

Article

Multi-Modal Collaborative Evacuation During Mass Gatherings via Distributional Reinforcement Learning

Wensi Wang ¹, Xiangsen Xu ¹, Liangmu Hou ¹ and Bin Yu ^{2,*}

¹ College of Transportation Engineering, Dalian Maritime University, Dalian 116026, China; wws@dmlu.edu.cn (W.W.)

² School of Transportation Science and Engineering, Beihang University, Beijing 100191, China

* Correspondence: yuybbincc@163.com

Highlights

Please indicate how your work links to systems science via your contributions to systems practice, theory, and/or methodology.

- This study develops a two-layer stochastic optimization framework integrating strategic fleet sizing with real-time dispatching, contributing a unified decision architecture for multi-modal evacuation management.
- A distributional reinforcement learning algorithm with categorical value representation is introduced to model return distributions under travel time uncertainty, advancing risk-aware policy learning in stochastic transportation networks.

What are the main findings and/or the implications of the main findings?

- The proposed framework reduces evacuation completion time by 31.5% and passenger waiting time by 48.6% compared to shuttle-only operations, with only 4.0% performance degradation under increasing travel time uncertainty.
- Online decisions are executed in approximately 3.25 s per step, well within the 2-min decision interval, supporting practical deployability in real-time traffic management.

Abstract

Large-scale public events generate concentrated passenger demand during egress periods, often overwhelming urban transit systems. This paper proposes a multi-modal evacuation framework that coordinates in-service buses temporarily diverted from existing lines and dedicated shuttle vehicles pre-positioned at depots. The problem is formulated as a two-layer stochastic optimization under travel time uncertainty: the upper layer determines pre-event shuttle fleet sizing, while the lower layer makes real-time dispatching decisions for both modes. We propose an Uncertainty-Aware Reinforcement Learning framework with Categorical DQN (UARL-CD) that learns a robust dispatching policy through a reward function aligned with the lower-level objective, explicitly accounting for travel time uncertainty via distributional value representation and stochastic training, with an action masking mechanism enforcing operational constraints. Simulation experiments based on a realistic stadium evacuation scenario demonstrate that the proposed framework significantly outperforms deterministic optimization and rule-based strategies, achieving a 31.6% reduction in evacuation completion time and a 48.4% reduction in average passenger waiting time compared to shuttles alone, while maintaining robustness to travel time uncertainty with only 4.0% performance degradation and online decisions executed within the 2-min decision interval.



Academic Editor: Emilio Larrodé

Received: 8 August 2026

Revised: 4 September 2026

Accepted: 7 September 2026

Published: 11 September 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 \(CC BY\)](#) license.

Keywords: multi-modal evacuation; reinforcement learning; categorical DQN; travel time uncertainty

1. Introduction

Large-scale public events, such as sports tournaments, concerts, and festivals, generate concentrated passenger demand during egress periods, often overwhelming existing transportation infrastructure. Thousands of attendees simultaneously leaving a venue create severe congestion at exits and surrounding road networks, while regular transit services operating on fixed schedules and routes are ill-equipped to handle such sporadic, high-intensity demand peaks. Dedicated shuttle fleets offer a potential solution but require substantial capital investment and are constrained by depot capacity. Moreover, passengers have heterogeneous destination preferences. Some need to reach nearby metro or rail hubs, others require access to ride-hailing pickup zones or parking facilities. A single mode cannot efficiently serve this diversity of travel needs.

In this context, a multi-modal evacuation strategy offers a superior alternative. By coordinating regular buses diverted from existing lines and dedicated shuttle vehicles pre-positioned at depots, the system can simultaneously serve passengers with different destination requirements. Buses, with their large capacity and extensive route coverage, handle the initial surge from pickup points, while shuttles provide direct, high-quality connections to major hubs, such as metro stations and ride-hailing pickup zones, bypassing intermediate stops and reducing overall travel time. This synergy makes more efficient use of available resources.

However, this multi-modal coordination introduces substantial operational complexity: decisions across modes are interlinked, strategic fleet sizing interacts with real-time dispatching, and travel times are inherently uncertain. The operator must continuously decide which bus or shuttle to dispatch, to which pickup point, for how long to dwell, and—for shuttles—to which hub to connect, all while balancing evacuation efficiency against regular service disruption and operating costs. The combination of multi-modal coordination, two-layer decision making (fleet sizing and real-time dispatch), and operational uncertainty arising from travel times renders the problem highly challenging from an optimization standpoint [1].

To address these challenges, we propose an Uncertainty-Aware Reinforcement Learning (UARL) framework that integrates a two-layer optimisation structure with a distributional reinforcement learning algorithm for real-time decision making. The upper layer solves the pre-event fleet sizing problem using sample average approximation, evaluating candidate shuttle counts through repeated simulations of the lower layer. The lower layer is formulated as a Markov Decision Process and solved using a Categorical DQN (UARL-CD) algorithm, which learns a dispatching policy that coordinates both buses and shuttles simultaneously. The contributions of our approach are:

First, we propose a unified two-layer optimization framework that coordinates multi-modal evacuation systems—integrating in-service buses temporarily diverted from existing lines and dedicated shuttle vehicles pre-positioned at depots—within a single decision hierarchy that couples pre-event strategic fleet sizing with real-time tactical dispatching.

Second, we develop a dynamic stochastic evolution model that tracks real-time vehicle positions, passenger loads, waiting queues at pickup points and regular stations, and hub states (metro train arrival times and ride-hailing vehicle supply rates) under uncertain travel times. This model captures the full spectrum of multi-modal coordination, enabling

synergistic interactions not only between buses and shuttles but also with downstream metro and ride-hailing services.

Third, we introduce a novel UARL-CD algorithm that learns robust, real-time dispatching policies. Unlike standard value-based methods that estimate only expected Q-values, our approach learns the full return distribution under travel time uncertainty, enabling the agent to account for variance and tail risks.

Fourth, a constraint-aware action masking mechanism is further incorporated to enforce operational constraints as hard constraints during both training and execution, ensuring feasible decisions at all times.

This paper focuses on a typical yet highly challenging class of evacuation problems: single-venue mass gatherings with a radial multi-modal feeder structure—a scenario commonly encountered in professional football matches, large-scale concerts, and regional sporting events. This class is characterized by concentrated demand surges, short decision intervals, and severe multi-modal coupling. While larger multi-node urban networks and diverse event types present additional complexities, establishing a rigorous and validated solution for this representative class serves as an essential methodological benchmark.

The remainder of this paper is organized as follows. Section 2 reviews the related literature. Section 3 defines the problem. Section 4 presents the system dynamics, including vehicle movements, passenger states, hub operations, constraints, and the objective function. Section 5 describes the UARL-CD algorithm. Section 6 reports the numerical experiments and results. Section 7 concludes the paper and discusses directions for future work.

2. Literature Review

Vehicle scheduling in the context of mega-event evacuation has drawn increasing scholarly interest. Teng et al. (2011) constructed a mixed-integer programming model incorporating multiple vehicle types to reduce the fleet size required for temporary shuttle operations [2]. Tian et al. (2023) introduced a two-phase planning framework that integrates data-driven techniques with demand-responsive strategies for designing tailored bus services in planned special events [3]. Santanam et al. (2024) developed ridership forecasting models for public transit systems during special events and combined them with optimized train timetables [4]. Yin et al. (2024) formulated a two-stage recoverable robust optimization model addressing joint location-allocation and evacuation planning under uncertainties in both supply and demand [5].

The integration of multiple transport modes in evacuation planning has also been extensively examined. Su et al. (2024) proposed a dual-mode system that coordinates buses with e-hailing services, incorporating a geofencing-based evacuation approach to shorten clearance times [6]. Jiang et al. (2022) developed a macro-level optimization model for large-scale multi-modal evacuation that coordinates pedestrian movement with metro, bus, and taxi systems, achieving notable improvements over single-mode alternatives [7]. Zhang et al. (2019) addressed a dynamic system-optimal multi-modal traffic evacuation problem using a mixed-integer linear programming formulation that accounts for private vehicles, emergency buses, and rail transit, and employed a Lagrangian relaxation-based heuristic for solution decomposition [8].

Flexible bus scheduling strategies have emerged as a promising avenue for improving operational performance. Kliewer et al. (2006) devised a time-space network for the multi-depot vehicle scheduling problem that enables vehicles to cover multiple routes through coordinated bus service planning [9]. Gu et al. (2018) put forward a strategy for dynamically assigning buses to bridging routes during metro service disruptions [10]. Li et al. (2019) extended the mixed-fleet bus scheduling framework by incorporating refueling station placement, simultaneously considering service delivery, passenger mobility, and emission

objectives [11]. Liu et al. (2024) investigated the scheduling of public transport vehicles with flexible departure times, aiming to minimize both fleet size and passenger schedule deviations [12]. Li et al. (2024) tackled the multi-route bus fleet allocation problem using a combination of distributionally robust optimization and reinforcement learning [13].

Uncertainty modeling plays a vital role in evacuation planning, given the prevalence of data inaccuracies, environmental variability, and measurement errors [14]. Robust optimization and stochastic programming have emerged as two primary methodologies in this domain [15]. Yan and Tang (2008) proposed an integrated framework for intercity bus routing and scheduling that accounts for stochastic travel times [16]. Shen et al. (2016) developed a probabilistic scheduling model based on stochastic trip durations [17]. Chu et al. (2022) designed a two-stage stochastic programming model that combines fixed-route bus services with dial-a-ride transportation under demand uncertainty [18]. However, obtaining reliable probability distributions can be difficult when historical records are scarce. Fuzzy programming has been employed to address epistemic uncertainty [19], and more recently, distributionally robust fuzzy optimization has been introduced to capture ambiguity in possibility distributions [20]. An et al. (2026) advanced this line of research by developing a distributionally robust optimization approach for customized bus scheduling using parametric interval-valued fuzzy variables [21].

Solution techniques for large-scale evacuation problems have also received substantial attention [22,23]. Variable neighborhood search (VNS) and its extensions have been applied to vehicle scheduling problems [24,25]. More recent contributions have integrated reinforcement learning into VNS frameworks to enable adaptive neighborhood selection. Qin et al. (2021) developed a reinforcement learning-based hyper-heuristic that employs distributed proximal policy optimization to orchestrate multiple metaheuristics dynamically [26]. Zhang et al. (2022) proposed a deep reinforcement learning-driven hyper-heuristic using Double Deep Q-Networks for dynamic heuristic selection [22]. An et al. (2026) [21] incorporated the KL-UCB algorithm and sliding window mechanisms into VNS to adaptively choose neighborhood structures, thereby improving computational efficiency. These developments underscore the promise of reinforcement learning-enhanced metaheuristics for tackling complex scheduling problems [21].

Despite significant progress, two critical gaps remain. First, from an evacuation scheduling perspective, strategic fleet sizing and real-time dispatching are almost always treated as separate problems, with pre-event resource deployment optimized independently of operational policies. Moreover, most existing studies coordinate only a subset of transport modes without systematically incorporating the full spectrum of in-service buses, dedicated shuttles, metro systems, and ride-hailing services, along with dynamic hub states that critically affect passenger waiting times. Second, from a reinforcement learning perspective, existing RL-based evacuation studies predominantly rely on standard value-based methods that estimate only expected returns, failing to capture the variance and tail risks inherent in stochastic travel times. To address these gaps, we propose an UARL framework that integrates strategic fleet sizing with real-time dispatching, explicitly accounts for travel time uncertainty through distributional value representation, coordinates multiple modes with real-time hub states, and enforces operational constraints via action masking to ensure feasible decisions at all times.

3. Problem Description

This study addresses the passenger evacuation problem from large-scale event venues under uncertain travel times. Figure 1 presents a conceptual illustration of the multi-modal evacuation scheme. The venue is located at the center, surrounded by three types of facilities: pickup points (blue squares) where passengers board, regular bus stations (gray

circles) on existing transit lines, and depots (larger gray circles) where shuttle vehicles are pre-positioned before the event. Two transport modes are coordinated: (i) in-service buses (green lines) are temporarily diverted from their regular bus lines (black dashed lines) to serve pickup points, then return to their original routes after dropping off evacuees; and (ii) dedicated shuttle vehicles (red dashed lines) travel from depots to pickup points, transport passengers to nearby hubs (metro stations or ride-hailing zones, yellow circles), and then return to depots for re-dispatch. Once passengers arrive at hubs, they continue their onward journeys via metro lines or ride-hailing services.

