

Blind ML: An Ethical Analysis of Privacy-Preserving Threat Research

Joe Waller

April, 15 2026

Abstract

Blind ML is an ethical research initiative that explores how security research should be conducted when the environment under study contains sensitive, operationally dangerous, or potentially harmful user activity. Conventional abuse-detection systems generally retain raw text, message bodies, URLs, and other content-rich evidence for downstream review. These conventional approaches strengthen semantic interpretation and aid in performance enhancement efforts, but requiring a retained-data surface increases blast radius, exposes operators to harmful material, and produces an artifact that can easily become a surveillance archive. Blind ML asks whether a different posture is possible: can useful hostile-behavior analysis still be performed when raw content is removed before persistence, thereby restricting downstream learning to names-only telemetry, structural metadata, and derived scalar features?

The Moltbook Observer Project is the implementation testbed for that question. Its architecture combines edge obfuscation, schema-bound packet families, staged simulation and collection lanes, fail-closed posture controls, hardened runtime boundaries, and local analysis workflows. The strongest ethical claim of the project is not that metadata-only learning eliminates platform-moderation risk. The strongest claim is that privacy, safety, and governance can be treated as binding requirements inside an ML workflow rather than as a compliance language attached after the system is already built. Viewed in its intended mature form, Blind ML becomes a governed multi-regime research program in which synthetic data, canary data, live-adjacent data, and reviewed release views all contribute distinct kinds of evidence without breaking the underlying privacy contract.

This report evaluates that mature design as an ethical data-science case. It argues that the project is compelling not because it eliminates every uncertainty in hostile-behavior detection, but because it demonstrates a disciplined way to study such behavior without normalizing unsafe retention.

1 Introduction

Much of contemporary platform-safety work assumes that raw content is the natural substrate of detection. If an investigator wants to study phishing, harassment, prompt injection, social engineering, or malicious coordination, the default pipeline captures the original text and treats richer semantic access as a prerequisite for meaningful modeling. That assumption has practical advantages, but it also creates ethical and operational burdens. Retained raw content increases the number of people and operational surfaces exposed to harmful material and amplifies the consequences of loss or misuse of data. In an agentic environment, where automation can collect, process, and act on information quickly and at scale, these burdens become more consequential.

These governance concerns are prominent in increasingly-complex agent-systems discussions, where security, interoperability, and operational control are being treated as first-class policy questions rather than background implementation details [1, 2].

Project repository: <https://github.com/joediggidy/calamum-moltbook-observer/>. Tooling disclosure: This project was developed using a repository governance harness and an AI-assisted coding/curation workflow engineered by Joe Waller and enabled via an OpenAI API. These tools are used to support development and documentation, including summarization, and do not replace the author's judgment. Names such as CodeSentinel, ORACL, and Calamum appear as internal identifiers for governance, agent assistance, and ethical-security harnesses, respectively.

Blind ML begins with a different assumption. Instead of asking how much content a security-research stack can safely retain, it asks how much useful inference can survive after semantic payloads have already been stripped away. The guiding hypothesis is not that metadata is sufficient for every security task; the hypothesis is that structural, temporal, and interaction-level signals may remain useful enough to support a meaningful class of hostile-behavior research while dramatically reducing the privacy and exposure costs associated with content-rich pipelines.

Previous work on metadata-based abuse detection and privacy-preserving moderation supports the plausibility of this direction by showing that non-semantic or reduced-semantic features can still contribute meaningfully to safety-relevant modeling tasks [3, 5, 4].

That hypothesis makes the Moltbook Observer strong as an ethical investigation, because the ethical question is inseparable from the technical design. The central issue is not simply whether a model can be trained, but whether a model can be trained under the honor privacy, proportionality, operator safety, and honest reporting constraints proposed by this framework. The observer is therefore best understood not merely as a software artifact but as a case study in ethical security by design.

2 Role, Sector, and Practical Problem

The role represented by this project can best be described as a hybrid of an ethical security researcher, a data guardian, and a defensive system designer. The sector is cyber-security and platform integrity, with a direct secondary connection to applied machine learning and responsible data governance. These roles matter because they define the standard against which the work should be judged. A commercial moderation product might optimize first for throughput, coverage, or intervention speed. An ethical security-research project must also justify how its measurement practices affect the people and systems it studies. This role-and-sector framing is also reflected in the project’s formal alignment assessment.

The practical problem is straightforward to state and difficult to solve well. Online platforms need ways to study suspicious or hostile behavior. Researchers need enough signal to identify patterns worth investigating. Operators need tools that are reproducible, controlled, and auditable. At the same time, platform subjects retain privacy interests even when their behavior is public-facing or risky, and operators should not be casually exposed to high-risk content when a safer design is possible. The project therefore confronts a familiar tension between security utility and private-data restraint, but it does so in a context where governance failures have immediate technical consequences.

The motivating question can be stated plainly: how can a research system study hostile activity without turning itself into a second site of harm? Harm in this context is not limited to breach risk. It includes operator exposure, quiet over-retention, mission creep, misleading reporting, and the gradual normalization of expansive collection simply because richer data is easier to model. Blind ML responds by treating bounded observation as part of the scientific method. That decision makes the learning problem harder, but it also makes the ethical stakes explicit.

3 Blind ML and the Observer Testbed

Blind ML is best defined as a *privacy-preserving machine-learning posture* rather than a single algorithm. Its core claim is that security-relevant research can be organized around obfuscated, schema-bound telemetry rather than retained message bodies. In the Calamum system, downstream analysis sees timestamps, event types, counts, flags, hashed identifiers, prompt-risk helper indicators, and derived scalar summaries such as complexity or code density. It does not depend on storing the underlying body text as the primary modeling substrate.

The Calamum Moltbook Observer turns that posture into an executable testbed. The public project overview and methodology manual describe a system with five clear commitments: synthetic-first validation, names-only telemetry, strict separation of scientific telemetry from diagnostics and safety governance, fail-closed enforcement, and reproducible local analysis. Privacy, accountability, and operator restraint are treated not only as first-class goals, but have the ability to determine if the project is allowed to proceed at all. The relevant project evidence surfaces are the public project overview and the methodology manual.

The project is especially interesting when understood through its architecture. The dataset authority layer does not simply pass downstream tooling whatever columns happen to appear in a raw retained table. Instead, reviewed dataset views are released with explicit field roles already classified. Identity, provenance, governance, and modeling features are separated before train, evaluate, and score stages consume the data. Ethically, that mechanism matters because it prevents downstream convenience from redefining what the data is supposed to mean. Interpretation and admissibility remain tied to reviewed release packets rather than ad hoc feature selection performed late in the pipeline.

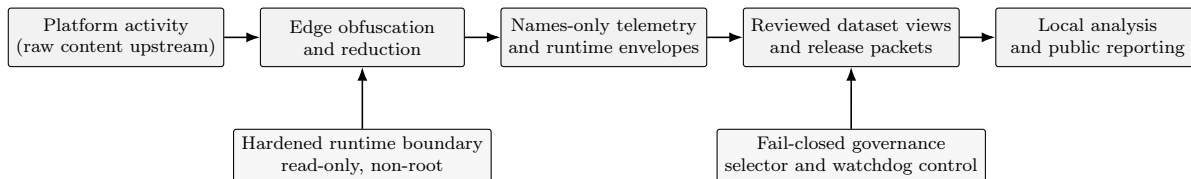


Figure 1: Governed Blind ML flow. Raw platform activity is reduced at the edge, converted into names-only telemetry, and released downstream only through reviewed dataset views and public reporting surfaces.

The strongest version of this paper is not a vague claim that the project is ethically thoughtful or operationally complete. It is a concrete description of what the fully realized experiment program is intended to look like. In its ideal state, Blind ML is not one model run against one convenient file. It is a controlled comparison across multiple data regimes, model families, thresholds, and reporting surfaces, all governed by the same names-only privacy contract. While the underlying premises and operational tooling have been thoroughly tested shown to be effective, the more robust operational features such as granular feature selection and dataset merging have been deferred to version 2 development.

3.1 Research Questions

The mature experiment lane is organized around four research questions. First, can structural metadata alone distinguish ordinary behavior from higher-risk behavior at a level that is useful for research triage? Second, can that usefulness be preserved across different data regimes, including synthetic, canary, live-adjacent, and reviewed honeypot slices? Third, can a conservative operating threshold be maintained so that the detector remains practically usable rather than simply sensitive? Fourth, can the governance architecture preserve truthful claims, reproducibility, and bounded retention while the experiment grows in complexity?

Taken together, these questions define success in a way that is distinct from ordinary leaderboard-style ML. The ideal outcome is not maximal predictive power at any cost. The ideal outcome is disciplined utility under a privacy-preserving observation contract.

3.2 Dataset Regimes

The mature experiment program relies on multiple dataset regimes because each one serves a different empirical and ethical role.

This regime-based design is important because it prevents the project from collapsing every research question into a single dataset and a single number. Synthetic data answers one class of question, benign canary data answers another, and real but names-only telemetry answers another still. The ethical benefit of this staged structure is that realism is introduced gradually and deliberately rather than treated as harmless by default.

3.3 Model Families and Comparison Strategy

Instead of treating one library call as the whole scientific contribution, the Observer’s experiment program compares multiple families of models. Heuristic baselines remain useful because they expose obvious structural rules and provide a floor for interpretation. Isolation Forest and related anomaly methods matter because the project can learn from unlabeled or weakly labeled regimes where conservative triage is the primary goal. Supervised models such as Logistic Regression and Random Forest become important once reviewed labels or

Dataset regime	Primary purpose	Ethical function	Ideal-state use
Synthetic TV-0 to TV-3 corpus	establish controlled truth conditions	supports low-risk validation before higher-risk observation	baseline supervised and unsupervised model development
Canary benign corpus	calibrate conservative false-positive posture	honors the precautionary principle by testing on safer traffic first	threshold tuning and benign-score distribution analysis
Live-adjacent names-only corpus	capture realistic operational structure	grounds the project in real telemetry without expanding raw-content retention	drift analysis, score distribution checks, robustness testing
Provenance-labeled companion view	support bounded reviewed comparisons	preserves label honesty by declaring what labels actually mean	limited supervised experiments with explicit interpretation boundaries
Reviewed honeypot slice	concentrate higher-risk but controlled observations	creates a tighter link between measurement and reviewed risk posture	focused high-risk evaluation and case-study analysis

Table 1: Dataset regimes in the intended mature Blind ML experiment program.

acceptable proxy labels exist, because they test whether the metadata-only regime supports discrimination beyond outlier ranking alone.

