blue envelope showing episode-level variance caused by the residual exploration rate

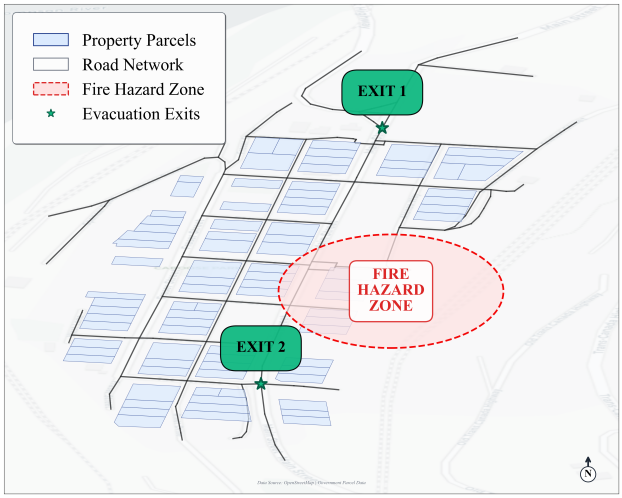

Figure 3. Lytton, BC study area showing the road network, residential parcel origins, fire-hazard zone, and designated exit nodes.

($\varepsilon_{\min} = 0.01$) occasionally steering vehicles into fire zones or dead ends. The cumulative success rate (right panel) follows a logarithmic trajectory, climbing quickly during early training and approaching ∼90% by episode 3,000. The rate does not reach 100% because the ε-greedy policy retains a small random-action probability throughout training, which periodically triggers fire-zone entries or loops that count as failures. The R_{progress} curriculum signal (Section 3.2.2) contributes meaningfully to convergence speed by providing intermediate gradient information that guides exploration toward exit-directed trajectories before the terminal reward has been fully back-propagated through the temporal-difference chain. The stable, asymptotic behaviour of both curves confirms that the mild non-Markovian action-masking via visit history H (Section 3.2.1) does not impede convergence in practice.

4.3 Routing Performance Comparison

The central evaluation compares the three routing strategies, SP, FASP, and MARL, on the full parcel-derived fleet. Table 3 reports the five performance metrics defined in Section 3.3. Including both SP and FASP as baselines serves to decompose the contributions of the framework: the comparison of SP vs. FASP isolates the effect of fire avoidance alone, while FASP vs. MARL isolates the contribution of congestion management.

Table 3. Routing performance comparison for the Lytton network. Bold values indicate the best result per metric. Percentage changes of MARL relative to FASP are shown in parentheses.

Metric	SP	FASP	MARL
C_{peak} (vehicles)	35	38	**10**
C_{mean} (vehicles)	8.2	10.4	**4.6**
U (%)	6.5	5.6	**10.0**
S (%)	64.3	100	**100**
L_{mean} (m)	**3,856**	4,638	4,982

Several observations follow from these results.

Fire Safety. The SP baseline, which is unaware of the fire zone, routes 35.7% of vehicles through hazardous nodes ($S = 64.3\%$). Both FASP and MARL achieve $S = 100\%$ by construction: FASP removes fire nodes from the graph entirely, while the MARL agents have learned to avoid fire nodes through the severe $R_{\text{fire}} = -200$ penalty during training, reinforced by the λ_{fire} deployment-time adjustment (Equation 6).

Congestion Reduction. The FASP baseline, while safe, concentrates all vehicles onto the shortest fire-free paths, which converge onto a small number of arterial corridors. This yields the highest peak congestion among all three strategies ($C_{\text{peak}} = 38$), even exceeding SP ($C_{\text{peak}} = 35$), because removing fire-zone edges forces previously dispersed traffic onto fewer alternatives. The MARL framework reduces C_{peak} to 10, a 74% reduction relative to FASP, because the batch-sequential deployment procedure (Algorithm 2) progressively raises the $\lambda_{\text{traffic}} \cdot c(v,u)$ penalty on saturated edges, causing subsequent batches to select alternative routes from the agents' pre-learned Q-value landscape. The increase in network utilisation from $U = 5.6\%$ (FASP) to $U = 10.0\%$ (MARL) confirms that the framework actively redistributes traffic onto secondary roads that the deterministic baselines ignore entirely.