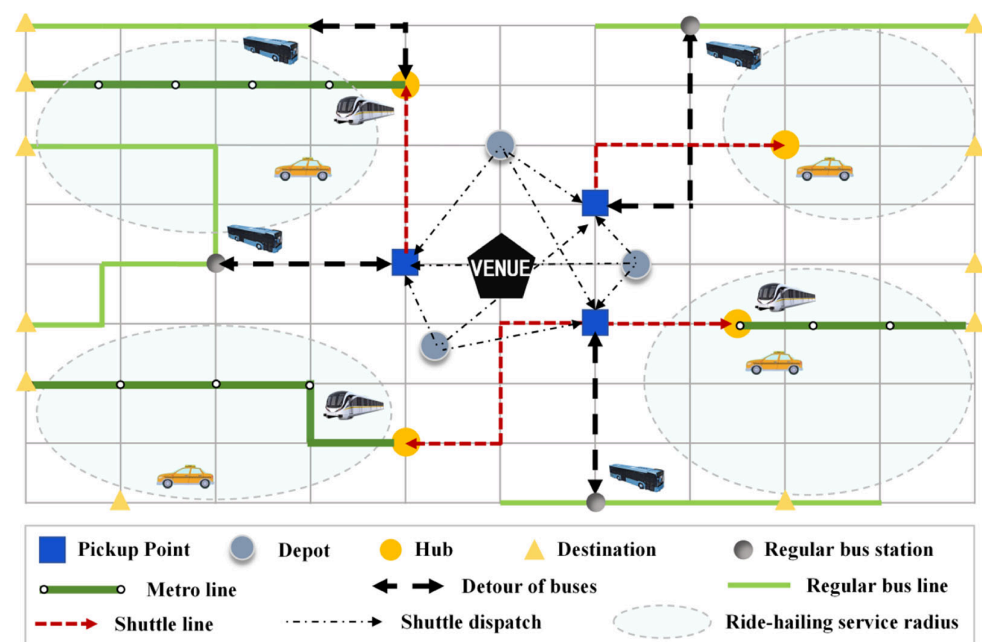


Figure 1. An illustration example of multi-mode evacuation scheme.

It is important to clarify the operational boundary conditions of this study. To ensure the efficiency of the multi-modal shuttle system and to eliminate the confounding effects of private vehicle interference, we assume that a temporary traffic restriction zone is enforced around the venue during the egress period. All non-authorized private cars are prohibited from entering the core area. Furthermore, the travel time parameters used throughout this study are calibrated to reflect congested urban conditions typical of stadium egress periods—accounting for junction delays, turning maneuvers, and reduced speeds due to dense traffic flow within the restricted zone. This ensures that while congestion propagation is eliminated through traffic management, the actual operating speeds remain realistic for post-event scenarios.

The problem is formulated as a two-layer stochastic optimization. The upper layer (pre-event) determines the number of shuttle vehicles R_d to pre-position at each depot $d \in \mathcal{D}$, balancing rental costs against expected operational performance. The lower layer (real-time) makes dynamic dispatching decisions at each decision epoch $t \in \mathcal{T}$: which bus or shuttle to dispatch, to which pickup point, for how long to dwell, and (for shuttles) to which hub to connect. The lower-level objective minimizes the expected total cost over the evacuation horizon, comprising evacuee waiting time, regular passenger disruption, dwell time penalties, and hub waiting time.

The problem is challenging due to three factors. First, the decisions concerning buses and shuttles are coupled, since sending a bus to a pickup point diminishes the need for shuttles while potentially disrupting regular service. Second, strategic fleet sizing and real-time dispatching are coupled. Third, travel times are inherently uncertain. The combination of

multi-modal coordination, two-layer decision making, and operational uncertainty exceeds the capabilities of conventional deterministic optimization. The objective is to learn a dispatching policy that minimizes the weighted sum of all cost components while respecting operational constraints, including vehicle capacity limits, maximum detour/round-trip times, minimum headways, and line dispatch balance.

4. Model Formulation

4.1. Notation

Table 1 summarizes the sets, parameters, and decision variables used in the model formulation.

Table 1. Sets, parameters, and decision variables of the model.

Symbol	Description
Sets and indices	
$\mathcal{L}$	Set of transit lines, indexed by l
$\mathcal{V}_l$	Set of vehicles on line l , indexed by v
$\mathcal{P}$	Set of pickup points near the venue, indexed by p
$\mathcal{N}_l$	Set of stations on line l , indexed by n
$\mathcal{T}$	Set of decision time points, indexed by t
Δt	Time step between decision points
$\mathcal{D}$	Set of depots, indexed by d
$\mathcal{H}$	Set of hubs, indexed by h
$\mathcal{V}^{\text{shuttle}}$	Set of shuttle vehicles
$\mathcal{V}_d^{\text{shuttle}}$	Set of shuttle vehicles pre-positioned at depot d
Parameters	
λ_p	Passenger arrival rate at pickup point p (passengers/min)
$\lambda_{l,n}$	Passenger arrival rate at station n of line l (passengers/min)
$C_{l,v}$	Capacity of vehicle v on line l
C^{shuttle}	Capacity of each shuttle vehicle (passengers)
$Q_{\min}$	Minimum effective capacity for bus dispatch
$T_{\max}^{\text{detour}}$	Maximum allowed bus detour time
$T_{\max}^{\text{shuttle}}$	Maximum allowed shuttle round-trip time
$H_{\min}$	Minimum headway between consecutive buses at same pickup point
$W_{\max}^{\text{normal}}$	Maximum allowed waiting time for regular passengers
ϕ_l	Maximum proportion of vehicles that can be dispatched from line l
v^{norm}	Normal bus operating speed (km/min)
v^{detour}	Bus speed while traveling to pickup points
v^{return}	Bus speed while returning to original route
v^{shuttle}	Shuttle operating speed
$\sigma_t, \sigma_d, \sigma_r, \sigma_{\text{shuttle}}$	Standard deviations of travel time uncertainty for normal, detour, return and shuttle phases
μ_p	Boarding rate at pickup point p
c_{vehicle}	Rental cost per shuttle vehicle
$R_d^{\max}$	Maximum number of shuttles that can be stored at depot d
T_h^{head}	Scheduled metro headway at hub h
C_h^{train}	Capacity of one metro train at hub h
C_h^{ride}	Capacity per ride-hailing vehicle
$\alpha, \beta, \delta, \eta_{\text{sh}}, \theta_{\text{metro}}, \theta_{\text{ride}}$	Weighting coefficients in the lower-level objective function
Bus State Variables	
$p_{l,v}(t)$	Position of bus v on line l at time t
$q_{l,v}(t)$	Passenger load of bus v on line l at time t
$s_{l,v}(t)$	Status of bus v on line l at time t : 0 = available, 1 = pickup, 2 = dwelling, 3 = returning
$\tau_{l,v}(t)$	Assigned pickup point for bus v on line l at time t (if any)
$d_{l,v}(t)$	Accumulated detour time for bus v on line l at time t
$rd_{l,v}(t)$	Remaining dwell time counter for bus v

Table 1. Cont.

Symbol	Description
Shuttle State Variables	
$p_{d,v}^{sh}(t)$	Position of shuttle v from depot d at time t
$q_{d,v}^{sh}(t)$	Passenger load of shuttle v from depot d at time t
$s_{d,v}^{sh}(t)$	Status of shuttle v at time t : 0 = available, 1 = pickup, 2 = dwelling, 3 = returning to hub, 4 = returning to depot
$\tau_{d,v}^{sh}(t)$	Assigned pickup point for shuttle v at time t (if any)
$\omega_{d,v}^{sh}(t)$	Assigned hub destination for shuttle v at time t
$rd_{d,v}^{sh}(t)$	Remaining dwell time counter for shuttle v
Passenger State Variables	
$w_p(t)$	Number of waiting passengers at pickup point p at time t
$w_{l,n}(t)$	Number of waiting passengers at station n of line l at time t
$W_p(t)$	Cumulative waiting time at pickup point p up to time t (passenger-minutes)
$W_{l,n}(t)$	Cumulative waiting time at station n of line l up to time t (passenger-minutes)
$w_h^{metro}(t)$	Number of waiting passengers at metro hub h at time t
$w_h^{ride}(t)$	Number of waiting passengers at ride-hailing hub h at time t
$W_h^{metro}(t)$	Cumulative waiting time at metro hub h up to time t (passenger-minutes)
$W_h^{ride}(t)$	Cumulative waiting time at ride-hailing hub h up to time t (passenger-minutes)
Hub State Variable	
$t_h^{next}(t)$	Time remaining until the next metro train arrives at hub h
$L_h^{train}(t)$	Current passenger load of the next arriving metro train at hub h
$r_h(t)$	Real-time ride-hailing vehicle supply rate at hub h
$b_h^{metro}(t)$	Number of passengers boarding metro at hub h at time t
$b_h^{ride}(t)$	Number of passengers picked up by ride-hailing at hub h
$u_{d,v,h}(t)$	Passengers unloaded from shuttle v at hub h at time t
Decision Variables	
$x_{l,v,p,t}$	Binary decision: dispatch bus v from line l to pickup point p at time t
$tw_{l,v,p,t}$	Dwell time of bus v from line l at pickup point p at time t
$y_{d,v,p,h,t}$	Binary decision: dispatch shuttle v from depot d to pickup point p and then to hub h , and then return to depot d at time t
$tw_{d,v,p,t}^{sh}$	Dwell time of shuttle v at pickup point p at time t
R_d	Number of shuttle vehicles pre-positioned at depot d (upper-level decision)

4.2. Vehicle Position Update

The position of each vehicle is updated based on its current status and the travel time uncertainty. However, vehicle speeds are not constant; they are influenced by real-time pedestrian density at the pickup zones they serve. To capture this critical interaction, we define a speed attenuation factor $\phi_p(t)$ for each pickup zone p , which reduces vehicle speed when crowds accumulate at the curbside. Let $\rho_p(t) = \omega_p(t)/A_p$ denote the pedestrian density (passengers per square meter) in the vicinity of pickup point p , where $\omega_p(t)$ is the current queue length and A_p is the effective standing area. The actual vehicle speed is then adjusted as: $\tilde{v}^{(\bullet)}(t) = v^{(\bullet)} \cdot \exp(-\kappa \cdot \rho_p(t))$. κ is a calibrated sensitivity coefficient. This formulation captures the phenomenon that when a massive crowd surges out of the stadium and occupies the curbside, the bus or shuttle must decelerate significantly regardless of the open-road speed limit. Consequently, the base speeds in all position update equations below— v^{norm} , v^{detour} , v^{return} , and $v^{shuttle}$ —are replaced by $\tilde{v}^{(\bullet)}(t)$ conditioned on the nearest pickup zone's real-time pedestrian density. This coupling makes the vehicle dynamics responsive to the very passengers they are trying to serve, reflecting realistic physical interference during stadium egress.

The update equations for each status are follows:

1. Normal operation ($s_{l,v}(t) = 0$):

$$p_{l,v}(t + \Delta t) = p_{l,v}(t) + \tilde{v}^{\text{norm}}(t) \cdot \Delta t \cdot (1 + \varepsilon_t), \varepsilon_t \sim N(0, \sigma_t^2) \quad (1)$$

this equation updates the vehicle position by adding the distance traveled during normal operation, which is the product of normal speed, time step, and a random factor representing travel time uncertainty.

2. En route to pickup point ($s_{l,v}(t) = 1$):

If vehicle v has been dispatched but has not yet reached the assigned pickup point:

$$p_{l,v}(t + \Delta t) = p_{l,v}(t) + \tilde{v}^{\text{detour}}(t) \cdot \Delta t \cdot (1 + \varepsilon_d), \varepsilon_d \sim N(0, \sigma_d^2) \quad (2)$$

this equation updates the vehicle position while traveling to a pickup point, considering detour speed and a different uncertainty factor.

If vehicle v reaches the pickup point at time t :

- Set $s_{l,v}(t + \Delta t) = 2$ (dwelling at pickup point).
- Record the assigned dwell time $tw_{l,v,p,t}$ (from the decision variable).
- Initialize a remaining dwell time counter $rd_{l,v}(t + \Delta t) = tw_{l,v,p,t}$.
- Apply the boarding process as per Section 4.3 to update $q_{l,v}(t)$.

3. Dwelling at pickup point ($s_{l,v}(t) = 2$):

If the vehicle is dwelling at a pickup point:

- Update remaining dwell time: $rd_{l,v}(t + \Delta t) = \max(0, rd_{l,v}(t) - \Delta t)$.
- Update vehicle position: the vehicle remains stationary $p_{l,v}(t + \Delta t) = p_{l,v}(t)$.
- Check dwell completion: if $rd_{l,v}(t + \Delta t) = 0$, set $s_{l,v}(t + \Delta t) = 3$ (start returning to original route).
- Update accumulated detour time: $d_{l,v}(t + \Delta t) = d_{l,v}(t) + tw_{l,v,p,t}$.

