A Provenance-Aware AI Framework for Detecting and Localizing Silent Data Integrity Failures in Enterprise Data Pipelines

Sanket Shah
Independent Researcher
United States
svshah26@gmail.com
ORCID: 0009-0005-6359-7988

Abstract—Silent data-integrity failures in enterprise pipelines can preserve schema validity and successful job completion while corrupting record counts, values, joins, aggregates, temporal consistency, or target state. This paper presents PAIR, a provenance-aware AI framework for detecting and localizing such failures by reconciling stage-level signatures across a multi-stage data pipeline. PAIR combines deterministic first-divergence analysis with a provenance-relative machine-learning ranker and a risk-aware selective policy. A benchmark of 36 injected failure scenarios was evaluated in Apache Spark. Under clean references, deterministic localization identified all 36 scenario origins. A 216-run ambiguity benchmark then introduced reference noise, stale references, and missing checkpoints. The deterministic baseline achieved 43.06% localization accuracy, whereas the final reference-aware hybrid policy achieved 75.00% overall recommendation accuracy and 91.90% within one pipeline stage. Its selective policy automatically localized 72 of 216 cases with 100% observed accuracy while routing the remaining cases to review. Ablation showed that removing relative provenance caused the largest accuracy loss, from 75.00% to 54.63%. Scalability experiments retained exact localization through one million rows. The results support combining deterministic reconciliation, provenance-relative AI ranking, and explicit abstention rather than applying unconstrained automation to ambiguous pipeline failures.

Keywords—data pipelines, data integrity, provenance, reconciliation, anomaly localization, machine learning, selective automation

I. Introduction

Modern enterprise analytics depends on multi-stage data pipelines that ingest, filter, normalize, join, deduplicate, aggregate, and materialize data for downstream applications. Operational monitoring commonly emphasizes whether jobs completed, whether infrastructure remained available, and whether expected outputs were produced. These signals are necessary but not sufficient. A pipeline may report successful execution even when the resulting data are incomplete, duplicated, semantically corrupted, incorrectly filtered, or aggregated under the wrong grouping logic. Such silent integrity failures are especially difficult because the absence of an execution exception can delay detection until downstream reports, applications, or users expose the defect.

Existing data-quality systems provide valuable mechanisms for constraint checking, anomaly detection, and validation at scale [1], [2]. Provenance research, meanwhile, establishes the importance of derivation history for understanding where data came from and how it was produced [3], [4]. However, an operational gap remains between detecting that two pipeline states disagree and determining where the first meaningful divergence occurred when observations are incomplete or the reference itself may be unreliable. In these conditions, blindly trusting either a deterministic comparison or an AI prediction can be unsafe.

PAIR addresses this gap as a hybrid localization framework. Its core principle is to use deterministic first-divergence reasoning when checkpoint evidence is trustworthy and sufficiently observable, and to invoke a provenance-relative machine-learning ranker only for ambiguity classes in which reference divergence makes deterministic reasoning unreliable. A risk-aware policy then decides whether the recommendation can be automatically localized or should be routed for human review.

This paper makes four contributions. First, it defines a benchmark of silent integrity failures across nine failure families and multiple pipeline stages. Second, it develops deterministic checkpoint signatures that combine cardinality, completeness, aggregate, fingerprint, distribution, and provenance evidence for first-divergence localization. Third, it introduces a provenance-relative stage-ranking and reference-aware fusion strategy for ambiguity conditions. Fourth, it evaluates a selective automation policy with statistical confidence intervals, ablation analysis, seed reproducibility, and scalability tests through one million rows.

II. Related work and research gap

A. Data Quality Validation

Large-scale data-quality systems have demonstrated that declarative constraints and aggregate statistics can detect missing values, duplicates, distribution anomalies, and other violations efficiently. Schelter et al. presented a production-oriented system that translates data-quality checks into Spark aggregation workloads and supports anomaly detection over historical quality metrics [1]. Breck et al. described production data validation for machine-learning pipelines, emphasizing that data anomalies can undermine downstream model behavior even when pipeline infrastructure continues to operate [2]. PAIR builds on the importance of data validation but focuses on a different question: once a silent integrity

failure is known to exist, which transformation stage most plausibly originated the divergence?

B. Provenance and Failure Localization

Data provenance captures derivation history and has long been studied as a mechanism for explaining how data products arise. Buneman et al. formalized why- and where-provenance concepts for database results [3], while Simmhan et al. surveyed provenance characteristics and workflow-oriented uses [4]. More recent work has moved provenance closer to operational data-science pipelines: Chapman et al. developed fine-grained provenance for inspecting data-preparation transformations [13], and Schelter et al. used record-level provenance to screen machine-learning data-preparation pipelines for correctness issues [14]. PAIR uses provenance in a narrower operational sense: candidate stages are evaluated relative to observed checkpoint transitions and reference behavior. The experiments therefore test not only whether a pipeline is anomalous, but where a silent integrity failure most likely originated.