This comparison strategy is ethically relevant in two ways. First, it avoids overstating the evidence from a single favorite method. Second, it respects the fact that privacy-preserving ML should be judged partly by how robust its conclusions are to methodological variation. An ethical research program should not let the boundary conditions of one algorithm become the only story told about a privacy-constrained feature space.

3.4 Evaluation Criteria

The ideal-state evaluation lane reports precision, recall, F1-score, false-positive rate, and threshold behavior wherever labels permit. Where labels are partial or intentionally bounded, the most relevant measures are ranking quality, threshold conservatism, and cross-regime stability. This is especially important because a privacy-preserving detector becomes ethically weaker if it floods benign activity with alarms and then asks human reviewers to absorb the cost.

Success is therefore defined in layers. At the first layer, the project must show that the metadata-only feature space is non-trivial. At the next, it must show that conservative thresholds can be set and reproduced. At the strongest layer, it must show that multiple models trained on admissible dataset views converge on results that are interpretable, bounded, and honest about their uncertainty. Those layers together form the empirical counterpart to the ethical thesis.

4 Stakeholders and Ethical Matrix

The ethical matrix for this project is strongest when it stays close to concrete stakeholders rather than abstract principles alone. Four stakeholder groups matter most: platform subjects whose behavior is being measured, researchers and operators who run the system, platform owners or governance actors who authorize or review the work, and public readers such as graders, peer reviewers, or auditors who rely on the final report.

For platform subjects, the major ethical question is whether the system preserves enough restraint to justify observation at all. Edge obfuscation and names-only persistence matter because they narrow the retained-data surface before downstream analysis occurs. That does not eliminate privacy risk; metadata can still reveal patterns of timing, activity intensity, and interaction structure. But it changes the nature of the risk from broad semantic retention to a more bounded inference problem, which is ethically preferable if the research question can still be answered.

Stakeholder group	Primary good sought	Main ethical risk	Design choices that matter most
Platform subjects	privacy, proportionality, bounded observation	metadata inference, over-collection, misclassification	edge obfuscation, names-only telemetry, staged escalation
Researchers and operators	safety, reproducibility, interpretability	exposure to harmful content, unsafe runtime shortcuts, overconfidence	synthetic-first validation, hardened runtime, fail-closed controls
Platform owners and governance actors	accountable monitoring, auditable controls	mission creep, opaque evidence chains, weak transition discipline	posture gates, selector-based dataset governance, report lineage
Readers, reviewers, and auditors	truthful claims, clear evidence boundaries	polished but misleading reporting, hidden gaps between local and public artifacts	citation packs, explicit packet routing, honest claim boundaries

Table 2: Primary stakeholder groups and ethical trade-offs in the Blind ML design.

For researchers and operators, safety is part of the method. A research environment that quietly exposes staff to harmful content, expands permissions for convenience, or rewards unsafe shortcuts is not ethically rescued by a good model score. This is why the hardened container boundary, non-root execution, and fail-closed posture checks matter so much in the paper. They are not side features; they are part of the system’s moral scaffolding.

For platform owners and governance actors, the project’s strongest contribution is that it makes authority explicit. Dataset selection, mode transitions, and report publication are not supposed to be implied by scattered artifacts. They are governed by named control-plane surfaces and explicit release or selector decisions. In the ideal state of the project, that makes it harder for downstream analysis to inherit un-reviewed data assumptions by accident.

For readers and reviewers, the ethical obligation is honesty. The report should help the reader see what is real, what is partial, and what remains open. That is why the project’s ethical matrix ties privacy, accountability, operator safety, traceability, and honest reporting together. Utility without clear boundaries would weaken the project rather than strengthen it.

5 Data Access, Selection, Curation, and Privacy

The project’s data methodology manual provides the clearest evidence for its privacy posture. The methodology manual defines names-only packet families for content items, interaction events, runtime envelopes, baseline telemetry, evidence packets, and public report packets. Across these families, the retained fields focus on counts, flags, hashes, derived scalars, event metadata, and run linkage. This matters because ethical claims about privacy must be tied to what the system actually retains, not to aspirational statements about being careful.

The project is strongest where it treats the retention boundary as a first-class contract. The observer retains structure and behavior signals, not raw semantic payloads, as the primary research substrate. That does not make the data ethically neutral. Hashed identifiers, timing patterns, and metadata correlations can still support inference. But it creates a principled narrower surface that is easier to justify than ordinary content-centric logging. The public project overview and methodology manual make this boundary explicit in the repository documentation.

Selection and curation are equally important. The data story becomes most coherent when it recognizes that not every dataset view should be treated as morally or analytically interchangeable. Synthetic corpora offer explicit truth conditions. Benign canary data supports conservative threshold calibration. Live-adjacent names-only telemetry offers realism but often weaker labels. Reviewed honeypot slices offer concentrated structure under higher scrutiny. In the ideal-state architecture, these are not merely separate files. They are separately governed releases with explicit field-role definitions and explicit interpretation boundaries.

That curation model is ethically preferable to a looser pipeline because it reduces the chance that downstream code will quietly treat provenance, identity, or governance metadata as if it were modeling signal. The mature system releases a reviewed feature view for model consumers rather than a mixed-authority raw table. That design preserves both privacy and analytical honesty. It also gives the paper a stronger answer to the curation question: the project does not simply minimize data quantity, it also minimizes ambiguity about what each retained field is allowed to mean.

The label question deserves special emphasis. Provenance labels, review flags, and source-lane markers can support bounded experiments, but they are not the same as semantic ground truth about hostile content. One of the project’s ethical strengths is its willingness to preserve that distinction rather than smoothing it over for narrative convenience.

6 Governance Structures and Defense in Depth

Governance in this project is implemented as layered technical control. At the methodology level, the system commits to synthetic-first validation, names-only persistence, strict separation of telemetry from governance signals, and reproducible local analysis. At the runtime level, it relies on multiple protective layers: obfuscation before persistence, a hardened Docker boundary, and a watchdog runtime that monitors for fail signatures and can terminate unsafe states. At the data-authority level, selector and manifest logic distinguish between ordinary local artifacts and reviewed dataset surfaces intended for modeling or reporting.

This layered structure matters because it makes governance inspectable. A privacy-preserving ML paper is weaker if the privacy claim lives only in prose while the operational stack remains permissive by default. By contrast, the current design points to concrete controls. The container hardening report demonstrates read-only filesystem enforcement, non-root execution as UID 10001, and removal of unnecessary binaries such as `ping`. The methodology manual identifies the runtime boundary and the expected fail-closed posture. The public project overview distinguishes public documentation from local evidence and makes the reporting boundary explicit. These claims are documented in the container hardening verification report, the data methodology manual, and the public project overview.

The mature version of this governance model is especially compelling because it treats the report lane as part of the ethics contract. Public-facing reporting surfaces are not supposed to invent certainty that the local authority layer does not support. Reporting discipline is therefore not secondary to the project’s ethics, it is one of the ways the project enforces them. When the system declines to overstate labels, thresholds, or evaluation coverage, it is aligning the narrative layer with the truth conditions of the evidence layer.

7 Evidence, Claim Boundaries, and Limitations

The current project materials already support several claims with substantial confidence. They support the claim that privacy-preserving telemetry is operationally defined; the data methodology manual describes concrete packet families and invariants. They support the claim that runtime restraint is implemented rather than merely promised; the hardening report and related runtime surfaces document actual containment controls. They support the claim that reporting and dataset authority matter; the project distinguishes between reader-facing public packets, local canonical artifacts, and reviewed dataset or citation surfaces.

Two retained figures from the current observer analysis lane are especially useful for this paper because they make the project’s claim boundaries visible. They do not prove full hostile-behavior classification. They do show that the current metadata-only anomaly lane has a structured score landscape and a legible threshold posture on benign canary traffic, and that each collection mode workflow analysis produces data useful to downstream processing layers in building an ever-richer and more intelligent model.

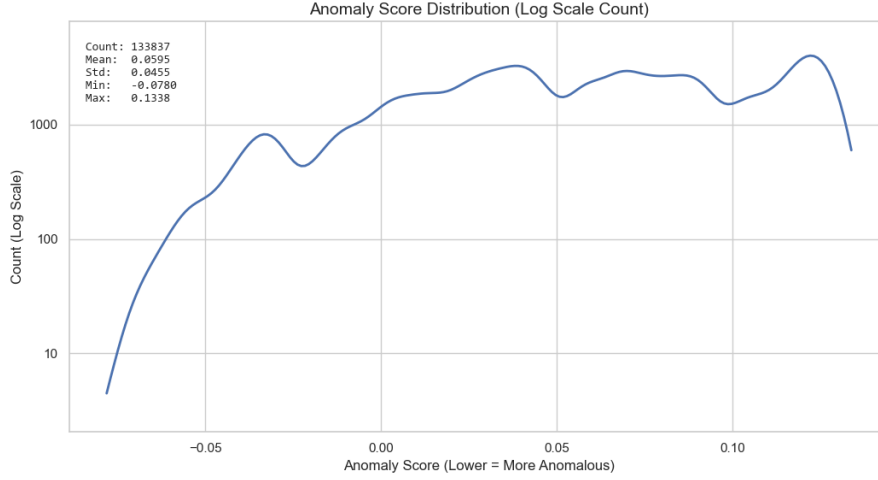


Figure 2: Benign canary anomaly-score distribution on a log-count scale. The broad, structured score landscape indicates that the metadata-only feature regime supports a non-trivial baseline rather than collapsing into a flat score mass.

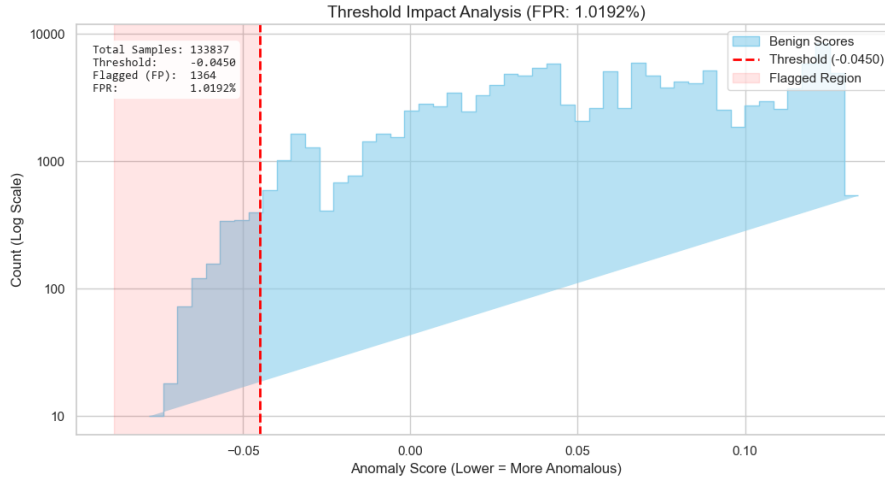


Figure 3: Threshold impact analysis for the benign canary calibration run. The selected operating point isolates a bounded flagged region while maintaining an approximately 1% benign false-positive rate.

