

Does Graph-Aware Context Improve Airport Arrival-Delay Forecasting? A Leakage-Audited Origin-Day Benchmark of U.S. Flights with Cross-Hub Transfer

Soufiane Momtaz

ENSET Mohammedia, Hassan II University of Casablanca, Morocco

Abstract

Reliable evaluation of AI forecasting models in operational transportation systems requires leakage control, semantically defensible prediction units, and transfer-aware validation. This paper introduces a leakage-audited origin-day benchmark for airport hub arrival-delay forecasting from arrival-only airline panels. The benchmark defines the origin-day as the finest defensible unit, excludes ex post delay-cause fields from predictive inputs, constructs a temporal heterogeneous hub graph, and evaluates local and relational XGBoost predictors under calendar-forward, rolling-origin, and fixed-interface destination-transfer protocols. The final locked evaluation uses 130,606 inbound American Airlines and Delta Air Lines arrivals to John F. Kennedy International Airport from 2022 to 2024, with an American Airlines Hartsfield–Jackson Atlanta International Airport panel reserved for transfer control. Under the final locked evaluation, the test evidence does not establish a robust advantage of the full graph-aware model over the local baseline: on the JFK 2024 support-filtered test set, full graph-aware WMAE is 20.136 minutes versus 20.108 for the local model, with a 7-day calendar-block bootstrap contrast of +0.029 minutes, 95% CI [-0.581, 0.635], Holm-adjusted $p = 0.991$. Hub-only augmentation significantly worsens WMAE, while carrier-origin features recover performance relative to hub-only without making the full model dominate the local baseline. Classification, rolling-origin robustness, and transfer-safe JFK–ATL controls lead to the same conservative conclusion. The contribution is therefore a reproducible, leakage-audited benchmark that shows where graph-aware context is operationally plausible but not automatically beneficial.

Keywords

Machine learning, trustworthy AI, leakage-audited modelling, temporal graphs, heterogeneous graphs, benchmark evaluation, flight delay forecasting, air transportation, transfer evaluation, negative results

1. Introduction

Reliable evaluation of artificial-intelligence forecasting models in operational transportation systems requires more than predictive accuracy. It also requires temporal leakage control, a prediction unit whose semantics are supported by the available data, and validation protocols that reveal whether a model remains useful outside the development setting. These requirements are especially important in air transportation, where delay forecasts may inform hub monitoring, resource planning and operational diagnostics, yet many available data panels are aggregated and do not contain trajectory revisions, sector traversals, runway-state changes or controller interventions.

Airport hub arrival-delay forecasting is naturally relational. The delay state of one inbound origin on a given day is shaped by its own recent history, by the wider state of the destination hub, by traffic from other origins, by carrier-origin activity and by the daily congestion regime. Classical work on delay propagation already showed that delays interact across airport networks rather than remaining isolated local events [1, 2]. Recent reviews and graph-based studies further confirm the field-wide move toward networked, causal and graph-aware formulations [3, 4, 5, 6, 7, 8, 9, 10, 11]. However, relational modelling is only scientifically defensible when the graph representation respects the information structure of the source data and when positive performance claims survive a locked temporal evaluation.

ICAIW 2026: Workshops at the 9th International Conference on Applied Informatics 2026, October 13–17, 2026, Orlando, United States

✉ soufiane.momtaz@gmail.com (S. Momtaz)

🆔 0009-0004-5650-2488 (S. Momtaz)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

This paper introduces a leakage-audited origin-day benchmark for airport hub arrival-delay forecasting from arrival-only airline panels. The benchmark uses the *origin-day* as the finest defensible prediction unit: all flights from one origin to one destination hub on one calendar day. This abstraction is deliberately conservative. It retains a relational daily state around the hub while avoiding unsupported claims about flight-event streams, within-day sequencing, trajectory memory or air-traffic-control actions that are not present in the panels.

The primary corpus comprises inbound arrivals of American Airlines (AA) and Delta Air Lines (DL) to John F. Kennedy International Airport (JFK) from 2022 to 2024. A separate American Airlines inbound-arrival panel to Hartsfield–Jackson Atlanta International Airport (ATL) during the same period is allocated to destination-holdout transfer assessment. The benchmark analyzes next-day mean-arrival-delay regression and next-day disruption classification under three complementary protocols: a strict calendar-year forward split, seven rolling quarter-ahead windows, and a transfer-safe ATL protocol restricted to variables observable at both destinations.

The paper is organized around four questions. **RQ1**: what leakage-audited benchmark can an arrival-only hub panel support without overclaiming temporal semantics? **RQ2**: does full graph-aware context robustly improve next-day arrival-delay prediction over a strong local baseline? **RQ3**: what do hub-only and carrier-origin ablations reveal about the relational feature blocks? **RQ4**: what can and cannot be inferred when JFK-trained models are evaluated on ATL through a common observable interface? These questions frame the work as a benchmark and evaluation study, not as a new graph neural architecture.

This paper makes five contributions. First, it performs a leakage audit appropriate for arrival-statistics panels and shows that delay-cause columns are *ex post* decompositions of realised arrival delay, not *ex ante* predictors. Secondly, it formalises a temporal heterogeneous origin-day graph with dynamic origin-day, carrier-origin-day and destination-day nodes plus static origin and carrier nodes. Thirdly, it defines a strict calendar-year split and a seven-window rolling-origin protocol. Fourthly, it benchmarks four feature families with identical XGBoost candidate rosters: local only, local plus hub, local plus carrier-origin, and full graph-aware context. Fifthly, it adds a fixed-interface JFK–ATL control and a dependence-aware calendar-block bootstrap that prevents unsupported promotion of small point-estimate differences.

The final empirical message is deliberately conservative. Full graph-aware context does not establish a robust JFK regression advantage over the local baseline; it also does not establish a robust classification advantage and is not temporally robust across seven rolling windows. Hub-only context is harmful in the primary run, while carrier-origin features recover much of that loss relative to hub-only. The ATL transfer control bounds what can be claimed under a reduced common interface rather than proving a destination-general graph advantage. Figure 1 summarises the locked benchmark construction and evaluation flow.

