Scalable and Privacy-Preserving Customer Support Automation in Banking Using Small Language Model Powered Multi-Agent Architectures

Anonymous ACL submission

Abstract

Large Language Model (LLM) deployments in regulated banking environments face critical barriers: prohibitive cloud inference costs, data residency requirements under PCI-DSS and GDPR, and risks of factual inconsistency. This paper presents an end-to-end SLM-powered multi-agent customer support system for banking, built around models with under 2 billion parameters and deployed entirely on-premises. Our architecture integrates a TinyBERT intent classifier (93.95% accuracy), a hybrid Retrieval-Augmented Generation (RAG) pipeline with cross-encoder reranking (+17.2% MRR@5), a MetaGPT-inspired publish-subscribe orchestration framework, and a three-layer MiniLM safety guardrail (98.645% accuracy, 250 ms latency). Gemma-3-1B-IT within the publish-subscribe architecture achieves a GPT-4o-mini judge score of 2.99/5, matching Gemini-2.5-Flash-Lite (2.90/5), while running at zero API cost and full data residency compliance. Our results demonstrate that domain adaptability and retrieval strategy outweigh raw parameter count, and that task-specific encoder models dramatically outperform general-purpose safety models such as Llama Guard 1B for banking compliance.

1 Introduction

The increasing replacement of human customer service assistants with AI agents in banking reflects a broader industry push toward delivering instant, scalable, and always-available support (Li et al., 2024). Unlike human-operated systems, AI agents can process thousands of simultaneous customer interactions while providing consistent responses, reducing operational costs, and improving response times for critical financial queries. However, in highly regulated domains such as banking, speed and automation alone are insufficient: AI systems must also provide factually correct, auditable, and privacy-preserving information (Gao et al., 2024). Incorrect financial guidance, inconsistent responses, or data leakage can result in severe regulatory penalties, reputational damage, and erosion of customer trust. Consequently, the transition from human assistants to AI agents has heightened the importance of developing trustworthy conversational systems that balance efficiency with strict guarantees of accuracy, security, and compliance.

The rapid adoption of conversational AI in banking has exposed a fundamental tension: state-of-the-art LLMs deliver impressive fluency but are incompatible with the privacy, cost, and regulatory constraints governing financial services. Cloud-based LLM APIs require transmitting customer queries to external servers, violating PCI-DSS data residency requirements and GDPR Article 44 cross-border transfer restrictions. At scale, API costs for GPT-4o-class models (∼$525/month for 10,000 daily queries) represent a significant operational burden for retail banks. Furthermore, general-purpose LLMs produce hallucinated financial advice with non-trivial frequency, creating compliance liability in regulated environments (Zheng et al., 2024). An analysis of 800 real-world LLM-generated QA pairs found that approximately 31.4% of responses contained hallucinated information, with 62.5% of those classified as fact-conflicting hallucinations that directly contradict the provided context, errors with severe reputational and legal consequences in banking.

Small transformer-based decoder-only models with 100M–5B parameters have recently closed the performance gap with larger counterparts on structured tasks through overtraining and data curation (Lu et al., 2025). Deployed on-premise, SLMs eliminate cloud data exposure while reducing inference cost by orders of magnitude. However, no prior work has demonstrated a complete SLM-powered multi-agent architecture validated end-to-end in a banking customer service setting.

This paper addresses four research questions:

- **RQ1:** How do SLM-based agents perform relative to LLM baselines across accuracy, latency, and cost?
- **RQ2:** Which RAG strategies most effectively improve factuality for SLMs?
- **RQ3:** Which agent orchestration patterns best support task allocation and scalability?
- **RQ4:** Which evaluation metrics most reliably measure factuality and compliance in SLM systems?

The primary contributions of this study are structured to directly address each research question:

- *(RQ1)* The first end-to-end SLM multi-agent banking chatbot evaluated on real FAQ data, demonstrating on-premise deployment at zero cloud API cost, with Gemma-3-1B-IT matching Gemini-2.5-Flash-Lite (2.99 vs. 2.90/5 overall) at a fraction of the cost.
- *(RQ1)* Empirical evidence that task-specific encoder models (MiniLM, TinyBERT) outperform general-purpose safety models (Llama Guard 1B) by 38+ percentage points in accuracy while being 134–210× faster.
- *(RQ2)* A cross-encoder reranking strategy delivering +17.2% MRR@5 and +13.8% nDCG@5 over dense-only retrieval with no recall penalty, identified as the sole driver of ranking improvement in structured FAQ corpora.
- *(RQ3)* Publish-subscribe agent orchestration reducing inter-agent latency by 20.5% and context token usage by 7.4% over the Supervisor-only pattern.
- *(RQ4)* A multi-dimensional evaluation framework demonstrating that lexical metrics (BLEU, ROUGE) actively invert model rankings in banking QA, while LLM-as-judge with two independent models provides the only reliable quality signal.
- Three rigorously curated and human-validated domain-specific datasets (intent classification, safety classification, general Q&A), each validated on 380 manually reviewed samples.

The remainder of this paper is structured as follows. Section 2 reviews related work. Section 3 presents the system architecture, datasets, and training methodology. Section 4 describes the evaluation framework. Section 5 reports and analyses all experimental results. Section 6 discusses future directions, and Section 6 concludes. Supplementary examples and tables are provided in Appendix A and Appendix B respectively.

2 Related Work

2.1 Small Language Models

TinyLlama (1.1B) (Zhang et al., 2024) and H2O-Danube3-500M (Pfeiffer et al., 2024) demonstrate that overtraining on large token counts enables compact models to approach larger counterparts on reasoning tasks. Phi-3-mini rivals Mixtral 8x7B on MT-Bench and MMLU benchmarks (Abdin et al., 2024). A comprehensive survey by Lu et al. (2025) confirms SLMs are approaching LLM-level performance on structured tasks through quantization and parameter-efficient fine-tuning, while Belcak et al. (2025) argues SLMs are the natural fit for repetitive, structured agentic tasks, a hypothesis this work validates in the banking domain.

2.2 Multi-Agent Orchestration

CAMEL (Li et al., 2023) introduced role-playing communication between agents but suffers from role-flipping instability. MetaGPT (Hong et al., 2024) addressed this through standardised operating procedures and a global message pool, which we adapt for customer service. ChatDev (Qian et al., 2024) demonstrated agent specialisation for structured workflows.