What the repository does not yet fully prove on its own is the strongest possible form of the ML argument. The ideal-state experiment still needs complete cross-regime model comparison, stable released feature views for all modeling lanes, and evaluation artifacts that can support broader performance claims without stretching the meaning of partial labels. This is not a weakness in the ethical framing of the paper. It is part of what makes the project ethically credible.

The limitations are therefore real and important. Metadata-only learning imposes a ceiling on what the system can know. Content-free features may preserve timing, structure, and interaction patterns, but they can miss semantic nuance that matters for certain categories of threat or abuse. Data minimization also does not eliminate privacy risk. Metadata can still reveal activity rhythms, coordination, and other behavioral signatures. Finally, bounded label surfaces and reviewed slices remain bounded. They must not be converted into stronger semantic conclusions than they warrant.

These limitations do not weaken the ethical value of the project. They strengthen it because the project insists that privacy, minimization, and governance remain binding even when they make the empirical problem harder. The premise that drives this project is not incumbent upon making ethical security easy, the

goal is to define a surface that prioritizes personal data rights and agent-focused operational safety as an ethically-minded front line of defense.

8 Recommendations for Mature Deployment and Study

The first recommendation is to keep dataset release authority close to the Librarian layer so downstream consumers receive reviewed views rather than mixed-authority raw tables. The second is to compare multiple model families and multiple dataset regimes so the privacy-preserving claim is not tied to one fragile method. The third is to publish threshold rationale, dataset-view identity, and claim boundaries alongside every major result packet. The fourth is to treat human review and governance gates as permanent features of the design rather than as temporary scaffolding to be discarded once the modeling stack becomes more capable.

Together, these recommendations point toward an ethical research program that matures without abandoning its founding constraints. The project is most convincing when it does not trade away privacy or honesty in the name of apparent progress.

9 Conclusion

Calamum Moltbook Observer is a strong DATA740 case because it turns ethics into system design. Blind ML, in this project, is not simply a preference for privacy over performance. It is an attempt to show that privacy, operator safety, disciplined curation, and honest reporting can be built into the architecture of a security-research workflow from the start. Edge obfuscation, names-only telemetry, staged data regimes, hardened runtime controls, selector-based dataset authority, and explicit report boundaries all contribute to a coherent ethical method.

Viewed in its ideal experimental state, the project offers a compelling answer to the course’s core questions. It identifies a socially meaningful technical problem, grounds that problem in a real sector and stakeholder structure, and responds with a design whose ethical meaning is expressed through concrete choices about data, governance, and evidence. Its broader contribution is therefore not that it eliminates every uncertainty in hostile-behavior detection. Its contribution is that it demonstrates a more defensible way to ask those questions in the first place.

Blind ML is worth taking seriously as an ethical data-science proposal because it argues that useful security research does not have to begin by normalizing unsafe retention. It can begin with constraint, stay accountable as it scales, and still produce evidence worthy of study.

Project Evidence Surfaces

The internal project documents most directly supporting this paper are the public project overview, the data methodology and packet contract, the formal alignment assessment, the Blind ML ethical matrix, and the container hardening verification report. These documents are treated as project evidence surfaces rather than external bibliography items.

References

- [1] Announcing the “AI Agent Standards Initiative” for Interoperable and Secure Innovation. 2026. NIST, February 18. <https://www.nist.gov/news-events/news/2026/02/announcing-ai-agent-standards-initiative-interoperable-and-secure>.
- [2] CAISI issues request for information about securing AI agent systems. 2026. NIST, January 12. <https://www.nist.gov/news-events/news/2026/01/caisi-issues-request-information-about-securing-ai-agent-systems>.
- [3] M. Casavantes, M. E. Aragón, L. C. González, and M. Montes-Y-Gómez. Leveraging posts’ and authors’ metadata to spot several forms of abusive comments in Twitter. *Journal of Intelligent Information Systems*, 61(2):519–539, 2023. <https://doi.org/10.1007/s10844-023-00779-z>.

- [4] P. Leonidou, N. Kourtellis, N. Salamanos, and M. Sirivianos. Privacy-preserving online content moderation: A federated learning use case. *arXiv preprint arXiv:2209.11843*, 2022. <https://arxiv.org/abs/2209.11843>.
- [5] Z. Zhou and T. Islam. Power of Social Media Meta Data: Enhancing Cyberbullying Detection. In *Advanced Sciences and Technologies for Security Applications*, pages 339–361, 2025. https://doi.org/10.1007/978-3-031-82031-1_19.

A Dataset Snapshot and Extended EDA

This appendix gathers the dataset surfaces, feature structure, and build summaries that support the Observer study. The main evidence base combines a benign canary calibration surface, a live-adjacent collection corpus, and a bounded provenance-labeled companion built from that collection. A smaller reviewed honeypot subset broadens the observed range and contributes bounded higher-risk proof of life through reviewed labeling, governed reviewed-dataset registration, and analysis-ready dataset preparation.

A.1 Regime overview

Figure 4 gives a reader-facing view of the four dataset surfaces referenced in this appendix. The canary surface anchors calibration, the live-adjacent corpus supplies the main observed collection base, the provenance-labeled companion adds bounded source-aware labels, and the reviewed honeypot subset broadens the operational picture under stricter runtime conditions with 103 retained rows spanning 50 unique reviewed records, 26 bounded labeled identities, and 24 explicitly unlabeled reviewed identities.

Canary calibration anchor 133,837 benign calibration samples 412,340 retained canary records Calibration surface plus broader benign reference corpus	Live-adjacent collection corpus 13,842 retained records Main observed collection base for the current analysis
Provenance-labeled companion 208 uniquely labeled records over the same 13,842-record collection Bounded source-aware labels for stratified analysis and later source-aware comparison work	Reviewed honeypot subset 103 retained rows across 50 unique reviewed records 26 bounded labeled identities; 24 explicitly unlabeled identities Bounded higher-risk proof of life under stricter runtime conditions

Figure 4: Dataset surfaces referenced in Appendix A. The calibration anchor, live-adjacent collection, bounded provenance companion, and reviewed honeypot proof-of-life subset each contribute a distinct public-facing role within the broader Observer workflow.

Canary counts. The canary box keeps two counts because they support different claims: 412,340 retained records describe the broader canary corpus, while 133,837 benign samples define the public calibration run cited in the main discussion.

A.2 Retained observation unit

Each retained row is one obfuscated event-level record keyed by a stable `record_id`. The dataset builder materializes timing, structure counts, surface markers, prompt-risk indicators, derived scalar features, and one-hot event-type flags into the `obfuscated_record_v1` schema. Within the retained modeling workflow, analysis operates on structural telemetry only: timestamps, lengths, counts, flags, and derived scalars remain available after reduction, while message text stays outside the stored ML surface. Additive fields that are absent in older records are normalized to stable zero or false defaults during extraction so the dataset contract remains consistent across the admitted collection regimes.

A.3 Feature families

Table 3 keeps this section at the family level rather than dumping raw field names. Appendix B can carry the stricter field-by-field contract later.

Feature family	Signals carried in this family
Identity and timing (2)	Stable record identity and event timing for deterministic splitting, ordering, and alignment across derived outputs.
Content structure (5)	Length, word-count, line-count, question-mark, and exclamation-mark signals that describe message shape without storing message text.
Surface markers (6)	Code-block, link, tag, and mention indicators that preserve interaction form and visible surface behavior.
Prompt-risk indicators (6)	Reduced flags and scores for prompt-injection style references, instruction-boundary mentions, and matched pattern activity.
Derived scalars (4)	Compact risk-oriented summaries such as complexity, code density, toxicity, and canary-source membership.
Event type (7)	One-hot interaction categories covering posts, replies, reposts, direct messages, follows, mentions, and unknown events.

Table 3: Feature families used in the live-adjacent dataset build.

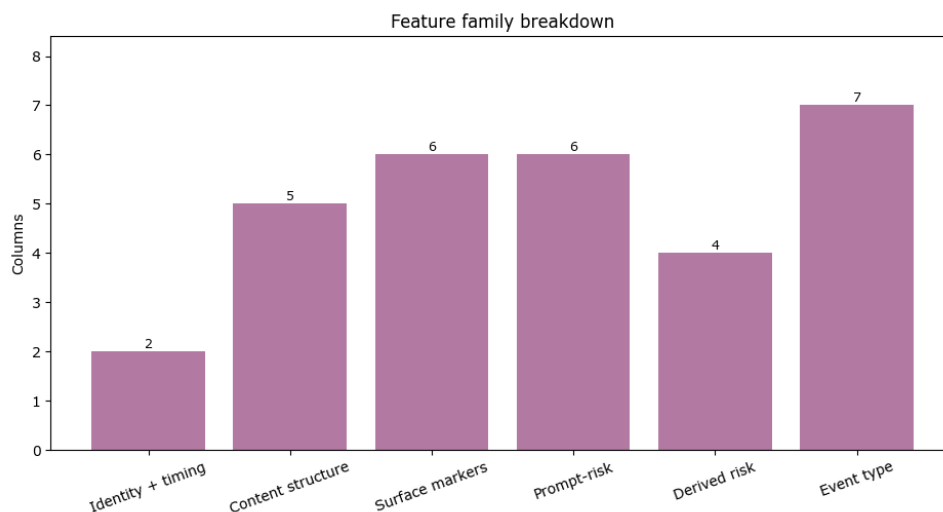


Figure 5: Feature-family counts in the live-adjacent dataset build.

This grouping keeps the observation model reader-facing. Identity and timing support deterministic splits, structural and surface markers preserve behavioral shape, reduced risk indicators capture the strongest blind signals, and event-type flags retain coarse interaction form without storing raw content.

