

Article

LLM-Driven Multi-Agent Coordinated Control for Urban Rail Transit Disruption Response

Hao Wu ¹ and Xiaoqing Zeng ^{2,*}¹ School of Government, Peking University, Beijing 100091, China; haoareu@stu.pku.edu.cn² College of Transportation Engineering, Tongji University, Shanghai 201804, China

* Correspondence: zengxq@tongji.edu.cn

Abstract

Short section delays in metro corridors can spread quickly through transfer hubs and return to operations as longer dwell times, local crowding, and downstream delay. Existing disruption–recovery studies often optimize train regulation, station control, and passenger guidance in separate modules, which makes it difficult to assemble a linked response that is both operationally coherent and auditable. This study develops an LLM-driven multi-agent decision framework for abnormal response in urban rail transit. Role-specific agents generate train-, station-, transfer-, and information-side measures; a safety-review agent removes infeasible actions; and an arbitration agent produces a structured *DecisionCard* for simulation backfeeding. A corridor-level mesoscopic model is built for the Shanghai Metro Line 2 section between East Nanjing Road and Lujiazui, covering four key transfer hubs. The evaluation combines five online LLM cases with a broader 3-by-3 delay–demand matrix and transfer-capacity sensitivity tests. Relative to B4-SOP-Limited, Online-MA-Rolling reduces total delay by 3.5–48.8% and peak transfer queues by 2.0–90.3% in the five online cases. Across the nine delay–demand combinations, the multi-agent strategy reduces total delay by 58.9% on average and transfer-queue peaks by 32.6% on average compared with B4. When transfer service capacity is reduced by 25%, however, queue relief collapses and the delay advantage can reverse under high load. The main contribution of the framework is, therefore, not the use of LLMs alone, but the production of linked, reviewable, and simulation-testable response plans within a clearly bounded operating envelope.

Keywords: urban rail transit; abnormal response; large language model; multi-agent collaboration; transfer congestion

MSC: 68T42; 68T50; 90B20; 90C27



Academic Editor: Yu-Wang Chen

Received: 9 June 2026

Revised: 8 July 2026

Accepted: 13 July 2026

Published: 6 August 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

As urban rail systems become increasingly intensive, interconnected, and transfer-dependent, operational risk is no longer confined to train movement alone, but extends to station organization, transfer channels, and passenger path choice. In peak periods, once a short section delay occurs, the arrival pattern of subsequent trains and the temporal distribution of transfer passengers can be altered within a short time window. This disturbance may then be amplified through major transfer hubs, producing queue accumulation, local crowding, and additional operational instability. Real-time recovery in rail and metro systems has long focused on timetable rescheduling, conflict resolution, and dispatch recovery, forming a relatively mature body of models and algorithms [1–3]. In recent years,

more studies have begun to jointly consider train operations, passenger flow, and station congestion, for example, by embedding passenger retention, platform capacity, and transfer continuity into integrated disruption-management frameworks [4–6].

Within metro networks, transfer stations function not only as capacity interfaces but also as key nodes where abnormal impacts can be amplified. Existing studies have explored network-level passenger-flow coordination, linked inflow control, and joint optimization of train timetables and station restrictions, providing important theoretical and practical support for congestion management under disturbed conditions [7–9]. Recent work further shows that multi-line coordinated adjustment, rolling-horizon timetable recovery, and collaborative station control can effectively mitigate congestion under short-term disruption [10–13]. At the passenger information level, dynamic route guidance through display systems can reduce pressure on affected stations and improve trip completion [14]. At the station level, measures such as escalator direction control, temporary barriers, and gate throttling directly affect density distribution in halls, platforms, and transfer channels, while simulation-based adaptive optimization has provided a quantitative basis for field organization [15]. Multi-agent simulation has also been used in domestic research to model trains, lines, networks, and passengers explicitly for flow-control analysis [16]. In parallel, congestion-propagation studies at rail hubs have characterized inter-node risk transmission through Bayesian networks and queueing logic [17]. Related transit-network research further indicates that global centrality and local network structure can shape spatial exposure risk in Beijing, and that the long-term evolution of metro-network topology across Chinese cities changes the structural conditions under which transfer pressure emerges [18,19]. Recent risk-topology modeling of urban rail safety relationships further highlights the networked nature of operational risk, while machine-learning-based departure-delay prediction using integrated real-time transit data shows the value of data-driven operational management for resilience analysis [20,21]. These studies jointly suggest that short-term disturbance should be analyzed within a coupled system of operations, station flow, and passenger routing rather than as a pure timetable deviation problem.

Despite this progress, coordinated abnormal response still faces several limitations. Train-side regulation and station-side control are often treated in separate models, while passenger guidance is frequently introduced as an external assumption rather than an endogenous decision component. As a result, the linked effect of train operations, passenger organization, and information intervention is not fully represented. In addition, even when optimization models produce numerically desirable solutions, practical emergency handling is constrained by safety rules, staffing conditions, resource accessibility, and responsibility boundaries across operational roles. A mathematically attractive solution, therefore, does not automatically become an executable operating order.

Large language models (LLMs) have shown strong potential in reasoning, planning, and tool use [22–24], and multi-agent LLM frameworks are increasingly used for task decomposition, collaboration, and review [25,26]. Transportation-oriented survey studies also indicate that LLMs may contribute to planning, management, and decision support, but their deployment in safety-critical transport settings still depends on well-defined interfaces, accountability, interpretability, and operational safeguards [27–29]. In urban rail abnormal response, letting an LLM directly output operating commands would introduce obvious risks, such as infeasible actions, safety conflicts, and unclear responsibility.

LLMs are not treated here as autonomous dispatchers. They are placed inside a decision chain with explicit role division, safety review, and simulation backfeeding. The focus is the coupled amplification of section delay and transfer congestion. The study builds a corridor-level mesoscopic simulation environment and uses structured state snapshots, linked control actions, and reviewed *DecisionCards* to support coordinated intervention

across train regulation, station organization, transfer coordination, and passenger information guidance. The intent is to strengthen joint response capability, not to replace the existing command system.

Compared with traditional rule-based control, the proposed framework does not simply trigger a fixed measure from a prewritten table; it uses role-specific agents to assemble a linked response from the current coupled state. Compared with optimization-based dispatching, it does not claim to replace rigorous timetable-recovery models; instead, it converts a complex abnormal state into a compact, reviewable, and simulation-testable operating plan. Compared with general multi-agent conversation, the agents here are constrained by an explicit action library, operational responsibilities, safety screening, and a structured output interface. The role of the LLM is, therefore, threefold: to generate local candidate measures under role prompts, to support cross-role plan synthesis, and to express the final plan in a traceable *DecisionCard*.

From the perspective of applied mathematics, the paper contributes a compact modeling chain that links state representation, normalized risk evaluation, constrained combinatorial action selection, and mesoscopic simulation backfeeding. The mathematical objects are intentionally interpretable: the state vector describes delay, queue, density, and operability; the risk function maps heterogeneous indicators to a common trigger scale; the arbitration problem formalizes the tradeoff among expected benefit, action conflict, and execution burden; and the backfeeding equations close the loop between decision and state evolution. This positioning is important because the proposed LLM component is used as a bounded decision generator inside an explicit mathematical and simulation structure, not as an unconstrained text interface.

The contribution of the paper lies in three parts. It first formalizes the feedback process through which a short section delay turns into transfer accumulation, dwell extension, and further delay, and translates that process into a simulation-ready representation. It then develops a role-based multi-agent mechanism in which local proposals, conflict checks, and arbitration are combined into a structured *DecisionCard* that can be traced and backfed. Finally, it evaluates the framework at two levels: visible online cases are used to expose the actual LLM decision chain, while a broader set of cross-condition and sensitivity experiments is used to test how stable the observed gains remain outside a handful of showcased snapshots.

The remainder of the paper is organized as follows. Section 2 presents the multi-agent decision framework, the coupled mesoscopic simulation model, the online case-selection logic, and the evaluation metrics. Section 3 reports the online decision process, baseline comparisons, cross-condition validation, sensitivity analysis, and single-agent comparison. Section 4 discusses the relationship with existing studies, deployment boundaries, ethical and regulatory issues, and transferability. Section 5 concludes the paper.

2. Materials and Methods

2.1. Overall Framework

The overall framework is shown in Figure 1. It is organized around state sensing, collaborative control, review and arbitration, and simulation backfeeding under abnormal-response conditions in urban rail transit. Abnormal events, train-operation states, station congestion, SOP rules, and executable control measures are first transformed into structured inputs. Role-specific agents then generate local proposals according to their assigned responsibilities. Unlike open-ended question answering, the LLMs in this study are embedded in an operational decision chain: proposal agents generate local control candidates, the safety-review agent checks action conflicts and feasibility constraints, and the arbitration agent synthesizes a final linked plan under coverage and implementation-burden

considerations. The output is a structured *DecisionCard*, which is then backfed into the simulation layer to update operating and congestion states for the next decision cycle.

