



HONG KONG MONETARY AUTHORITY
香港金融管理局



Practical Insights

from the Second
Cohort of the
GenA.I. Sandbox

Contents

Executive Summary	3
1. Introduction	4
1.1 GenA.I. and A.I. Agents and Their Core Capabilities in the Banking Industry	4
1.2 Background of GenA.I. Sandbox	4
1.3 Overview of Use Cases	5
<i>1.3.1 Key Factors for Consideration in Evaluating Applications</i>	<i>5</i>
<i>1.3.2 Key Functional Building Blocks</i>	<i>6</i>
<i>1.3.3 Observations and Lessons Learnt</i>	<i>7</i>
2. Foundation: Data Strategy and Preparation	7
2.1 Data Collection	7
2.2 Data Pre-processing	11
2.3 Data Augmentation	14
2.4 Data Quality Check	17
2.5 Continuous Data Updates	20
3. Approaches to solution design	21
3.1 Model Selection	21
3.2 Optimisation Strategies	24
<i>3.2.1 Prompt Engineering</i>	<i>26</i>
<i>3.2.2 Retrieval-Augmented Generation (RAG)</i>	<i>28</i>
<i>3.2.3 Parameter-Efficient Fine-Tuning (PEFT)</i>	<i>28</i>
<i>3.2.4 Knowledge Distillation</i>	<i>28</i>
3.3 Validation and Evaluation	30
<i>3.3.1 Model Validation</i>	<i>30</i>
<i>3.3.2 Model Evaluation</i>	<i>31</i>

3.4 A.I. Agents	36
3.4.1 Advantages of Multi-agent Systems	37
3.4.2 Challenges & Limitations	38
3.4.3 Agent Intent	39
3.4.4 Governance on A.I. Agents	41
4. Addressing Key Challenges and Risks	43
4.1 Ongoing Risks of GenA.I.	43
4.1.1 Hallucination and Inaccuracy	43
4.1.2 Bias and Fairness	50
4.1.3 Security and Privacy	52
4.1.3.1 Security	52
4.1.3.2. Privacy	54
4.1.4 Explainability and Transparency	54
4.2 Emerging Risks in More Autonomous A.I. Systems	56
4.3 A.I. vs. A.I.: An Evolving Security Dynamic	58
4.4 Adaptive Guardrails	61
4.5 A Multi-Layered Defence Against Deepfake Threats	61
5. Business Outcome	63
6. Conclusion	65
7. Appendix	67

Executive Summary

The second cohort of the Generative Artificial Intelligence (GenA.I.) Sandbox marks a significant maturation of the banking industry's adoption. The Hong Kong Monetary Authority (HKMA), in collaboration with the Hong Kong Cyberport Management Company Limited (Cyberport) has launched the GenA.I. Sandbox, the first of its kind, since August 2024. Building on the experimental phase of the first cohort, this second cohort signifies a strategic shift from validating basic GenA.I. capabilities to more secure, reliable, and scalable deployment of multimodal solutions as well as initial exploration of A.I. agents. The focus has also moved beyond simple text generation to supporting more complex banking workflows, with stronger emphasis on integrating A.I. governance into banks' three lines of defence.

In the second cohort, 27 use cases have been selected involving 20 banks and 14 technology partners. Notably, seven of these partnerships were established directly through the GenA.I. Sandbox Collaboratory workshops since July 2025, demonstrating the initiative's success in fostering early engagement and practical problem-solving between banks and technology providers. These use cases target critical domains including risk management, anti-fraud measures, and customer experience.

The insights from both the first and second cohorts cover the full development lifecycle, beginning with dynamic data strategies (Section 2) that support multimodal inputs and continuous learning. The report also details transition from solution design (Section 3) to multi-model stacks and multi-agent architectures. Under this approach, smaller and efficient models may handle high-volume of routine tasks, while larger models can manage complex reasoning. Key optimisation techniques, including Parameter-Efficient Fine-Tuning (PEFT), Knowledge Distillation, and Retrieval-Augmented Generation (RAG), are employed to offer fit-for-purpose performance and cost-efficiency.