2.3 Retrieval-Augmented Generation

Gao et al. (2024) survey RAG paradigms from naive to modular configurations. Hybrid retrieval combining dense embeddings with BM25 improved MAP@20 in knowledge-intensive tasks (Wang et al., 2024). Cross-encoder reranking by Nogueira and Cho (2019) demonstrated robust gains in passage retrieval, consistent with our ablation results.

2.4 LLMs in Finance

Li et al. (2024) surveys LLM applications in finance, highlighting hallucination risks. FinTextQA (Chen et al., 2024) shows BLEU and ROUGE are insufficient for financial QA evaluation, a finding this work replicates and extends. IntelliChain (Jiang et al., 2025) addresses financial regulation NLP but is limited to Chinese regulatory data, leaving the English banking customer-service domain under-explored.

3 Methodology

3.1 System Architecture

Figure 1 illustrates the high-level pipeline. User queries are first classified into three routing cate-

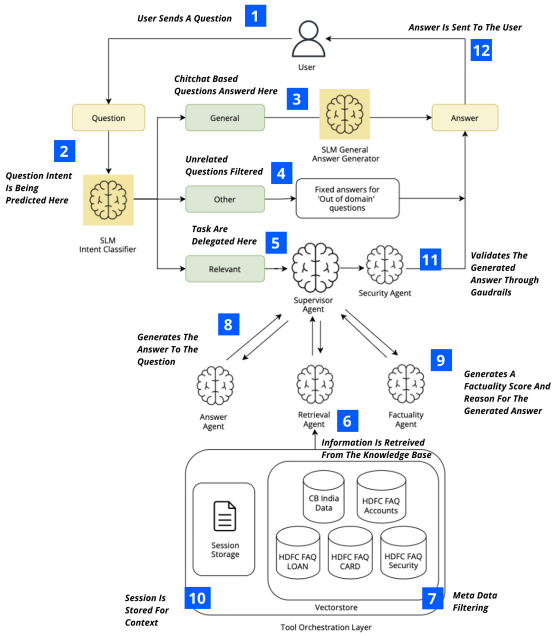

Figure 1: High-level system architecture. Queries are routed by intent class to a fine-tuned SLM (General), multi-agent workflow (Relevant), or fixed response (Other).

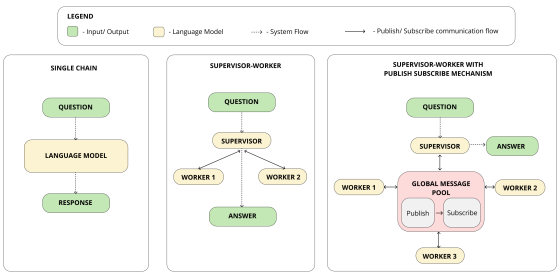

Figure 2: Three agent architectures evaluated: Single Chain (left), Supervisor-Worker (centre), and Publish-Subscribe (right). The publish-subscribe mechanism routes only subscribed message types to each agent, eliminating redundant token overhead.

Table 1: System component summary: selected models and empirical justifications derived from ablation results.

Component	Model	Justification
Intent Classif.	TinyBERT (14.5M)	Best latency–accuracy trade-off; 28 ms/query (Sec. 5.1)
General Q&A	OPT-1.3B (1.3B)	Highest LLM judge score (7.00/10) on Q&A task (Sec. 5.4)
Safety Guard.	MiniLM (66M)	98.6% accuracy at 250 ms vs. 52–60% for Llama Guard (Sec. 5.2)
Agent Backbone	Gemma-3-1B-IT (1B)	Best factuality + speed in multi-call agent context (Sec. 5.5)
Retrieval	Hybrid+Rerank	+17.2% MRR@5 vs. dense-only (Sec. 5.3)
Orchestration	Pub-Subscribe	20.5% latency reduction vs. Supervisor-only (Sec. 5.5)

gories by a lightweight intent classifier: *General* (greetings and chit-chat, answered by a fine-tuned SLM directly), *Relevant* (banking-specific queries routed to the multi-agent workflow), and *Other* (out-of-domain queries answered with a fixed deflection response).

The multi-agent workflow (Figure 2) implements three architectures of increasing complexity: (1) **Single Chain**, a direct LLM+RAG pipeline as the baseline; (2) **Supervisor-Worker**, centralised task delegation to specialised agents; and (3) **Publish-Subscribe**, a MetaGPT-inspired (Hong et al., 2024) global message pool where agents subscribe only to relevant message types, reducing token waste.

The five specialised agents are: a **Supervisor Agent** that parses intent and publishes structured task descriptions to the global message pool; a **Retrieval Agent** that queries an ensemble retriever combining dense FAISS embeddings with BM25, applies cross-encoder reranking, and publishes the top-3 ranked chunks; a **Generation Agent** (Gemma-3-1B-IT with 3-shot prompting) that generates the candidate response conditioned on retrieved context; a **Factuality Agent** (SLM-as-judge) that scores factual grounding against retrieved context on a 1–5 scale and returns a JSON object with `score` and `reason` fields, responses scoring below 3 trigger a fallback deflection rather than delivery of a potentially hallucinated answer; and a **Security Agent** (fine-tuned MiniLM classifier) that intercepts responses containing PII disclosures, unqualified financial advice, or competitor mentions before delivery. Appendix A.1 provides a worked trace of a representative query through the full Publish-Subscribe pipeline.

Table 1 summarises the final selected model for each component, together with the empirical justification.

3.2 Datasets and Pre-processing

The system relies on both publicly extracted and synthetically curated datasets to ensure domain specificity and generalisation.

3.2.1 Data Extraction.

We extracted the most recent FAQs and policy documents from the official HDFC Bank of India and Reserve Bank of India (RBI) websites using the BeautifulSoup library; only the latest policy documents were retained to ensure temporal consistency, yielding approximately 2,500 passages. For intent classification, the Bitext retail banking dataset (approximately 25,000 rows spanning 9 categories) was filtered to 8 HDFC-relevant banking categories. To handle out-of-domain questions, we sampled from Google's BoolQ dataset, which consists of 15,942 natural yes/no question pairs.

