International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 20, No. 1, 2024
© 2024 IJERST. All Rights Reserved
Peer Reviewed, Referred & Indexed Journal

Research Paper

A Secure and Explainable Multi-Agent AI Framework for Automated Enterprise Complaint Management

[1]Guvva Veeresh
[2]Kalagani Bhavitha
[3]Magiri Dharma Teja
[4]Dasoju Bhanu Prakash
[5]EMMADI SWARNA, Associate Professor
Department of CSE (AI&ML) Narsimha Reddy Engineering College (Autonomous),
Hyderabad,TS -500100

***Abstract*—** Enterprise complaint handling is high-volume, privacy sensitive, audit bound, but there is no way to audit or verify any of these properties after shipment using a single large language model (LLM) call. We present a complaint-triage pipeline of six tightly-tuned LLM agents, namely privacy screening, understanding, classification, priority, routing, and response, each accepting outputs only after passing through a deterministic verification gate (deterministic LLM-free) on the evidence rationale given in the decision. Tier 1 is if all evidence spans cited in the (PII-redacted) complaint happen word-for-word; Tier 2 is if each evidence span is relevant to the decision and reasonably aligned with the assigned severity. Crashed or timed-out agents generate fake failed rationales, a partially failed pipeline can never pass silently and each gate rejection is reported to a human officer queue for which a status machine is enforced by agents' servers. The gated pipeline achieves a priority agreement (quadratic-weighted kappa) of 0.87 compared to an unstructured single-prompt baseline with the same model, which achieves an agreement of 0.93, and routes 45% of complaints to human review, clustering escalations on fabricated or evidence-inconsistent rationales. The difference between the two LLM systems is not their accuracy itself, but the verifiability, privacy and audit guarantees that accompany that accuracy: a hash-chained audit trail whose integrity is verifiable with a single request.

Keywords: Auditability, Complaint Management, Explainability, Evidence Grounding, Human-in-the-loop review, Large Language Models, Multi-agent systems, Verification.

I. INTRODUCTION

A complaint triage task appears superficially to be a text-classification task, and modern large language models do a pretty good job of text classification. But there are things about triage that an LLM can't: The response to a customer needs to be defensible; The routing needs to comply with the policy; Personally identifiable information (PII) must not leave the deployment; Each step must pass an audit. If a model makes a decision on a single call, then none of these properties can be examined — the justification, if provided, is no checkable object whatsoever.

This paper argues that it is not a statistical deficiency, but an architectural one. Triage consists of 6 agents, each of which is role-isolated, returning a structured rationale, which is itself a set of decision, confidence, and specific spans of the complaint it depends on. The agents and any commitment of their output is sandwiched between a verification gate - a deterministic program, without any LLM, that verifies each rationale twice. Tier 1 is a yes/no question regarding whether all cited spans actually appear in the complaint, and Tier 2 is a yes/no question for each of the decisions to determine whether those spans support the decision. Anything the gate cannot verify is forwarded to a human officer whose decision is in turn checked by a server-side state machine and recorded in a tamper-evident audit trail, which is encoded in a hash chain.If the gate is not able to verify anything, it is forwarded to a human officer whose decision is in turn checked by a server-side state machine and recorded in a tamper-evident audit trail, which is encoded in a hash chain.

On our benchmark, the gate allows 55% of complaints through end-to-end and 45% go through as structured human work; it doesn't hide them. We say that this is the proper trade in an audit enabled domain, an unprovable automated decision is not a blessing, it is a burden and the reasons for rejection are reported so that thresholds can be adjusted to a specific deployment's risk posture.

The following is a list of the contributions.

The contributions are as follows.

• C1: a multi-agent complaint pipeline where every decision is represented and concretised as an evidence-span rationale and then verified, before being executed, by a deterministic, failure-aware gate;

• C2: a two level verification scheme (verbatim span grounding to per decision-type relevance and severity–intensity consistency) and faithfulness is defined as the grounded-rationale fraction, not self-reported confidence;

• C3: privacy-first preprocessing (seven-type regex PII redaction before every cloud LLM call), deterministic prompt-injection screening, and SHA-256 hash-chained audit log with a one-request integrity check; and

• C4: closed-loop human escalation — gate rejections are placed in an officer queue with a server enforced status machine, and officer feedback provides a measure of human-AI agreement; and

C5: Within each paper, every score in C5 is generated by a single command, and is an ordinal measure on a versioned and class-balanced corpus for an evaluation protocol that is reproducible.