2. Related work

Three recent reviews help position this study. Li et al. [3] synthesise flight-delay propagation research from a data, modelling and application perspective. Wandelt et al. [4] review machine-learning work on flight-delay prediction more broadly. Sun et al. [5] extend the discussion to network-delay prediction and emerging AI paradigms. Taken together, these reviews show a field moving from isolated flight- or airport-level prediction towards networked, causal and graph-based formulations, but they also show that benchmark design, leakage control and semantically defensible granularity are rarely the main object of study.

Long before the recent graph-learning wave, delay propagation was already analysed as a coupled network process. Analytical and queueing-based studies modelled how local congestion at one airport produces ripple effects across airport networks [1]. Network analyses later showed systemic propagation patterns at the scale of the United States airport system [2]. Subsequent work characterised propagation structure and causal dependence more explicitly, including airport-network characterisation [14], nonlinear and cross-mapped propagation inference [15], airline-level daily propagation models [16]

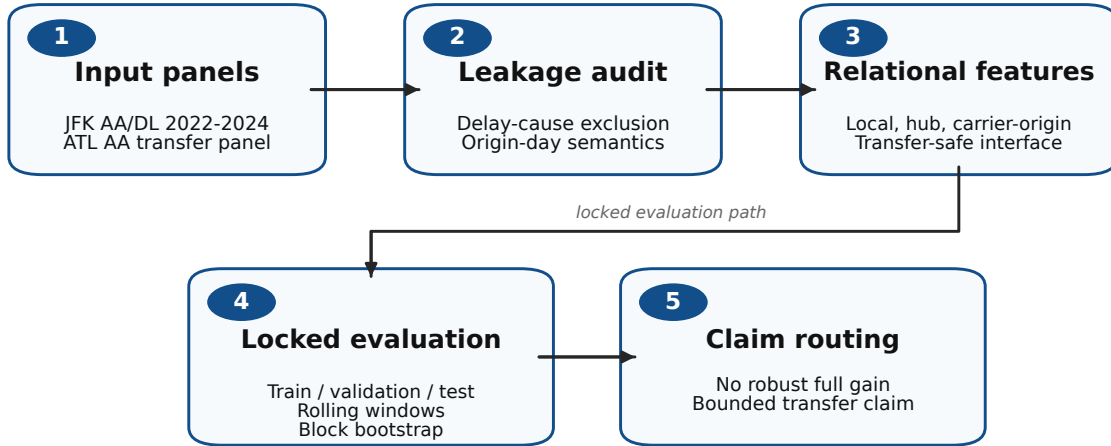


Figure 1: Locked benchmark flow. Arrival panels are leakage-audited and converted to an origin-day relational representation, evaluated under a locked temporal protocol, and routed to conservative claims when the evidence does not support a robust full graph-aware gain.

and causal risk modelling for flight delays [17]. These studies motivate relational modelling, but they mostly operate at airport-network or airline-network level rather than at the benchmark-design level addressed here.

A parallel predictive literature studied flight or airport delay forecasting with deep, multi-source or explainable models. Representative examples include deep flight-delay prediction on large operational datasets [18] and multi-source explainable machine learning that combines operational, airport and weather information [19]. Within graph-based aviation prediction itself, the field has moved from airport-network recurrent/graph hybrids to more explicit spatio-temporal and attention-based architectures. Examples include deep graph-embedded recurrent models for airport delay prediction [20], graph-to-sequence network-wide delay forecasting [6], nationwide airport throughput prediction with graph-attention recurrence [7], relation-attention models for 24-hour network-wide delay prediction [8], causality-guided graph convolution for propagated-delay prediction [11], spatiotemporal propagation networks [9] and long-horizon linear diffusion graph networks [10]. The common pattern is clear: graph structure can be useful when delay states are genuinely coupled through shared hub conditions, traffic flows or propagation channels.

From a graph-learning standpoint, the benchmark sits closer to typed and temporal relational modelling than to homogeneous static graph classification. Relational graph convolution offers a canonical typed-message-passing formulation [21]; heterogeneous graph transformers extend this idea with type-dependent attention [22]; and temporal graph attention extends inductive representation learning to dynamic graphs [23]. This paper does not claim one of these architectures as the main methodological contribution. Instead, they serve as reference points showing that a principled heterogeneous temporal benchmark is a prerequisite for any stronger model class.

Finally, explanation and feature diagnostics require caution. For tree ensembles, SHAP remains a practical attribution method [13]; for graph-native models, GNNExplainer and GraphFramEx highlight that explanation quality itself must be evaluated [24, 25]. The present paper therefore avoids treating post-hoc feature importance as proof of a causal graph benefit. Its primary diagnostic evidence comes from locked feature-block ablations and calendar-block uncertainty intervals.

Table 1

Source-panel and primary graph cardinalities before target-support filtering.

Object	Count
Inbound JFK flights, AA/DL	130,606
Inbound ATL transfer flights, AA	23,070
JFK origins	40
ATL origins	10
Days	1,096
JFK origin-day nodes	31,372
JFK carrier-origin-day nodes	39,561
JFK destination-day nodes	1,096
Origin-day next-day edges	31,187
Carrier-origin-day next-day edges	39,291

3. Data provenance and leakage-audited design

The benchmark is derived from three supplied *Detailed Statistics Arrivals* destination panels stored as HTML-based .xls exports. The footer of each export states “SOURCE: Bureau of Transportation Statistics”. Two exports cover inbound arrivals to JFK for AA and DL from 1 January 2022 to 31 December 2024. The third covers inbound AA arrivals to ATL over the same period and is reserved for external transfer analysis rather than for primary model selection. Each row corresponds to one completed arrival appearing in the destination panel and provides carrier code, flight number, tail number, origin airport, scheduled and actual arrival times, elapsed time, arrival delay, wheels-on time, taxi-in time, and five delay-cause columns.