Route Length Trade-off. The MARL framework produces a mean route length of $L_{\text{mean}} = 4{,}982$ m, which is 7.4% longer than the FASP baseline (4,638 m). This increase is a direct and expected consequence of congestion-aware rerouting: some vehicles are deliberately directed onto longer but less congested paths to alleviate bottlenecks elsewhere in the network. In the context of emergency evacuation, a sub-400 m average detour is operationally negligible compared to the 74% reduction in peak congestion, which directly correlates with time delays, queuing, and exposure to advancing hazards.

Spatial Flow Distribution. To complement the aggregate metrics, Figure 5 presents congestion heatmaps that visualise the spatial distribution of edge-level vehicle loads for the baseline and MARL strategies.

The contrast between the two panels is striking. Under baseline routing (top), traffic is funnelled onto a small number of corridors leading to the two exits, with the most heavily loaded edge carrying up to 35 vehicles. Several of these saturated corridors pass directly adjacent to or through the fire-hazard zone, exposing evacuees to unnecessary risk. Under MARL routing (bottom), traffic is visibly dispersed across a wider set of edges: more edges appear in the green–yellow range (1–5 vehicles each), while the deep red saturation visible in the baseline is largely eliminated. The fire-hazard zone is cleanly avoided, with no coloured edges entering the dashed perimeter. This lateral spread of traffic is a direct consequence of the $\lambda_{\text{traffic}} \cdot c(v,u)$ penalty in Equation 6: as the batch-sequential deployment proceeds, successive batches are progressively steered away from the corridors that earlier batches have already loaded, distributing flow onto secondary and tertiary roads. The visual evidence corroborates the quantitative reductions in C_{peak} and the increase in U reported in Table 3.

4.4 Scalability Under Increasing Fleet Demand

In practice, the number of evacuating vehicles depends on factors such as compliance rates and household size that are difficult to predict in advance. To assess the framework's robustness to demand uncertainty, the routing experiment is repeated with fleet sizes ranging from 50% to 200% of the nominal parcel count M. Figure 6 plots C_{peak} as a function of fleet size for all three strategies.

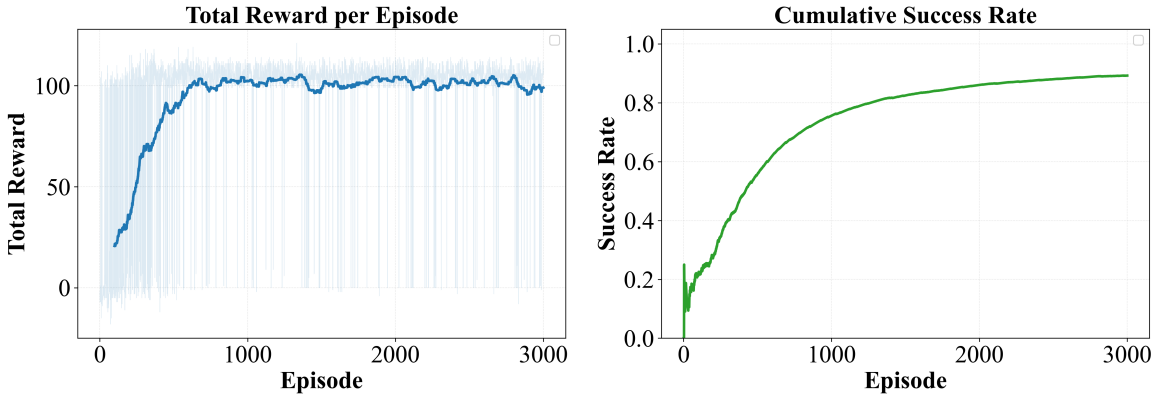

Figure 4. Training convergence over $N = 3{,}000$ episodes. Left: total reward per episode (raw values in light blue; smoothed trend in dark blue). Right: cumulative routing success rate. The reward stabilises near the terminal reward ($R_{\text{dest}} = +100$) by approximately episode 700, while the success rate rises to ∼90% following a logarithmic trajectory.

