information

MDPI

Article

Human Oversight During AI-Assisted Documentary Coding: An Empirical Study of Decisions, Interventions, and Traceability

Catherine Tan Lee Huan [1], Aaron Aw Teik Hong [1,*], Tekenate Ebibio Amah [2] and Anusuyah Subbarao [1]

[1] Faculty of Management, Multimedia University, Persiaran Multimedia, Cyberjaya 63100, Selangor, Malaysia; catherine.tan.lee@student.mmu.edu.my (C.T.L.H.); anusuyah.subbarao@mmu.edu.my (A.S.)
[2] Department of Computer Science, Faculty of Computing, Universiti Teknologi Malaysia (UTM), Skudai 81310, Johor, Malaysia; amatek008@gmail.com
* Correspondence: aaron.aw@mmu.edu.my

Abstract

A researcher agrees with an AI auditor that a coded category is unsupported and should be removed. The category is still present in the analysed dataset, and no record explains why. When decisions and data are recorded separately, such cases remain hidden, because neither record shows whether one reached the other. This retrospective study linked the two kinds of records in a completed project in which supervised AI systems coded 11,458 online reviews and 187 complaints about the Singapore–Malaysia confinement nanny market. Selected decisions were traced into the analysed data, and all 276 saved batch files were compared to identify changes that no decision explained. Whether a change could be traced depended on how it was made. Corrections logged by record and field were fully traceable. Recoding was not. Three recoded batches produced 109 changes, and only six of these could be linked to a decision about that record. Thirty-one saved files were also modified later, without their earlier contents being kept. Despite these gaps, none of the counts used in the project's analysis changed, although only rebuilding them afterwards revealed this. A recorded decision is not evidence that it was carried out. The study proposes records that would provide that evidence, for future projects to test.

Keywords: large language models; documentary coding; human oversight; audit trail; qualitative content analysis; traceability; research data governance

1. Introduction

Large language models (LLMs) are now used to code large collections of text. A coding instrument specifies the values that the model can assign and the evidence that each value needs, and the model is applied to thousands of documents to create a structured dataset. Performance on such tasks is often good but depends on the task and the dataset [1–3], so researchers verify the assigned values and decide on the ones that are challenged [4,5]. A ruling on a questioned value is only the start of oversight, however. Suppose an AI Auditor challenges a category given by an AI Coder, and the researcher agrees that the category should be removed. The removal must then be done in the saved output, the corrected file must be the one used for analysis, and any count using the field may be altered. All of these steps can go wrong even if the ruling is correct, and in batch coding there are more opportunities for failure. A rule clarified during production can be applied to batches coded prior to the rule. A batch coded a second time can change values nobody questioned, and a fix can be written over the file it corrects. Qualitative researchers maintain a record of

Check for updates

Academic Editor: Roser Morante

Received: 2 October 2026
Revised: 8 November 2026
Accepted: 16 November 2026
Published: 19 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

their analytical decisions [5], but a record of an approved correction is not evidence that the correction reached the stored data.

Current approaches to AI-assisted coding keep human judgement at the core. Researchers compare the values assigned by AI with human references [1,2], and they adjust coding instructions or verify outputs against the sources when the model's readings differ from the intended ones [4,6–9]. This work shows how well a model codes and how researchers maintain control of interpretation. It leaves open a narrower question that comes up once production has begun, namely whether a particular human ruling reached the data that were analysed. The records that might answer it are typically scattered. The concern and the ruling are stored in conversations or logs, and the value is stored in a saved file. The concepts of data provenance and dataset documentation provide ideas for linking such records [10–12], and this study uses these concepts for the decisions made during AI-assisted coding.

The study focuses on one completed project that coded online reviews and complaints about the Singapore–Malaysia confinement nanny market, where families in Singapore hire Malaysian nannies, either directly or via agencies, to care for the mother and baby in the weeks following birth. An AI Coder assigned the values, and two additional AI systems checked them, the Auditor against the sources and the Quality Controller against the file requirements. The Supervisor role was shared by two researchers who decided on the concerns raised. The unit of analysis is the oversight decision episode, starting with the concern and continuing through the human decision, any action taken, the saved files and the accepted data as far as the records go. These links are kept separate because a changed value in a later file does not show which decision, if any, produced it. The study started after coding was finished and both sets of data were locked, and thus is based on the records kept by the project. Its research question is as follows.

How are findings arising during AI-assisted documentary coding translated into decisions and interventions, and what can retained records establish about their consequences for the accepted data and the counts used in analysis?

The study makes two contributions. The first is methodological. The episode as the unit of analysis allows for the separation of a ruling from its implementation, and the reconstruction is a combination of episode tracing and a script-based census of saved files, which another research team can apply to its own completed project (File S5). The second is empirical. In this project, the link between decision and data held completely where corrections were logged by record and field and the files on either side of the change were retained. It broke where batches were recoded or saved files were later changed without their earlier contents being kept. That pattern identifies the records an AI-assisted coding project needs to keep.

2. Background

AI-assisted coding studies have primarily focused on the extent to which a model's values align with a reference and how researchers maintain control over interpretation during coding. After a researcher has ruled on a value during production, a further question arises, namely whether the ruling reached the data that were analysed. Answering the question draws on work beyond AI-assisted coding, on what makes human oversight effective and on the records that connect a change in data to its cause. Together these strands define the oversight decision episode used here as the unit of analysis.

2.1. AI-Assisted Coding and Human Judgement

Evaluations of LLM coding measure agreement with a reference. Gilardi et al. [1] discovered that ChatGPT performed better than crowd workers in the relevance, stance,

topic and frame annotation of tweets and news articles. Pangakis and Wolken [2] evaluated GPT-4 on 27 annotation tasks from 11 datasets against human reference values and observed that performance varied even across tasks within the same dataset. Sulaiman et al. [3] used GPT-4o mini to create a codebook from reviews of private hospitals in Selangor, Malaysia, compared human and LLM coding on a sample and then applied the codebook to the entire collection, publishing the prompts, codebook and frequency data as Supplementary Files. These studies determine the degree of similarity between AI coding and a reference. They do not follow what happens after a researcher questions a value during production.

Methodological work keeps interpretation under human control. Dunivin [4] reports researchers updating codebook definitions when an LLM's readings differ from human coding. Nguyen-Trung [6] has researchers direct ChatGPT through successive phases of thematic analysis, and Wiebe et al. [13] and Jafarian [14] place human checks at each stage of an iterative workflow. Jayawardene and Ewing [9] explain how researchers verify and adjust AI-generated codes and themes before accepting them. Bang et al. [7] compared LLM-generated quotations with the original transcripts, while Umer et al. [8] had a bilingual researcher verify the AI coding of Roman Urdu interviews and document the rationale for each correction. Wang et al. [15] introduce a verifier to score LLM labels, allowing human annotators to focus on the weakest labels. This work documents the process of interpretation and the process of revision, which is not the same type of evidence that a corrected value demands. A revised theme is supported with evidence from the text and the rationale for the new reading. A corrected value is established only when the saved output used for analysis contains it, and that second type of evidence is the subject of this study.

2.2. Oversight as the Power to Change Outcomes

Having a human reviewer in the process does not by itself make oversight effective. Green [16] examined 41 policies that require human oversight of government algorithms and asked whether humans can reliably provide the oversight that the policies require. Sterz et al. [17] argue that effective oversight requires the authority to alter the system or its effects, together with the information needed to judge the situation, and that the overseer must be able to act on a decision and must intend to do the job well. A recorded ruling shows that the overseer used the power to decide. Whether the decision took effect is shown only in the data.

2.3. Records That Link Decisions to Data

Several disciplines keep records that connect a change to its cause. Qualitative audit trails capture the rationale for methodological and analytical decisions [5], and document analysis involves analysing records in terms of their origin, purpose and context [18]. The provenance model of the World Wide Web Consortium (W3C) distinguishes entities, such as a source record or a saved dataset, from the activities that produce or modify them and the agents responsible, and names the relationships between them [10]. Documentation of a dataset captures the intent and content of a dataset and the choices made during its construction [11,12]. Regulated settings take it one step further and demand not only a version of the data, but also a history of the changes. The U.S. Food and Drug Administration and World Health Organization data-integrity guidance stipulates that changes must be attributable and that previous data must be retrievable [19,20], and 21 CFR 11.10(e) mandates secure, time-stamped audit trails that do not obscure previously recorded information [21]. These requirements do not apply to the coding project studied here, but they draw a distinction that this study relies on. A saved file contains a snapshot of the data, and a change record contains information about who changed the data and why. Research on data-versioning tools examines how datasets and pipelines evolve alongside the code that uses them [22].

There are two additional strands that relate to recoding and to prospective records. The outputs of repeated LLM classification can vary for the same input, even under settings intended to be deterministic [23,24], so a second coding run cannot be assumed to change only the value that prompted it. Lu et al. [25] present a tool that records the prompt, input and output of every unit of LLM-assisted analysis during the analysis process. If logging of that sort were to be done, it would resolve many of the questions discussed here. Most completed projects lack it, and this study asks what can be established without it.

2.4. The Oversight Decision Episode

An oversight decision episode starts with a concern about a value, a rule or a file and traces it through the connections between a decision and the data. The concern is recorded by an AI system or a researcher. The Supervisor then retains the value, approves a change or requests further verification. An approved change can be a change to a single field or a recoding of an entire batch, and its implementation is shown by a subsequent saved file or a correction entry naming the same record and field. The accepted dataset indicates which value was used in the analysis, and a calculation indicates whether the change affected a count. A changed count is only a changed result if the count is one that the analysis uses. Each link needs its own evidence, and a gap at any link leaves the later links unexplained. A changed value in a later file indicates that something has happened, but not what decision has caused it, and a ruling without a later file indicates what has been decided, but not that it has been done. One review may also include concerns for multiple fields, each of which may have a different decision, so episodes are defined by field, not by record.

3. Materials and Methods

The evidence for this study is the record of a coding project that was already completed. What that record can show depends on how it was produced during the project, by which roles and under which rules, and on how it was examined after the datasets were locked. Both are important for each subsequent finding, as is the scope of inference that a retrospective design permits.

