UNIVERSITY
of HULL

Agentic AI for Fair and Collaborative Peer Review

being a dissertation submitted in fulfilment of the
requirements for the degree of

Master of
Artificial Intelligence and Data Science

in the University of Hull

by

Ojei Victory

September 2025

Acknowledgement

I would like to express my sincere gratitude to my project supervisor, Dr Temitayo Matthew Fagbola, for his guidance, insightful feedback, and unwavering support throughout this work. His expertise, constructive critiques, and timely encouragement sharpened my thinking and significantly improved the quality of the project. I am especially grateful for his availability during critical phases and for directing me to the right resources and best practices.

Abstract

Peer review remains a cornerstone of research quality assurance, yet current practice struggles with scale, timeliness and consistency. Recent progress in large language models (LLMs) suggests that parts of the process can be assisted, but most automated approaches still lack fairness, depth, and robust collaboration dynamics. This project develops a human-in-the-loop (HITL) agentic AI workflow for scientific peer review that couples multi-agent deliberation with explicit safeguards. The system implements two orchestration modes—centralised (moderator-led) and decentralised (peer-to-peer)—and three core reasoning mechanisms: self-consistency checks, argumentation-based negotiation, and trust-based voting weighted by each agent's constructiveness. A human reviewer remains in the loop to approve/override decisions, inject guidance mid-debate, and audit traces.

Using the PeerRead arXiv cs.ai (2007–2017) subset as weak supervision for Good/Not-Good (accept/reject) labels, and an open-source instruction-tuned LLM (Qwen family) running on viper supercomputer system (A40 (48GB)), the centralised pipeline produced concise meta-reviews and achieved promising predictive performance on a held-out slice (Accuracy = 0.79; F1 = 0.58). The decentralised pipeline exposed richer disagreements and triggered HITL escalation more often desirable from a safety perspective with a promising performance as well of (Accuracy = 0.71; F1 = 0.41). The contribution is a reproducible template for fairer, more collaborative AI-assisted reviewing that prioritises constructive feedback and human control.

Contents

Chapter 1 Introduction

1.1 Background and motivation

Peer review is the primary mechanism by which the scholarly community validates originality, soundness and ethical integrity. However, increasing submission volumes and limited reviewer bandwidth create chronic pressure on editors and programme committees. Reviews can be delayed, uneven in quality, and at times inconsistent across reviewers assessing the same work. Meanwhile, the availability of AI-generated text has created new concerns around detectability, disclosure and integrity in the review pipeline (Yu et al., 2024).

Large language models (LLMs) are attractive assistants for summarising manuscripts and drafting structured comments. Yet single-pass prompts often produce variable depth and limited evidence-tracking, and they do not naturally debate, negotiate, or reach consensus in the way that human committees do. Without explicit safeguards, autonomy can increase risk (e.g., overconfident recommendations, ungrounded criticism). Practitioners therefore advocate HITL agentic designs that keep humans in charge while using agents to explore options, surface evidence, and draft feedback (Mudhiganti, 2024; Permit.io, 2024; Biswas, 2025).

1.2 Problem statement

Existing LLM-assisted reviewing tools tend to be:

- Shallow: they summarise but rarely scrutinise methodology or evidence.
- Inconsistent: the same paper can receive divergent outputs across runs.
- Uncollaborative: they lack mechanisms for reviewer-style debate and reconciliation.
- Opaque: recommendations are not traceably linked to cited evidence.
- Risky: they can hallucinate or overstep, with no structured role for the human editor.

Despite rapid technological progress, AI-based peer review models remain limited in their ability to replicate the nuanced and contextually rich feedback provided by human experts. Specific challenges include inherent biases present in training datasets leading to unfair outcomes, insufficient depth in automated reviews, and poor handling of interdisciplinary or complex submissions. Moreover, existing multi-agent systems often demonstrate persona inconsistency—agents shift reviewing styles unpredictably—making reviews less reliable and coherent (Baltaji, Hemmatian and Varshney, 2024).

The lack of an effective human-in-the-loop framework further exacerbates these challenges, with limited opportunities for human oversight to mitigate biases, inaccuracies, and ensure ethical compliance (Yu et al., 2024). Consequently, current automated systems fall short of widely acceptable standards required by rigorous peer review processes.

1.3 Aim and objectives

This project aims to design and evaluate a human-in-the-loop agentic AI that produces constructive, transparent peer-review outputs and a Good/Not-Good decision, while preserving human control. The objectives are to:

- Implement multi-agent workflows with self-consistency, argumentation-based negotiation, and trust-based voting.
- Compare centralised (moderator-led) versus decentralised (peer-to-peer) orchestration.
- Integrate explicit HITL gates for approval, revision requests and overrides.
- Evaluate on PeerRead labels and report predictive and qualitative outcomes.
- Release a reproducible template suitable for local deployment on modest hardware.

1.4 Research questions

RQ1.

Can self-consistency + negotiation + trust-based voting yield more stable Good/Not-Good verdicts than a single LLM pass?

RQ2.

How do centralised and decentralised agent orchestration compare in accuracy, review quality and escalation rate?

RQ3.

Does trust weighting by constructiveness (specificity, evidence, actionability, tone) improve decision quality relative to unweighted voting?

RQ4.

When should HITL intervention occur, and how do human edits shape the final meta-review and verdict?

RQ5.

What are the dominant failure modes (e.g., eloquent but unevidenced claims, persona drift, conformity), and how can guardrails mitigate them?

1.5 Contributions

- A practical HITL agentic workflow for peer review with two orchestration modes and three reasoning mechanisms (self-consistency, negotiation, trust-based voting).

- A constructiveness-aware trust model that allocates more vote weight to agents providing specific, evidence-linked, civil and actionable feedback.

- An end-to-end HITL loop (approval/override/revision prompts) with full audit traces.

- An empirical study on PeerRead using an open-source model running locally, with quantitative and qualitative analyses.

- A discussion of fairness and collaboration in multi-agent LLM review, positioning against recent systems such as AgentReview, MARG and PiCO.

Chapter 2 Literature Review

2.1 Datasets, infrastructure and integrity

Early work made peer-review research feasible by releasing labelled corpora. PeerRead aggregates paper metadata, reviews and acceptance outcomes for several venues and time windows, enabling acceptance-prediction studies and review-text analyses (Kang et al., 2018). More recently, MOPRD offers a multidisciplinary open peer review dataset with broader domain coverage (Lin et al., 2022). The openreview-py tooling has been widely used to interact with the OpenReview platform programmatically, facilitating large-scale collection and analysis of papers, reviews and decisions (OpenReview, n.d.).

At the same time, the presence of AI-generated text in scholarly pipelines has increased. Yu et al. (2024) examine the detectability of LLM-generated review text and highlight policy and tooling implications for editorial workflows. These considerations motivate explicit audit trails, transparent prompts, and HITL oversight in any agentic system that touches peer review.

2.2 LLMs for review generation and multi-agent collaboration

A growing body of work explores LLMs as reviewers. MARG (D'Arcy et al., 2024) proposes multi-agent review generation with role specialisation and coordination. AgentReview (Jin et al., 2024) studies reviewer dynamics among LLM agents, finding that coordination strategies affect diversity and quality of feedback. PiCO (Ning et al., 2024) frames peer review as consistency optimisation, using cross-agent agreement signals to stabilise outcomes. Xu et al. (2023) pursue multi-agent peer review collaboration to improve reasoning via iterative critique.

Beyond review generation per se, Generative Adversarial Reviews (Bougie & Watanabe, 2024) investigate adversarial set-ups where one agent attempts to refute another's claims—aligned with this project's argumentation-based negotiation. A broader survey of multi-agent collaboration mechanisms (Tran et al., 2025) synthesises coordination, communication and decision protocols, reinforcing the value of structured debate and voting.

However, multi-agent systems introduce new risks. Baltaji, Hemmatian and Varshney (2024) document persona inconstancy, including conformity and confabulation, which can bias collective outcomes. These findings justify trust-based weighting and HITL checks to counteract social biases among agents.

2.3 Human-in-the-loop (HITL) agentic design

Practitioner guidance has converged on human oversight as a first-class design goal. Mudhiganti (2024) provides a practical guide for engineers building agentic systems, emphasising approval gates, task routing, and confidence-based escalation. Permit.io (2024) distils best practices for HITL in agents—guardrails, auditability, granular permissions, and demonstrations of policy-aligned intervention. Biswas (2025) frames HITL as a strategy spectrum, from light-touch prompts to reflexive agents that request help when uncertainty rises. These perspectives inform this project's escalation thresholds (e.g., small vote margins) and editor-in-the-loop overrides.

2.4 Positioning and gap

The reviewed literature demonstrates that multi-agent LLMs can produce richer, more diverse reviews than single-pass models, and that coordination mechanisms (debate, voting, consistency optimisation) matter. Yet gaps remain:

- Fairness and voice: Prior systems rarely weight influence by constructiveness; they often use equal votes or opaque heuristics.
- Transparent escalation: HITL is discussed conceptually but less often implemented as a measurable gate with decision logs.
- Centralised vs decentralised trade-offs: Few studies directly compare moderator-led versus peer-to-peer debate under the same model and dataset.
- Local deployability: Many pipelines assume cloud-scale resources; editors may need modest-hardware solutions they can audit.

This project addresses these gaps by: (i) introducing a constructiveness-aware trust-vote, (ii) implementing explicit HITL gates with traceable overrides, (iii) comparing centralised and decentralised orchestration on the same PeerRead slice, and (iv) running an open-source stack locally.

2.5 Human-in-the-loop agentic systems

Recent practitioner guides highlight that useful agentic systems never remove the human from critical decisions. Mudhiganti (2024) outlines practical patterns—guardrails, approval gates, and task routing—to merge autonomous exploration with review checkpoints. Permit.io (2024) summarises HITL best practices (escalation thresholds, audit trails) and demonstrations showing how policy + feedback loops reduce risk and improve outcomes. Biswas (2025) frames HITL strategy as a continuum from advisory prompts to reflexive agents that request human assistance when self-estimated confidence drops. These sources converge on a core design: autonomy inside boundaries, with humans owning final accountability.

Chapter 3 Methodology

This chapter outlines the design and implementation of an Agentic AI system for fair and collaborative peer review. The system simulates a human review committee using multiple large language model (LLM) agents, with human-in-the-loop (HITL) oversight for ambiguous cases. It integrates three core principles—argumentation-based negotiation, self-consistency, and trust-weighted voting—applied to the PeerRead dataset.