Taken together, prior work covers several neighboring capabilities but not the complete PAIR decision problem. Data-quality validation focuses on detecting violated constraints or abnormal statistics [1], [2]. Classical provenance research explains derivation and lineage relationships [3]-[6], while production data and machine-learning pipeline studies emphasize lifecycle, dependency, and operational reliability challenges [7]-[9], [12]. Recent empirical work has also characterized root causes and processing problem areas in data pipelines [15], while recent provenance systems capture end-to-end or fine-grained transformation history for modern data-science workflows [13], [14], [16]. PAIR builds on these foundations but evaluates a narrower post-execution localization problem: how to combine ordered reconciliation, provenance-relative ranking, reference-state awareness, and selective abstention when a pipeline finishes successfully but its data state is wrong.

The technical novelty evaluated in PAIR is not a new anomaly detector or a replacement for lineage systems. It is the coupling of three mechanisms for post-execution failure localization: (1) deterministic ordered checkpoint reconciliation when the reference is trustworthy, (2) provenance-relative candidate-stage ranking when direct first-divergence reasoning becomes ambiguous, and (3) a reference-aware abstention policy that separates the predicted origin from permission to automate. In contrast, validation systems primarily determine whether data violate expected properties, while classical provenance systems primarily represent or query how data were derived. PAIR evaluates whether those forms of evidence can be combined to identify a likely originating stage and explicitly decline automatic localization when reference quality or observability is insufficient.

Accordingly, the empirical comparisons in this study are scoped to mechanisms implemented within the benchmark: deterministic first-divergence reconciliation is the non-ML baseline, direct stage classifiers provide a learned absolute-label comparison, provenance-relative candidate ranking tests the proposed relational formulation, and the final hybrid evaluates reference-aware fusion plus selective abstention. The study does not claim a head-to-head benchmark against an external provenance product or root-cause-localization system; such a comparison requires a shared pipeline, checkpoint model, and failure corpus and is left for independent follow-on evaluation.

C. Research Gap

The target problem differs from ordinary job-failure diagnosis. The pipeline can be technically successful, and the final output can still be wrong. It also differs from simple rule-based validation because the first anomalous checkpoint is not always the true origin when checkpoints are missing, reference values are noisy, or a stale reference creates structural divergence. The central research question is therefore whether a hybrid deterministic and AI-assisted approach can preserve high-confidence automation where evidence is reliable while avoiding unsafe automation under ambiguous evidence.

III. PAIR FRAMEWORK

A. Pipeline and Checkpoint Model

PAIR models a pipeline as an ordered sequence of observable stages. The experimental pipeline includes source, ingestion, filter, normalize, join, deduplicate, aggregate, and target states. Each checkpoint is summarized by a compact signature rather than by retaining a complete duplicate of the dataset. The signature captures complementary evidence families: row cardinality, field completeness, aggregate statistics, content fingerprints, value distributions, and provenance-relative relationships between adjacent stages.

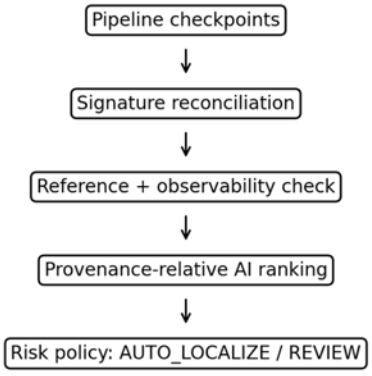


Fig. 1. PAIR end-to-end framework and decision flow

B. Deterministic First-Divergence Localization

For a clean reference, PAIR compares checkpoint signatures in pipeline order and identifies the earliest stage at which the observed execution materially diverges from the expected execution. This approach is intentionally deterministic. When reference integrity and checkpoint observability are strong, the framework does not invoke a statistical model merely to reproduce an answer already supported by direct evidence.

C. Ambiguity Detection and Provenance-Relative Ranking

Ambiguity arises when the earliest observed mismatch may not equal the true origin. PAIR distinguishes missing-checkpoint ambiguity from reference-divergence ambiguity. Missing checkpoints reduce observability but do not automatically invalidate the reference; therefore, the deterministic recommendation is retained and marked for review. Reference noise and stale-reference conditions can invalidate direct first-divergence assumptions. For these cases, candidate-stage records are constructed for every stage and ranked using features expressed relative to pipeline provenance and reference behavior. Random Forest, Logistic Regression, and Histogram Gradient Boosting models were evaluated during framework development.

D. Reference-Aware Fusion and Risk Policy

Reference state is inferred from observable checkpoint evidence rather than from the injected failure label. PAIR compares current checkpoint signatures with the available reference signatures. A clean reference shows no material reference-side divergence; value-only reference divergence changes value-oriented signature evidence while structural evidence remains consistent; structural stale-reference divergence changes structural evidence such as row-count, key-set, or schema-related signatures; and a missing-checkpoint condition is identified when a required checkpoint or reference signature is unavailable. The injected scenario label is retained only as ground truth for evaluation and is not provided to the routing logic or stage ranker.

