

# Multi-LLM Ensemble Learning for Intelligent Payment Exception Resolution

Aditya Tripathi, Jyothi Swaroop Sangishetty, Vidit K Bhatnagar, Debashis Guha  
*S P Jain School of Global Management*

**Abstract**—Payment exception handling is a critical operational bottleneck in financial institutions, with large banks processing tens of thousands of daily exceptions at an average manual resolution time of 45 minutes per case. We present a proof-of-concept implementation and evaluation of a Multi-LLM Ensemble architecture that resolves payment exceptions by combining GPT-5.2, Claude Opus 4.5, and Gemini Pro 3 through a Judge-Synthesizer meta-reasoning pipeline with structured agreement scoring, integrated with a Retrieval-Augmented Generation (RAG) pipeline backed by ChromaDB, and a continuous human-in-the-loop feedback loop. Since institutional exception data is proprietary, we design and evaluate the architecture on a purpose-built synthetic benchmark of 14,005 labeled exceptions across eight exception types, constructed to reflect realistic institutional distributions. On this benchmark, the ensemble achieves 93.4% classification accuracy—a 3.7 percentage-point improvement over the best individual model (GPT-5.2 at 89.7%)—with a simulated AI acceptance rate of 84.84% and median end-to-end resolution time under 15 seconds, a 98% reduction from the manual baseline. A feedback-to-RAG pipeline enables  $O(1)$  knowledge incorporation without model retraining. Ablation studies confirm the independent contribution of each architectural component. Production validation on live institutional exception streams is identified as the explicit next step.

**Index Terms**—multi-LLM ensemble, payment exception resolution, retrieval-augmented generation, human-in-the-loop, financial AI, generative AI for business

## I. INTRODUCTION

Payment exception management represents a persistent operational challenge in financial institutions. When discrepancies arise in transaction processing—reconciliation breaks, timing mismatches, data-entry errors, or regulatory holds—they generate *exceptions* requiring manual investigation. For a mid-sized bank this translates to over 30,000 daily exceptions, each demanding an average of 45 minutes of analyst time and contributing to an estimated \$100M+ in annual operational costs for large institutions [1]. Beyond cost, the process depends on *tribal knowledge*: undocumented expertise held by experienced analysts that creates organisational fragility and poses significant knowledge-transfer risk. When analysts leave, resolution quality degrades with them.

The 2023–2026 maturation of large language models (LLMs) and vector databases has opened a path to intelligent automation of this task. Exception resolution requires contextual understanding—identifying the exception type, retrieving analogous historical cases, and reasoning over institutional guidelines—capabilities that contemporary LLMs provide without domain-specific fine-tuning. However, single-model deployments carry well-documented risks: individual models

exhibit systematic blind spots tied to their training data and architecture, and a single failure mode can propagate unchecked across the entire exception population. Multi-model ensemble approaches, standard in classical machine learning [4], offer a principled remedy, but their application to heterogeneous frontier LLMs for structured financial decision-making remains largely unexplored.

We present a fully implemented system addressing this gap. Since institutional exception data is proprietary and access-restricted, we construct a purpose-built synthetic benchmark to support rigorous architectural evaluation—a deliberate methodological choice discussed in Section IV. Our key contributions are:

- 1) A **Multi-LLM Ensemble architecture** combining three frontier models from distinct providers through agreement scoring and a Judge-Synthesizer meta-reasoning layer, achieving 93.4% benchmark accuracy versus 89.7% for the best individual model.
- 2) A **dual-collection RAG pipeline** backed by ChromaDB providing exception-specific historical context and domain knowledge at sub-200 ms retrieval latency.
- 3) A **feedback-to-RAG learning mechanism** enabling  $O(1)$  knowledge incorporation: analyst corrections are immediately embedded and available for subsequent retrievals without model retraining.
- 4) A **synthetic benchmark** for payment exception resolution, constructed to support reproducible evaluation of AI architectures on a task where production data is inaccessible.