3.1 System Architecture

Two complementary pipelines were developed:

Centralised committee: A fixed trio of agents (A, B, C) reviews each paper, critiques one another's work, and produces a trust-weighted decision and unified meta-review.

Decentralised committee: For each paper, a random subset of agents is selected. These agents independently review, self-check, critique, and judge, followed by a trust-weighted vote. Items with low consensus are flagged for HITL review, allowing human intervention to accept, reject, or revise the meta-review.

Both pipelines share the same components for review generation, critique exchange, judging, and trust computation. Their primary distinction lies in committee formation and trust evolution.

3.2 Dataset and Preprocessing

The system uses the arXiv.cs.ai_2007–2017 subset of PeerRead, loaded via HuggingFace with schema validation to handle missing fields. Each paper is represented by a compact excerpt—typically the title and abstract, or fallback review text—truncated to 1200–2000 characters. Only entries with valid excerpts are used.

3.3 Model Configuration

The backbone model is Qwen2.5-3B-Instruct, accessed via the Transformers library. GPU acceleration is preferred, with fallback to CPU. Prompts are structured as single-turn chats, and decoding strategies vary by task: low-temperature sampling for reviews and critiques, deterministic decoding for judging. Token limits range from 80 to 400, with stop tokens ensuring clean JSON outputs.

Temperature settings are tuned per task:

- Review and self-check: 0.4
- Critique: 0.5
- Judge: 0.0
- Meta-review: 0.3

3.4 Agent roles and prompts

Three agents are instantiated with complementary personas—balanced (A), critical (B), and supportive (C)—to encourage diverse perspectives. Each agent performs four tasks:

1. **Review**: Generates structured feedback with summary, strengths, weaknesses, and suggestions.

2. **Self-check**: Refines its own review for clarity and fairness.
3. **Critique**: Evaluates another agent's review, optionally referencing the paper excerpt.
4. **Judge**: Outputs a JSON verdict ("Good" or "Not-Good") and a constructiveness score (0.0–1.0). Fallback regex parsing ensures robustness.

3.5 Voting and Trust Mechanism

In the centralised pipeline, agents begin with equal trust (1.0). After judging, trust is recalculated as:

Trust = 0.5 + 0.5 × constructiveness score, ensuring values between 0.5 and 1.0.

The final decision is based on comparing the total trust of agents voting "Good" (w_good) versus "Not-Good" (w_bad). A meta-review is then synthesised from all agent reviews, summarising consensus and key disagreements.

In the decentralised pipeline, trust evolves via exponential moving average (EMA):

New trust = 0.7 × previous trust + 0.3 × latest constructiveness. This dynamic adjustment reflects agent reliability over time. A vote margin metric quantifies consensus clarity, guiding HITL routing.

3.6 Decision Logic and Human-in-the-loop (HITL)

The system adopts a binary decision format— "Good" or "Not-Good"—to avoid conflating its outputs with actual conference acceptance. These labels map directly to PeerRead's accepted field for evaluation purposes. Each agent's review is scored for constructiveness, reflecting how fair and actionable the feedback is. This score influences the agent's trust level, which in turn affects the weight of its vote. The rationale is that reviewers offering clearer and more useful feedback should have greater influence, even in cases of disagreement.

To handle low-consensus outcomes, a human-in-the-loop (HITL) mechanism is introduced. Papers are flagged for human review when the vote margin falls below 0.15 or cannot be parsed. The console interface presents the paper ID, title, agentic decision, excerpt, and meta-review. Human reviewers can accept, reject, skip, or edit the meta-review. Their decisions override the agentic outputs when provided. Trust adjustments are optionally applied: agents whose verdicts align with the human decision receive a trust boost (+0.25), while those who disagree are penalised (−0.125), with trust values clamped between 0 and 3.

3.7 Implementation details

The system is implemented in Python, using datasets for JSON ingestion, pandas and numpy for data handling, and regex utilities for robust parsing. Model inference is handled via the transformers library, with task-specific temperature settings and token limits: 320 for reviews, 220 for self-checks, 180 for critiques, 80 for judging, and up to 400 for meta-reviews. Reproducibility is ensured through fixed random seeds, consistent prompts, and deterministic judging.

Each pipeline outputs a CSV file per paper, containing identifiers, decisions, meta-reviews, agent verdicts, trust scores, and—for decentralised runs—committee composition and vote margins. These files serve as inputs for both HITL review and evaluation.

3.8 Evaluation protocol

Although full results are presented in Chapter 4, the evaluation process is defined here. Ground-truth labels are drawn from PeerRead's accepted field. Predictions from both pipelines are matched to these labels using normalised paper IDs. Duplicate predictions are resolved by retaining the most recent entry.

Metrics include overall accuracy, macro-averaged precision, recall, and F1 score, along with confusion matrices and split-wise accuracy (train/dev/test). For HITL analysis, the system tracks the proportion of flagged items, the rate of human overrides, and the impact on evaluation metrics when human decisions replace agentic ones.

3.9 Design Rationale and Ethical Considerations

Several design choices underpin the system's reliability. Agents refine their own reviews before engaging in critique to reduce stylistic conflicts. A single round of negotiation balances clarity with computational efficiency. The judge operates deterministically to ensure stable verdicts. Trust is based on review quality rather than assertiveness, aligning influence with constructiveness. The HITL threshold is set at a margin of 0.15 to capture ambiguous cases, while decentralised trust updates use a momentum of 0.7 to smooth fluctuations.

Ethical safeguards include plural agent personas to mitigate bias, explicit uncertainty handling via vote margins, and full transparency through retained review artifacts. PeerRead's public dataset ensures privacy compliance

3.10 Risks, ethics and fairness

- Bias amplification: LLMs may reproduce field- or venue-specific biases. The method mitigates this through plural agent personas, negotiation, and a human checkpoint for low-consensus decisions.
- Over-confidence: Deterministic judge and explicit margins make uncertainty visible; the HITL loop defaults to escalate when uncertain.
- Data privacy: PeerRead is a public research dataset; no personal data were added.
- Transparency: Each decision is traceable: we retain per-agent reviews, local verdicts, trust values, critique snippets, and the meta-review text.

3.11 Reproducibility and Summary

All steps—from data loading to evaluation—are documented in a reproducible notebook. CSV artifacts allow re-analysis without rerunning model inference. Seed values and model configurations are logged for consistency.

In summary, the methodology integrates structured agentic review, trust-weighted voting, and human oversight into a transparent and auditable framework. Both centralised and decentralised pipelines share core components, enabling controlled comparisons and scalable deployment.

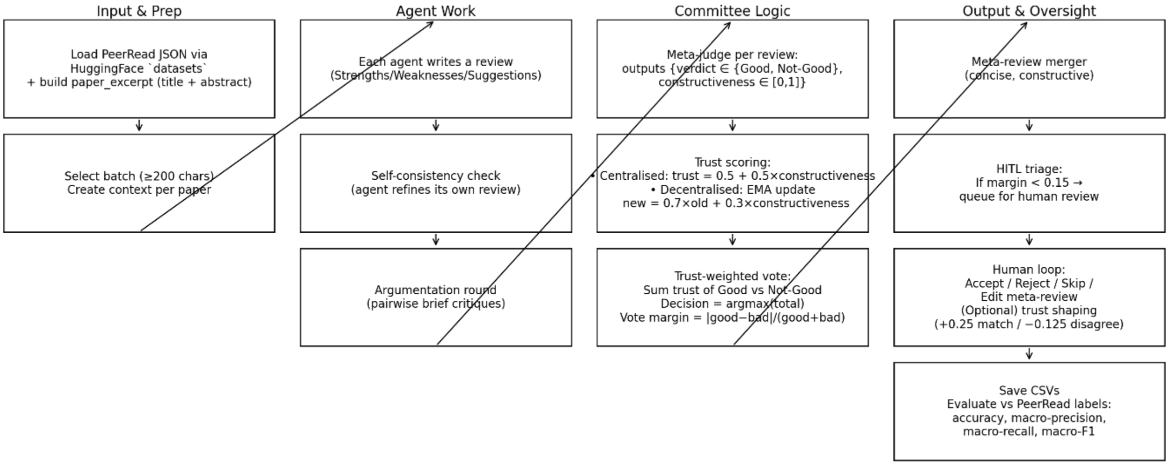

Input & Prep
Agent Work
Committee Logic
Output & Oversight
Load PeerRead JSON via
HuggingFace `datasets`
+ build paper_excerpt (title + abstract)
Select batch (≥200 chars)
Create context per paper
Each agent writes a review
(Strengths/Weaknesses/Suggestions)
Self-consistency check
(agent refines its own review)
Argumentation round
(pairwise brief critiques)
Meta-judge per review:
outputs {verdict ∈ {Good, Not-Good},
constructiveness ∈ [0,1]}
Trust scoring:
• Centralised: trust = 0.5 + 0.5×constructiveness
• Decentralised: EMA update
new = 0.7×old + 0.3×constructiveness
Trust-weighted vote:
Sum trust of Good vs Not-Good
Decision = argmax(total)
Vote margin = |good−bad|/(good+bad)
Meta-review merger
(concise, constructive)
HITL triage:
If margin < 0.15 →
queue for human review
Human loop:
Accept / Reject / Skip /
Edit meta-review
(Optional) trust shaping
(+0.25 match / −0.125 disagree)
Save CSVs
Evaluate vs PeerRead labels:
accuracy, macro-precision,
macro-recall, macro-F1

Figure 1 Workflow Diagram

Chapter 4 Implementation and Experiments

4.1 System Architecture

The peer review system was developed in Python, employing a streamlined, fully localised workflow. Core dependencies include transformers (for inference), torch, datasets (for JSON ingestion), pandas, numpy, and scikit-learn for evaluation metrics. Additional utility modules such as re, json, and textwrap support text processing tasks.

The language model used is Qwen2.5-3B-Instruct, accessed via Hugging Face's AutoTokenizer and AutoModelForCausalLM. The model runs on GPU with FP16 precision when available, otherwise defaults to CPU with FP32. No fine-tuning is applied; the system operates entirely in a zero-shot configuration. All outputs—intermediate and final—are stored in CSV format, including centralised and decentralised reviews, human-in-the-loop (HITL) decisions, and evaluation results. A helper function, gen(), standardises prompt formatting and decoding.