3.2.2 Synthetic Dataset Creation.

To ensure the system generalises well to unseen or adversarially phrased queries, synthetic data was generated using Llama-405B and Gemini-2.5-Flash via few-shot prompting. All generated data was validated by human evaluators for quality before use. Three task-specific datasets were curated:

- **Intent Classification Dataset (1,500 rows, 8 categories):** The Bitext corpus was filtered to eight HDFC-relevant categories (accounts, loans, investments, cards, operations, security, general, other). Llama-405B was prompted with 3 category-specific examples per call to generate queries across varied conversational tones, formal, casual, and abbreviated, simulating real customer phrasing diversity. The *other* category was augmented with synthetic questions about politics, sports, and celebrities to extend the out-of-domain boundary beyond BoolQ's yes/no format. Representative samples from each category are provided in Appendix A.2.1.
- **Safety Classification Dataset (1,600 rows, 4 categories):** The four safety classes, PII disclosure, Bad Advice, Competitor Information, and Safe, were each constructed with exactly 400 synthetically generated rows using few-shot prompting with 3 class-specific examples per call, enforcing strict class balance. PII examples contained realistic account numbers, passwords, and national identification references; Bad Advice examples included legally or financially incorrect guidance; Competitor Information examples mentioned rival banks by name; Safe examples were factual, compliant, and complete responses. Sample rows for each class are provided in Appendix A.2.2.
- **General Q&A Dataset (1,500 rows):** Llama-405B generated QA pairs in a three-part instruction-tuning format: a system instruction providing HDFC context from the official "About" section, a user query, and an assistant response that redirects the user to relevant HDFC banking services. All sequences were padded to 512 tokens. A sample instruction-tuning row is provided in Appendix A.2.3.

A multi-turn evaluation dataset of 100 conversations (2–4 turns each) was additionally generated using Gemini-2.5-Flash to produce ground-truth follow-up questions with pronoun-dependent references, simulating real conversational dynamics. A sample multi-turn conversation with ground-truth references is provided in Appendix A.2.4.

3.2.3 Manual Validation.

To guarantee data quality, a random sample of 380 records was drawn from each generated dataset and manually reviewed against three criteria: (1) relevance to the HDFC banking domain; (2) consistency in labelling and formatting; and (3) accurate depiction of HDFC-specific information. Fewer than 2% of reviewed rows were found to be malformed or mislabelled and were removed before training.

3.2.4 Dataset Analysis and Pre-processing.

Class distribution analysis on the HDFC FAQ corpus revealed that *insurance* and *cards* were over-represented while *fundtransfer* had minimal entries. We applied TF-IDF vectorisation to aggregate all questions per intent class into a single class document, then computed pairwise cosine similarity:

$$\text{sim}(C_i, C_j) = \frac{\mathbf{v}_i \cdot \mathbf{v}_j}{\|\mathbf{v}_i\| \, \|\mathbf{v}_j\|}$$

This revealed high semantic overlap between *fundtransfer* and *accounts*; the two classes were merged to prevent downstream routing ambiguity. The final pre-processed dataset splits are summarised in Table 9 (Appendix B).

The RAG knowledge base (approximately 2,500 passages) was chunked into 512-token segments with 50-token overlap and indexed into Chroma vector collections by intent category, enabling metadata-filtered retrieval.

3.3 Component Training

3.3.1 Intent Classifier.

TinyBERT (14.5M), DistilBERT (66M), and all-MiniLM were fine-tuned with 5-fold stratified cross-validation (LR = 2e-5, batch = 16, 3 epochs). Training loss decreased from 0.7 to approximately

0.01 by the second epoch. TinyBERT was selected based on its accuracy–latency trade-off: 93.95% test accuracy at approximately 28 ms per query.

3.3.2 Safety Guardrail.

The same three encoder architectures were fine-tuned on the 1,600-row safety dataset alongside Llama Guard 1B (base and fine-tuned) as external baselines. Training used 5-fold cross-validation, LR = 2e-5, batch = 8, 2 epochs, maximum 128 tokens. MiniLM was selected as the production guardrail.

3.3.3 General Q&A SLM.

Eleven SLMs (125M–7B) were fine-tuned using LoRA (rank 16) + 4-bit quantisation with LR = 2e-4 for 3 epochs, enabling models up to 7B parameters to train on a 16 GB GPU. An epoch ablation (1, 3, 5 epochs) on OPT-1.3B identified 3 epochs as optimal (Section 5.6).

3.4 Retrieval Pipeline

The retrieval pipeline supports three configurations: **Dense-only** (all-MiniLM-v2 FAISS embeddings), **Hybrid** (dense + BM25 EnsembleRetriever), and **Hybrid + Rerank** (hybrid retrieval followed by a cross-encoder reranker, top-$k = 5$). Metadata filtering routes each query to its intent-matched Chroma collection, restricting retrieval to one of six banking categories without sacrificing recall.

3.5 Conversation History Management

Five context management strategies were evaluated over 100 multi-turn conversations (2–4 turns each): **No History** (lower bound), **Full Buffer** (entire conversation), **Summarise** (first 8 words of each prior turn), **Window N=2** (last 2 turns only), and **Window+Summarise** (older turns compressed, recent 2 turns in full). To illustrate the practical difference between strategies, consider the 3-turn conversation in Appendix A.2.4 (RBI Repo Rate → home loan → fixed rate):

- **No History:** Receives only "What if I switch to a fixed rate instead?"; cannot resolve what "fixed rate" refers to. Produces a generic response.
- **Full Buffer:** Receives all three turns; correctly resolves "fixed rate" in the context of the Repo Rate discussion. Produces the most accurate response but consumes the most tokens.
- **Summarise:** Compresses prior turns to "What is the current RBI repo" and "How does that rate affect"; understands the topic but loses the loan-type nuance.
- **Window N=2:** Retains turns 2 and 3 in full; maintains the home loan context but discards the initial repo rate question.
- **Window+Summarise:** Compresses turn 1 to 8 words and keeps turns 2–3 in full; strongest balance between context fidelity and token efficiency.

4 Evaluation Methodology

To ensure the robustness of the SLM-powered multi-agent system, we employed a multi-dimensional evaluation framework combining quantitative classification metrics, retrieval ranking scores, and qualitative LLM-as-a-judge assessments.