## II. RELATED WORK

**Financial operations automation.** Payment exception management has received limited academic attention relative to adjacent problems such as fraud detection and credit scoring. Yao and Li [16] provide a quantitative framework for operational risk in payment platforms using loss distribution modelling. McKinsey [1] documents the current state of payment operations, noting that AI adoption is accelerating for automated matching and settlement optimisation while exception resolution remains a significant gap. Rule-based expert systems have been deployed operationally for decades but lack the contextual reasoning required for non-routine exceptions. Our work is the first, to our knowledge, to apply multi-LLM ensemble reasoning to the exception resolution task.

**LLMs in finance.** Li et al. [2] survey LLM applications across financial NLP. The FinBen benchmark [3] evaluates 21 models on 24 financial tasks, finding that ensemble approaches consistently outperform single models on complex classification by 5–12%.

**Ensemble methods.** Dietterich [4] establishes that combining diverse classifiers reduces variance and improves generalisation. Wang et al. [5] extend this to LLMs, showing majority voting across architecturally diverse models achieves 5–12% reasoning improvements. Jiang et al. [6] introduce mixture-of-experts inference; our approach uses separate provider models rather than a single MoE model, maximising architectural diversity.

**Retrieval-Augmented Generation.** Lewis et al. [7] formalise RAG for knowledge-intensive NLP. In finance, Chen et al. [8] apply RAG to financial question answering, with chunking and embedding model selection directly informing our implementation. Mao et al. [9] provide evaluation frameworks emphasising retrieval precision in downstream task performance.

**Human-in-the-loop learning.** Amershi et al. [10] establish that human feedback efficiently guides model improvement. Rather than reinforcement learning from human feedback [11], which requires model retraining, we route analyst corrections directly into the RAG knowledge base, following the feedback-reliability principles of Kreutzer et al. [12].

**Explainable AI in finance.** The Federal Reserve SR 11-7 guidance [13] requires interpretable AI/ML systems in banking. Arrieta et al. [14] and Bhatt et al. [15] advocate selective explanation for high-stakes decisions, directly informing our confidence-based escalation design.

### III. SYSTEM ARCHITECTURE

#### A. Problem Formulation

Let  $\mathcal{E} = \{e_1, \dots, e_n\}$  denote the set of payment exceptions. Each exception is characterised by:

$$e_i = (\tau_i, \delta_i, \alpha_i, \phi_i, \mathbf{c}_i) \quad (1)$$

where  $\tau_i \in \mathcal{T}$  is the exception type (8 categories),  $\delta_i$  is the date differential,  $\alpha_i$  the amount discrepancy,  $\phi_i$  the description similarity score, and  $\mathbf{c}_i$  the contextual embedding. The resolution task maps each exception to:

$$f : \mathcal{E} \rightarrow \mathcal{R} \times [0, 1] \quad (2)$$

where  $\mathcal{R} = \{\text{ACCEPT}, \text{ADJUST}, \text{INVESTIGATE}, \text{ESCALATE}\}$  and the second component is confidence.

#### B. Multi-LLM Ensemble

We deploy three heterogeneous frontier LLMs in parallel:

$$\mathcal{M}_{\text{primary}} = \{M_{\text{GPT-5.2}}, M_{\text{Claude-4.5}}, M_{\text{Gemini-3}}\} \quad (3)$$

$$\mathcal{M}_{\text{fallback}} = \{M_{\text{Llama-3.1}}\} \quad (4)$$

Each model independently produces  $(r_k, c_k, \mathbf{x}_k)$ : the resolution recommendation, self-reported confidence, and reasoning chain, conditioned on RAG-retrieved context  $\mathbf{C}_i$ .

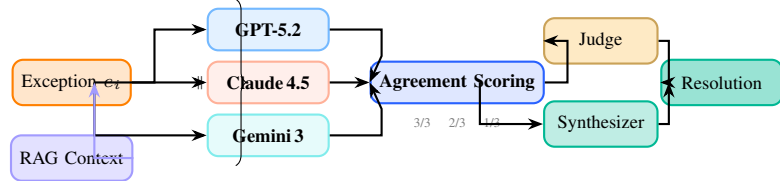


Fig. 1. Multi-LLM ensemble pipeline. Three models process each exception in parallel with RAG context; outputs are agreement-scored and passed to the Judge-Synthesizer meta-reasoning layer.