4. Returning to original route ($s_{l,v}(t) = 3$).

If vehicle v is returning to its original route and has not yet rejoined:

$$p_{l,v}(t + \Delta t) = p_{l,v}(t) + \tilde{v}^{\text{return}}(t) \cdot \Delta t \cdot (1 + \varepsilon_r), \varepsilon_r \sim N(0, \sigma_r^2) \quad (3)$$

this equation updates the vehicle position while returning to the original route, with return speed and corresponding uncertainty.

If vehicle v has rejoined:

- Set $s_{l,v}(t + \Delta t) = 0$ (normal operation).
- Set $\tau_{l,v}(t + \Delta t) = \varphi$.

4.3. Passenger State Update

The number of waiting passengers at each location is updated based on arrivals and boardings. The cumulative waiting time is also updated accordingly.

Pickup point waiting passengers:

$$w_p(t + \Delta t) = w_p(t) + \lambda_p \cdot \Delta t - \sum_{l \in \Lambda} \sum_{v \in \zeta_l} b_{l,v,p}(t) \quad (4)$$

this equation calculates the number of waiting passengers at pickup point p at the next time step, which equals the current waiting passengers plus new arrivals minus the total number of passengers who boarded vehicles at this point.

Line station waiting passengers:

$$w_{l,n}(t + \Delta t) = w_{l,n}(t) + \lambda_{l,n} \cdot \Delta t - \sum_{v \in \zeta_l} b_{l,v,n}(t) \quad (5)$$

similarly, this equation updates the number of waiting passengers at station n of line l, considering arrivals and boardings by vehicles on that line.

Cumulative waiting time:

$$W_p(t + \Delta t) = W_p(t) + w_p(t) \cdot \Delta t \quad (6)$$

this equation accumulates the total waiting time at pickup point p by adding the product of current waiting passengers and time step.

$$W_{l,n}(t + \Delta t) = W_{l,n}(t) + w_{l,n}(t) \cdot \Delta t \quad (7)$$

similarly, this accumulates the total waiting time at station n of line l.

4.4. Boarding Process

When a bus arrives at a pickup point, the number of passengers boarding is determined by the minimum of three quantities: available waiting passengers, available vehicle capacity, and maximum passengers that can board during the dwell time.

When bus v arrives at pickup point p:

$$b_{l,v,p}(t) = \min(w_p(t), C_{l,v} - q_{l,v}(t), \mu_p \cdot t w_{l,v,p,t}) \quad (8)$$

this equation determines the number of passengers boarding vehicle v at pickup point p, which is the minimum of: (1) waiting passengers at the point, (2) available capacity on the vehicle, and (3) maximum passengers that can board given the boarding rate and dwell time.

Update vehicle load:

$$q_{l,v}(t^+) = q_{l,v}(t^-) + b_{l,v,p}(t) \quad (9)$$

this equation updates the vehicle load by adding the number of boarded passengers.

4.5. Vehicle Availability Constraints

Single dispatch per vehicle:

$$\sum_{p \in P} x_{l,v,p,t} \leq 1 - I\{s_{l,v}(t) \in \{1, 2\}\}, \forall l \in L, v \in V, t \in T \quad (10)$$

this constraint ensures that at each time t, a bus can be dispatched to at most one pickup point, and only if it is not already on a detour (status 1 or 2).

4.6. Shuttle Vehicle Dynamics

In addition to buses, depot-based shuttle vehicles are dispatched to pickup points to transport evacuees directly to hubs (metro stations or ride-hailing points). The dynamics of shuttles are defined below.

4.6.1. Position Update for Shuttles

Let $s_{d,v}^{\text{sh}}(t) \in \{0, 1, 2, 3, 4\}$ denote the status of shuttle v from depot d:

0 = idle at depot.

1 = en route to pickup point.

2 = dwelling/boarding at pickup point.

3 = en route to hub.

4 = returning to depot.

1. Idle at depot ($s_{d,v}^{\text{sh}}(t) = 0$)

$$p_{d,v}^{\text{sh}}(t + \Delta t) = p_{d,v}^{\text{sh}}(t) \quad (11)$$

2. En route to pickup point ($s_{d,v}^{\text{sh}}(t) = 1$)

$$p_{d,v}^{\text{sh}}(t + \Delta t) = p_{d,v}^{\text{sh}}(t) + \tilde{v}^{\text{shuttle}}(t) \cdot \Delta t \cdot (1 + \varepsilon_{\text{sh}}), \quad \varepsilon_{\text{sh}} \sim N(0, \sigma_{\text{shuttle}}^2) \quad (12)$$

If the shuttle reaches the pickup point:

- Set $s_{d,v}^{\text{sh}}(t + \Delta t) = 2$
 - Record the assigned dwell time $tw_{d,v,p,t}^{\text{sh}}$.
 - Initialize remaining dwell time counter $rd_{d,v}^{\text{sh}}(t + \Delta t) = tw_{d,v,p,t}^{\text{sh}}$.
 - Apply the boarding process to update $q_{d,v}^{\text{sh}}(t)$.
3. Dwelling at pickup point ($s_{d,v}^{\text{sh}}(t) = 2$).
 - Update remaining dwell time: $rd_{d,v}^{\text{sh}}(t + \Delta t) = \max(0, rd_{d,v}^{\text{sh}}(t) - \Delta t)$
 - Vehicle remains stationary.
 - Check dwell completion: if $rd_{d,v}^{\text{sh}}(t + \Delta t) = 0$, set $s_{d,v}^{\text{sh}}(t + \Delta t) = 3$ (en route to hub).
 4. En route to hub ($s_{d,v}^{\text{sh}}(t) = 3$)

The update of $p_{d,v}^{\text{sh}}$ is identical to that in Equation (12).

If the shuttle reaches hub h:

- Update all passengers: $q_{d,v}^{\text{sh}}(t^+) = 0$.
 - Update hub waiting passengers $w_h^{\text{hub}}(t)$ (see Section 4.7).
 - Set $s_{d,v}^{\text{sh}}(t + \Delta t) = 4$ (return to the idle at depot).
5. Returning to depot ($s_{d,v}^{\text{sh}}(t) = 4$)

The shuttle travels back from the hub to its original depot. The position update is also identical to that in Equation (12).

If the shuttle reaches depot d:

- Set $s_{d,v}^{\text{sh}}(t + \Delta t) = 0$ (idle at depot, available for re-dispatch).
- Reset $p_{d,v}^{\text{sh}}$ to the depot's position.

4.6.2. Shuttle Boarding Process

The number of passengers boarding a shuttle at pickup point p is determined by the allocation mechanism in Section 4.8. In general:

$$b_{d,v,p}^{\text{sh}}(t) = \min\left(w_p(t) \cdot \omega_p(t), C^{\text{shuttle}} - q_{d,v}^{\text{sh}}(t), \mu_p \cdot tw_{d,v,p,t}^{\text{sh}}\right) \quad (13)$$

where $\omega_p(t)$ is the proportion of waiting passengers allocated to shuttles.

4.7. Hub Dynamics with Real-Time Operational Status

Hubs are of two types: metro stations and ride-hailing pickup zones. Their real-time status is captured by state variables defined below. The dynamics are discretized with time step Δt . At each time step, the following updates occur in sequence: (i) shuttle unloads arrive at hubs; (ii) metro trains may arrive and board passengers; (iii) ride-hailing pickups occur; (iv) waiting times are accumulated.

4.7.1. Metro Station Dynamics

1. Shuttle unloads (arrival of new evacuees)

Before any metro boarding happens, shuttles that have reached hub h during the interval $[t, t + \Delta t)$ unload their passengers. The total number of passengers unloaded is $\sum_{d,v} u_{d,v,h}(t)$. These passengers are added to the waiting queue:

$$w_h^{\text{metro}}(t^-) = w_h^{\text{metro}}(t) + \sum_{d \in \Delta} \sum_{v \in \zeta} u_{d,v,h}(t) \quad (14)$$

here t^- denotes the state just after unloading but before any train arrival processing.

2. Train arrival check

If $t_h^{\text{next}}(t) > 0$, no train arrives during this time step. The countdown decreases by Δt :

$$t_h^{\text{next}}(t + \Delta t) = \max(0, t_h^{\text{next}}(t) - \Delta t) \quad (15)$$

if $t_h^{\text{next}}(t) = 0$, a train is currently arriving. The boarding process occurs immediately.

3. Boarding process upon train arrival

When a train arrives, the number of passengers who can board is limited by both the waiting queue and the remaining capacity on the train:

$$b_h^{\text{metro}}(t) = \min(w_h^{\text{metro}}(t^-), C_h^{\text{train}} - L_h^{\text{train}}(t)) \quad (16)$$

After boarding, the train load increases and the waiting queue decreases:

$$L_h^{\text{train}}(t^+) = L_h^{\text{train}}(t) + b_h^{\text{metro}}(t) \quad (17)$$

$$w_h^{\text{metro}}(t^+) = w_h^{\text{metro}}(t^-) - b_h^{\text{metro}}(t) \quad (18)$$

The train then departs, and the next train arrival countdown is reset:

$$t_h^{\text{next}}(t^+) = T_h^{\text{head}} \quad (19)$$

if multiple trains could arrive within the same time step (e.g., Δt larger than headway), the above process is repeated until no more arrivals occur in this step.

4. If no train arrives ($t_h^{\text{next}}(t) > 0$), then $t_h^{\text{next}}(t + \Delta t)$ is set as above, and $w_h^{\text{metro}}(t^+) = w_h^{\text{metro}}(t^-)$.

5. Cumulative waiting time update

Regardless of whether a train arrived, the cumulative waiting time increases by the product of the waiting queue (after unloading and boarding) and the time step:

$$W_h^{\text{metro}}(t + \Delta t) = W_h^{\text{metro}}(t) + w_h^{\text{metro}}(t^+) \cdot \Delta t \quad (20)$$

4.7.2. Ride-Hailing Hub Dynamics

1. Shuttle unloads (arrival of new evacuees)

As with metro hubs, passengers unloaded from shuttles are added to the waiting queue.

$$w_h^{\text{ride}}(t^-) = w_h^{\text{ride}}(t) + \sum_{d \in \Delta} \sum_{v \in \zeta} u_{d,v,h}(t) \quad (21)$$

2. Ride-hailing pickup process

During the interval $[t, t + \Delta t)$, the total ride-hailing capacity available is $r_h(t) \cdot C_h^{\text{ride}} \cdot \Delta t$, where C_h^{ride} denotes the capacity per ride-hailing vehicle. In this study, we assume a maximum of 4 passengers per ride-hailing vehicle, consistent with the common ride-pooling practice of allowing carpooling, and the actual number of passengers picked up is limited by both this available capacity and the current waiting queue:

$$b_h^{\text{ride}}(t) = \min(w_h^{\text{ride}}(t^-), r_h(t) \cdot C_h^{\text{ride}} \cdot \Delta t) \quad (22)$$

After pickup, the waiting queue decreases:

$$w_h^{\text{ride}}(t^+) = w_h^{\text{ride}}(t^-) - b_h^{\text{ride}}(t) \quad (23)$$

3. Supply rate update

The supply rate $r_h(t)$ can follow an external stochastic process. This study uses a simple mean-reverting model:

$$r_h(t + \Delta t) = r_h(t) + \kappa(\bar{r} - r_h(t))\Delta t + \sigma_r \sqrt{\Delta t} \cdot \zeta(t) \quad (24)$$

where $\zeta(t) \sim N(0, 1)$.

4. Cumulative waiting time update.

$$W_h^{\text{ride}}(t + \Delta t) = W_h^{\text{ride}}(t) + w_h^{\text{ride}}(t^+) \cdot \Delta t \quad (25)$$

4.8. Capacity Constraints

4.8.1. Bus Constraints

1. Vehicle capacity limit:

$$q_{l,v}(t) \leq C_{l,v}, \forall l \in \Lambda, v \in \zeta_l, t \in T \quad (26)$$

this constraint ensures that the passenger load on each vehicle never exceeds its capacity.

Minimum effective capacity for dispatch:

$$x_{l,v,p,t} = 1 \Rightarrow C_{l,v} - q_{l,v}(t) \geq Q_{\min}, \forall l \in \mathcal{L}, v \in \mathcal{V}_l, p \in \mathcal{P}, t \in T \quad (27)$$

this constraint ensures that if a vehicle is dispatched ($x_{l,v,p,t} = 1$), it must have at least $Q_{\min}$ available capacity to be effective for evacuation. When $x_{l,v,p,t} = 0$, the constraint is trivially satisfied and imposes no requirement on the vehicle's remaining capacity.

2. Time Constraints

Maximum detour time:

$$d_{l,v}(t) \leq T_{\max}^{\text{detour}} \quad (28)$$

where $T_{\max}^{\text{detour}}$ is the maximum allowed detour time.