4.1 Classification and RAG Evaluation

The intent and safety classification models were evaluated using standard multi-class metrics: Accuracy, Macro-F1, and Average and P95 Latency. For the RAG retrieval pipeline, retrieved chunk quality was assessed using Mean Reciprocal Rank (MRR@k), Precision@k, Recall@k, and Normalised Discounted Cumulative Gain (nDCG@k) at $k \in \{5, 10\}$. An ablation study was conducted across Dense-only, Hybrid, and Hybrid+Rerank configurations to isolate the specific contribution of BM25 ensemble retrieval and cross-encoder reranking independently.

4.2 Multi-Agent and Multi-Turn Validation

Evaluating generated response quality in a regulated domain requires moving beyond simple lexical metrics (BLEU, ROUGE), which fail to capture factual nuance. We implemented an LLM-as-a-judge framework using two independent judge models, GPT-4o-mini and Gemini-3-Flash, to score responses on a 1–5 scale across three dimensions:

- **Factual Accuracy:** Whether the response is fully grounded in the retrieved context without hallucinations or extrapolations.
- **Completeness:** Whether all aspects of the user's query were comprehensively addressed.
- **Clarity and Naturalness:** Whether the response is grammatically correct and adopts a professional banking tone.

An illustrative scoring example contrasting a factually grounded response against a fluent but ungrounded one is provided in Appendix A.3.1.

Table 2: Intent Classification: 5-fold CV and unseen test accuracy with inference latency. TinyBERT achieves the best latency–accuracy trade-off at approximately 28 ms per query.

Model	CV Acc.	Test Acc.	Macro-F1	Lat. (s)
DistilBERT	0.9982	0.9436	0.9437	41.7
MiniLM	0.9981	0.9513	0.9516	18.1
TinyBERT	0.9935	0.9395	0.9393	**8.4**
DistilGPT2	0.9251	—	—	—

Table 3: Response Safety Guardrail Model Comparison. MiniLM selected for production (best latency at near-peak accuracy). P95 latency for Llama Guard 1B base reaches 87,028 ms, making it operationally unviable.

Model	Acc. (%)	Macro-F1	Avg Lat. (ms)
DistilBERT	**99.062**	**99.061**	584.69
MiniLM	98.645	98.641	**250.86**
TinyBERT	94.270	94.210	1,194.75
LlamaGuard (Base)	60.420	55.220	33,679.40
LlamaGuard (FT)	52.500	48.860	52,616.90

For multi-turn evaluation, 100 conversations (2–4 turns each) were generated with Gemini-2.5-Flash producing ground-truth responses. The five context strategies were compared using Semantic Similarity (SentenceBERT cosine similarity), BLEU, and ROUGE-L against ground truth.

5 Results and Discussion

5.1 Intent Classification and Validation

Task definition and Setup. Intent classification maps user queries into one of eight banking categories (`accounts`, `loans`, `cards`, `investments`, `operations`, `security`, `general`, or `other`) for downstream routing (examples in Appendix A.3.2). Three lightweight encoders and one decoder model (DistilGPT2) were evaluated using 5-fold stratified cross-validation ($1,700$ samples/class) and an unseen test set of over $14,000$ stylized synthetic queries.

Table 2 reports cross-validation and test metrics alongside inference latency. While encoders achieved near-perfect cross-validation scores (0.9935–0.9982), test accuracy dropped by 4–5 percentage points across all models, illustrating the generalisation cost of limited training stylistic diversity.

TinyBERT (14.5M parameters) achieved 93.95% test accuracy at 8.4 s over 300 queries (≈28 ms/query), matching DistilBERT's accuracy while being $5\times$ faster. Conversely, DistilGPT2 underperformed (92.51%), confirming that decoder-only architectures are poorly suited for bidirectional discriminative tasks. Error analysis revealed that minor misclassifications were restricted to semantic overlaps between *accounts* and *operations* (4 of 602 predictions).

Selected model: $\text{TinyBERT}_{\text{General-4L-312D}}$ was selected for deployment due to its negligible 28 ms per-query routing latency.

5.2 Response Safety Classification

Task definition and Setup. The safety guardrail screens candidate model responses against four labels: `PII Disclosure`, `Bad Advice`, `Competitor Information`, or `Safe` (examples in Appendix A.3.3). Non-compliant outputs are replaced with standard deflections. Three fine-tuned encoders were benchmarked against Llama Guard 1B variants on a balanced 1,600-row safety dataset using 5-fold cross-validation.

As shown in Table 3, MiniLM delivered 98.645% accuracy at a rapid 250 ms average latency, whereas Llama Guard 1B base stalled at 60.42% accuracy and 33,679 ms latency. Fine-tuning Llama Guard 1B caused catastrophic forgetting, degrading accuracy to 52.50% due to a structural misalignment between its generative pre-training objectives and discriminative 4-class classification. Furthermore, Llama Guard's P95 latency spiked to 87,028 ms, making it unviable. **Deploying general-purpose safety APIs without domain-specific validation poses a major financial compliance risk.**

5.3 RAG Retrieval Ablation

Analysis. The Retrieval Agent pulls relevant FAQ passages from the knowledge base, evaluated over 100 queries at $k \in \{5, 10\}$ (worked example in Appendix A.3.4). As detailed in Table 4, dense-only and hybrid configurations yielded identical metrics at $k = 5$ (Recall 0.95, MRR 0.720, nDCG 0.763). Keyword-based BM25 provided zero benefits due to the standardized vocabulary of banking FAQs, and even degraded recall at $k = 10$ (0.95 to 0.89) by displacing valid dense results.

The cross-encoder reranker proved to be the sole driver of ranking optimization, delivering a **+17.2% MRR@5** and **+13.8% nDCG@5** improvement with zero recall penalty. By shifting the average rank of the first relevant document from 1.39 to

Table 4: RAG Retrieval Ablation at $k = 5$ and $k = 10$ (100 queries). The cross-encoder reranker is the sole driver of ranking improvement; BM25 contributes no benefit for structured FAQ retrieval.

Strategy	$k = 5$				$k = 10$			
	Rec.	MRR	Prec.	nDCG	Rec.	MRR	Prec.	nDCG
Dense-only	0.95	0.720	0.412	0.763	0.95	0.720	0.206	0.763
Hybrid	0.95	0.720	0.412	0.763	0.89	0.695	0.192	0.711
Hyb.+Rerank	**0.95**	**0.844**	**0.412**	**0.868**	**0.89**	**0.812**	**0.192**	**0.830**

Table 5: Single-turn generation cross-model comparison (11 SLMs, 3 epochs, 60 queries). LLM judge scores diverge substantially from lexical metrics: Llama-3.2-1B achieves the highest ROUGE-L but the lowest judge score, while OPT-1.3B leads in human-aligned quality.