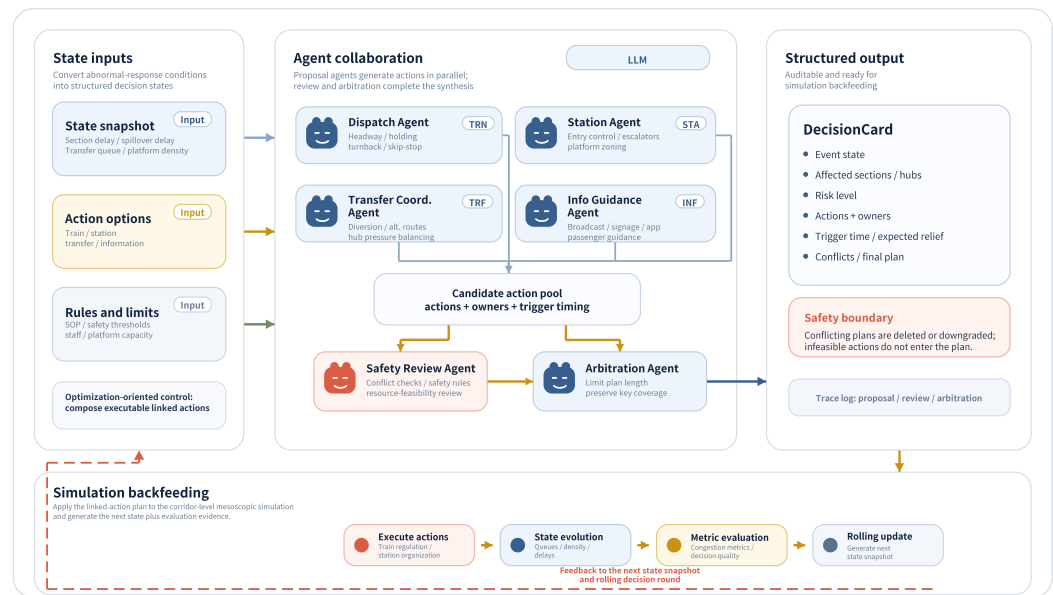


Figure 1. LLM-driven multi-agent collaborative decision framework. Solid arrows indicate information and control flow, dashed arrows indicate review or feedback links, and colors distinguish state inputs, role-specific agents, safety review, arbitration, and simulation backfeeding.

2.2. Abnormal-State Representation and Collaborative Action Generation

Simulation time is discretized at one-minute intervals, with $t \in \{0, 1, \dots, T\}$. At any decision epoch, the system state is written as

$$s_t = (d_t^s, d_t^o, q_t^r, q_t^p, \rho_t, m_t, \chi_t), \quad (1)$$

where d_t^s denotes the delay in the directly affected section, d_t^o denotes downstream spillover delay, q_t^r is the transfer-queue size, q_t^p is the platform-queue size, ρ_t is platform density, m_t is the transfer-failure rate, and χ_t collects external constraints such as dispatch operability and information compliance. The risk level at time t is defined as

$$R_t = \omega_1 \frac{d_t^s}{\bar{d}} + \omega_2 \frac{d_t^o}{\bar{d}_o} + \omega_3 \frac{q_t^r}{\bar{q}_r} + \omega_4 \frac{\rho_t}{\bar{\rho}}, \quad \sum_{k=1}^4 \omega_k = 1, \quad (2)$$

where $\bar{d}$, $\bar{d}_o$, $\bar{q}_r$, and $\bar{\rho}$ are reference thresholds for the corresponding indicators. According to R_t , the current state is classified into low-, medium-, and high-risk levels to trigger control actions of different intensity. These thresholds are not universal safety constants. In the reported experiments, $\bar{d}$, $\bar{d}_o$, and $\bar{q}_r$ are selected as scenario-calibration anchors that mark the transition from ordinary fluctuation to abnormal amplification in the corridor testbed, while $\bar{\rho}$ follows the platform safety-density bound used later in the parameter table. The risk weights are set equally, $\omega_1 = \omega_2 = \omega_3 = \omega_4 = 0.25$, to avoid giving priority to only train-side or passenger-side indicators before the simulation comparison. In practical deployment, these thresholds and weights should be recalibrated with local operation logs, safety rules, and expert review.

The linked action space is defined as

$$\mathcal{A} = \mathcal{A}^{\text{trn}} \cup \mathcal{A}^{\text{sta}} \cup \mathcal{A}^{\text{trf}} \cup \mathcal{A}^{\text{inf}}, \quad (3)$$

where $\mathcal{A}^{\text{tn}}$ denotes train-side actions, $\mathcal{A}^{\text{sta}}$ station-side actions, $\mathcal{A}^{\text{trf}}$ transfer-coordination actions, and $\mathcal{A}^{\text{inf}}$ information-guidance actions. Their composition and responsibility mapping are listed in Table 1.

Table 1. Composition of the collaborative control action set.

Action Category	Responsible Agent	Count	Main Control Objects
Train-side actions	Dispatch agent	5	Headway adjustment, connection holding, skip-stop candidates, turnback candidates, and downstream spillover-delay mitigation
Station-side actions	Station agent	7	Platform inflow, gates, escalators, passageways, temporary buffering, and field staffing organization
Transfer coordination	Transfer coordination agent	3	Upstream offloading, parallel-hub diversion, and transfer-pressure redistribution
Information guidance	Information guidance agent	5	Broadcast, app notification, signage, staggered-travel advice, and destination-group guidance

Concrete actions are specified at the action-library level rather than left to free-form LLM output. For example, train-side actions include headway rebalancing, selective connection holding, and skip-stop candidates; station-side actions include entrance metering, gate throttling, escalator reorganization, and temporary platform zoning; transfer-coordination actions include upstream offloading and parallel-hub diversion; and information actions include broadcast announcements, app messages, station signage, and destination-group guidance. Thus, $\mathcal{A}$ defines the feasible response vocabulary, while the LLM agents choose and combine measures inside that bounded vocabulary.

The local proposal function of agent i is written as

$$\mathcal{P}_{i,t} = \pi_i(s_t, \mathcal{A}_i, \mathcal{G}_i), \quad i \in \{\text{D, S, T, I}\}, \quad (4)$$

where $\mathcal{A}_i$ is the action subset accessible to agent i , and $\mathcal{G}_i$ contains role responsibility, operational rules, and prompting logic. Operationally, $\pi_i(\cdot)$ is implemented as a constrained LLM proposal function: the prompt exposes the current state, the role-specific action subset, and SOP constraints, and the agent must return action identifiers with short reasons and expected effects. It is, therefore, closer to a bounded decision-support policy than to an unconstrained natural-language answer. After merging all proposals, safety filtering is applied:

$$\mathcal{P}_t^{\text{safe}} = \left\{ a \in \bigcup_i \mathcal{P}_{i,t} \mid h(a, s_t) = 1 \right\}, \quad (5)$$

where $h(a, s_t)$ is the safety-feasibility function. If actions a and b are mutually conflicting, then $c_{ab} = 1$; otherwise, $c_{ab} = 0$. The arbitration agent solves the following combinatorial selection problem:

$$\max_x \quad \sum_{a \in \mathcal{P}_t^{\text{safe}}} v_a(s_t) x_a - \lambda_1 \sum_{a,b} c_{ab} x_a x_b - \lambda_2 \sum_a r_a x_a, \quad (6)$$

$$\text{s.t.} \quad \sum_{a \in \mathcal{P}_t^{\text{safe}}} x_a \leq K, \quad (7)$$

$$x_a \in \{0, 1\}, \quad a \in \mathcal{P}_t^{\text{safe}}, \quad (8)$$

where $v_a(s_t)$ is the expected benefit of action a under state s_t , r_a denotes execution burden or resource occupancy, and K is the maximum number of actions permitted in the

final plan. In this study, $v_a(s_t)$ is operationalized through a transparent ordinal rubric rather than a fitted black-box utility model. Each feasible action is scored according to its expected contribution to section-delay relief, downstream-delay relief, transfer-queue relief, and crowding-risk relief under the current risk profile. The burden term r_a combines staffing requirement, passenger disturbance, implementation latency, and resource occupation. All scores are normalized before arbitration; the conflict penalty is set to dominate any single marginal benefit, and the burden penalty is used only to discourage unnecessarily heavy plans. The reported online cases use $K = 4$, matching the operational preference for a compact linked action set that can still cover multiple roles. The arbitration result is written into a structured *DecisionCard*:

$$\mathcal{D}_t = (s_t, \mathcal{H}_t, R_t, \mathcal{A}_t^*, \mathcal{O}_t, \tau_t, \mathcal{C}_t, \mathcal{F}_t), \quad (9)$$

where $\mathcal{H}_t$ denotes the affected sections and hubs, $\mathcal{A}_t^*$ is the final action set, $\mathcal{O}_t$ indicates ownership, τ_t is the trigger time, $\mathcal{C}_t$ records conflict review, and $\mathcal{F}_t$ is the final linked plan.