A.4 Extended EDA summary

The comparison-ready build summarized here remains tightly tied to the live-adjacent collection corpus. Its input slices contribute 13,718 live-adjacent records and 124 canary-sourced records, yielding the 13,842-row dataset used for the current build. The deterministic split manifest preserves a near-exact 70/15/15 allocation with 9,587 training rows, 2,297 validation rows, and 1,958 test rows. Among the visuals gathered here, the broader input-volume reference view, *Split balance*, and *Feature family breakdown* add the clearest appendix value because they make both dataset context and dataset structure legible without overloading the main body.

Build snapshot.

- **Input composition:** 13,718 live-adjacent records and 124 canary-sourced records.
- **Deterministic split:** 9,587 training rows (69.26%), 2,297 validation rows (16.59%), and 1,958 test rows (14.15%).
- **Bounded labeled companion:** 208 uniquely labeled records, with 84 live-source labels and 124 canary-source labels.
- **Visuals kept in this appendix:** broader input-volume context, feature-family breakdown, and split balance, with the regime map above carrying the broader dataset story.
- **Calibration visuals held for later use:** score distribution and threshold overlay.

The inserted input-volume plot now serves as broader reference context rather than as a direct rendering of the admitted 13,842-row live-adjacent build. It shows the larger canary/reference slice family feeding the canary-aligned build lineage, with the 133,837-sample benign calibration slice appearing as the dominant visible input. The live-adjacent build counts remain the ones listed in the snapshot bullets above.

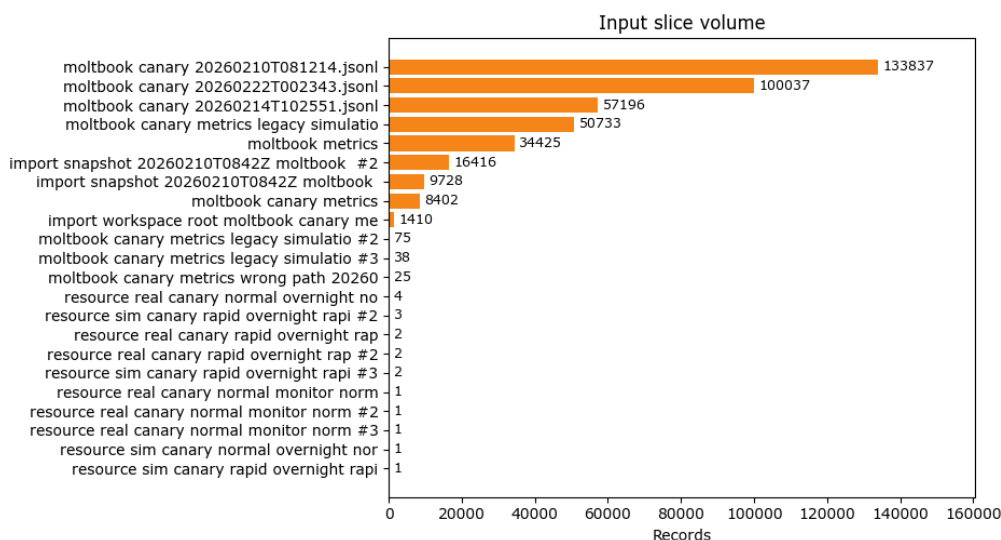


Figure 6: Relative input-slice volumes for the broader canary/reference build family. The largest visible slice corresponds to the 133,837-sample benign calibration surface, so this figure should be read as intake context rather than as the admitted composition of the 13,842-row live-adjacent comparison-ready build.

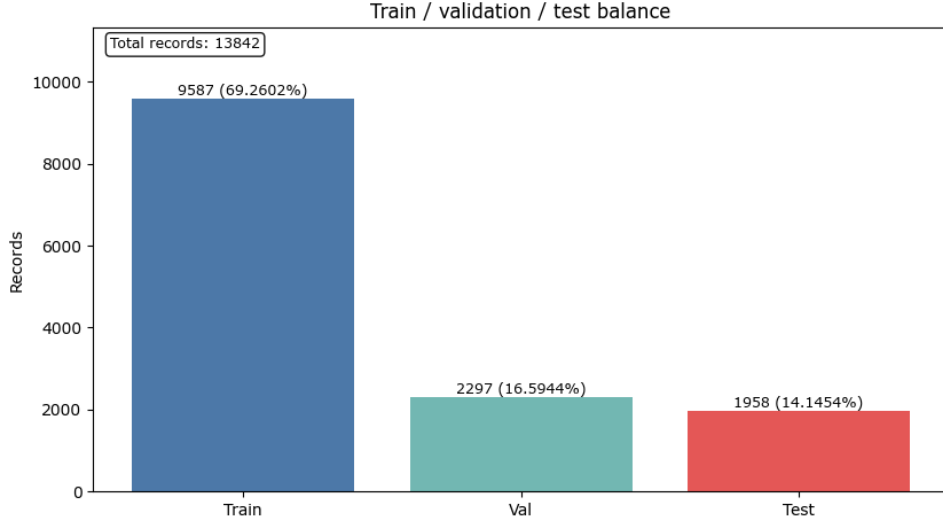


Figure 7: Deterministic train, validation, and test split for the live-adjacent comparison-ready build.

Taken together, the input-volume view, feature-family breakdown, and split-balance outputs show two distinct layers of Appendix A evidence. The input-volume figure situates the broader canary/reference intake behind the calibration lineage, while the feature-family and split-balance figures describe the structure of the admitted 13,842-row live-adjacent build summarized in the bullets above. This separation keeps the larger reference intake visible without confusing it with the composition of the current comparison-ready corpus.

B Feature Contract

This appendix documents the current privacy-safe feature contract for the Observer analysis lane. The governing distinction is between the broader obfuscated telemetry schema accepted at ingestion time and the narrower dataset table emitted for modeling. The schema carries the fields needed to validate and normalize privacy-safe telemetry, while the builder writes a reproducible CSV contract composed only of structural metadata and derived scalars.

B.1 Current admitted columns

Table 4 lists the current admitted feature families exactly as they are materialized into the live-adjacent dataset bundle. The emitted table carries 30 columns in total and matches the feature contract recorded in the dataset manifest and downstream training artifacts.

Feature family	Current emitted columns	Reader-facing role
Stable identity and timing (2)	<code>record_id</code> , <code>ts_epoch</code>	Preserve stable row identity, deterministic split alignment, and event timing without retaining message text.
Content-shape counts (5)	<code>content_length</code> , <code>content_length_words</code> , <code>line_count</code> , <code>question_count</code> , <code>exclamation_count</code>	Describe message size and visible shape through counts only.
Surface markers (6)	<code>has_code_block</code> , <code>code_block_count</code> , <code>has_link</code> , <code>link_count</code> , <code>tags_count</code> , <code>mentions_count</code>	Capture visible interaction form, formatting, and link or tagging behavior without retaining the underlying content.
Prompt-risk indicators (6)	<code>contains_ignore_previous</code> , <code>contains_system_prompt_reference</code> , <code>contains_developer_message_reference</code> , <code>contains_env_var_reference</code> , <code>prompt_injection_score</code> , <code>matched_pattern_count</code>	Reduce prompt-manipulation cues into flags and counts rather than storing matched strings or raw prompts.
Derived scalars and regime flag (4)	<code>f_complexity</code> , <code>f_code_density</code> , <code>f_toxicity</code> , <code>is_canary</code>	Carry compact numeric summaries and a canary-membership flag for audit and regime tracking.
Event-type one-hot fields (7)	<code>type_post</code> , <code>type_reply</code> , <code>type_repost</code> , <code>type_dm</code> , <code>type_follow</code> , <code>type_mention</code> , <code>type_unknown</code>	Normalize event-kind strings into a fixed model-ready vocabulary.

Table 4: Current admitted feature contract for the Observer dataset builder.

B.2 Materialization rules

The builder applies a small set of stable transformation rules before the dataset reaches the training and evaluation lane.

- **Stable row identity.** Each record receives a deterministic `record_id` so feature rows, split assignments, and optional labels can be joined safely across artifacts.
- **Timestamp reduction.** The emitted `ts_epoch` value prefers the promoted scalar `f_timestamp_epoch`; if that scalar is absent, the builder falls back to parsing timestamp or `ts`.
- **Boolean normalization.** Source booleans are written into the CSV as binary integer flags, so fields such as `has_code_block`, `has_link`, the `contains_*` indicators, `is_canary`, and all `type_*` columns appear as 0/1 values in the admitted feature table.
- **Pattern reduction.** The broader schema can carry `matched_pattern_labels` as an array of strings, but the current dataset contract reduces that surface to the scalar `matched_pattern_count`.
- **Event-type normalization.** Source event labels are mapped into seven one-hot columns, with `type_unknown` serving as the fallback when the incoming event type falls outside the current vocabulary.
- **Backward-compatible defaults.** Additive uplift fields that are missing in older records are normalized to stable zero or false defaults so earlier retained rows remain admissible under the same contract.
- **Separate label channel.** When a labeled companion is present, labels are written to a separate `labels.csv` surface keyed by `record_id` rather than embedded inside `features.csv`.

B.3 What stays outside the feature table

The broader obfuscated-record schema permits additional routing, provenance, and integrity fields, including `packet_family`, `packet_version`, `venue_id`, `entity_kind`, `source_id_hash`, `author_hash`, `sender_hash`, `matched_pattern_labels`, `node_id`, `mode`, `kind`, `signature`, and `tv_id`. Those fields remain useful for ingestion, governance, or auxiliary reporting, but they are not part of the current admitted modeling matrix. This boundary matters for two reasons: it keeps hashed actor identifiers and raw matched-pattern strings out of the learning surface, and it prevents runtime or routing metadata from quietly becoming a stand-in for the privacy-safe behavioral signals the study is actually trying to evaluate.

B.4 How the contract supports the current analytical core

Current training and scoring routines load the feature contract from the dataset manifest, then remove `record_id` and `is_canary` before constructing learned feature vectors. The remaining numeric columns feed the current ApexLab-backed Isolation Forest and Random Forest workflows. In practical terms, that means the admitted model surface is still fully names-only: no message bodies, no raw links, no matched-pattern label strings, no hashed actor identifiers, and no secret-bearing runtime fields enter the learned representation. What remains is the structural evidence the analytical core uses—timing, shape, formatting markers, risk cues, and compact derived scalars—kept in a form that is reproducible, auditable, and narrow enough to preserve the Blind ML privacy boundary.

