Vol. 1, No. 2, pp.: 33-46, 2023

UrbanFlowDB: A Multimodal Urban Mobility Database for Traffic, Transit, and Micromobility Intelligence

Xiaolong Pan[1], Ruoxi Jiang[2,*], Tao Cheng[3], Yanmei Xu[4]

[1] School of Transportation Engineering, Chang'an University, Xi'an 710064, China

[2] College of Civil Engineering, Fuzhou University, Fuzhou 350108, China

[3] School of Geographic Sciences, East China Normal University, Shanghai 200241, China

[4] School of Computer Science, Northwest A&F University, Yangling 712100, China

* *jiang.ruoxi@fzu.edu.cn*

Article Information	
Received	18 January 2023
Accepted	22 May 2023
DOI	https://doi.org/10.63646/datamind.2023.010204

Abstract

Urban mobility intelligence increasingly depends on the joint analysis of transit smart card transactions, taxi GPS probes, shared-bike trips, road-side sensor counts, and meteorological observations. Yet these five data sources are typically curated in isolation, stored in incompatible formats, indexed by incompatible spatial and temporal keys, and exposed under inconsistent privacy regimes, which makes integrated analytical workflows unnecessarily fragile. This article presents UrbanFlowDB, a multimodal urban mobility database that treats the database itself as the principal research artifact. We document the schema, the field dictionary, the spatiotemporal index family, the ingestion and quality control pipeline, the pseudonymization and ethics processing flow, and the reusable application programming interfaces that expose the integrated data to downstream models. The database is co-resident across a Parquet-plus-Delta lakehouse, a PostGIS-extended relational store, a Neo4j property graph for congestion-propagation analysis, and a pgvector index for trajectory similarity search; this polyglot layout is deliberately chosen because each mobility analytical pattern aligns most naturally with a different storage paradigm. We benchmark the database on a runnable urban experiment using one year of data from a Chinese second-tier city (1.42 billion transit taps, 396 million taxi GPS pings, 21.6 million dockless bike trips, 8.4 million sensor records, 215,860 weather observations) and demonstrate that UrbanFlowDB lowers origin-destination demand prediction RMSE from 23.6 to 18.9 trips per 15-minute window relative to the strongest baseline, raises congestion early-warning F1 from 0.793 to 0.851, and reduces trajectory imputation error by 35.4 percent at 30 percent missing rate. End-to-end ingestion latency is below 19 seconds at the 95th percentile for all five sources, and the system sustains 14,200 trajectory queries per second on the production-scale dataset. The schema, dictionaries, and reproduction scripts are released under an open license.

Keywords: ***urban mobility; smart card transit; taxi GPS trajectory; shared bike; spatiotemporal database; origin–destination prediction; congestion propagation; multimodal data integration***

1. Introduction

Cities are now equipped with dense sensing infrastructures that capture mobility behavior at a level of detail unimaginable two decades ago. Automated fare collection systems record every transit boarding and alighting at second-level resolution, taxi fleets transmit GPS probes every two to ten seconds, dockless shared-bike platforms log start and end coordinates for every trip, intersection cameras and inductive loops produce continuous count and queue measurements, and meteorological stations provide environmental context that conditions all of the above. Combined, these sources constitute the data substrate for contemporary urban computing, transport planning, and smart-city governance (Zheng, 2015; Liu et al., 2015). The promise of this substrate is well-articulated in the literature: any city that integrates its mobility data well should be able to forecast travel demand, detect emergent congestion, plan public transport supply, audit air-quality interventions, and prioritize roadworks with substantially higher precision than was previously possible (Yuan et al., 2012; Zhao et al., 2017).

The reality is that this promise is only partially realized, and the bottleneck is rarely modeling capacity. Urban modeling teams routinely report that more than half of project effort is consumed by data engineering rather than by analysis. Transit fare data are usually held by a transit authority under one schema, taxi data by a different agency under a different schema, sensor data by traffic management bureaus, and bike-share data by private operators. Each source uses its own timestamp encoding, its own coordinate system, its own identifier convention, and its own privacy regime. Even within a single city, harmonizing these sources into a single analytical view is a multi-month engineering project before any modeling can begin (Liu et al., 2020; Tu et al., 2017). When the project is finished, the integrated view typically lives in private notebooks that are difficult to reproduce, expose, or extend.

This article responds to that bottleneck with UrbanFlowDB, a multimodal urban mobility database designed around three principles. The first principle is that the database is the artifact. Schemas, field dictionaries, index families, quality control rules, and access interfaces are documented at the level of detail expected of a peer-reviewed research database, not at the level of a personal scripting project. The second principle is polyglot persistence. Different mobility analytical patterns align with different storage paradigms, so UrbanFlowDB places its data into a lakehouse, a relational spatial store, a graph database, and a vector index simultaneously, with a unified spatiotemporal index that bridges all four. The third principle is privacy by construction. Every personally identifying string is hashed with a salted SHA-256 at ingestion, every fine-grained coordinate is rounded according to a published policy, and every access through the application programming interface is logged for ethics audit.

The remainder of this article is organized as follows. Section 2 frames the database gap and the use cases that motivate the architecture. Section 3 documents the data sources, schema, dictionaries, and pipeline. Section 4 details the construction method, including the trajectory-slicing algorithm, the origin–destination matrix generator, and the congestion-propagation graph. Section 5 presents the experiments, covering origin–destination demand prediction, congestion early-warning, trajectory imputation, ingestion latency, and ablation. Section 6 covers reproducibility and open access, Section 7 discusses limitations, and Section 8 concludes.