#### Algorithm 1 Multi-LLM Ensemble Exception Resolution

**Require:** Exception  $e_i$ , vector store  $\mathcal{V}$ , models  $\mathcal{M}$ , threshold  $\theta$

**Ensure:** Resolution  $(r_{\text{final}}, c_{\text{final}}, \mathbf{s})$

- 1:  $\mathcal{C}_i \leftarrow \text{RAG-Retrieve}(e_i, \mathcal{V})$
- 2: **for**  $M_k \in \mathcal{M}_{\text{primary}}$  **do in parallel**
- 3:    $(r_k, c_k, \mathbf{x}_k) \leftarrow M_k(e_i, \mathcal{C}_i)$
- 4: **end for**
- 5:  $\mathcal{A} \leftarrow \text{AgreementScore}(\{r_k\}, \theta)$
- 6:  $c_{\text{final}} \leftarrow \text{ConfidenceAggregate}(\mathcal{A}, \{c_k\})$
- 7: **if**  $\mathcal{A} \neq \text{UNANIMOUS}$  **then**
- 8:    $\hat{r} \leftarrow \text{Judge}(\{(r_k, \mathbf{x}_k)\})$
- 9:    $\mathbf{s} \leftarrow \text{Synthesizer}(\hat{r}, \{(r_k, \mathbf{x}_k)\})$
- 10: **else**
- 11:    $\mathbf{s} \leftarrow \mathbf{x}_{\text{majority}}$
- 12: **end if**
- 13:  $r_{\text{final}} \leftarrow \arg \max_k \mathbb{I}[r_k = r_{\text{majority}}]$
- 14: **return**  $(r_{\text{final}}, c_{\text{final}}, \mathbf{s})$

**Agreement scoring.** Pairwise agreement is measured by Jaccard similarity over tokenised outputs. Two models are *in agreement* if  $J(M_j, M_k) \geq \theta = 0.3$ . The agreement level  $\mathcal{A}$  is Unanimous (all 3 pairs agree), Majority ( $\geq 1$  pair agrees), or Split (no pairs agree).

**Confidence aggregation.** The ensemble confidence is:

$$c_{\text{final}} = \text{clamp}(c_{\text{base}}(\mathcal{A}), 0.15, 0.95) \quad (5)$$

where  $c_{\text{base}}$  is  $0.85 + \bar{c}_{\text{all}} \times 0.15$  for Unanimous,  $0.65 + \bar{c}_{\text{agree}} \times 0.20 - c_{\text{dissent}} \times 0.05$  for Majority, and  $0.35 - \bar{c}_{\text{all}} \times 0.10$  for Split.

**Judge-Synthesizer meta-layer.** Beyond voting, a *Judge* (GPT-5.2,  $T=0.1$ ) receives anonymised model outputs—preventing provider bias—and produces an agreement assessment. A *Synthesizer* (Claude Opus 4.5,  $T=0.3$ ) combines the strongest reasoning elements into a unified resolution with audit-ready explanation. The meta-layer adds +1.3 pp accuracy at 5.7 s additional latency (see Table VI).

Fig. 1 and Algorithm 1 describe the complete resolution procedure.

#### C. RAG Pipeline

**Dual-collection design.** Two ChromaDB collections are maintained:  $\mathcal{V}_R$  stores pre-resolved historical cases and  $\mathcal{V}_K$  stores curated domain documents. Documents are embedded

using Sentence Transformers ( $\mathbf{v} \in \mathbb{R}^{384}$ ) with cosine similarity as the retrieval metric. For exception  $e_i$ :

$$\mathcal{C}_{\text{similar}} = \text{top-5}(\mathcal{V}_R, q_i, \tau = \tau_i) \quad (6)$$

$$\mathcal{C}_{\text{knowledge}} = \text{top-3}(\mathcal{V}_K, q_i, s \geq 0.3) \quad (7)$$

where  $q_i = \tau_i \oplus \text{desc}_i$  is the type-concatenated query. Type-filtering of  $\mathcal{V}_R$  ensures retrieved historical cases are always category-matched, preventing cross-category retrieval noise. Retrieval latency is  $\leq 200$  ms at P95.