C Evidence Lineage and Reader-Facing Traceability

This appendix shows how the main evidence surfaces fit together. The evidence chain begins with a benign calibration anchor, a live-adjacent collection snapshot, and a bounded provenance-labeled companion drawn from that same collection. It then continues through the build, training, evaluation, and scoring summaries that carry the same study surfaces forward into the key figures and metrics. The result is a stable, readable progression from dataset authority to observed findings.

C.1 Evidence chain across the current study

The current evidence base rests on three public-facing anchors. First, the *Canary V1 Inference Report* establishes the calibration story with 133,837 benign examples, a threshold near -0.0451 , and a benign false-positive rate close to 1%. Second, the live-adjacent current-collection snapshot establishes the 13,842-row corpus described in Appendix A. Third, the provenance-labeled companion adds the 208 uniquely labeled `record_id` values that support bounded source-aware comparison work and the admitted feature contract.

From there, the analytical story advances through four aligned summary layers. The build summary materializes the dataset bundle and deterministic split. The training summary records the available source-aware model path and the admitted feature contract. The evaluation summary records threshold-publication posture for that path, and the scoring summary carries the broader canary-distribution view that supports the calibration story. Read together, these surfaces preserve one continuous evidence chain across the project.

C.2 Current collection pair

The current collection pair is organized as a main empirical base plus a focused companion layer. The current-collection snapshot contributes 13,718 recent live records and 124 recent canary records, yielding the 13,842-row corpus used throughout the current analysis. The provenance-labeled companion builds on that same corpus and adds 208 uniquely labeled `record_id` values for source-aware analysis and later source-aware comparison work.

Within that companion, the label values have a direct reader-facing meaning. Label 0 marks the recent live slice, and label 1 marks the recent canary slice. This compact label layer supports provenance-aware analysis while keeping the broader dataset story unified around one stable collection snapshot.

C.3 Reviewed honeypot proof-of-life chain

Beyond the main current-collection pair, the evidence base now also includes a bounded reviewed honeypot proof-of-life chain. The reviewed subset contains 103 retained rows spanning 50 unique reviewed records, with 26 bounded labeled identities and 24 explicitly unlabeled reviewed identities. That higher-risk subset has already passed through governed reviewed-dataset registration and into the same analysis-ready dataset preparation workflow used elsewhere in the study. A selector-visible registration step and emitted comparison-baseline packet make that handoff explicit in the current authority spine. It therefore extends the lineage story from collection and review into governed higher-risk admission and downstream consumability without overstating the present scale of multi-class evidence.

C.4 How the summary surfaces stay aligned

A reader-facing summary in this project consistently carries four ingredients:

- **dataset identity**, so the reader can see which study surface a result belongs to;
- **workflow stage**, so the role of each summary is clear inside the larger pipeline;
- **core outputs**, such as dataset bundles, metrics, figures, or score distributions;
- **interpretive role**, so each surface contributes a specific analytical claim.

That structure is what keeps the lineage legible. The build summary clarifies dataset materialization, the training and evaluation summaries clarify the source-aware comparison path, and the canary calibration surface plus scoring summary clarify the calibration story. Each surface adds one part of the overall picture while preserving a shared study identity.

C.5 Analytical role

This lineage supports dataset-role interpretation, calibration, companion-label meaning, and current experimental interpretation. Appendix C carries the connective tissue that links those ideas into one forward-moving evidence story. In that form, lineage serves the reader directly: it shows how the live-adjacent corpus, the bounded companion, and the stage-specific summaries fit together into a coherent research contribution.

D Additional Figures and Result Displays

This appendix gathers the supplementary displays that deepen the current results story while keeping the core discussion selective. The retained additions fall into three roles: a fuller view of benign canary calibration, a broader score-surface view for the 412,340-record comparison corpus, and a bounded reviewed honeypot proof-of-life note.

D.1 Supplementary calibration views

Benign canary calibration remains the clearest empirical anchor for the Blind ML workflow. Figure 8 shows the full 133,837-sample benign distribution on a linear count scale, which makes the dominant benign mass easier to read than a tail-focused display alone. Figure 9 then shows how the selected threshold near -0.045 isolates only the extreme lower tail, flagging 1,364 benign samples and keeping the benign flagged share at approximately 1.02%.

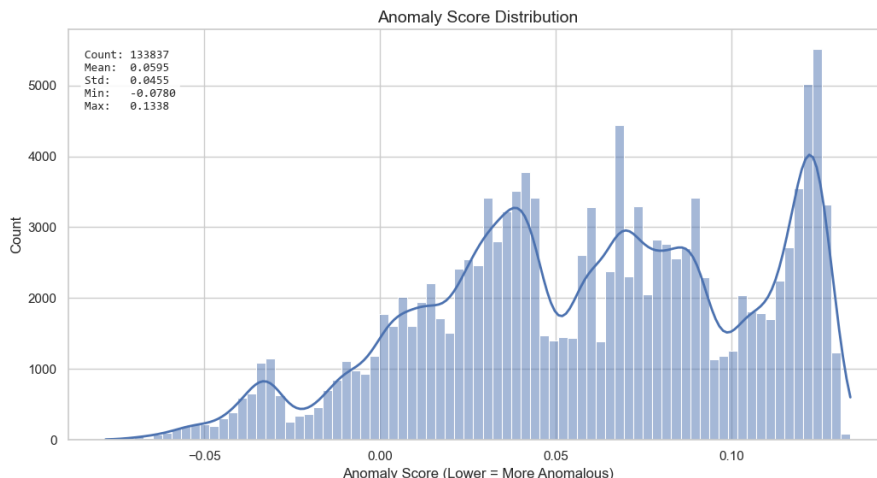


Figure 8: Linear-count view of anomaly scores across the 133,837 benign samples used in the Canary V1 calibration study.

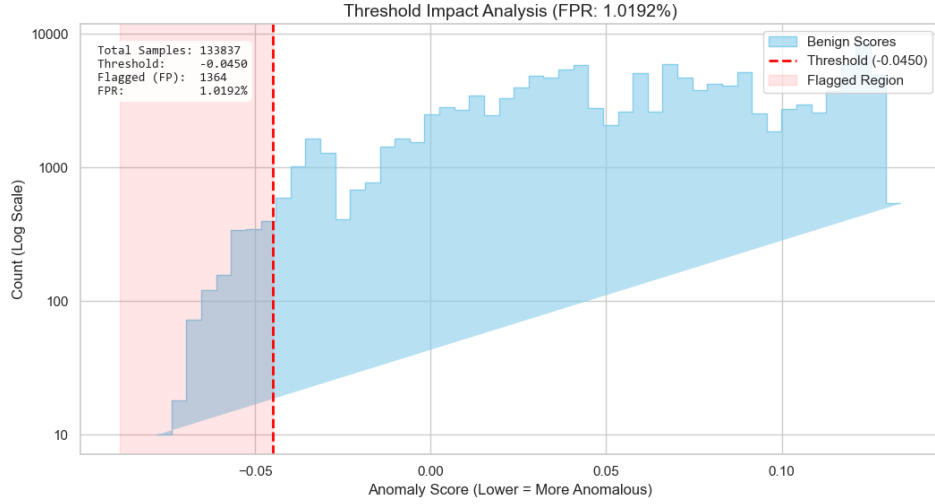


Figure 9: Threshold-impact view for the Canary V1 calibration run. The highlighted lower tail contains 1,364 benign samples, corresponding to an approximate 1.02% flagged share at the selected operating point.

D.2 Broader comparison score surface

The larger canary-aligned comparison corpus extends the calibration story beyond the smaller benign anchor. Figure 10 shows the full score surface for 412,340 scored records under the current unsupervised lineage. The paired evaluation packet publishes a threshold of 0.434329 and selects 4,478 records for review, a flagged share of 1.086%. Read together, the calibration and comparison views show the same conservative pattern: the review boundary sits at the lower edge of the score mass rather than near its center.

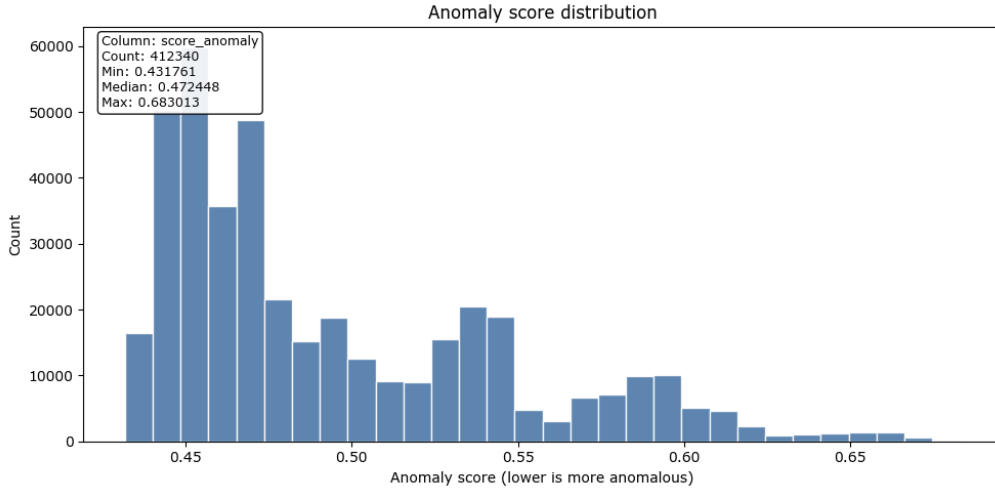


Figure 10: Score distribution for the 412,340-record comparison corpus used in the current canary-aligned scoring lane. The paired evaluation packet selects a 1.086% lower-tail slice for review.

D.3 Reviewed higher-risk proof of life

The current paper also includes a separate higher-risk evidence thread in the form of a bounded reviewed honeypot subset. That subset spans 103 retained rows, 50 unique reviewed records, 26 bounded labeled identities, and 24 explicitly unlabeled reviewed identities. It shows that reviewed labeling, governed reviewed-dataset registration, and downstream analysis-ready dataset preparation already function in a higher-risk regime. This evidence complements the calibration and canary-score figures above while keeping the claim

envelope disciplined, with a full honeypot train/evaluate benchmark family positioned as a later expansion step.

E Limitations and Threats to Validity

This appendix makes the current claim envelope explicit. The strongest completed quantitative anchor remains benign-regime anomaly calibration on 133,837 benign examples, and the current evidence base also includes a bounded reviewed honeypot subset that demonstrates governed reviewed labeling, reviewed-dataset registration, and downstream analysis-ready dataset preparation within the same workflow. Table 5 summarizes the main validity boundaries, and the subsections below explain each one in reader-facing terms.