3.1. Design and Scope of Inference

The study is a retrospective case study of one completed coding project, constructed from two enquiries. The first reconstructs selected oversight episodes from the production records. The second is a census of all saved batch files and all final-stage correction records, indicating the number of changes in the saved data that can be associated with a recorded action. The study also recalculates seven counts that the underlying project's analysis uses, five from the reviews and two from the complaints (Section 3.2), to determine whether any change to the coded data affected them. It examines whether decisions reached the data, not whether coded values were correct. Following document analysis and case-study practice, each conclusion is tied to the records that support it [18,26]. A connection between a decision and a change is only asserted if the records connect the same record and field, and if they do not, the connection is reported as unresolved.

3.2. Setting and Documentary Collections

The underlying research project collected reviews and complaints about the Singapore–Malaysia confinement nanny market. Families in Singapore engage Malaysian confinement nannies, directly or through employment agencies, to help with the mother's recovery and the baby's care, usually in the family's home, for a short period after birth. The documents tell of service experiences, employment arrangements and reported problems, and this study looks at the records of how they were coded, not the experiences they tell of. The

Agency Google Reviews (AGR) collection contains 11,458 reviews from 15 agency profiles, which are anonymised in the coded data. The reviews were organised into 230 batches, 229 of 50 records and one of eight. The accepted AGR dataset has 30 columns. Five are pre-filled from the source (batch, agency label, record identifier, year and star rating), 24 are primary coding fields in which the Coder applies the coding rules, and one holds notes. A *field* is a column, and a *value* is what a field holds for one record. There are some fields that contain one value. *Multi-tag fields* may contain multiple categories, each separated by a comma, and the coding instructions indicate when a field is empty.

The Community Complaints (CC) collection contains 187 complaint and warning records from publicly accessible social-media posts, forums, blogs and news sources, including both agency-mediated and direct-hire arrangements. They are divided into 19 batches, 18 of ten records and one of seven. The accepted CC dataset has 15 columns, comprising four pre-filled columns (record identifier, year, engagement and country), ten primary coding fields and one coder-notes column. Review identifiers start with a letter, such as A_0002, and complaint identifiers start with CR_, such as CR_060. The researchers collected the complaints from these sources with a project tool that also managed duplicate entries. For each complaint, the tool stored a short summary and a list of key points, which helped the Coder find the relevant passages. Coding values were assigned from the original text in the content field. The key points were only a preliminary list for the complaint-category field nanny_service_complaint_categories. The Coder retained a listed category only when the text supported it, and added any category the text supported that was not listed. The one exception was a record with minimal text, which the instrument allowed to be coded from its title and key points, with a note in coder_notes saying so (File S2). Both collections have a stated cutoff of 31 March 2026. They were built with different criteria and are not samples of one population, and their contents are accounts written by others, including unverified allegations.

Seven counts from the two accepted datasets are used in the underlying project's analysis. Five are counts of reviews, namely those containing a complaint (complaint_review = 1), the five-star reviews among them, those containing both a complaint and an endorsement, the one- or two-star reviews with endorsement evidence, and those describing an incentive for a positive review (nanny_incentivised = 1). The other two are counts of complaints coded with one of two values in the fraud field fraud_typology. The value Concurrent Booking is used when a nanny or agency has taken bookings or deposits from multiple families for the same or overlapping dates, and Unauthorized Swap is used when a different nanny from the one agreed was sent without the family's consent. Complaint CR_060, the only accepted complaint coded from its title and key points, is not in either count.

3.3. Roles and Production Procedure

Three AI systems and one human role divided the production work (Table 1). The Coder coded each batch according to the Coding Instructions and the approved rules. The Auditor checked selected records, first coding each from the source text and the applicable rules and then comparing its values with the Coder's output. The Quality Controller checked saved files for structural faults and impermissible values, so an invalid interpretation stored in a valid row and a valid interpretation stored in a malformed row fall to different checks. A warning from either checking system was a concern for the Supervisor to consider, not a confirmed error. The Supervisor role was shared by two researchers. The first made most rulings after consulting the second, and both participated in coding and checking both collections. The Supervisor was the final authority on disputed interpretations, rule changes, corrections and the selection of accepted outputs, and only the Supervisor could initiate a coding or recoding run by giving the command to the Coder. When a finding

required a decision, the Supervisor responded in writing and treated a finding without a response as acceptance of the value as coded. Those acceptances were not recorded on a case-by-case basis.

The public coding instruments (Files S1 and S2) give the field definitions and coding criteria the Coder followed, and Figure 1 shows the materials each role received from the Supervisor and what it returned.

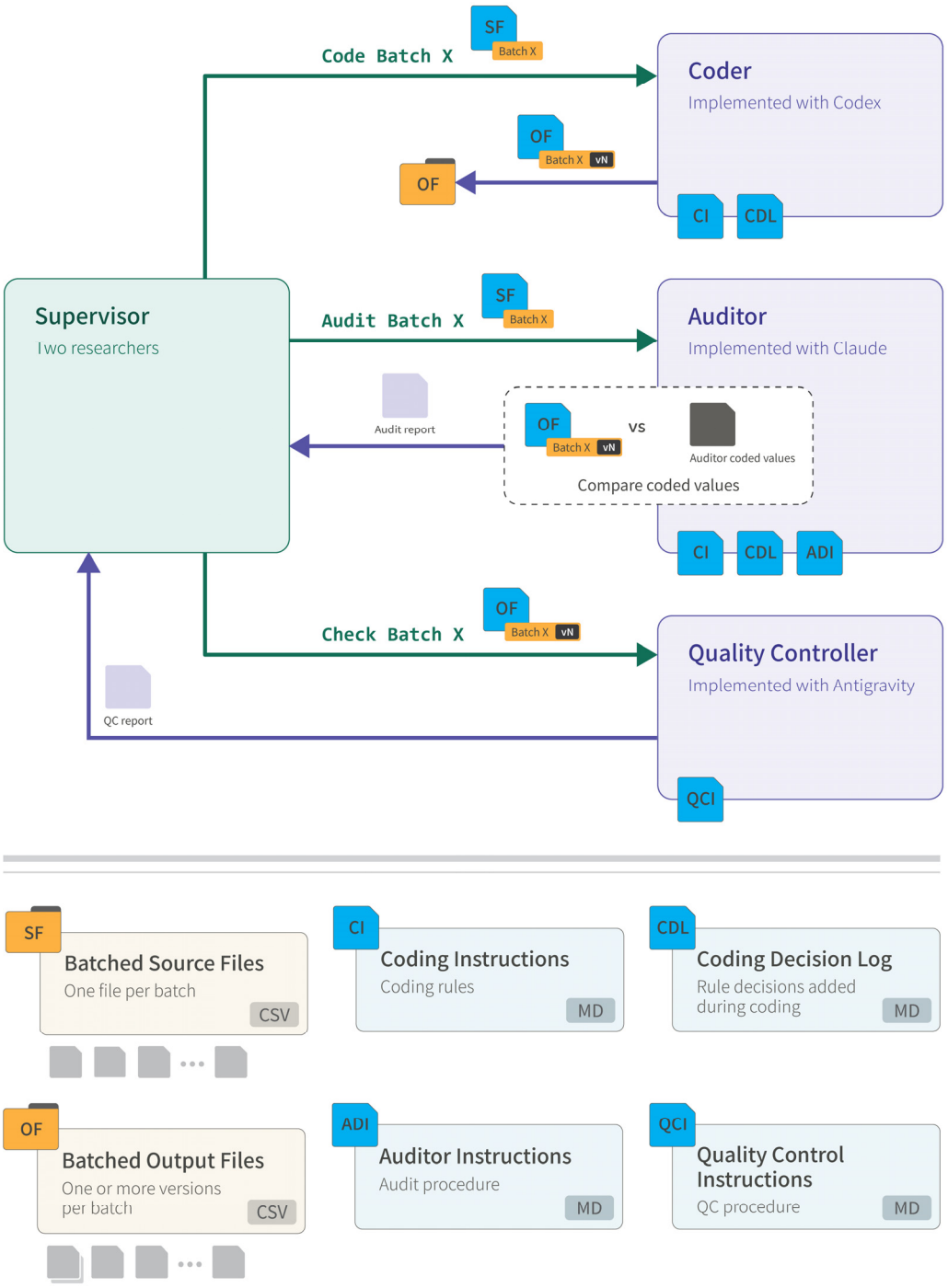


Figure 1. Coding, audit and quality-control procedures set out in the instructions for each role.

Table 1. Production roles, tools and the dates the records establish.

Role	Tool or Researchers	Work Performed	Dates Established for the Role
Coder	Codex, ChatGPT GPT-5.5 (OpenAI)	Assigned values to source batches and saved coded outputs	AGR coding from 28 May to 2 June 2026. CC coding on 8 June 2026
Auditor	Claude, Claude Sonnet 4.6 (Anthropic)	Coded selected records from the source and rules, then compared its values with the Coder's	CC checks on 8 and 19 June 2026. Complete range not recorded
Quality Controller	Antigravity, Gemini 3.1 Pro (Google)	Checked saved files for structural faults and impermissible values	Complete dates of use not recorded
Supervisor	Two researchers holding one decision-making role	Ruled on concerns, approved or rejected changes, clarified rules, started coding runs and accepted outputs	Throughout production. Dates of individual rulings not recorded

3.4. Coding Rules, File Versions and Accepted Datasets

Each collection had its own Coding Instructions and Coding Decision Log (CDL). The instructions defined the fields, permitted values and evidence required for each category, and the CDL recorded the rules and clarifications the Supervisor adopted during the work. A CDL entry is a record of a rule being approved, and the impact of that on records coded prior to the CDL entry is a different question. There are 21 decisions in the AGR CDL. Decisions 1 to 15 were made during a pre-production trial of the Coding Instructions, and Decisions 16 to 21 were made during the audit of AGR Batch 001, each naming the record that prompted the decision. The Coder's progress log records the number of CDL entries read for each coding run. The first run of Batch 001 read none of the 15 entries that existed at the time, because the session instruction told the Coder to read entries labelled as "rules" while the CDL labelled its entries as "decisions". The Supervisor corrected the instruction and had Batch 001 recoded, and the second run and all subsequent AGR runs recorded in the progress log read all 21 decisions.