The raw exports were supplied to the study as files rather than harvested through a documented public API. The manuscript therefore treats them as source-limited inputs: the paper works with derived benchmark tables and evaluation artefacts, not with redistributed raw panels. All time fields are used only in the destination-local clock-time form in which they appear in the panels, and the benchmark aggregates them to the day level rather than reconstructing cross-time-zone event sequences. By construction, the benchmark covers only arrivals that appear in the destination panels; non-arrivals that never materialise as arrival records are outside scope.

The JFK primary corpus contains 130,606 flights, 40 origins and 1,096 consecutive days. No exact duplicates were found, and no duplicates were found under a reasonable flight key consisting of carrier, date, flight number, origin, destination, and scheduled arrival time. The primary graph derived from this corpus contains 31,372 origin-day nodes, 39,561 carrier-origin-day nodes and 1,096 destination-day nodes. The transfer panel contains 23,070 ATL-bound AA arrivals and 10 origins. Table 1 summarises the source scale.

The most important leakage-audit finding concerns the delay-cause columns. For every JFK flight with realised arrival delay of at least 15 minutes, the sum of `delay_carrier_min`, `delay_weather_min`, `delay_nas_min`, `delay_security_min` and `delay_late_aircraft_min` exactly equals the observed arrival delay. These fields are therefore ex post decompositions of the target, not ex ante predictors. They are retained only for auditability and are excluded from all predictive features, rolling statistics, selectors and diagnostics.

The second audit finding concerns temporal meaning. The panels contain day-level schedule and realised-time information, but they do not reliably identify aircraft rotations beyond individual tails, nor do they contain departure queues, runway state changes, sector traversals or trajectory amendments. The finest defensible benchmark unit is therefore not the flight-event stream or the trajectory version, but the origin-day. This is the central semantic decision of the paper.

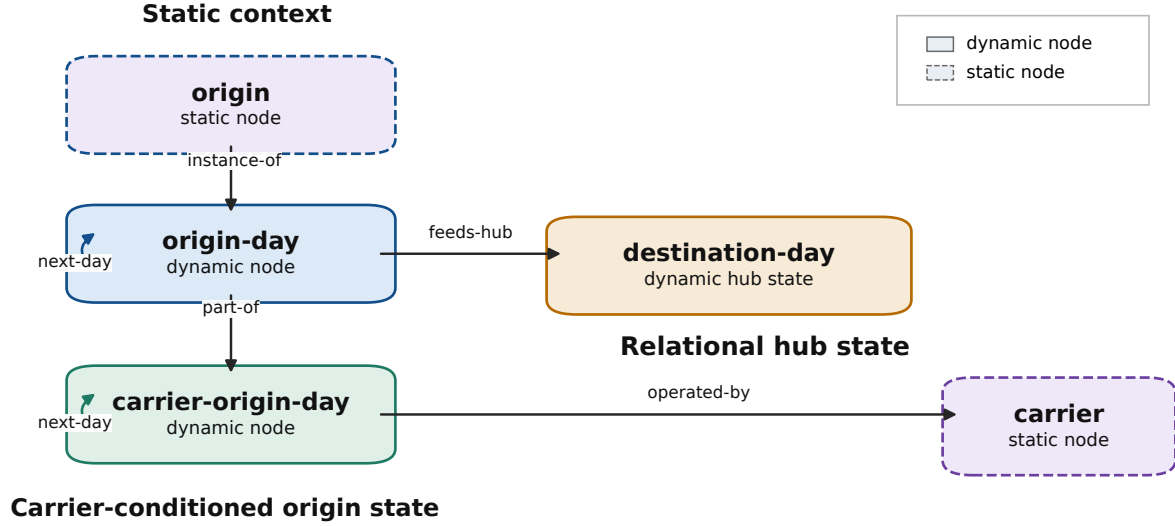


Figure 2: Temporal heterogeneous origin-day graph. The graph supports day-level local, hub, and carrier-origin context while deliberately avoiding trajectory or within-day control semantics that are absent from the source panels.

4. Temporal heterogeneous origin-day graph

4.1. Node and edge semantics

Let o denote an origin airport, c a carrier, h a destination hub and t a calendar day. The dynamic node types are as follows.

- **origin-day** $v_{o,h,t}^{od}$: all flights from origin o to hub h on day t ;
- **carrier-origin-day** $v_{c,o,h,t}^{cod}$: the subset of those flights operated by carrier c ;
- **destination-day** $v_{h,t}^{dd}$: the aggregated state of the destination hub on day t .

Two static node types retain longer-term context: origin nodes v_o^o and carrier nodes v_c^c . The resulting graph is heterogeneous and temporal,

$$\mathcal{G} = (\mathcal{V}, \mathcal{E}, \tau, \psi),$$

where τ maps nodes to types and ψ maps edges to relation types. Edges connect origin-day nodes to their carrier-origin-day components, to the same-day destination-day hub node, and to the next-day node for the same origin and destination. Carrier-origin-day nodes are also linked forward in time and to their static carrier node. Figure 2 gives the graph schema.

4.2. Feature blocks

The final protocol evaluates four feature families. M_L uses local origin-day, calendar, and lagged origin-history features. M_H adds hub-state and hub-history features to M_L . M_C adds carrier-origin features to M_L . M_G is the full graph-aware representation, combining local, hub, and carrier-origin blocks. Before one-hot expansion, the primary JFK feature counts are 52 for M_L , 77 for M_H , 72 for M_C , and 97 for M_G ; after train-fitted one-hot expansion, they become 71, 96, 91, and 116, respectively. The transfer-safe interface uses 45 raw local predictors and 60 raw local-plus-hub predictors, expanding to 64 and 79 columns respectively.

All features are computed from the closed-source day t or from histories before or at t . No target-day $t + 1$ value is used as a predictor. Origin identity is fitted on the training period only, with unknown validation or test origins handled explicitly rather than retrospectively refitting the encoder.

Table 2

Locked support-filtered evaluation populations. The JFK TRAIN and VALIDATION rows are used for model fitting and selection; the JFK 2024 and ATL 2024 rows are opened only in the locked test stage.