This cohort focuses on governance and risk mitigation (Section 4). As A.I. systems become more autonomous, traditional controls become increasingly insufficient. Participating banks in the second cohort have developed robust frameworks to address emerging risks, including hallucination, bias, and security threats. A key innovation explored is the "A.I. vs. A.I." paradigm, where secondary A.I. systems or heterogeneous models are used to validate, challenge, and monitor primary A.I. outputs. Combined with adaptive guardrails and intent-anchored agent boundaries, this approach allows a balance between autonomy and accountability, securing regulatory compliance, as well as human oversight.

In addition, the second cohort has demonstrated that A.I. agents can significantly enhance operational efficiency, strengthen risk controls, and elevate customer experience when implemented within a robust governance framework. The trials confirm that the value of agentic A.I. lies in combining fit-for-purpose models, intent-anchored governance, defined workflows, and evidence-based validation. Building on these outcomes, the industry is now poised for GenA.I. Sandbox++, which will further explore the responsible integration of agentic A.I. into core financial processes, enabling innovation to proceed with the overreaching principles of accountability, resilience and trust.

1. Introduction

1.1 GenA.I. and A.I. Agents and Their Core Capabilities in the Banking Industry

GenA.I. is a subset of artificial intelligence capable of generating text, images, or other forms of outputs in response to instructions. Within the banking industry, GenA.I. offers a range of transformative capabilities. These include automating and tailoring customer service through intelligent chatbots and virtual assistants, generating personalised products and services, augmenting risk assessment and fraud detection by leveraging vast amounts of structured and unstructured data. These go beyond earlier deployment of GenA.I. to streamline back-office operations by automating routine tasks such as document processing, information retrieval and content preparation.

A.I. agents are A.I.-enabled applications, which may incorporate GenA.I. models, designed to assist users in performing defined tasks through natural-language interaction, information retrieval, analysis, content generation, or the use of pre-approved tools and workflows. They can support customer-facing and internal banking processes by bringing together relevant information, tools and functions within a defined scope.

By responsibly harnessing these capabilities, banks can improve operational efficiency, elevate customer satisfaction, support risk control and compliance strategies, and maintain a competitive edge in the rapidly evolving financial technology (Fintech) landscape.

1.2 Background of GenA.I. Sandbox

The GenA.I. Sandbox, launched by the HKMA in collaboration with the Cyberport, aims to promote responsible innovation in A.I. including GenA.I. across the banking industry. It provides a risk-controlled environment for Authorized Institutions to develop and test their innovative A.I. solutions to address real-world banking use cases.

In the first cohort started in 2024, the participating banks explored a wide range of innovative applications. A total of 15 innovative use cases, from 10 banks and four technology partners ([Appendix](#)), have been selected from over 40 submissions following a rigorous prioritisation process¹. The pilot use cases mainly revolve around enhancing risk management, anti-fraud measures and customer experience. Following the completion of the trials in the first cohort of the GenA.I. Sandbox, an insight report was issued by the HKMA on 31 October 2025².

Building on the positive response to the inaugural cohort, the HKMA and Cyberport announced the launch of the second cohort of the GenA.I. Sandbox in April 2025³. A key addition to the second cohort is the introduction of the GenA.I. Sandbox Collaboratory (Collaboratory), a platform comprising a series of practical workshops that facilitates early engagements between banks and technology providers. These workshops are designed to drive the conversion of problem statements into practical use cases, which can then be trialled in the GenA.I. Sandbox. The Collaboratory series featured over

1 HKMA announced inaugural cohort of GenA.I. Sandbox, 19 Dec 2024, <https://www.hkma.gov.hk/eng/news-and-media/press-releases/2024/12/20241219-5/>

2 Practical Insights from the GenA.I. Sandbox, 31 October 2025. <https://brdr.hkma.gov.hk/eng/doc-ldg/docId/getPdf/20251031-6-EN/20251031-6-EN.pdf>

3 HKMA launched second cohort of GenA.I. Sandbox, 28 Apr 2025. <https://www.hkma.gov.hk/eng/news-and-media/press-releases/2025/04/20250428-5/>

80 one-on-one sessions between over 10 technology firms and 20 banks.

Following the Collaboratory workshops and the application process, the HKMA announced the selected use cases for the second cohort of the GenA.I. Sandbox in October 2025⁴. The second cohort comprised 27 use cases involving 20 banks and 14 technology partners (Appendix). Of these, seven partnerships were newly established through the Collaboratory arrangement.