2.3. Coupled Simulation Model of Operation and Passenger-Flow Evolution

To avoid excessive dependence on fine-grained pedestrian parameters under limited data availability, this study uses a mesoscopic event-driven simulation to represent delay propagation and passenger accumulation. The purpose is not to reproduce every passenger and every train in detail, but to capture the feedback relationship among disturbance, transfer queues, platform crowding, and dwell-time extension. The transfer-queue update is written as

$$q_{t+1}^r = [q_t^r + \lambda_t^r + \alpha_d d_t^s - \mu_t^r(\mathcal{A}_t^*)]_+, \quad (10)$$

where λ_t^r is the regular transfer-arrival rate, $\alpha_d d_t^s$ is the extra transfer accumulation induced by section delay, and $\mu_t^r(\cdot)$ is the transfer-dissipation capacity affected by station control and information guidance. The platform-queue update is

$$q_{t+1}^p = [q_t^p + \lambda_t^p + \alpha_r q_t^r - \mu_t^p(\mathcal{A}_t^*)]_+. \quad (11)$$

The dwell-time extension Δw_t is jointly determined by platform density and queue level:

$$\Delta w_t = \beta_1 \frac{q_t^p}{\mu_t^p + \varepsilon} + \beta_2 [\rho_t - \rho^{safe}]_+, \quad (12)$$

where ρ^{safe} is the platform safety-density threshold. Downstream spillover delay is updated by

$$d_{t+1}^o = [d_t^o + \gamma_w \Delta w_t - \gamma_a e(\mathcal{A}_t^*)]_+, \quad (13)$$

where $e(\mathcal{A}_t^*)$ denotes the recovery contribution of train-side actions. Together, these equations represent the coupled mechanism emphasized in this paper: section delay changes train arrivals and transfer rhythms; transfer and platform accumulation then extend dwell time and weaken operational recovery, which, in turn, feeds back into further delay.

2.4. Online LLM Agents and the DecisionCard Interface

In the online experiment, all agents receive the same state snapshot s_t and the same action set as input, but operate under different role prompts and responsibility boundaries. In this paper, an “agent” is not a simple rule trigger, but a decision proxy oriented toward a specific responsibility, equipped with a relatively independent strategy-generation space and capable of producing local proposals while participating in collaborative review and synthesis. It should also be emphasized that real online LLM invocation occurs only at

critical snapshot time t_0 ; subsequent rolling backfeeding is executed by an offline planner rather than by invoking multiple LLMs every minute. This experimental boundary keeps cost under control and makes the process reproducible, while avoiding exaggerated claims that the LLM directly and continuously commands the metro system.

The *DecisionCard* generation process is shown in Table 2.

Table 2. Multi-agent *DecisionCard* generation process.

Input	State snapshot s_t , linked action set $\mathcal{A}$, operational rule set $\mathcal{G}$, and conflict matrix C .
Step 1	Dispatch, station, transfer-coordination, and information-guidance agents generate local candidate measures $\mathcal{P}_{i,t}$ within their respective control scopes. Candidate measures are merged, and the safety-review agent screens out mutually conflicting actions, infeasible actions, and actions violating safety rules, yielding $\mathcal{P}_t^{safe}$.
Step 2	
Step 3	The arbitration agent selects the final action set $\mathcal{A}_t^*$ according to risk level, category coverage, and execution burden.
Step 4	Event state, affected sections, risk level, owners, trigger time, conflict notes, and the final linked plan are written into a structured <i>DecisionCard</i> .
Step 5	The <i>DecisionCard</i> is backfed into the simulation layer to evaluate operational recovery, transfer congestion, and decision quality.

2.5. Safety Review and Simulation Backfeeding

In a safety-critical system, the goal of decision generation is not to maximize the number of actions, but to ensure that candidate actions satisfy resource, rule, and safety boundaries. The safety-review stage is, therefore, not intended to replace human responsibility, but to block infeasible or high-risk LLM outputs before they enter simulation and the final action order. Let u_a denote resource accessibility, b_a the safety-redline condition, and o_a the SOP compliance of action a . Then

$$h(a, s_t) = \mathbb{I}(u_a = 1)\mathbb{I}(b_a = 1)\mathbb{I}(o_a = 1). \quad (14)$$

For a reviewed plan, the simulation backfeeding equation is

$$s_{t+1} = F(s_t, \mathcal{A}_t^*, \xi_t), \quad (15)$$

where ξ_t denotes uncertain factors such as stochastic arrivals, compliance fluctuations, and operating disturbance. Through backfeeding, different decision mechanisms can be compared in terms of total delay, transfer-queue peak, platform-density threshold duration, transfer-failure rate, and action conflict.

2.6. Study Scenario, Online Case Selection, and Experimental Settings

Because real-world operational data are difficult to access in full and are also subject to safety constraints, the experiments in this paper are mainly based on reproducible simulation data. The simulation object is defined as “one disturbed corridor + four key transfer hubs”, namely the Shanghai Metro Line 2 corridor from East Nanjing Road to Lujiazui, together with People’s Square, East Nanjing Road, Century Avenue, and Longyang Road. Figure 2 presents the study scenario. Rather than building a system-wide digital twin, this paper focuses on a corridor-level mesoscopic environment around one disturbed corridor and its key transfer nodes, so that abnormal propagation and linked control can be examined under controllable conditions. The scenario families S1–S4 are used to vary delay intensity and demand level, whereas online cases CS01–CS05 are extracted as critical state snapshots for LLM decision making and simulation backfeeding. These two sets, therefore,

belong to different layers: *S* denotes base operating scenarios, while *CS* denotes online case samples. In addition to the five online cases, this paper further introduces a full 3×3 delay–demand validation matrix and transfer-service-capacity sensitivity settings in order to test whether the conclusions remain stable beyond a small number of visible examples (see Table 3).

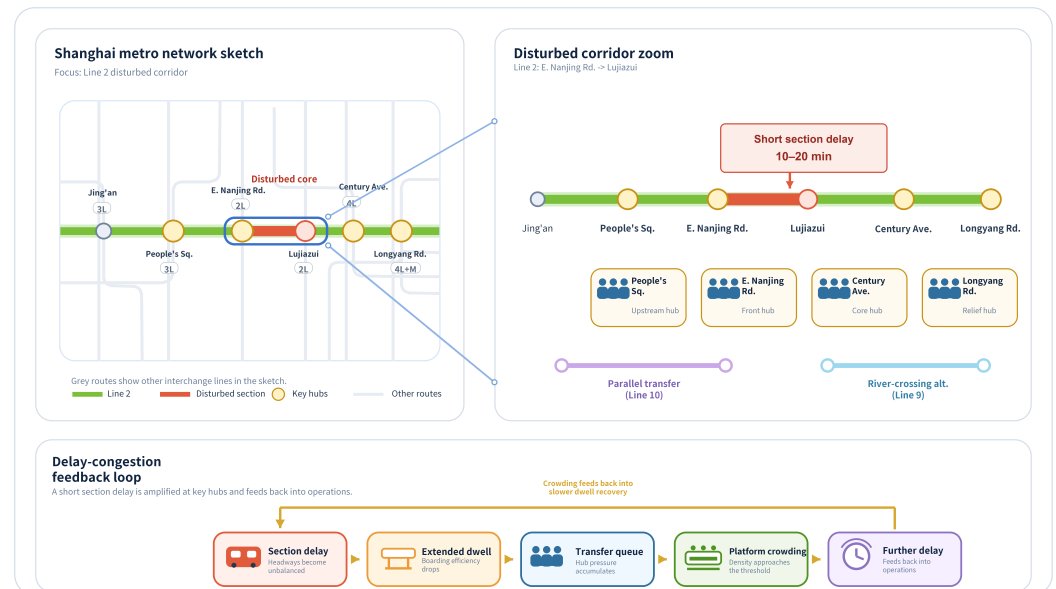


Figure 2. Study scenario: the disturbed Line 2 corridor and its key transfer hubs.

Table 3. Selection logic of the five online cases.

