Execution-Time State Projection for Dynamic Air-Ground Logistics: A Digital-Twin Cooperative Q-Learning Approach

Songlin Luo *, Lili Yu, Yao Wang

College of Economics and Management, Anyang University, Anyang 455000, China

* Corresponding Author Email: luosonglin94@163.com

Abstract. In dynamic air-ground delivery, the operating conditions observed when an order is assigned may no longer hold when the selected resource actually becomes available. This timing gap is especially relevant when congestion, temporary airspace restrictions, or UAV endurance conditions change while tasks are waiting in a queue. To address this issue, we introduce execution-time state projection into a cooperative Q-learning scheduler. For every candidate vehicle or UAV, the scheduler estimates the state expected at the earliest feasible service-start time and uses this information when choosing the transport mode and resource rule. A deterministic execution mask is retained to prevent infeasible UAV actions from being dispatched. The mechanism is examined on a synthetic county-rural network containing 18 service nodes, three ground vehicles, and five UAVs. Six operating conditions are considered, with 50 matched replications for each method-scenario pair. The largest differences appear under the compound disruption. Relative to the rolling-horizon heuristic, mean response time decreases by 15.8%, P95 response time by 18.4%, the modeled energy index by 7.3%, and recovery time by 17.4%. These four differences remain significant after Holm adjustment ($p<0.001$). By contrast, the change in on-time delivery is only 0.65 percentage points and is not statistically significant ($p=0.053$). When execution-time projection is removed, mean response increases by 22.0%, and the stale-state policy proposes infeasible UAV actions for 16.9% of orders. Forecast-severity perturbations of ±10% and ±20% weaken performance gradually rather than abruptly. The results indicate that the main benefit arises from using state information that is closer to the actual execution time, rather than from a general advantage of reinforcement learning over heuristic dispatch.

Keywords: Digital twin, cooperative multi-agent reinforcement learning, execution-time projection, air-ground logistics, dynamic scheduling, resilience.

1. Introduction

Coordinated use of ground vehicles and UAVs has received sustained attention in last-mile logistics. Early studies mainly examined whether road and air resources could be synchronized effectively [1-3]. Subsequent work moved closer to operational practice by considering payload limits, endurance, launch and retrieval requirements, and energy consumption [4, 5]. More recent models have incorporated multiple vehicles, dynamically arriving requests, uncertain road travel times, aerial congestion, and synchronization reliability [6–10]. One modeling assumption, however, remains common across much of this literature: a dispatch decision is usually based on the system state observed when the order is assigned.

That assumption becomes problematic when the selected resource is not immediately available. A truck or UAV may remain occupied for several minutes, or considerably longer under congestion. During that interval, road conditions can deteriorate, an airspace restriction may begin, or the feasible endurance of a UAV may change. The assignment may therefore be reasonable at the decision instant but unattractive by the time service actually starts. This paper studies that particular mismatch between assignment-time information and execution-time conditions. It does not attempt to establish whether UAV delivery is generally preferable to conventional road delivery.

Digital-twin methods provide a way to maintain and project operational state [11, 12], while cooperative reinforcement learning offers a structured means of coordinating related scheduling choices [13-15]. The question considered here is more specific than simply combining these two techniques. We ask which state should be used when a queued order is evaluated: the state observed

now, or the state expected when a candidate resource becomes available. Our design uses the latter. Three elements are examined. First, a resource-specific state is projected to the candidate service-start time. Second, that projected information is passed to a centralized joint Q-learning policy that selects a transport mode and a resource rule. Third, the effect of information timing is tested through matched simulations, an ablation without projection, backlog-recovery analysis, and forecast-bias experiments.

2. Related Work and Research Gap

Existing air-ground routing studies make clear that synchronization, travel-time uncertainty, energy consumption, and shared resource availability can alter both feasibility and delivery performance [4, 6-10]. Digital-twin studies approach the problem from a different direction, emphasizing state updating and short-horizon projection [11, 12]. Cooperative-learning studies, meanwhile, show how related decisions can be evaluated under a common objective [13-15]. The unresolved point is the timing of the information supplied to the scheduler. If a resource can start immediately, the current state and the execution state are almost the same. Once queues form, this is no longer true. The present experiments therefore keep the action structure and execution constraints unchanged and vary the timing of the state information. This makes it possible to examine whether execution-time projection itself contributes to the observed scheduling differences.

