Rule conformance and referential integrity in a transit data extract

A controlled GTFS fault-injection study

October 2026

Abstract

This study examines whether aggregate rule-conformance scores are sufficient to detect a failed relationship in a derived transit-data product. A deterministic projection of a frozen public MBTA GTFS feed selected 500 trips and 10,870 stop-time events. One referenced stop was removed from the consumer stop dimension while its two referencing events were retained. The resulting event fixture was evaluated under a lenient 11-check profile and a stricter 18-check profile. An independently implemented verifier assessed relational membership, key uniqueness and the distinction between source and consumer data. [E1–E3]

The lenient profile passed all checks. The strict profile failed three required-field checks, identifying six missing-attribute applications across the two affected events. Strict completeness conformance nevertheless rounded to 99.99%; validity and uniqueness conformance were 100.00%. The independent verifier detected two unresolved stop references and confirmed that the omitted stop existed in the frozen source. Repeated primary evaluations produced byte-identical outputs within each profile. [E1–E4]

These results demonstrate that a high aggregate score can coexist with a failed declared consumer contract. Interpretation therefore requires the evaluated rule set, application denominators, relationship coverage and decision gate. The experiment is limited to a non-representative extract and an intentionally introduced fault; it does not measure route-planning behavior or passenger impact.

Input events	Failed strict checks	Violating applications
10,870	3 of 18	6

Contents and notation

Section	Pages
1–2. Study design, source data and consumer model	3–4
3–4. Construction procedure and experimental controls	5–6
5–6. Complete primary evaluation results	7–8
7–8. Aggregate results and denominator reconstruction	9–10
9–10. Affected records and failure propagation	11–12
11–12. Independent verification and threats to validity	13–14
13–15. Decision semantics and operational implications	15–17
16–17. Limitations and further work	18–19
Appendices A–C. Rules, terminology and reproducibility	20–22

Evidence notation

Numerical findings and record-level observations are referenced to four classes of retained experimental evidence. Public documentation references are listed separately in Appendix C.

Key	Evidence
E1	Primary evaluation reports, check-level outputs, aggregate comparison and repeated-output records.
E2	Frozen public source, fixture manifest, derived consumer tables, event records and preregistered design.
E3	Independent verification results, reference-membership tests and source-row checks.
E4	Recorded reproduction and preservation checks.

Observed measurements are distinguished from calculations derived from those measurements. Illustrative decision policies and proposed follow-up studies are identified as analysis or future work rather than experimental outcomes.

1. Study design and data source

The General Transit Feed Specification (GTFS) defines a structured schedule-data exchange with references between tables, including stop-time references to stops.[1] The input was a frozen snapshot of the official MBTA static GTFS download, documented through the agency's developer material.[2][3]

Frozen source
Referenced stop exists

→

Bounded projection
500 trips · 10,870 events

→

Consumer-only loss
Omit one stop row

Question and preregistered expectation

Can aggregate conformance remain high while a critical relationship in a consumer view fails? The predicted pattern was that lenient event-field checks would pass, stricter joined-attribute checks would fail, and an independent relationship check would identify the missing target. [E1–E3]

Source snapshot attribute	Recorded value
Publisher	Massachusetts Bay Transportation Authority (MBTA)
Feed version	Fall 2026, 2026-10-02T16:00:50+00:00, version D
Recorded schedule range	25 September–12 December 2026
Selection strategy	First 500 trip rows in lexical trip_id order
Intervention	Remove stop_id 70506 only from the consumer stop dimension

Bounded table	Rows
agency	2
routes	4
trips	500
stop_times	10,870
stops	121
calendar	1
calendar_dates	0
feed_info	1

Before the omission, the selected events referenced 122 stops. The bounded stop dimension contains 121 afterward; all 10,870 event rows remain. These are deterministic selection choices, not a random or representative sample. [E2]

2. Consumer data model

The flattened fixture contains 10,870 event objects and 11 fields. The table summarizes the retained event objects. Its missing-value counts use JSON null, which is the representation of the deliberately failed stop join. [E2]