Boundary	Current evidence anchor	Implication for interpretation
Metadata-only observation surface	30-column structural feature table built from timing, counts, markers, prompt-risk indicators, derived scalars, and event-type flags	Downstream models can screen behavioral structure, but distinctions that depend on retained semantic content or longer discourse context remain outside the admitted representation.
Provenance label posture	208 uniquely labeled records, with label 0 marking the recent real/live slice and label 1 marking the recent sim/canary slice	This label surface supports bounded source-aware comparison work inside the current corpus, not full hostile-behavior ground truth.
Threshold and regime transfer	Benign calibration near a 1% false-positive target on 133,837 samples and a broader canary flagged share of 1.086% on 412,340 records	Numeric cutoffs are run-specific and score-surface specific, so they should be interpreted within their own lineage rather than copied across regimes.
Reviewed higher-risk breadth	The reviewed honeypot companion spans 103 retained rows, 50 unique reviewed records, 26 labeled identities, and an explicit unlabeled remainder	Regime breadth is real, but multi-class prevalence claims and cross-regime generalization remain bounded.

Table 5: Current Appendix E validity boundaries.

E.1 Metadata-only observation ceiling

The retained model surface is deliberately narrow. The admitted feature table keeps only privacy-safe structural telemetry: timing, length, counts, visible formatting markers, prompt-risk indicators, derived scalar summaries, and one-hot event types. It excludes retained message text, raw matched-pattern strings, and actor-identifying fields from the learned representation. That boundary is central to the Blind ML contribution, because it keeps the downstream analytical surface names-only and reproducible. It also creates an explicit information ceiling. Distinctions that depend on lexical meaning, longer conversational context, sarcasm, or topic-specific semantics are only partially recoverable once the retained representation has been reduced to structure alone. The current results should therefore be read as evidence that privacy-safe behavioral screening is possible under a constrained observation model, not as evidence that every harmful-behavior distinction survives full semantic stripping.

E.2 Provenance-label supervision boundary

The active labeled companion is a provenance layer derived from the existing canary-membership field already present in the approved feature table. Its 208 uniquely labeled `record_id` values divide into 84 recent real/live records and 124 recent sim/canary records. Those labels preserve source-slice truth for the current corpus and make the dataset comparison-ready for stratified analysis. They do not define a hostile-behavior ground-truth benchmark. In consequence, this label layer supports bounded source-aware comparison work inside the current corpus, while fuller multi-class threat discrimination still depends on broader reviewed labels.

E.3 Threshold transfer and score-surface comparability

The current project evidence contains several threshold-bearing surfaces, each with a different interpretive role. The benign calibration report on 133,837 samples places the operating point near -0.0451 in order to hold the benign false-positive rate at approximately 1%. A later canary-facing evaluation summary on 412,340 records reports a threshold of 0.434329 and a 1.086% lower-tail review share under the same lower-is-more-anomalous scoring direction. These values support a consistent guardrail idea—conservative lower-tail review at a bounded operating point—but they do not live on one universal numeric scale. Each threshold belongs to its own run lineage, score surface, and label posture, so direct numeric transfer across calibration, canary comparison, and reviewed higher-risk settings would overstate comparability.

E.4 External validity and regime breadth

The study already spans more than one observed regime: a large benign canary corpus, a live-adjacent collection corpus, a provenance-labeled companion built from that live corpus, and a reviewed higher-risk honeypot subset. This breadth is a real strength because it shows that the same privacy-safe feature contract can survive across different collection conditions. At the same time, the strongest labeled high-risk evidence remains small. The reviewed honeypot companion covers 103 retained rows, which collapse to 50 unique reviewed records, of which 26 received bounded labels while 24 remained explicitly unlabeled. That scale is appropriate for demonstrating that a reviewed higher-risk regime exists and can be governed honestly, but it remains too small to support broad prevalence claims, robust multi-class comparison, or strong cross-regime generalization. More reviewed labels, longer time coverage, and broader venue diversity would all improve the external validity of the present findings.

Taken together, these limitations define the current scientific boundary in a constructive way. The project has already established a governed metadata-only workflow, a strong benign-calibration result, bounded label-ready extensions, and a reviewed higher-risk proof-of-life surface. The next empirical gain comes from expanding reviewed label coverage and controlled cross-regime evaluation. Governance maturity and runtime discipline strengthen traceability throughout that process, and larger labeled evidence will further strengthen performance claims.

F Privacy Boundary, Governed Custody, and Protected Release Notes

This appendix makes the paper’s data-handling boundary explicit. The retained analytical lane begins only after reduction into names-only structural telemetry, and the same governed workflow keeps dataset custody, integrity checks, protected release, and reader-facing publication aligned without collapsing them into one ambiguous authority surface.

F.1 Normal analytical privacy boundary

The normal ML lane persists structural telemetry and derived scalars rather than retained message text. In the current admitted workflow, the stored analytical surface is built from event timing, event class, lengths, counts, surface markers, prompt-risk indicators, deterministic row identity, and compact derived features such as complexity, code density, and toxicity. Earlier safety handling may still inspect raw content briefly inside the controlled upstream environment, but that text stays outside the stored dataset, the admitted learned feature table, and the reader-facing publication packets discussed elsewhere in the paper.

This same names-only rule shapes supporting artifacts. Values that would be unsafe to expose in a static appendix are carried through hashes, selectors, or reference-handle fields whose names end in `_ref` rather than copied into the released artifact itself. In practical terms, routing fields, raw matched-pattern strings, and hashed actor identifiers remain governance-side context rather than part of the current learned surface.

F.2 Names-only custody and derived-publication discipline

The project keeps source authority, derived artifacts, and publication views in separate roles. Approved dataset selectors, manifest-backed dataset definitions, and run-ledger surfaces carry the canonical machine-readable lineage for admission, preparation, evaluation, and release. Generated figures, collection packets, and aggregate report pages sit downstream as reader-facing publication surfaces rebuilt from those governed artifacts. This one-way relationship matters methodologically: published packets can summarize a run clearly for readers, but they do not become configuration inputs or a second machine-readable authority layer.

That boundary also explains why the project treats generated artifacts conservatively. Public documentation may reference canonical local runtime paths and evidence families, but those references remain descriptive. They help the reader follow the workflow without turning local runtime evidence into tracked public authority.

F.3 Integrity-checked vault role in governance and reproducibility

Governed custody is enforced through an integrity-checked librarian vault rather than through ad hoc file placement. In the current observer workflow, the normal operating posture is the locked state: approved dataset selectors, admission snapshots, and protected release projections remain available for signed operations while the integrity layer compares the current managed-surface checksum against a recorded baseline checksum and preserves the associated audit surfaces. Manual maintenance is bounded rather than informal: unlock opens a deliberate maintenance window, relock restores the normal posture, and rebaseline refreshes the checksum anchor once the controlled mutation is complete.

This integrity layer contributes directly to reproducibility. Registration and protected-release mutations refresh the vault baseline, preserve tracked-file counts and checksum evidence, and append names-only audit records. As a result, dataset admission and later protected release remain attributable to specific governed mutations instead of becoming undocumented file copies.

F.4 Protected-source request, attestation, and release flow

Protected-source dataset access follows a short delegated release sequence once an approved selector has been resolved to a ready manifest.

1. A signed `request.json` packet records the requester, requested action, expiry, access class, source binding, and dataset-manifest checksum.
2. That request is verified before release proceeds.
3. A signed `attestation.json` packet binds the request hash to the selector identity, readiness state, source binding, and grant decision at the governed dataset-authority layer.
4. A signed `release_receipt.json` packet then binds the request and attestation hashes to the released manifest path and the granted protected-source action.
5. The release mutation refreshes the vault baseline and appends an audit record so the delegated handoff remains reviewable after the fact.

Local approved datasets resolve directly from the admitted manifest path, but protected-source datasets enter this delegated request $\rightarrow$ attestation $\rightarrow$ release flow so the release surface stays attributable, bounded, and checksum-coupled to the same governed custody layer.

F.5 Reader-facing publication views versus canonical authority surfaces

Table 6 summarizes the authority split used throughout the paper.

Surface family	Reader-facing role	Authority meaning
Approved dataset selectors, manifest-backed dataset definitions, release receipts, and run ledgers	Machine-readable custody and lineage	Canonical local authority for dataset identity, admission, preparation, and governed release.
Tracked docs/reports/ collection packets, dated stage packets, and packet-declared figures	Human-facing publication views	Derived from canonical local artifacts; used for reading and citation support without replacing local machine-readable authority.
Manuals, policy docs, and methodology notes	Public contract and interpretation guide	Describe boundaries, paths, and expectations without becoming runtime evidence themselves.

Table 6: Authority split between canonical local artifacts and reader-facing publication surfaces.

This split keeps the paper reader-first without flattening the governance model. The narrative can point readers to public packets and validation publications, while the underlying selector catalogs, manifests, release receipts, and run ledgers remain the canonical local surfaces that make those publications reproducible.

G KEYSMITH Secret-Authority Lane

KEYSMITH is the current v1 secret-authority lane for higher-risk Moltbook registration and credential minting inside the broader Observer governance stack. Its operator-facing authority remains a mint-only surface, and its purpose is narrow by design: obtain the vendor credential without printing it, route the human only through the non-secret claim URL, and keep the secret inside a sealed-drop handoff that can be imported into the stricter runtime lanes without copy/paste handling.

G.1 Host dry-run versus sandbox live-mint boundary

The operator-facing authority surface remains `observerctl ops keysmith mint`, backed by a single underlying `keysmith mint` command family. The dry-run path is intentionally permissive enough for local verification: it can execute from the host, writes into a timestamped `local_untracked/keysmith_exports/` directory, and produces a deterministic-shaped artifact set without contacting the vendor. Live minting is stricter. A non-dry-run execution must occur in the approved sandbox/container lane, where `KEYSMITH_SANDBOX=1` marks the controlled execution surface and the output root is constrained to the sandbox-approved export directory.

That boundary is fail-closed rather than advisory. KEYSMITH denies non-dry-run execution outside the sandbox lane, denies outputs inside the tracked project tree outside `local_untracked/`, denies non-empty output directories to avoid sealed-drop reuse, and denies non-allowlisted hosts. In the current implementation, the allowlisted canonical vendor host is `www.moltbook.com`, and the sandbox runner mounts the project-local export root into the container so the resulting names-only artifact bundle remains reviewable after execution.