3. Method

3.1 System Setting And Projected State

The simulated system contains one county-level depot, a set of rural demand nodes N, ground resources G, and UAV resources D. Deliveries are dispatched directly from the depot to customer nodes; truck-carried UAV launch and retrieval are not modeled. Order i enters the system at ai, with payload wi, delivery distance di, and deadline hi. A candidate resource j may already be occupied and becomes available at bj. Its earliest possible service-start time is therefore $t^s_{ij} = \max(a_i, b_j)$. Let $r_{j,t}$ denote the resource state observed at the current event and c_t the prevailing disturbance context. The operational twin applies the projection operator Π to estimate the candidate state $\tilde{r}_{ij}$ at t^s_{ij}. In this study, the twin is used as an event-level decision model rather than as a detailed physical replica of the vehicle or UAV.

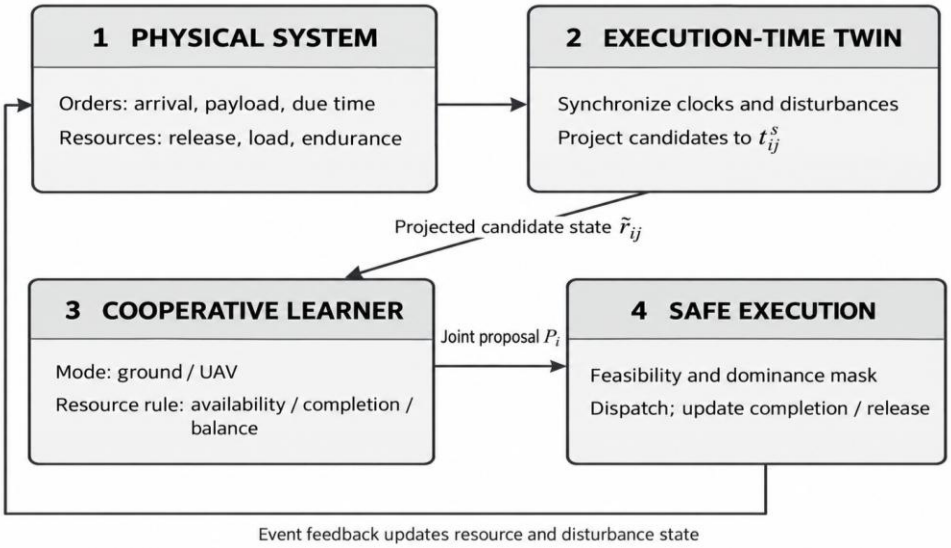


Figure 1. Execution-time digital-twin architecture for cooperative air-ground scheduling.

Figure 1 shows how information moves through the scheduler. New orders and current resource states are first recorded in the physical layer. The twin then updates resource release times and

disturbance conditions and projects each candidate resource to its expected service-start time. The learner evaluates the resulting candidate states and produces a joint mode/resource-rule proposal. Before the assignment is executed, the mask checks payload, endurance, airspace availability, and dominance conditions. The realized completion time and resource release time are then returned to the system state. The projected-state model and the ablation model use the same execution checks; the difference lies in the state supplied to the learner.

3.2 State Encoding, Joint Action, And Learning

The learning component contains two linked decision roles. One determines whether an order is assigned to ground or UAV service, and the other chooses the resource-allocation rule. Both roles share the same reward and joint Q-table; the implementation is therefore centralized. For order i, the feature vector is $x_i = (d_i, w_i, \ell_i)$, where $\ell_i = h_i - a_i$ denotes initial time slack. The set J_i contains all candidate resources considered for that order. A fixed encoder φ combines the order features, disturbance context, and projected candidate states into the finite state Z_i. The same encoder is used in the projected-state and no-projection variants.