The accumulated detour time $d_{l,v}(t)$ consists of current accumulated time from previous detours and expected additional time if the vehicle is dispatched at time t . To account for travel time uncertainty, we use a chance constraint. Assuming normally distributed travel times with means μ and variances σ^2 , this can be converted to:

$$d_{l,v}(t) + (\Delta t_{l,v,p} + \Delta t_{l,v,p}^{\text{return}}) \cdot x_{l,v,p,t} + z_{1-\gamma} \cdot \sqrt{\sigma_d^2 + \sigma_r^2} \cdot x_{l,v,p,t} \leq T_{\max}^{\text{detour}} \quad (29)$$

where $\Delta t_{l,v,p}$ and $\Delta t_{l,v,p}^{\text{return}}$ are the expected travel time to pickup point and return time, $z_{1-\gamma}$ is the $(1 - \gamma)$ -quantile of the standard normal distribution.

3. Operational Constraints

Minimum headway:

$$\left| t_{l,v,p}^{\text{dep}} - t_{l,v',p}^{\text{dep}} \right| \geq H'_{l,\min}, \dots \forall l \in \mathcal{L}, v \neq v' \in \mathcal{V}_l, p \in \mathcal{P} \quad (30)$$

this constraint ensures that the departure times of consecutive vehicles at each pickup point are separated by at least the minimum headway $H_{\min}$.

Maximum waiting time for regular passengers:

$$\frac{w_{l,n}(t)}{\lambda_{l,n}} \leq W_{\max}^{\text{normal}}, \dots \forall l \in \mathcal{L}, n \in N_l, t \in T \quad (31)$$

this constraint ensures that the estimated waiting time for regular passengers (waiting passengers divided by arrival rate) does not exceed the maximum allowed waiting time.

Line dispatch balance:

$$\frac{1}{|T|} \sum_{t \in T} \sum_{v \in V} \sum_{p \in \Pi} x_{l,v,p,t} \leq \phi_l \cdot |\zeta_l|, \dots \forall l \in \mathcal{L} \quad (32)$$

this constraint ensures that on average, no more than ϕ_l proportion of buses from line l are dispatched, preventing over-exploitation of any single line.

4.8.2. Shuttle Constraints

1. Shuttle capacity limit

$$q_{d,v}^{\text{sh}}(t) \leq C^{\text{shuttle}}, \forall d \in \Delta, v \in \mathcal{V}_d^{\text{shuttle}}, t \in T \quad (33)$$

this constraint ensures that the passenger load on any shuttle vehicle never exceeds its physical capacity C^{shuttle} .

2. Maximum round-trip time (chance constraint)

$$P\left(T_{d,v,p,h}^{\text{shu,act}} \leq T_{\max}^{\text{shuttle}}\right) \geq 1 - \gamma, \forall d, v, p, h \quad (34)$$

where $T_{d,v,p,h}^{\text{shu,act}}$ is the actual random round-trip time from depot d to pickup point p and to hub h , and then return to depot. This chance constraint requires that the round-trip time does not exceed the maximum allowed $T_{\max}^{\text{shuttle}}$ with probability at least $1 - \gamma$. To make it computationally tractable, we assume travel times are normally distributed with mean $\mu_{d,p,h}^{\text{sh}}$ and variance $(\sigma^{\text{shuttle}})^2$. The constraint is then reformulate as:

$$\mu_{d,p,h}^{\text{sh}} + z_{1-\gamma} \cdot \sigma^{\text{shuttle}} \leq T_{\max}^{\text{shuttle}} \quad (35)$$

where $z_{1-\gamma}$ is the $(1 - \gamma)$ -quantile of the standard normal distribution. This ensures robust dispatching under travel time uncertainty.

3. Depot availability constraint

$$\sum_{v \in \mathcal{V}_d^{\text{shuttle}}} I\left\{s_{d,v}^{\text{sh}}(t) = 0\right\} \geq \sum_{p \in \mathcal{P}} \sum_{h \in H} y_{d,v,p,h,t}, \forall d \in \mathcal{D}, t \in T \quad (36)$$

the left-hand side counts the number of shuttle vehicles currently idle at depot d . The right-hand side sums over all dispatch decision $y_{d,v,p,h,t}$ that would sent shuttles from depot d at time t .

4.8.3. Pre-Event Upper-Level Constraints

These constraints govern the strategic (pre-event) decision of how many shuttle vehicles to position at each depot. They are defined before the real-time operation and remain fixed throughout the evacuation.

1. Total budget constraint

$$\sum_{d \in D} R_d \leq R^{\max} \quad (37)$$

where R_d is the number of shuttles pre-positioned at depot d , and $R^{\max}$ the maximum number of shuttles that can be stored at depot d .

2. Maximum shuttles per depot

$$0 \leq R_d \leq R_d^{\max}, \forall d \in D \quad (38)$$

4.9. Passenger Demand Allocation Mechanism at Pickup Points

In the multi-modal system, evacuees waiting at a pickup point p are served by whichever vehicle (bus or shuttle) arrives first with available capacity. To enable the optimization to balance the two modes, we introduce a dynamic allocation variable $\omega_p(t)$ that represents the target proportion of waiting passengers intended for shuttle vehicles.

Let $\omega_p(t) \in [0, 1]$ be the proportion of waiting passengers at pickup point p that are expected to board shuttles during the next decision interval. It is derived from the relative service pressure:

$$\omega_p(t) = \frac{\frac{1}{\bar{T}_p^{\text{shuttle}}(t)} \cdot \text{RemShuttle}(t)}{\frac{1}{\bar{T}_p^{\text{bus}}(t)} \cdot \text{RemBus}(t) + \frac{1}{\bar{T}_p^{\text{shuttle}}(t)} \cdot \text{RemShuttle}(t)} \quad (39)$$

where $\bar{T}_p^{\text{bus}}(t)$ is estimated average waiting time for a bus to arrive at p based on current positions and travel time distributions, $\bar{T}_p^{\text{shuttle}}(t)$ is estimated waiting time for a shuttle to arrive at p , $\text{RemBus}(t)$ is total remaining capacity of buses already dispatched to p plus those that can be dispatched within the next window, $\text{RemShuttle}(t)$ is analogous for shuttles.

This formulation allocates more passengers to the mode that provides faster service, weighted by available capacity. When a shuttle arrives at p , the actual number of boarding passengers is:

$$b_{d,v,p}^{\text{sh}}(t) = \min(w_p(t) \cdot \omega_p(t), C^{\text{shuttle}} - q_{d,v}^{\text{sh}}(t), \mu_p \cdot tw_{d,v,p,t}^{\text{sh}}) \quad (40)$$

When a bus arrives at p , the boarding number is:

$$b_{d,v,p}^{\text{bus}}(t) = \min(w_p(t) \cdot (1 - \omega_p(t)), C_{l,v} - q_{l,v}(t), \mu_p \cdot tw_{l,v,p,t}) \quad (41)$$

the remaining waiting passengers after boarding update $w_p(t + \Delta t)$ as in Section 4.3, but with the boarding terms replaced by the sum of b^{sh} and b^{bus} .

4.10. Objective Function

Minimize the expected total cost considering travel time uncertainty:

Upper-level objective (pre-event):

$$\begin{aligned} \min_{R_d} \{ & \sum_{d \in D} c^{\text{vehicle}} R_d + E_s[J_{\text{lower}}(R_d)] \} \\ \text{Subject to :} & \\ E_e[T_{\text{evac}}(R_d)] & \simeq T_{\text{max}} \end{aligned} \quad (42)$$

where $J_{\text{lower}}(R_d)$ is the minimal expected lower-level cost given the pre-positioned shuttle fleet, $T_{\text{evac}}(R_d)$ is the expected evacuation completion time for fleet configuration R_d , and T_{max} is the maximum acceptable evacuation time stipulated by the emergency management plan. This constraint reflects the practical requirement that service level takes precedence over cost minimization in emergency operations.

The real-time (lower-level) objective is to minimize the expected total cost over the evacuation horizon T , considering travel time uncertainty, operational penalties, and hub congestion. The objective function is defined as:

$$\min E_\epsilon \left[\begin{aligned} & \alpha \sum_{p \in P} W_p(T) + \beta \sum_{l \in L} \sum_{n \in N_l} W_{l,n}(T) + \delta \sum_{l \in L} \sum_{v \in V_l} \sum_{p \in P} \sum_{t \in T} x_{l,v,p,t} \cdot \theta_{k(l,v,p,t)} \\ & + \eta_{\text{sh}} \sum_{d \in D} \sum_{v \in V_d^{\text{shuttle}}} \sum_{p \in P} \sum_{t \in T} y_{d,v,p,h,t} \cdot \theta_{k(d,v,p,t)}^{\text{sh}} + \theta_{\text{metro}} \sum_{h \in H^{\text{metro}}} W_h^{\text{metro}}(T) + \theta_{\text{ride}} \sum_{h \in H^{\text{ride}}} W_h^{\text{ride}}(T) \end{aligned} \right] \quad (43)$$

the lower-level objective function consists of six components¹, evacuee waiting at pickup points, regular passenger waiting, penalty for bus dwell times, penalty for shuttle dwell times, waiting at metro hubs and waiting at ride-hailing hubs.

All cumulative waiting time terms ($W_p(T)$, $W_{l,n}(T)$, $W_h^{\text{metro}}(T)$, $W_h^{\text{ride}}(T)$) are updated according to the dynamics in Sections 4.3, 4.7.1 and 4.7.2. The expectation $\mathbb{E}_\epsilon$ is taken over the random travel time disturbances.

These objective components form a well-balanced reward signal that guides the agent to make real-time dispatching decisions under uncertainty, trading off evacuation speed and regular service disruption. Firstly, the waiting time terms (α , β , θ_{metro} , θ_{ride}) directly capture the primary goal of minimizing passenger delay. They provide immediate per-step feedback, enabling the agent to learn policies that reduce queue accumulation. Secondly, the dwell time penalties (δ , η_{sh}) address the temporal opportunity cost: prolonged boarding delays vehicle return, reducing future service frequency. Without this penalty, the agent might adopt a myopic strategy that clears the current queue with an excessively long stop while harming overall throughput. Hence, these terms act as a regulariser that aligns short-term actions with long-term system efficiency.

5. Uncertainty-Aware Reinforcement Learning with Categorical DQN

We formulate the dynamic bus dispatching problem as a Markov Decision Process under travel time uncertainty. The proposed UARL-CD algorithm learns a policy that jointly controls both bus and shuttle dispatches while explicitly modeling the distribution of returns to account for environmental stochasticity. The agent observes the full system state at discrete epochs $t \in \mathcal{T}$ and selects action for all available vehicle. After executing the actions, the environment transitions to a new state and returns a reward signal aligned with the lower-level objective.

The proposed UARL-CD adopts a centralized-training-decentralized-execution architecture. During training, a single global Q-network is updated using system-wide rewards and state information, capturing inter-vehicle coordination through shared observations. During execution, each vehicle's action is selected independently from this unified network based on the current global state, enabling real-time decisions without the non-stationarity and credit assignment challenges of multi-agent formulations.

5.1. State Representation

The state s_t at time t must capture all information required for making coordinated dispatching decisions. It is defined as:

$$s_t = \{ \mathbf{P}_t, \mathbf{Q}_t, \mathbf{S}_t, \mathbf{P}_t^{\text{sh}}, \mathbf{Q}_t^{\text{sh}}, \mathbf{S}_t^{\text{sh}}, \mathbf{W}_t, \mathbf{D}_t, \mathbf{\Lambda}_t, \mathbf{M}_t, \mathbf{\Phi}_t, \mathbf{H}_t \}$$

where $\mathbf{P}_t$ is the position of all buses $\{p_{l,v}(t)\}_{\forall l,v}$, $\mathbf{Q}_t$ is passenger loads of all buses $\{q_{l,v}(t)\}_{\forall l,v}$, $\mathbf{S}_t$ is bus statuses $\{s_{l,v}(t)\}_{\forall l,v}$ (0 = available, 1 = en route to pickup, 2 = dwelling, 3 = returning), $\mathbf{P}_t^{\text{sh}}$ is the position of all shuttle vehicles $\{p_{d,v}^{\text{sh}}(t)\}_{\forall d,v}$, $\mathbf{Q}_t^{\text{sh}}$ is the passenger loads of all shuttles $\{q_{d,v}^{\text{sh}}(t)\}_{\forall d,v}$, $\mathbf{S}_t^{\text{sh}}$ is the statuses of all shuttles $\{s_{d,v}^{\text{sh}}(t)\}_{\forall d,v}$, $\mathbf{W}_t$ is waiting passengers at pickup points $\{w_p(t)\}_{p \in \mathcal{P}}$ and at regular stations $\{w_{l,n}(t)\}_{\forall l,n}$, $\mathbf{D}_t$ is accumulated detour times $\{d_{l,n}(t)\}_{\forall l,n}$, $\mathbf{\Lambda}_t$ is the resent travel time uncertainty realizations (moving averages over the past k time steps), $\mathbf{M}_t$ is recent boarding rates $\{\mu_p(t)\}_{p \in \mathcal{P}}$ at each pickup point p , calculated as moving averages of actual boarding per minute. $\mathbf{\Phi}_t$ is dynamic passenger allocation proportions at pickup points $\{\varphi_p(t)\}_{p \in \mathcal{P}}$, $\mathbf{H}_t$ is real time hub status, containing for each hub $h \in \mathcal{H}$:

$$\mathbf{H}_t = \{ w_h^{\text{metro}}(t), t_h^{\text{next}}(t), L_h^{\text{train}}(t), w_h^{\text{ride}}(t), r_h(t) \}$$

where $w_h^{\text{metro}}(t)$ and $w_h^{\text{ride}}(t)$ are waiting queues at metro and ride-hailing hubs, $t_h^{\text{next}}(t)$ is the time until the next metro train, $L_h^{\text{train}}(t)$ is the current train load, and $r_h(t)$ the ride-hailing vehicle supply rate.