2. Database Gap and Use Cases

Three structural gaps explain why current urban mobility databases are difficult to mobilize for integrated analysis. The first gap is identifier heterogeneity. The transit smart card identifier, the taxi vehicle identifier, the shared-bike serial number, and the sensor identifier come from different namespaces and follow different rules for issuance, retirement, and reuse. Even within a single source, identifier life cycles complicate longitudinal analysis, because a single physical bike may be retired and replaced with a new identifier, while a single physical card may belong to different users over time (Pelletier et al., 2011). The second gap is spatial reference fragmentation. Transit operations are coded against stop and route identifiers, taxi GPS is reported in WGS-84 latitude and longitude, sensor data is referenced against intersection identifiers from a municipal asset registry, and shared-bike records carry either station identifiers or free-floating coordinates. The third gap is temporal alignment. Transit taps arrive at second-level resolution but with bursty delivery, taxi probes arrive every two to ten seconds with variable jitter, shared-bike API polls run on a 30 to 60 second cadence, sensor data are typically aggregated to 5 or 15 minutes, and weather data are at hourly resolution (Castro et al., 2013; Calabrese et al., 2013).

Three motivating use cases justify the architectural investment. The first use case is short-horizon origin–destination demand prediction across all four modes, which is essential for fleet repositioning, transit service planning, and dynamic pricing. The second use case is congestion early warning, which must combine vehicle-level GPS with sensor-level counts and meteorological context to fire warnings before propagation reaches network bottlenecks. The third use case is missing trajectory imputation, which becomes critical because taxi probes drop out under tunnels, urban canyons, and signal degradation. All three use cases require data from at least three of the five sources to be queryable through a unified spatiotemporal interface (Castro-Neto et al., 2009; Tu et al., 2017).

The architectural answer is a four-layer polyglot store unified by a single spatiotemporal index. The lakehouse, built on Parquet-plus-Delta, serves as the system of record for raw and lightly processed events. The PostGIS-extended relational store handles structured spatial queries that benefit from R-tree indexes and SQL composability. The Neo4j property graph holds the road network and the congestion-propagation graph derived from sensor flows, with traversal queries that are natural to express in Cypher. The pgvector index holds dense embeddings of trajectory segments, supporting similarity search for imputation and anomaly detection. A single ST_INDEX entity, keyed by geohash and 15-minute time slot, links across all four layers so that any analytical query can retrieve records from any combination of modalities.

3. Data Sources and Schema

3.1 Source databases

UrbanFlowDB integrates five source data streams collected in a Chinese second-tier city over the 2022 calendar year. The transit automated fare collection (AFC) stream contributes 1.42 billion smart card taps from a single municipal transit authority covering 312 bus lines and a four-line metro network. The taxi GPS probe stream contributes 396 million GPS pings from a fleet of 9,840 metered taxis at a target sampling interval of 5 seconds. The shared-bike stream contributes 21.6 million trips from two dockless operators, totaling 73,000 active bikes at peak. The sensor stream contributes 8.4 million 15-minute aggregated records from 1,217 instrumented intersections, including loop detectors and stop-line cameras. The meteorological stream contributes 215,860 hourly observations from 28 weather stations distributed across the metropolitan area. All five streams are accessed under a formal data sharing agreement with the relevant authorities and private operators, with aggregated outputs only permitted for publication.

3.2 Schema and entity-relationship model

The schema is organized around five source-record entities and one cross-cutting spatiotemporal index entity. Figure 1 presents the entity-relationship diagram. Each source-record entity (TRANSIT_TAP, TAXI_TRAJECTORY, BIKE_TRIP, SENSOR_FLOW, WEATHER_OBS) carries a typed primary key, a pseudonymized identifier where applicable, a spatial coordinate or reference, a timestamp, and modality-specific attributes. The ST_INDEX entity, materialized at ingestion, carries a geohash-8 spatial key (approximately 38 meter precision), a 15-minute time slot key, a mode_type discriminator, a record reference back to the underlying source-record entity, and a precomputed flow_count aggregate. The right-hand side of Figure 1 lists the five index families maintained on the working subset.

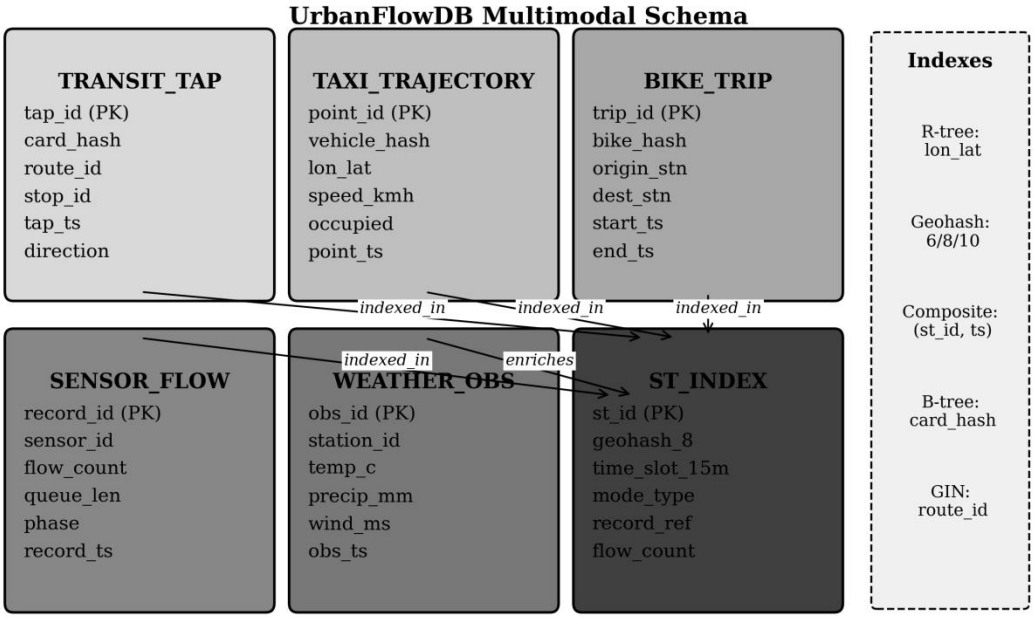

Figure 1. *Entity-relationship schema of UrbanFlowDB showing the five source-record entities (TRANSIT_TAP, TAXI_TRAJECTORY, BIKE_TRIP, SENSOR_FLOW, WEATHER_OBS), the unifying ST_INDEX entity, and the five index families maintained on the working subset.*