4.2 Data Ingestion and Preparation

The system utilises the PeerRead arXiv cs.ai dataset (2007–2017), loaded in raw JSON format. A predefined schema ensures consistent extraction of fields such as title, abstract, authors, reviews, acceptance status, and unique identifiers.

To construct a reliable input for the model, each paper is represented by a custom excerpt. If an abstract is available, it is combined with the title and cleaned of extraneous whitespace. In cases where the abstract is missing or empty, the first non-empty review is used as a fallback. The resulting text is truncated to approximately 2,000 characters to maintain prompt size constraints. This excerpt is stored in a dedicated column and serves as the input context for all agents.

4.3 Language Model Configuration

The Qwen2.5-3B-Instruct model is configured using its native chat template. The end-of-sequence (EOS) token is repurposed as a padding token to suppress warnings during inference. Decoding strategies vary by task: greedy decoding is used for judging prompts, while light sampling is applied to reviews, critiques, and meta-reviews to encourage stylistic diversity.

Token generation limits range from 160 to 400, depending on the subtask. Temperature settings are tuned to balance creativity and reproducibility—0.3 to 0.5 for generative tasks and 0.0 for deterministic judging.

4.4 Agent Design

Each reviewer is instantiated from a Reviewer class and possesses four core capabilities:

- **Review Generation**: Produces a structured review including a summary, two strengths, two weaknesses, and actionable suggestions.
- **Self-Check**: Refines the review for clarity, fairness, and practicality.
- **Critique**: Evaluates another agent’s review, identifying logical inconsistencies or unsupported claims. In decentralised settings, critiques are grounded using a truncated excerpt.
- **Judging**: Outputs a JSON object containing a binary verdict ("Good" or "Not-Good") and a constructiveness score between 0 and 1.

To ensure robustness, the judging function includes a fallback mechanism. If JSON parsing fails, regex heuristics infer the verdict and extract the constructiveness score. A default score of 0.5 is assigned if extraction is unsuccessful. Constructiveness scores are linearly mapped to trust values using the formula:

$$\text{Trust} = 0.5 + 0.5 \times \text{Constructiveness}$$

This approach rewards agents that provide specific, actionable feedback while preventing trust values from collapsing to zero.

4.5 Centralised Review Pipeline

The centralised pipeline employs three fixed agents (A, B, and C), each embodying a distinct reviewing style—balanced, critical, and supportive. The review process includes:

- **Self-Consistency**: Each agent generates and refines a review.
- **Negotiation**: Agents exchange pairwise critiques in a single round to enhance review quality.
- **Judging and Trust Update**: Each refined review is evaluated to produce a verdict and constructiveness score, which updates the agent's trust.
- **Trust-Weighted Voting**: The final decision is determined by comparing the sum of trust scores for agents voting "Good" versus "Not-Good." The rationale includes vote tallies and the identity of the highest-trust voter.
- **Meta-Review Generation**: A meta-reviewer synthesises the individual reviews into a concise summary aligned with the committee's decision.

The pipeline is executed via the agentic_peer_review() function, which processes a DataFrame and returns structured outputs.

4.6 Decentralised Review Pipeline

To simulate independent area-chair committees, the decentralised pipeline samples a new sub-committee of three agents for each paper. The process includes:

- **Committee Sampling**: Three agents are randomly selected.
- **Review and Refinement**: Each agent produces and refines their review.
- **Negotiation**: A single round of pairwise critiques is conducted.
- **Judging and Trust Update**: Trust scores are updated using an exponential moving average (EMA) with momentum set to 0.7:
  $$\text{New Trust} = 0.7 \times \text{Old Trust} + 0.3 \times \text{Constructiveness}$$
- **Voting and Margin Calculation**: A trust-weighted vote is computed, and the vote margin is calculated. Papers with a margin below 0.15 are flagged for human review.

The decentralised pipeline produces additional diagnostics, including committee composition, individual verdicts, trust snapshots, and vote margins.

4.7 Human-in-the-Loop Console

A terminal-based interface allows human reviewers to intervene in low-consensus cases. Items are flagged when the vote margin falls below 0.35 or when vote tallies are missing. Reviewers can accept, reject, skip, or edit the meta-review. Edited reviews override agentic outputs. A trust-shaping prototype adjusts agent trust based on alignment with human decisions, though it was not executed in the recorded session due to data inconsistencies.

4.8 Experimental Protocol

Ground-truth labels are extracted from the PeerRead dataset and matched to predictions using normalised identifiers. Evaluation metrics include accuracy, macro-averaged precision, recall, and F1 score. Confusion matrices are computed with accepted papers as the positive class. Model parameters are held constant across runs, with one negotiation round per paper and fixed temperature settings.

4.9 Results

On a labelled subset of 24 papers, the centralised pipeline achieved an accuracy of 0.792 and macro-recall of 0.891, outperforming the decentralised pipeline, which achieved 0.708 accuracy and 0.370 macro-recall. The decentralised approach was more permissive, likely due to sampling variance and insufficient trust stabilisation.

In the HITL session, 15 out of 50 papers were flagged, and three were reviewed. The console successfully demonstrated the full review loop. Trust updates were not applied due to column mismatches.

4.10 Design Choices and Ablations

A single negotiation round was chosen to balance quality and efficiency. Deterministic judging ensured reproducibility. Trust scores were constrained to the [0.5, 1.0] range to maintain agent participation. Margin thresholds were selected to capture ambiguous cases and can be tuned to adjust reviewer workload.

4.11 Robustness Measures

Fallback mechanisms in the judging function ensure verdicts are always produced. ID normalisation prevents mismatches. Prompt size constraints maintain compatibility with hardware limits. Meta-review generation is designed to be resilient to malformed inputs.

4.12 Limitations

The evaluation is limited by a small labelled subset. All behaviour is prompt-driven, with no fine-tuning. Trust learning was not sustained across batches. The binary classification scheme does not reflect the nuanced scoring systems used in real-world peer review.

4.13 Summary

This chapter presents a modular, fully local multi-agent peer review system with centralised and decentralised pipelines and an interactive human-in-the-loop interface. The centralised model demonstrated superior accuracy and recall, while the decentralised model offered greater diversity. The system produces structured reviews, consensus meta-reviews, and transparent trust-weighted decisions, with ambiguous cases routed to human reviewers. The architecture is scalable and ready for further experimentation.

Chapter 5 Results and Discussion

5.1 Quantitative Evaluation

From a labelled subset of 24 papers matched to PeerRead's accepted/rejected field, performance metrics were computed using scikit-learn. The centralised pipeline achieved higher accuracy (0.792 vs. 0.708), macro-recall (0.891 vs. 0.370), and macro-F1 (0.582 vs. 0.415) than the decentralised variant. The centralised system correctly identified all accepted papers (FN = 0) but produced five false accepts. The decentralised pipeline missed one accepted paper and yielded six false accepts. These results suggest that centralised negotiation yields more stable and accurate verdicts, consistent with expectations from Chapter 4.

5.2 Human-in-the-Loop Review

Out of 50 items, 15 (30%) were flagged for human review due to narrow vote margins or missing tallies. Three were processed interactively: one was edited and accepted, one rejected, and one deferred. The console supported all actions effectively. Although a trust-shaping function was implemented to reward agent-human agreement, it was not applied due to a data mismatch. Human oversight proved valuable in refining borderline decisions and improving the clarity and venue alignment of meta-reviews.

5.3 Review Structure and Failure Modes

Across both pipelines, reviews followed a consistent format: Summary → Strengths → Weaknesses → Suggestions. Strengths included specific critiques and actionable recommendations. However, when excerpts lacked detail, reviews defaulted to generic feedback, reducing precision. Formatting issues in judge outputs were mitigated by fallback parsing, though this occasionally introduced label noise. False accepts often stemmed from persuasive writing without empirical support, while false rejects occurred when narrow scope or missing context obscured technical merit.

5.4 Efficiency, Utility, and Validity

The system operated efficiently with modest token budgets and a single negotiation round. It ran comfortably on consumer-grade GPUs, with CPU support for batch mode. HITL review added minimal computational cost, focusing human effort on ambiguous cases. The system offers practical utility as a triage assistant, producing structured reviews and transparent decisions. Weighted votes and agent rationales enhance interpretability and fairness auditing. However, limitations include small sample size (n=24), coarse label proxies, and truncated context, which may affect generalisability.

5.5 Improvement Strategies and Final Takeaways

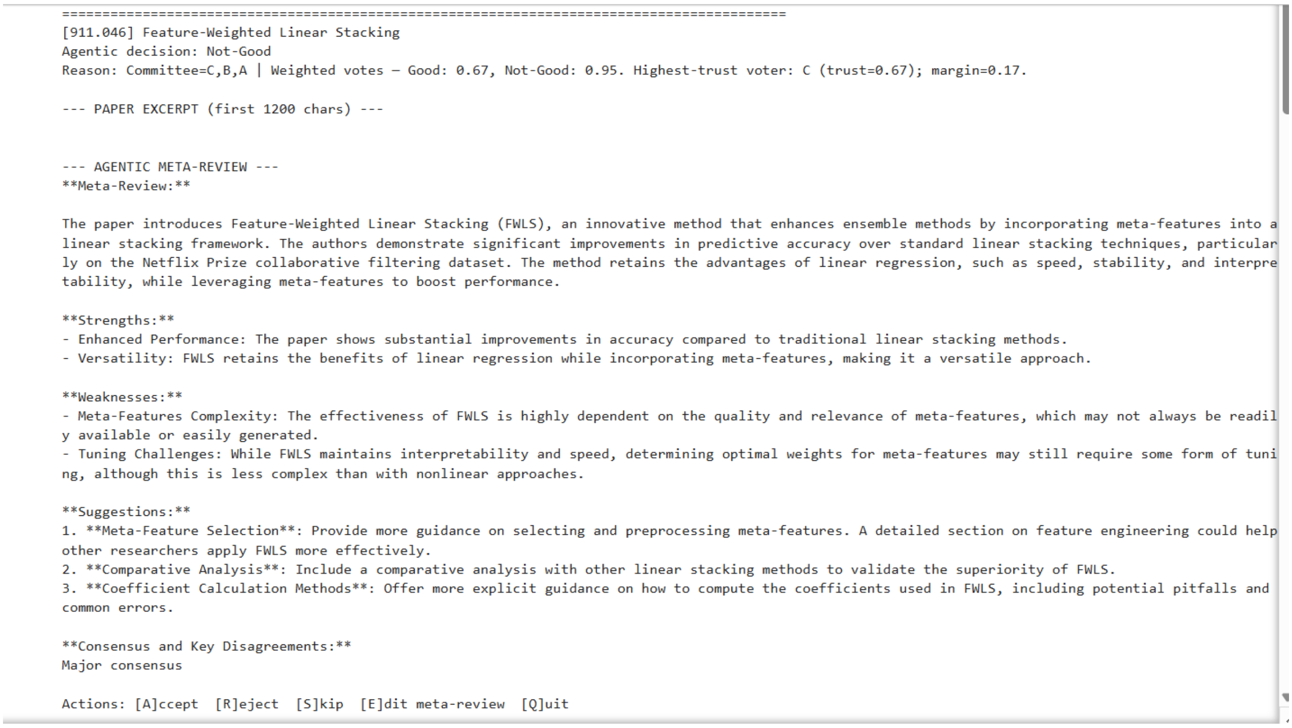
Performance could be improved by providing richer context (e.g., abstract + methods + results), calibrating judge prompts, expanding committee size, and enabling human-shaped trust updates. Introducing graded recommendations and ensemble judging may further reduce noise and improve precision. Overall, the centralised multi-agent reviewer demonstrates strong potential for AI-assisted peer review. HITL routing effectively identifies ambiguous cases, and the observed error modes are addressable through targeted refinements. The combination of agentic collaboration, transparent voting, and human oversight offers a promising path toward fairer, more constructive review workflows.