5.2. Action Space with Discrete Dwell Time

The agent must select actions for all available buses and all idle shuttles simultaneously. For computational tractability, dwell times are discretized into K levels (e.g., 2, 4, 6 min). Each level k corresponds to a predetermined dwell time ϑ_k via a mapping function $f: k \mapsto \vartheta_k$.

1. Bus Action Space

For each bus v on line l with status $s_{l,v}(t) = 0$, the action is:

$$a_{l,v}^{\text{bus}} \in \mathcal{A}_{l,v}^{\text{bus}} = \{0\} \cup \{(p, k) : p \in \mathcal{P}, k \in \{1, \dots, K\}\}$$

where 0 denotes no dispatch (continue normal operation), (p, k) denotes dispatch the bus to pickup point p with dwell time level k .

2. Shuttle Action Space

For each idle shuttle v at depot d (status $s_{d,v}^{\text{sh}}(t) = 0$), the action is:

$$a_{d,v}^{\text{sh}} \in \mathcal{A}_{d,v}^{\text{sh}} = \{0\} \cup \{(p, h, k) : p \in \mathcal{P}, h \in \mathcal{H}, k \in \{1, \dots, K\}\}$$

where 0 leaves the shuttle idle, and (p, h, k) dispatches the shuttle to pickup point p , dwells for ϑ_k minutes, then proceeds to hub h (metro or ride-hailing). The shuttle's return to depot after unloading is automatic.

3. Joint Action

The joint action at time t is the concatenation of all individual vehicle actions:

$$a_t = \left[\left\{ a_{l,v}^{\text{bus}} \right\}_{\forall l,v: s_{l,v}(t)=0}, \left\{ a_{d,v}^{\text{sh}} \right\}_{\forall d,v: s_{d,v}^{\text{sh}}(t)=0} \right]$$

The total number of possible joint actions is:

$$\prod_{(l,v:s_{l,v}(t)=0)} (1 + |\mathcal{P}| \times K) \times \prod_{(d,v:s_{d,v}^{sh}(t)=0)} (1 + |\mathcal{P}| \times |\mathcal{H}| \times K)$$

which is extremely large. The categorical DQN handles this by outputting independent Q-value distributions for each vehicle–action pair.

Although actions are selected independently per vehicle, coordination is maintained through three mechanisms. First, all selections condition on the shared global state, aligning decisions toward globally optimal allocation. Second, the shared reward signal trains the network to maximize system-wide efficiency rather than individual vehicle performance. Third, the action masking mechanism (detailed in Section 5.3) enforces operational constraints locally, including headway and dispatch balance limits, ensuring that independently chosen actions collectively constitute a feasible joint action. Thus, independent selection is a computationally efficient implementation of a coordinated policy, not a relaxation of the coordination problem.

5.3. Constraint-Aware Action Masking

To avoid sampling infeasible actions, we apply action masking during both training and execution. For each vehicle, invalid actions are disabled (masked) according to the following rules.

1. Bus Action Masking

A bus action (p, k) is masked if any of the following holds:

- Vehicle not available: $s_{l,v}(t) \neq 0$.
- Insufficient capacity: (p, k) is masked if $C_{l,v} - q_{l,v}(t) < Q_{\min}$.
- Detour time limit would be exceeded: (p, k) is masked if $d_{l,v}(t) + \hat{T}_{l,v,p} + \vartheta_k > T_{\max}^{\text{detour}}$, where $\hat{T}_{l,v,p}$ is the estimated round-trip travel time to point p and back using mean travel times.
- Line dispatch balance exceeded: Actions from line l are masked if the current dispatch ratio exceeds ϕ_l .

2. Shuttle Action Masking

A shuttle action (p, h, k) is masked if:

- Shuttle not idle: $s_{d,v}^{sh}(t) \neq 0$.
- Round-trip time limit would be exceeded: $\mu_{d,p,h}^{sh} + z_{1-\gamma} \sigma^{\text{shuttle}} + \vartheta_k > T_{\max}^{\text{shuttle}}$.
- No idle shuttles at depot: The left-hand side of the depot availability constraint is zero, but the action would require a shuttle; this is already covered by the idle check.
- Hub queue exceeds soft limit: if the target hub's current waiting queue $w_h^{\text{metro}}(t)$ or $w_h^{\text{ride}}(t)$ is above a predefined threshold, the action can be masked to avoid worsening congestion.

Masked actions are excluded from the agent's action selection, ensuring policy compliance with operational constraints.

5.4. Reward Function

The reward r_t received after executing joint action a_t and transitioning to s_{t+1} is designed to match the lower-level objective as closely as possible, but expressed as an incremental signal:

$$r_t = - \left[\begin{array}{l} \alpha \sum_{p \in P} \Delta W_p(T) + \beta \sum_{l \in L} \sum_{n \in N_l} \Delta W_{l,n}(T) + \delta \sum_{\text{bus dispatches}} \vartheta_k \\ + \eta_{\text{sh}} \sum_{\text{shuttle dispatches}} \vartheta_k^{\text{sh}} + \theta_{\text{metro}} \sum_{h \in H^{\text{metro}}} \Delta W_h^{\text{metro}}(T) + \theta_{\text{ride}} \sum_{h \in H^{\text{ride}}} \Delta W_h^{\text{ride}}(T) \end{array} \right]$$

where $\Delta W_p(t) = W_p(t) - W_p(t-1)$ is the increment in cumulative waiting time at pickup point p during the last Δt minutes (similarly for $\Delta W_{l,n}(t)$, $\Delta W_h^{\text{metro}}(t)$, $\Delta W_h^{\text{ride}}(t)$). The sums over dispatches count the number of buses and shuttles that were actually dispatched in action a_t . All weight α , β , δ , η_{sh} , θ_{metro} , θ_{ride} are non-negative and chosen to reflect the relative importance of each cost component.

This reward function provides immediate, per-step feedback that aligns with the long-term objective. The dwell penalties act as regularizers, preventing the agent from adopting myopic strategies that clear queues via excessively long stops at the expense of future service frequency.

5.5. A Hybrid Algorithm for Multi-Modal Evacuation

5.5.1. Algorithm Framework

This paper solves the two-layer problem using a hybrid approach. Upper layer (pre-event) optimizes the number of shuttle vehicles R_d to pre-position at each depot, balancing fixed investment cost against expected evacuation performance. Then, given a fixed R_d , lower layer (real-time) with the UARL-CD agent learns a dispatching policy for buses and shuttles under travel time uncertainty.

The overall algorithm is shown as Algorithm 1, which consists of an outer loop over candidate upper-level decisions, and an inner loop that trains the UARL-CD agent and evaluates the expected lower-level cost $J_{\text{lower}}(R_d)$.

In this work, we optimize the upper level using sample average approximation (SAA) with a small set of candidate fleet configurations (e.g., grid search over a bounded integer space). For each candidate $\mathbf{R} = (R_d)$, we run M independent training-evaluation cycles of the lower-level UARL-CD algorithm and average the resulting total costs. The configuration with the lowest total cost (upper-level vehicle cost plus average lower-level cost) is selected.

Algorithm 1: Two-Layer UARL-CD for Multi-Modal Evacuation

Input: Environment env, atom size G , value range $[V_{\min}, V_{\max}]$, replay buffer capacity $|\mathcal{D}|$, dwell time levels ϑ_k , exploration rate ϵ , discount factor γ , target update interval C , number of lower-level episodes J , number of upper-level samples M , candidate fleet set $\mathcal{R}_{\text{cand}}$ (e.g., all integer combinations of R_d satisfying budget and max constraints).

Output: Optimal pre-vent fleet $\mathbf{R}^*$ and learned lower-level policy π^*

1. Upper-level optimization (pre-event):

For each candidate fleet configuration $\mathbf{R} \in \mathcal{R}_{\text{cand}}$:

Set $\text{total_upper_cost} = \sum_d c^{\text{vehicle}} R_d$

For $m = 1$ to M :

Train a lower-level UARL-CD agent (steps 2–4 below) with fixed $\mathbf{R}$, obtaining an estimate of the expected lower-level cost $\hat{J}_{\text{lower}}^{(m)}(\mathbf{R})$.

Compute average lower-level cost $\bar{J}^{\text{lower}}(\mathbf{R}) = \frac{1}{M} \sum_m \hat{J}_{\text{lower}}^{(m)}(\mathbf{R})$.

Store total cost $C_{\text{total}}(\mathbf{R}) = \text{total_upper_cost} + \bar{J}_{\text{lower}}(\mathbf{R})$.

Select $\mathbf{R}^* = \arg \min_{\mathbf{R} \in \mathcal{R}_{\text{cand}}} C_{\text{total}}(\mathbf{R})$

2. Lower-level training (real-time policy learning) with optimal $\mathbf{R}^*$:

Initialize online network Q_{θ} with categorical output for each bus-action and shuttle-action pair. Initialize target network $Q_{\theta^-} \leftarrow Q_{\theta}$.

Initialize replay buffer $\mathcal{D} = \emptyset$.

Algorithm 1: Cont.

For episode = 1 to J :
 Reset environment to initial state s_0 .
 For $t = 1$ to T :
 a. Observe state s_t .
 b. For each available bus (l, v) and each idle shuttle (d, v) , compute valid action sets $A_{bus}^{valid}(l, v)$ and $A_{sh}^{valid}(d, v)$ using constraints in Section 4.
 c. Select joint action a_t :
 With probability ϵ : randomly select a_t from valid actions.
 Otherwise: for each vehicle, select $a = \arg \max_{a' \in A^{valid}} \sum_{i=1}^G z_i \cdot p_i(s_t, a'; \theta)$.
 d. Execute a_t in environment (simulate one step: movement, boarding, hub events).
 Observe reward r_t and next state s_{t+1} .
 e. Store transition (s_t, a_t, r_t, s_{t+1}) in $\mathcal{D}$.
 f. Sample mini-batch $\mathcal{B}$ from $\mathcal{D}$.
 For each (s, a, r, s') in $\mathcal{B}$:
 Compute target distributions using target network (greedy actions in s' with masking, $T_{z_j} = \text{clip}(r + \gamma z_j)$, then L2 projection to atoms)
 Compute KL loss: cross-entropy between target and predicted distributions
 Update θ via gradient descent on loss via Adam.
 Every C steps: $\theta^- \leftarrow \theta$.
 End for.
 After training, evaluate the learned policy on M_{eval} independent test episodes (with new random seeds) to obtain $\hat{J}_{lower}(\mathbf{R}^*)$.
 3. Return $\mathbf{R}^*$ and the trained lower-level policy π^* derived from Q_θ (greedy action selection with masking)

5.5.2. UARL-CN Algorithm for Lower-Level

We adapt the categorical DQN [27] algorithm for lower-level to handle the multi-discrete action space with action masking. Instead of estimating the expected Q-value, the network outputs a categorical distribution over a fixed set of atoms for each vehicle–action pair.

1. Distributional Value Representation

Let $\{z_i\}_{i=1}^G$ be a set of atoms equally spaced between V_{min} and V_{max} . For each individual vehicle–action pair (l, v, a) for buses or (d, v, a) for shuttles, the network outputs a probability vector $\mathbf{p}(s_t, a) = [p_1, \dots, p_G]$ such that $\sum_i p_i = 1$. The Q-value is approximated as the expectation:

$$Q(s_t, a) \approx \sum_{i=1}^G z_i \cdot p_i(s_t, a).$$

because actions are selected independently per vehicle (but the joint action is executed simultaneously), the algorithm uses independent action selection for computational feasibility: for each available vehicle, the agent chooses the action that maximizes the expected Q-value among the valid actions, given the current state.