Role	Period	Destination	Rows	Notes
TRAIN	2022	JFK	5,780	support ≥ 3
VALIDATION	2023	JFK	6,143	support ≥ 3
TEST primary	2024	JFK	6,180	support ≥ 3
TEST sensitivity	2024	JFK	10,937	all labelled
TEST transfer	2024	ATL	1,157	support ≥ 3
TEST transfer sensitivity	2024	ATL	2,486	all labelled

4.3. Targets, support and splits

For each source node $v_{o,h,t}^{od}$, the regression target is the mean arrival delay on the exact next calendar day for the same origin and destination:

$$y_{o,h,t+1}^{reg} = \frac{1}{|\mathcal{F}_{o,h,t+1}|} \sum_{f \in \mathcal{F}_{o,h,t+1}} d_f^{arr},$$

where $\mathcal{F}_{o,h,t+1}$ denotes the flights from origin o to destination h on day $t + 1$ and d_f^{arr} denotes realised arrival delay in minutes.

The classification target flags next-day disruption:

$$y_{o,h,t+1}^{cls} = \mathbb{I} \left[y_{o,h,t+1}^{reg} \geq 15 \right].$$

The support rule is an inclusion filter, not part of the class definition. The confirmatory population requires at least three next-day flights, `next_n_flights` ≥ 3 , in TRAIN, VALIDATION, and TEST. Rows with at least one but fewer than three next-day flights are retained only in an all-labelled sensitivity using already frozen primary models and thresholds. Rows with no exact next-calendar-day flight are unlabelled and excluded.

Temporal evaluation follows a strict calendar-year split. Source days in 2022 define training, source days in 2023 define validation, and source days in 2024 define test. Splits are defined by source, or forecast-issue, date; no predictor from a target date is included in the source features. Table 2 reports the locked support-filtered evaluation populations.

The benchmark is not a single-season snapshot: it covers three complete calendar years and is evaluated through a forward split and seven rolling-origin windows rather than through random temporal mixing.

5. Experimental design

5.1. Models and selection

The four compared models share the same XGBoost candidate roster [12]; only the admitted feature block differs. The local model M_L is the reference baseline. The hub-only model M_H tests whether one-hop destination-day state helps when added to local features. The carrier-origin model M_C tests carrier-conditioned origin context without hub-state augmentation. The full model M_G combines local, hub, and carrier-origin features and is the only model described as full graph-aware in the primary experiment. All predictive models remain gradient-boosted trees over engineered relational features; the paper does not claim graph neural message passing.

For reproducibility, Table 3 reports the selected XGBoost configuration used by the final common-configuration locked evaluation. The candidate roster contained 12 configurations and was searched on VALIDATION only. The same selected candidate, H01, was then loaded unchanged for the primary

Table 3

Selected XGBoost configuration for the final common-configuration locked evaluation. Selection was performed on VALIDATION only and frozen before TEST; no model was refit at test time.

Parameter	Selected value
Candidate identifier	H01
n_estimators	250
max_depth	3
learning_rate	0.05
min_child_weight	1
subsample	0.8
colsample_bytree	0.8
gamma	0.0
reg_alpha	0.0
reg_lambda	1.0
max_bin	256
tree_method, device	hist, CPU
booster, grow policy	gbtree, depthwise
random_state	20260908

JFK common-configuration regression and classification results for M_L , M_H , M_C and M_G , and for the fixed-interface transfer-safe local and hub models. Independent rolling sensitivity selections are archived in the selected-model manifest and do not change the claim-determining common-configuration results.

Model fitting is restricted to TRAIN, candidate selection and classification-threshold selection are restricted to VALIDATION, and TEST is opened only after feature sets, model configurations, thresholds and claim-revision rules have been sealed. Classification thresholds maximise validation F1 and are then frozen for TEST. The common-configuration results select the same candidate family across variants; an independent-selection sensitivity is retained in the evidence package but does not change the main claim.

5.2. Metrics and uncertainty

Regression is reported with mean absolute error (MAE) and flight-weighted MAE (WMAE), the latter weighting each source node by its number of next-day flights. Classification is reported with area under the precision-recall curve (AUPRC), area under the receiver operating characteristic curve (AUROC) and F1 at the validation-selected threshold.

The confirmatory uncertainty analysis uses paired calendar-block bootstraps with block lengths of 1, 7 and 14 days and 5,000 replicates per block length. The claim-determining family is the 7-day block analysis with Holm adjustment. The primary regression contrasts are $M_G - M_L$, $M_H - M_L$ and $M_G - M_H$ in WMAE minutes. The primary classification contrasts use AUPRC. Rolling-origin robustness is summarised by mean performance and by the number of windows in which M_G beats M_L .

5.3. Locked evaluation governance

The results reported here are produced by a locked test evaluation protocol. The protocol sealed the final feature construction, model-selection outputs, thresholds, bootstrap design, and mechanical claim-routing rules before opening TEST. The TEST stage loaded the validation-selected models directly, performed zero model refits, generated predictions, and then reconstructed the full metric and bootstrap tables through an independent consistency verifier. The verifier confirmed the locked metrics, bootstrap summaries, prediction files, and claim-routing outputs without changing any result.

Table 4

JFK 2024 support-filtered primary results. Lower is better for MAE/WMAE; higher is better for AUPRC/AUROC/F1.

Variant	Description	WMAE	MAE	AUPRC	AUROC	F1
M_L	Local	20.108	21.185	0.362	0.654	0.423
M_H	Local + hub	21.628	22.822	0.336	0.643	0.413
M_C	Local + carrier-origin	19.941	20.928	0.369	0.655	0.426
M_G	Full graph-aware	20.136	20.952	0.341	0.650	0.427

5.4. External transfer protocol

The ATL add-on stresses the benchmark outside the destination used for model development. Because ATL contains only AA arrivals, the full JFK feature interface cannot be transferred without inventing DL-specific carrier-origin blocks that do not exist in the ATL panel. A transfer-safe interface is therefore defined in the narrow sense of schema safety:

$$\phi^{TS}(u_t) = \phi_{JFK}^{full}(u_t) \cap \phi_{ATL}^{obs}(u_t),$$

where ϕ_{JFK}^{full} is the full JFK feature map and ϕ_{ATL}^{obs} is the set of predictors actually computable from ATL observations at forecast time. The transfer-safe local model M_L^{TS} uses the destination-agnostic local subset of this interface. The transfer-safe hub model M_H^{TS} augments it with shared one-hop hub-state features. Both are trained on JFK 2022, tuned only on JFK 2023, and applied to JFK 2024 and ATL 2024 without destination-specific refitting.