The final fusion policy uses deterministic localization for clean-reference and missing-checkpoint conditions, while a learned stage ranker is used for reference-divergent cases. The strongest evaluated ranker in the final policy was Histogram Gradient Boosting. A separate risk layer converts the recommendation into either AUTO_LOCALIZE or REVIEW_REQUIRED. Clean-reference cases with full observability and value-only reference divergence were eligible for automatic localization; missing checkpoints and structural stale-reference divergence were routed to review. This separation makes recommendation quality and automation safety distinct evaluation objectives.

E. End-to-End Decision Procedure

PAIR processes a completed pipeline execution in four logical steps. First, it constructs checkpoint signatures for each observable stage and compares them with the corresponding reference signatures. Second, it evaluates the reference state and checkpoint coverage to determine whether the evidence satisfies deterministic assumptions. Third, when those assumptions hold, the earliest material divergence is returned as the recommended origin. When the reference is value-divergent but structurally comparable, candidate-stage features are passed to the provenance-relative ranker. Fourth, the risk policy converts the recommendation into AUTO_LOCALIZE or REVIEW_REQUIRED. The policy output therefore contains both a candidate origin and an automation decision; these are intentionally separate because a useful recommendation need not be safe for unattended operational action.

The decision procedure is designed to preserve explainability. A deterministic recommendation can be traced to the first checkpoint whose signature differs from its reference. A ranked recommendation can be inspected through the stage-relative feature vector and confidence score. A review decision additionally records the policy reason, such as a missing checkpoint or structural reference divergence. This separation supports operational auditability: engineers can distinguish what the framework believes happened, what evidence produced that belief, and whether the framework considers the evidence strong enough for automatic localization.

IV. Experimental Methodology

A. Benchmark Design

The benchmark contains 36 deterministic scenarios covering completeness, duplication, quality, semantic, transformation, aggregation, temporal, representation, and mixed-fault behavior. Representative faults include missing and unexpected records, partial batches, replayed batches, duplicate records, fan-out joins, under- and over-deduplication, null injection, value corruption, incorrect filters, wrong join keys, grouping errors, stale snapshots, truncation, key corruption, and combined faults. Each scenario has a known injected origin stage, enabling exact localization measurement.

The deterministic sensitivity benchmark evaluated multiple severities and seeds. Boundary testing then introduced ultra-low perturbations, noisy references, stale references, missing checkpoints, and multi-fault cases. An expanded ambiguity dataset contained 216 labeled runs across eight failure types and four observation conditions: clean reference, missing checkpoint, reference noise, and stale reference. Candidate-stage expansion produced 1,728 stage-ranking records.

The evaluation is organized as four explicit test groups. Test Group 1 evaluates 36 injected silent-failure scenarios under full checkpoint observability. Test Group 2 stresses the deterministic method with severity variation and boundary conditions including reference noise, stale references, missing checkpoints, and multi-fault cases. Test Group 3 evaluates learning and reference-aware fusion on the 216-run ambiguity benchmark, including random-split diagnostics, leave-one-failure-type-out generalization, candidate-stage ranking, and ablation tests. Test Group 4 evaluates implementation scalability on four representative failure origins from 10,000 through 1,000,000 rows. Each group reports its corresponding localization, ranking, coverage, uncertainty, or runtime metrics in Section V.

B. Failure Taxonomy and Injection Protocol

The 36-scenario benchmark is organized into nine four-scenario groups: cardinality, duplication, completeness, semantic, transformation, aggregation, temporal, representation, and mixed-fault behavior. Representative cases include missing and unexpected records, partial batches and batch replay, duplicate records and join fan-out, null injection and sparse-attribute loss, value corruption and mapping swaps, incorrect filters and wrong join keys, double counting and dropped categories, stale snapshots and out-of-order batches,

string truncation and key corruption, and compound faults that combine two error mechanisms. Each scenario records the injected failure type, the true originating stage, perturbation severity, random seed, affected-row count, and before/after cardinality so that localization predictions can be evaluated against explicit ground truth.

Failure injection is applied to an otherwise successful pipeline rather than by forcing task exceptions. This design intentionally targets silent integrity defects: executions remain technically valid while one or more data invariants change. Checkpoint signatures capture complementary evidence including row-count and distinct-key changes, duplicate and null deltas, aggregate drift, distributional changes, fingerprints, and provenance-relative differences. The benchmark therefore tests whether the framework can identify where trustworthy data first became inconsistent, rather than merely detecting that a Spark job failed.