$$t^s_{ij} = \max(a_i, b_j), \quad \tilde{r}_{ij} = \Pi(r_{j,t}, c_t; t^s_{ij}) \tag{1}$$

$$Z_i = \varphi(x_i, c_t, \{\tilde{r}_{ij} : j \in J_i\}) \tag{2}$$

$$P_i = \arg\max_{(A \in \mathcal{A}_i)} Q(Z_i, A), \quad A_i = M(Z_i, P_i) \tag{3}$$

$$R_i = -(\lambda_1 T_i + \lambda_2 L_i + \lambda_3 E_i + \lambda_4 I_i) \tag{4}$$

$$y_i = R_i + \gamma \max_{(A' \in \mathcal{A}_{i+1})} Q(Z_{i+1}, A') \tag{5}$$

$$Q(Z_i, P_i) \leftarrow Q(Z_i, P_i) + \alpha[y_i - Q(Z_i, P_i)] \tag{6}$$

Equation (3) distinguishes the action proposed by the learner from the action that is finally executed. P_i denotes the joint proposal, whereas M applies the deterministic feasibility mask and returns the executed action Ai. If the proposed action violates an operating constraint, the simulator uses the fallback action and sets I_i=1. The reward then reflects the realized fallback outcome together with the infeasibility penalty. In (4), T_i,L_i,and E_i represent response-time, lateness, and modeled-energy costs.The coefficients λ_1–λ_4 remain unchanged in the two learning variants. Equations (5) and (6) give the one-step target and Q-value update. No bootstrap term is used at a terminal event. Sensitivity to the reward weights and state discretization is left outside the present experiment.

3.3 Training Protocol And Online Overhead

The Q-table is trained for 2,200 episodes. The learning rate is set to α=0.13, the discount factor to γ=0.95, and the exploration probability is reduced from 0.75 to 0.04 over training. The learned table is fixed before evaluation begins. Separate deterministic seed ranges are used for training and testing. Within each test replication, all competing methods receive the same realized demand sequence. Average online decision times are 0.110 ms per order for DT-MARL, 0.049 ms for RHH, and 0.037 ms for Greedy-RT under the same implementation. These values indicate that online computation is small in the present benchmark, although they should not be interpreted as evidence of scalability to substantially larger fleets.

3.4 Online Decision Cycle

When an order arrives, the twin first refreshes resource availability, workload, endurance, and the current disturbance condition. For each candidate j, it calculates t^s_{ij} and then obtains $\tilde{r}_{ij}$ through Π. These candidate states are encoded into Z_i, after which the learner generates proposal P_i. The execution mask checks the proposal before an assignment is released. Once service is completed, the realized completion time, energy use, and next resource-availability time are written back to the event state.

The distinction becomes relevant whenever a resource remains busy long enough for the operating environment to change. For example, a vehicle released after the onset of road congestion faces a different condition from the one observed when the order first entered the queue. The ablation is constructed to capture only this difference in information timing. MARL w/o DT retains the same state encoding, action structure, reward, training procedure, and execution mask, but it evaluates the assignment with the current or stale resource state. The comparison therefore targets the contribution of execution-time projection rather than differences in the learning architecture.

Algorithm 1. Online DT-MARL scheduling cycle

1) observe the order and current system state; 2) compute candidate release and start times; 3) project each candidate state to service start; 4) encode Z_i and select P_i; 5) apply M; 6) dispatch the accepted or fallback action; 7) update the event state after service.

4. Experimental Design

The experiments use a synthetic benchmark so that the effect of information timing can be examined without relying on proprietary operating records. The network contains 18 rural delivery nodes located 5 – 30 km from the depot. Three ground vehicles and five heterogeneous UAVs are available. Orders enter the system during a 480-min arrival period. Deliveries already in the queue are allowed to continue after the arrival horizon, which is why response and recovery times may exceed 480 min. Six operating settings are considered: normal service, a demand surge, road congestion, an airspace restriction, UAV battery degradation, and a compound disruption. All numerical results reported below refer to this simulated environment and should not be read as field estimates for a particular carrier or county.

Table 1. Parameter settings used in the simulation experiments.