Case	Parent	Trigger (min)	Disturbance Stage	Load Condition	Operability	Why Selected
CS01	S2	20	Onset	Normal	Normal	Captures the early amplification stage, where linked intervention is still able to suppress queue growth before transfer congestion becomes dominant.
CS02	S2	28	Amplifying	Normal	Normal	Represents the developed congestion state of a medium disturbance and stresses multi-role coordination after queue accumulation has already emerged.
CS03	S3	24	Early severe stress	Normal	Normal	Tests whether the framework remains effective when disturbance severity increases even before the system reaches full saturation.
CS04	S4	24	Critical	High	Normal	Represents the most coupled high-demand state and is used to stress linked control under strong delay–congestion interaction.
CS05	T04	24	Constrained response	Normal	Constrained	Introduces reduced dispatch operability and tests whether transfer coordination and information guidance can partially compensate for limited train-side control space.

The five online cases are, therefore, not intended to exhaust the whole condition space. Instead, they are representative critical snapshots selected along two dimensions: amplification-stage difference and decision-difficulty difference. CS01 and CS02 correspond to the onset and amplification stages of a medium disturbance, CS03 corresponds to a severe-delay stress state, CS04 is the high-load critical case, and CS05 is taken from a constrained-operability sensitivity setting. The broader robustness of the framework is examined later through supplementary cross-condition validation and parameter sensitivity analysis.

To ensure transparency and reproducibility, the main simulation and decision parameters are summarized in Table 4. The simulation uses a one-minute step with a total horizon of 105 min, including 15 min of warm-up, 60 min of control, and 30 min of recovery observation. The planned train headway is set to 3.0 min to represent a peak-hour service

level in the core corridor. Baseline passenger-arrival demand is anchored at 260 pax/min and is scaled by 0.9, 1.0, and 1.15 to form three loading levels of approximately 234, 260, and 299 pax/min. The platform safety threshold is set to 4.0 pax/m², and transfer service capacity is modeled heterogeneously across key hubs as 72–106 pax/min, with an average of about 86 pax/min. These magnitudes are not treated as direct field measurements; rather, they are used as calibration anchors whose order of magnitude is aligned with metro disruption-recovery, station-control, and simulation-based crowd-management studies [5,12,15,16]. Parameters that cannot be directly verified from complete operating data are further examined later through cross-condition validation and service-capacity sensitivity tests. The supporting computation and table-processing scripts were implemented in Python 3.12. Online LLM experiments are conducted through the DashScope service in OpenAI-compatible mode with temperature 0.1 and without activating reasoning mode. In the multi-agent experiment, the dispatch and station agents use qwen-plus-2025-07-28, the transfer-coordination and information-guidance agents use qwen-plus-latest, and the safety-review and arbitration agents use qwen3.6-plus. The single-agent baseline uniformly uses qwen3.6-plus. This allocation reflects task difficulty: proposal agents mainly need stable role-specific generation, whereas safety review, arbitration, and the single-agent baseline require stronger cross-role reasoning and consistency checking. To control cost and preserve reproducibility, real online invocation only occurs at critical state snapshots, while rolling backfeeding is executed by an offline planner.

Table 4. Experimental settings, parameter ranges, and provenance.

Item	Value/Range	Role in Model	Basis
Simulation object	One disturbed corridor + four key transfer hubs	Mesoscopic testbed for coupled abnormal propagation	Scenario design for a controllable corridor-level study
Line and section	Line 2, East Nanjing Road–Lujiazui	Disturbed corridor prototype	Urban-core to river-crossing corridor selected for strong transfer coupling
Time step and horizon	1 min; 105 min	Decision and evolution resolution	Modeling assumption; 15 min warm-up, 60 min control, 30 min recovery
Planned headway	3.0 min	Baseline train-service frequency	Peak-hour engineering anchor for the corridor prototype
Passenger-arrival demand	234/260/299 pax/min	Low-, normal-, and high-load demand levels	Scenario multipliers 0.9/1.0/1.15 applied to the 260 pax/min calibration anchor
Delay intensity	10/15/20 min	Short-term disturbance severity	Scenario design for light, medium, and severe delay conditions
Platform safety threshold	4.0 pax/m ²	Threshold-exceedance indicator	Literature-aligned safety-density bound used for mesoscopic evaluation
Transfer service capacity	72–106 pax/min	Queue dissipation bottleneck at transfer hubs	Literature-aligned range plus corridor calibration; mean about 86 pax/min
Online case set	CS01–CS05	Real LLM snapshot evaluation	Representative critical-state sampling rather than exhaustive coverage
Online LLM models	Qwen-series models	Decision generation at critical snapshots	Experimental setup; role-specific calls for the multi-agent setting and qwen3.6-plus for the single-agent baseline

The evaluated strategies are B1, B2, B3, B4, B4-SOP-Limited, SingleAgent, and Online-MA-Rolling. Table 5 summarizes their meanings. B4 and B4-SOP-Limited are deliberately kept as strong rule-based references rather than weak placeholders. They represent the type of structured response that can be assembled from SOP rules, dispatch heuristics,

and station-control experience without LLM collaboration. The main comparison in this paper focuses on B4-SOP-Limited, Online-MA-Rolling, and SingleAgent, because this contrast tests whether the proposed multi-agent chain adds value beyond a feasible rule-based joint response and beyond a single all-purpose LLM.

Table 5. Definition of evaluated benchmark and LLM-assisted strategies.

Strategy	Control Scope	Operational Meaning
B1	No linked control	Disturbed evolution without coordinated intervention; used as the lower reference for disruption impact.
B2	Station-side control only	Applies local station measures such as entry metering, gate throttling, and platform organization without train-side or information-side coordination.
B3	Train-side control only	Applies dispatch measures such as headway rebalancing and connection holding without station or passenger-guidance coordination.
B4	Rule-based joint control	Combines train, station, transfer, and information measures through heuristic SOP logic; represents a structured non-LLM joint-control baseline.
B4-SOP-Limited	Restricted rule-based joint control	Uses the same SOP-style logic as B4 but restricts action intensity and availability under realistic implementation constraints.
SingleAgent	One LLM agent	A single LLM receives the full state and action library and directly produces a complete <i>DecisionCard</i> .
Online-MA-Rolling	Proposed multi-agent chain	Role-specific LLM agents generate proposals, safety review filters infeasible measures, arbitration forms a compact <i>DecisionCard</i> , and simulation backfeeding evaluates the plan.

2.7. Evaluation Metrics

Let $\mathcal{T}^+$ denote the evaluation window after the trigger time. Total delay is defined as

$$M_D = \sum_{t \in \mathcal{T}^+} (d_t^s + d_t^o). \quad (16)$$

The peak transfer queue is defined as

$$M_Q = \max_{t \in \mathcal{T}^+} q_t^r. \quad (17)$$

The duration of platform-density threshold exceedance is

$$M_\rho = \sum_{t \in \mathcal{T}^+} \mathbb{I}(\rho_t > \rho^{safe}). \quad (18)$$

For strategy x relative to baseline y , the improvement rate is defined as

$$I(M; x, y) = \frac{M_y - M_x}{M_y} \times 100\%. \quad (19)$$

If M is a lower-is-better indicator, then $I > 0$ means that strategy x outperforms baseline y . Decision-quality indicators further include the average number of actions, conflict minutes, SOP compliance, online token consumption, and response latency. For the radar chart, all dimensions are normalized to a 0–100 scale and used only as a visual aid for result interpretation rather than as a separate optimization objective.

For the cross-condition validation and transfer-service sensitivity experiments, each condition is simulated with 20 random seeds. Because B4 and the multi-agent strategy are

evaluated under the same seed in each condition, the stochastic comparison is treated as a paired design. In addition to mean improvement rates, the repeated runs are used to report standard deviations, seed-level consistency, and paired significance tests for total delay and peak transfer queue. The five online LLM cases are still interpreted mainly as traceable decision snapshots, whereas the repeated simulation experiments provide the main evidence on stochastic robustness.

3. Results

3.1. Online Decision Process

To avoid presenting LLM-based collaborative decision making as an opaque black box, this paper first demonstrates one online run at a critical state snapshot. Taking the high-load state *P1_B_S4* as an example, the triggered section delay is 10.11 min, the downstream spillover delay is 42.83 min, the transfer queue is 713.78 passengers, the platform queue is 257.83 passengers, and the system risk level is high. Figure 3 shows the process through which the multi-agent system generates a *DecisionCard*.