C. Metrics

Primary metrics are exact stage-localization accuracy, mean stage distance between the predicted and true origin, within-one-stage accuracy, top-k ranking accuracy, and mean reciprocal rank. For the final policy, evaluation additionally reports overall recommendation accuracy, automatic-localization coverage, automatic-localization accuracy, and recommendation accuracy among review-required cases. Wilson confidence intervals and bootstrap confidence intervals were used for key proportions, and paired bootstrap differences were used for ablation comparisons.

D. Ambiguity Dataset and Learning Protocol

To move beyond ideal observability, the ambiguity benchmark generated 216 labeled runs spanning clean references, missing checkpoints, reference noise, and stale references. These runs produced 1,728 candidate-stage records for provenance-relative ranking. Candidate features describe how each stage differs from its reference context rather than encoding a failure-type identifier. Random Forest, logistic regression, and histogram gradient boosting rank candidate stages, while leave-one-failure-type-out evaluation measures whether ranking behavior transfers to failure mechanisms excluded from training. This separation is important because a useful operational localizer should reason from observable evidence rather than memorize scenario names.

For the ambiguity benchmark, routing and ranking inputs are computed only from checkpoint observations, reference comparisons, checkpoint availability, and provenance-relative candidate-stage features. The known injected origin is withheld from inference and used only afterward to score localization accuracy. Comparison rules distinguish material signature divergence from normal agreement, while the ranker receives candidate-stage features describing local divergence and relative upstream/downstream evidence. AUTO_LOCALIZE is allowed only when the applicable policy has sufficient observable evidence and confidence; otherwise the case is assigned REVIEW_REQUIRED. Thus, benchmark condition labels are evaluation metadata rather than inputs to the final decision.

The routing decision is evaluated independently from the injected ground-truth label. Checkpoint availability and reference-side signature comparisons first determine the observable evidence state. Clean-reference evidence follows deterministic first-divergence localization. Missing checkpoints preserve the deterministic recommendation but force REVIEW_REQUIRED because the evidence chain is incomplete. Reference-divergent cases are evaluated structurally: value-oriented divergence with structurally comparable checkpoints can invoke the candidate-stage ranker, whereas structural stale-reference divergence is routed to REVIEW_REQUIRED. The final AUTO_LOCALIZE decision therefore depends on observable reference state and checkpoint coverage rather than on the known scenario type used to score the experiment.

The final policy does not automatically act on every model recommendation. Reference state and observability determine whether a result is eligible for automatic localization. Clean-reference cases with full observability remain deterministic. Value-only reference divergence can invoke the learned ranker when the observed evidence supports reliable ranking. Missing checkpoints and structural reference divergence are routed to REVIEW_REQUIRED. This policy converts localization accuracy into a risk-aware decision process in which automation coverage and automation accuracy are reported separately.

E. Scalability and Reproducibility

For reproducibility, each checkpoint signature is composed from the signal families used throughout the experiments: row cardinality and distinct-key behavior, duplicate and null/completeness deltas, aggregate drift, distributional changes, content fingerprints, and provenance-relative upstream/downstream differences. Candidate-stage ranking uses these observable stage-relative signals rather than a failure-type identifier. The learning evaluation reports both a conventional random split and the stricter leave-one-failure-type-out protocol, in which each of the eight failure types is withheld in turn. Random Forest, logistic regression, and histogram gradient boosting are compared under this protocol, and the final hybrid uses histogram gradient boosting because it produced the strongest evaluated stage-distance behavior. Repeated evaluations use fixed seeds 42, 101, and 202, and experiment outputs are retained as machine-readable CSV/JSON artifacts.

Scalability experiments used representative failures originating at filter, normalize, join, and target stages. Dataset sizes ranged from ten thousand to one million rows. Reconciliation runtime and effective throughput were recorded while exact localization was checked at every size. Reproducibility was evaluated across seeds 42, 101, and 202.

F. Research Integrity Controls

All reported benchmark outcomes were generated by executable experiment scripts rather than manually assigned predictions. Fixed seeds (42, 101, and 202) were used for repeated evaluations, scenario metadata retained the injected ground-truth stage, and result artifacts were written to

machine-readable CSV/JSON summaries. Unit and integration tests were executed throughout framework development. The final statistical analysis reports Wilson confidence intervals for proportions, paired accuracy differences for ablations, and bootstrap confidence intervals for overall accuracy. These controls are intended to make the reported claims traceable to reproducible experimental outputs.

V. Results

A. Deterministic Localization Under Full Observability

In the initial 36-scenario benchmark, all scenarios completed without error and all injected failures were detected. Blind first-divergence localization identified the exact origin stage in 36 of 36 scenarios. Repeated sensitivity tests at severities 0.001, 0.01, and 0.05 also maintained perfect detection and localization for the evaluated cases. These results establish that deterministic reconciliation is sufficient when the reference is trustworthy and relevant checkpoints are observable.