The CC CDL has eight decisions. The first six were carried over from the AGR work, and Decisions 7 and 8 were added during complaint coding, so Batches 01 to 05 were coded with six decisions, Batch 06 with seven and Batches 07 to 19 with eight. Decisions 7 and 8 both concern complaints in which a nanny or agency received a deposit, and the nanny never arrived to start work. Because no service period began, the timing field review_timing, which indicates whether a complaint was written during or after service, had no obvious value for such a complaint. CC Decision 7 set it to after_service whenever the family wrote after the booking failed. It was declared applicable from Batch 01, although Batches 01 to 05 had been coded before it existed. CC Decision 8 set the threshold for the fraud field, under which an accepted deposit followed by non-arrival is coded No-Show whether or not a refund was requested. Coded outputs were saved by batch. A recode created a new saved version of the batch, while the *Apply Fixes* operation could change the latest file without saving a new one. At finalisation, the chosen batch versions were merged, checked and locked on 19 July 2026. The accepted AGR file contains 11,458 records and was created from 214 first-version and 16 second-version batch files. The accepted CC file has 187 records and was built from eight first-version and 11 second-version files. The lock record gives each accepted file's dimensions and SHA-256 hash, which identify the accepted versions (Table A11).

3.5. Episode Inventory and Reconstruction

There are 15 entries in the case inventory (Table 2). Fourteen (E01 to E14) were prepared from the remaining records during the reconstruction, based on Auditor–Supervisor

conversations, audit and Quality Controller reports, CDL entries, correction logs and comparisons of saved outputs. The logged correction to Review A_5016 was discovered to alter a calculated count, so the fifteenth, E15, was added. No complete register of production decisions exists, so the inventory is the set of cases that the surviving records allow to be followed, not a sample drawn from a larger list. Nine entries relate to individual records, seven of which are from production and two from the later pilot, and Appendix B provides evidence for these. The remaining six relate to aggregate issues and are reported with the results they relate to.

Table 2. Case inventory.

Entry	Subject	Reported in
E01	Review `A_0033`, removal of `emotional_support` and two further categories	Section 4.3, Appendix B
E02	Review `A_0047`, timing ruling and removal of `parent_training`	Section 4.2, Appendix B
E03	Review `A_0002`, timing ruling that kept the coded value	Section 4.2, Appendix B
E04	Pilot, Review `A_0011`, category omitted by all four outputs	Section 4.7, Appendix B
E05	Pilot, Review `A_0033`, threshold for recording mixed language	Appendix B
E06	Design of the production audit sample	Section 3.7
E07	Reconstructed review category counts	Appendix C.1, Table S2
E08	Complaint `CR_052`, fraud value accepted as `No-Show`	Section 4.3, Appendix B
E09	Complaint `CR_007`, permitted values of `first_time_parent`	Appendix B
E10	Complaint `CR_074`, formal escalation	Appendix B
E11	CC Batch 08, Auditor conversations and aggregate log disagree	Section 4.1, Appendix C.1
E12	Reconstructed complaint category counts	Appendix C.1, Table S2
E13	Format checks on the pilot outputs	Not reported separately
E14	Human coding in the pilot	Sections 3.8 and 4.7, Appendix C.4
E15	Review `A_5016`, removal of `experience` after warning W02	Sections 4.3 and 4.6, Appendix B

Each episode was reconstructed from the source text and the concern, then the human ruling, any authorised action, later saved files and the accepted dataset, and Table 3 lists what each type of record can establish on its own. Records were joined using the record_id, which is the same across audit material, rule records, batch versions and accepted datasets. Values were compared field by field, and each category in a multi-tag field was considered separately, so a change from baby_care, mother_care to mother_care removes one category. No gap in the records was filled by assuming that an undocumented correction had or had not been made.

Table 3. Evidence used to reconstruct oversight decisions and the limit of each source.

Evidence Source	What It Can Establish	What It Cannot Establish Alone
Original source and Coding Instructions	What the document said and how the field or category was defined	Whether a concern was raised and how the Supervisor responded
Auditor–Supervisor conversation, audit report or Quality Controller warning	What was questioned and, where recorded, the human ruling	Whether a proposed change was made in the saved data
CDL and correction authorisation	The rule or action approved, and any stated scope	Whether every affected record was checked or changed
Correction record and saved batch versions	Recorded actions and the values present in particular files	Why every difference occurred, or who made an undocumented change
Accepted datasets, batch selection and lock records	Which saved values and file versions were selected for analysis	Whether a value was correct, or how each earlier change occurred
Calculations and the counts used in analysis	Whether a change affected a specified count, and whether the analysis uses that count	That a particular ruling caused a changed count, without a supported connection

Each link in an episode is given one of five evidence statuses.

- **Documented.** A surviving record directly supports the decision, action or file state.
- **Partly documented.** Some steps have direct support, but a material connection is missing.
- **Reconstructed.** The connection was pieced together after production by matching individual records that have survived.
- **Conflicted.** Two relevant sources disagree on a material value, date or event.
- **Unresolved.** The available evidence does not show what happened at that step.

3.6. Census of Saved Versions and Correction Records

Two full sets of retained records were included in the census. The first was all 276 saved batch files, with each of the 27 batches that had a second version compared to its first. The second was each final-stage correction-log entry, compared to each cell that is different between the combined files saved prior to final corrections and the accepted files. Concerns and decisions that did not appear in either set are not included in the census. After the lock, Codex (ChatGPT GPT-5.5, OpenAI) carried out the census under written instructions, writing and running Python scripts on a read-only copy of the project archive, using the Python runtime bundled with Codex (Python 3.12). The census results are generated by that code, and the scripts, when run on the same archive, reproduce the census results. There were parser errors in two preliminary runs. One parser did not capture all record-identifier formats in the correction logs, and the other parser did not read dates in the Decision Logs. Both errors were identified by comparing the output to the project's validation report, and the corrected code was used to generate the results presented here.

Files were matched on record_id and compared field by field. Multi-tag fields were compared as sets, meaning that a reordering of the same categories is not a change. A changed single-value field is one change, and each category added to or removed from a multi-tag field is one change, so the number of changed cells and the number of individual changes are not the same. Empty cells, whitespace-only cells and cells containing the literal value 0 were distinguished, since they have different meanings in the coding rules. Each change between saved batch versions was then grouped by the strongest record that could explain it. The tests were applied in a fixed order, and a change was assigned to the first group whose test it met. A change was *named for its record* when an audit report, Auditor conversation, correction log or CDL example identified that record and category, and, for a single-value field, the new value. A change was *rule-addressed* if a CDL entry directed the Coder to use, restrict or exclude that category, or defined the value of that single-value field, without naming the specific change. All other changes were classed as *not addressed by a recorded rule*, which simply means that no rule entry matched. For rule-addressed changes under a rule that points in one direction, the direction of the change was compared with the rule's. Label-standardisation rules are directionless and parent_training was handled separately, as it was introduced in AGR Decision 7 and restricted in AGR Decision 18. The rule map is given in Appendix A.

The creation and modification times were used to detect saved files that were modified after they were created. The script flagged any file that was last modified more than five minutes after it was created. The gaps between creation and last modification fall into two well-separated groups. Any threshold in the range between the longest gap (3.87 min) among the 245 files not flagged and the shortest gap (39.72 min) among the 31 flagged files will flag the same files. A timestamp only indicates that a file has been modified, not by whom or which values changed. In the final correction stage, the correction entries were matched to the changed cells by record and field in both directions, so that the census could identify changed cells without an entry and entries without a changed cell, and a separate whitespace entry and notes on file repairs were kept apart from the main entries. The seven

counts used in the underlying project's analysis were recalculated on the combined files before final corrections and on the accepted files. In a third state, the accepted versions of the three batches recoded during production were replaced with their earliest retained versions, and every other batch selection was kept. This checks if those recodes shifted the seven counts under the retained files. It does not restore the first-run state of a batch whose first-version file was subsequently changed.

The two researchers checked 37 census results by hand against the original files. The sample was purposive, with the aim of ensuring that each type of census result was tested at least once and that the locations where the parsers failed were tested again. It included all seven recalculated counts, ten of the 109 recode changes (covering all three recoded batches and every group) and 16 cells changed at the final stage. Of those, six were in the reviews, including Review A_5016 and records the faulty parser had misread, and ten were in the complaints. Four audit recommendations from both collections were also included in the sample, two of which were identified by the census in the accepted data and two of which were not identified by the census. A check sheet was used to record expected values prior to opening files. Thirty-six of the 37 items agreed, and Table S3 shows all items and the census result and what the hand check found. For Complaint CR_072, the census reported the recommended categories as missing, but they are present in the accepted file, and the corrected result is reported. Because the sample was selected for coverage, it does not provide an estimate of the error rate for the census.

3.7. Production Audit Records

Selected records were examined by the Auditor during production. The first review audits were conducted as coding was done, and later audited batches were selected at intervals across the middle and late stages of production and at each transition from one agency's reviews to the next. Within an audited batch, the Auditor Instructions required records to be drawn at random, and the audit reports record the random seed used. Every complaint batch was audited. Each check of a record was entered in an aggregate audit log, so a record checked twice, for example, after its batch was recoded, was counted twice. The logs record a *discrepancy* when the Auditor counted at least one error in that check. The counting conventions changed during production. The Auditor could bring up a borderline case with the Supervisor without it being considered a discrepancy. Partway through production, the Auditor and the Supervisor also agreed that a category the Auditor determined to be unsupported in nanny_service_endorsement_categories, the field recording which aspects of the nanny's work a reviewer praised, would be noted and not counted. Some counts were also reduced due to the Supervisor's disagreement with a finding. A counted discrepancy is therefore the Auditor's judgement under the conventions of that check, not an independently verified error, and the audit counts cannot estimate an error rate for either collection. Checking occasions, fields and categories within multi-tag fields are reported as separate units throughout.

3.8. Post-Production Pilot

A separate pilot was conducted after both datasets were locked. Its records were the first production batch of each collection, the 50 reviews of AGR Batch 001 and the ten complaints of CC Batch 01. These records were coded independently by each of the two researchers who had served as Supervisor, and the Coder and Auditor generated separate outputs for the same records. The researchers then went back to the sources and agreed on reference values. The comparison includes the primary coding fields and excludes the pre-filled source fields and notes, resulting in 24 fields per review and ten per complaint, or 1200 AGR and 100 CC field values. Each category in multi-tag fields was also compared.