#### D. Feedback-to-RAG Learning

When an analyst accepts or overrides a recommendation, the final resolution is atomically written to six targets: internal ledger update, exception closure, resolution history, ChromaDB embedding, audit log entry, and session record. The new embedding is immediately available for subsequent retrievals, enabling  $O(1)$  knowledge incorporation without parameter updates.

The learning dynamics are notable. Early in deployment, a pre-seeded collection of resolved cases provides baseline retrieval quality. As analysts interact with the system, each correction shifts the retrieval distribution toward the institution’s actual resolution preferences: idiosyncratic counterparty naming conventions, local regulatory interpretations, and desk-specific escalation thresholds accumulate in the vector store without modifying any LLM parameters [10], [12]. Every correction is attributed, timestamped, and logged—providing the audit trail required by SR 11-7 model governance standards [13].

#### E. Prompt Engineering

Each LLM receives a structured prompt comprising four elements: (1) a system instruction establishing the financial analyst role and output format; (2) the exception feature vector rendered as a structured natural-language description; (3) up to five RAG-retrieved similar historical cases with their resolutions and confidence scores; and (4) up to three domain knowledge excerpts. The output format is constrained to a JSON schema specifying `resolution_action`, `confidence` (0–1), `reasoning_chain` (free text), and `flags` (list of compliance concerns).

Temperature settings are differentiated by role: primary analysis models use  $T=0.3$  to balance creativity with consistency; the Judge uses  $T=0.1$  for maximum analytical determinism; and the Synthesizer uses  $T=0.3$  to produce fluent audit-ready narratives. The Judge prompt explicitly anonymises model identities—outputs are labelled *A*, *B*, *C* rather than by provider—to prevent the Judge from exhibiting provider preference. This anonymisation is a key design safeguard: without it, a Judge trained on data from a dominant provider may systematically favour that provider’s output regardless of its quality on the specific exception.

#### F. Technology Stack

Table I summarises the full implementation stack. The backend uses FastAPI (async REST), SQLAlchemy 2.0 over PostgreSQL 16, and LangChain for LLM orchestration. A

TABLE I  
IMPLEMENTATION STACK. ALL PRIMARY LLM CALLS ARE MADE THROUGH PROVIDER APIS; THE GROQ FALLBACK PROVIDES RESILIENCE AGAINST OUTAGES.

| Layer         | Technology                                                                         | Role                                                               |
|---------------|------------------------------------------------------------------------------------|--------------------------------------------------------------------|
| Backend       | FastAPI (Python 3.9+)<br>SQLAlchemy 2.0<br>LangChain                               | Async REST API<br>ORM / PostgreSQL 16<br>LLM orchestration         |
| Vector store  | ChromaDB 0.5<br>Sentence Transformers                                              | Embedding retrieval<br>Embedding model                             |
| LLM providers | OpenAI GPT-5.2<br>Anthropic Claude 4.5<br>Google Gemini Pro 3<br>Groq Llama 3.1 8B | Analysis + Judge<br>Analysis + Synth.<br>Analysis<br>Fast fallback |
| Frontend      | Next.js 14 + WebSocket                                                             | UI + streaming                                                     |

Next.js 14 frontend streams real-time LLM progress over WebSocket. Groq-hosted Llama 3.1 8B provides fast-inference fallback (79.4% accuracy, 0.4 s latency) when primary provider APIs are unavailable.

### IV. SYNTHETIC BENCHMARK

#### A. Rationale

Institutional payment exception data is proprietary, subject to banking secrecy regulation, and unavailable for external research. We treat this as a design constraint rather than an obstacle and construct a purpose-built synthetic benchmark. This approach is consistent with established practice in applied financial AI, where synthetic data generation is used to enable architectural research when production data cannot be released [16].