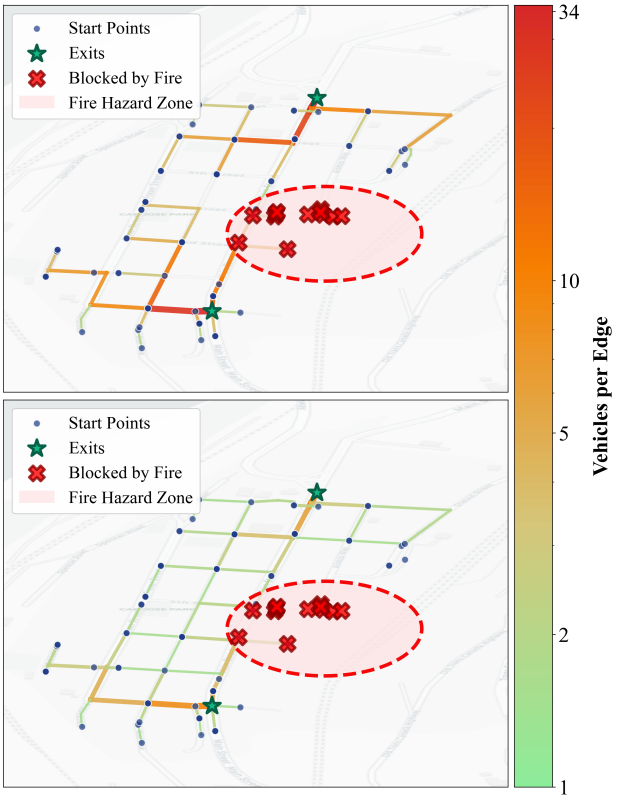

Figure 5. Edge-level congestion heatmaps for the Lytton network. Top: baseline (fastest-path) routing. Bottom: MARL routing. Edge colour intensity is proportional to the vehicle count $c(e)$, with the shared colour bar ranging from 1 (green) to 35 (red) vehicles per edge. Blue dots denote vehicle start points (v_i^*), green stars denote exit nodes (D), and red crosses mark fire-blocked nodes (V_f) within the dashed fire-hazard zone.

Both SP and FASP exhibit approximately linear growth in C_{peak}: because every vehicle independently selects the same deterministic shortest (or shortest-safe) path, each additional vehicle increments the count on the same critical edge. The MARL framework, by contrast, exhibits sub-linear scaling. At low fleet sizes, vehicles encounter minimal congestion and the MARL routing closely mirrors the FASP trajectories. As the fleet grows and edge-usage counters begin to exceed the penalty thresholds (τ_{med}, τ_{high}), the batch-sequential feedback loop activates the

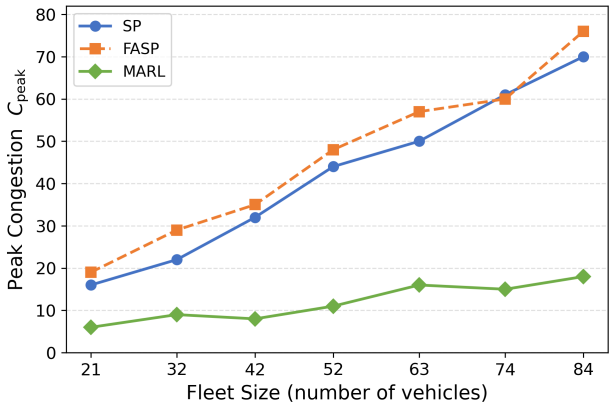

Figure 6. Peak congestion (C_{peak}) as a function of fleet size for the Lytton network. SP and FASP exhibit approximately linear growth, while MARL scales sub-linearly owing to the batch-sequential feedback mechanism. The performance gap widens with increasing demand.

rerouting mechanism: each successive batch of vehicles "sees" the congestion generated by earlier batches and shifts onto the alternative corridors encoded in the Q-value landscape. The result is a progressively flattening congestion curve. The absolute performance gap between MARL and the baselines therefore widens with increasing demand, confirming that the framework provides the greatest benefit precisely in the most operationally critical high-demand scenarios.

4.5 Sensitivity Analysis

Two deployment-time hyperparameters govern the behaviour of the traffic-aware adjustment (Equation 6): the traffic sensitivity weight λ_{traffic} and the routing batch size B. This section examines the sensitivity of the routing outcomes to variations in each parameter.

4.5.1 Effect of the Traffic Sensitivity Weight. The parameter λ_{traffic} controls the magnitude of the per-vehicle congestion penalty applied during deployment. To understand its effect, we fix $B = 10$ and vary $\lambda_{\text{traffic}} \in \{0, 2, 5, 10, 20, 50\}$, recording C_{peak}, U, and L_{mean}. Figure 7 plots the results.