Because the same researchers coded the records and set the reference, the pilot measures agreement within one team. It did not change either locked dataset or add a production rule.

3.9. Handling of Identifying Information

The CC records came from open or publicly accessible sources. Closed groups and forums with restricted access were excluded. Source accounts could include names, contact details and allegations about people or organisations. Some of that identifying material was submitted to commercial AI services during preparation and production without every identifier being removed beforehand. Original record identifiers were kept internally for comparison, and identifying text not needed for the analysis is not reported.

4. Results

Across the episodes and the census, the link between a human decision and the accepted data held completely at the final correction stage and broke most often during production recoding (Table 4). The records establish each link to a different extent. The units used in the rows of Table 4 are thus different, and the rows indicate the extent of the evidence at each link without providing a sequence of totals.

Table 4. How far the evidence reaches at each link of an oversight episode.

Link	What the Records Establish	Unit
1. Concern recorded	72 checks with a counted discrepancy (AGR 57, CC 15). 78 category findings (AGR 58, CC 20)	Checking occasions. Categories
2. Decision recorded	29 rule decisions in the two CDLs, 8 of them added during production in response to a named record. Individual rulings survive only where the Supervisor replied in writing	CDL entries
3. Change linked to a decision	Final stage, 110 of 110 changed cells match a log entry. Production recodes, 6 of 109 changes named for their record	Changed cells. Individual changes
4. Change present in accepted data	62 of 78 proposed audit outcomes present	Categories
5. Calculated count changed	At the final stage, experience 4 → 3, `premium_standard` 288 → 287 and `good_matching` 502 → 504	Counts
6. Count used in analysis changed	0 of 7	Counts used in the underlying analysis

4.1. Concerns Raised During Production

The audit counts record where the Auditor looked and what it questioned (Table 5). In the reviews, 50 of the 58 category findings were in categories that the Auditor determined were not supported, and 47 of the 58 were in the nanny-service endorsement field. That concentration was met with one decision by the Supervisor for the entire field, and that decision was not to recode the field. No count in the underlying project's analysis is based on that field. In the complaints, the direction was reversed, and all 20 findings were for categories the Auditor considered to be missing from the complaint-category field. No category finding was made from AGR Batch 001, whose concerns were addressed in the Auditor conversation reported in Section 4.2.

Table 5. Concerns raised by the Auditor during production.

Measure	AGR	CC
Checking occasions	195	37
Checking occasions with at least one counted discrepancy	57	15
Category findings	58	20
Marked `overcoded` (category judged unsupported)	50	0
Marked `undercoded` (category judged missing)	8	20
Category findings in `nanny_service_endorsement_categories`	47	0
Records with a category finding	48	14
Batches with a category finding	21	Not recorded

Note. A checking occasion is a single check of a single record, thus a record checked twice is counted twice. A discrepancy was counted if the Auditor found at least one error in a check, according to the conventions outlined in Section 3.7. The CC log omits a check of two complaints in Batch 08 that an Auditor conversation describes (E11). Sources are Appendix C.1 and Table S2.

4.2. Supervisor Decisions

The Auditor conversation for the first review batch retains four rulings on three reviews (Table 6). Two rejected the Auditor's suggestion and retained the coded value. The proposal to change Review A_0002 from unknown to after_service was rejected because a review written in the past tense does not indicate when it was written. In Review A_0047, the Supervisor maintained the timing value during_service as the review showed that the nanny had not yet departed, and in the same review, the Supervisor accepted the concern about parent_training and clarified the rule in Decision 18. The fourth ruling accepted the concern about emotional_support in Review A_0033 and clarified the rule in Decision 17. Apart from these conversations, no register of individual rulings was kept. The two CDLs hold 29 rule decisions, eight of which were added during production in response to a named record, and because the Supervisor treated silence as acceptance, the number of findings accepted without a reply cannot be recovered.

Table 6. Rulings preserved from AGR Batch 001.

Record	Field or Category	Auditor's Proposal	Ruling	Accepted Value
`A_0002`	`review_timing`	`unknown` → `after_service`	Rejected	unknown
`A_0047`	`review_timing`	`during_service` → `after_service`	Rejected	`during_service`
`A_0047`	`parent_training`	Remove category	Accepted, Decision 18	Removed
`A_0033`	`emotional_support`	Remove category	Accepted, Decision 17	Removed

Note. Each row is one ruling on one field or category of one record. The review_timing field can have the values during_service, after_service and unknown. Sources are the Auditor–Supervisor conversation for Batch 001 and Appendix B.

4.3. From Ruling to Accepted Data

Two episodes illustrate the entire process from concern to accepted value (Table 7a). In Review A_0033, the Supervisor accepted the concern about emotional_support and clarified the rule in Decision 17, and neither Batch 001 version 2 nor the locked dataset contains the category. In Review A_5016, the Quality Controller warning W02 questioned experience under Decision 3, which allows the category only when a reviewer gives the nanny's experience as a reason for hiring her. The final correction log authorised its removal, and neither the revised batch nor the accepted dataset contains it. The same comparisons also show breaks in both directions. The recode that removed emotional_support from A_0033 also removed baby_care and mother_care. Restrictive rules apply to both categories (Decisions 1 and 16), but no ruling names them for this record, so the recode changed more than the ruling named. In Review A_0177, the break is the reverse. During the audit

conversation, the Supervisor agreed to remove good_matching, and the accepted data still includes it, but there is no record of why.

Table 7. From ruling to accepted value.

(**a**) **Selected Episodes**

Case	Saved Values	Status
`A_0033`, `emotional_support` removed under Decision 17	Batch 001 v1 `baby_care`, `mother_care`, `parent_training`, `cooking_standard`, `emotional_support`. v2 and `locked` `parent_training`, `cooking_standard`. One of three removals named in the ruling	Documented ruling. Two removals rule-addressed only
`A_5016`, `experience` removed after warning W02	Batch 101 v1 `baby_care`, `experience`, `mother_care`. v2 and accepted `baby_care`, `mother_care`	Documented
`A_0177`, agreed removal of `good_matching`	Category still present in accepted data	Agreed change not implemented
`CR_052`, `None` → `No-Show` in fraud_typology	`No-Show` in both saved versions and accepted data. `None` in no retained file	Conflicted

(**b**) **Audit Findings Compared with Accepted Data**

Outcome	AGR	CC	Total
Proposed outcome present	45	17	62
Proposed outcome absent	13	3	16
Absent, covered by the field-level decision not to recode	6	0	6
Absent, agreed change not implemented	1	0	1
Absent, no recorded response	6	3	9

Note. Panel (b) counts categories, each one category in one field of one record. A proposed outcome is present when a category the Auditor judged unsupported is not present in the accepted value, or a category the Auditor judged missing is present. A present outcome does not indicate a ruling, but rather that the accepted data are consistent with the proposal. Sources are Appendices B and C.2.

Complaint CR_052 demonstrates a third pattern, where the ruling and the accepted value are the same, but the history between them is missing. The Auditor reported the fraud value as None for a complaint that described a deposit followed by the nanny's non-arrival, the Supervisor accepted No-Show, and Decision 8 clarified the rule. Both saved versions of CC Batch 06 already contain No-Show, and the only difference between them is in coder_notes. The first version was modified about four hours after its creation, on the day of the audit. The instructions allowed a fix to be written into an existing file, which would result in this pattern, but no record of a fix for CR_052 survives, so the episode is classified as conflicted. These cases are expanded to the entire audit record (Table 7b) by comparing each itemised audit finding with the accepted data. Of the 78 category findings, 62 are the same as the outcome the Auditor proposed. Seven of the other 16 have a recorded explanation. Six fall under the Supervisor's decision not to recode the nanny-service endorsement field, and one is the unimplemented removal in A_0177. The other nine (six in the reviews and three from the last complaint batch) have no recorded response.

4.4. Matching Final-Stage Corrections to Changed Data

The chain was fully maintained at the final correction stage (Table 8). All cells that are different between the combined files saved prior to final corrections and the accepted files correspond to a correction-log entry for the same record and field in both collections. The match was also performed from log entries to cells. The seven logged pairs that show

no difference between the two files come primarily from the intermediate repair of Review A_0982, a malformed row whose three changed values each match an entry. At this stage, no complaint category changed, as all 30 complaint changes were to coding notes.

Table 8. Final-stage corrections.

Measure	AGR	CC
Records changed	59	30
Cells changed	80	30
Changed cells matched to a log entry for the same record and field	80 of 80	30 of 30
Analytical cells	79	0
Structural or administrative cells	1	0
coder_notes cells	0	30
Main correction-log entries	86	30
Distinct record-field pairs in the logs	87	30
Logged pairs with no difference between the two files	7	0
CSV rows given missing trailing empty fields (file-format repair)	818	

Note. A cell is a single field of a single record, and a logged pair is a unique record and field identified in the correction logs. The AGR log also contains a note about the structure of the malformed row of Review A_0982 and a separate entry on whitespace, but these are not counted among the main entries. Log entries are undated, so a value that was changed and then restored between the two saved files leaves no difference and is not counted. Source is Appendix C.3.

4.5. Recoding and Later Changes to Saved Files

The most frequent break in the chain was production recoding. The Supervisor had three review batches coded a second time, with the second runs differing from the first in 76 cells, representing 109 individual category or value changes (Table 9a). Six of those changes were named for their record, each in the CDL entry that cited that review as its example. Another 71 involved a category or value that a recorded rule addresses, and 32 involved none. The changes in the two larger recodes are in opposite directions (Table 9b). In Batch 001, 36 of the 38 changes governed by a one-directional rule went in the rule's direction, and all 36 removed restricted categories. This is consistent with the rule states, as the first run read no CDL entries, and the second read all 21. Batch 029 was the opposite. Fourteen of its 15 changes that were governed by a one-directional rule added categories the rules restrict, eight of them mother_care and six baby_care, and no recorded rule accounts for those additions. The reason for recoding Batch 001 is recorded, but no reason was recorded for the Supervisor's requests to recode Batches 005 and 029, and the rule state of the first run of Batch 029 is missing from the progress log (Appendix A.2). With the exception of a subsequent review of a sample of Batch 001 records, no retained record indicates that the changed values were reviewed prior to acceptance of the recoded files.