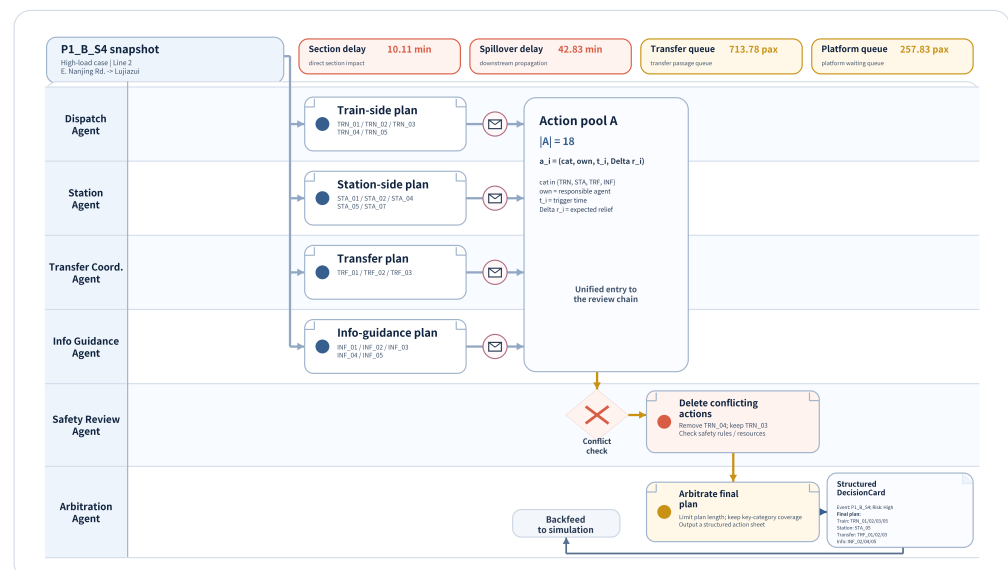


Figure 3. Multi-agent handoff process for *DecisionCard* generation under case *P1_B_S4*. Blue dots denote candidate action groups; orange dots denote reviewed or final outputs; envelope icons denote proposal submission to the action pool; and the cross sign denotes conflict checking and removal.

In this case, the dispatch agent concentrates on headway balancing and selective connection holding, the station agent on entry metering, gate control, escalator organization, and platform zoning, the transfer-coordination agent on upstream offloading and parallel-route diversion, and the information-guidance agent on broadcast, app notification, and signage. The safety-review agent removes TRN_04 because it conflicts with the current resource condition and retains TRN_03. The arbitration agent then compresses the proposal set into a linked plan that still covers the train, station, transfer, and information sides. What matters here is not the raw number of actions. It is the fact that proposal, review, and synthesis are explicit and traceable.

3.2. Comparison Between Online-MA-Rolling and B4-SOP-Limited

Table 6 compares Online-MA-Rolling with B4-SOP-Limited on the five representative online cases. Online-MA-Rolling improves both indicators in all five cases, although the size of the gain varies sharply with state. In CS01, which captures the onset of a medium disturbance, early headway adjustment and passenger guidance cut total delay by 48.8% and the

transfer-queue peak by 90.3% relative to B4. In CS04, the high-load critical case, the linked plan combines entry metering, headway balancing, upstream offloading, and broadcast guidance, reducing total delay by 22.1% and the transfer-queue peak by 42.5%. In CS05, where dispatch operability is restricted, queue relief is limited, but total delay still falls by 16.1%, which suggests that transfer coordination and information guidance can partly compensate for a reduced train-side action space.

Table 6. Backfeeding simulation results of B4-SOP-Limited and Online-MA-Rolling.

Case	Scenario	Trigger Time (min)	Initial Online Actions at t_0	Delay vs. B4 (%)	Queue Peak vs. B4 (%)
CS01	Medium, early	20	TRN_01 headway reb.; INF_01 broadcast	48.8	90.3
CS02	Medium, amplifying	28	TRN_03 conn. hold; STA_02 gate throttle; TRF_03 upstream offload; INF_01 broadcast	3.5	20.2
CS03	Severe, early	24	TRN_01 headway reb.; TRF_03 upstream offload; INF_01 broadcast	7.8	36.5
CS04	High-load case	24	STA_01 entry metering; TRN_01 headway reb.; TRF_03 upstream offload; INF_01 broadcast	22.1	42.5
CS05	Low operability	24	INF_01 broadcast; TRF_03 upstream offload; TRN_03 conn. hold	16.1	2.0

Figure 4 shows the same comparison graphically. The contrast is not between a useless baseline and a strong model; B4-SOP-Limited already provides a workable response. The difference is that once state combinations become harder and several operational roles must move together, the rule-based plan more often leaves gaps in coverage or reacts too mildly. The multi-agent chain helps most in those situations.

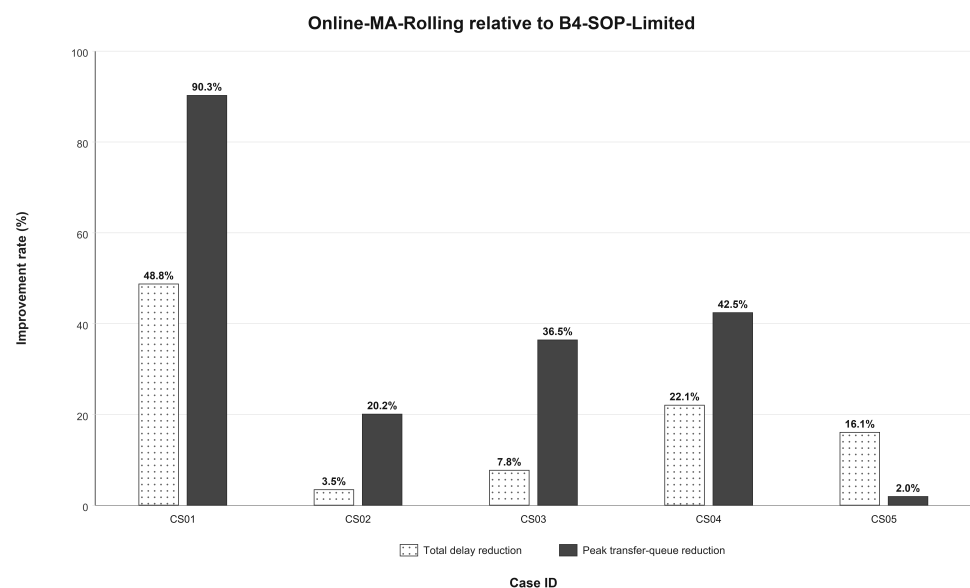


Figure 4. Improvement of Online-MA-Rolling relative to B4-SOP-Limited.

3.3. Expanded Cross-Condition Validation

The five online cases are used to reveal how the LLM-based decision chain behaves at critical snapshots, but they are not the sole evidence base of this paper. To test whether the observed gains persist beyond these visible examples, the same corridor-level simulation is further evaluated on a full 3×3 matrix formed by three delay intensities (10, 15, and 20 min) and three demand multipliers (0.90, 1.00, and 1.15). Table 7 reports the comparison between

the multi-agent linked strategy and B4. Across all nine combinations, the multi-agent strategy yields positive improvements in both total delay and transfer-queue peak. Delay reduction ranges from 45.7% to 75.7%, with an average of 58.9%, whereas queue reduction ranges from 7.7% to 63.4%, with an average of 32.6%. The repeated-seed analysis supports the same conclusion: over 20 paired stochastic runs per cell, delay improvement is positive in 20/20 seeds for every cell, and queue improvement is positive in at least 18/20 seeds even in the most stressed cell. Paired tests show that both delay and queue differences relative to B4 are significant in all nine cells ($p < 0.001$).

Table 7. Expanded cross-condition validation of the multi-agent linked strategy relative to B4.

Delay (min)	Demand Mult.	Delay vs. B4 (%)	Queue vs. B4 (%)	Interpretation
10	0.90	62.7	63.4	Early disturbance under light load still leaves substantial room for linked recovery and queue suppression.
10	1.00	65.8	48.8	Delay pressure rises with demand, but coordinated control still suppresses both delay and transfer accumulation.
10	1.15	75.7	29.5	Heavier demand increases delay-side recovery benefit, while queue relief starts to narrow.
15	0.90	45.7	46.1	Under medium disturbance and low load, both operational and passenger-flow indicators improve clearly.
15	1.00	46.0	28.5	The baseline medium case retains stable delay relief, while queue relief is more moderate.
15	1.15	67.8	9.3	Under medium delay and high load, delay control remains strong but queue suppression becomes limited.
20	0.90	50.8	36.2	Even under severe delay, linked control remains effective when demand pressure is relatively lower.
20	1.00	48.8	23.4	Severe delay with normal demand still shows robust gains, though queue reduction weakens.
20	1.15	67.4	7.7	In the most stressed delay–demand cell, the delay benefit remains positive while queue relief becomes marginal.

Figure 5 visualizes the same evidence as a delay–demand robustness landscape. The left panel shows that delay-side gains remain strong throughout the matrix, whereas the right panel shows that queue-side gains narrow progressively toward the high-delay, high-demand corner. This pattern is consistent with the interpretation that queue relief is more sensitive than delay relief to system saturation.