At $\lambda_{\text{traffic}} = 0$, the deployment-time adjustment is disabled and the system reduces to a fire-aware greedy policy over the nominal Q-values, producing congestion levels comparable to FASP.

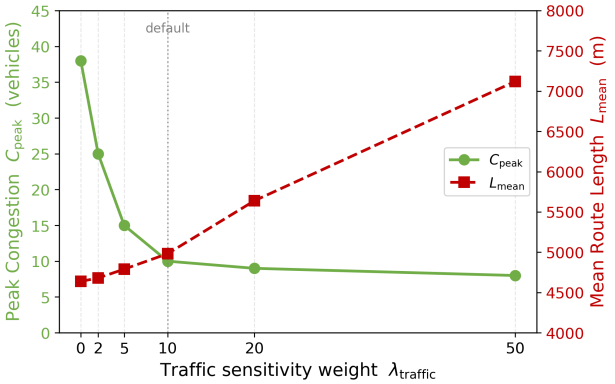

Figure 7. Effect of λ_{traffic} on peak congestion C_{peak} (left axis, green) and mean route length L_{mean} (right axis, red) for the Lytton network. The dotted vertical line marks the default value $\lambda_{\text{traffic}} = 10$.

As λ_{traffic} increases toward the default value of 10, C_{peak} declines steadily while L_{mean} increases only modestly, indicating that the agents are successfully shifting traffic onto alternative corridors without incurring excessive detour costs. Beyond $\lambda_{\text{traffic}} \approx 20$, however, diminishing returns set in: C_{peak} stabilises while L_{mean} rises sharply, as the agents begin aggressively avoiding edges that carry only one or two vehicles and forcing vehicles into unnecessarily long routes. The default value of $\lambda_{\text{traffic}} = 10$ sits at the inflexion point of this trade-off, providing the largest congestion reduction per unit of additional travel distance.

4.5.2 Effect of the Routing Batch Size. The batch size B determines how frequently edge-usage counters are updated during deployment (Algorithm 2). A smaller B provides more frequent feedback but increases the computational overhead of counter updates and path aggregation. Figure 8 examines $B \in \{1, 5, 10, 25, M\}$, where $B = M$ corresponds to routing all vehicles simultaneously (i.e., a single batch with no intermediate counter updates).

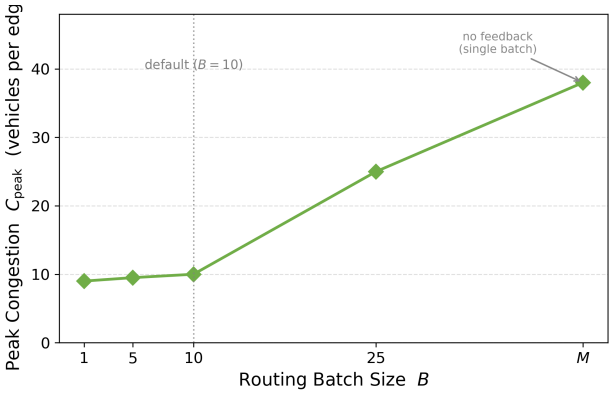

Figure 8. Effect of batch size B on peak congestion C_{peak} for the Lytton network. At $B = M$ (single batch), the system receives no congestion feedback and reverts to FASP-like performance. Diminishing returns are evident below $B \approx 10$.

When $B = M$ (single batch), the system has no opportunity to observe emergent congestion, and all vehicles are routed based solely on the nominal Q-values. This eliminates the traffic-aware feedback loop and reverts behaviour to the FASP-like baseline. As B decreases, agents receive progressively higher-resolution traffic feedback, yielding steadily lower C_{peak}. Notably, the marginal improvement diminishes below $B \approx 10$: at this point the feedback granularity is sufficient to capture the dominant congestion patterns, and further refinement yields only marginal gains while increasing the number of counter-update cycles. The default value of $B = 10$ therefore represents an effective balance between congestion awareness and computational efficiency. Additionally, the batch model is operationally realistic: in a real evacuation, vehicles depart in temporal waves rather than simultaneously, and the batch abstraction mirrors this staggered departure pattern.