6. Results

6.1. Primary JFK 2024 results

Table 4 reports the primary results on the 2024 JFK support-filtered TEST set. The central result is negative but informative: the full graph-aware model M_G does not improve WMAE over the local baseline M_L . The point difference is small and in the wrong direction for an improvement claim: 20.136 minutes for M_G versus 20.108 minutes for M_L . The carrier-origin ablation M_C has the best point WMAE and AUPRC, but its direct contrast against M_L was not part of the claim-determining Holm family and is treated as descriptive ablation evidence.

Figure 3 visualises the same result. Full graph-aware context is not a reliable winner. The useful scientific output is therefore not a positive graph-aware performance claim, but a final evidence boundary: relational feature engineering can be plausible while still failing to dominate a well-specified local baseline under locked temporal evaluation.

6.2. Confirmatory block-bootstrap contrasts

Table 5 gives the claim-determining 7-day calendar-block bootstrap results. The main contrast $M_G - M_L$ is +0.028786 minutes, with a 95% interval from -0.580885 to 0.635348 and a Holm-adjusted p -value of 0.991402. This excludes any robust conclusion that the full graph-aware model improves regression over the local baseline.

The ablation contrasts are sharper. The hub-only model is significantly worse than the local baseline by +1.520829 minutes. Adding carrier-origin features to the hub-only model recovers -1.492043 minutes relative to M_H . This supports a narrow block-level conclusion: carrier-origin features repair much of the hub-only degradation. It does not support the stronger conclusion that the full graph-aware model beats M_L .

For classification, M_G obtains an AUPRC of 0.340616 against 0.361976 for M_L . The full-minus-local AUPRC contrast is -0.021360 , with a 7-day block interval of $[-0.043984, 0.002908]$ and Holm-adjusted $p = 0.170366$. The final classification conclusion is therefore also *no robust full graph-aware gain*.

Locked primary test results

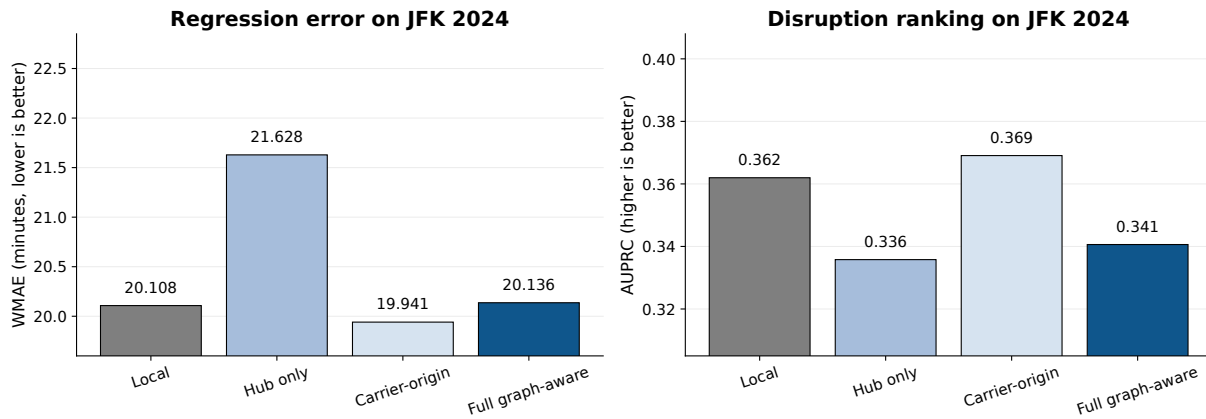


Figure 3: Primary JFK 2024 results. The full graph-aware model does not reduce WMAE relative to the local baseline and has lower AUPRC. The carrier-origin ablation is retained as descriptive block evidence, not as a confirmatory promoted claim.

Table 5

Locked claim-determining regression contrasts using 5,000 paired 7-day calendar-block bootstrap replicates. Positive WMAE contrasts mean the first model is worse than the comparator.

Contrast	Estimate	95% CI	Holm p
$M_G - M_L$	+0.028786	[-0.580885, 0.635348]	0.991402
$M_H - M_L$	+1.520829	[0.876995, 2.162127]	0.001200
$M_G - M_H$	-1.492043	[-2.050674, -0.985658]	0.001200

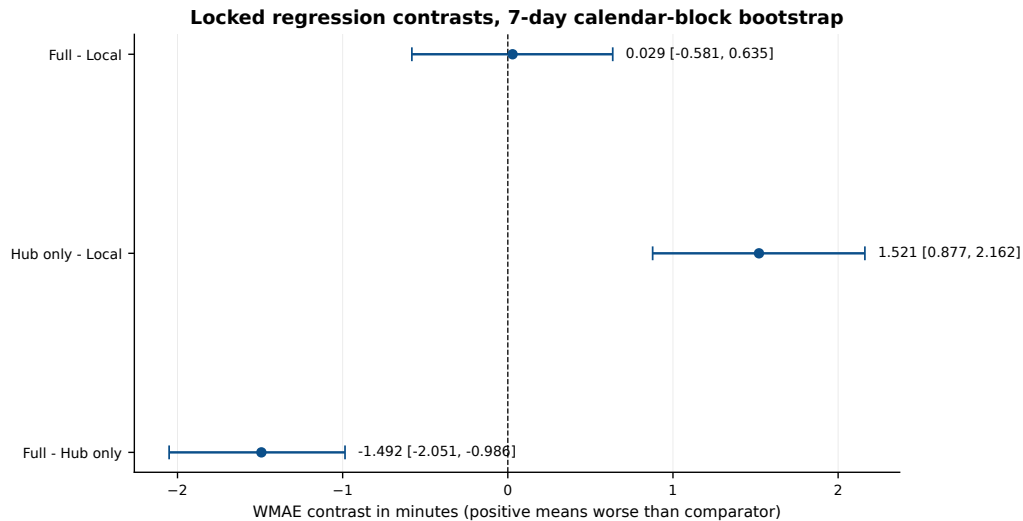


Figure 4: Locked 7-day block-bootstrap regression contrasts. The full model is statistically indistinguishable from the local baseline, hub-only context is harmful, and the carrier-origin increment inside the full model recovers performance relative to hub-only.