Table 7 and Figure 5 point to a clear split between the two indicators. Delay reduction stays strong across the whole matrix, which means the framework is not leaning on one favorable case. Queue reduction remains positive in all nine cells, but the margin narrows as delay and demand rise together. The five online cases should, therefore, be read as visible cuts through a broader performance landscape, not as isolated success stories.

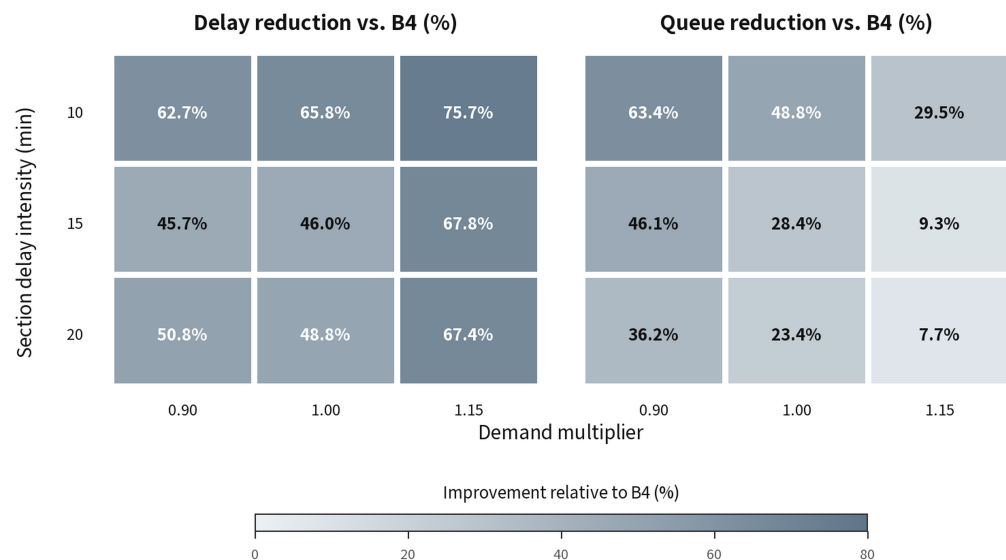


Figure 5. Delay–demand robustness heatmap of the multi-agent strategy relative to B4.

3.4. Parameter Sensitivity and Physical-Bottleneck Effects

Demand sensitivity is already embedded in the 3×3 delay–demand matrix above. To further evaluate whether the conclusions rely on one particular transfer-capacity assumption, the service capacity of key transfer hubs is uniformly scaled by $0.75\times$, $1.00\times$, and $1.25\times$ in a medium-disturbance case (S2) and a high-load case (S4). Table 8 reports the resulting performance changes relative to B4. Each capacity setting is again evaluated with 20 paired random seeds. Under baseline and enhanced capacity, the multi-agent strategy keeps significant delay and queue advantages ($p < 0.001$ for both indicators). Under the $0.75\times$ capacity setting, however, the queue deterioration is also statistically significant, and in the high-load S4 case, the delay reversal is significant as well ($p < 0.001$). Thus, the sensitivity result is not a single-run artifact but a repeated-seed boundary pattern.

Table 8. Sensitivity of the multi-agent strategy to transfer-service capacity scaling.

Scenario	Capacity Scale	Delay vs. B4 (%)	Queue vs. B4 (%)	Interpretation
S2	$0.75\times$	15.3	−133.5	After a 25% capacity loss, the queue benefit turns into queue deterioration even though delay relief remains positive.
S2	$1.00\times$	46.0	28.5	Under baseline capacity, the multi-agent strategy preserves stable gains on both indicators.
S2	$1.25\times$	43.1	65.9	Capacity enhancement substantially strengthens queue suppression in the medium-disturbance case.
S4	$0.75\times$	−9.8	−169.3	In the high-load case, a severe physical bottleneck overwhelms the policy advantage and reverses both indicators.
S4	$1.00\times$	67.4	7.7	Under baseline capacity, delay relief remains strong but queue improvement is already limited.
S4	$1.25\times$	58.6	43.5	Capacity expansion restores a clear queue benefit and strengthens system recovery.

The sensitivity results also make the modeling boundary much clearer. The demand anchor of 260 pax/min is not a single-point claim because it is tested directly through the wider demand matrix. Transfer-service capacity is likewise not assumed to be fixed: once it is reduced by 25%, queue relief collapses in both the medium- and high-load cases, and in S4, even the delay advantage turns negative. The framework should, therefore, be read as a

control layer that works within a feasible infrastructure envelope, not as a remedy for a severe capacity shortfall.

3.5. Comparison Between Multi-Agent Collaboration and a Single-Agent LLM

To further answer whether multi-agent collaboration is necessary, the same state snapshots and action set are also fed into a single-agent LLM, which directly outputs a complete *DecisionCard* and is then backfed into simulation. Table 9 reports the comparison between multi-agent collaboration and the single-agent LLM.

Table 9. Performance comparison between multi-agent collaboration and a single-agent LLM.

Case	Scenario	Delay vs. B1 (M/S)	Queue vs. B1 (M/S)	MA Gain vs. SA	Observation
CS01	Medium onset	55.5%/20.5%	91.4%/53.2%	Delay +35.0% Queue +38.2%	MA is more stable
CS02	Medium amplification	14.1%/17.2%	24.8%/61.2%	Delay −3.1% Queue −36.4%	SA is stronger here
CS03	Severe early stress	23.6%/19.7%	41.6%/43.0%	Delay +3.9% Queue −1.4%	MA is better on delay
CS04	High-load critical	33.9%/1.6%	43.1%/14.0%	Delay +32.3% Queue +29.1%	MA is more robust
CS05	Low operability	45.8%/−51.5%	2.0%/−12.6%	Delay +97.3% Queue +14.6%	SA becomes unstable

Figure 6 presents the absolute indicator values of the two decision mechanisms. The multi-agent framework is not uniformly dominant on every local indicator. For example, in CS02, the single-agent policy is stronger in queue suppression, and in CS03, the two methods are close in queue performance. However, in high-load and low-operability scenarios, the single-agent LLM is more likely to produce locally effective yet structurally unbalanced action sets, and in CS05, it even causes deterioration relative to B1 in total delay. This indicates that the advantage of multi-agent collaboration lies less in producing the most aggressive local response each time than in reducing the probability of serious policy failure through role coverage, safety review, and arbitration.

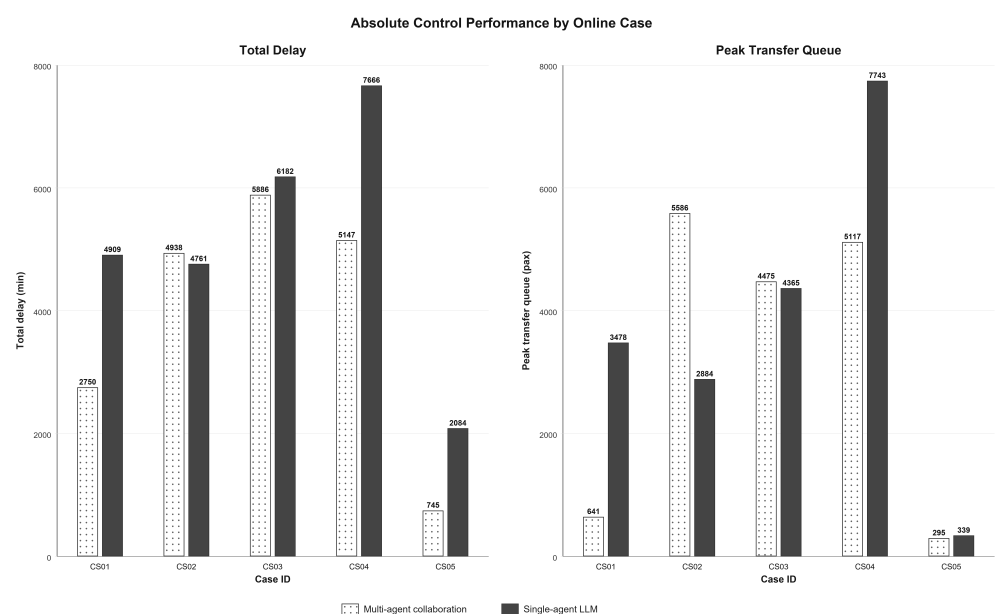


Figure 6. Absolute control performance of multi-agent collaboration and the single-agent LLM.

Figure 7 summarizes the same contrast from a broader capability perspective. The multi-agent framework scores higher in delay control, queue control, linked-action coverage, and robustness, while the single-agent setting remains lighter in cost and coordination overhead. In practice, that points to a straightforward division of use: multi-agent collaboration is better suited to high-risk and tightly coupled disruption states, whereas a single-agent LLM is more plausible as a low-risk advisory tool.