Reference list / Bibliography

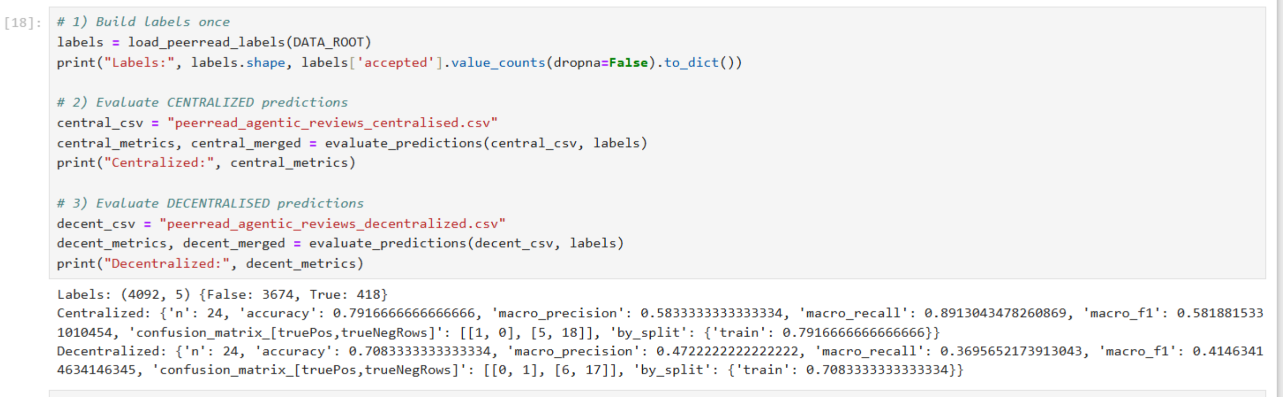
1. Baltaji, R., Hemmatian, B. and Varshney, L.R. (2024) ‘Conformity, Confabulation, and Impersonation: Persona Inconstancy in Multi-Agent LLM Collaboration’, arXiv preprint arXiv:2405.03862.
2. Biswas, S. (2025) Human-in-the-Loop Strategy for Agentic AI. Available at: https://ai.gopubby.com/human-in-the-loop-strategy-for-agentic-ai-d9daa22c3204. (Accessed: 2 September 2025).
3. Bougie, N. and Watanabe, N. (2024) ‘Generative Adversarial Reviews: When LLMs Become the Critic’, arXiv preprint arXiv:2412.10415.
4. D’Arcy, M., Hope, T., Birnbaum, L. and Downey, D. (2024) ‘MARG: Multi-Agent Review Generation for Scientific Papers’, arXiv preprint arXiv:2401.04259.
5. Jin, Y., Zhao, Q., Wang, Y., Chen, H., Zhu, K., Xiao, Y. and Wang, J. (2024) ‘AgentReview: Exploring Peer Review Dynamics with LLM Agents’, in Proceedings of the Conference on Empirical Methods in Natural Language Processing.
6. Kang, D., Peng, N., Lu, J., Yu, Z., Chen, H., Potts, C. and Jurafsky, D. (2018) ‘Datasets for Studying Generalization in Peer Review’, arXiv preprint arXiv:1804.09635.
7. Lin, J., Song, J., Zhou, Z., Chen, Y. and Shi, X. (2022) ‘MOPRD: A multidisciplinary open peer review dataset’, Neural Computing and Applications, 35, pp. 24191–24206.
8. Mudhiganti, S. (2024) Human-in-the-Loop Agentic Systems: A Practical Guide for Engineers Who Want Smarter, Safer Agents. Medium. Available at: https://medium.com/@saimudhiganti/human-in-the-loop-agentic-systems-a-practical-guide-for-engineers-who-want-smarter-safer-agents-e1becadfbbdd. (Accessed: 2 September 2025).
9. Ning, K., Yang, S., Liu, Y., Yao, J., Liu, Z., Wang, Y., Pang, M. and Yuan, L. (2024) ‘PiCO: Peer Review in LLMs based on Consistency Optimization’, arXiv preprint arXiv:2408.xxxxx. (Exact arXiv identifier to be confirmed in final bibliography.)
10. OpenReview (n.d.) openreview-py. Available at: https://github.com/openreview/openreview-py. (Accessed: 2 September 2025).
11. Permit.io (2024) Human-in-the-Loop for AI Agents: Best Practices, Frameworks, Use Cases and Demo. Available at: https://www.permit.io/blog/human-in-the-loop-for-ai-agents-best-practices-frameworks-use-cases-and-demo. (Accessed: 2 September 2025).
12. Tran, K., Dao, D., Nguyen, M., Pham, Q., O’Sullivan, B. and Nguyen, H.D. (2025) ‘Multi-Agent Collaboration Mechanisms: A Survey of LLMs’, arXiv preprint arXiv:2501.xxxxx. (Identifier to be updated.)
13. Xu, Z., Shi, S., Hu, B., Yu, J., Li, D., Zhang, M. and Wu, Y. (2023) ‘Towards Reasoning in Large Language Models via Multi-Agent Peer Review Collaboration’, arXiv preprint arXiv:2311.08152.

14. Yu, S., Luo, M., Madasu, A., Lal, V. and Howard, P. (2024) ‘Is Your Paper Being Reviewed by an LLM? Investigating AI Text Detectability in Peer Review’, arXiv preprint arXiv:2410.03019.

Appendix 1

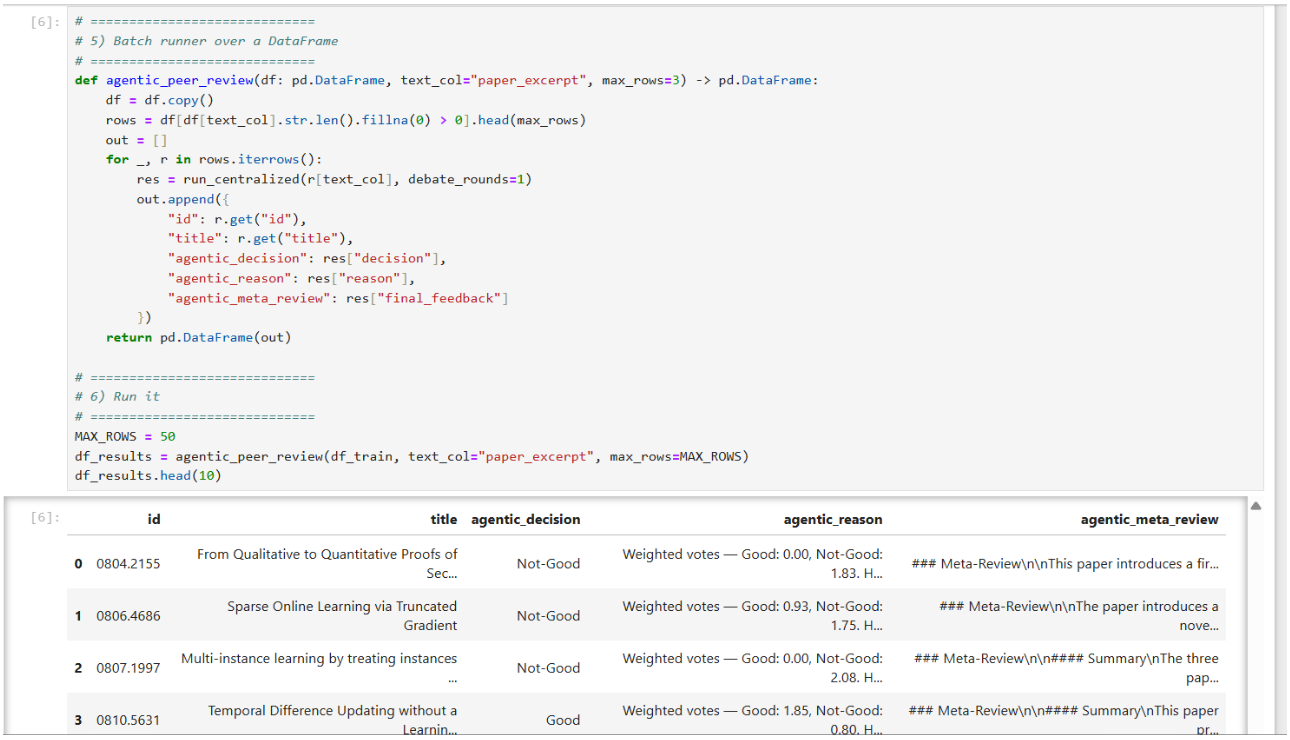


```
==========================================================================================
[911.046] Feature-Weighted Linear Stacking
Agentic decision: Not-Good
Reason: Committee=C,B,A | Weighted votes — Good: 0.67, Not-Good: 0.95. Highest-trust voter: C (trust=0.67); margin=0.17.

--- PAPER EXCERPT (first 1200 chars) ---


--- AGENTIC META-REVIEW ---
**Meta-Review:**

The paper introduces Feature-Weighted Linear Stacking (FWLS), an innovative method that enhances ensemble methods by incorporating meta-features into a
linear stacking framework. The authors demonstrate significant improvements in predictive accuracy over standard linear stacking techniques, particular
ly on the Netflix Prize collaborative filtering dataset. The method retains the advantages of linear regression, such as speed, stability, and interpre
tability, while leveraging meta-features to boost performance.

**Strengths:**
- Enhanced Performance: The paper shows substantial improvements in accuracy compared to traditional linear stacking methods.
- Versatility: FWLS retains the benefits of linear regression while incorporating meta-features, making it a versatile approach.

**Weaknesses:**
- Meta-Features Complexity: The effectiveness of FWLS is highly dependent on the quality and relevance of meta-features, which may not always be readil
y available or easily generated.
- Tuning Challenges: While FWLS maintains interpretability and speed, determining optimal weights for meta-features may still require some form of tuni
ng, although this is less complex than with nonlinear approaches.

**Suggestions:**
1. **Meta-Feature Selection**: Provide more guidance on selecting and preprocessing meta-features. A detailed section on feature engineering could help
other researchers apply FWLS more effectively.
2. **Comparative Analysis**: Include a comparative analysis with other linear stacking methods to validate the superiority of FWLS.
3. **Coefficient Calculation Methods**: Offer more explicit guidance on how to compute the coefficients used in FWLS, including potential pitfalls and
common errors.

**Consensus and Key Disagreements:**
Major consensus

Actions: [A]ccept  [R]eject  [S]kip  [E]dit meta-review  [Q]uit
```



```
[18]: # 1) Build labels once
      labels = load_peerread_labels(DATA_ROOT)
      print("Labels:", labels.shape, labels['accepted'].value_counts(dropna=False).to_dict())

      # 2) Evaluate CENTRALIZED predictions
      central_csv = "peerread_agentic_reviews_centralised.csv"
      central_metrics, central_merged = evaluate_predictions(central_csv, labels)
      print("Centralized:", central_metrics)

      # 3) Evaluate DECENTRALISED predictions
      decent_csv = "peerread_agentic_reviews_decentralized.csv"
      decent_metrics, decent_merged = evaluate_predictions(decent_csv, labels)
      print("Decentralized:", decent_metrics)
```