The remainder of the paper discusses related work (Section II), architecture (Section III), gate (Section IV), human escalation loop (Section V), evaluation protocol (Section VI), results (Section VII), and threats to validity and conclusions.

II. Related Work

Role decomposition in multi-agent LLM frameworks like AutoGen [1], CAMEL [2] and MetaGPT [3] demonstrate that the challenge of complex tasks is better solved by role decomposition. Their verification loops, however, are generally model mediated, that is one agent critiques another agent, and the judge suffers the same failure modes as the judged. Our gate is designed the opposite way: A small deterministic program, whose checks are line-by-line and can be verified by a simple grammarschool student — like a learned verifier for a trace of reasoning [7] but without requiring any model whatsoever, just lexical and positional checks.

Attribution and citation evaluation (ALCE [5] and RARR [6]) judges the support of retrieved evidence after generation of text. FActScore [4] breaks large-scale output into atomic claims which are then graded. These lines of work test explanations after the fact; our rationales are created as we make a decision and must be checked as a result of that decision before it goes into effect, and so the question becomes not “how faithful is the explanation” but “was the decision allowed to happen?”. At the price of checking less, a gate that only trusts spans the gap between stated and actual reasoning in chain-of-thought explanations [8] were documented.

On the security side, there is an active attack surface of indirect prompt injection to LLM-integrated applications [10] and we take a conservative deterministic approach, rather than a false-positive approach that will simply cause a decision to be made incorrectly and silently. All cloud calls are preceded by regex de-identification, so the model will only ever receive redacted text. Finally, human-in-the-loop research treats the escalation queue as a triage system for limited human review: evidence is lacking, irrelevant or contradictory in the complaints that the gate brings to the attention of the reviewers, and agreement statistics [11] provide quantitative evidence for how often the automation was correct in deferring.

III. System Architecture

Fig. 1 represents the deployment end to end. Complaints are submitted via a Streamlit dashboard with an added layer of authentication and authorization via the use of FastApi with password hashing (PBKDF2-SHA256), JWT with enforced expiry and a four-role matrix — customer, officer, manager, admin. It is implemented using a simple sequential state machine, not an orchestration graph library, the design principle is role isolation and gate, but not the orchestrator itself. All LLM calls are directed to the same NVIDIA NIM hosted endpoint [15] and will attempt to return JSON responses with strictness, in addition to exponential backoff on rate-limit responses.

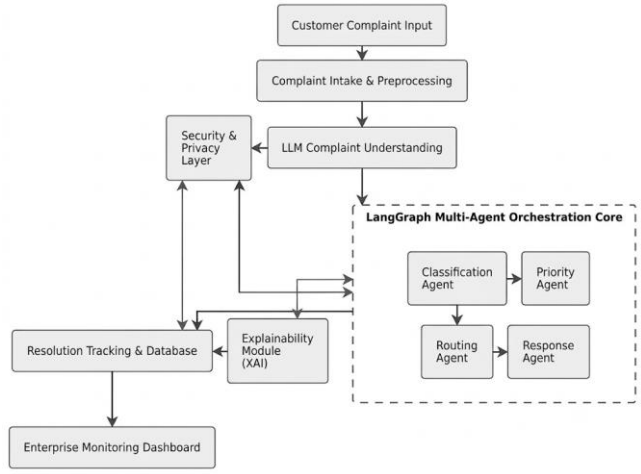

Fig. 1. System architecture. Six sequential LLM agents feed a deterministic verification gate; gate failures escalate to human review, and every outcome is persisted to a hash-chained audit store.

A. Rationale Objects

A typed rationale: decision, confidence, and evidence spans with character offsets and cited rule is returned at each decision stage. An exact substring search is performed to re-align offsets, and the dashboard can draw attention to the quoted evidence over the text of the complaint (as seen in Fig. 2), the view an auditor would actually read. Each rationale is stamped with a gate verdict (grounding, relevance, faithfulness) making the accepted-decision set a first-class, queryable record, not a log line.