Model	BERTScore F1	ROUGE-L	Judge (/10)	High-Q (%)
OPT-1.3B	0.6764	0.3056	**7.00**	**67.2**
Llama-3.2-1B (FT)	**0.6995**	**0.5115**	2.92	8.7
Qwen-2.5-1.5B (FT)	0.5945	0.2099	4.27	21.6
Qwen-2.5-3B (FT)	0.6281	0.2500	4.87	16.4
Qwen-3-4B	0.6215	0.2320	4.00	9.8
Gemma-3-1B (FT)	0.4959	0.3099	6.10	48.3
Gemma-3-4B	0.5959	0.2232	3.70	16.3
Gemma-2-7B (FT)	0.5949	0.2097	6.31	51.1
SmolLM-360M	0.6371	0.4619	5.75	31.7
SmolLM-125M	0.6339	0.4005	5.05	20.0
Mixtral-7B	0.5728	0.1926	4.05	20.0

1.18, the reranker guarantees that the primary context chunk contains the target answer, which is crucial for parameter-constrained SLMs bound to narrow context windows.

5.4 Cross-Model Generation Comparison

Analysis and Selection. The Generation Agent produces responses conditioned on the top-3 retrieved passages, evaluated across 11 fine-tuned SLMs (3 epochs, LoRA rank 16, 4-bit quantization, 60 evaluation queries) using an LLM judge (Appendix A.3.5). As shown in Table 5, OPT-1.3B led the group with a 7.00/10 judge score and a 67.2% high-quality response rate, despite registering a lower ROUGE-L (0.3056) than Llama-3.2-1B (0.5115). This inversion underscores that lexical overlap fails to track factual alignment. Crucially, scaling parameters beyond 1–2B offered no benefits, as seen by Gemma-3-4B underperforming Gemma-3-1B (3.70 vs. 6.10).

Model selection: OPT-1.3B was designated for standalone general Q&A. However, Gemma-3-1B-IT was chosen as the multi-agent backbone; OPT-1.3B's multi-call inference latency overhead (up to 239.90 s per 100 queries) makes it operationally prohibitive at scale compared to Gemma-3-1B-IT's 98.08 s runtime.

Table 6: Publish-Subscribe architecture: GPT-4o-mini judge evaluation (100 queries, /5 per dimension). Gemma-3-1B-IT achieves the highest factuality–completeness balance; Gemini-2.5-Flash-Lite achieves higher clarity but at the cost of factual grounding.

Model	Context	Fact.	Comp.	Clarity	Overall
Gemma-3-1B-IT	Raw	**2.89**	**2.79**	3.30	**2.99**
	Formatted	2.76	2.53	3.39	2.89
Gemini-2.5-Flash-Lite	Raw	2.10	1.75	**4.84**	2.90
	Formatted	1.71	1.43	4.66	2.60
OPT-1.3B	Raw	2.74	2.46	2.93	2.71
	Formatted	2.75	2.31	3.43	2.83

Table 7: Average response time (seconds per 100 single-turn queries) across all three architectures and models. Single Chain is fastest but lacks safety and factuality agents; Publish-Subscribe is recommended for regulated deployment.

Model	Context	Single Chain	Supervisor	Pub-Sub
Gemma-3-1B-IT	Raw	17.92	158.95	128.60
	Formatted	18.65	123.45	**98.08**
Gemini-2.5-Flash-Lite	Raw	41.89	41.77	33.24
	Formatted	29.68	30.21	33.87
OPT-1.3B	Raw	136.06	215.60	236.82
	Formatted	83.94	164.89	239.90

5.5 Multi-Agent Architecture Evaluation

Analysis. System architectures were evaluated across 100 single-turn queries using GPT-4o-mini (Table 6) and Gemini-3-Flash (Table 10, Appendix B) judges. Gemma-3-1B-IT achieved the highest overall score (2.99/5), balancing factuality and completeness. Conversely, cloud-hosted Gemini-2.5-Flash-Lite excelled in clarity (4.84/5) but scored poorly in factual grounding (2.10/5), proving that fluid outputs can compromise regulatory compliance.

As measured in Table 7, the Publish-Subscribe pattern dropped Gemma-3-1B-IT latency by **20.5%** over the Supervisor-only pattern (123.45 s to 98.08 s) and compressed average context lengths by 7.4%, translating to 69.4 hours of cumulative processing savings daily at a 10,000-query scale. Context structural formatting shaved off an extra 22.3% latency by scrubbing out redundant preamble tokens.

5.5.1 Multi-Turn Evaluation.

Table 8 benchmarks five history tracking strategies over 100 multi-turn sessions. For Gemma-3-1B-IT, **Window+Summarise** maximized semantic similarity (0.490) and BLEU (0.054), while **Full Buffer** registered peak holistic judge quality (1.64/5).

The `WINDOW_N2` strategy was the slowest configuration (565.99 s) due to iterative tokenization overhead induced by text slicing. The wide multi-turn

Table 8: Multi-turn context strategy evaluation (100 conversations, 2–4 turns each). Window+Summarise achieves the best semantic fidelity for SLMs; Full Buffer leads on holistic judge quality. The SLM–LLM gap is widest in multi-turn settings (1.64 vs. 3.97/5).

Model	Metric	No Hist.	Full Buf.	Summ.	Win. N2	Win.+Summ.
Gemma-3 1B-IT	Sem. Sim.	0.442	0.483	0.436	0.464	**0.490**
	BLEU	0.025	0.049	0.031	0.035	**0.054**
	ROUGE-L	0.127	**0.156**	0.136	0.135	0.150
	Judge (/5)	1.32	**1.64**	1.40	1.40	1.52
Gemini-2.5 Flash	Sem. Sim.	0.601	**0.844**	0.575	0.790	0.822
	BLEU	0.136	**0.485**	0.129	0.441	0.444
	ROUGE-L	0.319	**0.669**	0.310	0.618	0.628
	Judge (/5)	2.52	**3.97**	2.40	3.59	3.70

performance gap between the SLM and Gemini-2.5-Flash (1.64 vs. 3.97) establishes contextual coherence as a key frontier for SLM architectures. The deployment recommendation is to use Full Buffer up to 5 turns, shifting to Window+Summarise for longer interactions.