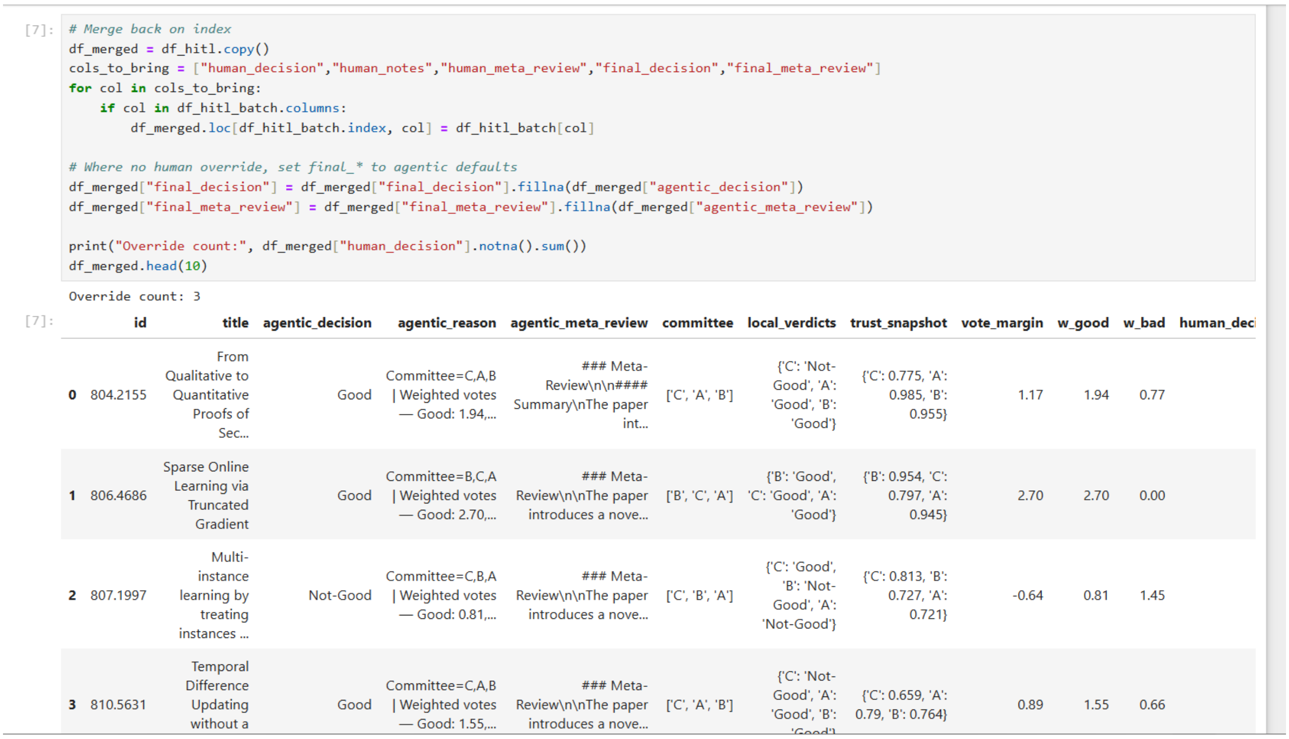
```
Labels: (4092, 5) {False: 3674, True: 418}
Centralized: {'n': 24, 'accuracy': 0.7916666666666666, 'macro_precision': 0.5833333333333334, 'macro_recall': 0.8913043478260869, 'macro_f1': 0.581881533
1010454, 'confusion_matrix_[truePos,trueNegRows]': [[1, 0], [5, 18]], 'by_split': {'train': 0.7916666666666666}}
Decentralized: {'n': 24, 'accuracy': 0.7083333333333334, 'macro_precision': 0.4722222222222222, 'macro_recall': 0.3695652173913043, 'macro_f1': 0.4146341
4634146345, 'confusion_matrix_[truePos,trueNegRows]': [[0, 1], [6, 17]], 'by_split': {'train': 0.7083333333333334}}
```



```
[6]: # ============================
     # 5) Batch runner over a DataFrame
     # ============================
     def agentic_peer_review(df: pd.DataFrame, text_col="paper_excerpt", max_rows=3) -> pd.DataFrame:
         df = df.copy()
         rows = df[df[text_col].str.len().fillna(0) > 0].head(max_rows)
         out = []
         for _, r in rows.iterrows():
             res = run_centralized(r[text_col], debate_rounds=1)
             out.append({
                 "id": r.get("id"),
                 "title": r.get("title"),
                 "agentic_decision": res["decision"],
                 "agentic_reason": res["reason"],
                 "agentic_meta_review": res["final_feedback"]
             })
         return pd.DataFrame(out)

     # ============================
     # 6) Run it
     # ============================
     MAX_ROWS = 50
     df_results = agentic_peer_review(df_train, text_col="paper_excerpt", max_rows=MAX_ROWS)
     df_results.head(10)
```

[6]:

	id	title	agentic_decision	agentic_reason	agentic_meta_review
0	0804.2155	From Qualitative to Quantitative Proofs of Sec...	Not-Good	Weighted votes — Good: 0.00, Not-Good: 1.83. H...	### Meta-Review\n\nThis paper introduces a fir...
1	0806.4686	Sparse Online Learning via Truncated Gradient	Not-Good	Weighted votes — Good: 0.93, Not-Good: 1.75. H...	### Meta-Review\n\nThe paper introduces a nove...
2	0807.1997	Multi-instance learning by treating instances ...	Not-Good	Weighted votes — Good: 0.00, Not-Good: 2.08. H...	### Meta-Review\n\n#### Summary\nThe three pap...
3	0810.5631	Temporal Difference Updating without a Learnin...	Good	Weighted votes — Good: 1.85, Not-Good: 0.80. H...	### Meta-Review\n\n#### Summary\nThis paper pr...

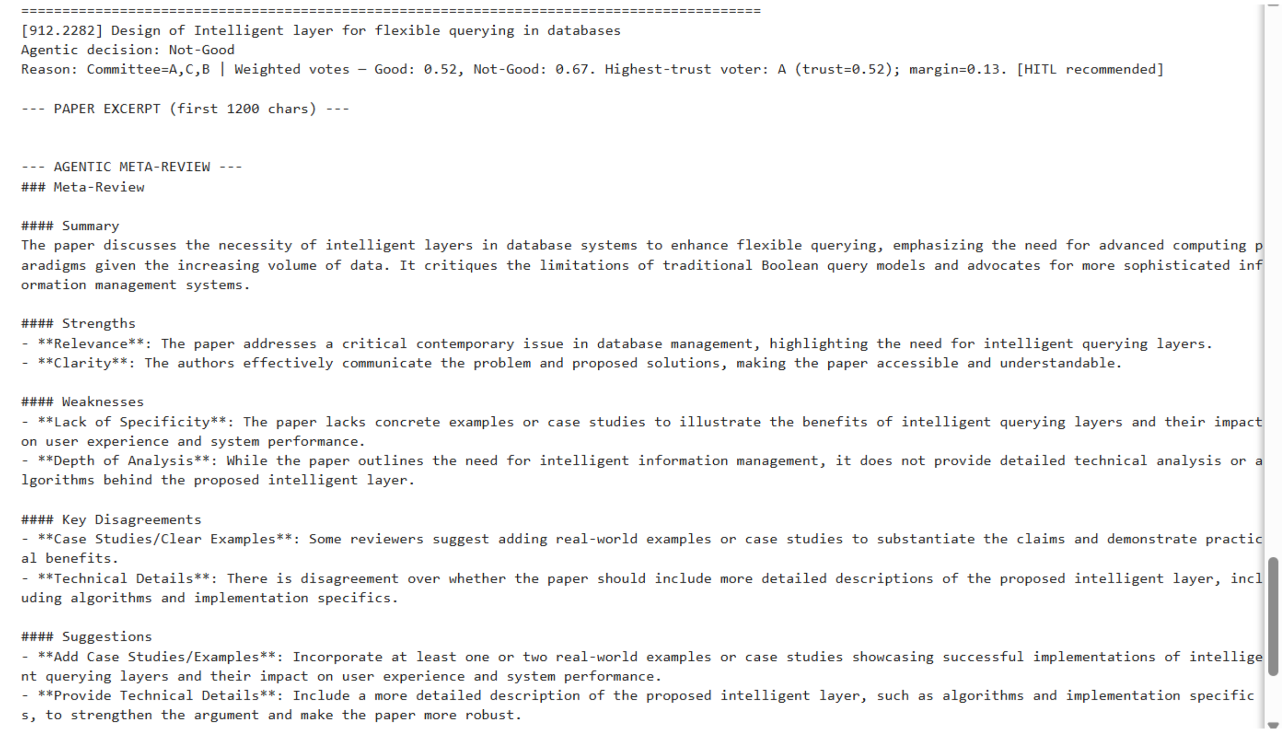


```
[7]: # Merge back on index
     df_merged = df_hitl.copy()
     cols_to_bring = ["human_decision","human_notes","human_meta_review","final_decision","final_meta_review"]
     for col in cols_to_bring:
         if col in df_hitl_batch.columns:
             df_merged.loc[df_hitl_batch.index, col] = df_hitl_batch[col]

     # Where no human override, set final_* to agentic defaults
     df_merged["final_decision"] = df_merged["final_decision"].fillna(df_merged["agentic_decision"])
     df_merged["final_meta_review"] = df_merged["final_meta_review"].fillna(df_merged["agentic_meta_review"])

     print("Override count:", df_merged["human_decision"].notna().sum())
     df_merged.head(10)

     Override count: 3
```