Item	Setting
Timing	Arrivals 0-480 min; disruption 180-300 min; service continues until the queue clears
Network	18 rural nodes; 5-30 km from depot
Fleet	Ground: 3 vehicles, 38/42/46 km h^{-1}; UAV: 5 units, 55-65 km h^{-1}, payload 1.5-3.0 kg, endurance 42-48 min
Demand / compound shock	Poisson mean 68; due windows 60/100/150 min; surge + road ×1.65 + wind ×1.28 + partial closure + endurance ×0.82
Joint actions	Mode: ground/UAV; resource rules: earliest availability, earliest predicted completion, workload-energy balance
Learning	2,200 episodes; α=0.13; γ=0.95; ε: 0.75→0.04; fixed state encoding and reward weights; shared joint Q-table
Evaluation	50 paired replications per scenario; training and evaluation use disjoint seed ranges

Four adaptive dispatch policies are included in the comparison. Greedy-RT assigns the order to the currently feasible resource with the earliest estimated completion time. RHH supplements this rule with workload and UAV-scarcity penalties. MARL w/o DT uses the same learning framework and execution mask as DT-MARL but relies on current-state information. DT-MARL replaces that information with the projected candidate state. Greedy-RT and RHH do not receive candidate execution-time projections. A ground-only policy is also retained as a basic operational reference, but it is not treated as a principal competitor in the statistical comparison.

4.1 Paired Evaluation And Statistical Inference

To reduce simulation noise in method-to-method comparisons, all adaptive policies are exposed to the same realized order stream within a replication. This produces matched observations across methods. Confidence intervals are calculated from the 50 replication-level statistics using the Student-t distribution. For the five prespecified outcomes in the compound scenario, differences between DT-MARL and RHH are tested with two-sided paired Wilcoxon signed-rank tests. Holm correction is

applied to the resulting p-values. These tests describe uncertainty arising from repeated simulation runs. They do not account for uncertainty in model specification or for the gap between the benchmark and field operation.

Performance is not judged from a single indicator. A heavily loaded system can show only a small change in the proportion of orders delivered on time while still experiencing a meaningful reduction in extreme delay or recovery time. We therefore examine mean response, P95 response, on-time delivery, modeled energy use, rejected infeasible proposals, and post-disruption recovery together.

4.2 Evaluation Metrics And Recovery Criterion

Response time is defined as the interval from order arrival to completed delivery. P95 response is calculated separately within each replication from the order-level response-time distribution. The on-time rate records the proportion of orders completed no later than their assigned deadlines. Recovery is measured from the end of the disruption period to the point at which unfinished orders return to the predefined pre-disruption band and remain there for the required stability interval. The same recovery rule is used for all methods. The reported energy index combines the consumption coefficients assigned to the ground and UAV resources; it is intended for comparison inside the simulation and is not a calibrated estimate of real fuel or electricity use.

5. Results and Discussion

5.1 Cross-Scenario Service Reliability

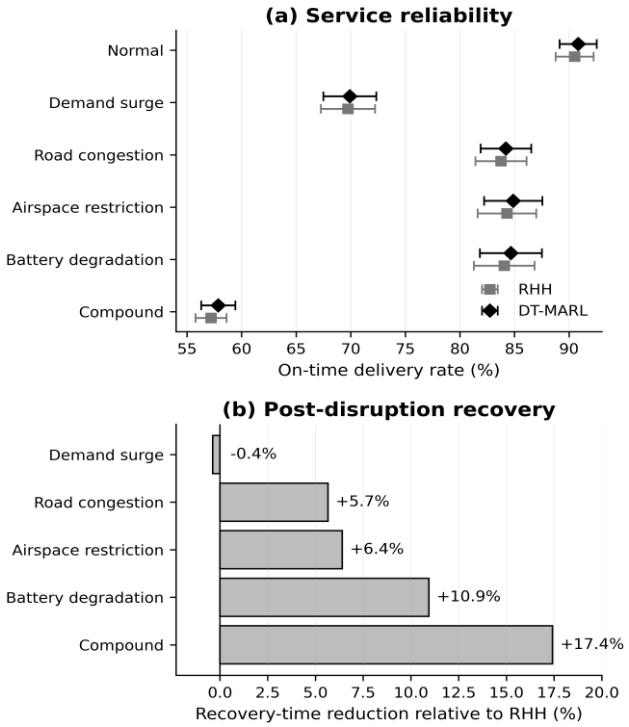


Figure 2. Cross-scenario evidence: (a) on-time delivery with Student-t 95% confidence intervals; (b) recovery-time reduction relative to RHH.