5.6 Training Duration Ablation

Fine-tuning duration tests on OPT-1.3B (1, 3, and 5 epochs, details in Table 11, Appendix B) identified a non-monotonic cliff-edge performance drop. Extending training from 3 to 5 epochs caused a **59.6% collapse** in judge scores (7.00 to 2.83). This severe decay indicates total dataset memorization within 4–5 epochs, corrupting foundational language patterns and resulting in repetitive token loops (example in Appendix A.3.6). This establishes a hard 3-epoch training ceiling for low-data SLM regimes and shows that evaluation requires qualitative human-aligned tracking rather than basic training loss tracking.

5.7 Lexical vs. LLM-as-Judge Divergence

The generation evaluations reveal a clear discrepancy between surface text metrics and automated qualitative reviews. Llama-3.2-1B secured the highest ROUGE-L (0.5115) but the lowest judge rating (2.92/10), because it routinely mirrored the language patterns of the prompt reference while omitting actual answer content. Conversely, OPT-1.3B synthesized accurate, contextually rephrased resolutions, netting a high judge rating (7.00/10) despite low surface string scores (0.3056). Consistent rankings across independent GPT-4o-mini and Gemini-3-Flash setups prove that multi-model LLM-as-judge frameworks offer the most accurate, secure, and discriminating verification signifiers for customer interactions in compliance-driven domains.

6 Conclusions

This paper demonstrated that SLM-powered multi-agent architectures can match or exceed cloud LLMs for single-turn banking customer service while operating entirely on-premise, at zero API cost, and in compliance with PCI-DSS and GDPR data residency requirements. We now revisit our four research questions with specific empirical answers.

RQ1 (SLM vs. LLM baselines): On-premise SLMs are competitive with cloud LLMs at a fraction of the cost. Gemma-3-1B-IT in the Publish-Subscribe architecture achieves a GPT-4o-mini judge score of 2.99/5, matching Gemini-2.5-Flash-Lite (2.90/5), while running at zero API cost and full data residency compliance. Moreover, domain-specific encoder models (MiniLM, TinyBERT) outperform general-purpose alternatives (Llama Guard 1B) by 38 percentage points in safety classification accuracy while reducing inference latency by 134–210×.

RQ2 (RAG strategies for SLMs): Cross-encoder reranking is the critical component for SLM factuality. Hybrid retrieval (dense + BM25) provides no improvement over dense-only retrieval for structured FAQ corpora (+0% MRR@5), whereas adding the cross-encoder reranker delivers +17.2% MRR@5 and +13.8% nDCG@5 with no recall penalty. For SLMs constrained to 512-token context windows, ranking quality directly translates into factuality gains.

RQ3 (Agent orchestration): The Publish-Subscribe pattern is the optimal orchestration architecture for task allocation and scalability. It reduces per-query latency by 20.5% over the Supervisor-only pattern (123.45 s → 98.08 s per 100 queries) and cuts average context token usage by 7.4%, yielding compounding savings at production scale.

RQ4 (Evaluation metrics): LLM-as-judge with multiple independent models is the only reliable quality signal for banking conversational AI. Lexical metrics (BLEU, ROUGE-L) actively invert model rankings: Llama-3.2-1B scores highest in ROUGE-L (0.5115) but lowest in judge quality (2.92/10), while OPT-1.3B leads in judge score (7.00/10) with a much lower ROUGE-L (0.3056). Rankings are consistent across two independent judges (GPT-4o-mini and Gemini-3-Flash), providing convergent validity. BLEU and ROUGE-L should not be used as primary evaluation criteria for domain-specific QA in regulated settings.

Limitations

Several constraints bound the generalisability of these findings. First, the system was evaluated exclusively on HDFC Bank of India's FAQ corpus; applicability to other banking institutions, multilingual environments, or regulatory regimes requires further validation. Second, the multi-turn ground truth (100 conversations) was synthetically generated by Gemini-2.5-Flash, introducing potential quality bias in favour of outputs stylistically resembling that model. Third, adversarial robustness, prompt injection and jailbreak attacks, was not tested, meaning the safety guardrail's real-world failure modes remain partially characterised. Fourth, no live human evaluators participated in the final output assessment; all quality signals were automated. Fifth, token-based cost analysis relies on projected rather than live deployment data, and the system's conversational memory is session-scoped, limiting cross-session continuity.

Three directions emerge directly from the present limitations. **Knowledge Graph Integration**: incorporating a banking-domain knowledge graph via LightRAG (Guo et al., 2025) could improve multi-entity relational reasoning beyond document-level retrieval. **Persistent Cross-Session Memory**: replacing session-scoped buffers with vectorstore-backed long-term memory (Wu et al., 2025) would enable customer history retention under GDPR data-minimisation constraints. **Multilingual Expansion**: fine-tuning multilingual SLMs (mBERT, BLOOM variants) on translated FAQ data for Hindi, Tamil, and Bengali would substantially extend HDFC's customer coverage. Additionally, adversarial stress-testing of the MiniLM guardrail under injection and jailbreak conditions is needed before production deployment.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, and 110 others. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *Preprint*, arxiv:2404.14219 [cs].

Peter Belcak, Greg Heinrich, Shizhe Diao, Yonggan Fu, Xin Dong, Saurav Muralidharan, Yingyan Celine Lin, and Pavlo Molchanov. 2025. Small language models are the future of agentic AI. *Preprint*, arxiv:2506.02153 [cs].

Jian Chen, Peilin Zhou, Yining Hua, Yingxin Loh, Kehui Chen, Ziyuan Li, Bing Zhu, and Junwei Liang. 2024. Fintextqa: A dataset for long-form financial question answering. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 6025–6047. Association for Computational Linguistics.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey. *Preprint*, arxiv:2312.10997 [cs].

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. 2025. LightRAG: Simple and fast retrieval-augmented generation. *Preprint*, arxiv:2410.05779 [cs].

Sirui Hong, Mingchen Zhuge, Jiaqi Chen, Xiawu Zheng, Yuheng Cheng, Ceyao Zhang, Jinlin Wang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. 2024. MetaGPT: Meta programming for a multi-agent collaborative framework. *Preprint*, arxiv:2308.00352 [cs].