Boundary experiments exposed the limits of that assumption. The deterministic first-divergence method also serves as the non-ML comparison baseline because it localizes solely from ordered checkpoint reconciliation without a learned ranker. Exact deterministic localization fell to 9 of 15 cases when reference noise, stale references, and missing checkpoints were introduced. The expanded 216-run ambiguity benchmark yielded an overall deterministic localization rate of 43.06 percent. Clean-reference accuracy remained 100 percent, but reference-noise and stale-reference cases failed systematically under this non-ML baseline.

B. Final Risk-Aware Policy

The final reference-aware hybrid achieved 75.00 percent overall recommendation accuracy across 216 ambiguity runs. The 95 percent confidence interval was 68.82 to 80.30 percent. The policy automatically localized 72 runs, corresponding to 33.33 percent coverage, with 100 percent observed accuracy and a 95 percent confidence interval of 94.93 to 100 percent. The remaining 144 runs were routed to review and had 62.50 percent recommendation accuracy. Thus, the policy did not equate a model recommendation with permission to automate.

C. Learning and Generalization Results

Within a conventional random train/test split of the 216 ambiguity records, machine-learning models appeared very strong: Random Forest reached 97.22 percent stage-localization accuracy and logistic regression reached 100 percent. These values were not treated as the primary generalization claim because random splitting allows closely related failure mechanisms to occur in both training and test data. A stricter leave-one-failure-type-out experiment therefore withheld each of eight failure types in turn. Under that protocol, direct stage classification generalized poorly: the deterministic baseline achieved 43.06 percent accuracy, Random Forest 18.06 percent, and logistic regression 37.50 percent. This negative result motivated the provenance-relative candidate-ranking formulation rather than direct multiclass stage prediction.

With candidate-stage ranking, cross-scenario performance improved. Aggregated leave-one-failure-type-out results produced top-1 accuracies of 53.24 percent for Random Forest, 50.00 percent for logistic regression, and 53.70 percent for histogram gradient boosting. Random Forest and histogram gradient boosting both reached 80.09 percent top-2 accuracy. The corresponding mean stage distances were 1.0463, 1.3611, and 0.9630 stages, respectively. These results show that ranking candidate origins relative to provenance context transfers better to unseen failure mechanisms than directly learning an absolute stage label.

Reference-aware fusion then combined the complementary strengths of deterministic reasoning and learned ranking. Using histogram gradient boosting as the ranker, the hybrid reached 75.00 percent overall recommendation accuracy with mean localization distance 0.3238 stages and 91.90 percent of recommendations within one stage of the true origin. The learned ranker was invoked for 44.44 percent of ambiguity runs rather than for every execution. This selective use of learning is central to PAIR: the model is introduced specifically where reference divergence makes deterministic first-divergence reasoning unreliable.

D. Accuracy by Ambiguity Condition

Condition-level results explain where the hybrid design succeeds and where uncertainty remains. Clean-reference and reference-noise cases both achieved 100 percent recommendation accuracy. Missing-checkpoint cases achieved 71.88 percent and were deliberately routed to review because incomplete observability weakens the evidence chain. Stale-reference cases were the most difficult condition at 43.75 percent, showing that structural reference drift remains a significant limitation.

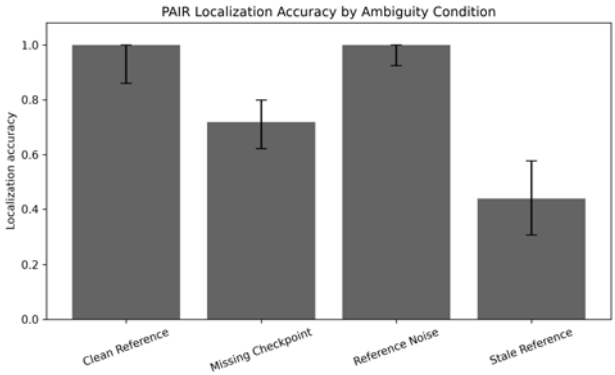


Fig. 1. Recommendation accuracy by ambiguity condition with 95 percent confidence intervals

E. Ablation Analysis

Ablation results show that the final framework depends on multiple complementary signals, but relative provenance is the most consequential. Removing relative-provenance features reduced overall accuracy from 75.00 percent to 54.63 percent, a 20.37 percentage-point decrease with a paired 95 percent confidence interval of 13.88 to 26.85 points. Removing aggregate signals caused the second-largest drop, 12.04 points. Distribution, completeness, cardinality, and fingerprint signals produced smaller but measurable degradations.

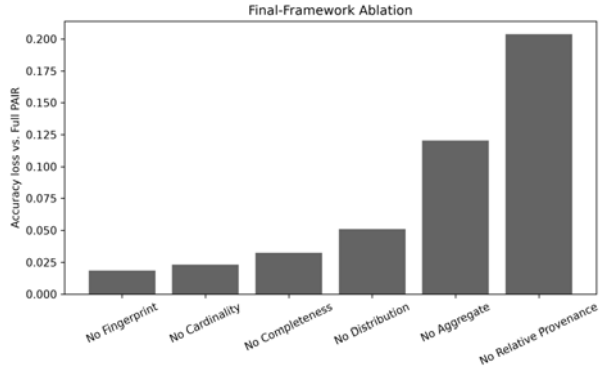


Fig. 2. Accuracy reduction when individual signal families are removed from the final framework.