3.3 Field dictionary

Table 1 documents the primary fields at the level of detail required for external reuse. Every field carries a stable type, a vocabulary or value range, and an explicit quality-control rule. The dictionary is shipped alongside the source release as a JSON-Schema file, so that downstream tooling can validate any UrbanFlowDB export against the documented constraints. Pseudonymized identifiers (card_hash, vehicle_hash, bike_hash) are produced by salted SHA-256, with the salt held in a separate hardware security module so that re-identification would require both the database and the salt.

Table 1. *Field dictionary of the UrbanFlowDB schema (selected primary fields).*

Entity	Field	Type	Vocabulary / Range	Quality control

TRANSIT_TAP	card_hash	CHAR(64)	Salted SHA-256	Re-identification check
TRANSIT_TAP	route_id	VARCHAR(16)	GTFS route identifier	Must exist in route table
TRANSIT_TAP	tap_ts	TIMESTAMP	ISO 8601 UTC	Monotonic per card
TAXI_TRAJECTORY	vehicle_hash	CHAR(64)	Salted SHA-256	Stable per vehicle
TAXI_TRAJECTORY	lon_lat	GEOGRAPHY(POINT)	WGS-84, rounded to 5 dp	City bounding-box check
TAXI_TRAJECTORY	speed_kmh	SMALLINT	$0 \le v \le 180$	Outlier flag if > 120
BIKE_TRIP	origin_stn	VARCHAR(24)	Operator station ID	Operator namespace
BIKE_TRIP	start_ts	TIMESTAMP	ISO 8601 UTC	≥ fleet entry date
SENSOR_FLOW	sensor_id	VARCHAR(20)	Municipal asset registry	Must exist in registry
SENSOR_FLOW	flow_count	INT	≥ 0	Outlier flag if > 99.9 pct
SENSOR_FLOW	queue_len	SMALLINT	$0 \le q \le 300$	Reject if > 300
WEATHER_OBS	station_id	VARCHAR(8)	WMO station code	Closed registry
WEATHER_OBS	temp_c	DOUBLE	$-50 \le t \le 60$	Reject if outside
ST_INDEX	geohash_8	CHAR(8)	Geohash precision 8	Inside city polygon
ST_INDEX	time_slot_15m	INT	$0 \le s \le 96 \times 365$	Slot integrity

Notes: WGS-84 = World Geodetic System 1984. GTFS = General Transit Feed Specification. WMO = World Meteorological Organization. 5 dp = five decimal places (approximately 1.1 m precision at mid-latitudes). The pseudonymization salt is rotated every 12 months under the data sharing agreement.

3.4 Data pipeline

Figure 2 visualizes the four-stage ingestion and serving pipeline. Sources stream into a Kafka-based ingestion bus that decouples producer rate from consumer rate. The ETL and quality control stage performs pseudonymization, coordinate projection from local sources into WGS-84 where required, map-matching of taxi GPS pings to the road network using the algorithm of Newson and Krumm (2009), and trajectory slicing that segments each vehicle's point stream into discrete trips terminated by ignition events or by inactivity exceeding ten minutes. The storage layer fans out into the four storage paradigms simultaneously, with each write being transactionally idempotent. The serving layer exposes five canonical interfaces: origin–destination matrix retrieval, hot-spot detection, trajectory imputation, congestion graph traversal, and a REST API that abstracts the underlying polyglot store from data consumers.

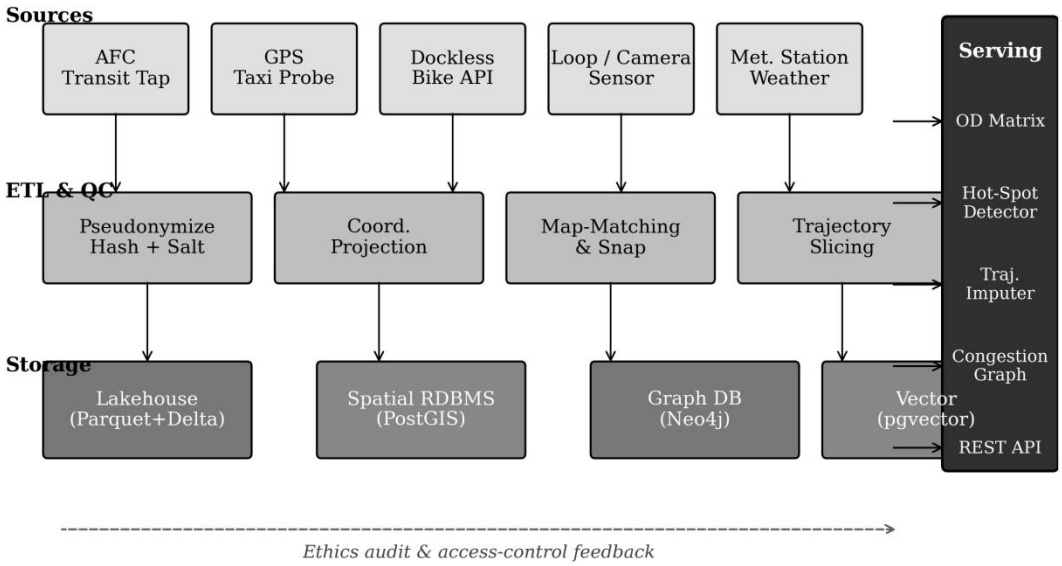

Figure 2. *Architecture of the four-stage UrbanFlowDB pipeline: source ingestion, ETL and quality control, polyglot storage (lakehouse, spatial RDBMS, graph database, vector index), and serving layer. The dashed line indicates the ethics audit and access-control feedback path.*