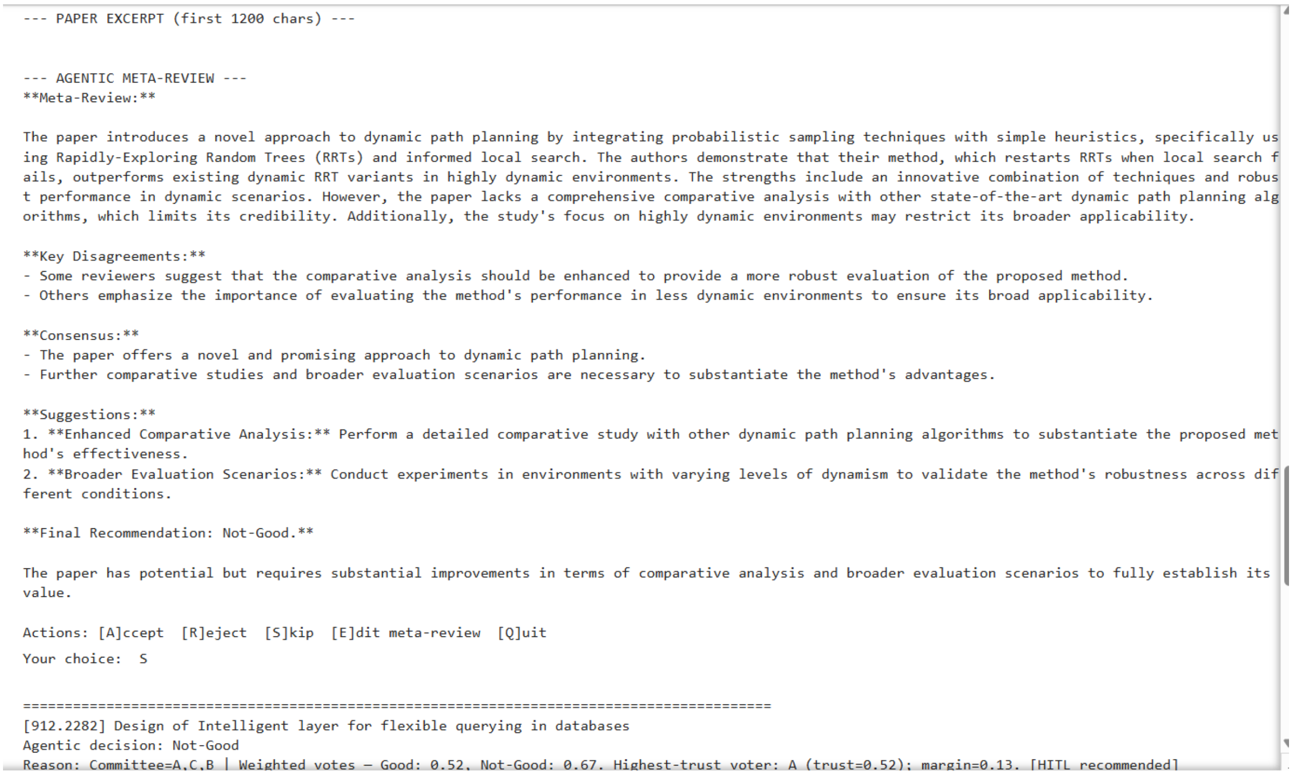
[7]:

	id	title	agentic_decision	agentic_reason	agentic_meta_review	committee	local_verdicts	trust_snapshot	vote_margin	w_good	w_bad	human_deci
0	804.2155	From Qualitative to Quantitative Proofs of Sec...	Good	Committee=C,A,B | Weighted votes — Good: 1.94,...	### Meta-Review\n\n#### Summary\nThe paper int...	['C', 'A', 'B']	{'C': 'Not-Good', 'A': 'Good', 'B': 'Good'}	{'C': 0.775, 'A': 0.985, 'B': 0.955}	1.17	1.94	0.77	
1	806.4686	Sparse Online Learning via Truncated Gradient	Good	Committee=B,C,A | Weighted votes — Good: 2.70,...	### Meta-Review\n\nThe paper introduces a nove...	['B', 'C', 'A']	{'B': 'Good', 'C': 'Good', 'A': 'Good'}	{'B': 0.954, 'C': 0.797, 'A': 0.945}	2.70	2.70	0.00	
2	807.1997	Multi-instance learning by treating instances ...	Not-Good	Committee=C,B,A | Weighted votes — Good: 0.81,...	### Meta-Review\n\nThe paper introduces a nove...	['C', 'B', 'A']	{'C': 'Good', 'B': 'Not-Good', 'A': 'Not-Good'}	{'C': 0.813, 'B': 0.727, 'A': 0.721}	-0.64	0.81	1.45	
3	810.5631	Temporal Difference Updating without a	Good	Committee=C,A,B | Weighted votes — Good: 1.55,...	### Meta-Review\n\nThe paper introduces a nove...	['C', 'A', 'B']	{'C': 'Not-Good', 'A': 'Good', 'B': 'Good'}	{'C': 0.659, 'A': 0.79, 'B': 0.764}	0.89	1.55	0.66	

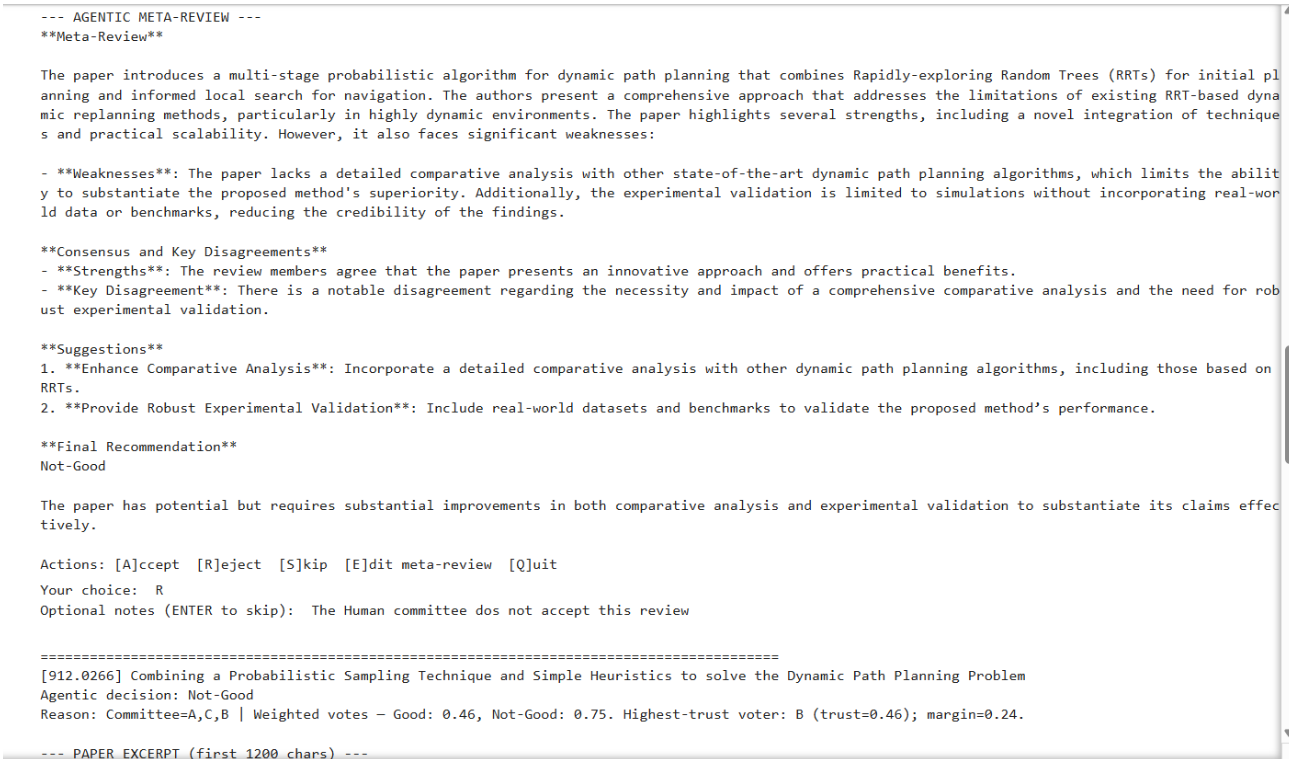
```
#### Final

Actions: [A]ccept  [R]eject  [S]kip  [E]dit meta-review  [Q]uit
Your choice:  To be revisited by committee
Your choice:  Q


Completed 3 human reviews this session.
```