Shijia Jiang, Yongfu Dai, Haochen Jia, Yuxin Wang, and Hao Wang. 2025. IntelliChain stars at the regulations challenge task: A large language model for financial regulation. In *Proceedings of the Joint Workshop of the 9th Financial Technology and Natural Language Processing (FinNLP), the 6th Financial Narrative Processing (FNP), and the 1st Workshop on Large Language Models for Finance and Legal (LLMFinLegal)*, pages 371–384. Association for Computational Linguistics.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023. CAMEL: Communicative agents for "mind" exploration of large language model society. *Preprint*, arxiv:2303.17760 [cs].

Yinheng Li, Shaofei Wang, Han Ding, and Hang Chen. 2024. Large language models in finance: A survey. *Preprint*, arxiv:2311.10723 [q-fin].

Zhenyan Lu, Xiang Li, Dongqi Cai, Rongjie Yi, Fangming Liu, Xiwen Zhang, Nicholas D. Lane, and Mengwei Xu. 2025. Small language models: Survey, measurements, and insights. *Preprint*, arxiv:2409.15790 [cs].

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*.

Pascal Pfeiffer, Philipp Singer, Yauhen Babakhin, Gabor Fodor, Nischay Dhankhar, and Sri Satish Ambati. 2024. H2o-danube3 technical report. *Preprint*, arxiv:2407.09276 [cs].

Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, Juyuan Xu, Dahai Li, Zhiyuan Liu, and Maosong Sun. 2024. ChatDev: Communicative agents for software development. *Preprint*, arxiv:2307.07924 [cs].

Qinglun Wang, Ji Yuan, Xinlong Huang, Jia Yan, Rixin Xiao, and Xianfeng Ding. 2024. Hybrid retrieval systems based on LLMs embedding and enhancement.

Yaxiong Wu, Sheng Liang, Chen Zhang, Yichao Wang, Yongyue Zhang, Huifeng Guo, Ruiming Tang, and Yong Liu. 2025. From human memory to ai memory: A survey on memory mechanisms in the era of llms. *Preprint*, arXiv:2504.15965.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024. TinyLlama: An open-source small language model. *Preprint*, arxiv:2401.02385 [cs].

Danna Zheng, Mirella Lapata, and Jeff Z. Pan. 2024. How reliable are LLMs as knowledge bases? rethinking facutality and consistency. *Preprint*, arxiv:2407.13578 [cs].

A Supplementary Examples

A.1 End-to-End Pipeline Trace

The following example traces a representative user query through all five agents of the Publish-Subscribe pipeline, illustrating the message flow from intent classification through safety screening.

End-to-End Pipeline Trace

User: "What documents do I need for a home loan?"

Step 1 — Intent Classifier: *Relevant* → `loans` category.

Step 2 — Retrieval Agent: Retrieves top-3 chunks from the `loans` Chroma collection (e.g., HDFC Home Loan eligibility FAQ). Cross-encoder reranker promotes the "documents required" chunk from rank 3 to rank 1.

Step 3 — Generation Agent: Produces: *"To apply for an HDFC home loan, you will need: identity proof (Aadhaar/Passport), address proof, last 3 months' salary slips, and property documents."*

Step 4 — Factuality Agent: Scores 4/5; `reason`: "Response is grounded in retrieved context; minor paraphrasing only."

Step 5 — Security Agent: Class → *Safe*. Response delivered to user.

A.2 Dataset Construction Examples

A.2.1 Intent Classification Samples

The following rows illustrate the tonal diversity introduced during synthetic data generation (Section 3.2.2), covering formal, casual, abbreviated, and out-of-domain utterances.

Sample intent classification rows:

Formal: "Could you please advise on the applicable processing fee for a personal loan application?" → `loans`

Casual: "hey how do i check my balance?" → `accounts`

Abbreviated: "min bal savings acc?" → `accounts`

Out-of-domain: "Who won the last FIFA World Cup?" → `other`

A.2.2 Safety Classification Samples

The following rows provide one representative example per safety class, illustrating the boundaries between harmful and compliant model outputs.

Sample safety classification rows (one per class):

PII: "Your account number is 5021XXXX and your PIN is usually your birthdate." → `PII Disclosure`

Bad Advice: "You can withdraw from a fixed deposit any time without penalty." → `Bad Advice`

Competitor: "SBI and ICICI offer lower home loan rates than HDFC at this time." → `Competitor Information`

Safe: "HDFC FD rates currently range from 3% to 7.25% depending on tenure." → `Safe`

A.2.3 General Q&A Instruction-Tuning Sample

The following row shows the three-part instruction-tuning format used to fine-tune the General Q&A SLM component.

Sample general Q&A instruction-tuning row:

System: "You are a helpful assistant for HDFC Bank India. Respond to general queries and guide the user to relevant services."

User: "Good morning! How are you today?"

Assistant: "Good morning! I'm here and ready to assist you with your HDFC banking needs. How can I help you today?"

A.2.4 Multi-Turn Ground-Truth Conversation

The following 3-turn conversation illustrates the pronoun-dependent reference structure used in multi-turn evaluation (Section 3). It is also the example used in the context strategy comparison in Section 3.

Sample multi-turn conversation (ground-truth):

Turn 1 — User: "What is the current RBI repo rate?"
Turn 1 — Reference: "The current RBI repo rate is 6.50%."

Turn 2 — User: "How does *that rate* affect my home loan EMI?"
Turn 2 — Reference: "Since the repo rate is 6.50%, floating-rate home loan interest is typically set 2–3% above it, meaning your EMI will adjust accordingly."

Turn 3 — User: "What if I switch to a fixed rate instead?"
Turn 3 — Reference: "Fixed-rate loans are unaffected by repo rate changes but usually carry a slightly higher base rate at the time of signing."

A.3 Evaluation Examples

A.3.1 LLM-as-Judge Scoring

The following example contrasts a factually grounded response (high factuality score) with a fluent but ungrounded response (high clarity, low factuality), illustrating the trade-off central to the judge evaluation framework in Section 4.

LLM-as-Judge scoring example:

Query: "What is the minimum balance for an HDFC savings account?"

Retrieved context: "HDFC Bank requires a minimum average monthly balance of 10,000 for metro branches and 5,000 for semi-urban/rural branches."