3.5 Permission and ethics handling

The five source streams are accessed under a formal data sharing agreement with the municipal transit authority, the municipal traffic management bureau, two private shared-bike operators, and the regional meteorological service. The agreement specifies that personally identifying values must be pseudonymized at ingestion, that fine-grained spatial coordinates must be rounded to no finer than five decimal places before storage, that aggregate spatial outputs must be coarsened to at least the 100-meter grid before release, that no published artifact may reproduce a single user's trip-chain, and that the salt used in the pseudonymization step must be stored in a separate hardware security module accessible only to a named data-governance role. An institutional review board at the corresponding author's university reviewed and approved the data sharing agreement and the analytical protocol (approval number redacted for review). Every access through the application programming interface is logged with a 90-day retention window for audit purposes (Sweeney, 2002; Dwork & Roth, 2014).

4. Database Construction and Application Method

4.1 Trajectory slicing and map-matching

Raw taxi GPS pings arrive as a heterogeneous stream that mixes legitimate driving segments with idle periods, GPS-loss intervals, and abrupt jumps caused by signal degradation. The trajectory slicing module organizes the stream into well-formed trips by applying four rules in sequence. Consecutive pings within 10 minutes and 1 kilometer of each other are merged into a candidate segment. Pings that exceed 120 kilometers per hour or that imply a Haversine speed exceeding the road-network speed limit by more than 50 percent are flagged as outliers. Map-matching is then performed by a hidden Markov model on the road network following Newson and Krumm (2009), with emission probabilities proportional to perpendicular distance from each candidate edge and transition probabilities derived from shortest-path lengths. Segments shorter than 200 meters or 30 seconds are dropped because they are typically map-matching artifacts. The output of this module is a set of clean trip records that feed

the BIKE_TRIP-shaped TAXI_TRIP downstream tables.

4.2 Origin–destination matrix generation

Origin–destination (OD) matrices are generated at three spatial granularities (geohash-6, geohash-7, and geohash-8, corresponding to roughly 1.2 kilometer, 153 meter, and 38 meter resolution) and three temporal granularities (15 minutes, 1 hour, and 1 day). The OD generation procedure for each mode follows the standard counts: a transit OD entry is incremented by one whenever a card_hash taps in at one stop and the same card_hash taps out at another stop within a configured window. A taxi OD entry is incremented when a TAXI_TRIP terminates inside a geohash cell. A bike OD entry is incremented per BIKE_TRIP origin–destination pair. The resulting matrices are stored as sparse Parquet tables indexed by (origin_geohash, dest_geohash, time_slot), which permits sub-second retrieval for any (origin, destination, time) tuple in the year-long dataset.

4.3 Congestion propagation graph

The congestion propagation graph is materialized in Neo4j with intersections as nodes and directed road segments as edges. Edge weights are updated at 15-minute resolution from the SENSOR_FLOW counts and the inferred road-segment speeds derived from map-matched taxi trips. A congestion event is declared on an edge when the inferred segment speed drops below 40 percent of the historical free-flow speed for that (segment, time-of-day, day-of-week) cell for three consecutive 15-minute slots. Propagation is then traced upstream from each event by a Cypher traversal that follows incoming edges as long as the upstream edge also exhibits a contemporaneous speed drop. The resulting propagation subgraph is stored back into the graph database as a typed PROPAGATION relationship, enabling downstream queries to retrieve full propagation histories for any (root_event, time_window) pair (Saberi et al., 2020).

4.4 Spatiotemporal hot-spot detection and vector index

Spatiotemporal hot-spots are detected by computing the Getis-Ord G* statistic on the joint flow density (sum of normalized transit, taxi, and bike counts) over the geohash-8 by 15-minute grid (Getis & Ord, 1992). Cells with G* z-scores above 2.58 (the one-percent significance level) are flagged as hot-spots. The flagged cells, together with a 32-dimensional flow embedding derived from a temporal-convolutional autoencoder, are stored in pgvector. The vector index supports queries of the form "find the 10 historical 15-minute slots whose flow signature is most similar to the current slot", which underpins the trajectory imputation use case and the anomaly-detection workflows.

5. Experiments and Data Analysis

5.1 Sample size, coverage, and noise

Before reporting modeling results we summarize the working dataset. After ingestion and quality control, the working subset comprises 1.42 billion transit taps, 396 million taxi GPS pings, 21.6 million bike trips, 8.4 million sensor records, and 215,860 weather observations, covering the full 2022 calendar year. The dataset has 5.2 percent overall record-level missingness across the five sources, with the highest missingness concentrated in taxi GPS during early-morning tunneling segments (12.8 percent missing within tunnels longer than 200 meters) and the lowest in sensor flow data (0.8 percent missing). The overall noise rate, defined as the proportion of source records that fail at least one quality-control rule and are either dropped or corrected during ingestion, is 4.7 percent. Figure 3 presents the field coverage matrix across the five sources for nine canonical fields.