Field	Lineage	Observed JSON type	Nulls
arrival_time	Event or trip linkage	string	0
departure_time	Event or trip linkage	string	0
route_id	Event or trip linkage	string	0
route_short_name	Route-dimension join	string	0
service_id	Event or trip linkage	string	0
stop_id	Event or trip linkage	string	0
stop_lat	Stop-dimension join	null / string	2
stop_lon	Stop-dimension join	null / string	2
stop_name	Stop-dimension join	null / string	2
stop_sequence	Event or trip linkage	string	0
trip_id	Event or trip linkage	string	0

Preserve identity separately from attributes

The affected event rows still contain stop_id=70506. Presence of that identifier is therefore not evidence that the referenced stop row exists in the consumer dimension. The missing name and coordinates are downstream symptoms of the unresolved relationship, not three unrelated source omissions.

String storage does not imply invalid values

Identifiers, times and coordinate values in nonmissing records are stored as strings. The saved numeric-type checks accept the tested coordinate values. Their result concerns numeric compatibility under the evaluator’s rules, not a requirement that the original JSON token itself be a number.

> **What this schema does not supply**
>
> The extract does not establish geographic accuracy, stop accessibility correctness, semantic route validity, freshness compliance or a working routing service. Those require additional contracts and evidence.

3. Fixture construction and evaluation

Construction and evaluation sequence

- Read the eight core schedule tables from the unmodified frozen archive; parse CSV with UTF-8 BOM tolerance.
- Sort trip identifiers as literal strings; retain the first 500 trip rows and all referencing stop-time events.
- Select referenced routes, services and stops. Sort events by trip identifier and a stable numeric-or-string stop-sequence key.
- Require one selected stop row with identifier 70506, then omit it only from the consumer stop dimension.
- Join event rows to trip, route and incomplete stop dimensions. Preserve missing joined values as JSON null.
- Evaluate the identical event fixture under both fixed rule profiles, repeating each primary evaluation.
- Independently check key uniqueness, reference membership and joined-attribute loss; inspect the full frozen source separately.

What each profile measures

Check family	Lenient	Strict
Required source-derived identifiers	4 checks	4 checks
Stop sequence: required, integer, ≥0	3 checks	3 checks
Arrival/departure: required + time pattern	4 checks	4 checks
Duplicate rows and composite event uniqueness	Not checked	2 checks
Joined stop name/latitude/longitude required	Not checked	3 checks
Latitude/longitude numeric compatibility	Not checked	2 checks
Total checks / distinct scopes	11 / 7	18 / 12

> **Separate implementations, separate responsibilities**
>
> The primary strict profile detects missing joined attributes; it is not itself a full foreign-key validator. Direct relational target-existence checks belong to the separate verifier. The profiles express a bounded consumer contract, not every GTFS requirement.

Controls included fixed input and rule hashes, repeated primary execution, preservation of source values, and a verifier that did not import primary evaluator logic. The manipulated variable in the main comparison was rule coverage on the same damaged view—not a live before/after routing workflow. [E1–E4]

4. Experimental controls

Element	Fixed or varied	Reason
Frozen source archive	Fixed	Preserves the public source boundary and enables fingerprint checks.
Trip selection and event ordering	Fixed	Avoids changing the slice while comparing rule profiles.
Consumer stop omission	Fixed injected condition	Both primary profiles see the same deliberately damaged view.
Event bytes	Fixed	Opposite verdicts cannot be attributed to different event inputs.
Rule profile	Varied	The main comparison changes coverage from 11 to 18 checks.
Separate verifier logic	Different implementation	Tests references and expected outcomes without importing primary evaluator logic.

Causal scope

The source row exists; the consumer row is absent; two retained events reference it; all three joined stop attributes on those events are null. The primary failures and separate membership test agree with the preregistered loss. This is a coherent causal chain within the constructed fixture. [E1–E3]

Study exclusions

- It does not compare two production pipelines or demonstrate that one vendor is safer.
- It does not measure the entire schedule before and after a real outage.
- It does not estimate the population frequency of missing stops.
- It does not establish sensitivity or specificity across many fault classes.