6.3. Rolling-origin multi-window robustness

A single held-out year can overstate or understate robustness if its operational regime is unusually easy or hard. The final evaluation includes seven expanding rolling-origin windows with quarter-ahead tests from 2023Q2 through 2024Q4. Table 6 summarises the common-configuration results.

The rolling outcome reinforces the conservative primary conclusion. In regression, $M_G - M_L$

Table 6

Seven-window rolling-origin summary under the common configuration. Wins are counted against M_L ; for WMAE, a win means lower error.

Task	Variant	Mean	Std.	Wins vs. M_L
WMAE	M_L	19.638	3.124	–
WMAE	M_H	20.157	3.334	1/7
WMAE	M_C	19.695	3.027	3/7
WMAE	M_G	19.892	3.446	3/7
AUPRC	M_L	0.338	0.122	–
AUPRC	M_H	0.337	0.126	3/7
AUPRC	M_C	0.341	0.116	5/7
AUPRC	M_G	0.336	0.117	3/7

Seven-window rolling-origin evaluation

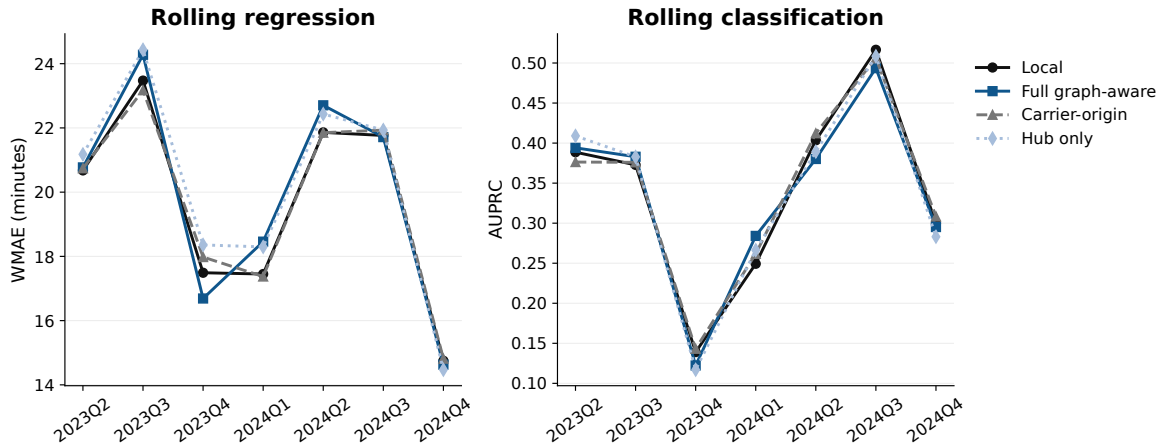


Figure 5: Rolling-origin results. The full graph-aware model is not temporally robust against the local baseline: it wins only 3/7 windows for WMAE and 3/7 windows for AUPRC.

Table 7

Fixed-interface transfer-safe results. The transferred hub model M_H^{TS} is the local transfer-safe interface plus shared hub-state features; full carrier-origin transfer is not evaluated because the ATL panel is AA-only.

Destination	Variant	WMAE	MAE	AUPRC	AUROC	F1
JFK	M_L^{TS}	20.146	21.210	0.363	0.657	0.425
JFK	M_H^{TS}	21.711	22.989	0.325	0.629	0.405
ATL	M_L^{TS}	22.060	23.488	0.565	0.677	0.610
ATL	M_H^{TS}	21.675	23.002	0.549	0.669	0.613

has a mean WMAE difference of +0.253814 minutes, and M_G wins only three of seven windows. In classification, the mean AUPRC difference is -0.002445 , and M_G again wins only three of seven windows. Figure 5 shows that the model ranking varies by quarter. This is precisely the type of instability that a robust benchmark should expose rather than hide.

6.4. Fixed-interface JFK–ATL transfer control

The transfer-safe control asks: does the shared local-plus-hub interface, trained at JFK and frozen after JFK validation, retain an advantage when applied to ATL? Table 7 reports the support-filtered results.

The 7-day block-bootstrap WMAE contrast $M_H^{TS} - M_L^{TS}$ is +1.565469 minutes at JFK, with 95% CI [0.814077, 2.411839]. At ATL, the same contrast is -0.384692 minutes with 95% CI $[-0.963058,$

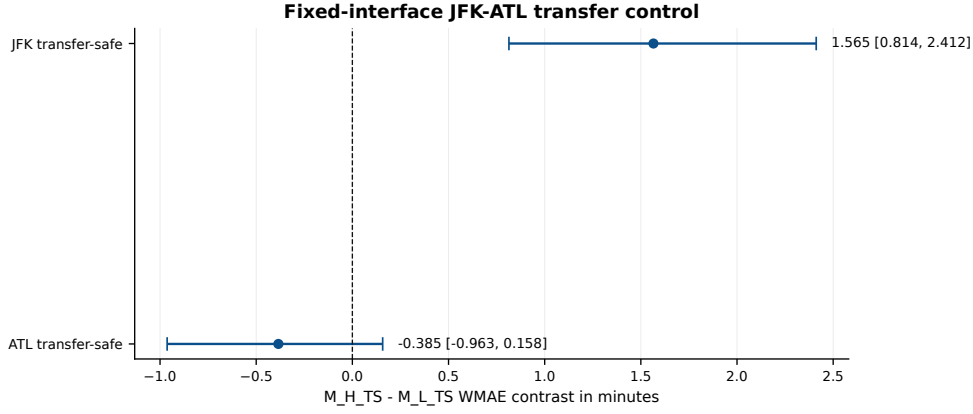


Figure 6: Fixed-interface JFK–ATL transfer control. The JFK transfer-safe hub contrast is already adverse; the ATL contrast crosses zero. The evidence therefore precludes attributing a lost graph-aware advantage specifically to the destination shift.