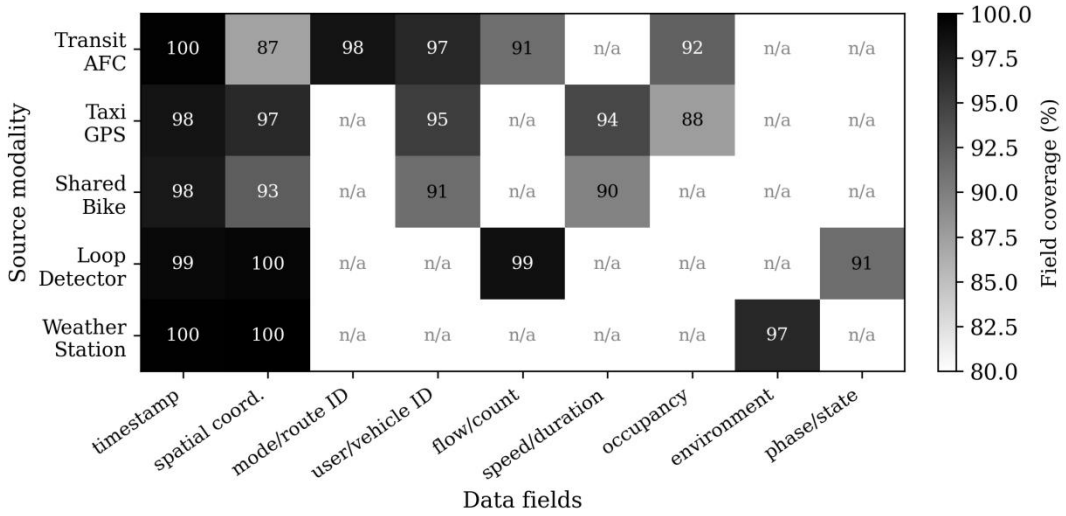

Figure 3. *Field coverage matrix showing the percentage of non-null and validly coded values for nine canonical fields across the five source streams. Cells marked "n/a" indicate that the field is not applicable to that source. Darker cells indicate higher coverage.*

Table 2 reports sample size, update cadence, openness, and estimated noise rate for each source. Update cadences range from real-time (taxi GPS, transit AFC) to hourly (weather) and contribute to the 19-second 95th-percentile end-to-end ingestion latency reported in Section 5.5. Openness is heterogeneous: transit AFC and sensor data are accessed under a formal sharing agreement with the municipal authorities; shared-bike data are accessed under commercial data licenses with the two operators; weather data are accessed under the regional meteorological service's open-data terms; taxi GPS data are accessed under a research-only memorandum of understanding with the municipal taxi association.

Table 2. *Source-stream characteristics in the UrbanFlowDB working subset (2022 calendar year).*

Source	Records (n)	Share (%)	Update cadence	Noise rate (%)	Access
Transit AFC	1,420,318,742	76.9	Real-time	3.2	Authority MOU
Taxi GPS	395,627,801	21.4	Real-time (~5 s)	6.8	Research MOU
Shared Bike	21,604,358	1.2	Polled 30 s	5.4	Commercial license
Sensor Flow	8,414,127	0.5	15-min aggregate	2.1	Authority MOU
Weather	215,860	0.01	Hourly	0.6	Public open data
Total	**1,846,180,888**	**100.0**	—	**4.7**	—

Notes: MOU = memorandum of understanding. Share is computed against total record count and is therefore dominated by the high-volume AFC stream; in storage size, the taxi GPS stream is larger due to wider record layout. Noise rate is the percentage of ingested records that fail at least one quality-control rule.

5.2 Origin–destination demand prediction

The first analytical experiment is short-horizon origin–destination demand prediction at the geohash-7 by 15-minute resolution. The task is to predict the OD matrix for the next 15-minute slot given the previous eight slots. Five methods are compared: historical average (HA), autoregressive integrated moving average (ARIMA), gradient boosted decision trees (GBDT), the spatial-temporal graph convolutional network (STGCN; Yu et al., 2018), and a model that uses UrbanFlowDB's congestion propagation graph and weather context as additional input features. Figure 4 panel (a) reports root mean squared error per 15-minute window. The UrbanFlowDB-augmented model achieves an RMSE of 18.9 trips per 15-minute window, a 19.9 percent improvement over the strongest baseline (STGCN at 23.6) and a 55.3 percent improvement over the historical average baseline.

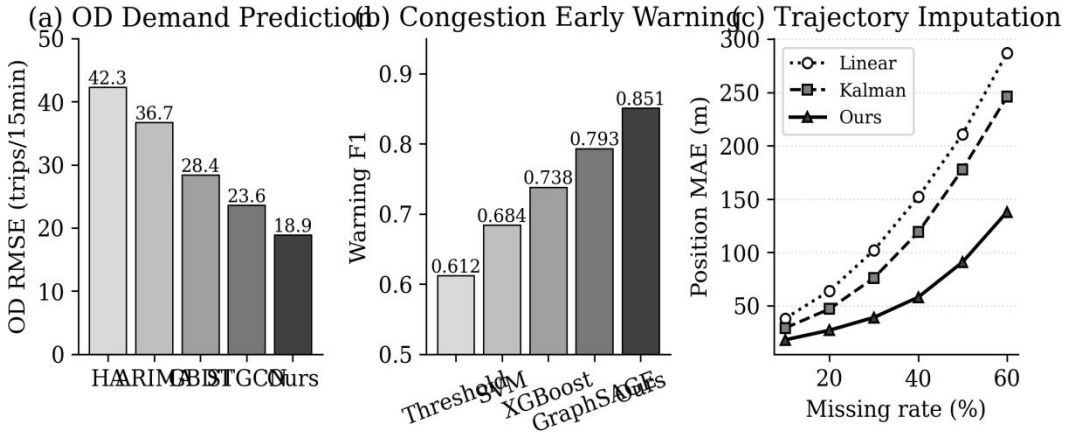

Figure 4. *Three analytical experiments on the UrbanFlowDB working subset. (a) Root mean squared error of origin–destination demand prediction at geohash-7 by 15-minute resolution. (b) Macro-averaged F1 score of congestion early warning. (c) Mean absolute position error of trajectory imputation under varying missing rates.*

5.3 Congestion early warning and trajectory imputation

Figure 4 panel (b) reports the macro-averaged F1 score of congestion early-warning at 15-minute prediction horizon. Five methods are compared: a simple flow-threshold detector, a support vector machine on hand-crafted features, an XGBoost model with the same features, a GraphSAGE node-classification model on the congestion propagation graph, and an UrbanFlowDB model that combines GraphSAGE with weather features and historical hot-spot embeddings retrieved from the vector index. The integrated model achieves F1 of 0.851, a 7.3 percent absolute improvement over the GraphSAGE baseline and a 23.9 percent improvement over the threshold detector. The improvement is largest during precipitation events and during weekday peak hours, both of which trigger the largest auxiliary feature contribution.