A separate loss occurred due to changes made to existing files. Production instructions resulted in a new version of the file for each recode, but a fix could be written into the existing file. Table 10 indicates that 31 of the 276 batch files saved were modified long after they were created, and their earlier contents were not retained. The majority of these changes occur within the same three days, over a week after the batches were coded. The loss extends to the comparisons themselves, as the first saved version of Batch 001 was changed after the second version was created.

No recorded recheck of earlier batches followed the rules added during complaint coding. CC Decision 7 was declared applicable from Batch 01, and no saved file indicates that Batches 01 to 05 were rechecked. Those complaints that are already coded as No-Show in those batches comply with it. All three have the timing value after_service that Decision 7 requires, and none has the service category without the fraud category No-Show under Decision 8. This comparison cannot identify a complaint that mentioned a deposit and non-arrival but did not get either category. The fraud labels Concurrent Booking and

Unauthorized Swap already appear in the first saved complaint batch, which was never modified, and a check of all 187 complaints against their sources for those two labels after the lock found no further qualifying complaint.

Table 9. Changes made by the three production recodes.

(**a**) **Changes by Batch**					
Batch	**Changed Cells**	**Individual Changes**	**Named for Record**	**Rule-Addressed**	**No Recorded Rule**
001	44	70	6	52	12
005	4	4	0	0	4
029	28	35	0	19	16
Total	76	109	6	71	32

(**b**) **Rule-Addressed Changes by Direction**		
Direction	**Batch 001**	**Batch 029**
In the rule's direction	36	1
Against the rule's direction	2	14
`parent_training` (AGR Decision 7 introduces, AGR Decision 18 restricts)	11	4
Label or value rules, no direction	3	0

Note. A changed cell is one field of one record. An individual change is one changed value in a single-value field, or one category added to or removed from a multi-tag field, so one cell may contain multiple changes. The groups in panel (a) are defined in Section 3.6. Only Batches 001 and 029 are covered in panel (b), as Batch 005 did not have any rule-addressed changes. A change is against a rule's direction if, for instance, it adds a category that the rule restricts. A match with a rule does not imply that the rule is responsible for the change. Source is Appendix A.3.

Table 10. Saved batch files and later modifications.

Measure	**AGR**	**CC**	**Total**
Saved batch files	246	30	276
Batches with a second version	16	11	27
Production recodes	3	0	3
Final-stage second versions	13	11	24
Final-stage second versions differing only in CSV formatting	4		4
Files modified more than five minutes after creation	21	10	31
Modified about 40 min to four hours after creation			3
Modified 11 days or more after creation			28
Of those, modified within the same three days, more than a week after coding			27

Note. Each count is of saved batch files. The last three rows break down the 31 flagged files and are not split by collection. Any threshold between the longest gap between creation and modification of unflagged files (3.87 min) and the shortest gap between creation and modification of flagged files (39.72 min) will flag the same 31 files. A timestamp indicates that a file has been modified, but not by whom or which values changed. Source is Appendix C.5.

4.6. Counts Used in the Analysis

There were no changes in the seven counts used in the underlying project's analysis (Table 11). They were the same before and after final corrections, and again when the three recoded batches were substituted by their earliest retained versions. Other calculated counts did change. The number of eligible reviews with the category of experience dropped from four to three, and the only record whose removal accounts for the drop is Review A_5016. The count of good_matching increased from 502 to 504 at the final stage and was 503 when the recoded batches were restored, while the number of premium_standard = 1 decreased from 288 to 287 at the final stage.

Table 11. Counts under three dataset states.

Count	Before Final Corrections	Accepted	Recoded Batches Restored
Counts used in the underlying analysis			
AGR reviews containing a complaint (`complaint_review = 1`)	725	725	725
Of those, five-star reviews	309	309	309
AGR reviews containing both a complaint and an endorsement	509	509	509
One- or two-star AGR reviews with endorsement evidence	48	48	48
AGR reviews describing an incentive for a positive review (`nanny_incentivised = 1`)	19	19	19
CC complaints coded `Concurrent Booking` in `fraud_typology`	9	9	9
CC complaints coded `Unauthorized Swap` in `fraud_typology`	6	6	6
Other calculated counts			
`premium_standard = 1`	288	287	
`good_matching`, of 1702 reviews with `agency_service_endorsement = 1`	502	504	503
`experience`, of 10,814 reviews with `nanny_service_endorsement = 1`	4	3	

Note. Each count is the number of records that satisfy the condition in the first column. The three columns are three versions of the data. "Before final corrections" is the combined file saved immediately before the final correction stage. "Accepted" is the locked file used for analysis. "Recoded batches restored" is the accepted file with AGR Batches 001, 005 and 029 replaced by the earliest saved versions of those batches. This column may not reflect the original first run of Batch 001, as the first saved version of Batch 001 was modified after the second version of the batch was created. Blank cells indicate that the count was not computed for that version. Source is Section 3.6.

4.7. The Post-Production Pilot

Agreement among the pilot outputs did not guarantee a correct reading (Table 12). In Review A_0011, all four outputs failed to include night_duty, the category for a nanny who cares for the baby at night while the mother sleeps, and only a return to the source added it to the pilot reference. For 21 review values and one complaint value, the two AI outputs were in agreement with each other and different from the reference. The production data already had night_duty for A_0011, as the recoded Batch 001 had added it, so the pilot changed its own reference and left the locked dataset unchanged.

Table 12. Agreement in the post-production pilot.

Measure	AGR	CC
Records	50	10
Primary field values compared	1200	100
Researcher 1 and Researcher 2 agree	1173	86
Coder and Auditor agree	1149	83
Field values where the four outputs do not all agree	122	36
AI outputs agree with each other but differ from the reference	21	1

Note. Each count is of field values, one primary coding field for one record, giving 24 per review and ten per complaint. The four outputs are the two researchers' independent codings, the Coder's output and the Auditor's output. The two researchers agreed on the reference values afterwards, after returning to the sources. Category-level comparisons are in Appendix C.4.

5. Discussion

The findings suggest one answer to the question of where oversight could and could not be traced. At one point in the project, the same Supervisor and the same rules yielded a full trail, and at another point, a weak trail, and the difference lay in how changes were recorded. That explanation has implications for the documentation of oversight and for what can be assumed to have been accomplished by a recode or a correction. It also explains

what a retrospective reconstruction can contribute to prospective logging and what records a future project requires.

5.1. Traceability Depends on How Implementation Is Recorded

At the final correction stage, each correction was an entry naming a record and a field, and the files on both sides of the stage were kept. Without the previous file, a log could not show that an entry was carried out, and two files without a log could show that a value was changed but not on whose authority. With both, each modification of the accepted data is associated with a recorded instruction. Production recoding kept the files but had no entry for each change, and saved files that were subsequently changed did not retain an earlier version. Complaint CR_052, which the study classifies as conflicted, shows the cost, since the value the Auditor reported is in no retained file. Regulated electronic records draw the same distinction between retaining previous data and attributing each change [19–21], and provenance models draw it between an entity and the activity that created it [10]. A documentary coding project is not bound by those requirements. In this project, oversight could be traced where a saved version was paired with a change entry, and either one without the other left the decision and the data unconnected.

5.2. A Decision to Keep a Value Is an Outcome of Oversight

Of the four rulings that were saved from the first batch of reviews, two maintained the value that the Auditor challenged. Such a decision leaves nothing in the data, and the accepted dataset cannot distinguish between a value that the Supervisor has looked at and accepted and a value that no one has questioned. This gap was expanded by the working arrangement, as explicit rulings were only recorded when a finding required a decision and silent acceptances were never recorded. Methodological work on LLM coding describes researchers adapting instructions when the model's readings do not align with the intended definitions [4,6]. The cases here add that a decision not to change a value is also an outcome of oversight. In the terms of Sterz et al. [17], it is an exercise of the power to decide, and only a written entry shows that the power was exercised.

5.3. A Recode Is a New Coding of the Batch

A recode applies the entire instruction set to the entire batch. It thus alters what the Coder now reads differently, not only the value that caused the run, and the direction of those changes does not have to be according to the rules. Batch 001 moved towards the rules, as its rule states predict. Batch 029 went against them and added categories that the rules restrict, and there is no recorded rule that accounts for the additions. The results of repeated LLM classification can vary for the same input [23,24], so a second run is better understood as a new coding of the batch than as a correction of it. The practical implication is that a recoded batch must be compared with the batch it replaces in its entirety before it is accepted, and each change must be considered as authorised or as requiring review. The comparison rules in the census (Section 3.6) could serve that purpose during production. In this project, no such comparison is recorded before acceptance, except for a subsequent examination of a sample of Batch 001 records, and the complete comparison was performed by the census after the lock, when the changed values could be described but not examined before use.

5.4. Changed Data and the Counts Used in Analysis

A correction can change a calculated count without changing any count the analysis uses, and the two claims need different evidence. The removal of experience from Review A_5016 affected a category count and none of the seven counts used in the underlying

analysis. Establishing the second result required rebuilding those seven counts after the lock, since there was no register that associated them with a calculation and a dataset version. Dataset documentation already records how a dataset was created and what it is used for [11,12]. A register of the counts used in analysis, each with its calculation and dataset version, would extend that documentation to the point where data become claims and would make this test routine.

5.5. Agreement and the Return to the Source

Agreement among coders says little about whether a category was correctly applied when the coders share an omission. In the pilot, all four outputs omitted night_duty from Review A_0011, and it was only by going back to the source that it was found. A shared omission results in neither disagreement nor low confidence, which are the bases for verification designs that route weak labels to human review [15]. The production audits had the same limit for errors common to the Coder and the Auditor, because after the first audit the reports listed differences and named the sampled records that had none. Checks that return to the source on records chosen without regard to agreement are needed to find omissions every coder shares.

5.6. What a Retrospective Reconstruction Adds

Tools that store each prompt, input and output as the analysis proceeds [25] would have answered many of the questions this study had to reconstruct. This project was not run with such a tool. The reconstruction used here needed only the decision records, saved output files and correction logs the project kept, and it reports each link with an explicit evidence status instead of assuming the missing ones (File S5). The empirical result also has implications for prospective design. The link held where corrections were itemised and versions kept, and failed where they were not, so a prospective system needs a change entry for every new data state as well as a log of model interactions. Work on human oversight asks whether humans can effectively review algorithmic output [16,17]. The episodes here demonstrate that review is the starting point of oversight, not the end. A ruling exercises authority over the data only if a recorded action brings it into the file accepted for analysis.