Table 2. Cross-scenario mean response and recovery comparison.

Scenario	RHH response (min)	DT-MARL response (min)	Recovery reduction vs RHH (%)
Normal	46.5	46.3	—
Demand surge	90.7	90.5	-0.4%
Road congestion	57.7	57.1	5.7%
Airspace restriction	55.0	54.3	6.4%
Battery degradation	57.8	56.4	10.9%
Compound	190.1	160.1	17.4%

The cross-scenario comparison shows that the projected-state policy offers little advantage in the two settings where the operating environment remains relatively stable. Under normal service, mean response is almost unchanged. A similar pattern appears during the demand surge. Differences become larger once the disturbance affects conditions that may change while an order is waiting. Recovery is 5.7% shorter under road congestion, 6.4% shorter under the airspace restriction, 10.9% shorter under battery degradation, and 17.4% shorter in the compound case. Demand surge again behaves differently: recovery is 0.4% slower than under RHH. Because road, airspace, and endurance states remain unchanged in that scenario, projecting them forward contributes little useful information.

This pattern is consistent with the mechanism being tested. Projection has limited value when the present state remains a good description of the state at service start. It becomes more useful when queued work crosses a disturbance boundary. The demand-surge case is important for this reason: it shows that the method does not improve performance automatically whenever load increases.

5.2 Compound Disruption: Magnitude And Significance

Table 3. Compound-disruption comparison between RHH and DT-MARL (mean ± 95% CI).

Metric	RHH	DT-MARL	Gain vs RHH	Holm-adjusted p
Mean response (min)	190.1±12.3	160.1±10.8	15.8%	<0.001
P95 response (min)	582.9±34.8	475.8±34.0	18.4%	<0.001
On-time rate (%)	57.2±1.5	57.9±1.6	+0.65 pp	0.053
Energy index	449.7±17.4	416.7±16.9	7.3%	<0.001
Recovery (min)	626.5±36.2	517.4±34.8	17.4%	<0.001

Note. Entries are replication means with Student-t 95% confidence intervals based on 50 matched runs. The reported p-values come from paired two-sided Wilcoxon signed-rank tests and are adjusted with the Holm procedure across the five prespecified outcomes.

The compound disruption produces the clearest difference between DT-MARL and RHH. Mean response decreases from 190.1 to 160.1 min, while P95 response decreases from 582.9 to 475.8 min. The corresponding reductions are 15.8% and 18.4%. The modeled energy index also falls, from 449.7 to 416.7, and recovery time falls from 626.5 to 517.4 min. These changes amount to 7.3% and 17.4%, respectively. For each of these four measures, the direction of the paired difference favors DT-MARL in all 50 replications, and the Holm-adjusted Wilcoxon tests remain below 0.001. Median paired reductions are 30.1 min for mean response, 98.9 min for P95 response, 30.2 units for the energy index, and 97.5 min for recovery. On-time delivery behaves differently. Its increase is only 0.65 percentage points and does not reach statistical significance after adjustment p=0.053.

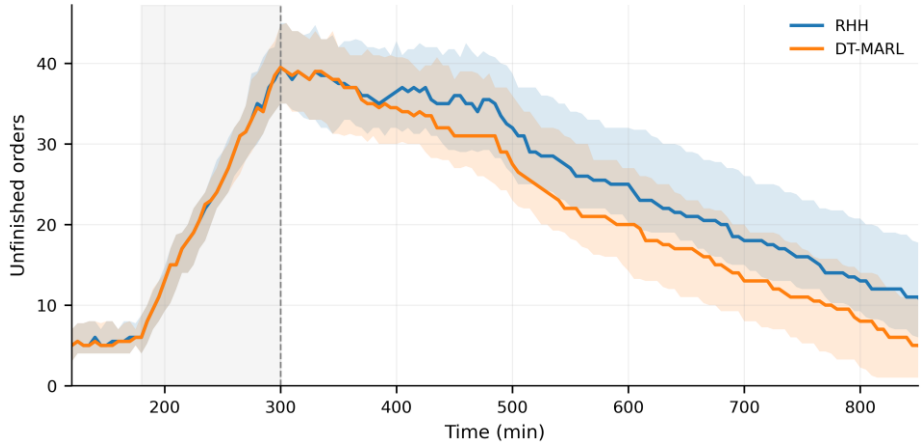


Figure 3. Compound-disruption backlog trajectory. Lines show medians and ribbons show the interquartile range across 50 paired replications.