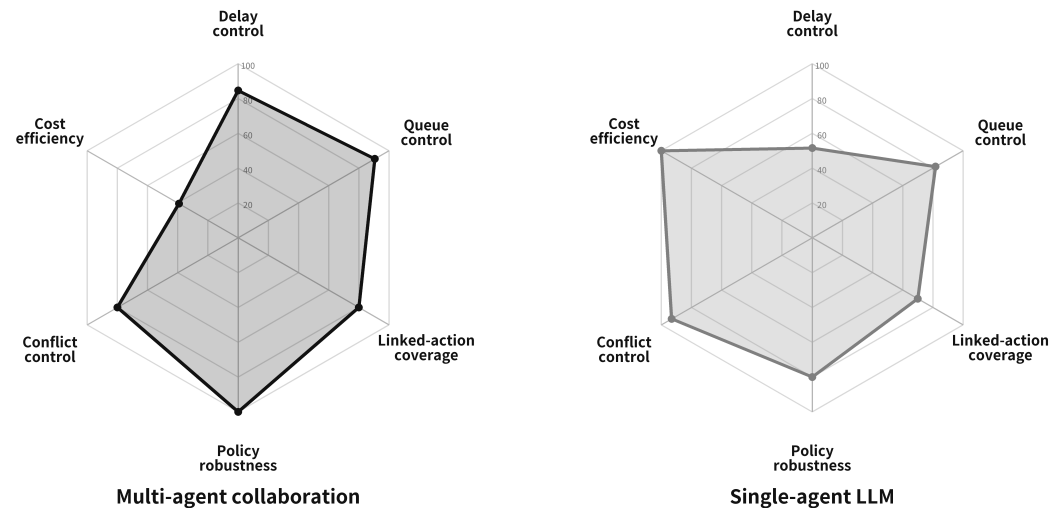


Figure 7. Multi-dimensional capability portrait of multi-agent collaboration and the single-agent LLM.

4. Discussion

4.1. Positioning the Framework Against Existing Disruption-Management Studies

The first issue is how the present results should be read against the broader disruption-management literature. Classical railway studies mainly examine train rescheduling, rerouting, and conflict resolution in real time [1–3]. That line of work remains indispensable, but metro disturbances usually do not stay on the train-operation layer alone. Once transfer stations begin to absorb delayed arrivals, the operational problem quickly becomes a coupled one involving dwell extension, platform accumulation, and path-choice adjustment. This is precisely why later studies moved toward integrated models that combine timetable recovery, passenger reassignment, station constraints, and coordinated inflow control [4–6,10–13].

The results reported here are largely consistent with that shift in emphasis. In the five online cases, the linked strategy is most effective when train regulation, station organization, transfer coordination, and information guidance are all allowed to move together. In the broader 3×3 delay–demand matrix, delay reduction remains positive in every cell, whereas queue relief weakens steadily toward the high-delay, high-demand corner. That pattern is in line with previous studies on metro passenger-flow control and hub congestion propagation: coordinated measures can redistribute pressure and postpone local saturation, but once transfer channels become the dominant bottleneck, passenger-side gains narrow much faster than train-side gains [7,8,15,17]. The same logic also helps explain why route guidance and transfer coordination matter more in cases with limited train-side operability, which agrees with recent work on passenger information systems and network-level collaborative recovery [10,11,14].

Table 10 summarizes the position of this study relative to several representative lines of prior work. The point is not that the present framework replaces optimization-based recovery models. Rather, it sits at the interface between optimization and operations,

converting a coupled disruption state into a reviewed multi-role response that can still be tested in simulation.

Table 10. Positioning of this study relative to representative lines of prior work.

Theme	Representative Studies	Main Emphasis in Existing Literature	Position of This Study
Train recovery and rescheduling	[1–3,10,11]	Real-time rescheduling, conflict resolution, rolling recovery, and multi-line schedule adjustment	Retains train-side control, but evaluates it together with station, transfer, and information measures in one linked response chain
Integrated passenger–operation control	[4–6,13]	Joint handling of timetable recovery, passenger reassignment, overcrowding, and collaborative flow control	Focuses on the delay–transfer coupling mechanism and on corridor-level linked control under short-term disruption
Station organization and route guidance	[7,8,12,14,15]	Inflow restriction, station organization, route guidance, and adaptive local congestion control	Shows when passenger-side and information-side measures compensate for a restricted train-side action space
LLM-based decision support	[23–27,29]	Reasoning, tool use, multi-agent interaction, and bounded deployment of LLMs in decision support	Converts LLM outputs into reviewed <i>DecisionCards</i> and verifies them through simulation backfeeding rather than direct command issuance

4.2. Operational Meaning, Deployment Boundary, and Remaining Limitations

The second issue concerns what the framework actually contributes in operational terms. Its contribution does not lie in producing the most aggressive action set at every snapshot. The comparison with the single-agent LLM makes that clear. In some medium cases, a single agent can remain locally competitive; in CS02, for instance, it delivers stronger queue suppression. The more consistent difference appears when the state becomes tighter and the response must remain balanced across several roles. Under those conditions, the multi-agent chain is less likely to produce plans that are locally plausible but structurally incomplete. That is where the safety-review and arbitration stages matter. They do not guarantee optimality, but they reduce the chance of a policy that ignores one operational side entirely. This design choice is also closer to current thinking in the LLM-agent literature, which emphasizes tool grounding, explicit interaction protocols, and post-generation checks rather than unconstrained text generation [23–26]. Transportation-oriented surveys make the same point in more practical terms: if LLMs are to support safety-critical decisions, they need bounded interfaces, traceability, and accountable human oversight [27–29].

Practical deployment would require more safeguards than those used in this simulation study. First, response latency must be bounded because an abnormal-response suggestion is useful only if it arrives within the dispatch decision window. Second, the system should keep an auditable record of the input state, role proposals, rejected actions, arbitration reasons, and final *DecisionCard*. Third, the LLM output should remain advisory until reviewed by authorized operators; it should not directly issue train-control or crowd-control commands. Fourth, unexpected operating states should trigger a fallback to approved SOP procedures or to a human-only decision mode. These requirements are ethical and regulatory as much as technical, because public-transport decisions affect passenger safety, responsibility allocation, and institutional accountability.

The capacity-sensitivity analysis also clarifies the deployment boundary of the method. When transfer-service capacity stays near or above the baseline, the linked strategy retains a measurable advantage over B4. Once transfer capacity is reduced by 25%, however, queue performance deteriorates sharply, and in the high-load case, the delay advantage even turns negative. This is an important result because it prevents the framework from

being overstated. The method should be understood as a coordination layer operating within a feasible infrastructure envelope, not as a substitute for physical capacity at transfer hubs. That interpretation is consistent with metro studies showing that station control, train adjustment, and passenger guidance are most effective when they act on a system that still has residual handling capacity [9,12,13,15].

The framework is also transferable only at the level of modeling logic and interface design, not at the level of fixed parameter values. For another metro network, the state vector, role division, safety-review logic, and *DecisionCard* format can be reused, but the thresholds, risk weights, action library, SOP constraints, transfer capacities, demand profiles, and compliance assumptions must be recalibrated. Networks with radial topologies, dense grid structures, airport–rail connections, or strongly asymmetric transfer hubs may, therefore, produce different bottleneck patterns. The appropriate transfer path is to rebuild the local scenario library, validate the mesoscopic parameters with local AFC and train-operation data where available, and then retest the multi-agent chain under the new topology.

The study still has clear limitations. The simulation is corridor-level and mesoscopic; it is neither a full timetable-recovery engine nor a microscopic pedestrian model. Real online LLM calls are limited to critical snapshots, while the rolling backfeeding stage is handled by an offline planner rather than a fully online closed loop. In addition, several parameters are calibrated against literature ranges and controlled assumptions instead of full operational logs. A stronger validation path would combine AFC records, train-operation logs, and in-station monitoring data, then extend the framework from one disturbed corridor to multi-line and multi-hub settings with human-in-the-loop verification.

5. Conclusions

This study examines abnormal response in urban rail transit from the coupled perspective of section delay and transfer congestion and develops an LLM-driven multi-agent decision framework around that problem. The point is not to let an LLM replace dispatching authority. The point is to place LLM-based reasoning inside a decision chain that can be reviewed, filtered, and tested against simulation.