F. Scalability and Reproducibility

Exact localization was preserved for all four representative scenarios at every evaluated dataset size. Mean reconciliation time increased from 10.74 seconds at ten thousand rows to 753.92 seconds at one million rows. Effective throughput peaked at approximately 2,350 rows per second at one hundred thousand rows and then declined to approximately 1,327 rows per second at one million rows. The non-monotonic throughput indicates that the current implementation is not a pure linear-scaling system and that Spark execution overhead, caching behavior, and stage-specific operations remain optimization opportunities.

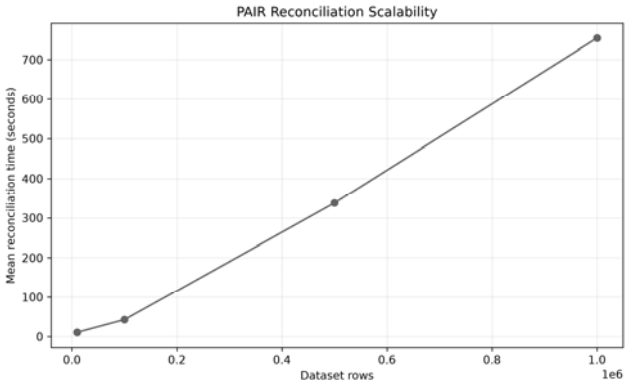


Fig. 3. Mean reconciliation runtime across dataset sizes

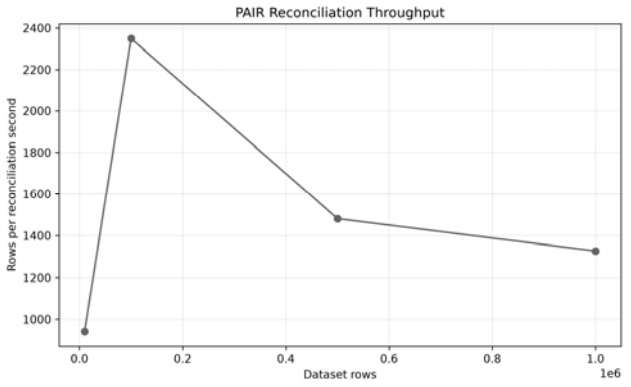


Fig. 4. Effective reconciliation throughput across dataset sizes

TABLE I. SCALABILITY RESULTS ACROSS DATASET SIZES

Rows	Mean reconciliation time	Throughput	Exact localization
10,000	10.74 s	941.8 rows/s	100%
100,000	42.88 s	2,350.1 rows/s	100%
500,000	337.72 s	1,480.8 rows/s	100%
1,000,000	753.92 s	1,326.8 rows/s	100%

Across seeds 42, 101, and 202, final-policy accuracy was 73.61, 75.00, and 76.39 percent, respectively. The narrow spread supports reproducibility of the aggregate result under the tested seed variation. Bootstrap estimation produced a mean overall accuracy of 75.00 percent with a 95 percent interval of approximately 68.98 to 80.56 percent.

G. Scalability Interpretation

The scalability measurements should be interpreted as an implementation-feasibility result in the tested Spark environment, not as a universal enterprise-scale performance claim. The benchmark was executed on one Spark environment and the reconciliation procedure performs multiple signature computations and checkpoint comparisons. Different cluster sizes, partitioning strategies, persistence policies, storage formats, and network conditions can materially change runtime. The supported finding is therefore limited to the observation that PAIR remained executable through one million rows without changing its decision logic or losing localization accuracy in the four selected scalability scenarios.

VI. DISCUSSION

A. Why Selective Automation Matters

The principal result is not that AI should replace deterministic reconciliation. The experiments support the opposite conclusion: deterministic evidence is preferable when it is trustworthy, while AI is most useful as a targeted ambiguity resolver. In the final policy, only one third of ambiguity runs were eligible for automatic localization, yet those cases achieved 100 percent observed accuracy. Higher-risk cases were not forced through the same automation path. This architecture converts uncertainty into an explicit operational state rather than hiding it behind a single prediction score.

B. Interpretation of the Provenance Ablation

The 20.37-point degradation after removing relative provenance is important because it distinguishes PAIR from a generic anomaly classifier. The ranker is not merely learning which raw quality metric is large. It benefits from knowing how candidate stages relate to upstream and downstream observations and to reference behavior. This supports the design premise that localization is a relational problem across a pipeline path, not only a classification problem over an isolated dataset snapshot.