The backlog trajectories help explain why the response-time statistics diverge. During the disruption, both methods accumulate unfinished orders at a similar rate. After the disturbance window ends, the queue under DT-MARL begins to decline earlier and remains below the RHH trajectory. Vehicle and UAV travel speeds are unchanged between the two methods. The difference therefore comes from assignment timing: projected-state decisions avoid some choices that would not begin service until road, airspace, or endurance conditions have worsened.

5.3 Ablation: what the digital twin contributes

Table 4. Ablation results in the compound regime (mean ± 95% CI).

Metric	DT-MARL	MARL w/o DT	Reduction vs MARL w/o DT
Mean response (min)	160.1±10.8	205.3±14.3	22.0%
P95 response (min)	475.8±34.0	618.1±37.2	23.0%
Energy index	416.7±16.9	459.1±18.1	9.3%
Recovery (min)	517.4±34.8	660.9±39.0	21.7%
Rejected infeasible UAV proposals (%)	0.0	16.9	Eliminated

Note. Percentage reductions are measured relative to MARL w/o DT. “Rejected infeasible UAV proposals”refers to UAV actions proposed by the current/stale-state policy but blocked by the execution mask before dispatch. None of these actions is executed.

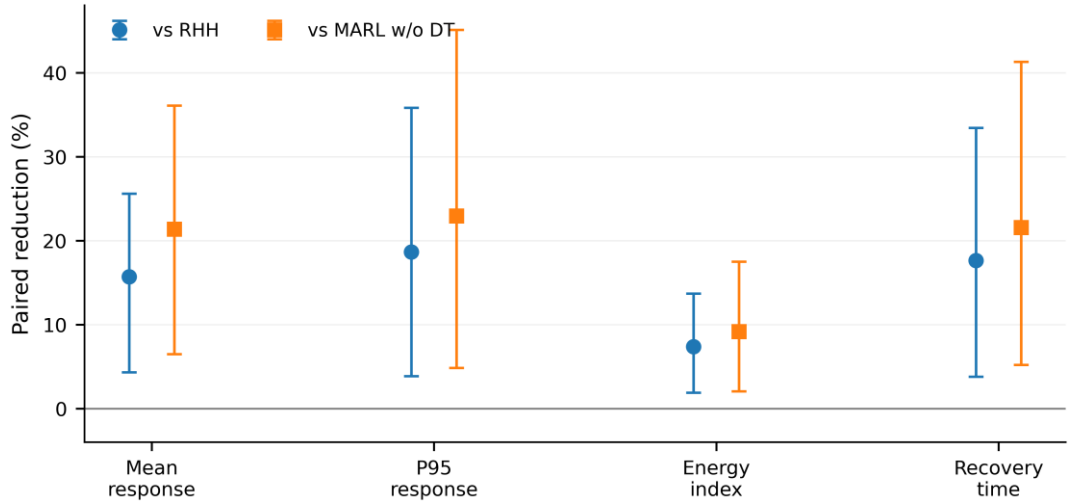


Figure 4. Paired reductions delivered by DT-MARL relative to RHH and MARL w/o DT in the compound regime.

The ablation comparison keeps the learning schedule, state encoding, action structure, reward, and execution mask unchanged. Only execution-time projection is removed. Under the compound disturbance, mean response then increases from 160.1 to 205.3 min, and recovery increases from 517.4 to 660.9 min. The no-projection policy also proposes infeasible UAV actions for 16.9% of orders, although these actions are intercepted by the mask before execution. Compared with MARL w/o DT, the projected-state model reduces mean response by 22.0%, P95 response by 23.0%, the energy index by 9.3%, and recovery time by 21.7%. Because the two learning variants differ primarily in the timing of the state information, this comparison provides a more direct test of the proposed mechanism than the heuristic comparison alone.