Three conclusions follow from the experiments. First, short section delays in peak periods do not remain local for long. They reshape transfer arrivals, increase platform accumulation, extend dwell time, and then feed back into downstream operations. That coupling is the basic reason why linked train, station, transfer, and information control is needed. Second, the multi-agent framework produces measurable benefits beyond a few showcase moments. The five online cases expose the actual decision chain, while the broader 3×3 delay–demand matrix shows that the strategy keeps positive gains over B4 across all nine supplementary combinations. Third, the advantage of multi-agent collaboration lies mainly in robustness and structural balance. A single-agent LLM can still perform well in some medium cases, but it is more vulnerable to policy imbalance when the state becomes tighter. Even the multi-agent framework, however, remains bounded by physical transfer capacity; once that bottleneck becomes severe, the room for policy improvement contracts quickly. For real implementation, the framework should be treated as a human-supervised decision-support layer that must be locally recalibrated, audited, and validated before use in another network.

Author Contributions: Conceptualization, H.W. and X.Z.; methodology, H.W.; software, H.W.; validation, H.W. and X.Z.; formal analysis, H.W.; investigation, H.W.; data curation, H.W.; writing—original draft preparation, H.W.; writing—review and editing, H.W. and X.Z.; visualization, H.W.; supervision, X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The simulation outputs, figures, and manuscript source files used in this study are contained in the minimal dataset package. Additional materials are available from the corresponding author on reasonable request.

Acknowledgments: During the preparation of this manuscript, the authors used ChatGPT (OpenAI) for language polishing and drafting support. The authors reviewed and edited the output and take full responsibility for the content of this publication.

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

References

1. Cacchiani, V.; Huisman, D.; Kidd, M.; Kroon, L.; Toth, P.; Veelenturf, L.; Wagenaar, J. An overview of recovery models and algorithms for real-time railway rescheduling. *Transp. Res. Part B Methodol.* **2014**, *63*, 15–37. [\[CrossRef\]](#)
2. D'Ariano, A.; Corman, F.; Pacciarelli, D.; Pranzo, M. Reordering and local rerouting strategies to manage train traffic in real time. *Transp. Sci.* **2008**, *42*, 405–419. [\[CrossRef\]](#)
3. Corman, F.; D'Ariano, A.; Pacciarelli, D.; Pranzo, M. Bi-objective conflict detection and resolution in railway traffic management. *Transp. Res. Part C Emerg. Technol.* **2012**, *20*, 79–94. [\[CrossRef\]](#)
4. Besinovic, N.; Wang, Y.; Zhu, S.; Quaglietta, E.; Tang, T.; Goverde, R.M.P. A matheuristic for the integrated disruption management of traffic, passengers and stations in urban railway lines. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 10380–10394. [\[CrossRef\]](#)
5. Gao, Y.; Kroon, L.; Schmidt, M.; Yang, L. Rescheduling a metro line in an over-crowded situation after disruptions. *Transp. Res. Part B Methodol.* **2016**, *93*, 425–449. [\[CrossRef\]](#)
6. Zhu, Y.; Goverde, R.M.P. Integrated timetable rescheduling and passenger reassignment during railway disruptions. *Transp. Res. Part B Methodol.* **2020**, *140*, 282–314. [\[CrossRef\]](#)
7. Liu, J. Research on metro network passenger-flow coordinated control based on monitoring and early warning of station passenger-density coefficients. *Urban Mass Transit* **2021**, *24*, 4–9. (In Chinese). [\[CrossRef\]](#)
8. Yao, X.; Zhao, P.; Qiao, K.; Yu, D. Coordinated passenger-flow control model for urban rail transit networks. *J. Cent. South Univ. (Sci. Technol.)* **2015**, *46*, 342–350. (In Chinese)
9. Meng, F.; Yang, L.; Lu, Y.; Guo, R. Coordinated optimization of urban rail train timetables and station inflow control considering skip-stop strategies. *J. Transp. Syst. Eng. Inf. Technol.* **2021**, *21*, 156–162. (In Chinese). [\[CrossRef\]](#)
10. Wang, F.; Wang, P.; Hao, X.; Yang, R.; Xu, R. Collaborative rescheduling of train timetables to relieve passenger congestions in an urban rail transit network: A rolling horizon approach. *Int. J. Transp. Sci. Technol.* **2025**, *19*, 193–210. [\[CrossRef\]](#)
11. Wang, L.; Jin, J.G. Multi-line coordinated schedule adjustment optimization for metro short-term disruption recovery. *Comput. Ind. Eng.* **2026**, *213*, 111770. [\[CrossRef\]](#)
12. Zhang, P.; Liu, R.; Yang, L.; Gao, Z. Model and algorithm of coordinated flow controlling with station-based constraints in a metro system. *Transp. Res. Part E Logist. Transp. Rev.* **2021**, *148*, 102258. [\[CrossRef\]](#)
13. Lu, Y.; Yang, L.; Yang, H.; Zhou, H.; Gao, Z. Robust collaborative passenger flow control on a congested metro line: A joint optimization with train timetabling. *Transp. Res. Part B Methodol.* **2023**, *168*, 27–55. [\[CrossRef\]](#)
14. Wang, X.; Xu, X.; Wong, M.; Liu, J. Distributed dynamic route guidance via passenger information display systems for subway disruption management. *Transp. Res. Part C Emerg. Technol.* **2025**, *179*, 105316. [\[CrossRef\]](#)
15. Zhou, Y.; Zhong, M.; Li, Z.; Xu, X.; Hua, F.; Pan, R. Simulation-based adaptive optimization for passenger flow control measures at metro stations. *Simul. Model. Pract. Theory* **2025**, *138*, 103021. [\[CrossRef\]](#)
16. Lu, G.; Lei, Y.; Zhang, H. Urban rail transit flow-control strategies based on multi-agent simulation. *J. Tongji Univ. (Nat. Sci.)* **2022**, *50*, 1189–1197. (In Chinese). [\[CrossRef\]](#)
17. Zhao, W.; Xu, M. A passenger flow congestion propagation Bayesian network model for urban rail transit hubs. *Reliab. Eng. Syst. Saf.* **2026**, *266*, 111655. [\[CrossRef\]](#)
18. Chai, H.; Sun, T. Impact of transit network on local COVID-19 infection risk from the perspectives of global network centrality and local network structure: A case study of Beijing. *Cities* **2025**, *165*, 106133. [\[CrossRef\]](#)
19. Chai, H.; Sun, T.; Wang, Y. Characterizing a long-time evolution of metro network topology: Evidence from metro networks in 34 Chinese cities. *Transp. Res. Rec.* **2026**. [\[CrossRef\]](#)
20. Li, M.; Zhou, X.; Liu, J.; Ma, W.; Li, X. Topological modeling and analysis of urban rail transit safety risk relationship. *J. Intell. Manag. Decis.* **2022**, *1*, 108–117. [\[CrossRef\]](#)
21. Passarella, R.; Ayu, A.P.; Marieska, M.D.; Isbatudinia; Solehan, N.; Veny, H.; Rahmat, R.F. Leveraging real-time GTFS and integrated data for high-accuracy LRT departure delay prediction using optimized machine learning. *Int. J. Transp. Dev. Integr.* **2025**, *9*, 790–803. [\[CrossRef\]](#)
22. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2017; pp. 5998–6008.

23. Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.; Cao, Y. ReAct: Synergizing reasoning and acting in language models. In Proceedings of the International Conference on Learning Representations, Kigali, Rwanda, 1–5 May 2023.
24. Schick, T.; Dwivedi-Yu, J.; Dessi, R.; Raileanu, R.; Lomeli, M.; Hambro, E.; Zettlemoyer, L.; Cancedda, N.; Scialom, T. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2023; Volume 36, pp. 68539–68551.
25. Wang, L.; Ma, C.; Feng, X.; Zhang, Z.; Yang, H.; Zhang, J.; Chen, Z.; Tang, J.; Chen, X.; Lin, Y.; et al. A survey on large language model based autonomous agents. *Front. Comput. Sci.* **2024**, *18*, 186345. [[CrossRef](#)]
26. Wu, Q.; Bansal, G.; Zhang, J.; Wu, Y.; Li, B.; Zhu, E.; Jiang, L.; Zhang, X.; Zhang, S.; Liu, J.; et al. AutoGen: Enabling next-gen LLM applications via multi-agent conversation framework. *arXiv* **2023**, arXiv:2308.08155.
27. Jin, Y.; Ma, J. A survey of large language models in transportation planning: Modelling, design and decision-making. *Transp. A* **2026**. [[CrossRef](#)]
28. Zhang, Z.; Sun, Y.; Wang, Z.; Nie, Y.; Ma, X.; Li, R.; Sun, P.; Ban, X. Large language models for mobility analysis in transportation systems: A survey on forecasting tasks. *Transp. Res. Rec.* **2026**, *2680*, 756–774.
29. Nie, T.; Sun, J.; Ma, W. Exploring the roles of large language models in reshaping transportation systems: A survey, framework, and roadmap. *Artif. Intell. Transp.* **2025**, *1*, 100003. [[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.