> **Unmeasured counterfactuals**
>
> A repaired copy, an untouched consumer copy and additional omission locations would strengthen the causal and generalization tests. Their expected behavior must be specified before execution; this report does not claim those additional evaluations occurred.

5. Lenient evaluation

Every executed lenient check is listed below in the preserved evaluation order. None examines joined stop attributes or direct foreign-key target existence. Consequently, populated event identifiers and well-formed times can pass even when the consumer stop lookup is incomplete. [E1]

#	Check / target	Applications	Violations	Status
1	required / trip_id	10,870	0	passed
2	required / route_id	10,870	0	passed
3	required / service_id	10,870	0	passed
4	required / stop_id	10,870	0	passed
5	required / stop_sequence	10,870	0	passed
6	type / stop_sequence	10,870	0	passed
7	min / stop_sequence	10,870	0	passed
8	required / arrival_time	10,870	0	passed
9	regex / arrival_time	10,870	0	passed
10	required / departure_time	10,870	0	passed
11	regex / departure_time	10,870	0	passed

Coverage interpretation

Seven required-field checks contribute 76,090 completeness applications. Four validity checks contribute 43,480 applications. No uniqueness rule is evaluated. Both measured dimensions report 100.00%, while uniqueness is unmeasured rather than perfect.

> **Coverage limitation**
>
> The evaluator passed the checks it was asked to perform. A false-negative classification would require the missing relationship to be part of the profile's declared detection target. The stronger criticism is incomplete coverage for a consumer requiring resolvable stop attributes.

6. Strict evaluation

The strict profile adds duplicate-row and composite-key checks, three joined-attribute requirements and two coordinate-type checks. Only the three required joined-attribute checks fail. Direct foreign-key checks remain separate. [E1]

#	Check / target	Applications	Violations	Status
1	max_duplicate_rows / all event fields	10,870	0	passed
2	unique_together / trip_id + stop_sequence	10,870	0	passed
3	required / trip_id	10,870	0	passed
4	required / route_id	10,870	0	passed
5	required / service_id	10,870	0	passed
6	required / stop_id	10,870	0	passed
7	required / stop_sequence	10,870	0	passed
8	type / stop_sequence	10,870	0	passed
9	min / stop_sequence	10,870	0	passed
10	required / arrival_time	10,870	0	passed
11	regex / arrival_time	10,870	0	passed
12	required / departure_time	10,870	0	passed
13	regex / departure_time	10,870	0	passed
14	required / stop_name	10,870	2	failed
15	required / stop_lat	10,870	2	failed
16	type / stop_lat	10,868	0	passed
17	required / stop_lon	10,870	2	failed
18	type / stop_lon	10,868	0	passed

The latitude and longitude type checks each evaluate 10,868 present values; the two nulls in each column are not numeric-type failures. All 18 checks are evaluated at check level: a reduced application denominator is different from an entirely not-evaluated check.

7. Aggregate results

Measure	Lenient	Strict
Rule status	passed	failed
Process exit code	0	1
Passed / total checks	11 / 11	15 / 18
Failed checks	0	3
Not-evaluated checks	0	0
Distinct scopes	7	12
Evaluated rule applications	119,570	195,656
Violating rule applications	0	6

Dimension scores and their denominators

Profile / dimension	Applications	Violations	Score
Lenient / completeness	76,090	0	100.00%
Lenient / uniqueness	0	0	Not measured
Lenient / validity	43,480	0	100.00%
Strict / completeness	108,700	6	99.99%
Strict / uniqueness	21,740	0	100.00%
Strict / validity	65,216	0	100.00%

```
Conformance % = (evaluated applications − violating applications)
÷ evaluated applications × 100
```

Strict completeness: (108,700 − 6) ÷ 108,700 × 100 = 99.99448022079117%, displayed as **99.99%**. The denominator counts rule applications, not unique records, stops, or a percentage of the entire feed that is “correct.” [E1]

Required checks evaluate every row. Other column checks exclude missing values: null coordinates fail completeness but do not also fail numeric-type checks. A dimension with zero evaluated applications is unmeasured (null), not 100%. This explains lenient uniqueness and why strict validity can remain 100.00%.