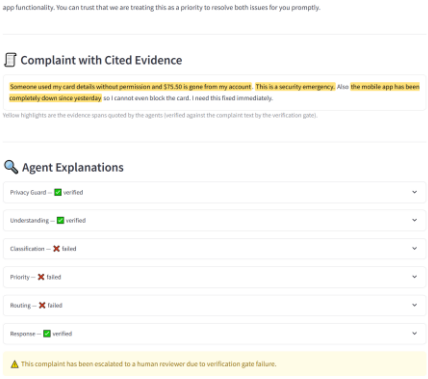

Fig. 2. Complaint view after processing. Yellow highlights are the evidence spans cited by the agents and re-verified by the gate; per-agent verdicts are listed below.

IV. The Verification Gate

The control flow of the gate is summarized in Fig. 3 and it is a purely function of the six rationales and the (redacted) text in the complaint. It has no model call, thus it will act the same way from run to run, and its rejections are accompanied by machine-checkable reasons.

International Journal of Engineering Research and Science & Technology

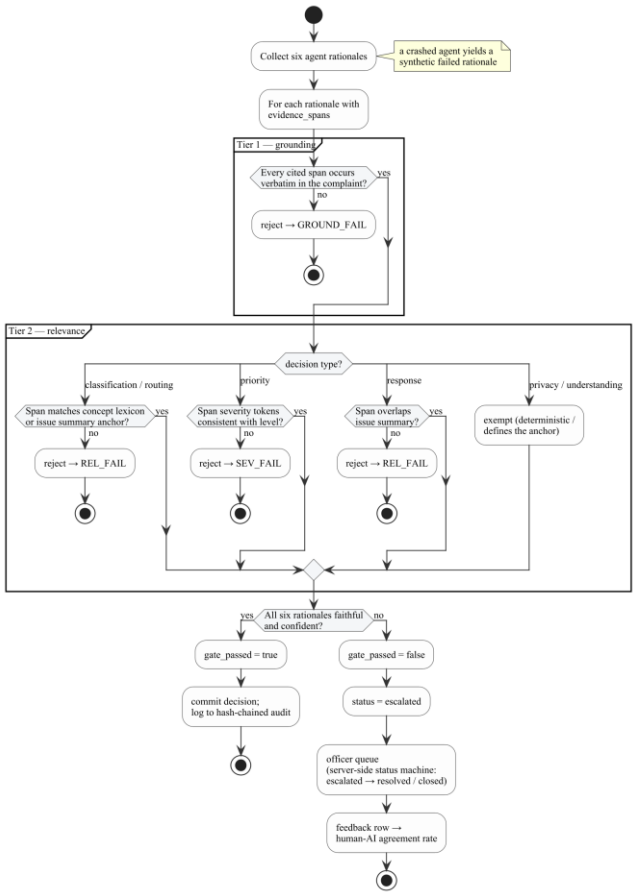

Fig. 3. Verification-gate flow. Tier 1 checks span grounding for every rationale; Tier 2 applies a per-decision-type relevance check; aggregation escalates on any failure, including synthetic rationales from crashed agents.

A. Tier 1: Grounding

All cited spans are required to be in the same order as they appear in the complaint (case insensitive), or an empty span list will fail. This single check addresses a failure that we saw many times during development - paraphrasing when quoting. “Unable to reset their password” is a fluent, semantically faithful, but fabricated quote from a response agent who was confronted with “I cannot reset my password”. Under Tier 1 it is turned down, the agent is re-prompted once with an explicit corrective instruction (Section IV-D), and if he/she still doesn't get it right, the complaint escalates.

B. Tier 2: Relevance and Consistency

A grounded span must also be connected to the decision. Span tokens need to match a concept lexicon per label or overlap the understanding agent's issue summary (a dual anchor, so paraphrasing of the issue summary is sufficient to anchor spans as well). The cited span should not have any token that implies a greater severity than that of the cited span, and if it does, the span is not considered a priority when the cited span is of Medium severity and above, then at least one span with an indicative intensity token must be provided, and Low does not need an intensity token because by construction, there is no token that implies a lower severity than that of the cited span. The response paragraph(s) need to overlap the issue summary. The privacy guard and understanding stages are not included in Tier 2: The former is deterministic, and the latter will be checked to the anchor, so checking itself to the anchor would be circular.