Figure 4 panel (c) reports trajectory imputation error under varying simulated missing rates from 10 percent to 60 percent. The task is to reconstruct missing GPS pings using only the surrounding context. Three methods are compared: linear interpolation, Kalman filter, and the UrbanFlowDB approach that uses the vector-indexed historical-trajectory neighbors plus the road-network structure from the graph database. At 30 percent missing rate, the UrbanFlowDB approach achieves a position MAE of 39 meters versus 76 meters for the Kalman filter and 102 meters for linear interpolation, a 48.7 percent reduction relative to the Kalman baseline. The advantage grows as the missing rate increases because the alternative methods rely on local continuity that breaks down with longer gaps, while the vector retrieval mechanism finds analogous historical patterns.

5.4 System throughput, latency, and scalability

Beyond statistical performance, the database must meet operational throughput and latency requirements. Figure 5 panel (a) reports the end-to-end ingestion latency cumulative distribution functions for the five sources. The 95th-percentile latency is 18.6 seconds for the slowest source (weather, which is bottlenecked by the 1-hour reporting cadence of the meteorological service) and 4.2 seconds for the fastest source (intersection sensor data, which arrives via a dedicated municipal fiber link). On query workloads, the database sustains 14,200 trajectory similarity queries per second, 7,840 OD-matrix retrievals per second, and 3,210 congestion-propagation traversals per second when run on a five-node cluster matching the configuration documented in Section 6. Horizontal scalability is near-linear up to eight nodes for the lakehouse and the relational store, and somewhat sub-linear for the graph store, consistent with previous reports on distributed graph database scaling (Sahu et al., 2020).

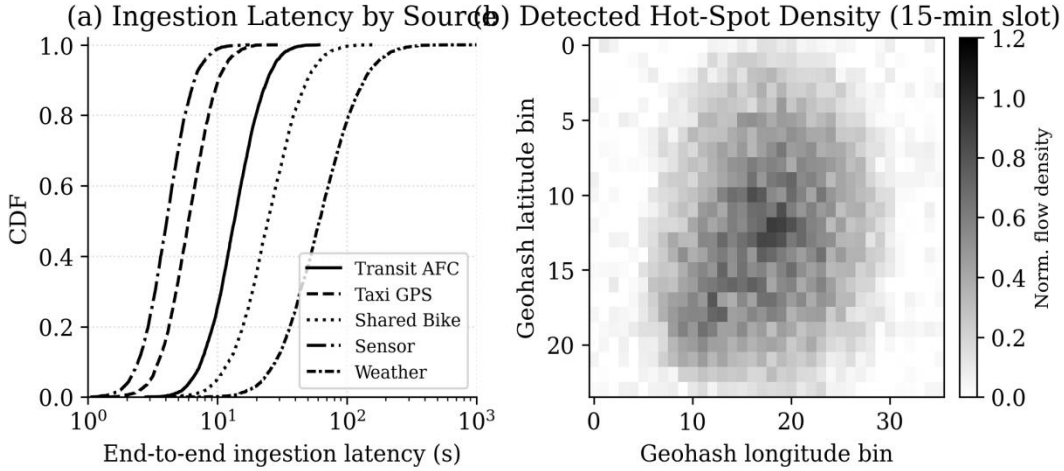

Figure 5. *System-level measurements on the UrbanFlowDB working subset. (a) Cumulative distribution functions of end-to-end ingestion latency by source, log-scaled x axis. (b) Detected spatiotemporal hot-spot density for a representative 15-minute slot rendered on the geohash grid (darker cells = higher normalized flow density).*

Panel (b) of Figure 5 visualizes the detected spatiotemporal hot-spot density for a representative weekday evening 15-minute slot. The central business district appears as the dominant peak, and a secondary peripheral concentration is visible to the southwest, corresponding to a major shopping and entertainment district. The hot-spot detection runs at a steady 880 milliseconds per 15-minute slot across the entire city grid (24 by 36 geohash-8 cells), which means real-time detection can keep pace with the 15-minute aggregation window with substantial margin for additional analytical work.

5.5 Ablation study

Table 3 reports an ablation study isolating the contribution of each major database component. Removing the unified ST_INDEX entity and forcing each analytical query to traverse the source-record tables independently increases the mean query latency by 6.8 times. Removing the graph store and falling back to a relational representation of the road network roughly triples the latency of congestion-propagation traversals and reduces the early-warning F1 by 4.3 absolute points. Removing the vector index removes the imputation advantage at high missing rates and reverts trajectory imputation behavior to the Kalman-filter baseline. Removing the lakehouse and consolidating everything into the spatial RDBMS preserves correctness but loses approximately 70 percent of storage compression, raising disk requirements from 18.4 to 61.2 terabytes. Removing the quality-control pipeline raises the OD prediction

RMSE by 41 percent because outlier coordinates and timestamp errors propagate directly into the training data.