Coverage increased 1.63636× by checks and 1.63633× by applications. Those descriptive ratios are not quality-improvement factors. Scores under different coverage contracts are not intrinsic quality rankings.

8. Score calculation and denominators

Dimension	Lenient construction	Strict construction
Completeness	7 required checks × 10,870 = 76,090	10 required checks × 10,870 = 108,700
Validity	4 present-value checks × 10,870 = 43,480	4 × 10,870 + 2 × 10,868 = 65,216
Uniqueness	No applications	2 dataset checks × 10,870 = 21,740
All dimensions	119,570 applications	195,656 applications

Completeness has six violating applications. Validity has none among values it evaluated. Uniqueness has none across its two dataset-level checks. Application-weighted aggregation therefore answers a different question from counting affected events or evaluating the critical stop relationship. [E1]

Rounding sensitivity

Case	Applications	Violations	Unrounded score	Display
Measured case	108,700	6	99.99448022%	99.99%
Illustration only: 10× denominator	1,087,000	6	99.99944802%	100.00%
Illustration only: 100× denominator	10,870,000	6	99.99994480%	100.00%

Only the first row is an observed experiment result. The other rows are pure arithmetic illustrations with six violations held fixed; no additional data or evaluation was produced. They show why rounding and denominator size can make a percentage look perfect without removing the counted violations.

Units of analysis

- **Record:** one event. Two distinct events are affected.
- **Check:** one constraint on a target. Three strict checks fail.
- **Application:** one evaluation under a check. Six required-field applications violate their rule.

> **Interpretation**
>
> The completeness denominator counts individual rule applications. It neither estimates record-level accuracy nor represents an operational-risk weighting.

9. Record-level findings

The public source identifies stop 70506 as Gilman Square, at latitude 42.387831 and longitude −71.096629, with parent station place-gilmn. That source row remained present. Only the bounded consumer stop table omitted it. [E2–E3]

Field	Frozen source stop	Joined consumer value
stop_id	70506	70506 retained on the event
stop_name	Gilman Square	null
stop_lat	42.387831	null
stop_lon	-71.096629	null

Exact affected event identifiers

Row*	Trip ID	Route	Sequence	Arrival	Stop ID
36	77655682	Green-B	7	06:38:00	70506
97	77655684	Green-C	7	06:17:00	70506

Both departure times equal their respective arrival times. The shared service identifier is LRV20264-hlb46016-Saturday-01. Identifiers and times remain strings.

Primary failed checks

Failed check	Evaluated	Violations	Rows*
required:stop_name	10,870	2	36, 97
required:stop_lat	10,870	2	36, 97
required:stop_lon	10,870	2	36, 97

Count failures without double-counting defects

There are two affected events, three failed checks and six failed field applications. The separate verifier’s two unresolved stop references describe the same injected loss; adding them to the six field violations would double-count overlapping consequences.

The two events are approximately 0.01840% of the 10,870-event extract. This small affected share helps explain the high aggregate conformance; it does not measure operational severity.

* Row references are one-based positions in this fixed event fixture, not permanent transit identifiers. The corresponding zero-based positions are 35 and 96.

10. Failure propagation

Source
Stop 70506 exists

→

Consumer dimension
One stop row omitted

→

Joined events
Two events × three nulls

Observation	Supported interpretation	Not established
Source stop row is present	The public source includes this referenced stop.	Full feed accuracy or certification.
Consumer stop row is absent	The intended controlled omission is visible.	A naturally occurring transport incident.
Event stop_id values remain populated	An identifier-presence rule can pass after target loss.	Successful reference resolution.
Name, latitude and longitude become null	The joined view carries consequences of the missing target.	Three separate source defects.
Two references fail membership	The separate check localizes a relationship failure.	Measured routing failure or passenger harm.