The second cohort reflects a progression from experimentation with GenA.I. capabilities towards their secure, reliable and scalable implementation. It places emphasis on integrating A.I. governance into the three lines of defence of banks, exploring “A.I. vs. A.I.” approaches for proactive risk management, and developing adaptive guardrails and self-correcting mechanisms.

1.3 Overview of Use Cases

1.3.1 Key Factors for Consideration in Evaluating Applications

The prioritisation of GenA.I. Sandbox proposals was guided by several key factors (Figure 1):

Level of innovation: The extent to which the proposed solution introduces novel ideas, methodologies or models with potential to create new digitalisation opportunities;

Complexity of the solutions: The technical sophistication and intricacy of the proposed solution to promote advancement and added value;

Expected contribution to the industry: The potential for the proposed solution to make a meaningful impact on the financial services sector, such as by addressing industry-wide challenges through replicable and scalable application; and

Adherence to the principle of fair use: The degree to which the proposed solution is designed to use the pooled computing power in a fair, responsible, and efficient manner.



Figure 1 – Key Factors for Consideration in Evaluating Applications

4 HKMA announced second cohort of GenA.I. Sandbox, 15 Oct 2025, <https://www.hkma.gov.hk/eng/news-and-media/press-releases/2025/10/20251015-4/>

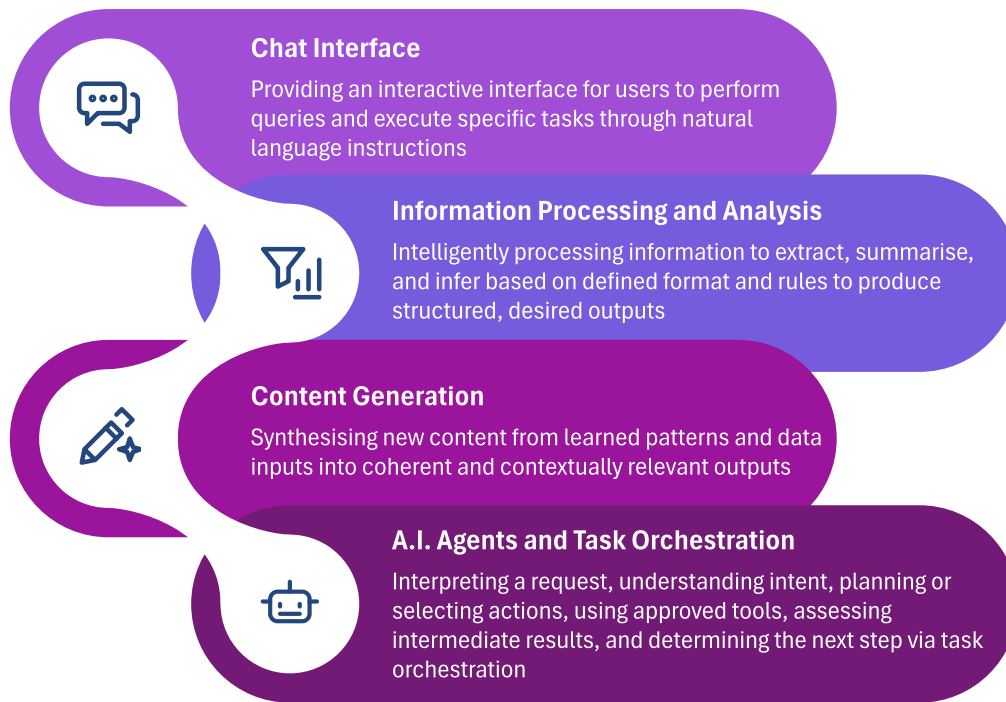


Figure 2 – Key Functional Building Blocks

1.3.2 Key Functional Building Blocks

Across various use cases, GenA.I. capabilities can be categorised into several core functional building blocks that serve as foundational components for broader applications (Figure 2):

Chat Interface: This provides an interactive interface for users to engage with Large Language Models (LLMs), enabling users to perform queries and execute specific tasks through natural language instructions.

Information Processing and Analysis: This leverages GenA.I. to automate and optimise the handling of information by intelligently processing information to extract, summarise, and infer based on defined format and rules to produce structured, desired outputs. Applications are not limited to text while participating banks also applied GenA.I. to scanned documents, images and audio recordings, including identity documents and recorded interactions, using multimodal models or supporting technologies such as optical

character recognition and speech-to-text.