5.4 Robustness To Forecast Miscalibration

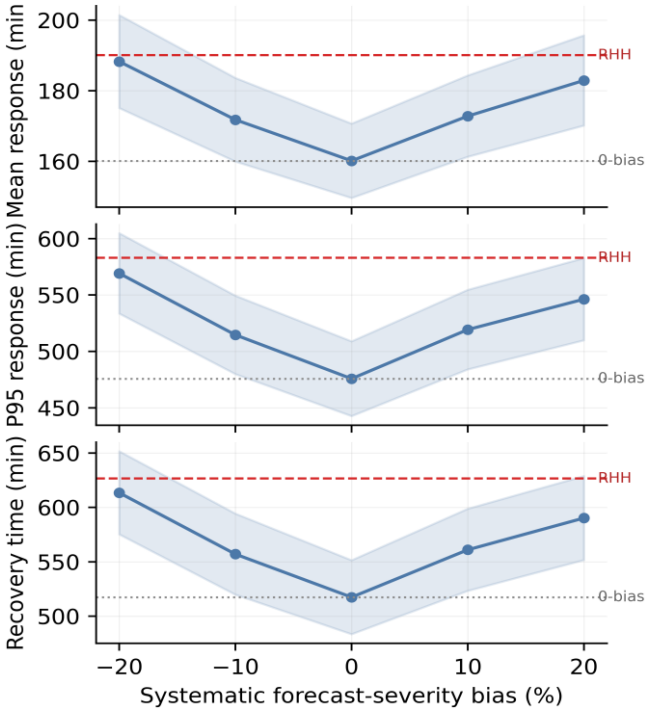


Figure 5. Forecast-bias sensitivity in the compound regime.

Forecast-severity bias is introduced after training so that the learned policy itself is not retrained for each perturbation. With a ±10% bias, mean response remains between 171.8 and 172.8 min, while recovery remains between 557.0 and 561.0 min. Even when the bias reaches±20%, both measures remain below the corresponding RHH values. Performance nevertheless worsens as the forecast moves away from the baseline in either direction. The experiment therefore indicates some tolerance to moderate severity error within the tested range. It does not provide evidence about errors in disturbance timing or location, correlated forecast failures, or conditions lying outside the training distribution.

5.5 Implications and Limitations

The operational value of projection depends on the length of the gap between assignment and service. If resources are lightly loaded and orders can start almost immediately, current-state dispatch may be sufficient. The benefit becomes more plausible when queues are long enough for road or air conditions to change before service begins. In that setting, UAV capacity is better treated as a selective resource for orders whose projected conditions justify its use. Evaluation should also include extreme delay and recovery behavior, because an on-time percentage alone may hide large differences in post-disruption performance.

Several features of the benchmark limit direct interpretation of the numerical gains. Ground vehicles and UAVs operate as parallel depot-to-node resources rather than as truck-UAV launch-and-recovery teams. Disturbances are represented by discrete regimes, and the demand stream is simulated.

The reported percentage improvements should therefore be interpreted as within-model algorithmic effects. They are not forecasts of savings for an actual county, logistics firm, or aircraft platform. Using common random numbers improves the precision of paired comparisons, but it does not reduce uncertainty about the size of a real-world effect.

The operational twin is intentionally simple. It updates resource release times and disturbance states and projects them to a candidate start time. It does not ingest live telemetry, spatially resolved wind forecasts, detailed battery dynamics, or probabilistic forecast distributions. A field implementation would require these components to be estimated and validated from operating data. The present learning model also fixes the reward weights and the state discretization. The sensitivity of the results to those choices has not been examined.

External validation would require a different level of data. At minimum, local travel times, parcel-size distributions, regulatory constraints, launch and landing facilities, and vehicle-specific energy curves would need to be calibrated. Testing on operating days not used during calibration would also be necessary. Larger fleets introduce an additional question because the state and action spaces increase quickly; the current computation-time results should therefore not be extrapolated to large-scale deployment.

The experiments point to a fairly specific explanation for the observed difference. The twin changes the state presented to the learner by estimating conditions at service start. The ablation removes that projection while keeping the remaining learning structure fixed. The forecast-bias experiment then weakens the accuracy of the projected information. Together, the two tests indicate where the reported benefit comes from and also show the limits of the evidence available from the present benchmark.