Source attribution

If the stop were also absent from the frozen source, the interpretation would change: the experiment could no longer attribute its absence solely to the declared consumer intervention. Fingerprinting the source and checking the source row guard against that attribution error. [E2–E3]

Fixture invariants

- The number of event rows remains 10,870 after the omission.
- The stop dimension loses one row, changing 122 referenced stops to 121 consumer stop rows.
- Only stop identifier 70506 appears in the missing-join set.
- The failed required-field checks point to the same two event positions.

> **Remediation boundary**
>
> Restoring the omitted dimension row is a plausible repair to test next. This report does not claim a repaired view was passed through a live consumer or that remediation restored service.

11. Independent verification

The verifier uses separate standard-library CSV/JSON parsing, set membership and counting logic. It does not import the primary evaluator. “Independent” here means independently implemented logic, not a second organization or an external human review. [E3]

Checks on the bounded consumer tables

Relationship	Evaluated	Violations
routes → agency	4	0
trips → routes	500	0
trips → service	500	0
stop_times → trips	10,870	0
stop_times → stops	10,870	2

The verifier also checks uniqueness of agency, route, trip and stop keys and the trip-plus-stop-sequence key. No duplicate groups were found in those checks. The joined stop-dimension check identifies the same two affected events. [E3]

Checks on the frozen full source

Relationship	Evaluated references	Violations
routes → agency	402	0
trips → routes	122,478	0
trips → services	122,478	0
stop_times → trips	3,155,220	0
stop_times → stops	3,155,220	0

Reproducibility evidence

- Original evaluations on 4 October 2026 were repeated with byte-identical outputs within each profile.
- A recorded reproduction on 9 October 2026 evaluated each profile twice, matching the historical report bytes; the separate verifier also passed its expected-result checks.
- The retained records support source-fingerprint checks, repeated-output byte equality, affected-record inspection and bounded reference-membership checks.

> **Verification outcome**
>
> The verifier passes because it reproduces the expected injected failure and confirms the source boundary. Its consumer-contract disposition is “not conformant.” The strict evaluator’s exit code 1 is the expected rule-failure result, not evidence of a broken executable.

12. Threats to validity

Risk to the conclusion	Existing safeguard	Residual limitation
Changed input between profiles	Fixed fixture and recorded input hashes; repeated reports.	A shared construction error could affect both profiles.
Primary evaluator hides the relationship failure	Separate reference membership and joined-attribute checks.	The verifier still reflects the declared contract.
Source defect misattributed to consumer loss	Source-stop presence and full-source reference checks.	Untested GTFS constraints remain outside the evidence.
Nondeterministic report output	Repeated primary report bytes agree within each profile.	Cross-environment byte equivalence is not established.
Unexpected additional bounded fault	Other checked references and key counts have no violations.	Untested semantic defects may still exist.
Internal agreement mistaken for independent review	Implementation independence is stated explicitly.	No external organization or domain-owner validation.

Separate verifier mechanics

The verifier constructs identifier sets from bounded tables, checks references against those sets, counts duplicate keys and compares the missing-join set with the declared loss. It also inspects the frozen full source and reads back repeated primary outputs. Key/reference preparation trims surrounding whitespace under its own parsing logic; this is not a measured study of normalization variants. [E3]

> **Expected-failure testing**
>
> A successful verification process means the saved observations match the expected fault pattern and no listed verification problem was reported. It does not mean the consumer fixture is acceptable for use. In this experiment, successful verification accompanies a “not conformant” consumer gate.

The check ledgers and dimensional totals are consistent with the retained primary outputs. The verification scope covers those outputs and the specified reference tests; it does not establish behavior in another execution environment.