C. Operational Use

In an enterprise deployment, PAIR can operate after a pipeline completes or after a downstream reconciliation check

signals a mismatch. Checkpoint signatures can be persisted as lightweight observability artifacts. When a divergence is detected, the framework can produce a candidate origin stage, evidence summary, risk decision, and review requirement. AUTO_LOCALIZE should be interpreted as permission to identify the likely origin automatically, not as permission to mutate production data automatically. Remediation can remain governed by existing change-control and approval processes.

D. Observed Failure Modes and Design Implications

The ambiguity experiments reveal two qualitatively different causes of localization error. Missing checkpoints remove evidence but do not necessarily invalidate the reference; deterministic reasoning therefore remains useful and achieved 71.88 percent recommendation accuracy in that condition. Stale references are more problematic because the comparison baseline itself can encode structural differences unrelated to the injected failure. The resulting 43.75 percent accuracy is a warning against treating every reference mismatch as causal evidence. PAIR consequently treats structural reference divergence as a review condition rather than an automation opportunity.

Ablation results reinforce this interpretation. Removing aggregate signals reduced overall accuracy by 12.04 percentage points, whereas removing relative provenance reduced it by 20.37 points and reduced automatic-localization accuracy by 47.22 points. Cardinality, completeness, fingerprint, and distribution signals each contributed smaller but measurable changes. The framework therefore benefits from signal diversity, while provenance-relative evidence provides the strongest contribution to the final risk-aware configuration.

VII. Threats to Validity and Limitations

The study uses controlled synthetic failure injection with known origin labels. This provides strong ground truth for localization but does not reproduce every dependency pattern, schema, data distribution, operational behavior, or naturally occurring failure mode of a production enterprise platform. The 36-scenario benchmark is broad but finite, and the ambiguity benchmark expands only eight selected failure types. No production incident corpus or second independent pipeline is evaluated in the present study. Results therefore demonstrate controlled feasibility within the tested benchmark rather than universal localization accuracy or real-world enterprise reliability.

PAIR depends on checkpoint and reference signatures that are sufficiently complete, current, and trustworthy. If reference signatures are noisy, stale, incorrectly generated, or missing important evidence, localization quality can decrease even when the pipeline itself executes successfully. This limitation is visible in the ambiguity benchmark: stale-reference cases achieved 43.75 percent recommendation accuracy, and missing-checkpoint cases achieved 71.88 percent. A production implementation should therefore validate reference freshness and checkpoint coverage before allowing automatic localization, and should route uncertain reference or incomplete-observability cases to human review.

The learned ranking component is statistical rather than deterministic, so its recommendation can retain residual uncertainty even when provenance-relative features are available. PAIR limits this risk by using the learned ranker only for selected ambiguity conditions and by separating AUTO_LOCALIZE from REVIEW_REQUIRED decisions; a model score is therefore not treated as proof of the failure origin. The scalability study also evaluates four representative failure scenarios in a single Spark execution environment, so runtime results should not be generalized directly to other cluster, storage, memory, executor, or network configurations.

VIII. Conclusion

This paper presented PAIR, a provenance-aware hybrid framework for detecting and localizing silent data integrity failures in enterprise data pipelines. PAIR combines deterministic checkpoint reconciliation, provenance-relative stage ranking, reference-aware fusion, and risk-aware selective automation. Deterministic localization was exact across all 36 clean benchmark scenarios, but ambiguity experiments showed that missing checkpoints and unreliable references can substantially weaken naive first-divergence reasoning. The final policy achieved 75.00 percent overall recommendation accuracy across 216 ambiguity runs and automatically localized 33.33 percent of cases with 100 percent observed accuracy, while routing higher-risk cases for review. Relative provenance was the most important evaluated signal family, with its removal reducing accuracy by 20.37 percentage points. In the tested single Spark environment, exact localization was preserved for four representative scalability scenarios through one million rows; this is an implementation-feasibility result rather than evidence of universal enterprise-scale performance. These results support a design in which AI augments, rather than replaces, deterministic reconciliation and in which automation authority is explicitly conditioned on evidence quality and operational risk.

IX. Future Work

In addition, the present evaluation uses completed batch-style pipeline runs and does not experimentally evaluate continuously changing or streaming pipelines. Future work should evaluate PAIR under streaming workloads, evolving reference state, checkpoint updates, and concept or data drift. Finally, the 100 percent AUTO_LOCALIZE accuracy was observed on 72 benchmark runs and should not be interpreted as guaranteed perfect accuracy in deployment. The benchmark also uses one designed eight-stage pipeline and synthetic injected faults, so the results establish controlled feasibility rather than universal enterprise reliability. Additional validation on independent pipelines, naturally occurring incidents, and different Spark or data-platform configurations is needed to test generality. The present study does not include a second independent pipeline or naturally occurring production incidents; those validations remain necessary to determine how well the routing and ranking behavior transfers beyond the designed eight-stage benchmark. A further priority is comparative evaluation on shared real-world workloads against existing provenance-based diagnostic and root-cause-localization approaches, using identical failure corpora and observability constraints.