5.7. Practices for Recording Oversight

Each break observed in this project corresponds to a record that was not kept. The practices that follow from those breaks are listed in Table 13 and offered for future projects to test. The evidence supporting each is provided in File S4.

Table 13. Proposed recording practices and the case behind each.

No.	Practice	What It Records or Requires	Case in This Project
1	Record every ruling	A decision and its reason for every flagged concern, including a decision to keep the value	Review `A_0002`, the silent acceptances
2	Separate authorisation from implementation	The authorised action and, separately, the evidence that it was carried out	Review `A_0177`
3	Never overwrite a saved file	Every new data state saved as a new version, with its hash recorded when it is created	Complaint `CR_052`, the 31 files modified after creation
4	Compare every recode	A comparison of each recoded batch with the batch it replaces, and review of every unexplained change before acceptance	Batch 029
5	Record rule states and rechecks	The rules available to each coding run, and the earlier records rechecked when a rule was given earlier scope	CC Decision 7

Table 13. *Cont.*

No.	Practice	What It Records or Requires	Case in This Project
6	Match changes in both directions	Changed cells matched to correction entries and correction entries matched to changed cells	The final stage, Review `A_0982`
7	Identify audited records	The identifier of every source record the Auditor checks	The five CC Batch 01 records recovered later
8	Return to the source	A check against the source as well as a comparison of outputs	Review `A_0011`
9	Register the counts used in analysis	Each count with its calculation and dataset version	Table 11
10	State units and denominators	The counting unit and denominator for every count of actions, records, cells or repairs	Tables 4–12

5.8. Scope

The study examines one completed project with two collections built for different purposes. The episodes are the cases that can be traced in the surviving records. Changes overwritten during production and rulings never written down are not included in the census, which includes the saved batch versions and final-stage correction records. The results reveal the records that a long AI-assisted coding process requires to be traced. They do not estimate the frequency of each type of break in other projects.

6. Conclusions

In this project, human oversight resulted in decisions to retain questioned values and decisions to change questioned values, both of which were outcomes of review. Whether an approved change reached the analysed data depended on how its implementation was recorded. Every change in the accepted data was traced to an instruction where corrections were logged by record and field and the files on both sides were kept. Where batches were recoded, most changes could not be traced to any ruling, and where saved files were subsequently modified, their earlier versions were lost. None of the seven counts used in the underlying project's analysis changed, but individual corrections resulted in changes to other calculated counts. Recording a human decision is therefore not enough to show that oversight was effective. A project also needs evidence that each decision was implemented in the saved data, and a link from each count used in analysis to the dataset version behind it. The practices suggested here follow from the gaps in one project and are offered for others to test.

Supplementary Materials: The following supporting information can be downloaded at https://www.mdpi.com/article/10.3390/info17111177/s1, Table S1: Selected oversight-episode evidence; Table S2: Audit counts and denominator definitions; Table S3: Hand-checked census results; File S1: Agency Google Reviews coding instrument; File S2: Community Complaints coding instrument; File S3: Oversight decision register template, with field definitions and example rows; File S4: Oversight recording checklist; File S5: Retrospective reconstruction protocol.

Author Contributions: Conceptualization, C.T.L.H. and T.E.A.; methodology, C.T.L.H., A.A.T.H., T.E.A. and A.S.; investigation, C.T.L.H. and T.E.A.; writing—original draft preparation, C.T.L.H.; writing—review and editing, C.T.L.H., A.A.T.H. and T.E.A.; supervision, A.A.T.H., T.E.A. and A.S.; funding acquisition, A.A.T.H. All authors have read and agreed to the published version of the manuscript.

Funding: This APC was funded by Multimedia University, Cyberjaya, Selangor, Malaysia.

Institutional Review Board Statement: Ethical review and approval for the research project were waived by the Research Ethics Committee, Multimedia University (MMU), through its Research

Management Centre (RMC). The exemption determination was subsequently clarified by MMU in a supplementary letter. MMU classified the project as minimal risk and cited Guideline 1(3) and Guideline 1(4)(c) of the Ministry of Health Malaysia Medical Research & Ethics Committee (MREC) Guidelines for Ethical Review of Clinical Research or Research Involving Human Subjects as the basis for the exemption. This study is a retrospective analysis of previously collected, publicly accessible documentary records, comprising Agency Google Reviews and Community Complaints. No participants were recruited, and no interviews, surveys or direct interactions with individuals were conducted. Identifying information in the records, including its processing by commercial AI services, was handled as set out in Section 3.9.

Informed Consent Statement: Informed consent was not obtained from the individuals who wrote the publicly accessible reviews and complaints analysed in this study. The study is a retrospective documentary analysis of previously collected public records, and the researchers did not recruit, contact, interview or otherwise interact with the individuals concerned. The records were analysed as documentary accounts of reported experiences and allegations, not as independently verified accounts of events.

Data Availability Statement: The selected-episode evidence table (Table S1), the audit-count and denominator table (Table S2), the hand-checked census results (Table S3), the two coding instruments (FileS S1 and S2), the oversight decision register template with example rows (File S3), the recording checklist (File S4) and the reconstruction protocol (File S5) are publicly available as Supplementary Materials. The accepted Agency Google Reviews and Community Complaints datasets are retained by the research team and are not publicly available, because some coding notes refer to identifying details in the source texts. Appendix C reports their file dimensions and SHA-256 hashes, which identify the accepted dataset versions. The original source texts, complete Auditor conversations, Coding Decision Logs and other operational records are also withheld, because their release could expose individuals or link them to sensitive allegations. The Python scripts used for the census are not released. They encode the folder structure and file names of the private project archive, including the names of unpublished dataset files, and they run only on the withheld batch files, datasets, correction logs and Decision Logs. Section 3.6 gives the comparison and classification rules, and File S5 sets out each step of the reconstruction. The Supplementary Materials allow the reported evidence and methods to be examined, but they do not permit source-level reanalysis of every coded record or reconstruction of every production decision.

Acknowledgments: During the preparation of this study, the authors used Codex (ChatGPT GPT-5.5, OpenAI) for dataset coding, Claude (Claude Sonnet 4.6, Anthropic) for source-grounded auditing, and Antigravity (Gemini 3.1 Pro, Google) for output-level quality control, as described in Section 3.3. After production, Codex wrote and ran Python scripts on a read-only copy of the project archive to compare saved versions, match correction records to changed data and recalculate the counts reported in Section 4 and Appendix C. The two researchers checked 37 of these results by hand against the original files (Section 3.6 and Table S3). The authors also used Claude Sonnet 4.6 (Anthropic) and ChatGPT GPT-5.5 (OpenAI) to assist with drafting and revising sections of this manuscript. In all uses, these tools operated under written instructions and human oversight. They did not independently determine the study design, the coding rules, the interpretation of findings or the conclusions. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest. The funding sponsors had no role in the study design, data collection, analysis or interpretation, writing, or decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

AGR	Agency Google Reviews
AI	Artificial intelligence
APC	Article processing charge
CC	Community Complaints

CDL	Coding Decision Log
CFR	Code of Federal Regulations
CSV	Comma-separated values
LLM	Large language model
MMU	Multimedia University
MREC	Medical Research & Ethics Committee
QC	Quality control
RMC	Research Management Centre
SHA-256	Secure Hash Algorithm 256-bit
W3C	World Wide Web Consortium

Appendix A. Coding Fields, Rule States and Rule Map

Appendix A.1. Coding Instruments and Fields

The public coding instruments provide the field definitions, allowed values and examples for the two collections, the review instrument (File S1) and the complaint instrument (File S2). They are not complete copies of the operational prompts or the CDLs, which quote or refer to identifying source text. The fields and categories listed in the results are defined in Table A1. The category strings are the values that are stored.

Table A1. Fields and values discussed in the results.

Field or Category	What It Records	Values or Evidence Relevant Here
`review_timing`	Whether the review was written during or after the engagement	`during_service`, `after_service`, or `unknown` when no reliable time cue is present
`nanny_service_endorsement_categories`	Aspects of the nanny's work a reviewer praised	Multi-tag field including `baby_care`, `mother_care`, `night_duty`, `parent_training`, `emotional_support`, `cooking_standard` and `experience`
`parent_training`	Teaching a parent a specific newborn-care skill	General advice without described teaching does not qualify
`emotional_support`	A described action supporting the mother's emotional wellbeing	General praise for warmth or a reassuring personality does not qualify alone
`experience`	The nanny's experience given as a reason for selecting her	General praise for an experienced nanny is insufficient
`night_duty`	Care of the baby at night so that the mother can sleep	Feeding or settling the baby at night
`agency_service_endorsement_categories`	Aspects of the agency's service a reviewer praised	Multi-tag field including `good_matching`, matching a nanny to the family's needs
`premium_standard`	Whether a review identifies a higher service tier for the nanny	`1` when such a tier is mentioned. It does not measure whether the tier's advertised quality was achieved
`fraud_typology`	Alleged booking fraud or financial deception in a complaint	`None` is the literal value for no fraud element. `No-Show` applies under the deposit and non-arrival rule
`nanny_service_complaint_categories`	Problems with the nanny's service alleged in a complaint	Multi-tag field
`first_time_parent`	Whether the family was new to or experienced with parenting or confinement care	`0`, `1` and `2`, where `2` marks an experienced or repeat parent
`formal_escalation`	Whether the family pursued a formal legal or regulatory route	`Yes` can be supported by a filed police report. `No` and `Not mentioned` have separate meanings
`coder_notes`	Short notes on coding or context	A note field, not a coding category

Appendix A.2. Rule States of Coding Runs

Table A2. Decisions recorded as available in the Coder's progress log.