13. Decision semantics

Measurement, contract and fitness

Analytical layer	Result	Interpretation
Primary rule evaluation	Lenient: passed. Strict: three checks failed.	The verdict depends on the declared coverage contract.
Consumer referential contract	Two unresolved stop references.	The consumer view is not conformant with the tested stop-resolution requirement.
Fitness for the intended application	Not established.	No representative route-planning or live consumer evaluation was performed.

Rule conformance is an observation about specified checks. Admission is a policy decision about whether a candidate satisfies the requirements of a particular consumer. Fitness for purpose requires evidence that the admitted data is suitable for that application. The experiment demonstrates disagreement between these layers without requiring any measured score to be arithmetically incorrect.

Information required for an admission decision

Information	Purpose
Rule contract	Identifies which constraints produced the reported verdicts.
Coverage and exclusions	Shows whether critical fields, joins and relationships were actually tested.
Decision policy	Defines thresholds, critical-rule vetoes and exception conditions.
Consumer evidence	Connects structural observations to representative application behavior.

For a consumer requiring populated stop attributes and resolvable stop references, the injected loss is a contract failure. A different consumer may apply another policy; such a policy would require an explicit justification rather than an inference from the aggregate percentage.

> **Implication**
>
> The result supports reporting critical relationship failures alongside aggregate measurements. It does not establish a universal zero-violation requirement or a validated operational acceptance threshold.

14. Alternative decision policies

The following gate outcomes are analytical read-backs of the saved measurements. They are not additional evaluator executions or validated consumer policies. They show why the gate must be named before a score is used as approval.

Illustrative gate	Read-back outcome	Reason
All measured dimension scores ≥99%	Pass	99.99%, 100.00% and 100.00% meet the threshold.
No failed primary checks	Fail	Three strict checks failed.
Zero unresolved stop references	Fail	Two event references cannot resolve in the consumer dimension.
Representative consumer-fitness evidence available	Not established	No live consumer or representative journey evaluation was performed.

A filled evidence card for this fixture

Card part	Current entry
Measurement	Fixed 18-check strict profile; six violating required-field applications.
Coverage	10,870 events; 12 primary scopes; selected core references checked separately.
Named contract gate	Required stop relationships and attributes are not conformant.
Fitness evidence	Absent for a real route-planning decision; diagnostic-only status is appropriate.
Exception authority	No operational waiver or accountable consumer approval was supplied.

> **Policy scope**
>
> The experiment does not validate a 99% threshold, mandate zero violations for every dataset or define acceptable operational risk. A consumer must identify its decision, consequences and critical requirements before selecting a gate.

The four-part card makes uncertainty visible rather than forcing one score to represent conformance, completeness of testing and permission to deploy at once.

15. Operational implications

The following admission procedure is an operational recommendation derived from the observed failure mechanism. Its implementation and efficiency have not been evaluated.

Step	Decision owner should require	Record retained
1. Freeze the candidate	Identify the source snapshot and derived consumer version.	Dataset identity and integrity fingerprints.
2. Declare the intended use	Name the consuming decision and its critical dependencies.	Consumer contract and rationale.
3. Execute checks	Run versioned rules; distinguish rule failures from execution errors.	Per-check status, denominators and evidence.
4. Inspect coverage	Expose untested fields, relationships and failure modes.	Coverage and exclusion statement.
5. Apply the gate	Make critical vetoes and exception conditions explicit.	Pass, hold, fail or inconclusive disposition.
6. Test fitness	Use representative consumer cases and domain review.	Behavioral outcomes and review decision.

Escalation when a critical relationship fails

Hold the affected candidate, preserve the evidence, locate the missing target boundary and determine whether the fault belongs to source acquisition, projection, transport or join logic. Re-evaluate a separate repaired candidate only after the proposed repair is explicit; do not silently edit frozen evidence.

Exception handling

If a consumer can legitimately tolerate a missing relationship, document why, the affected scope, the responsible decision owner and the expiry or recheck condition. An exception changes an admission policy; it does not erase a measured violation or prove the data became intrinsically correct.