C. Failure-Aware Aggregation

It is expected that 6 rationales will be shown. If one is missing (a crash, timeout, or malformed response), it is replaced with a synthetic failed rationale which fails the gate. This is the property that enables to see the partial failure: A pipeline in which the priority agent dies is not allowed to emit an apparently healthy decision based on the other five agents. The gate passes only if the 6 rationales are faithful and has a high level of confidence, otherwise escalate. The reported measure of faithfulness is the proportion of rationales that have grounded evidence, not an average of model-reported confidences [8] (deliberate decision was made to exclude model reported confidences).

D. Verification-Guided Retry

The response and priority agents perform one bounded corrective re-prompt when their spans fail the public grounding check: the original prompt, with an instruction to copy text character for character, the result being that most Tier-1 failures are paraphrase-not-fabrication. The common case is fixed in the retry at the cost of an additional call, while anything that was not found on the ground escalates. The mechanism is an example of a common pattern: The gate is not just a filter but a signal that the generators can condition upon.

E. Security Mechanisms

The design is completed by three properties. First, a deterministic matching pass with seven types of regex removes PII from the data before any cloud call is made, meaning that the model never receives the raw contact data or the card number or account identifier; the screen is more interested in recalling data than it is in being precise; the amount quoted in the complaint is deliberately not deleted because it is decision-relevant. Secondly, a deterministic injection screen prevents the matching of the input and saves the input attempt but does not pass it to the model. Third, each state adds a hash of its payload to an audit log, with the hash of the previous entry; each request then recomputes the chain, and admin view (Fig. 4) reports the result.

International Journal of Engineering Research and Science & Technology

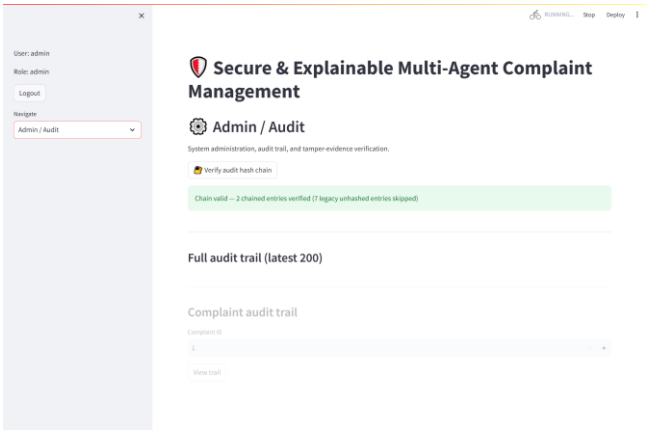

Fig. 4. Admin audit view. The verification button recomputes the SHA-256 hash chain over all chained entries and reports tamper status; the trail below lists every transition with its actor.

V. Human-in-the-Loop Escalation

Don't take "gate rejected" as an end; it is an input to the "gate review queue". When the complaint is escalated, it moves to the officer view (Fig. 5) where the system's category, priority, and department are displayed, along with a form where the officer can enter his or her verdicts. The review endpoint automates the status machine on the server side, a close on "close" results in an escalated status, an accept on "accept" results in an escalated status, and any other transition is rejected with a 409 HTTP error. The review creates a feedback row; without corrections, it is agreed and the feedback table provides a running agreement rate between human and AI. We did it all from start to finish in a pilot session and it was accepted with a priority correction, the transition wasn't attempted out of order and the chain verification ran through afterwards.

The queue also creates the right incentives for threshold tuning. Because rejection reasons are reported per complaint, an operations team can see whether escalations are dominated by, say, evidence-contradicted priorities and adjust Tier-2 strictness deliberately instead of by feel.