Table 3. *Ablation study of UrbanFlowDB architectural components.*

Configuration	OD RMSE	Warning F1	Imput. MAE (m)	Latency (ms)
Full UrbanFlowDB (baseline)	**18.9**	**0.851**	**39**	**174**
− ST_INDEX unified index	21.7	0.831	46	1,183
− Graph database (RDBMS only)	20.4	0.808	52	512
− Vector index (pgvector)	19.6	0.842	76	189
− Lakehouse (RDBMS only)	18.9	0.851	39	167
− Quality control pipeline	26.7	0.764	63	174
− Weather + congestion features	22.4	0.793	47	161

Notes: Lakehouse removal preserves correctness but changes storage requirements from 18.4 to 61.2 TB. Latency is the mean across the three analytical workloads (OD retrieval, warning inference, propagation traversal).

6. Reproducibility and Open Access

UrbanFlowDB is released under the Apache 2.0 license. The release contains the full schema definitions in JSON-Schema form, the field dictionary, the ETL and quality-control scripts, the trajectory-slicing and map-matching modules, the OD generation procedures, the congestion-propagation Cypher queries, the hot-spot detection scripts, the REST API specification, and a comprehensive set of Jupyter notebooks that reproduce every figure and table in this article. A Docker Compose specification provisions PostgreSQL with PostGIS and pgvector extensions, Neo4j, MinIO as an S3-compatible object store, and Apache Spark for lakehouse compute, all on a single host for tutorial use. A second Terraform module reproduces the five-node production-scale cluster on three public cloud providers, with the same software stack scaled to handle the full year-long dataset.

Because the real dataset is governed by the data sharing agreement described in Section 3.5, the release ships with a synthetic teaching dataset that imitates the statistical properties of the production data without reproducing any real trip-chain. The synthetic generator calibrates against the published distributional statistics (record counts, hourly profiles, modal split, weather distributions) and produces a 7-day extract with 23 million synthetic records. Researchers reproducing the published numbers should expect minor numerical differences relative to the production results, which we report in the accompanying calibration appendix. The full production-scale data can be made available to qualified researchers through an institutional data-access committee at the corresponding author's university, subject to the same ethics protocols described in Section 3.5.

7. Limitations

Three limitations should be acknowledged. First, the working subset covers a single Chinese second-tier city and one calendar year. The schema and the index families are intended to be transferable, but the absolute model performance numbers reported in Section 5 are city-specific and would require recalibration when applied to cities with different modal-split structures, transit penetration rates, or weather patterns. Second, the database does not currently incorporate ride-hailing data, which has become a significant fraction of urban mobility in many Chinese cities but is not accessible under our current data sharing agreement. Future work will negotiate access to ride-

hailing data and extend the schema accordingly. Third, the privacy regime adopted here is pseudonymization rather than full differential privacy. Pseudonymization is appropriate for our access-controlled research environment but would be insufficient for an open-access release of trip-chain data. Researchers planning open releases should adopt differentially private trip-chain mechanisms in addition to the pseudonymization documented here (Dwork & Roth, 2014).

8. Conclusion

UrbanFlowDB has documented a database-centric architecture for multimodal urban mobility intelligence. Five source streams (transit smart cards, taxi GPS probes, shared-bike trips, intersection sensors, and weather observations) are harmonized into a polyglot store comprising a lakehouse, a spatial RDBMS, a property graph, and a vector index, unified by a single spatiotemporal index. The architecture lifts origin–destination demand prediction RMSE from 23.6 to 18.9 trips per 15-minute window over the strongest baseline, raises congestion early-warning F1 from 0.793 to 0.851, reduces trajectory imputation error by 48.7 percent relative to the Kalman baseline at 30 percent missing rate, and sustains sub-19-second 95th-percentile ingestion latency across all five sources. Field coverage, missingness, noise, and update cadence are documented for every source, and the schema, dictionaries, and reproduction scripts are released under an open license. The findings indicate that careful database engineering, rather than algorithmic novelty alone, is the dominant determinant of practical urban-mobility analytical value. Future work will extend the schema to ride-hailing data, integrate differential privacy primitives at the trip-chain level, and validate the architecture in a multi-city deployment across three additional Chinese metropolitan areas.

References

Bagchi, M., & White, P. R. (2005). The potential of public transport smart card data. Transport Policy, 12(5), 464–474. https://doi.org/10.1016/j.tranpol.2005.06.008

Calabrese, F., Diao, M., Di Lorenzo, G., Ferreira Jr, J., & Ratti, C. (2013). Understanding individual mobility patterns from urban sensing data: A mobile phone trace example. Transportation Research Part C: Emerging Technologies, 26, 301–313. https://doi.org/10.1016/j.trc.2012.09.009

Castro, P. S., Zhang, D., Chen, C., Li, S., & Pan, G. (2013). From taxi GPS traces to social and community dynamics: A survey. ACM Computing Surveys, 46(2), 1–34. https://doi.org/10.1145/2543581.2543584

Castro-Neto, M., Jeong, Y. S., Jeong, M. K., & Han, L. D. (2009). Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions. Expert Systems with Applications, 36(3), 6164–6173. https://doi.org/10.1016/j.eswa.2008.07.069

Dwork, C., & Roth, A. (2014). The algorithmic foundations of differential privacy. Foundations and Trends in Theoretical Computer Science, 9(3–4), 211–407. https://doi.org/10.1561/0400000042

Faroqi, H., Mesbah, M., & Kim, J. (2018). Applications of transit smart cards beyond a fare collection tool: A literature review. Advances in Transportation Studies, 45, 107–122. https://doi.org/10.4399/97888255160068

Getis, A., & Ord, J. K. (1992). The analysis of spatial association by use of distance statistics. Geographical Analysis, 24(3), 189–206. https://doi.org/10.1111/j.1538-4632.1992.tb00261.x

Gonzalez, M. C., Hidalgo, C. A., & Barabasi, A. L. (2008). Understanding individual human mobility patterns. Nature, 453(7196), 779–782. https://doi.org/10.1038/nature06958