> **Unmeasured operational outcomes**
>
> No acceptance workflow, exception process, monitoring cost, remediation time or business benefit was evaluated. Those are deployment questions for a later study, not outcomes implied by this report.

16. Limitations

Limitation	Effect on interpretation
Deterministic lexical sampling	The first 500 trip identifiers are not a probability sample of the complete schedule.
Single controlled fault	One deliberately omitted stop establishes a mechanism, not its prevalence across feeds or fault classes.
No application-level outcome	No route planner, representative journey, rider outcome or service impact was measured.
Partial GTFS coverage	Selected core tables and relationships were checked; complete specification compliance was not assessed.
Declared consumer rules	The stricter profile is a bounded contract, not a universal definition of data quality.
Implementation-level independence	The verifier uses separate logic but is not an independent organizational or domain-owner review.
Frozen source snapshot	The findings do not describe the current contents of the mutable public download endpoint.

Internal and external validity

The retained source row, controlled consumer omission and matching affected-event references support the proposed causal explanation within this fixture. Generalization remains limited because the study does not vary selection strategy, fault location, consumer implementation or operational context.

Conclusion

The same consumer event fixture passed a lenient profile and failed a stricter profile while retaining 99.99% strict completeness conformance. Independent membership checks localized the failure to two consumer stop references and confirmed the referenced stop in the original source. The evidence demonstrates why conformance scores must be interpreted with coverage and decision semantics; it does not establish a defect in the MBTA source or an observed operational harm.

17. Further work

Further work should test whether the observed structural failure changes a declared consumer outcome and whether the mechanism persists across datasets and implementations.

Study extension	Design requirement	Outcome to measure
Untouched, damaged and repaired controls	Fixed candidate copies and one declared mutation per comparison.	Whether the fault produces the predicted change and whether restoration reverses it.
Representative journey cases	Predeclared origins, destinations, service dates and expected results.	Observed consumer behavior under the stated decision rule.
Independent consumer implementation	Separate implementation with identical scenario inputs.	Agreement or disagreement at the application boundary.
Multiple feeds and fault classes	Predeclared source selection, omission locations and analysis plan.	Generality of the failure mechanism and limits of detection.
Domain review	Explicit review questions and acceptance criteria.	Whether the tested constraints support the intended application.

Evaluation criteria

- Preserve immutable source snapshots and record each derived-candidate change.
- Demonstrate the failed relationship directly rather than inferring it solely from an aggregate score.
- Compare observed consumer behavior against a declared reference outcome.
- Retain negative findings, unexpected failures and implementation disagreement.
- Separate structural conformance, application behavior and operational impact in the analysis.

A repaired candidate and broader comparisons are proposed follow-up conditions, not completed evaluations. Claims about remediation effectiveness or production suitability require those additional measurements.

Appendix A. Rule definitions

The following lists the two fixed rule definitions. Both profiles use the same event fixture. These rules are a consumer-view checkpoint, not a full GTFS validator. [E1–E2]

Lenient · 11 checks

```
dataset: {}
columns:
  trip_id:
    required: true
  route_id:
    required: true
  service_id:
    required: true
  stop_id:
    required: true
  stop_sequence:
    required: true
    type: integer
    min: 0
  arrival_time:
    required: true
    regex: '^(?:[0-9]{2}|[0-9]{3}):[0-5][0-9]:
[0-5][0-9]$'
  departure_time:
    required: true
    regex: '^(?:[0-9]{2}|[0-9]{3}):[0-5][0-9]:
[0-5][0-9]$'
```

Strict · 18 checks

```
dataset:
  max_duplicate_rows: 0
  unique_together:
    - [trip_id, stop_sequence]
columns:
  trip_id:
    required: true
  route_id:
    required: true
  service_id:
    required: true
  stop_id:
    required: true
  stop_sequence:
    required: true
    type: integer
    min: 0
  arrival_time:
    required: true
    regex: '^(?:[0-9]{2}|[0-9]{3}):[0-5][0-9]:
[0-5][0-9]$'
  departure_time:
    required: true
    regex: '^(?:[0-9]{2}|[0-9]{3}):[0-5][0-9]:
[0-5][0-9]$'
  stop_name:
    required: true
  stop_lat:
    required: true
    type: number
  stop_lon:
    required: true
    type: number
```