Collection and Batch	Decisions Available	What the Records Establish
AGR Batch 001, version 1	0 read of the 15 then existing	The session instruction asked for entries labelled as rules
AGR Batch 001, version 2	21	Recoded at the Supervisor's request after the first-batch audit produced Decisions 16 to 21 and the session instruction was corrected
AGR Batch 005, versions 1 and 2	21	Recoded at the Supervisor's request. No reason recorded
AGR Batch 029, version 1	Not recorded	The saved file exists, but its run is missing from the progress log
AGR Batch 029, version 2	21	Recoded at the Supervisor's request. No reason recorded
Other accepted AGR batches	21	Common decision count for the accepted review batches
CC Batches 01 to 05	6	Coded before Decisions 7 and 8
CC Batch 06	7	Coded after Decision 7 and before Decision 8
CC Batches 07 to 19	8	Coded with all complaint decisions

Note. A coding or recoding run could only be initiated by the Supervisor. The decision count in the progress log indicates the number of rules that were available to a run, not that all of the values in the output of a run were in compliance with those rules.

Appendix A.3. Rule Map Used to Classify Recode Changes

A category that was only referred to in a rule as an example of another code was not considered to be addressed by that rule. For example, Decision 20 does not call for a change to cooking_family, but rather identifies it as an alternative to coding incidental sibling care.

Table A3. CDL entries used to classify changes in the production-recoded AGR batches.

Field	Category or Value	Decision	Rule Direction or Purpose
`nanny_service_endorsement_categories`	`mother_care`	1	Restricts
`nanny_service_endorsement_categories`	`emotional_support`	2, 17	Restricts
`nanny_service_endorsement_categories`	`experience`	3	Restricts
`nanny_service_endorsement_categories`	`overall_support`	6	Disallows
`nanny_service_endorsement_categories`	`parent_training`	7, 18	Introduces, then restricts
`nanny_service_endorsement_categories`	`baby_care`	16	Restricts
`nanny_service_endorsement_categories`	`older_sibling_care`	20	Restricts
`agency_service_endorsement_categories`	`good_matching`, `trained_nannies`	5	Standardises labels
`agency_service_complaint_categories`	`high_cost`, `overwhelming_process`, `adjustment_needed`	9	Disallows
`selection_pathways`	`agency_assigned`	4	Permits
`first_time_parent`	`1`	19	Defines value
`agency_response_described`	`Replacement offered and fulfilled`	21	Defines value
`provider_identification_type`	`Named and internally searchable`	12, 15	Defines value

Table A3. *Cont.*

Field	Category or Value	Decision	Rule Direction or Purpose
`agency_service_complaint_categories`	`poor_matching`, `poor_screening` and other standardised labels	5, 8	Standardises labels
`review_context`	Blank when `complaint_review = 0`	11	Defines value
`agency_service_endorsement`	`0` without a specific agency action	13	Defines value
`review_changed`	`1` for a review beginning `Update:`	14	Defines value
All binary fields	`0` or `1` only	10	Permitted values

Appendix B. Selected Oversight Episode Evidence

The nine inventory entries relating to individual records are listed in Table A4, with the complete evidence record for each listed in Table S1. Identifiers are not consecutive, but follow the case inventory. There are two file histories that relate to these episodes. The first saved version of AGR Batch 001 was modified after the second version was created, so the comparison for Review A_0033 is between the two saved versions, and the first version is not necessarily the first-run output. In the case of Review A_0982, which had a malformed row, three of the field values are different between the combined files before and after the final corrections, and each of those differences corresponds to a log entry, while the log also records intermediate repairs that leave no difference between those two files.

Table A4. Episodes concerning individual records and the strongest conclusion their records support.

Episode and Record	Concern and Ruling	Implementation and Accepted Value	Status of the Open Link
E01, AGR `A_0033`	Auditor questioned `emotional_support`. Supervisor agreed and clarified the evidence requirement in Decision 17	Batch 001 v1 has `baby_care`, `mother_care`, `parent_training`, `cooking_standard`, `emotional_support`. v2 and accepted data have `parent_training`, `cooking_standard`	The removals of `baby_care` and `mother_care` are rule-addressed under Decisions 16 and 1 but named in no ruling for this record. Partly documented
E02, AGR `A_0047`	Supervisor rejected a timing change from `during_service` to `after_service` and accepted removal of `parent_training`, clarified in Decision 18	Timing remains `during_service`. `parent_training` removed in Batch 001 v2 and accepted data	Documented
E03, AGR `A_0002`	Auditor proposed `after_service`. Supervisor kept unknown because the past tense did not establish when the review was written	unknown in Batch 001 v1, v2 and accepted data	Documented. No correction authorised
E04, pilot `A_0011`	All four pilot outputs omitted `night_duty`. A return to the source added it to the pilot reference	Batch 001 v1 also omitted `night_duty`. The recoded version and locked dataset include it	Pilot finding. No production correction followed from it

Table A4. *Cont.*

Episode and Record	Concern and Ruling	Implementation and Accepted Value	Status of the Open Link
E05, pilot `A_0033`	Pilot outputs differed on the threshold for recording language as mixed or English	No new rule. Locked dataset unchanged	Pilot ambiguity, not a production intervention
E08, CC `CR_052`	Auditor reported `None` in `fraud_typology` and proposed `No-Show`. Supervisor accepted, and Decision 8 clarified the rule	Both saved Batch 06 versions and the accepted data contain `No-Show`. The versions differ only in `coder_notes`	The earlier value `None` is in no retained file. Batch 06 v1 was modified on the audit day. Conflicted
E09, CC `CR_007`	The earlier wording of `first_time_parent` did not permit the stored value 2. A later authorised clarification allowed 2 for experienced parents	The accepted instrument permits 2	A rule clarification. The classification of the complaint itself was not rechecked
E10, CC `CR_074`	Auditor questioned whether the family had taken formal action	A later source check supports `formal_escalation = Yes` because the mother filed a police report	The production ruling sequence is not recovered. Partly documented
E15, AGR `A_5016`	Quality Controller warning W02 questioned `experience` under Decision 3. Removal authorised in the final correction log	Batch 101 v1 has `baby_care, experience, mother_care`. v2 and accepted data have `baby_care, mother_care`. The `experience` count fell from four to three of 10,814 eligible reviews	Documented. No change to the seven counts used in analysis

Appendix C. Counts, Denominators and File Records

Appendix C.1. Production Audit Measures

Table A5. Measures derived from the production source audits.

Measure	AGR	CC	Basis
Checking occasions in the aggregate logs	195	37	A record can contribute more than once
Checking occasions with a counted discrepancy	57	15	At least one error counted by the Auditor under the conventions of that check
Field positions compared, historical audit summary	4290	444	As counted in the summary made during production
Primary field positions compared, recounted	4680	370	Checking occasions multiplied by the 24 AGR or 10 CC primary fields
Reconstructed multi-tag category instances	419	251	Across 156 identifiable checked AGR records and 32 identifiable CC rows

Note. The recounted row is a separate calculation and is not a replacement for the historical summary. The identifiers of five CC Batch 01 complaints (CR_001, CR_003, CR_005, CR_007 and CR_009) were recovered from an Auditor conversation after the historical summary was made. The category values were not captured in a format that the reconstruction could count, so the CC category figure covers only the 32 rows whose identity could be matched. Two Auditor conversations for CC Batch 08 describe a check of two complaints and a check of three, and the aggregate log shows only the check of three (E11).

Table A6. AGR category findings by category, as labelled by the Auditor.

Category	Field	Overcoded	Undercoded	Total
baby_care	Nanny service	0	7	7
breastfeeding_support	Nanny service	9	0	9
cleaning_standard	Nanny service	5	0	5
cooking_family	Nanny service	4	0	4
experience	Nanny service	1	0	1
mother_care	Nanny service	7	1	8
night_duty	Nanny service	1	0	1
older_sibling_care	Nanny service	5	0	5
parent_training	Nanny service	7	0	7
fast_response	Agency service	5	0	5
good_matching	Agency service	2	0	2
smooth_admin	Agency service	4	0	4
Total		50	8	58

Note. The labels overcoded (category judged unsupported) and undercoded (category judged missing) are the Auditor's, and they were not checked independently against the 48 source reviews. "Nanny service" is nanny_service_endorsement_categories and "Agency service" is agency_service_endorsement_categories. All 20 CC findings are undercoded, and all are in nanny_service_complaint_categories.

Appendix C.2. Audit Findings Compared with the Accepted Data

A finding matches when a category that the Auditor considered unsupported is absent from the accepted value, or a category that the Auditor considered missing is present. Each field was split on commas, and categories were compared exactly.

Table A7. Category findings that do not match the accepted value.

Group	AGR Records	CC Records
Noted without being counted, pending a field recode that was not carried out (6)	B_0005, B_0012 (two categories), B_0035 (parent_training), B_0037, C_0001	None
Agreed removal not implemented (1)	A_0177	None
No Supervisor reply found (9)	A_2431, A_5958, and baby_care in B_0012, B_0035, D_0154 and G_0041	Financial Exploitation in CR_181, Neglecting Baby in CR_185 and Ignoring Instructions in CR_186, all from Batch 19

Note. The number in brackets is the number of category findings in each group. Review B_0012 contributes two findings to the first group and one to the third.

Appendix C.3. Final-Stage Counts and Units

Table A8. Correction records and changed cells between the two final-stage dataset versions.

Measure	AGR	CC	Unit
Changed records	59	30	Records
Changed cells	80	30	Field values of one record
Changed cells matched to a log entry	80 of 80	30 of 30	Matched by record and field
Main correction entries	86	30	Log entries

Table A8. *Cont.*

Measure	AGR	CC	Unit
Structure note	1	0	Note on the malformed row `A_0982`
Separate whitespace entry	1	0	Log entry
Distinct record-field pairs in the logs	87	30	Pairs
Logged pairs without a difference between the two files	7	0	Pairs, mainly the intermediate repair of `A_0982`

Note. The number of logged pairs (87) is greater than the number of changed cells (80) because the two measures count different units, and because the log records intermediate repairs of the malformed row of Review A_0982 that do not result in a difference between the two files. The seven pairs that do not differ are not lost changes. Restoring missing trailing empty fields in 818 CSV rows and ignoring two empty lines were repairs to the file format, not coding corrections.

Appendix C.4. Post-Production Pilot

Table A9. Agreement among the four pilot outputs on primary coding fields.

Comparison	AGR	CC
Researcher 1 and Researcher 2 agree	1173 of 1200	86 of 100
Coder and Auditor agree	1149 of 1200	83 of 100
Field values without agreement across all four outputs	122 of 1200	36 of 100
AI outputs agree with each other but differ from the reference	21	1

Table A10. Category additions and omissions relative to the pilot reference.