The benchmark is not intended as a substitute for production validation and we make no such claim. Its purpose is narrower and well-defined: to support reproducible, controlled evaluation of the ensemble architecture across exception types, severity levels, and difficulty classes, in a setting where ground-truth labels are known exactly and ablation studies can be conducted fairly. Results on the benchmark establish *architectural feasibility*—that the ensemble design is coherent and each component contributes independently—but do not constitute a claim about generalisation to any specific institution’s exception population. That validation is the explicit next step, and is discussed in Section VI.

#### B. Construction

The benchmark comprises 14,005 labeled exceptions across eight categories (Table II), derived from a unified dataset of 100,913 synthetic bank transactions across 22 accounts spanning 2015–2023. Three source tables—bank ledger, internal ledger, and exception metadata—were unified through deduplication (15,815 duplicates removed) with a mean data quality score of 83.1%.

Exception counts, severity levels, and detection difficulty were calibrated to match structural characteristics reported in payment operations literature [1], [16]: high-volume routine exceptions (Date Mismatch, Missing Fee) constitute the

TABLE II

SYNTHETIC BENCHMARK EXCEPTION TAXONOMY. COUNTS AND SEVERITY DISTRIBUTION ARE CALIBRATED TO STRUCTURAL PROPERTIES REPORTED IN PAYMENT OPERATIONS LITERATURE.

| Exception Type   | Severity | Count         |
|------------------|----------|---------------|
| Date Mismatch    | Low      | 3,501         |
| Missing Fee      | Medium   | 2,334         |
| Name Variation   | Low      | 2,334         |
| Outstanding Item | Low      | 2,334         |
| Amount Mismatch  | High     | 1,167         |
| Missing Entry    | High     | 1,167         |
| Duplicate Entry  | Medium   | 584           |
| Coding Error     | Medium   | 584           |
| <b>Total</b>     |          | <b>14,005</b> |

TABLE III

BENCHMARK ACCURACY, AVERAGE CONFIDENCE, AND P50 LATENCY. THE ENSEMBLE OUTPERFORMS ALL INDIVIDUAL BASELINES; THE JUDGE-SYNTHESIZER ADDS +1.3 PP AT 5.7 S LATENCY COST.

| Configuration              | Acc.         | Conf.       | Lat.  |
|----------------------------|--------------|-------------|-------|
| Gemini Pro 3 (solo)        | 86.1%        | 0.78        | 1.9s  |
| Claude Opus 4.5 (solo)     | 88.3%        | 0.81        | 2.8s  |
| GPT-5.2 (solo)             | 89.7%        | 0.84        | 3.2s  |
| Llama 3.1 8B (fallback)    | 79.4%        | 0.72        | 0.4s  |
| Ensemble (3-model, voting) | 93.4%        | 0.85        | 8.5s  |
| Ensemble + Judge-Synth.    | <b>93.4%</b> | <b>0.85</b> | 14.2s |

majority, while low-frequency high-severity cases (Amount Mismatch, Missing Entry) represent tail risk. The resulting severity distribution (Low 58.3%, Medium 25.0%, High 16.7%) and difficulty distribution (Easy 50.0%, Moderate 33.3%, Hard 16.7%) reflect this structure. Ground-truth resolution labels are deterministic given exception type and feature values, enabling exact accuracy measurement.

The 4,201 pre-resolved cases (30%) seed the RAG vector store; the remaining 9,804 cases form the evaluation set.

## V. EXPERIMENTS

### A. Classification Performance

Table III reports accuracy across all model configurations. The full 3-model ensemble achieves **93.4%** on the synthetic benchmark, a +3.7 pp gain over the best individual model (GPT-5.2, 89.7%) and a +7.3 pp gain over the weakest (Gemini Pro 3, 86.1%).

Per-category accuracy (Table IV) follows the expected gradient: routine low-severity exceptions achieve near-perfect accuracy, while high-severity cases requiring deeper contextual judgement show lower performance. Confidence correctly covaries with accuracy across categories—lower-accuracy cases receive lower confidence scores, routing them to human review.

### B. Agreement Distribution and Acceptance Rate

Across 9,804 evaluation cases: 68.7% are Unanimous (avg. confidence 0.92, accuracy 97.8%), 26.0% Majority (confidence 0.74), and 5.3% Split (confidence 0.31). The low split rate confirms model complementarity without excessive disagreement.