Content Generation: This leverages the core capability of GenA.I. to synthesise new content—such as text, image, and multimedia—from learned patterns and data inputs. It synthesises underlying data, brand guidelines, and creative prompts into coherent and contextually relevant outputs. This basic ability to create, rather than merely retrieve, serves as a fundamental enabler for service customisation.

Building on the core capabilities above, a number of banks in the second cohort further explored an additional functional building block:

A.I. Agents and Task Orchestration: A.I. agents enable the pursuit of a defined objective through multiple steps. Within defined permissions and controls, A.I. agents can interpret a request, understand intent, plan or select actions, use approved tools and data sources, assess intermediate results, and determine the appropriate next step. Task orchestration coordinates how and when these steps are

executed, what information may be passed between them, and when the process needs to stop or escalate. It may follow a fixed workflow and does not require an A.I. model to determine the path. Together with guardrails, access controls, monitoring and human oversight, A.I. agents enable controlled execution of complex, multi-step processes.

1.3.3 Observations and Lessons Learnt

Based on lessons learnt and observations arising from the technical trials of the second cohort of the GenA.I. Sandbox, this report provides practical reference points for the development and implementation of future use cases in the banking industry. Sections 2 to 4 of this report set out key considerations and practices relating to **data strategy and preparation, solution design**, and **addressing key challenges**, respectively. Across these areas, the report also

incorporates considerations relevant to the exploration and use of **A.I. agents**, where applicable.

These practical insights may support the achievement of stronger A.I. governance and risk management, including as set out under the HKMA's updated Supervisory Guidance for Responsible Adoption of Artificial Intelligence (the Guidance), framed under **"ART" (Accountability, Resilience and Trust)**. Throughout the report, individual practices and observations will be marked to indicate the "A", "R" and/or "T" dimensions to which they are most closely related. A simple legend is provided below to explain the relevant markings. This report should also be read in conjunction with the Guidance. The report sets out non-exhaustive examples of how the Guidance may be applied in practice.

Legend

A Accountability

R Resilience

T Trust

2. Foundation: Data Strategy and Preparation

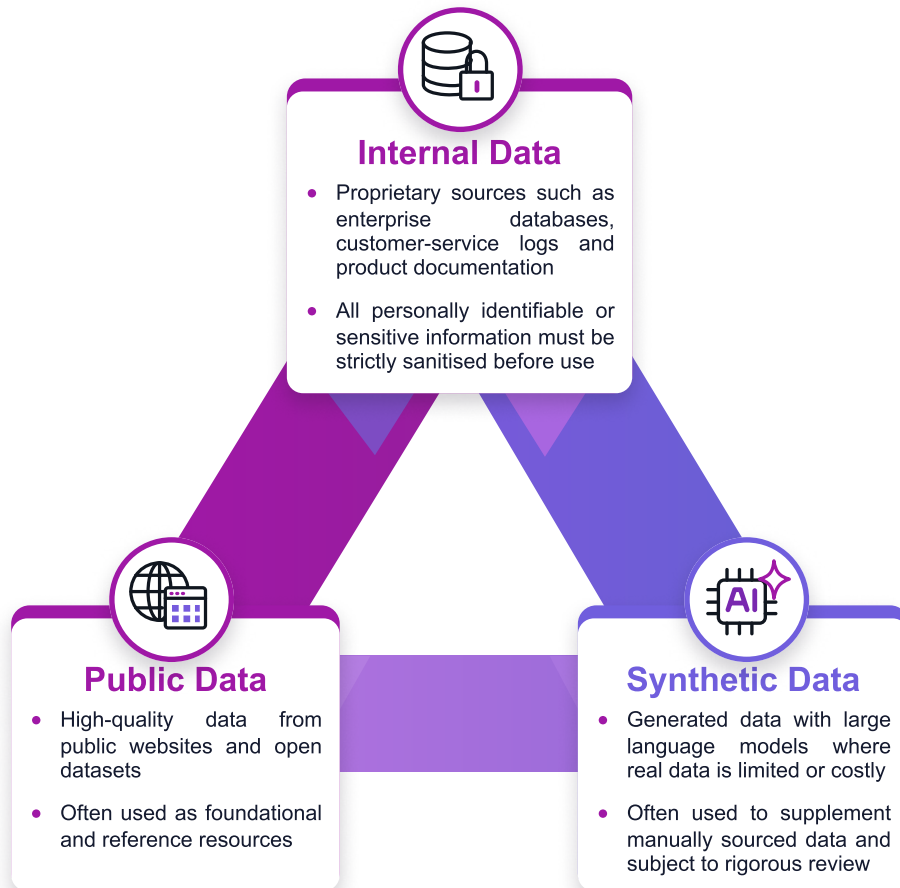
As highlighted in the first cohort, the performance, reliability, accuracy and contextual relevance of GenA.I. solutions depend fundamentally on the quality of the data used for training, fine-tuning and testing. Dataset planning and preparation are among the most critical and resource-intensive stages of development, requiring careful sourcing, cleaning, labelling and curation to ensure that the data are representative, unbiased and fit for purpose.