X. Practical deployment implications

The results suggest that PAIR should be deployed as a decision-support layer rather than as an unconstrained automatic repair mechanism. Clean-reference cases can use deterministic first-divergence localization, while ambiguous cases should use the learned ranker only when the available checkpoint and reference evidence supports that route. The AUTO_LOCALIZE decision should remain restricted to cases that satisfy the observable-evidence and confidence conditions; all other cases should be sent to REVIEW_REQUIRED. This separation is important because the ambiguity benchmark showed that stale references and missing checkpoints can materially reduce recommendation accuracy even when the pipeline job itself completes successfully. In operational use, reference freshness, checkpoint coverage, and signature completeness should therefore be monitored as part of the localization workflow. The scalability experiment further shows feasibility in the tested Spark environment through one million rows, but it should not be interpreted as proof of universal enterprise-scale performance. These deployment constraints preserve the paper's central risk-aware principle: automation is useful when evidence is strong, while uncertainty should remain visible and trigger human review.

XI. References

[1] S. Schelter, D. Lange, P. Schmidt, M. Celikel, F. Biessmann, and A. Grafberger, “Automating large-scale data quality verification,” Proc. VLDB Endow., vol. 11, no. 12, pp. 1781–1794, 2018, doi: 10.14778/3229863.3229867.

[2] E. Breck, N. Polyzotis, S. Roy, S. E. Whang, and M. Zinkevich, “Data validation for machine learning,” in Proc. 2nd SysML Conf., 2019.

[3] P. Buneman, S. Khanna, and W.-C. Tan, “Why and where: A characterization of data provenance,” in Database Theory—ICDT 2001, LNCS 1973, pp. 316–330, 2001, doi: 10.1007/3-540-44503-X_20.

[4] L. Simmhan, B. Plale, and D. Gannon, “A survey of data provenance in e-science,” ACM SIGMOD Record, vol. 34, no. 3, pp. 31–36, 2005, doi: 10.1145/1084805.1084812.

[5] R. Bose and J. Frew, “Lineage retrieval for scientific data processing: A survey,” ACM Computing Surveys, vol. 37, no. 1, pp. 1–28, 2005, doi: 10.1145/1057977.1057978.

[6] J. Cheney, L. Chiticariu, and W.-C. Tan, “Provenance in databases: Why, how, and where,” Foundations and Trends in Databases, vol. 1, no. 4, pp. 379–474, 2009, doi: 10.1561/1900000006.

[7] N. Polyzotis, S. Roy, S. E. Whang, and M. Zinkevich, “Data management challenges in production machine learning,” in Proc. ACM SIGMOD Int. Conf. Management of Data, pp. 1723–1726, 2017, doi: 10.1145/3035918.3054782.

[8] N. Polyzotis, S. Roy, S. E. Whang, and M. Zinkevich, “Data lifecycle challenges in production machine learning: A survey,” ACM SIGMOD Record, vol. 47, no. 2, pp. 17–28, 2018, doi: 10.1145/3299887.3299891.

[9] D. Sculley et al., “Hidden technical debt in machine learning systems,” in Advances in Neural Information Processing Systems 28, pp. 2503–2511, 2015.

[10] L. Breiman, “Random forests,” Machine Learning, vol. 45, pp. 5–32, 2001.

[11] M. Zaharia et al., “Apache Spark: A unified engine for big data processing,” Communications of the ACM, vol. 59, no. 11, pp. 56–65, 2016.

[12] D. Xin, H. Miao, A. G. Parameswaran, and N. Polyzotis, “Production machine learning pipelines: Empirical analysis and optimization opportunities,” in Proc. ACM SIGMOD Int. Conf. Management of Data, pp. 2639–2652, 2021, doi: 10.1145/3448016.3457566.

[13] A. Chapman, L. Lauro, P. Missier, and R. Torlone, “Supporting better insights of data science pipelines with fine-grained provenance,” ACM Trans. Database Syst., vol. 49, no. 2, Art. no. 6, pp. 1–42, 2024, doi: 10.1145/3644385.

[14] S. Schelter, S. Guha, and S. Grafberger, “Automated provenance-based screening of ML data preparation pipelines,” Datenbank-Spektrum, vol. 24, pp. 187–196, 2024, doi: 10.1007/s13222-024-00483-4.

[15] H. Foidl, V. Golendukhina, R. Ramler, and M. Felderer, “Data pipeline quality: Influencing factors, root causes of data-related issues, and processing problem areas for developers,” J. Syst. Softw., vol. 207, Art. no. 111855, 2024, doi: 10.1016/j.jss.2023.111855.

[16] M. Schlegel and K.-U. Sattler, “Capturing end-to-end provenance for machine learning pipelines,” Inf. Syst., vol. 132, Art. no. 102495, 2025, doi: 10.1016/j.is.2024.102495.