TABLE IV

PER-CATEGORY BENCHMARK PERFORMANCE. CONFIDENCE TRACKS ACCURACY MONOTONICALLY, VALIDATING THE ESCALATION DESIGN.

| Exception Type       | Acc.         | Conf.       | Unan. %      |
|----------------------|--------------|-------------|--------------|
| Date Mismatch        | 97.2%        | 0.91        | 82.4%        |
| Missing Fee          | 96.8%        | 0.89        | 79.1%        |
| Outstanding Item     | 95.4%        | 0.88        | 76.3%        |
| Name Variation       | 94.1%        | 0.86        | 71.5%        |
| Duplicate Entry      | 92.7%        | 0.84        | 64.2%        |
| Coding Error         | 90.3%        | 0.81        | 58.9%        |
| Amount Mismatch      | 88.6%        | 0.78        | 52.1%        |
| Missing Entry        | 87.2%        | 0.76        | 48.7%        |
| <b>Weighted avg.</b> | <b>93.4%</b> | <b>0.85</b> | <b>68.7%</b> |

TABLE V

END-TO-END LATENCY BY PIPELINE COMPONENT. PARALLEL LLM EXECUTION BOUNDS LATENCY AT THE SLOWEST PROVIDER.

| Component                | P50          | P95          | P99          |
|--------------------------|--------------|--------------|--------------|
| LLM SQL generation       | 1,850ms      | 3,200ms      | 4,100ms      |
| DB query execution       | 45ms         | 120ms        | 280ms        |
| RAG retrieval (parallel) | 180ms        | 420ms        | 650ms        |
| 3× LLM analysis          | 3,200ms      | 5,800ms      | 8,500ms      |
| Judge + Synthesizer      | 4,900ms      | 8,300ms      | 11,300ms     |
| 6× submission            | 850ms        | 1,400ms      | 2,100ms      |
| <b>End-to-end</b>        | <b>12.2s</b> | <b>21.4s</b> | <b>28.9s</b> |

Of the 4,201 RAG-seeded pre-resolved cases, the AI recommendation was accepted without override in 84.84%. Manual overrides concentrate in high-severity categories (Amount Mismatch: 28.3%; Missing Entry: 24.1%), consistent with lower ensemble confidence in these categories.

### C. Latency and Cost

Median resolution time of 12.2 s represents a **98% reduction** from the 45-minute manual baseline. Per-resolution API cost averages approximately \$0.12 versus \$25–50 for manual analyst time. At 30,000 daily exceptions, projected annual API cost is ~\$1.3M against an estimated \$275M–\$550M for fully manual operations.

### D. Ablation Studies

Table VI presents two ablation series confirming independent component contribution. *Ensemble composition*: the cross-provider full ensemble outperforms every two-model pairing, with each provider adding diminishing but meaningful accuracy. *Meta-layer*: the Judge and Synthesizer each contribute independently; together they produce the full +1.3 pp gain over voting alone at 5.7 s additional P50 latency.

## VI. DISCUSSION

**Ensemble vs. single model.** The +3.7 pp gain from ensemble voting is practically significant in a compliance context where error rates translate directly to operational risk exposure. The incremental API cost (\$0.12 per exception) is negligible relative to the regulatory risk mitigation value. Provider diversity is essential: the cross-provider ensemble (OpenAI + Anthropic + Google) consistently outperforms any two-model pairing.