6. Conclusion

This study examines whether dynamic air-ground dispatch should rely on the state observed when an order is assigned or on the state expected when the selected resource begins service. We implement the second option through an execution-time projection layer linked to a centralized cooperative Q-learning policy. The two approaches behave similarly when operating conditions remain stable. Their performance separates when queues extend across changes in road congestion, airspace access, or UAV endurance. In the compound disturbance, the projected-state policy produces lower mean and tail response times, lower modeled energy use, and faster backlog recovery. The ablation indicates that these differences are associated with the timing of state information rather than with the learning structure alone. The forecast-bias experiment also suggests that moderate error in disturbance severity does not immediately remove the benefit. These findings remain limited to the simulated setting. Application to field operations will require calibrated travel and energy data, live or forecast state information, tests of reward and discretization choices, and evaluation with larger fleets.

Acknowledgements

This work was supported by the Anyang Science and Technology Plan Project (2026C02GH098) and the Henan Provincial Key Scientific Research Project for Higher Education Institutions (27AQ630001).

References

[1] Murray, C. C., & Chu, A. G. (2015). The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. Transportation Research Part C: Emerging Technologies, 54, 86–109.

[2] Agatz, N., Bouman, P., & Schmidt, M. (2018). Optimization approaches for the traveling salesman problem with drone. Transportation Science, 52(4), 965–981.

[3] Carlsson, J. G., & Song, S. (2018). Coordinated logistics with a truck and a drone. Management Science, 64(9), 4052–4069.

[4] Macrina, G., Di Puglia Pugliese, L., Guerriero, F., & Laporte, G. (2020). Drone-aided routing: A literature review. Transportation Research Part C: Emerging Technologies, 120, 102762.

[5] Zhang, J., Campbell, J. F., Sweeney, D. C., & Hupman, A. C. (2021). Energy consumption models for delivery drones: A comparison and assessment. Transportation Research Part D: Transport and Environment, 90, 102668.

[6] Gao, J., Zhen, L., & Wang, S. (2023). Multi-trucks-and-drones cooperative pickup and delivery problem. Transportation Research Part C: Emerging Technologies, 157, 104407.

[7] Gu, R., Liu, Y., & Poon, M. (2023). Dynamic truck-drone routing problem for scheduled deliveries and on-demand pickups with time-related constraints. Transportation Research Part C: Emerging Technologies, 151, 104139.

[8] Meng, S., Li, D., Liu, J., & Chen, Y. (2024). The multi-visit drone-assisted routing problem with soft time windows and stochastic truck travel times. Transportation Research Part B: Methodological, 190, 103101.

[9] She, R., & Ouyang, Y. (2024). Hybrid truck-drone delivery under aerial traffic congestion. Transportation Research Part B: Methodological, 185, 102970.

[10] Xing, J., Guo, T., & Tong, L. (2024). Reliable truck-drone routing with dynamic synchronization: A high-dimensional network programming approach. Transportation Research Part C: Emerging Technologies, 165, 104698.

[11] Guo, X., Li, Y., Cui, S., & Zhong, R. Y. (2026). Digital twin-enabled cooperative autonomous truck platooning framework for modeling, cost estimation, and decision-making in transportation systems. Transportation Research Part E: Logistics and Transportation Review, 207, 104596.

[12] Murugaiyan, P., Kundu, T., Wagner, S. M., & Sundarakani, B. (2026). Digital twin architecture for drone logistics operations. IEEE Transactions on Engineering Management, 73, 3738–3753.

[13] Shou, Z., Chen, X., Fu, Y., & Di, X. (2022). Multi-agent reinforcement learning for Markov routing games: A new modeling paradigm for dynamic traffic assignment. Transportation Research Part C: Emerging Technologies, 137, 103560.

[14] Mak, S., Xu, L., Pearce, T., Ostroumov, M., & Brintrup, A. (2023). Fair collaborative vehicle routing: A deep multi-agent reinforcement learning approach. Transportation Research Part C: Emerging Technologies, 157, 104376.

[15] Li, Y., Li, X., & Gao, L. (2025). Real-time scheduling for production-logistics collaborative environment using multi-agent deep reinforcement learning. Advanced Engineering Informatics, 65, 103216.