Semantics that affect interpretation

- required checks evaluate all rows; type and pattern checks operate on nonmissing values.
- The time pattern allows two- or three-digit hours and constrains minutes and seconds to 00–59.
- Composite uniqueness covers the pair (trip_id, stop_sequence); whole-row duplicate detection is a separate check.
- Required stop attributes detect failed joins in this fixture. Direct foreign-key membership remains a separate-verifier responsibility.
- A dimension with no evaluated applications is reported as null rather than a perfect score.

Appendix B. Terminology

Term	Definition
Rule conformance	The outcome of specified constraints evaluated against a fixed dataset.
Coverage	The records, fields, relationships and failure modes included in the evaluation.
Evaluated application	One unit counted under an executed check; its denominator follows the check's semantics.
Not evaluated	No evaluation units for a check or dimension; distinct from a passing result.
Consumer contract	Requirements declared for a derived data product and its intended use.
Critical gate	An admission policy capable of withholding approval despite favorable aggregate measurements.
Fitness for purpose	Evidence that the data and its controls are suitable for the named application.
Diagnostic evidence	Measurement or controlled demonstration without sufficient application-level acceptance evidence.
Independent implementation	Separate verification logic, which does not by itself establish organizational independence.

Distinguishing failure counts

A missing dimension row can cause multiple event references and multiple attribute checks to fail. In this study, one omitted stop produces two affected events, three failed checks and six violating applications. These counts describe different units and are not additive measures of independent defects.

Distinguishing correctness claims

A reported percentage is a property of its rule applications and aggregation method. It is not, without additional evidence, an estimate of complete-source accuracy, the share of journeys that work, or the probability of an operational failure.

Appendix C. Reproducibility and references

Reproduction contract

Use the frozen snapshot identified below, not a fresh download. Rebuild the deterministic projection, omit stop 70506 only from the consumer stop table, retain all event references, and apply the exact rules in Appendix A. Run each profile twice. Independently recompute key uniqueness, reference membership and source-stop presence.

Expected observations: 10,870 events; 121 consumer stops after omission; 11/11 lenient checks pass; strict has 15 pass and 3 fail with 6 violations; strict completeness 99.99%; two unresolved stop references; the omitted stop present in the source. Compare output bytes within the same evaluator environment.

The experiment used a deterministic Python rule evaluator and a separate standard-library CSV/JSON verifier. A complete dependency lock is absent from the retained evidence package, so byte-identical cross-environment reproduction is not established. Reproduction should preserve the frozen inputs and use separate output destinations.

Dataset fingerprints (SHA-256)

Frozen official source archive
`173a28225a8ae27cb069693c839ce2b7abe9240f7a840adefa49a160641593ea`

Consumer event fixture
`a6dcdf22b2a4b80d9006a5ea785189d284cb26064000a94347766c706ff61212`

Bounded consumer tables
`e8fcf8ba6778205b96065a2eb01aa255e7c445c7b2e0b6a76cf9a85c4f38ee26`

Evidence records

E1 Saved primary results, comparison and repeated-output records. **E2** Frozen source, fixture manifest, consumer tables/events and preregistered design. **E3** Separate verifier and full-source relationship results. **E4** Recorded reproduction and preservation checks.

Public references

[1] GTFS Schedule Reference
https://gtfs.org/documentation/schedule/reference

[2] MBTA GTFS developer documentation
https://www.mbta.com/developers/gtfs

[3] Official MBTA static GTFS feed (mutable download endpoint)
https://cdn.mbta.com/MBTA_GTFS.zip

Public references provide specification and source context; they do not independently substantiate this study's measurements. Numerical findings come from the preserved experiment evidence. The specification context corresponds to the documentation retained with the experiment.