G.2 Sealed-drop bundle shape and names-only handoff

Once the mint succeeds, the shipped bundle is small and deliberate. Humans receive only the claim URL and never the credential value itself; the secret remains in the sealed-drop file and in downstream import surfaces that report presence only.

Artifact	Public-facing role	Current handling posture
<code>claim_url.txt</code>	Human claim/verification route	Non-secret URL recorded in plaintext so the operator can complete the vendor ceremony without viewing the credential.
<code>sealed_drop.bin</code>	Secret handoff surface	Stores the API key bytes in the bounded export directory; the value is never printed or logged.
<code>keysmith_audit.jsonl</code>	Names-only execution evidence	Records start, artifact-write, and completion events using paths, timestamps, and byte counts rather than secret material.
<code>keysmith_result.json</code>	Manifest of emitted outputs	Lists the artifact paths, bundle notes, and execution metadata in a reader-safe JSON surface.
Import and persistence helpers	Session and user-environment hydration	Load the sealed drop into the current process or Windows user environment while reporting presence only, not the key value itself.

Table 7: Current KEYSMITH artifact bundle and handling posture.

The important design point is that the sealed drop is the handoff boundary, not a human-readable checkpoint. The current shipped code treats it as a short-window plaintext-at-rest artifact inside the bounded export root, while the audit trail records only names-only metadata such as artifact paths and sealed-drop byte length. This is why the paper can describe KEYSMITH as a secret-authority surface rather than as an ordinary registration helper.

G.3 Authority role and readiness nuance

Within the broader authority matrix, KEYSMITH does not create the **live** and **honeypot** regimes; those stricter collection regimes already exist inside the governed observer continuum. Instead, KEYSMITH supplies the credential-acquisition and hydration path required for real-source operation while Watchdog and the observer runtime continue to own the downstream fail-closed movement gates. In practice, this means the runtime still refuses real-source work when `MOLTBOOK_API_KEY` is absent, even though the rest of the collection stack is already admitted and operational.

That distinction explains the current status nuance. The shipped KEYSMITH lane can truthfully report that the surface is present and that host dry-run remains available while live mint authority is still sandbox-only. In the current operator status packet, this split can appear as a human-readable success state for the shipped surface alongside a JSON-only nonzero readiness code when the sandbox lane is not armed. That is shell-local readiness signaling, not evidence that the broader **live** or **honeypot** collection reality is hypothetical. Those regimes already belong to the admitted governed workflow; KEYSMITH simply keeps their credential path inside a separate, controlled access surface.

H Sandbox Validation and Empirical Rehearsal Lane

This appendix documents the stable sandbox family exposed through `observerctl sandbox` as the project’s retained empirical rehearsal and validation surface. The operator-facing contract is intentionally compact: `list`, `show`, and `run` expose the stable probe catalog and execute one named definition, while `sandbox runs list` and `sandbox runs show` reopen retained runs for later review. The current retained lane is organized around exact-name-only stable definitions grouped by validation role rather than by implementation chronology. A separate `feedback-loop` simulation remains available for temp-only experimentation, but it is not part of the indexed retained review spine summarized here.

H.1 Grouped probe families

The current stable sandbox catalog covers metadata integrity, DS-wizard hydration and state continuity, DS-wizard truthfulness, publication alias coherence, baseline-monitor runtime continuity, and librarian governance. Table 8 summarizes those families at the role level rather than replaying the full historical order in which the probes were added.

Probe family	Current stable definitions	Behavior validated
Metadata contract and regression	metadata-contract, metadata-contract-regression	Confirms that retained sample and index fields remain present and that contract regressions surface as reviewable failures rather than silent schema drift.
DS-wizard hydration and persisted state	ds-wizard-hydration, ds-wizard-stale-state-continuity, ds-wizard-durability, ds-wizard-labeled-eval-contract-coherence	Confirms selector hydration, stale-state recovery, persisted draft continuity, and labeled-evaluation report coherence for the dataset workflow lane.
DS-wizard execution truthfulness	ds-wizard-blocked-execute-truthfulness, ds-wizard-execute-failure-truthfulness	Confirms that blocked or failing execute paths emit truthful previews, reason codes, and fail-closed review packets instead of implying a successful run.
Publication alias coherence	ds-alias-coherence	Confirms that build, train, evaluate, and score artifacts stay under the registered alias root and that unresolved aliases fail closed instead of drifting into fallback publication paths.
Baseline-monitor runtime and continuity	baseline-monitor-runtime, validation-cycle-lineage, baseline-monitor-restart-continuity, baseline-monitor-state-recovery	Confirms runtime liveness, retention continuity, validation-cycle lineage, restart continuity, and malformed-state recovery for the watchdog-adjacent runtime surface.
Librarian governed exchange and vault controls	librarian-access-exchange, librarian-vault-controls	Confirms protected-source request, attestation, and release flow; signature verification; vault baseline and audit updates; and lock, unlock, rebaseline, and verify control behavior.

Table 8: Current sandbox probe families and the behaviors they validate.

H.2 Append-only retained review surfaces

Each retained sandbox family writes into a stable subtree under `report_tmp/`, typically with per-run material under `runs/<run_id>/` and an append-only `run_index.jsonl` file at the family root. Those index rows preserve the definition identity, timestamp, result, report paths, and next-bite status so later review does not depend on a single latest-run overwrite. The per-run artifacts are intentionally paired: a JSON report preserves the machine-readable check matrix, while a Markdown companion preserves the same run as a compact review surface.

The CLI mirrors that storage discipline. Before execution, `observerctl sandbox show <definition>` surfaces purpose, outputs, guardrails, and run-index destination for the selected stable definition. After execution, `observerctl sandbox runs show <run_id>` summarizes retained checks, command results, and surfaced findings for the saved run. In current retained examples, baseline-monitor runtime runs preserve monitor-state and resource-retention evidence, DS-wizard truthfulness runs preserve command previews and validation blockers, and librarian governance runs preserve request, attestation, release, and vault-control artifacts. Names-only terminal output and no-secret-printing rules remain part of the documented guardrail surface throughout that review flow.

H.3 Why this lane is stronger than smoke testing

This sandbox lane is stronger than a smoke-test family because it validates artifact content, publication coherence, continuity behavior, and governance semantics rather than only exit codes. The retained baseline-monitor runtime probe, for example, checks that monitor state is written, `resource_normal` retention continuity is real, and the resulting evidence packet carries the expected readiness and reporting sections. The librarian access-exchange probe confirms that protected-source request, attestation, and release artifacts are written and signature-verified while the corresponding vault baseline and vault audit surfaces are refreshed. The librarian vault-controls probe then confirms that registration is denied while the vault is locked and that rebaseline and verify return the managed surface to an `ok` integrity state.

The same pattern holds for the DS-facing probes. Hydration, stale-state, durability, alias-coherence, and truthfulness runs verify what the system actually writes and how it fails, not merely whether a command starts. In methodological terms, that is why the project treats `observerctl sandbox` as a retained empirical rehearsal lane: each run leaves a reviewable evidence packet about successful checks, denied transitions, and emitted artifacts, which makes the lane a stronger validation surface than a conventional smoke-test suite.

I Runtime State, Posture Mapping, and Transition Contract

This appendix defines the small runtime vocabulary behind the watchdog and posture language used in the main body. In the current `observerctl` authority spine, runtime truth is not a bare mode label or a process-start event. The canonical state is the tuple (*source, mode*), and movement between states is carried by names-only gate, set, transition, and evidence packets. That distinction matters reader-facing because starting the observer process does not by itself change the authoritative route state. Route truth changes only when the saved runtime state and its control packet move together.

I.1 Canonical tuple and posture map

Table 9 summarizes the current state vocabulary. The `source` axis identifies whether the collection lane is simulated or real-source, the `mode` axis identifies the active operational lane, and the trigger posture is derived from mode rather than chosen independently.

Runtime element	Current values	Reader-facing meaning
<code>source</code> axis	<code>sim</code> , <code>real</code>	Distinguishes simulated collection from real-source collection and determines whether real-source dependency checks apply.
<code>mode</code> axis	<code>watch</code> , <code>canary</code> , <code>live</code> , <code>honeypot</code>	Identifies the current operational lane and the target named state carried in gate and transition packets.
Trigger posture	<code>watch/canary</code> → <code>isolation</code> ; <code>live/honeypot</code> → <code>lockdown</code>	Determines the watchdog cadence and the stricter prerequisite bundle expected for higher-risk moves.

Table 9: Canonical runtime tuple and derived posture mapping.

Because posture is mode-derived, `live` and `honeypot` are not merely descriptive labels. They activate the lockdown family, while `watch` and `canary` remain isolation paths. In practice, any request targeting `live` or `honeypot`, or promoting `sim` to `real`, is treated as a stricter path.

I.2 Transition classes and gate expectations

The public control surface stays intentionally small. `ops mode current` and `ops mode list` expose the saved runtime state, `gate` produces a read-only go/no-go packet, `set` applies a freshly admitted change, and `transition` performs the guarded gate → set → evidence workflow. The important reader-facing point is that these are different layers of authority: a runtime may be running, yet the authoritative route state remains unchanged until the state packet itself moves.

Transition class	Meaning	Gate expectation
No-op	Requested state already matches the saved state	Denied fail-closed rather than silently re-admitted; current tests assert <code>policy_denied:no_op_transition</code> .
Lateral move	Source stays fixed while mode changes	Must prove state coherence, heartbeat freshness, target support, posture validity, and security linkage before the move is admitted.
Source escalation	<code>sim</code> → <code>real</code>	Adds the real-source dependency and higher-risk readiness checks on top of the ordinary mode gate.
Source de-escalation	<code>real</code> → <code>sim</code>	Remains an explicit packeted move rather than an implicit fallback, preserving a truthful state history.

Table 10: Current transition classes and their reader-facing gate behavior.

For the stricter lockdown targets, the gate does more than check whether a process is alive. Current code and tests require coherent runtime state, watchdog and observer freshness, a valid trigger posture, security-report linkage, escalated heartbeat and baseline-validation cadence, an active baseline-monitor runtime, retained `resource_normal` continuity, and a complete baseline-window proof surface. The same packet family remains names-only throughout, surfacing fields such as `timestamp_utc`, `from_state`, `to_state`, `decision`, `reason_codes`, posture information, and `security_report_ref` rather than secret-bearing runtime material.