2. Training Procedure

The training uses a replay buffer $\mathcal{D}$ and target network with parameters θ^- . At each update, a mini-batch of transition (s, a, r, s') is sampled. For each transition, the target distribution is computed as follow. Firstly, for each vehicle, compute the greedy action in the next state s' using valid actions only:

$$a^* = \arg \max_{a' \in \mathcal{A}_{\text{valid}}(s')} \sum_{i=1}^G z_i \cdot p_i(s', a'; \theta^-)$$

Secondly, obtain the target distribution for atom z_j :

$$T_{z_j} = \text{clip}(r + \gamma \cdot z_j, V_{\min}, V_{\max})$$

where γ is the discount factor.

Then, project the T_{z_j} onto the original atom grid using as L2 projection as in the standard categorical DQN. This yields a target probability vector $\hat{\mathbf{p}}$ for each vehicle–action pair.

The loss is the Kullback–Leibler divergence between the predicted distribution $\mathbf{p}(s, a; \theta)$ and the target distribution $\hat{\mathbf{p}}$:

$$\mathcal{L}(\theta) = \sum_{\text{vehicle–action pairs in } a} D_{\text{KL}}(\hat{\mathbf{p}} \| \mathbf{p}(s, a; \theta)).$$

the network parameter θ are updated via gradient decent using the Adam optimizer. The target network θ^- is updated every C steps.

The pseudocode of the two-layer UARL-CD algorithm is shown below.

6. Numerical Experiments

To validate the effectiveness and robustness of the proposed UARL-CD algorithm for real-time multi-modal evacuation in large-scale event evacuations, we conduct a series of simulation-based experiments under stochastic travel time conditions.

6.1. Experimental Setup

6.1.1. Network and Parameter Settings

All simulation parameters are calibrated based on a field investigation of the Dalian Suoyuwan Football Stadium. The evacuation scenario is modeled after a major stadium event with 20,000 attendees, of which 80% are assumed to use the multi-modal evacuation system. The total attendance of 20,000 passengers represents a mid-sized mass gathering event, which is a common scale for professional football matches and large-scale concerts. Consistent with observed egress behavior at similar events, approximately 80% of these attendees are assumed to utilize the multi-modal evacuation system, with the remaining 20% departing via private vehicles, walking, or other unmodeled modes. The simulation operates with a decision time step $\Delta t = 2$ min and a maximum of $T = 60$ decision steps during which the RL agent actively makes dispatch decisions. The simulation terminates only when all passengers have been evacuated from the system, with the evacuation completion time recorded as the moment when the last passenger get on a vehicle. This design ensures that the actual clearance time is measured for all strategies, providing a fair and complete comparison.

The evacuation scenario includes $|P| = 4$ designated pickup zones within a 500 m radius of the venue, $|\mathcal{H}| = 4$ hubs (two metro stations and two ride-hailing pickup zone), and a regular transit network consisting of 25 bus lines operating within a 3 km radius, each with 8–10 buses and headways ranging from 4 to 6 min. Each line serves on average 15 regular stations, where passenger arrival rates $\lambda_{l,n}$ follow a uniform distribution with

mean 5–10 passengers/minute during the peak egress period. The three depots are located 2 km and 3 km from the venue, respectively, each with a maximum storage capacity of 20 shuttle vehicles, with a per-vehicle rental cost of 1000 RMB.

Travel time and speed parameters, including bus speeds and shuttle speed are set based on the actual road network characteristics surrounding the stadium and measured transit operating speeds during event egress periods, with the detour speed slightly reduced to account for turning and merging maneuvers.

Metro and ride-hailing parameters, including metro headway (4 min), and ride-hailing supply dynamics, are calibrated based on the actual public transportation infrastructure serving the venue and surrounding urban areas.

The weighting coefficients in the lower-level objective are set based on operational priorities in evacuation management: passenger waiting time at pickup points receives the highest weight ($\alpha = 1.0$) as the primary evacuation goal, while hub congestion weights ($\theta_{metro} = \theta_{ride} = 0.8$) are slightly lower since passengers at hubs have already completed the primary pickup phase and typically have better waiting conditions; regular service disruption ($\beta = 0.5$) is assigned the lowest weight because emergency evacuation takes precedence over normal transit operations, though still accounted for; and shuttle operation ($\eta_{sh} = 1.0$) is weighted equally to a to discourage excessively long dwell times that reduce shuttle throughput and delay vehicle returns.

Key model parameters are summarized in Table 2, while the algorithm hyperparameters of UARL-CD are listed in Table 3.

Table 2. Parameter settings.

Parameter	Symbol	Value
Bus capacity	$C_{l,v}$	70 passengers
Shuttle capacity	$C^{shuttle}$	50 passengers
Minimum effective capacity for bus dispatch	Q_{min}	30 passengers
Maximum bus detour time	T_{max}^{detour}	30 min
Minimum headway between buses at same pickup point	H_{min}	5 min
Max waiting time for regular passengers	W_{max}^{normal}	15 min
Max dispatch proportion per line	ϕ_l	0.4
Bus speed (km/min)	$v^{norm}, v^{detour}, v^{shuttle}$	0.5, 0.4, 0.5
Shuttle speed (km/min)	$v^{shuttle}$	0.5
Metro headway	T_h^{head}	4 min
Passenger arrival rate at pickup point p	λ_p	80–120 passengers/min
Regular station arrival rate	$\lambda_{l,n}$	5–15 passengers/min
Boarding rate	μ_p	20 passengers/min
Travel time uncertainty std	$\sigma_t, \sigma_d, \sigma_r, \sigma_{shuttle}$	0.15, 0.15, 0.15, 0.15
Confidence level for chance constraints	$1 - \gamma$	0.95
Weighting coefficients	$\alpha, \beta, \delta, \eta_{sh}, \theta_{metro}, \theta_{ride}$	1.0, 0.5, 1.0, 1.0, 0.8, 0.8

Table 3. UARL-CD Algorithm Hyperparameters.

Hyperparameter	Value
Learning rate	5×10^{-4}
Discount factor	0.99
Number of atoms	51
Training episodes	1500
Mini-Batch size	128
Hidden layer sizes	[256, 256] with ReLU
Value range $[V_{min}, V_{max}]$	$[-200, 50]$

6.1.2. Simulation Environment

The environment is implemented in Python 3.10 using a discrete-time, event-driven simulator. At each time step $\Delta t = 2$ min: (i) all vehicle positions are updated using the stochastic equations in Sections 4.2 and 4.6; (ii) arrivals at pickup points trigger boarding according to the allocation mechanism (Section 4.9); (iii) arrivals at hubs trigger metro boarding (if a train is present) or ride-hailing pickups (based on real-time supply); (iv) cumulative waiting time variables are incremented; and (v) the reward r_t is computed as per Section 5.4. Random seeds are fixed for reproducibility across different strategies.

In the simulation environment, the ride-hailing supply rate $r_h(t)$ evolves according to the stochastic mean-reverting process defined in Equation (24). At each time step, a random perturbation $\xi(t) \sim N(0, 1)$ is sampled to generate different supply trajectories across episodes. This exposes the agent to a wide range of supply conditions during training, enabling it to learn policies that are robust to fluctuations in ride-hailing availability. The current supply rate $r_h(t)$ is included in the state representation $\mathbf{H}_t$, allowing the agent to condition its dispatching decisions on real-time supply conditions.

6.2. Results and Analysis

To validate the effectiveness and efficiency of the proposed UARL-CD framework, we conduct comprehensive simulation experiments under the stochastic environment defined in Section 6.1. The performance of our approach is compared against four alternative strategies: (1) Shuttle-only UARL-CD, a variant of our algorithm that uses the same RL framework and training procedure but with all bus dispatch actions disabled. This strategy serves as an ablation component that isolates the marginal contribution of integrating in-service buses into the evacuation system—a standard practice in ablation studies to demonstrate the necessity of each system component. (2) a Deterministic Optimization (DO) model solved as a Mixed-Integer Linear Programming (MILP) problem using expected travel times. This strategy is included as a legitimate optimization-based approach that practitioners would consider as a benchmark, demonstrating that even with perfect information about expected travel times, deterministic planning falls short under stochastic conditions. (3) a Fixed-Interval Dispatch (FID) rule-based policy that dispatches both buses and shuttles at fixed 8-min intervals. FID represents a simple yet practically implementable rule-based strategy commonly used in real-world contingency planning for its ease of execution and low communication overhead, providing a lower-bound performance reference for adaptive decision-making. (4) a Standard DQN variant that shares the same state representation and stochastic training environment as UARL-CD, differing only in its value representation (expected Q-values rather than a categorical distribution), to isolate the contribution of distributional value representation to robustness. Each algorithm is evaluated over 10 independent simulation runs with different stochastic realizations of travel times and passenger arrivals.

6.2.1. Upper-Level Fleet Optimization Results

The upper-level optimization determines the number of shuttle vehicles to pre-position at each depot before the event, balancing rental costs against expected operational performance. Figure 2 illustrates the total expected cost as a function of the total number of shuttle vehicles, evaluated at discrete candidate fleet sizes of 30, 35, 40, 45, 50, 55, and 60 (the maximum physical capacity across three depots). For each candidate total, the fleet is distributed as evenly as possible among the three depots to balance spatial coverage and reduce deadhead travel time.

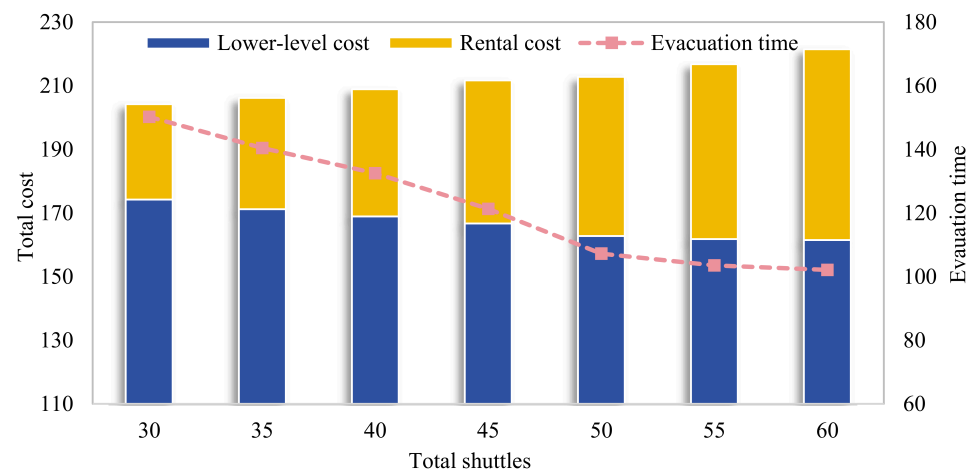


Figure 2. Upper-level cost comparison across candidate fleet sizes.

As shown in Figure 2, the total cost increases monotonically with fleet size, as the rental cost of additional shuttles consistently outweighs the marginal operational savings. From a purely cost-minimization perspective, the smallest fleet (30 shuttles) would appear optimal. However, the evacuation completion time exhibits a decreasing trend as more shuttles become available, reflecting improved service capacity. The upper-level objective defined in Equation (42) explicitly incorporates a hard constraint on evacuation completion time to reflect the practical priority of service level in emergency management.

As indicated by the dashed curve in Figure 2, fleet sizes below 50 shuttles result in evacuation completion times exceeding the acceptable threshold, failing to satisfy this hard service-level requirement. At 50 shuttles, the evacuation time is reduced to 106.2 min. For fleet sizes above 50, the marginal reduction in lower-level cost diminishes to negligible levels, while rental costs continue to accrue without commensurate operational benefit. Therefore, the 50-shuttle configuration thus represents the minimum fleet size that guarantees the required service level, making it the most pragmatic choice for real-world deployment.

To validate the reasonableness of the near-equal distribution (17, 17, 16) obtained from the upper-level grid search, we conduct a sensitivity analysis comparing this optimal configuration against several deliberately unbalanced allocations, all with a total fleet size of 50 shuttles. As shown in Table 4, the uniform configuration consistently outperforms all unbalanced alternatives across multiple performance metrics.

Table 4. Sensitivity analysis of depot allocation strategies.

Allocation	Total Objective ($\times 10^3$)	Evacuation Completion Time (min)
(17, 17, 16)	212.8	106.8
(25, 15, 10)	218.6	112.4
(30, 10, 10)	222.5	118.7
(10, 25, 15)	217.9	112.1
(15, 10, 25)	219.3	113.5

The unbalanced configurations, such as (25, 15, 10), (30, 10, 10), (10, 25, 15), and (15, 10, 25), exhibit total cost increases ranging from 2.3% to 4.6% compared to the uniform solution. This cost penalty stems primarily from two sources: (i) increased deadhead travel distances, as shuttles from overloaded depots must traverse longer distances to serve pickup zones on the opposite side of the venue; and (ii) imbalanced depot utilization, where certain depots become bottlenecks due to excessive dispatch requests while others remain underutilized.