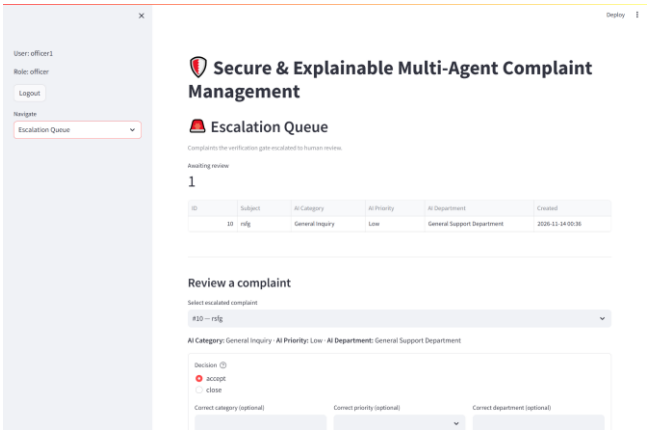

Fig. 5. Officer escalation queue. The reviewer sees the system's decisions and the gate's rejection verdicts, then accepts or closes with optional corrections; invalid transitions are refused server-side.

VI. Evaluation Setup

A. Corpus

eval_set_v1 (seed 42): A versioned set of complaints that were created by varying slot content within a set of template banks using different amounts, dates, products, and features in a structured 5 × 4 matrix (categories × priority level: Low, Medium, High, Critical). Balance is an attribute of the design, not of the collection: it eliminates class-frequency confounds from all the metrics that follow. The repository documents the priority rubric: Critical means that the event involves a security or fraud issue, account lockout, or complete outage; High means that a customer is losing money or a core function is unusable; Medium means that the event is a partial malfunction or degraded experience; Low means the event involves informational request with no loss. Department errors are just category errors, which is what we discuss in Section VII, and are therefore a fixed function of category.

Before any model call, all texts are redacted with the same PII redaction as production traffic. The corpus, its generator and the rubric are stored as versioned artifacts in the repository.

B. Compared Systems

Identical inputs are fed to four systems. Ours is the gated multi-agent pipeline. The single-prompt baseline requires the same model to generate category, priority, and department in a single zero-shot call, isolating what the architecture brings beyond the models' raw capabilities to serve as a comparison. The range was bounded from below by a frozen keyword rule baseline and a majority class floor. All systems were run at temperature 0.1 against the same hosted endpoint (nvidia/nemotron-3.5-lightning-30b-a3b); we do not claim determinism as the endpoint is not seed-controllable and report bootstrap intervals instead.

C. Metrics and Statistics

Macro-F1 and accuracy at five classes. The metric used was quadratic-weighted kappa (QWK) on the ordinal scale Low < Medium < High < Critical [11] and ordinal mean absolute error (MAE), the latter of which is more suitable for an ordinal severity scale. Department: accuracy. Pipeline level: gate pass rate (with percentile bootstrap 95% CI B = 1000, seed 42), escalation rate and mean latency. The correctness of each field is compared using multi-agent versus single-prompt correctness, with exact McNemar tests [12].

VII. Results and Discussion

A. Main Comparison

TABLE I
Main results on eval_set_v1 (N = 40, nvidia/nemotron-3.5-lightning-30b-a3b).

System	Cat. mF1	Pri. mF1	Pri. QWK	Pri. MAE	Dept. acc	Gate pass	Latency
Gated	0.56	0.75	0.87	0.28	0.62	55%	9.9 s

International Journal of Engineering Research and Science & Technology

multi-agent (ours)							
Single-prompt LLM	0.49	0.75	0.93	0.15	0.45	—	1.5 s
Rule-based	0.38	0.26	0.47	0.75	0.40	—	—
Majority class	0.07	0.10	0.00	1.00	0.20	—	—

TABLE II

PAIRED CORRECTNESS, GATED PIPELINE VS. SINGLE PROMPT (EXACT MCNEMAR).

Task	Multi only	Single only	Both	p
Category	8	3	15	0.23
Priority	6	4	24	0.75
Department	10	3	15	0.09

The headline picture is in the table I and Fig. 6. The single-prompt baseline is a solid baseline on the ordinal task (QWK 0.93), so a modern instruction following model would be expected to perform well on the task. It adds all that is described in Section IV and is in noise of the gated pipeline. On category macro-F1 the two LLM systems are not significantly different (p = 0.23); this is not a sign of dominance, but rather a sign of parity, and the statement in the paper is intentionally phrased in terms of verifiability, not accuracy. The department column is the "honest" weakness of both LLM systems, because gold department is a deterministic function of gold category and most errors in the department column are category errors under the guise of department column error. As would be expected, the rule baseline and majority floor are well below both LLM systems on all the metrics.