Output	Unsupported Additions	Supported Omissions
AGR Researcher 1	60	57
AGR Researcher 2	55	51
AGR Coder	54	11
AGR Auditor	63	19
CC Researcher 1	13	69
CC Researcher 2	14	60
CC Coder	0	0
CC Auditor	16	23

Note. An unsupported addition is a category that is present in the output but not in the reference, and a supported omission is a category that is present in the reference but not in the output. One additional CC Auditor difference, in selection_pathways, was noted outside of the category additions and is not included. The counts are for the pilot records only.

Appendix C.5. Dataset Identity and Saved Batch Versions

Table A11. Identity of the two accepted datasets at the lock.

Collection	Rows	Columns	Unique Record IDs	SHA-256
AGR	11,458	30	11,458	`2ba1c3c68afd85043f030b14d38ecb109eb6eee4f8faf2bcee90d950be762df5`
CC	187	15	187	`6c5d31507ae716d1cfc6063c60751632ceb423c86de99365ed6b7f4b8d5c67e6`

Note. Every record has an identifier, and no new lock was created during the verification of the files for this study. The hashes are SHA-256 hashes of the accepted versions of the files, but the datasets are not public, so the hashes cannot be verified outside the project.

Table A12. Second batch versions and what their comparison establishes.

Mechanism	Batches	Comparison Result
Recode during production	AGR 001, 005 and 029	76 changed cells and 109 individual changes
Final-stage second version	AGR 003, 011, 020, 027, 074, 088, 101, 156, 181, 182, 188, 210 and 223	85 changed cells between version pairs. Four second versions differ only in CSV formatting
Final-stage second version	CC 01, 05, 06, 08, 09, 11, 12, 13, 17, 18 and 19	30 changed cells, all in `coder_notes`

Note. The counts are not cumulative across the rows, as each row compares two different pairs of files. The 85 AGR cells compare each final-stage second version to its first version, and the 80 cells in Table 8 compare the combined files before and after final corrections. Production recoding changed 76 cells, which are not counted in either. The batch manifest lists the accepted version of every batch.

Appendix C.6. Records Created During Production and Compiled Afterwards

The records created during production are the Coding Instructions, the Coding Decision Logs, the Coder's progress log, the saved batch files, the Auditor conversations and audit reports, the aggregate audit logs, the Quality Controller reports, the final-stage correction logs and the lock record. Records compiled after the lock are the case inventory and episode evidence (Appendix B and Table S1), the census and its recalculated counts, the reconstructed audit figures (Table S2), the post-production pilot and the check of all 187 complaints for two fraud labels. An evidence status is assigned to each link in an episode, indicating the type of record supporting the link.

References

1. Gilardi, F.; Alizadeh, M.; Kubli, M. ChatGPT outperforms crowd workers for text-annotation tasks. *Proc. Natl. Acad. Sci. USA* **2023**, *120*, e2305016120. [CrossRef] [PubMed]
2. Pangakis, N.; Wolken, S. Keeping humans in the loop: Human-centered automated annotation with generative AI. *Proc. Int. AAAI Conf. Web Soc. Media* **2025**, *19*, 1471–1492. [CrossRef]
3. Sulaiman, M.H.; Muda, N.; Abdul Razak, F. Analyzing patient complaints in web-based reviews of private hospitals in Selangor, Malaysia, using large language model-assisted content analysis: Mixed methods study. *JMIR Form. Res.* **2025**, *9*, e69075. [CrossRef] [PubMed]
4. Dunivin, Z.O. Scaling hermeneutics: A guide to qualitative coding with LLMs for reflexive content analysis. *EPJ Data Sci.* **2025**, *14*, 28. [CrossRef]
5. Nowell, L.S.; Norris, J.M.; White, D.E.; Moules, N.J. Thematic analysis: Striving to meet the trustworthiness criteria. *Int. J. Qual. Methods* **2017**, *16*, 1609406917733847. [CrossRef]
6. Nguyen-Trung, K. ChatGPT in thematic analysis: Can AI become a research assistant in qualitative research? *Qual. Quant.* **2025**, *59*, 4945–4978. [CrossRef]
7. Bang, S.H.; Han, S.; Turchioe, M.R.; Ellison, M.; Dai, S.; Happ, M.B.; Russell, D.; Masterson Creber, R. Developing and evaluating human-led and large language model-assisted hybrid deductive-inductive workflows for qualitative analysis. *J. Am. Med. Inform. Assoc.* **2026**, *33*, 2079–2091. [CrossRef] [PubMed]
8. Umer, M.; Asif, M.; Xue, S.; Jones, B.D.M.; Dennis, C.-L.; Naeem, F.; Mulsant, B.H.; Husain, M.I. Using ChatGPT for thematic analysis of qualitative interviews in cultural research: A methodological investigation. *Asian J. Psychiatr.* **2026**, *122*, 105071. [CrossRef] [PubMed]
9. Jayawardene, V.; Ewing, M.T. Generative AI-augmented thematic analysis. *Int. J. Mark. Res.* **2026**, *68*, 162–193. [CrossRef]
10. Moreau, L.; Missier, P. (Eds.) PROV-DM: The PROV Data Model. W3C Recommendation. 30 April 2013. Available online: https://www.w3.org/TR/2013/REC-prov-dm-20130430/ (accessed on 28 October 2026).
11. Gebru, T.; Morgenstern, J.; Vecchione, B.; Vaughan, J.W.; Wallach, H.; Daumé, H., III; Crawford, K. Datasheets for datasets. *Commun. ACM* **2021**, *64*, 86–92. [CrossRef]

12. Hutchinson, B.; Smart, A.; Hanna, A.; Denton, R.; Greer, C.; Kjartansson, O.; Barnes, P.; Mitchell, M. Towards accountability for machine learning datasets: Practices from software engineering and infrastructure. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency (FAccT '21), Virtual Event, Canada, 3–10 March 2021*; ACM: New York, NY, USA, 2021; pp. 560–575. [CrossRef]
13. Wiebe, J.P.; Khan, R.; Burns, S.; Slotta, J.D. Qualitative research in the age of LLMs: A human-in-the-loop approach to hybrid thematic analysis. In *Proceedings of the 19th International Conference of the Learning Sciences (ICLS 2025), Helsinki, Finland, 10–13 June 2025*; International Society of the Learning Sciences: Bloomington, IN, USA, 2025; pp. 1123–1131. [CrossRef]
14. Jafarian, M. Human-in-the-loop LLM-assisted thematic analysis: A validated, reproducible tutorial for large-scale qualitative research. *Behav. Res. Methods* **2026**, *58*, 329. [CrossRef] [PubMed]
15. Wang, X.; Kim, H.; Rahman, S.; Mitra, K.; Miao, Z. Human-LLM collaborative annotation through effective verification of LLM labels. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems (CHI '24), Honolulu, HI, USA, 11–16 May 2024*; ACM: New York, NY, USA, 2024; pp. 1–21. [CrossRef]
16. Green, B. The flaws of policies requiring human oversight of government algorithms. *Comput. Law Secur. Rev.* **2022**, *45*, 105681. [CrossRef]
17. Sterz, S.; Baum, K.; Biewer, S.; Hermanns, H.; Lauber-Rönsberg, A.; Meinel, P.; Langer, M. On the quest for effectiveness in human oversight: Interdisciplinary perspectives. In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency (FAccT '24), Rio de Janeiro, Brazil, 3–6 June 2024*; ACM: New York, NY, USA, 2024; pp. 2495–2507. [CrossRef]
18. Bowen, G.A. Document analysis as a qualitative research method. *Qual. Res. J.* **2009**, *9*, 27–40. [CrossRef]
19. U.S. Food and Drug Administration. Data Integrity and Compliance with Drug CGMP: Questions and Answers; Guidance for Industry. U.S. Food and Drug Administration: Silver Spring, MD, USA, 2018. Available online: https://www.fda.gov/regulatory-information/search-fda-guidance-documents/data-integrity-and-compliance-drug-cgmp-questions-and-answers (accessed on 6 November 2026).
20. World Health Organization. Annex 4: Guideline on data integrity. In *WHO Expert Committee on Specifications for Pharmaceutical Preparations: Fifty-Fifth Report*; WHO Technical Report Series No. 1033; World Health Organization: Geneva, Switzerland, 2021. Available online: https://www.who.int/publications/m/item/annex-4-trs-1033 (accessed on 6 November 2026).
21. Electronic Code of Federal Regulations. Title 21, Part 11, § 11.10: Controls for Closed Systems. Available online: https://www.ecfr.gov/current/title-21/chapter-I/subchapter-A/part-11/subpart-B/section-11.10 (accessed on 6 November 2026).
22. Barrak, A.; Eghan, E.E.; Adams, B. On the co-evolution of ML pipelines and source code: Empirical study of DVC projects. In *Proceedings of the 2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), Honolulu, HI, USA, 9–12 March 2021*; IEEE: New York, NY, USA, 2021; pp. 422–433. [CrossRef]
23. Reiss, M.V. Testing the reliability of ChatGPT for text annotation and classification: A cautionary remark. *arXiv* **2023**, arXiv:2304.11085. [CrossRef]
24. Atıl, B.; Aykent, S.; Chittams, A.; Fu, L.; Passonneau, R.J.; Radcliffe, E.; Rajagopal, G.R.; Sloan, A.; Tudrej, T.; Ture, F.; et al. Non-determinism of "deterministic" LLM system settings in hosted environments. In *Proceedings of the 5th Workshop on Evaluation and Comparison of NLP Systems (Eval4NLP 2025), Mumbai, India, 23 December 2025*; Association for Computational Linguistics: Kerrville, TX, USA, 2025; pp. 135–148. [CrossRef]
25. Lu, M.H.; Ellegood, R.; Rodriguez-Ramirez, R.; Blumert, S. Affording process auditability with QualAnalyzer: An atomistic LLM analysis tool for qualitative research. *arXiv* **2026**, arXiv:2604.03820. [CrossRef]
26. Ebneyamini, S.; Sadeghi Moghadam, M.R. Toward developing a framework for conducting case study research. *Int. J. Qual. Methods* **2018**, *17*, 1–11. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.