0.157583]. Since the transfer-safe hub interface is already worse than local at JFK, the ATL result cannot be interpreted as the disappearance of a robust full-interface relational advantage, and the common observable interface does not provide a robust, attributable destination-transfer advantage.

7. Discussion

Four scientific conclusions emerge from the final evaluation.

Firstly, the origin-day abstraction remains valuable. It is strong enough to support meaningful forecasting while remaining faithful to the input panels. It preserves relational information that a purely flight-independent framing would miss, but it does not pretend to model trajectories, sectors or event-level operations. This is a useful middle ground for many operational tables in aviation.

Secondly, graph-aware context is not automatically beneficial. The original motivation for relational features remains operationally plausible, but the final locked evaluation rejects the stronger claim that full graph-aware context robustly improves the primary JFK test set. This distinction matters for trustworthy AI: a representation can be semantically defensible and still fail to outperform a leaner baseline under a rigorous temporal protocol.

Thirdly, the ablation result is more informative than a binary “graph helps” conclusion. Hub-only features are harmful in the primary run. Carrier-origin features then recover most of that harm relative to hub-only, and the carrier-origin ablation has the best descriptive point estimate. This suggests that relational context must be decomposed rather than treated as a monolithic graph-aware block. Operationally, not every hub aggregate is useful simply because it is network-derived.

Fourthly, transfer interpretation must be interface-specific. The ATL control is transfer-safe only in a schema sense: every transferred predictor is computable from observed ATL fields at forecast time, with no synthetic carrier slots, destination-specific padding, or post-hoc imputation. It is not transfer-complete because it necessarily drops the carrier-origin channels that require a comparable multi-carrier panel. The transfer result therefore bounds portability under common observables rather than proving that all graph-aware hub models fail to transfer.

The benchmark also has limitations. The raw panels were source-limited files and are not redistributed with the manuscript; the controlled release focuses on derived benchmark artefacts and evaluation evidence. The graph is inbound and hub-centred, not a national network. Delay causes are leakage-prone and therefore excluded from predictive inputs, which removes one potentially informative but operationally illegitimate signal. The external ATL transfer uses the schema-intersection transfer-safe interface rather than the full JFK carrier-origin schema. Finally, the current paper uses gradient-boosted baselines over relational features rather than a full heterogeneous graph neural network because the benchmark must be defensible before more complex graph-native architectures are promoted.

8. Conclusion

This paper introduced a leakage-audited origin-day benchmark for airport hub arrival-delay forecasting. The benchmark is built from 130,606 inbound flights to JFK operated by AA and DL across 2022–2024, with an ATL AA panel used for fixed-interface transfer control. Its central contribution is a trustworthy AI evaluation design: target leakage is audited, the prediction unit is semantically defensible, temporal validation is calendar-forward, rolling-origin robustness is measured, and transfer is restricted to features observable at both destinations. Full graph-aware context does not establish a robust improvement over the local baseline on the JFK 2024 support-filtered test set. The main WMAE contrast is effectively zero and statistically unsupported; classification does not show robust AUPRC gain, rolling-origin windows do not support temporal robustness, and the fixed-interface ATL control precludes a stronger destination-transfer attribution. At the same time, the ablation evidence is scientifically useful: hub-only context is harmful, carrier-origin context recovers much of that harm, and relational design should therefore be audited at the feature-block level. The resulting origin-day benchmark provides a reproducible basis for future heterogeneous graph learning, destination adaptation, transfer evaluation and operational diagnostics in hub-centric arrival forecasting.

Declaration on Generative AI

During the preparation of this work, the author used AI tools (OpenAI ChatGPT) for language editing, LaTeX conversion support, consistency checking, and final version formatting assistance. The author reviewed and edited the content as needed and takes full responsibility for the final version.

References

- [1] Pyrgiotis, N., Malone, K.M. and Odoni, A. (2013) 'Modelling delay propagation within an airport network', *Transportation Research Part C: Emerging Technologies*, Vol. 27, pp. 60–75. doi: 10.1016/j.trc.2011.05.017.
- [2] Fleurquin, P., Ramasco, J.J. and Eguiluz, V.M. (2013) 'Systemic delay propagation in the US airport network', *Scientific Reports*, Vol. 3, Article 1159. doi: 10.1038/srep01159.
- [3] Li, C., Mao, J., Li, L., Wu, J., Zhang, L., Zhu, J. and Pan, Z. (2024) 'Flight delay propagation modeling: Data, methods, and future opportunities', *Transportation Research Part E: Logistics and Transportation Review*, Vol. 185, Article 103525. doi: 10.1016/j.tre.2024.103525.
- [4] Wandelt, S., Chen, X. and Sun, X. (2025) 'Flight delay prediction: A dissecting review of recent studies using machine learning', *IEEE Transactions on Intelligent Transportation Systems*, Vol. 26, No. 4, pp. 4283–4297. doi: 10.1109/TITS.2025.3528536.
- [5] Sun, M., Tian, Y., Li, J., Wu, C.-L., Peng, L. and Xu, S. (2026) 'A review of network delay prediction and advances in large language models for air traffic', *Artificial Intelligence Review*, Vol. 59, Article 36. doi: 10.1007/s10462-025-11400-w.
- [6] Bao, J., Yang, Z. and Zeng, W. (2021) 'Graph to sequence learning with attention mechanism for network-wide multi-step-ahead flight delay prediction', *Transportation Research Part C: Emerging Technologies*, Vol. 130, Article 103323. doi: 10.1016/j.trc.2021.103323.
- [7] Zhu, X., Lin, Y., He, Y., Tsui, K.-L., Chan, P.W. and Li, L. (2022) 'Short-term nationwide airport throughput prediction with graph attention recurrent neural network', *Frontiers in Artificial Intelligence*, Vol. 5, Article 884485. doi: 10.3389/frai.2022.884485.
- [8] Zheng, H., Wang, Z., Zheng, C., Wang, Y., Fan, X., Cong, W. and Hu, M. (2024) 'A graph multi-attention network for predicting airport delays', *Transportation Research Part E: Logistics and Transportation Review*, Vol. 181, Article 103375. doi: 10.1016/j.tre.2023.103375.
- [9] Wu, Y., Yang, H., Lin, Y. and Liu, H. (2024a) 'Spatiotemporal propagation learning for network-wide flight delay prediction', *IEEE Transactions on Knowledge and Data Engineering*, Vol. 36, No. 1, pp. 386–400. doi: 10.1109/TKDE.2023.3286690.