5. CONCLUSION

This study presented a congestion-aware multi-agent reinforcement learning framework for wildfire evacuation routing that incorporates parcel-level demand, distributed decision-making, and real-time congestion and hazard penalties. Applied to Lytton, the framework reduced peak congestion by 74% (from 38 to 10 vehicles per edge) relative to fire-aware shortest-path routing, achieved complete fire-zone avoidance ($S = 100\%$ versus 64.3% for conventional shortest paths), and increased network utilisation by 79%, all at a cost of only 7.4% additional mean travel distance. Sensitivity analyses confirmed that the default hyperparameters ($\lambda_{\text{traffic}} = 10$, $B = 10$) sit at effective operating points, and the sub-linear scalability of peak congestion under increasing fleet demand indicates that the framework provides the greatest benefit in the most operationally critical scenarios. These results demonstrate that distributed RL policies yield safer, more balanced evacuation flows and offer a foundation for integration with dynamic fire-spread data and real-time traffic monitoring.

References

Boeing, G., 2017. OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers, environment and urban systems*, 65, 126–139.

Cova, T. J., Johnson, J. P., 2003. A network flow model for lane-based evacuation routing. *Transportation research part A: Policy and Practice*, 37(7), 579–604.

de Oliveira e Silva, R. A., Cui, G., Rahimi, S. M., Wang, X., 2022. Personalized route recommendation through historical travel behavior analysis. *GeoInformatica*, 26(3), 505–540.

Ekaputra, R. A., Kee, S.-H., Yee, J.-J., 2024. Optimizing Urban-Scale evacuation strategies through disaster victim aggregation modification. *IEEE Access*, 12, 73581–73598.

Gai, W.-m., Deng, Y.-f., Li, J., Du, Y., Ye, F.-q., 2014. A bi-objective optimization problem about rescue route during disaster time. *Proceedings of the 33rd Chinese Control Conference*, IEEE, 8906–8910.

Gu, Y. et al., 2023. A metaverse-based teaching building evacuation training system with deep reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 53(4), 2209–2219.

Guman, J., O'Brien, J. C., Pondoc, C., Kochenderfer, M. J., 2024. PyroRL: A reinforcement learning environment for wildfire evacuation. *Journal of Open Source Software*, 9(101), 6739.

Hamacher, H. W., Tjandra, S. A., 2001. Mathematical modelling of evacuation problems: A state of art. Report, Fraunhofer-Institut für Techno- und Wirtschaftsmathematik ITWM.

Jain, A., 2025. AI-Based Route Optimization in Urban Public Transport Networks. *International Journal of Advanced Research in Computer Science and Engineering (IJARCSE)*, 1(1), 30–37.

Jain, P., Barber, Q. E., Taylor, S. W., Whitman, E., Castellanos Acuna, D., Boulanger, Y., Chavardès, R. D., Chen, J., Englefield, P., Flannigan, M. et al., 2024. Drivers and impacts of the record-breaking 2023 wildfire season in Canada. *Nature Communications*, 15(1), 6764.

Jain, P., Castellanos-Acuna, D., Coogan, S. C., Abatzoglou, J. T., Flannigan, M. D., 2022. Observed increases in extreme fire weather driven by atmospheric humidity and temperature. *Nature Climate Change*, 12(1), 63–70.

Li, J. et al., 2023. A reinforcement learning-based routing algorithm for large street networks. *International Journal of Geographical Information Science*, 37(12), 2562–2588.

Link, E. D., Maranghides, A., 2023. Burnover events identified during the 2018 Camp Fire. *International journal of wildland fire*, 32(6), 989–997.

Lopez, P. A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.-P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., Wießner, E., 2018. Microscopic traffic simulation using sumo. *2018 21st international conference on intelligent transportation systems (ITSC)*, Ieee, 2575–2582.

Mamuji, A. A., Rozdilsky, J. L., 2019. Wildfire as an increasingly common natural disaster facing Canada: understanding the 2016 Fort McMurray wildfire. *Natural Hazards*, 98(1), 163–180.

Mas, E. et al., 2024. Reinforcement learning-based tsunami evacuation guidance system. *International Journal of Disaster Risk Reduction*, 111, 105023.

McEvoy, A., Nielsen-Pincus, M., Holz, A., Catalano, A. J., Gleason, K. E., 2020. Projected impact of mid-21st century climate change on wildfire hazard in a major urban watershed outside Portland, Oregon USA. *Fire*, 3(4), 70.