In the second cohort, it is observed that the adoption of GenA.I. applications has increasingly

shifted towards A.I. agents and multimodal capabilities. These applications generate content from both structured and unstructured data, including financial documents, images, audio and video. This evolution requires a transition from static datasets to dynamic, self-improving data ecosystems. Unlike previous models that relied on large volumes of static text, modern A.I. agents require continuously evolving and verifiable context to function effectively while processing multi-modal data inputs.

2.1 Data Collection

R T Data collection is critical to ensuring that datasets are inclusive, reliable and fit for purpose. Participating banks in the GenA.I.

Figure 3 – Channels of Data Sources **R T**

Sandbox drew on diverse and independent sources (Figure 3), including publicly available and proprietary datasets, and curated data across key dimensions such as product lines, languages and customer personas.

While some banks adopted a cautious, manual approach during initial training to reduce the risk of errors and irrelevant content, others combined manual collection with automated techniques. For example, a bank built an A.I. agent with a large language model to generate a robotic process automation (RPA) workflow that crawled accessible links on its public website and loaded the content into a datastore. This created a comprehensive, accurate and source-aligned searchable knowledge base to support future

development.

Participating banks generally partitioned curated datasets into training, validation and test sets following conventional ratios such as 70-20-10 or 80-10-10. The training set, which comprises the largest share, is used to train the model to identify patterns and relationships in the data. The validation set supports model selection and hyperparameter tuning during development, helping to identify and mitigate overfitting. The test set is held back from both training and validation and used only for final evaluation, providing an unbiased assessment of the model's accuracy, robustness and expected performance on unseen data.

2nd cohort

Case Study: Designing a Traceable and Balanced Dataset for an A.I.-Augmented RPA Solution

R T

Background

A bank developed a fine-tuning dataset to support an A.I.-augmented Robotic Process Automation (RPA) solution. The dataset was designed to help the model respond effectively to routine business requests while consistently identifying and rejecting non-compliant, risky or unauthorised requests. Key design objectives included data traceability, compliance alignment, balanced learning, realistic conversational capability and scalability across different model sizes and project stages.

Approach**1. Select a suitable dataset format**

The dataset was constructed in formats compatible with model fine-tuning frameworks. The dataset was stored in JSONL (JSON Lines) format, enabling efficient loading, parsing and scaling within the LLaMA Factory training framework. Each line comprised one compact and independent JSON object, which reduced unnecessary formatting and resource consumption during training. In this case, the bank selected an instruction format for natural language-driven RPA process generation.

Data Format	Applicable Scenarios	Example
Instruction format	Command-based tasks and user instructions execution	<pre>{"instruction": "Generate an RPA workflow to download a daily sales report from a web portal and save it to a shared folder.", "input": "Portal URL", "output": "1.Open the portal URL and log in using secure vault credentials. 2.Go to Reports > Sales > Daily Sales Report. 3.Set the report date to the current business date and select XLSX format. 4.Download the report and rename it Daily_Sales_Report_YYYYMMDD.xlsx. 5.Move the file to the destination folder and verify its existence. 6.Log completion with the file name and timestamp. 7.If an error occurs, capture a screenshot, log the details, and notify the Operations mailbox."}</pre>
Completion format	Single-turn conversation or classification tasks	<pre>{"prompt": "Classify the customer sentiment as positive, negative, or neutral: The new online banking interface is easy to use and much faster than before.", "completion": "positive"}</pre>
Conversational format	Multi-turn conversation, such as chatbots and customer service assistant	<pre>{"messages": [{"role": "system", "content": "You are a customer-service assistant for a bank. Do not provide financial advice."}, {"role": "user", "content": "How can I reset my online banking password?"}, {"role": "assistant", "content": "Use the 'Forgot Password' option on the online banking sign-in page and follow the identity-verification instructions. If you cannot complete the process, contact the bank's customer-service hotline."}]}</pre>

Table 1 - Examples of Dataset Format

2. Embed traceability and compliance metadata

Each data sample contained seven core fields:

- A unique dialogue identifier for audit tracking;
- The number of dialogue turns;
- The user query;
- The expected model response;
- The applicable scenario category;
- The risk type, including whether the case was compliant or required rejection; and
- The creation date and applicable compliance-rule version.