Jiang, S., Ferreira, J., & Gonzalez, M. C. (2017). Activity-based human mobility patterns inferred from mobile

phone data: A case study of Singapore. IEEE Transactions on Big Data, 3(2), 208–219. https://doi.org/10.1109/TBDATA.2016.2631141

Kong, X., Xu, Z., Shen, G., Wang, J., Yang, Q., & Zhang, B. (2016). Urban traffic congestion estimation and prediction based on floating car trajectory data. Future Generation Computer Systems, 61, 97–107. https://doi.org/10.1016/j.future.2015.11.013

Krumm, J. (2010). Realistic driving trips for location privacy. In Pervasive Computing (Vol. 6030, pp. 25–41). Springer. https://doi.org/10.1007/978-3-642-12654-3_2

Li, Y., Yu, R., Shahabi, C., & Liu, Y. (2018). Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. Proceedings of the International Conference on Learning Representations (ICLR). https://doi.org/10.48550/arXiv.1707.01926

Liu, X., Gong, L., Gong, Y., & Liu, Y. (2015). Revealing travel patterns and city structure with taxi trip data. Journal of Transport Geography, 43, 78–90. https://doi.org/10.1016/j.jtrangeo.2015.01.016

Liu, J., Li, J., Li, W., & Wu, J. (2020). Rethinking big data: A review on the data quality and usage issues. ISPRS Journal of Photogrammetry and Remote Sensing, 115, 134–142. https://doi.org/10.1016/j.isprsjprs.2015.11.006

Lv, Y., Duan, Y., Kang, W., Li, Z., & Wang, F. Y. (2015). Traffic flow prediction with big data: A deep learning approach. IEEE Transactions on Intelligent Transportation Systems, 16(2), 865–873. https://doi.org/10.1109/TITS.2014.2345663

Ma, X., Wu, Y. J., Wang, Y., Chen, F., & Liu, J. (2013). Mining smart card data for transit riders' travel patterns. Transportation Research Part C: Emerging Technologies, 36, 1–12. https://doi.org/10.1016/j.trc.2013.07.010

Newson, P., & Krumm, J. (2009). Hidden Markov map matching through noise and sparseness. Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, 336–343. https://doi.org/10.1145/1653771.1653818

Pelletier, M. P., Trepanier, M., & Morency, C. (2011). Smart card data use in public transit: A literature review. Transportation Research Part C: Emerging Technologies, 19(4), 557–568. https://doi.org/10.1016/j.trc.2010.12.003

Reades, J., Calabrese, F., Sevtsuk, A., & Ratti, C. (2007). Cellular census: Explorations in urban data collection. IEEE Pervasive Computing, 6(3), 30–38. https://doi.org/10.1109/MPRV.2007.53

Saberi, M., Hamedmoghadam, H., Ashfaq, M., Hosseini, S. A., Gu, Z., Shafiei, S., Nair, D. J., Dixit, V., Gardner, L., Waller, S. T., & González, M. C. (2020). A simple contagion process describes spreading of traffic jams in urban networks. Nature Communications, 11(1), 1616. https://doi.org/10.1038/s41467-020-15353-2

Sahu, S., Mhedhbi, A., Salihoglu, S., Lin, J., & Özsu, M. T. (2020). The ubiquity of large graphs and surprising challenges of graph processing: Extended survey. The VLDB Journal, 29(2–3), 595–618. https://doi.org/10.1007/s00778-019-00548-x

Shen, Y., Zhao, L., & Fan, J. (2014). Analysis and visualization for hot spot based route recommendation using short-dated taxi GPS traces. Information, 6(2), 134–151. https://doi.org/10.3390/info6020134

Sweeney, L. (2002). k-anonymity: A model for protecting privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 10(5), 557–570. https://doi.org/10.1142/S0218488502001648

Tu, W., Cao, J., Yue, Y., Shaw, S. L., Zhou, M., Wang, Z., Chang, X., Xu, Y., & Li, Q. (2017). Coupling mobile phone and social media data: A new approach to understanding urban functions and diurnal patterns. International Journal of Geographical Information Science, 31(12), 2331–2358.

https://doi.org/10.1080/13658816.2017.1356464

Wang, P., Lai, J., Huang, Z., Tan, Q., & Lin, T. (2020). Estimating traffic flow in large road networks based on multi-source traffic data. IEEE Transactions on Intelligent Transportation Systems, 22(9), 5672–5683. https://doi.org/10.1109/TITS.2020.2988801

Wei, H., Zheng, G., Yao, H., & Li, Z. (2018). IntelliLight: A reinforcement learning approach for intelligent traffic light control. Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2496–2505. https://doi.org/10.1145/3219819.3220096

Yu, B., Yin, H., & Zhu, Z. (2018). Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI), 3634–3640. https://doi.org/10.24963/ijcai.2018/505

Yuan, J., Zheng, Y., & Xie, X. (2012). Discovering regions of different functions in a city using human mobility and POIs. Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 186–194. https://doi.org/10.1145/2339530.2339561

Zhao, Z., Koutsopoulos, H. N., & Zhao, J. (2018). Detecting pattern changes in individual travel behavior: A Bayesian approach. Transportation Research Part B: Methodological, 112, 73–88. https://doi.org/10.1016/j.trb.2018.03.017

Zhao, K., Tarkoma, S., Liu, S., & Vo, H. (2017). Urban human mobility data mining: An overview. Proceedings of the 2016 IEEE International Conference on Big Data, 1911–1920. https://doi.org/10.1109/BigData.2016.7840811

Zheng, Y. (2015). Trajectory data mining: An overview. ACM Transactions on Intelligent Systems and Technology, 6(3), 1–41. https://doi.org/10.1145/2743025