- [10] Wu, Y., Yang, J., Chen, X., Lin, Y. and Yang, H. (2024b) 'Long-term airport network performance forecasting with linear diffusion graph networks', *IEEE Transactions on Intelligent Transportation Systems*, Vol. 25, No. 11, pp. 18264–18278. doi: 10.1109/TITS.2024.3420423.
- [11] Sun, M., Tian, Y., Wang, X., Huang, X., Li, Q., Li, Z. and Li, J. (2024) 'Transport causality knowledge-guided GCN for propagated delay prediction in airport delay propagation networks', *Expert Systems with Applications*, Vol. 240, Article 122426. doi: 10.1016/j.eswa.2023.122426.
- [12] Chen, T. and Guestrin, C. (2016) 'XGBoost: A scalable tree boosting system', in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*, Association for Computing Machinery, New York, NY, USA, pp. 785–794. doi: 10.1145/2939672.2939785.
- [13] Lundberg, S.M. and Lee, S.-I. (2017) 'A unified approach to interpreting model predictions', in *Advances in Neural Information Processing Systems 30*, Curran Associates, Inc., Red Hook, NY, USA, pp. 4765–4774.
- [14] Li, Q. and Jing, R. (2021) 'Characterization of delay propagation in the air traffic network', *Journal of Air Transport Management*, Vol. 94, Article 102075. doi: 10.1016/j.jairtraman.2021.102075.
- [15] Guo, Z., Hao, M., Yu, B. and Yao, B. (2022) 'Detecting delay propagation in regional air transport systems using convergent cross mapping and complex network theory', *Transportation Research Part E: Logistics and Transportation Review*, Vol. 157, Article 102585. doi: 10.1016/j.tre.2021.102585.
- [16] Erdem, F. and Bilgiç, T. (2024) 'Airline delay propagation: Estimation and modeling in daily operations', *Journal of Air Transport Management*, Vol. 115, Article 102548. doi: 10.1016/j.jairtraman.2024.102548.
- [17] Truong, D. (2021) 'Using causal machine learning for predicting the risk of flight delays in air transportation', *Journal of Air Transport Management*, Vol. 91, Article 101993. doi: 10.1016/j.jairtraman.2020.101993.
- [18] Yu, B., Guo, Z., Asian, S., Wang, H. and Chen, G. (2019) 'Flight delay prediction for commercial air transport: A deep learning approach', *Transportation Research Part E: Logistics and Transportation Review*, Vol. 125, pp. 203–221. doi: 10.1016/j.tre.2019.03.013.
- [19] Pineda-Jaramillo, J., Munoz, C., Mesa-Arango, R., Gonzalez-Calderon, C. and Lange, A. (2024) 'Integrating multiple data sources for improved flight delay prediction using explainable machine learning', *Research in Transportation Business & Management*, Vol. 56, Article 101161. doi: 10.1016/j.rtbm.2024.101161.
- [20] Zeng, W., Li, J., Quan, Z. and Lu, X. (2021) 'A deep graph-embedded LSTM neural network approach for airport delay prediction', *Journal of Advanced Transportation*, Vol. 2021, Article 6638130, 15 pp. doi: 10.1155/2021/6638130.
- [21] Schlichtkrull, M., Kipf, T.N., Bloem, P., van den Berg, R., Titov, I. and Welling, M. (2018) 'Modeling relational data with graph convolutional networks', in Gangemi, A., Navigli, R., Vidal, M.-E., Hitzler, P., Troncy, R., Hollink, L., Tordai, A. and Alam, M. (Eds.): *The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, Proceedings*, Lecture Notes in Computer Science, Vol. 10843, Springer, Cham, pp. 593–607. doi: 10.1007/978-3-319-93417-4_38.
- [22] Hu, Z., Dong, Y., Wang, K. and Sun, Y. (2020) 'Heterogeneous graph transformer', in *Proceedings of The Web Conference 2020*, Association for Computing Machinery, New York, NY, USA, pp. 2704–2710. doi: 10.1145/3366423.3380027.
- [23] Xu, D., Ruan, C., Korpeoglu, E., Kumar, S. and Achan, K. (2020) 'Inductive representation learning on temporal graphs', in *International Conference on Learning Representations (ICLR 2020)*.
- [24] Ying, Z., Bourgeois, D., You, J., Zitnik, M. and Leskovec, J. (2019) 'GNExplainer: Generating explanations for graph neural networks', in *Advances in Neural Information Processing Systems 32*, Curran Associates, Inc., Red Hook, NY, USA, pp. 9244–9255.
- [25] Amara, K., Ying, Z., Zhang, Z., Han, Z., Zhao, Y., Shan, Y., Brandes, U., Schemm, S. and Zhang, C. (2022) 'GraphFramEx: Towards systematic evaluation of explainability methods for graph neural networks', in Rieck, B. and Pascanu, R. (Eds.): *Proceedings of the First Learning on Graphs Conference*, Proceedings of Machine Learning Research, Vol. 198, PMLR, pp. 44:1–44:23.