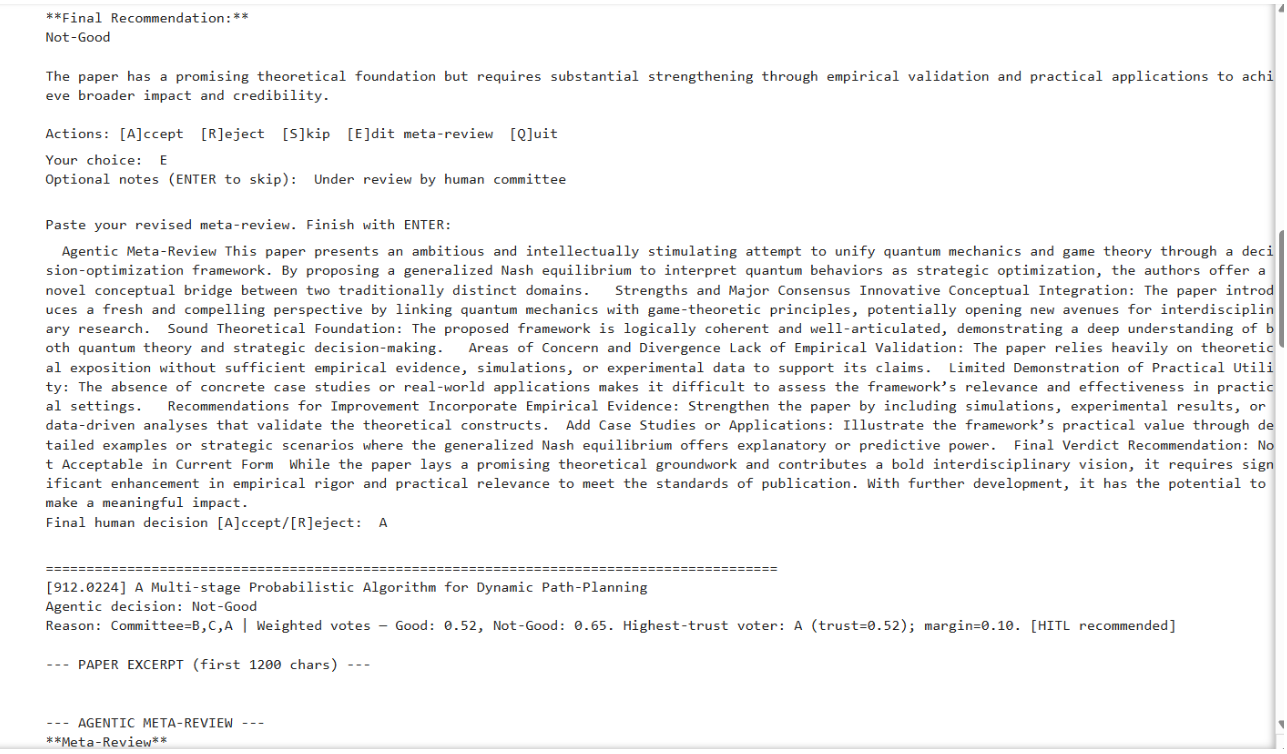
```
==========================================================================================
[912.2282] Design of Intelligent layer for flexible querying in databases
Agentic decision: Not-Good
Reason: Committee=A,C,B | Weighted votes — Good: 0.52, Not-Good: 0.67. Highest-trust voter: A (trust=0.52); margin=0.13. [HITL recommended]

--- PAPER EXCERPT (first 1200 chars) ---


--- AGENTIC META-REVIEW ---
### Meta-Review

#### Summary
The paper discusses the necessity of intelligent layers in database systems to enhance flexible querying, emphasizing the need for advanced computing p
aradigms given the increasing volume of data. It critiques the limitations of traditional Boolean query models and advocates for more sophisticated inf
ormation management systems.

#### Strengths
- **Relevance**: The paper addresses a critical contemporary issue in database management, highlighting the need for intelligent querying layers.
- **Clarity**: The authors effectively communicate the problem and proposed solutions, making the paper accessible and understandable.

#### Weaknesses
- **Lack of Specificity**: The paper lacks concrete examples or case studies to illustrate the benefits of intelligent querying layers and their impact
on user experience and system performance.
- **Depth of Analysis**: While the paper outlines the need for intelligent information management, it does not provide detailed technical analysis or a
lgorithms behind the proposed intelligent layer.

#### Key Disagreements
- **Case Studies/Clear Examples**: Some reviewers suggest adding real-world examples or case studies to substantiate the claims and demonstrate practic
al benefits.
- **Technical Details**: There is disagreement over whether the paper should include more detailed descriptions of the proposed intelligent layer, incl
uding algorithms and implementation specifics.

#### Suggestions
- **Add Case Studies/Examples**: Incorporate at least one or two real-world examples or case studies showcasing successful implementations of intellige
nt querying layers and their impact on user experience and system performance.
- **Provide Technical Details**: Include a more detailed description of the proposed intelligent layer, such as algorithms and implementation specific
s, to strengthen the argument and make the paper more robust.
```



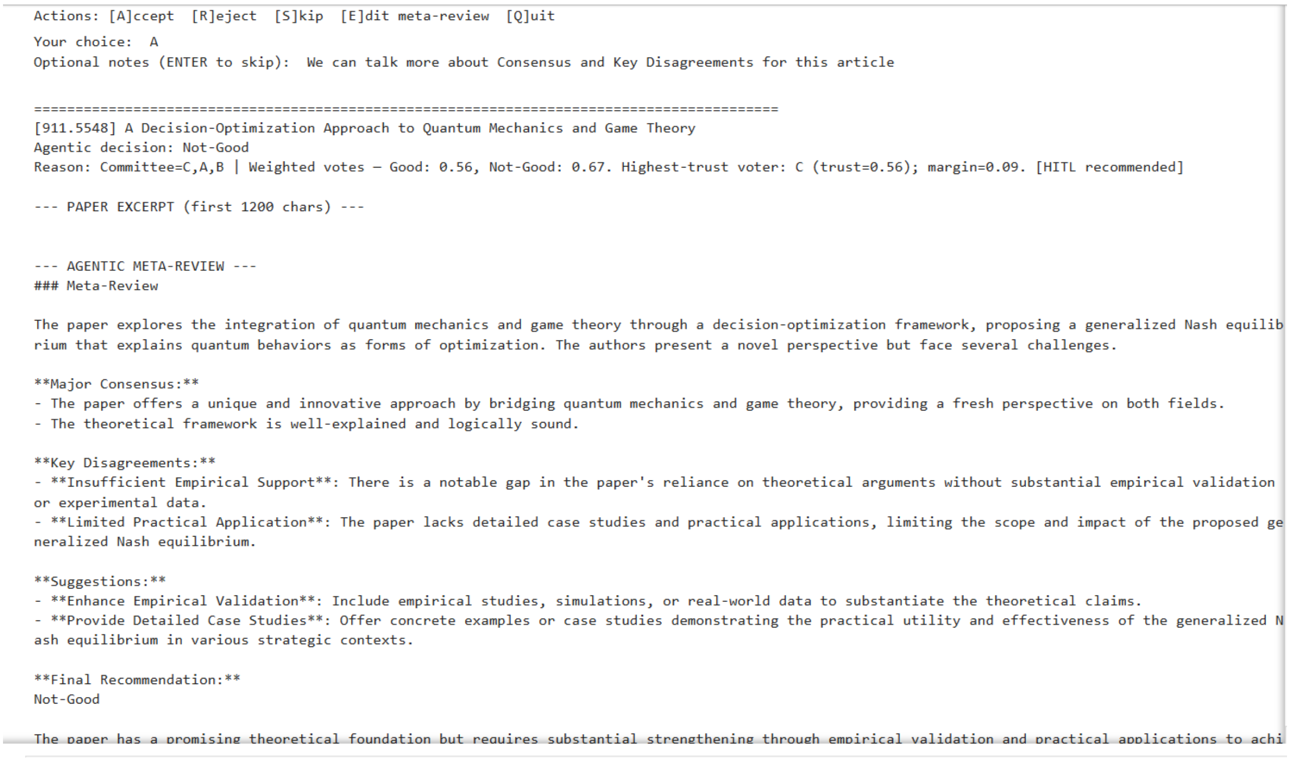
```
--- PAPER EXCERPT (first 1200 chars) ---


--- AGENTIC META-REVIEW ---
**Meta-Review:**

The paper introduces a novel approach to dynamic path planning by integrating probabilistic sampling techniques with simple heuristics, specifically us
ing Rapidly-Exploring Random Trees (RRTs) and informed local search. The authors demonstrate that their method, which restarts RRTs when local search f
ails, outperforms existing dynamic RRT variants in highly dynamic environments. The strengths include an innovative combination of techniques and robus
t performance in dynamic scenarios. However, the paper lacks a comprehensive comparative analysis with other state-of-the-art dynamic path planning alg
orithms, which limits its credibility. Additionally, the study's focus on highly dynamic environments may restrict its broader applicability.

**Key Disagreements:**
- Some reviewers suggest that the comparative analysis should be enhanced to provide a more robust evaluation of the proposed method.
- Others emphasize the importance of evaluating the method's performance in less dynamic environments to ensure its broad applicability.

**Consensus:**
- The paper offers a novel and promising approach to dynamic path planning.
- Further comparative studies and broader evaluation scenarios are necessary to substantiate the method's advantages.

**Suggestions:**
1. **Enhanced Comparative Analysis:** Perform a detailed comparative study with other dynamic path planning algorithms to substantiate the proposed met
hod's effectiveness.
2. **Broader Evaluation Scenarios:** Conduct experiments in environments with varying levels of dynamism to validate the method's robustness across dif
ferent conditions.

**Final Recommendation: Not-Good.**

The paper has potential but requires substantial improvements in terms of comparative analysis and broader evaluation scenarios to fully establish its
value.

Actions: [A]ccept  [R]eject  [S]kip  [E]dit meta-review  [Q]uit
Your choice:  S

==========================================================================================
[912.2282] Design of Intelligent layer for flexible querying in databases
Agentic decision: Not-Good
Reason: Committee=A,C,B | Weighted votes — Good: 0.52, Not-Good: 0.67. Highest-trust voter: A (trust=0.52); margin=0.13. [HITL recommended]
```