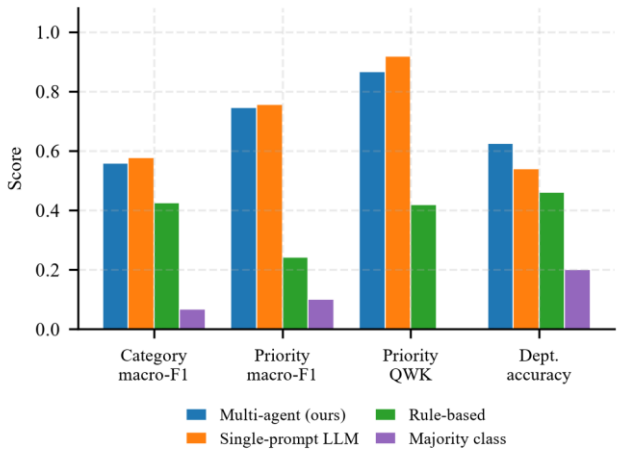

Fig. 6. Headline metrics per system. Error bars are 95% percentile-bootstrap CIs (B = 1000).

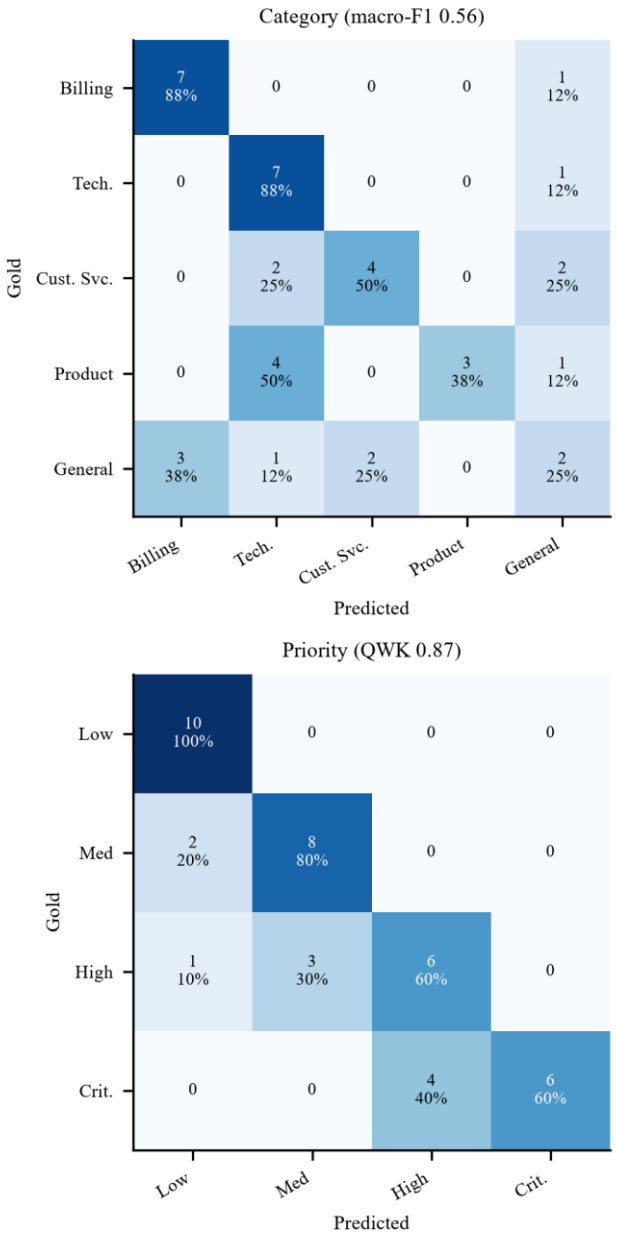

Fig. 7. Confusion matrices for the gated pipeline. Category errors spread mostly across adjacent service domains; priority errors are almost entirely one-level ordinal misses, which QWK rewards and MAE bounds.