The performance gap widens as the allocation becomes more skewed, confirming that uniformity is indeed the optimal strategy under our scenario.

It should be emphasized that this result is contingent upon the relatively symmetric spatial distribution and capacity of the three depots around the venue in our scenario, which were selected to provide balanced coverage of the surrounding road network. Under such conditions, a balanced allocation naturally minimizes the maximum travel distance from any depot to any pickup zone, thereby reducing overall deadhead mileage and ensuring equitable workload distribution across depots. If the depots were highly asymmetric in their locations or service conditions, the optimal allocation would likely deviate from uniformity. Thus, the optimal upper-level solution is $R_1^* = 17$, $R_2^* = 17$, $R_3^* = 16$, totalling 50 shuttles, which is adopted for all subsequent lower-level experiments.

6.2.2. Lower-Level Real-Time Performance Comparison

Table 5 reports the key performance metrics for the proposed two-layer UARL-CD and the three comparison strategies. The shuttle-only UARL-CD serves as a comparison strategy: if shuttles alone could achieve near-optimal performance, the complexity of coordinating bus detours would be unnecessary. The FID strategy, which uses both modes but with fixed rules, reveals the value of intelligent, adaptive coordination over simply having access to both vehicle types. The standard DQN, which employs the same state representation, action space, and neural network architecture as UARL-CD, but estimates only the expected Q-value rather than the full return distribution. This baseline is included to isolate and quantify the specific contribution of the distributional value representation in our framework.

Table 5. Average performance comparison across methods.

Algorithm	Total Objective ($\times 10^3$)	Evacuation Completion Time (min)	Avg. Passenger Waiting Time (min)
Shuttle-only	213.1 ± 4.8	155.8 ± 5.2	14.6 ± 0.8
DO	166.4 ± 3.1	126.1 ± 4.1	9.9 ± 0.6
FID	178.8 ± 3.6	133.4 ± 4.5	11.6 ± 0.7
Standard DQN	168.3 ± 2.9	118.5 ± 3.8	9.3 ± 0.5
UARL-CD	163.3 ± 2.6	106.8 ± 3.2	7.5 ± 0.4

Note: The total lower-level objective (a weighted sum of multiple cost components as defined in Section 4.10) is the quantity actually optimized by the RL agent, while the evacuation completion time and average passenger waiting time are reported as intuitive physical metrics to facilitate interpretation.

The proposed two-layer framework significantly outperforms all comparison methods. Compared to the shuttle-only UARL-CD, it reduces evacuation completion time by 31.5% and average passenger waiting time by 48.6%. The shuttle-only UARL-CD, despite using the same RL framework and the same 50-vehicle fleet, performs significantly worse than the full two-layer model. Shuttles must travel from depots to pickup points, then to hubs, and return to depots before being re-dispatched, which severely limits their effective throughput.

Compared to the FID strategy, which also uses both buses and shuttles but with fixed rules—our method reduces evacuation time by 19.9% and waiting time by 35.3%. This gap highlights the critical value of adaptive coordination: simply having access to both modes is insufficient; the ability to dynamically decide which vehicle goes where and for how long is what drives performance gains.

Compared to the DO model, which optimizes both modes using expected values, our UARL-CD achieves 15.3% lower evacuation time and 24.3% lower passenger waiting

time. The performance gap stems from the DO model's inability to adapt to stochastic real-time events; its decisions, made with deterministic travel time estimates, often become suboptimal when actual travel times deviate from expectations.

UARL-CD outperforms Standard DQN by 3.0% in total objective, reducing evacuation completion time from 118.5 to 106.8 min (9.9%). This performance gap directly quantifies the benefit of learning the full return distribution under travel time uncertainty. Standard DQN, which effectively optimizes for the mean performance, fails to account for the variance and tail risks inherent in stochastic travel times, leading to policies that are less robust when actual conditions deviate from expectations. In contrast, the distributional representation in UARL-CD enables the agent to capture the trade-off between mean and variance, resulting in more reliable and efficient dispatching decisions.

Furthermore, the relatively small standard deviations observed for UARL-CD across all metrics (e.g., evacuation completion time: ± 3.2 min; total objective: $\pm 2.6 \times 10^3$) indicate that the learned policy produces stable and reliable performance regardless of the specific stochastic realizations of travel times. In contrast, the shuttle-only and FID baselines exhibit larger standard deviations, reflecting their higher sensitivity to random variations in the environment. This consistency analysis confirms that the superior performance of UARL-CD is not only statistically significant but also consistently achievable across different random conditions.

6.2.3. Sensitivity to Travel Time Uncertainty

To assess the robustness of the proposed framework under varying levels of travel time variability, we conduct a sensitivity analysis by varying the standard deviation of travel time uncertainty σ from 0.10 to 0.25. Note that σ is not an optimized parameter but a preset environmental attribute representing the level of travel time stochasticity in the simulation. The sensitivity analysis is designed to evaluate how performance degrades as environmental uncertainty intensifies. Table 6 reports the total lower-level objective values for the proposed UARL-CD and the four alternative strategies under different σ levels.

Table 6. Total objective under different travel time uncertainty levels.

Algorithm	$\sigma = 0.10$	$\sigma = 0.15$	$\sigma = 0.20$	$\sigma = 0.25$
Shuttle-only	208.2 $\pm$ 4.5	213.1 $\pm$ 4.8	221.5 $\pm$ 5.2	232.2 $\pm$ 5.8
DO	162.2 $\pm$ 2.9	166.4 $\pm$ 3.1	171.9 $\pm$ 3.6	179.0 $\pm$ 4.2
FID	174.9 $\pm$ 3.4	178.8 $\pm$ 3.6	187.0 $\pm$ 4.1	196.5 $\pm$ 4.9
Standard DQN	163.8 $\pm$ 2.7	168.3 $\pm$ 2.9	172.5 $\pm$ 3.3	179.1 $\pm$ 3.8
UARL-CD	160.7 $\pm$ 2.4	163.3 $\pm$ 2.6	165.3 $\pm$ 2.8	167.1 $\pm$ 3.1

The proposed UARL-CD exhibits the strongest robustness, with only 4.0% degradation when σ increases from 0.10 to 0.25 ($(167.1 - 160.7)/160.7 \times 100\% = 4.0\%$). This near-flat trajectory confirms that the agent does not over-fit to a single noise level; it adapts via the uncertainty state component and stochastic training exposures.

In contrast, all comparison methods deteriorate substantially. DO shows a 10.4% increase, as its deterministic plans become increasingly suboptimal when actual travel times deviate from expectations. FID suffers the largest degradation (12.4%), because its fixed 8-min dispatch rules and rigid hub assignments cannot adjust to changing conditions. The shuttle-only variant shows an 11.5% increase, confirming that without bus detours, the system lacks sufficient flexibility to compensate for travel time fluctuations.

To isolate the source of this robustness, the introduced standard DQN shares the same state representation (including uncertainty-related features such as moving averages of travel times) and the same stochastic training environment as our UARL-CD. Both

algorithms are trained under identical realizations of travel time randomness and ride-hailing supply fluctuations, controlled by the same set of random seeds. The two variants differ only in their value representation: standard DQN estimates expected Q-values, while UARL-CD learns a full categorical distribution over returns. As shown in Table 5, under low uncertainty ($\sigma = 0.10$), the performance gap between UARL-CD and standard DQN is marginal. However, as uncertainty intensifies to $\sigma = 0.25$, standard DQN degrades by 9.3%, while UARL-CD degrades by only 4.0%. This divergence confirms that the robustness advantage originates primarily from the distributional value representation—the Categorical DQN’s ability to model the full return distribution.

6.2.4. Sensitivity to Weighting Coefficients

To examine the impact of the weighting coefficients in the lower-level objective on the performance of our method, we conduct a sensitivity analysis by systematically varying the four groups of weights (passenger waiting time, regular service disruption, shuttle operation, and hub congestion). For each group, we test both increased ($\times 1.5$) and decreased ($\times 0.5$) scenarios while keeping all other weights at their baseline values.

As shown in Table 7, the total objective of UARL-CD shifts predictably in response to weight adjustments. When an operator assigns higher priority to passenger waiting time (Wait + 50%), the agent reduces evacuation completion time at the expense of increased operational costs. Conversely, when regular service disruption is emphasized (Reg + 50%), the system imposes stricter limits on bus detours, resulting in slightly longer passenger waiting times but reduced impact on regular transit operations. Shuttle operation and hub congestion weights show similar trade-off effects.

Table 7. Sensitivity analysis of weighting coefficients.

Scenario	Weight Variation	Total Objective ($\times 10^3$)	Evacuation Completion Time (min)
Wait + 50%	$\alpha \times 1.5$	198.3	102.8
Wait – 50%	$\alpha \times 0.5$	140.5	110.1
Disruption + 50%	$\beta \times 1.5$	192.2	111.2
Disruption – 50%	$\beta \times 0.5$	151.5	103.7
Shuttle + 50%	$\eta_{sh} \times 1.5$	165.1	106.8
Shuttle – 50%	$\eta_{sh} \times 0.5$	160.2	105.7
Hub + 50%	$\theta_{metro}, \theta_{ride} \times 1.5$	168.4	105.5
Hub – 50%	$\theta_{metro}, \theta_{ride} \times 0.5$	165.9	108.2

Crucially, the observed variations are consistent with the direction of weight adjustments, confirming that the agent effectively adapts its dispatching policy to the specified priorities. This demonstrates that the UARL-CD framework offers operators a practical tool for multi-criteria decision-making.

6.2.5. Sensitivity to Demand Magnitude and Distribution

To assess whether the learned policy is overly specialized to the specific demand configuration of the case study, we conduct two additional sensitivity analyses that vary key demand-related parameters without retraining the UARL-CD agent.

Demand magnitude shift: We vary the total attendance from the nominal 16,000 passengers to 20,000 (+25%) and 24,000 (+50%), representing a significantly more congested event scenario. As shown in Table 8, the UARL-CD policy maintains its superiority over all baselines across these demand levels, confirming that the learned coordination principles are not sensitive to the absolute demand magnitude.

Table 8. Sensitivity to demand magnitude and destination distribution.

Scenario	Shuttle-Only	DO	FID	Standard DQN	UARL-CD
16,000 passengers	213.1 $\pm$ 4.8	166.4 $\pm$ 3.1	178.8 $\pm$ 3.6	168.3 $\pm$ 2.9	163.3 $\pm$ 2.6
20,000 passengers	245.3 $\pm$ 5.1	188.1 $\pm$ 3.5	202.6 $\pm$ 4.1	191.2 $\pm$ 3.4	184.5 $\pm$ 3.0
24,000 passengers	278.5 $\pm$ 5.5	210.4 $\pm$ 3.8	226.8 $\pm$ 4.6	214.4 $\pm$ 3.7	206.3 $\pm$ 3.4
Metro-dominated	208.6 $\pm$ 4.7	163.1 $\pm$ 3.0	175.6 $\pm$ 3.5	165.8 $\pm$ 2.8	160.3 $\pm$ 2.5
Ride-hailing-dominated	218.9 $\pm$ 5.0	168.3 $\pm$ 3.3	181.5 $\pm$ 3.8	171.0 $\pm$ 3.0	165.1 $\pm$ 2.7

Demand destination distribution shift: We alter the destination preference ratio between metro hubs and ride-hailing hubs to 70:30 (metro-dominated) and 30:70 (ride-hailing-dominated). These shifts test the agent's ability to adapt to different passenger OD patterns without retraining. The results (Table 6) show that UARL-CD consistently outperforms all baselines across both scenarios. This stability indicates that the agent learns robust resource allocation principles, rather than memorizing the specific destination preferences of the case study.

These results provide tangible evidence that the proposed framework generalizes beyond the specific demand configuration of the case study, offering preliminary validation of its applicability to varied event conditions.

6.2.6. Computational Performance

Table 9 summarizes the computational burden of each algorithm. The proposed UARL-CD shifts the majority of computation offline, requiring approximately 105 h of total training time. This includes: (i) the upper-level fleet search over candidate configurations with $M = 5$ lower-level runs each, taking approximately 90 h; and (ii) the final lower-level policy training with the optimal 50-shuttle fleet, taking approximately 15 h. Once trained, the online decision time is only 3.25 ± 0.3 s per time step, well within the 2-min decision interval, making it suitable for real-time deployment.

Table 9. Computational performance comparison.