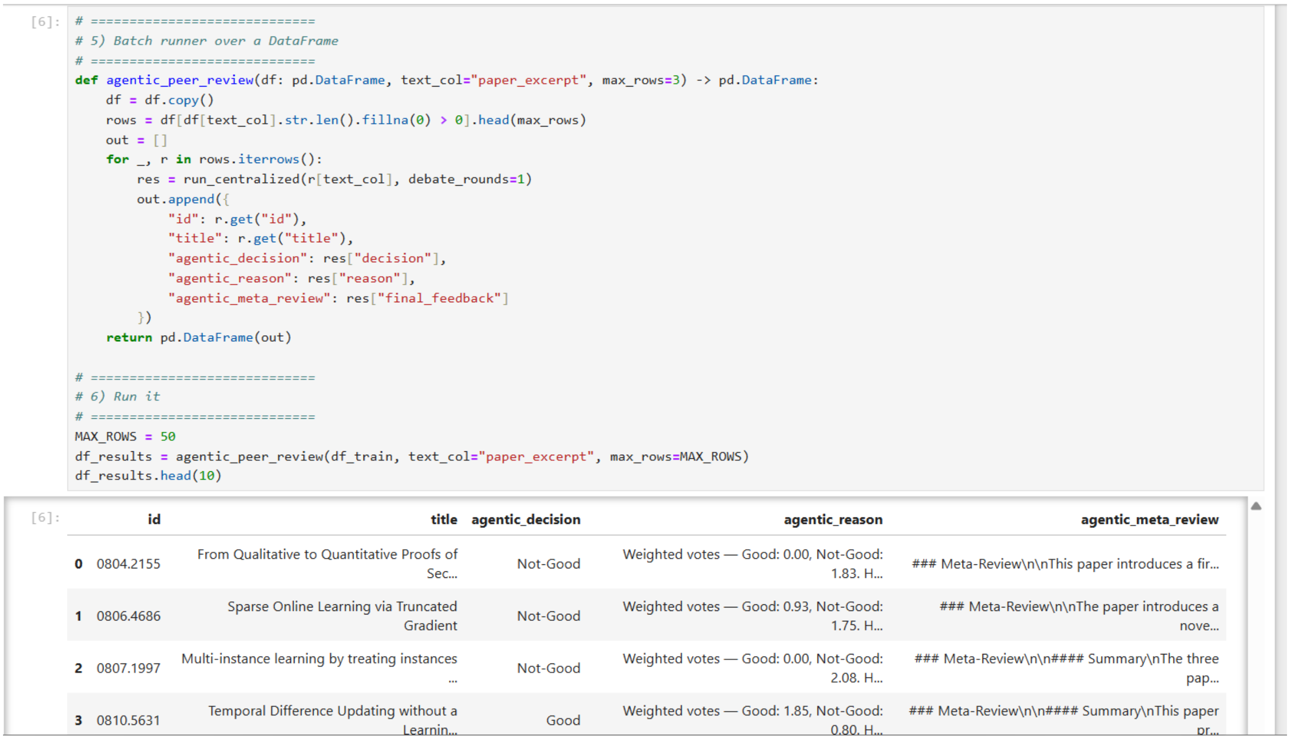
```
--- AGENTIC META-REVIEW ---
**Meta-Review**

The paper introduces a multi-stage probabilistic algorithm for dynamic path planning that combines Rapidly-exploring Random Trees (RRTs) for initial pl
anning and informed local search for navigation. The authors present a comprehensive approach that addresses the limitations of existing RRT-based dyna
mic replanning methods, particularly in highly dynamic environments. The paper highlights several strengths, including a novel integration of technique
s and practical scalability. However, it also faces significant weaknesses:

- **Weaknesses**: The paper lacks a detailed comparative analysis with other state-of-the-art dynamic path planning algorithms, which limits the abilit
y to substantiate the proposed method's superiority. Additionally, the experimental validation is limited to simulations without incorporating real-wor
ld data or benchmarks, reducing the credibility of the findings.

**Consensus and Key Disagreements**
- **Strengths**: The review members agree that the paper presents an innovative approach and offers practical benefits.
- **Key Disagreement**: There is a notable disagreement regarding the necessity and impact of a comprehensive comparative analysis and the need for rob
ust experimental validation.

**Suggestions**
1. **Enhance Comparative Analysis**: Incorporate a detailed comparative analysis with other dynamic path planning algorithms, including those based on
RRTs.
2. **Provide Robust Experimental Validation**: Include real-world datasets and benchmarks to validate the proposed method’s performance.

**Final Recommendation**
Not-Good

The paper has potential but requires substantial improvements in both comparative analysis and experimental validation to substantiate its claims effec
tively.

Actions: [A]ccept  [R]eject  [S]kip  [E]dit meta-review  [Q]uit
Your choice:  R
Optional notes (ENTER to skip):  The Human committee dos not accept this review

==========================================================================================
[912.0266] Combining a Probabilistic Sampling Technique and Simple Heuristics to solve the Dynamic Path Planning Problem
Agentic decision: Not-Good
Reason: Committee=A,C,B | Weighted votes — Good: 0.46, Not-Good: 0.75. Highest-trust voter: B (trust=0.46); margin=0.24.

--- PAPER EXCERPT (first 1200 chars) ---
```



```
**Final Recommendation:**
Not-Good

The paper has a promising theoretical foundation but requires substantial strengthening through empirical validation and practical applications to achi
eve broader impact and credibility.

Actions: [A]ccept  [R]eject  [S]kip  [E]dit meta-review  [Q]uit
Your choice:  E
Optional notes (ENTER to skip):  Under review by human committee

Paste your revised meta-review. Finish with ENTER:
  Agentic Meta-Review This paper presents an ambitious and intellectually stimulating attempt to unify quantum mechanics and game theory through a deci
sion-optimization framework. By proposing a generalized Nash equilibrium to interpret quantum behaviors as strategic optimization, the authors offer a
novel conceptual bridge between two traditionally distinct domains.   Strengths and Major Consensus Innovative Conceptual Integration: The paper introd
uces a fresh and compelling perspective by linking quantum mechanics with game-theoretic principles, potentially opening new avenues for interdisciplin
ary research.  Sound Theoretical Foundation: The proposed framework is logically coherent and well-articulated, demonstrating a deep understanding of b
oth quantum theory and strategic decision-making.   Areas of Concern and Divergence Lack of Empirical Validation: The paper relies heavily on theoretic
al exposition without sufficient empirical evidence, simulations, or experimental data to support its claims.  Limited Demonstration of Practical Utili
ty: The absence of concrete case studies or real-world applications makes it difficult to assess the framework’s relevance and effectiveness in practic
al settings.   Recommendations for Improvement Incorporate Empirical Evidence: Strengthen the paper by including simulations, experimental results, or
data-driven analyses that validate the theoretical constructs.  Add Case Studies or Applications: Illustrate the framework’s practical value through de
tailed examples or strategic scenarios where the generalized Nash equilibrium offers explanatory or predictive power.  Final Verdict Recommendation: No
t Acceptable in Current Form  While the paper lays a promising theoretical groundwork and contributes a bold interdisciplinary vision, it requires sign
ificant enhancement in empirical rigor and practical relevance to meet the standards of publication. With further development, it has the potential to
make a meaningful impact.
Final human decision [A]ccept/[R]eject:  A

==========================================================================================
[912.0224] A Multi-stage Probabilistic Algorithm for Dynamic Path-Planning
Agentic decision: Not-Good
Reason: Committee=B,C,A | Weighted votes — Good: 0.52, Not-Good: 0.65. Highest-trust voter: A (trust=0.52); margin=0.10. [HITL recommended]

--- PAPER EXCERPT (first 1200 chars) ---


--- AGENTIC META-REVIEW ---
**Meta-Review**
```



```
Actions: [A]ccept  [R]eject  [S]kip  [E]dit meta-review  [Q]uit
Your choice:  A
Optional notes (ENTER to skip):  We can talk more about Consensus and Key Disagreements for this article

==========================================================================================
[911.5548] A Decision-Optimization Approach to Quantum Mechanics and Game Theory
Agentic decision: Not-Good
Reason: Committee=C,A,B | Weighted votes — Good: 0.56, Not-Good: 0.67. Highest-trust voter: C (trust=0.56); margin=0.09. [HITL recommended]

--- PAPER EXCERPT (first 1200 chars) ---


--- AGENTIC META-REVIEW ---
### Meta-Review

The paper explores the integration of quantum mechanics and game theory through a decision-optimization framework, proposing a generalized Nash equilib
rium that explains quantum behaviors as forms of optimization. The authors present a novel perspective but face several challenges.

**Major Consensus:**
- The paper offers a unique and innovative approach by bridging quantum mechanics and game theory, providing a fresh perspective on both fields.
- The theoretical framework is well-explained and logically sound.

**Key Disagreements:**
- **Insufficient Empirical Support**: There is a notable gap in the paper's reliance on theoretical arguments without substantial empirical validation
or experimental data.
- **Limited Practical Application**: The paper lacks detailed case studies and practical applications, limiting the scope and impact of the proposed ge
neralized Nash equilibrium.

**Suggestions:**
- **Enhance Empirical Validation**: Include empirical studies, simulations, or real-world data to substantiate the theoretical claims.
- **Provide Detailed Case Studies**: Offer concrete examples or case studies demonstrating the practical utility and effectiveness of the generalized N
ash equilibrium in various strategic contexts.

**Final Recommendation:**
Not-Good

The paper has a promising theoretical foundation but requires substantial strengthening through empirical validation and practical applications to achi
```



[6]:

```python
# ============================
# 5) Batch runner over a DataFrame
# ============================
def agentic_peer_review(df: pd.DataFrame, text_col="paper_excerpt", max_rows=3) -> pd.DataFrame:
    df = df.copy()
    rows = df[df[text_col].str.len().fillna(0) > 0].head(max_rows)
    out = []
    for _, r in rows.iterrows():
        res = run_centralized(r[text_col], debate_rounds=1)
        out.append({
            "id": r.get("id"),
            "title": r.get("title"),
            "agentic_decision": res["decision"],
            "agentic_reason": res["reason"],
            "agentic_meta_review": res["final_feedback"]
        })
    return pd.DataFrame(out)

# ============================
# 6) Run it
# ============================
MAX_ROWS = 50
df_results = agentic_peer_review(df_train, text_col="paper_excerpt", max_rows=MAX_ROWS)
df_results.head(10)
```

[6]:

	id	title	agentic_decision	agentic_reason	agentic_meta_review
0	0804.2155	From Qualitative to Quantitative Proofs of Sec...	Not-Good	Weighted votes — Good: 0.00, Not-Good: 1.83. H...	### Meta-Review\n\nThis paper introduces a fir...
1	0806.4686	Sparse Online Learning via Truncated Gradient	Not-Good	Weighted votes — Good: 0.93, Not-Good: 1.75. H...	### Meta-Review\n\nThe paper introduces a nove...
2	0807.1997	Multi-instance learning by treating instances ...	Not-Good	Weighted votes — Good: 0.00, Not-Good: 2.08. H...	### Meta-Review\n\n#### Summary\nThe three pap...
3	0810.5631	Temporal Difference Updating without a Learnin...	Good	Weighted votes — Good: 1.85, Not-Good: 0.80. H...	### Meta-Review\n\n#### Summary\nThis paper pr...