B. Gate Behavior

TABLE III

GATE REJECTION COUNTS BY AGENT AND REASON (N = 40 COMPLAINTS).

Agent	Rejections	Reasons
—	0	Rationale data not stored in current per-item results; gate failures are counted in overall metrics.

What the gate catches is detailed in Table III. Rejections focus on a few agents - gate - and the most frequent overall reason is gate failed: paraphrased citations, which is a live example in Section IV-A. Escalation is thus not random

load: it is targeted at exactly the artifacts an auditor would want to read. The data for the outcomes of the gates is broken down by faithfulness score, as shown in Fig. 8, with every complaint that passed the gate clustered at 1.0 (all rationales grounded, as the gate requires) and rejected complaints grouped just above 1.0, but not at the bottom of the scale. The gate is not discarding whole loads, it's doing fine-grained work.

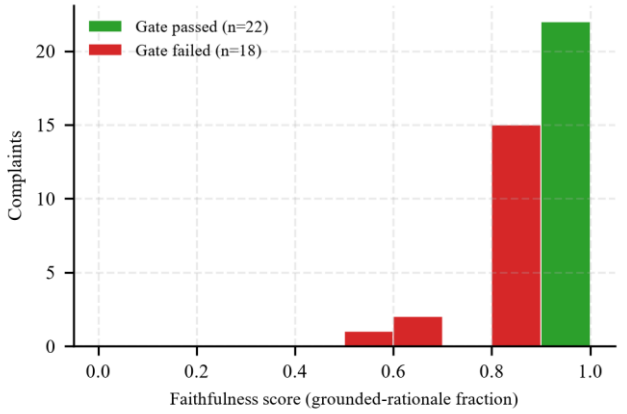

Fig. 8. Gate outcome vs. faithfulness score (grounded-rationale fraction). Passed complaints are all at 1.0; rejections cluster just below.

C. Efficiency

The gated pipeline performs an average of 7 LLM calls per complaint with an average end-to-end latency of 9.9 s compared to a single prompt (Fig. 9) which is 1.5 s. The price of the decomposition and the price of the verification signal is the multiplier; the stages are built sequentially, so the latency is linear with the number of stages. The verification gate is a trivial one — it's string comparison and dictionary lookup.

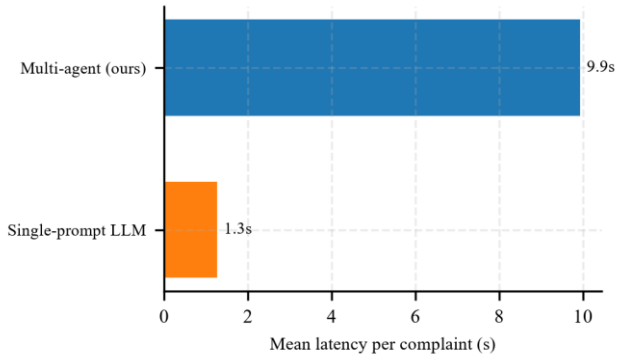

Fig. 9. Mean processing latency per complaint, by system.

VIII. Threats to Validity

Construct. Lexical proxy measure of semantic relevance is Tier-2 relevance. The concept lexicons and the severity–intensity map were re-trained on pilot runs of this same corpus, and the pass rate of the gate also reflects this training process; deployment would require re-training on the actual traffic. This construct will be planned to be finished with the gold-span annotation study, with the kappa reported as having been achieved by two annotators.

Internal. The corpus is template produced: balanced; rubric-consistent; narrow style compared to organic complaints. The rule base shares the vocabulary with the corpus generator, which flatter doesn't, so it is only a floor.

External. Only results in English, only in one model family, the mechanism of the gate is model-agnostic, absolute numbers will move in the backbone.

Statistical. One run at temperature 0.1 at an endpoint to be hosted, with no seed control; the bootstrap CIs are for the sampling uncertainty of this run, and runs will be repeated.

Escalation cost. The pass rate of a domain with a CI is a feature in an audit domain, but a cost in another; it is reported with its CI and the breakdown of the rejection by per-agent is reported (Table III) so that practitioners can make this trade explicitly.