Algorithm	Online Decision Time (s)	Offline Training Time (h)
Shuttle-only	3.18 $\pm$ 0.3	15
DO	18.8 $\pm$ 1.2	N/A
FID	<0.1	N/A
UARL-CD	3.25 $\pm$ 0.3	105 (upper 90 + lower 15)

Note: The upper-level search is performed once before the event, while the final policy is deployed in real time. The shuttle-only variant requires only the lower-level training (15 h), as the fleet is fixed.

The deterministic optimization (DO) model requires 18.8 ± 1.2 s per online decision, which is numerically within the 120-s decision interval for the tested instance. However, this computation time is instance-specific and grows non-linearly with problem size (e.g., number of bus lines, pickup zones, and depots), whereas UARL-CD inference scales linearly with the number of vehicles and remains nearly constant regardless of network complexity. Moreover, the MILP must be re-solved from scratch at each time step as new information arrives, imposing a recurring online burden that grows with the problem size. The FID policy is instantaneous but yields the worst performance among the two-mode strategies, as it cannot adapt to real-time conditions. The shuttle-only UARL-CD has identical online inference time because the network architecture is the same.

The key advantage of the proposed approach is the separation of planning and execution. The expensive upper-level optimization is performed offline, allowing the agent to exploit the full computational resources available in a planning office. Once the event

begins, the lower-level policy executes rapid neural network inference, enabling real-time responses to dynamic conditions, including unexpected congestion, fluctuating demand, and variable hub supply. This makes the framework practically deployable for transit agencies, where pre-event planning can be conducted well in advance, while operations must be agile and responsive.

The 3.25-s reported above refers exclusively to neural network inference—the time required for the trained policy to compute a dispatching decision from the current state observation. This is the component over which the algorithm has direct control.

In practice, the end-to-end latency of a real-time traffic management system comprises three additional components: (i) state sensing and GPS data transmission, (ii) data aggregation and pre-processing, and (iii) command dissemination to vehicles. Modern GPS-based fleet management systems typically report vehicle positions at 1-s intervals, with communication latency in urban V2X environments ranging from 30–50 ms under typical conditions to approximately 100–150 ms under peak load. Data aggregation and command dissemination add several seconds in practice, but these operations can be pipelined or executed in parallel with inference.

Even under conservative estimates—allowing 10 s for state sensing and GPS transmission, 5 s for data aggregation, 3.25 s for inference, and 5 s for command dissemination—the total end-to-end latency remains approximately 23 s, well within the 2-min decision interval. Moreover, the 2-min interval is a planning re-optimization cycle rather than a hard real-time deadline: if a single decision packet experiences delay, previously issued dispatch commands remain valid and physically executable (vehicles continue their current trips). The system only requires that the average decision latency remains below the inter-arrival time of significant demand changes.

We therefore maintain that the 3.25-s inference time, together with realistic allowances for communication and sensing delays, supports the practical deployability of the proposed framework in existing traffic management center infrastructures. For transit agencies, the key operational requirement is to ensure reliable V2X communication and efficient data pipelines—engineering challenges that are well within the capabilities of modern intelligent transportation systems.

7. Conclusions

This paper addresses the real-time multi-modal evacuation problem for large-scale public events under travel time uncertainty. We propose a two-layer optimization framework that coordinates in-service buses temporarily diverted from existing lines and dedicated shuttle vehicles pre-positioned at depots. The upper layer determines the pre-event fleet sizing, balancing rental costs against expected operational performance, while the lower layer makes real-time dispatching decisions for both modes. The problem is formulated as a stochastic optimization with coupled strategic and tactical decisions, capturing the inherent uncertainty in urban travel times and hub conditions.

To solve this challenging problem, we develop an Uncertainty-Aware Reinforcement Learning algorithm with Categorical DQN (UARL-CD). The agent learns a robust dispatching policy through a reward function directly aligned with the lower-level objective—minimizing evacuee waiting time, regular passenger disruption, dwell time penalties, and hub congestion. The algorithm explicitly accounts for travel time uncertainty through distributional value representation and stochastic training, while an action masking mechanism enforces operational constraints during both training and execution. This design enables the agent to make feasible, adaptive decisions in real time.

We validate the proposed framework through simulation experiments based on a realistic stadium evacuation scenario with 16,000 passengers. The results demonstrate that

the optimal pre-event fleet consists of 50 shuttles (distributed as 17, 17, 16 across three depots). The learned policy achieves a 31.6% reduction in evacuation completion time and a 48.4% reduction in average passenger waiting time compared to using shuttles alone. Compared to deterministic optimization, the proposed method reduces evacuation time by 14.9% and waiting time by 24.3%, while also outperforming fixed-interval dispatching by 19.8% and 35.4%, respectively. The policy remains robust under varying travel time uncertainty, exhibiting only a 4.0% performance degradation—far lower than the 10.6–11.9% degradations observed in other methods. Online decisions are executed in approximately 3.25 s per time step, well within the 2-min decision interval, making the framework practically deployable.

While this study provides a comprehensive solution for the single-venue evacuation scenario, we acknowledge several important directions for future extension. First, scaling the framework to larger, multi-node urban networks would require hierarchical state abstraction. For instance, grouping multiple pickup zones and depots into regional clusters to maintain sample efficiency. Second, the demand profiles of different event types (e.g., concerts vs. sports events vs. festivals) vary not only in magnitude but also in temporal dispersion and modal preferences. Incorporating heterogeneous demand patterns and adaptive demand prediction models would further enhance the framework's generalizability. Third, the current framework assumes known passenger arrival patterns; integrating real-time demand forecasting using streaming data would enable proactive rather than reactive dispatching. These extensions represent our immediate research priorities, and the modular architecture of UARL-CD allows for seamless integration of these enhancements.

Author Contributions: Conceptualization, W.W.; Methodology, W.W.; Validation, L.H.; Investigation, X.X. and L.H.; Writing—original draft, W.W.; Writing—review & editing, X.X. and B.Y.; Supervision, B.Y.; Funding acquisition, W.W. and B.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported in National Natural Science Foundation of China (52302403 and 52225209), the Fundamental Research Funds for the Central Universities (3132026179).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Xin, X.; Wang, S.; Zhang, T. Truck-drone supported humanitarian relief logistics network design: A two-stage distributionally robust optimization approach. *Transp. Res. Part C Emerg. Technol.* **2025**, *178*, 105231. [\[CrossRef\]](#)
2. Teng, J.; Chen, Y.; Zhao, M.; Yang, X.; Shen, B. Scheduling of feeder vehicles for intermodal services for special events. *Transp. Res. Rec.* **2011**, *2216*, 165–173.
3. Tian, Y.; Yang, D.; Wang, J.; Lin, W.; Jiang, Y.; An, S. Designing and planning sustainable customized bus service for departing attendees of planned special events: A two-phase methodology integrating data-driven and demand-responsive. *Sustain. Cities Soc.* **2023**, *96*, 104630.
4. Santanam, T.; Trasatti, A.J.; Hentenryck, P.V.; Zhang, H. Public transit for special events: Ridership prediction and train scheduling. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 8387–8403. [\[CrossRef\]](#)
5. Yin, Y.; Xu, X.; Wang, D.; Yu, Y.; Cheng, T. Two-stage recoverable robust optimization for an integrated location-allocation and evacuation planning problem. *Transp. Res. Part B Methodol.* **2024**, *182*, 102906. [\[CrossRef\]](#)
6. Su, H.; Wang, X.; Liu, W.; Zhang, X.; Xu, M. Modelling and accelerating the passenger evacuation process in a bi-modal system with bus and e-hailing modes after mass gathering events. *Travel Behav. Soc.* **2024**, *35*, 100713. [\[CrossRef\]](#)

7. Jiang, J.; Tu, W.; Kong, H.; Zeng, W.; Zhang, R.; Konecny, M. Large-scale urban multiple-modal transport evacuation model for mass gathering events. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 23059–23069. [\[CrossRef\]](#)
8. Zhang, Y.; Meng, Z.; Zheng, Y.; Ralescu, A. Schedule optimization under fuzzy constraints of vehicle capacity. *Fuzzy Optim. Decis. Mak.* **2019**, *18*, 131–150. [\[CrossRef\]](#)
9. Kliwer, N.; Mellouli, T.; Suhl, L. A time-space network based exact optimization model for multi-depot bus scheduling. *Eur. J. Oper. Res.* **2006**, *175*, 1616–1627. [\[CrossRef\]](#)
10. Gu, W.; Yu, J.; Ji, Y.; Zheng, Y.; Zhang, H.M. Plan-based flexible bus bridging operation strategy. *Transp. Res. Part C Emerg. Technol.* **2018**, *91*, 209–229. [\[CrossRef\]](#)
11. Li, L.; Lo, H.K.; Xiao, F. Mixed bus fleet scheduling under range and refueling constraints. *Transp. Res. Part C Emerg. Technol.* **2019**, *104*, 443–462. [\[CrossRef\]](#)
12. Liu, T.; You, H.; Gkotsalitis, K.; Cats, O. Human-machine collaborative decision-making approach to scheduling customized buses with flexible departure times. *Transp. Res. Part A Policy Pract.* **2024**, *187*, 104184. [\[CrossRef\]](#)
13. Li, X.; An, X.; Zhang, B. Minimizing passenger waiting time in the multi-route bus fleet allocation problem through distributionally robust optimization and reinforcement learning. *Comput. Oper. Res.* **2024**, *164*, 106568. [\[CrossRef\]](#)
14. Chen, K.; Yi, X.; Xin, X.; Shan, H. Spatial evolution of liner bunkering hubs in Asia–Europe: Green fuel adoption under demand and fuel price uncertainty. *Transp. Policy* **2025**, *178*, 103956. [\[CrossRef\]](#)
15. Xin, X.; Zhang, T.; Wang, X.; He, F.; Wu, L. Risk-averse distributionally robust optimization for construction waste reverse logistics with a joint chance constraint. *Comput. Oper. Res.* **2025**, *173*, 106829. [\[CrossRef\]](#)
16. Yan, S.; Tang, C.-H. An integrated framework for intercity bus scheduling under stochastic bus travel times. *Transp. Sci.* **2008**, *42*, 318–335. [\[CrossRef\]](#)
17. Shen, Y.; Xu, J.; Li, J. A probabilistic model for vehicle scheduling based on stochastic trip times. *Transp. Res. Part B Methodol.* **2016**, *85*, 19–31. [\[CrossRef\]](#)
18. Chu, J.C.; Chen, A.Y.; Shih, H. Stochastic programming model for integrating bus network design and dial-a-ride scheduling. *Transp. Lett.* **2022**, *14*, 245–257. [\[CrossRef\]](#)
19. Zhang, X.; Sun, H.; Qu, Y.; Wu, J. Dynamic optimal evacuation planning for multimodal urban traffic network. *J. Transp. Geogr.* **2025**, *129*, 104424. [\[CrossRef\]](#)
20. Yang, M.; Ma, H.; Li, X.; Shang, C.; Shen, Q. Bus bridging for rail disruptions: A distributionally robust fuzzy optimization approach. *IEEE Trans. Fuzzy Syst.* **2022**, *31*, 500–510. [\[CrossRef\]](#)
21. An, X.; Li, X.; Zhang, B. Flexible scheduling of customized bus for green mega-events: A distributionally robust optimization approach. *Comput. Oper. Res.* **2026**, *185*, 107249. [\[CrossRef\]](#)
22. Zhang, Y.; Bai, R.; Qu, R.; Tu, C.; Jin, J. A deep reinforcement learning based hyper-heuristic for combinatorial optimisation with uncertainties. *Eur. J. Oper. Res.* **2022**, *300*, 418–427. [\[CrossRef\]](#)
23. Liu, M.; Xin, X.; Wang, X.; Zhang, T.; Chen, K. Dual-channel slot sales strategy for container liner shipping companies with blockchain technology adoption. *Transp. Policy* **2025**, *162*, 200–220. [\[CrossRef\]](#)
24. Kalatzantonakis, P.; Sifaleras, A.; Samaras, N. A reinforcement learning-variable neighborhood search method for the capacitated vehicle routing problem. *Expert Syst. Appl.* **2023**, *213*, 118812. [\[CrossRef\]](#)
25. Karakostas, P.; Sifaleras, A. Learning-assisted improvements in adaptive variable neighborhood search. *Swarm Evol. Comput.* **2025**, *94*, 101887. [\[CrossRef\]](#)
26. Qin, W.; Zhuang, Z.; Huang, Z.; Huang, H. A novel reinforcement learning-based hyper-heuristic for heterogeneous vehicle routing problem. *Comput. Ind. Eng.* **2021**, *156*, 107252. [\[CrossRef\]](#)
27. Bellemare, M.; Dabney, W.; Munos, R. A Distributional Perspective on Reinforcement Learning. *arXiv* **2017**, arXiv:1707.06887. [\[CrossRef\]](#)

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.