Response A (Gemma-3-1B-IT): "The minimum balance is 10,000 for metro branches and 5,000 for semi-urban and rural branches." → **Factuality: 5/5**, Completeness: 4/5, Clarity: 4/5

Response B (Gemini-2.5-Flash-Lite): "HDFC Bank offers flexible savings accounts with competitive balance requirements tailored to your lifestyle and location." → **Factuality: 1/5**, Completeness: 1/5, Clarity: 5/5

A.3.2 Intent Classification Task Examples

The following examples illustrate the breadth of the 8-class intent routing task described in Section 5.1.

Intent classification examples:

Input: "How do I apply for a credit card?" → `cards`

Input: "When is the next EMI due for my car loan?" → `loans`

Input: "I forgot my net banking password" → `security`

Input: "What are your working hours?" → `operations`

Input: "Who is the CEO of Tesla?" → `other`

A.3.3 Safety Classification Task Examples

The following examples illustrate the 4-class safety screening task described in Section 5.2, including the fine-grained distinction between harmful and compliant banking responses.

Safety classification examples:

Input: "Your debit card PIN is usually the last 4 digits of your Aadhaar."
→ `PII Disclosure` (contains sensitive guidance on PIN formation)

Input: "You can break your HDFC fixed deposit before maturity without any charge."
→ `Bad Advice` (factually incorrect; premature FD closure incurs a penalty)

Input: "Axis Bank currently offers better credit card rewards than HDFC."
→ `Competitor Information` (direct competitor comparison)

Input: "HDFC Bank's personal loan interest rate starts from 10.50% per annum."
→ `Safe` (accurate, factual, and compliant)

A.3.4 RAG Cross-Encoder Reranking Example

The following example illustrates how the cross-encoder reranker reorders dense-retrieved passages, promoting the correct answer chunk from rank 3 to rank 1 and improving the query's MRR from 0.33 to 1.0.

Reranking example (query: "What is the NEFT

transfer limit per day?”):

Dense-only top-3 (before reranking):
Rank 1: “RTGS transfer: minimum 2 lakh, no upper limit...” (*wrong passage*)
Rank 2: “IMPS transfer: up to 5 lakh per transaction...”
Rank 3: “NEFT: no minimum; daily limit up to 25 lakh...” (*correct*)

After cross-encoder reranking:
Rank 1: “NEFT: no minimum; daily limit up to 25 lakh...” (*correct, promoted from rank 3*)

MRR for this query: Dense-only $= 1/3 = 0.33$; Reranked $= 1/1 = 1.0$.

A.3.5 Response Generation Quality Example

The following example contrasts two generation outputs on the same query, showing how repetition failure modes in fine-tuned SLMs can produce high ROUGE-L scores while being judged as low quality.

Generation quality example (query: “How do I activate international usage on my HDFC debit card?”):

Llama-3.2-1B (ROUGE-L: 0.51; Judge: 2.9/10):
“To activate international usage on your debit card, you can activate international usage on your card by activating international usage through NetBanking or by activating it at a branch.” — Repetitive, no actionable steps.

OPT-1.3B (ROUGE-L: 0.31; Judge: 7.0/10):
“Log in to HDFC NetBanking, go to Cards → Debit Card → International Usage, and enable the feature. Alternatively, call PhoneBanking at 1800-202-6161 or visit your nearest branch.”

A.3.6 Training Collapse at Epoch 5

The following example illustrates the degenerate output produced by OPT-1.3B after 5 training epochs on the 1,500-sample dataset, compared to the coherent 3-epoch response. This collapse motivates the 3-epoch training optimum identified in Section 5.6.

Degenerate output at epoch 5 (OPT-1.3B):

Query: “How do I apply for an HDFC credit card?”

Epoch 3 response (Judge: 7/10): *“You can apply for an HDFC credit card online at hdfcbank.com/cards, via the HDFC Bank app, or by visiting your nearest branch with your PAN card and income proof.”*

Epoch 5 response (Judge: 1/10): *“Apply apply apply credit card HDFC apply credit apply HDFC card HDFC apply apply apply...”*

B Supplementary Tables

B.1 Dataset Summary

Table 9 summarises the final pre-processed train/test splits for all three curated datasets, each manually validated on 380 randomly sampled records prior to training (Section 3.2).

Table 9: Pre-processed dataset summary. Each dataset was manually validated on 380 randomly sampled records prior to training.

Dataset	Total	Train	Test	Validated
Intent Classification	1,500	1,200	300	380
Safety Classification	1,600	1,280	320	380
General Q&A Pairs	1,500	1,350	150	380

B.2 Independent Judge Evaluation (Gemini-3-Flash)

Table 10 provides the independent Gemini-3-Flash judge scores for the Publish-Subscribe architecture, corresponding to the GPT-4o-mini evaluation in Table 6. Agreement across both judges confirms that the Gemma-3-1B-IT ranking is not an artefact of a single judge’s scoring behaviour.

Table 10: Publish-Subscribe architecture: independent Gemini-3-Flash judge evaluation (100 queries, /5 per dimension). Both judges agree: Gemma-3-1B-IT leads overall; Gemini-2.5-Flash-Lite leads in clarity but trails in factuality and completeness.

Model	Context	Fact.	Comp.	Clarity	Overall
Gemma-3-1B-IT	Raw	**2.29**	**1.84**	3.16	**2.43**
	Formatted	2.10	1.62	2.77	2.16
Gemini-2.5-Flash-Lite	Raw	1.98	1.68	**3.94**	2.53
	Formatted	1.76	1.37	3.41	2.18
OPT-1.3B	Raw	1.73	1.45	2.02	1.73
	Formatted	1.50	1.29	1.71	1.50

B.3 Training Duration Ablation

Table 11 presents the full epoch ablation results for OPT-1.3B referenced in Section 5.6. The LLM judge score collapses 59.6% from 3 to 5 epochs,

identifying 3 epochs as the hard training optimum for a 1,500-sample fine-tuning dataset.

Table 11: OPT-1.3B epoch ablation (LR = 2e-4, LoRA rank 16, 4-bit quantisation). The LLM judge score collapses 59.6% from 3 to 5 epochs, identifying 3 epochs as the hard training optimum for a 1,500-sample dataset.

Epochs	**BERT-F1**	**ROUGE-L**	**Judge (/10)**	**High-Q (%)**
1	0.651	0.281	5.80	44.1
3	0.676	0.306	**7.00**	**67.2**
5	0.589	0.217	2.83	12.5