I.3 Normalized denial surfaces and retained runtime proof

The denial surface is intentionally normalized and fail-closed. The current authority spine uses representative reason codes such as `policy_denied:no_op_transition`, `policy_denied:target_mode_unsupported`, `critical_check_failed:run_security_report_missing`, `critical_check_failed:watchdog_trigger_posture_invalid`, `critical_check_failed:lockdown_heartbeat_rate_not_escalated`, `critical_check_failed:lockdown_baseline_rate_not_escalated`, `critical_check_failed:baseline_monitor_runtime_inactive`, `critical_check_failed:resource_stream_retention_unavailable`, `critical_check_failed:resource_baseline_window_incomplete`, and `critical_check_failed:cpu_spike_lockdown`. For the lockdown path, the current test suite also fixes the activation-order vocabulary deterministically, so missing cadence, inactive monitor runtime, and incomplete retention or baseline evidence surface in a stable order rather than as ad hoc prose.

The retained probe family shows that this contract is empirical rather than purely documentary. The baseline-monitor runtime probe keeps the current state at `sim:canary` while generating a non-activation live-projection packet whose lockdown-prerequisite bundle is fully green, confirming active monitor runtime plus retained normal-stream and baseline-window evidence without falsely claiming that the system has already moved to `sim:live`. The validation-cycle lineage probe preserves append-only two-cycle linkage, so the latest cycle retains prior validation, baseline, and analysis anchors rather than overwriting them. The restart-continuity probe then shows the same continuity state marked `preserved` across a resumed live-mode cycle, while the malformed-state recovery probe marks continuity `degraded` with `major_check_failed:baseline_monitor_state_malformed`, repairs the saved anchors into a normalized form, and still writes a fresh validation cycle instead of discarding state silently.

Taken together, the runtime contract is watchdog-governed, names-only, and deliberately fail-closed. `watch` and `canary` are isolation states, whereas `live` and `honeypot` are lockdown requests that must prove tighter cadence and stronger continuity before the control plane will admit them. The reader-facing implication is straightforward: runtime authority in this project is not merely “which process is running,” but which (*source, mode*) state has been explicitly gated, written, and evidenced under the current posture rules.

J ApexLab Reference Validation

This appendix summarizes the 2026-03-24 ApexLab reference-validation run as supporting infrastructure evidence for the broader Observer data-science lane. Its purpose is to show that the lightweight statistical and regression helpers used by the current analytical stack remain numerically aligned with trusted reference implementations while preserving ApexLab’s low-dependency design. Within the paper’s claim structure, this appendix supports the analytical tooling boundary behind the lane and complements the main empirical contribution.

J.1 Validation frame

The retained evidence base for this appendix consists of three aligned surfaces: the ApexLab reference-validation runner `projects/apexlab/examples/reference_validation_run.py`, the machine-readable metrics artifact `projects/apexlab/logs/metrics/reference_validation_20260324.json`, and the companion validation report `projects/calamum-moltbook-observer/docs/reports/validations/APEXLAB_REFERENCE_VALIDATION_REPORT_20260324.md`. Together, those surfaces record the validation question, declared thresholds, observed deltas, and interpretation boundary in a form that is both reviewable and reproducible.

The validation question was simple and operationally important: are the current lightweight ApexLab comparison and regression helpers close enough to trusted SciPy and scikit-learn references to support the active Observer integration lane without adding heavyweight runtime dependencies? A passing answer in this context requires bounded numeric agreement under declared tolerance bands.

This appendix therefore presents the results in grouped numeric displays rather than one compressed narrative summary. The grouping follows the validation design itself: statistical comparison, OLS regression, and logistic regression.

Table 11 keeps one compact entry point at the front of the appendix, using the capability-group matrix idea from the earlier draft but tightening it into a denser numeric overview.

Capability group	Fixture and reference	Key bounded read	Primary interpretation
Statistical comparison	$n_a = 10, n_b = 10$; SciPy baselines for Mann-Whitney, KS, and Welch	All statistic deltas were 0.0; the largest p -delta was 1.88798×10^{-5} against a 5×10^{-2} bound	Strongest numerical alignment in the run; the lightweight statistics lane remained deep inside tolerance.
OLS regression	$n = 6$ two-feature fixture; scikit-learn LinearRegression	Max coefficient delta = 6.66134×10^{-15} ; max prediction delta = 7.10543×10^{-15}	Effectively exact on the retained fixture.
Logistic regression	$n = 10$ balanced binary fixture; scikit-learn LogisticRegression	Agreement = 1.0; probability MAE = 0.0323034; probability correlation = 0.991242; convergence flag = false at 12,000 iterations	Operational pass with a retained optimizer diagnostic rather than a flattened pass-only story.

Table 11: Overview matrix for Appendix J. This table provides the fast structural read, while the grouped tables below carry the detailed numeric comparisons.

J.2 Grouped metric displays

J.2.1 Statistical comparison

The statistical fixture uses two ten-sample groups. Supporting ApexLab diagnostics include a Mann-Whitney z score of -3.74185 , Welch degrees of freedom of 16.6121, and a Cohen’s d magnitude bucket of *large*. Table 12 then keeps the row-level comparison strictly numeric, including a utilization column defined as $|\Delta|/\text{bound}$ so smaller values immediately read as stronger margin.

Check	ApexLab	Reference	$ \Delta $	Bound	Util.	Pass
MW U	0.0	0.0	0.0	1×10^{-9}	0.0	Yes
MW p	1.82672×10^{-4}	1.82672×10^{-4}	1.57×10^{-17}	5×10^{-2}	3.13×10^{-16}	Yes
KS D	1.0	1.0	0.0	1×10^{-9}	0.0	Yes
KS p	1.88798×10^{-5}	0.0	1.88798×10^{-5}	5×10^{-2}	3.78×10^{-4}	Yes
Welch t	6.8470184108	6.8470184108	0.0	1×10^{-9}	0.0	Yes
Welch p	7.54063×10^{-12}	3.20956×10^{-6}	3.20955×10^{-6}	5×10^{-2}	6.42×10^{-5}	Yes
Cohen’s d	3.062079722	3.062079722	4.44×10^{-16}	1×10^{-9}	4.44×10^{-7}	Yes

Table 12: Statistical-comparison results for the 2026-03-24 ApexLab reference-validation run. Utilization is $|\Delta|/\text{bound}$, so smaller values indicate stronger margin inside the declared threshold.

The statistical helpers are the cleanest part of the run: the core test statistics matched exactly or to machine precision on the retained fixtures, while the approximation-based p -value differences stayed comfortably inside the declared tolerance bands.

J.2.2 OLS regression

The OLS fixture uses a six-row two-feature design matrix. Table 13 keeps the bounded checks comparison-facing, Table 14 adds the ApexLab fit diagnostics, and Table 15 keeps the learned coefficients side by side.

Metric	Observed value	Bound	Util.	Pass
Maximum absolute coefficient delta	6.66134×10^{-15}	1×10^{-9}	6.66×10^{-6}	Yes
Maximum absolute prediction delta	7.10543×10^{-15}	1×10^{-9}	7.11×10^{-6}	Yes

Table 13: Bounded OLS validation checks.

ApexLab diagnostic	Value
Sample size n	6
MAE	6.80937×10^{-15}
MSE	4.75946×10^{-29}
RMSE	6.89888×10^{-15}
R^2	1.0
Converged	true

Table 14: ApexLab OLS fit diagnostics on the retained validation fixture.

Term	ApexLab coefficient	Reference coefficient
Intercept	2.4999999999999916	2.4999999999999982
Feature 1	1.7500000000000007	1.75
Feature 2	-0.40000000000000013	-0.3999999999999999

Table 15: OLS coefficients from ApexLab and the scikit-learn reference fit.

The OLS lane is effectively exact at the scale that matters here: both fitted coefficients and downstream predictions are separated from the reference fit only by floating-point-level residuals.

J.2.3 Logistic regression

The logistic fixture is balanced ($n = 10$, positive rate = 0.5). Table 16 keeps the declared checks separate from the retained optimizer diagnostic, and Table 17 adds the side-by-side coefficient comparison that was missing from the earlier draft.

Metric	Observed value	Acceptance	Distance to bound	Pass
Accuracy delta	0.0	≤ 0.05	0.05 spare	Yes
Prediction agreement	1.0	≥ 0.90	0.10 above floor	Yes
Probability MAE	0.0323034	≤ 0.12	0.0876966 spare	Yes
Probability correlation	0.991242	≥ 0.98	0.0112416 above floor	Yes
Coefficient sign match	true	true	exact	Yes
ApexLab accuracy	1.0	Informational	—	—
ApexLab pseudo- R^2	0.949313	Informational	—	—
ApexLab log loss	0.0351335	Informational	—	—
Iterations used	12,000	Informational	—	—
Convergence flag	false	Retained diagnostic	visible by design	—

Table 16: Logistic-regression validation metrics and retained diagnostics.

Term	ApexLab coefficient	Reference coefficient
Intercept	-15.512864538686669	-79.33850982424279
Feature 1	6.326241197206793	28.26949240183999
Feature 2	1.071511617730478	11.83734630563448

Table 17: Logistic coefficients from ApexLab and the scikit-learn reference fit. Magnitude mismatch is expected here because the validation frame emphasizes probability quality, agreement, and sign consistency rather than coefficient equality across optimizers.

The logistic lane also passed the declared validation frame: predictions matched the reference model perfectly on the retained fixture, probability outputs stayed well inside the MAE and correlation bounds, and coefficient directions matched. The retained non-convergence flag remains important because it preserves the optimizer diagnostic instead of flattening the result into a pass-only story.

J.3 Interpretation boundary

This validation supports the use of ApexLab as lightweight analytical infrastructure inside the Observer workflow. Its scope is reference alignment on small deterministic fixtures, giving the current lane a clear and reproducible baseline. The main empirical claims of the paper still rest on the dataset, calibration, lineage, and governance evidence summarized in the earlier appendices. Appendix J adds a different kind of confidence: the supporting statistical and regression helpers behind that analytical lane have been externally checked under declared thresholds.

That distinction is why this appendix belongs at the end of the stack. The calibration and score-surface displays in Appendix D remain the reader-facing Observer outputs that matter most empirically, while this appendix explains why the lightweight analytical surfaces sitting beside that broader lane can be treated as methodologically credible supporting infrastructure.

J.4 Bottom line

For the current Observer integration lane, the appropriate conclusion is straightforward. The 2026-03-24 ApexLab reference-validation run passed all declared capability groups. The strongest alignment appears in the statistical-comparison and OLS sections, which matched their references to effectively exact precision on the retained fixtures. The logistic section also passed and is strong enough for present use, while its non-convergence diagnostic remains an honest signal worth carrying forward into later optimization work.