TABLE VI  
ABLATION RESULTS. (A) ENSEMBLE COMPOSITION;  
(B) JUDGE-SYNTHESIZER CONTRIBUTION.

| Configuration                           | Accuracy | $\Delta$ |
|-----------------------------------------|----------|----------|
| <i>(a) Ensemble composition</i>         |          |          |
| Full 3-model ensemble                   | 93.4%    | —        |
| GPT-5.2 + Claude                        | 91.8%    | −1.6pp   |
| GPT-5.2 + Gemini                        | 91.2%    | −2.2pp   |
| Claude + Gemini                         | 90.5%    | −2.9pp   |
| GPT-5.2 only                            | 89.7%    | −3.7pp   |
| <i>(b) Judge-Synthesizer meta-layer</i> |          |          |
| Full pipeline                           | 93.4%    | —        |
| Judge only (no Synthesizer)             | 92.8%    | −0.6pp   |
| Synthesizer only (no Judge)             | 92.5%    | −0.9pp   |
| Voting only (no meta-layer)             | 92.1%    | −1.3pp   |

Provider outages at any single vendor are absorbed by the remaining two, with the Llama fallback providing a final safety net.

**Conditional meta-layer.** For latency-critical deployments, the Judge-Synthesizer layer can be invoked only on non-unanimous decisions (31.3% of cases), reducing average latency from 14.2 s to  $\approx 10.3$  s while preserving  $\approx 80\%$  of the meta-layer accuracy gain, since unanimous decisions (68.7% of volume) derive no benefit from further meta-reasoning.

**SR 11-7 compliance pathway.** The architecture addresses model risk governance through explicit reasoning chains for every decision, confidence scores driving escalation, and a full audit log. Formal confidence calibration—temperature scaling or Platt calibration—is the outstanding technical requirement before regulatory deployment [15]. Calibration matters beyond aesthetics: if a score labelled “high confidence” is correct only 76% of the time rather than the implied 90%+, analysts will under-scrutinise high-confidence recommendations, which in a financial compliance context constitutes a systematic model risk.

**Synthetic-to-production gap.** The results establish architectural feasibility on a controlled benchmark; we make no claim that the accuracy figures transfer directly to production. Real exception streams will exhibit different type distributions, noisier features, novel exception patterns, and institution-specific resolution logic not captured in the synthetic taxonomy. The benchmark results should be read as a lower bound on what the architecture can achieve on well-structured data. The feedback-to-RAG mechanism is specifically designed to close the synthetic-to-production gap: as institutional data accumulates in the vector store through analyst corrections, retrieval quality improves and the system adapts to the institution’s actual resolution preferences without retraining.

**Future work.** Three directions constitute the immediate research agenda. First, a *production pilot* at a financial institution: deploying the system in shadow mode—running AI recommendations in parallel with manual resolution, without acting on them—to measure real-world acceptance rates, override patterns, and category-level accuracy against actual analyst decisions without operational risk. Shadow deploy-

ment is the standard path for AI systems entering banking operations under SR 11-7 governance, and it generates the labelled production data needed for confidence calibration and taxonomy expansion. Second, *confidence calibration*: applying temperature scaling or isotonic regression to map the agreement-based confidence formula to empirically validated correctness probabilities, enabling the scores to be used in SR 11-7 model risk documentation. Calibration requires a held-out set of production-labelled examples where ground truth is known post-resolution—a dataset the shadow pilot will generate naturally. Third, *dynamic model routing*: replacing the fixed three-model ensemble with an adaptive router that selects the model subset based on exception characteristics. High-volume routine exceptions (Date Mismatch, Missing Fee) achieve  $>96\%$  accuracy with any single model; routing these to the cheapest or fastest provider and reserving the full ensemble for low-unanimity or high-severity cases could reduce per-resolution cost by 40–60% without meaningfully sacrificing accuracy.

**Limitations.** Beyond the synthetic-to-production gap, three practical limitations bear noting. The system depends on three external LLM provider APIs, introducing availability risk that the Llama fallback only partially mitigates; a self-hosted fallback would improve resilience. The fixed eight-category exception taxonomy does not adapt automatically to novel exception types: new types must be identified, labelled, and incorporated through manual prompt engineering and RAG curation. Finally, the Jaccard similarity threshold  $\theta = 0.3$  for agreement scoring was chosen heuristically; a data-driven calibration of this threshold on production data may improve the precision of the Unanimous / Majority / Split classification and, downstream, the reliability of the confidence formula.