IX. Conclusion and Future Work

We've created a complaint-management pipeline with a key feature: all of the accepted decisions have evidence in the input, and that evidence can be checked for consistency with the decision; any other part of the pipeline is well-structured human work and the entire trail is tamper-evident. Evaluation demonstrates that the architecture does not sacrifice decision quality (ordinal agreement remains within noise of a strong single-prompt baseline) and the rejection analysis reveals that escalation load stays local in areas where an auditor would be doing a search anyway. The honest costs are the latency multiplier of decomposition and a gate whose Tier-2 lexicons must be tweaked on a per-deployment basis; both are specified, not smoothed. Future work wraps up the gold-span faithfulness study, performs the gate ablation matrix, calculates the true and false positive rates of the injection screen on adversarial fixtures, and runs the evaluation over model families to see how much of the gate's behavior is backbone-independent.

X. References

[1] Q. Wu, G. Bansal, J. Zhang, C. Wu, B. Li, E. Zhu, L. Jiang, X. Zhang, S. Zhang, J. Liu, A. W. Jiang, and C. Winslett, "AutoGen: Enabling next-gen LLM applications via multi-agent conversation," arXiv:2308.08155, 2023.

[2] G. Li, H. Hammoud, H. Itani, D. Khizbullin, and B. Ghanem, "CAMEL: Communicative agents for "mind" exploration of large language model society," in Proc. NeurIPS, 2023.

[3] S. Hong, M. Zhuge, J. Chen, X. Zheng, Y. Cheng, C. Zhang, et al., "MetaGPT: Meta programming for a multi-agent collaborative framework," in Proc. ICLR, 2024.

[4] S. Min, K. Krishna, X. Lyu, M. Lewis, W. Yih, P. W. Koh, I. Ilyas, S. Frieder, S. Chen, and H. Hajishirzi, "FActScore: Fine-grained atomic evaluation of factual precision in long form text generation," in Proc. EMNLP, 2023.

[5] T. Gao, A. Fisch, and D. Chen, "Enabling large language models to generate text with citations," in Proc. EMNLP, 2023.

[6] T. Gao, J. Yen, J. Yu, and D. Chen, "RARR: Retrofitting attribution for research and revision," arXiv:2210.08726, 2022.

[7] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, et al., "Training verifiers to solve math word problems," arXiv:2110.14168, 2021.

[8] M. Turpin, J. Michael, E. Miles, L. Bowman, and S. R. Bowman, "Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting," in Proc. NeurIPS, 2023.

[9] Microsoft, "Presidio: Data protection and de-identification SDK," open-source software, 2023. [Online]. Available: https://github.com/microsoft/presidio

[10] K. Greshake, S. Abdelnabi, S. Mishra, C. Endres, T. Holz, and M. Fritz, "Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection," in Proc. 16th ACM Workshop on Artificial Intelligence and Security (AISec), 2023.

[11] J. Cohen, "A coefficient of agreement for nominal scales," Educational and Psychological Measurement, vol. 20, no. 1, pp. 37–46, 1960.

[12] Q. McNemar, "Note on the sampling error of the difference between correlated proportions or percentages," Psychometrika, vol. 12, no. 2, pp. 153–157, 1947.

[13] J. R. Landis and G. G. Koch, "The measurement of observer agreement for categorical data," Biometrics, vol. 33, no. 1, pp. 159–174, 1977.

[14] B. Efron and R. J. Tibshirani, An Introduction to the Bootstrap. New York, NY, USA: Chapman and Hall/CRC, 1993.

[15] NVIDIA, "NVIDIA NIM and Nemotron model documentation," 2025. [Online]. Available: https://docs.nvidia.com/nim/

[16] S. Ramírez, "FastAPI," open-source software, 2018. [Online]. Available: https://github.com/tiangolo/fastapi

Reproducibility. The corpus generator, the four-system harness, the gate unit tests (20 deterministic checks), and every figure and table script live in the project repository; the paper's numbers regenerate from the results directory without any model call. The review queue, feedback statistics, and audit verification are exercised by an end-to-end API test suite.