Raei, B., Kinateder, M., Benichou, N., Gomaa, I., Wang, X., 2025. Are the data good enough? Spatial and temporal modeling of evacuee behavior using GPS data in a small rural community. *International Journal of Disaster Risk Reduction*, 116, 105054.

Raei, B., Sen, A., Kinateder, M., Bénichou, N., Wang, X., 2026. Optimizing evacuation routes for human mobility during wildfires: A case study of the 2023 McDougall Creek Wildfire. *Transportation Research Part D: Transport and Environment*, 153, 105216.

Rahayuda, I., Santiari, N., 2021. Dijkstra and bidirectional dijkstra on determining evacuation routes. *Journal of Physics: Conference Series*, 1803, IOP Publishing, 012018.

Rahimi, S. M., Far, B., Wang, X., 2020. Behavior-based location recommendation on location-based social networks. *GeoInformatica*, 24(3), 477–504.

Roughgarden, T., 2005. *Selfish routing and the price of anarchy*. MIT press.

Saadatseresht, M., Mansourian, A., Taleai, M., 2009. Evacuation planning using multiobjective evolutionary optimization approach. *European journal of operational research*, 198(1), 305–314.

Safarzadeh, R., Wang, X., 2024. Map matching on low sampling rate trajectories through deep inverse reinforcement learning and multi-intention modeling. *International Journal of Geographical Information Science*, 38(12), 2648–2683.

Safarzadeh, R., Wang, X., Raei, B., 2025. Chargenav: End-to-end fleet optimization and energy aware navigation for electric trucks. *Proceedings of the 19th International Symposium on Spatial and Temporal Data*, 247–251.

Takabatake, T. et al., 2025. Reinforcement learning-based optimization of tsunami evacuation paths: effectiveness and robustness in two coastal areas in Japan. *Reliability Engineering & System Safety*, 257, 111594.

Tang, H. et al., 2024. Strategizing equitable transit evacuations: A data-driven reinforcement learning approach. *arXiv preprint arXiv:2412.05777*.

Udhan, P., Ganeshkar, A., Murugesan, P., Permani, A. R., Sanjeeva, S., Deshpande, P., 2022. Vehicle route planning using dynamically weighted Dijkstra's algorithm with traffic prediction. *arXiv preprint arXiv:2205.15190*.

W Axhausen, K., Horni, A., Nagel, K., 2016. *The multi-agent transport simulation MATSim*. Ubiquity Press.

Wei, H., Zheng, G., Gayah, V., Li, Z., 2021. Recent advances in reinforcement learning for traffic signal control: A survey of models and evaluation. *ACM SIGKDD explorations newsletter*, 22(2), 12–18.

Wotton, B. M., Flannigan, M. D., Marshall, G. A., 2017. Potential climate change impacts on fire intensity and key wildfire suppression thresholds in Canada. *Environmental Research Letters*, 12(9), 095003.

Yang, X. et al., 2025. Deep reinforcement learning for real-time airport emergency evacuation using A3C. *Mathematics*, 13(14), 2269.

Zhang, L. et al., 2023. Deep reinforcement learning and 3D physical environments applied to crowd evacuation in congested scenarios. *International Journal of Digital Earth*, 16(1), 1301–1321.

Zhao, X., Xu, Y., Lovreglio, R., Kuligowski, E., Nilsson, D., Cova, T. J., Wu, A., Yan, X., 2022. Estimating wildfire evacuation decision and departure timing using large-scale GPS data. *Transportation research part D: transport and environment*, 107, 103277.

Zhong, Z., Ren, Z., 2026. Multi-Agent Reinforcement Learning Optimization for Urban Earthquake Emergency Evacuation With GIS-Based Real-Time Decision Support. *Journal of Earthquake and Tsunami*.

Zhou, Y., Yang, Y., Liu, D., Liu, Y., Namilae, S., Song, H., 2024. Real-time Deep Reinforcement Learning for Evacuation under Emergencies.

Zhu, Y., Li, H., Wang, Z., Li, Q., Dou, Z., Xie, W., Zhang, Z., Wang, R., Nie, W., 2022. Optimal evacuation route planning of urban personnel at different risk levels of flood disasters based on the improved 3D Dijkstra's algorithm. *Sustainability*, 14(16), 10250.