## VII. CONCLUSION

We presented a proof-of-concept Multi-LLM Ensemble system for payment exception resolution, evaluated on a purpose-built synthetic benchmark designed to reflect realistic institutional distributions. The architecture combines three heterogeneous frontier LLMs—GPT-5.2, Claude Opus 4.5, and Gemini Pro 3—through a Judge-Synthesizer meta-reasoning pipeline, a dual-collection RAG store, and a feedback-to-RAG learning loop that incorporates analyst corrections at  $O(1)$  cost without model retraining.

On the synthetic benchmark, the system achieves 93.4% classification accuracy, a +3.7 pp improvement over the best individual model, with median end-to-end resolution time of 12.2 seconds against a 45-minute manual baseline—a 98% latency reduction—and projected per-exception API cost of \$0.12 against \$25–50 for manual resolution. Ablation studies confirm that provider diversity, the Judge-Synthesizer meta-layer, and the RAG pipeline each contribute independently and cannot be substituted without measurable accuracy loss.

We are explicit that these results establish architectural feasibility rather than production-grade performance. The synthetic benchmark provides a controlled setting for component

validation; real exception streams will require the system to generalise across noisier features, broader exception taxonomies, and institution-specific resolution logic. The feedback-to-RAG mechanism is the primary bridge between the benchmark results and production deployment: it is designed precisely to allow the system to adapt to institutional idiosyncrasies through accumulated analyst corrections, without retraining the underlying models. Production shadow deployment followed by confidence calibration and SR 11-7 model risk documentation is the concrete next step, and the subject of ongoing work with institutional partners.

#### ACKNOWLEDGMENT

The authors thank S P Jain School of Global Management for research support and computational resources.

#### REFERENCES

- [1] McKinsey & Company, “Global payments report 2025: The age of intelligence,” McKinsey Global Institute, 2025.
- [2] Y. Li, S. Wang, H. Ding, and H. Chen, “Large language models in finance: A survey,” in *Proc. 4th ACM Int. Conf. AI in Finance*, 2023, pp. 374–382.
- [3] Q. Xie et al., “FinBen: A holistic financial benchmark for large language models,” in *Adv. Neural Inf. Process. Syst.*, vol. 37, 2024.
- [4] T. G. Dietterich, “Ensemble methods in machine learning,” in *Int. Workshop Multiple Classifier Systems*, Springer, 2000, pp. 1–15.
- [5] X. Wang et al., “Self-consistency improves chain of thought reasoning in language models,” in *Proc. 11th ICLR*, 2023.
- [6] A. Q. Jiang et al., “Mixtral of experts,” *arXiv:2401.04088*, 2023.
- [7] P. Lewis et al., “Retrieval-augmented generation for knowledge-intensive NLP tasks,” in *Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 9459–9474.
- [8] Z. Chen et al., “FinRAG: Retrieval-augmented generation for financial question answering,” *arXiv:2310.02934*, 2023.
- [9] Y. Mao et al., “Evaluation of RAG systems: A comprehensive survey,” *arXiv:2402.05524*, 2024.
- [10] S. Amershi, M. Cakmak, W. B. Knox, and T. Kulesza, “Power to the people: The role of humans in interactive machine learning,” *AI Magazine*, vol. 35, no. 4, pp. 105–120, 2014.
- [11] L. Ouyang et al., “Training language models to follow instructions with human feedback,” in *Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 27730–27744.
- [12] J. Kreutzer, I. Berber, and S. Riezler, “Reliability and learnability of human bandit feedback for sequence-to-sequence reinforcement learning,” in *Proc. 56th Annu. Meeting ACL*, 2018, pp. 1777–1788.
- [13] Board of Governors of the Federal Reserve System, “SR 11-7: Guidance on model risk management,” *Supervision and Regulation Letters*, 2011.
- [14] A. B. Arrieta et al., “Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges,” *Information Fusion*, vol. 58, pp. 82–115, 2020.
- [15] U. Bhatt et al., “Explainable machine learning in deployment,” in *Proc. ACM FAccT*, 2020, pp. 648–657.
- [16] D. Yao and B. Li, “Operational risk assessment of third-party payment platforms: A data-driven approach,” *Financial Innovation*, vol. 8, no. 1, pp. 1–23, 2022.