This structure enabled the project team to track each sample, assess coverage by scenario and risk type, and maintain the dataset as compliance requirements evolved.

3. Balance compliant and rejection scenarios

The dataset comprised 90% normal question-and-answer pairs and 10% rejection pairs. Normal scenarios covered core business needs, including natural-language-to-RPA process conversion, compliance-rule consultation and RPA-process optimisation. Rejection scenarios covered inappropriate or prohibited requests, such as illegal operations, non-compliant process designs and unauthorised data access. This balance helped the model perform useful business tasks without undertraining its ability to recognise and safely handle risky requests.

4. Separate data for training, evaluation and testing

The dataset was divided into an 85% training set, 5% evaluation set and 10% test set. The evaluation set contained representative normal and rejection samples across all scenarios and risk categories to support iterative tuning. The test set remained unseen throughout training and was reserved for final performance assessment, helping to reduce overfitting risk and provide a more reliable measure of model performance.

5. Design realistic conversation data

To reflect real-world interactions with the A.I.-augmented RPA solution, dialogue samples ranged from one to four turns. Simple, one-turn consultations accounted for 15% of cases; two-turn and three-turn interactions each accounted for 35%; and complex four-turn conversations represented the remaining 15%. This distribution enabled the model to handle both straightforward requests and more involved, multi-step problem-solving interactions.

6. Adopt a scalable deployment approach

The dataset was designed with an output-scale gradient to support training and testing across Qwen models of different parameter sizes, including 8B, 14B and 72B variants. Smaller datasets could be used for early debugging and testing, while larger datasets could support formal deployment. This approach allowed the team to align training resources with the relevant stage of development while preserving a consistent dataset structure and control framework.

Outcome

The dataset design provided a structured foundation for developing a compliant and operationally useful A.I.-augmented RPA solution. By combining standardised data formatting, comprehensive metadata, a controlled mix of compliant and rejection scenarios, independent testing and realistic multi-turn dialogues, the project team improved the model's trainability, traceability and ability to respond safely across a range of business interactions.

2.2 Data Pre-processing

R T Curated datasets generally require pre-processing before use. The participating banks applied pre-processing primarily to improve data quality, consistency and relevance to support efficient and accurate model training and inference and protect confidential and personal information throughout the model-development lifecycle.

Following data collection, banks cleansed, standardised and filtered data to improve data consistency, fine-tuning efficiency and the quality of model outputs. These processes also mitigated context overload and noise contamination, which occur when a model receives excessive, redundant or irrelevant information. Such information can consume the model's limited context window, increase latency and impair reasoning quality, particularly in agent-based A.I. systems. Common practices included:

Enhancing Data Quality for Model Performance

R T

- **Text normalisation:** Converting text into a consistent format, such as using lowercase text, removing unnecessary special characters, applying a standard date format (e.g. YYYY-MM-DD), and converting monetary values into a common currency (e.g. HKD).
- **Semantic standardisation:** Mapping equivalent terms to a controlled vocabulary. For example, "home loan", "mortgage loan" and "residential property financing" may be standardised as "residential mortgage" to ensure consistent representation of the same concept across datasets.
- **Algorithmic text cleaning:** Using regular expressions (regex) and parsing tools to remove HTML tags, markdown artefacts, tracking pixels, boilerplate text, headers, footers and cookie notices.
- **De-duplication:** Applying techniques such as MinHash or locality-sensitive hashing (LSH)

to identify and remove duplicate or near-duplicate documents and paragraphs. This reduces redundancy and prevents models from over-weighting repeated information.

- **Quality and information-density filtering:** Applying predefined heuristics or lightweight models, such as cross-encoders or smaller LLMs, to identify and exclude low-quality, incomplete, repetitive or low-value content.
- **Semantic chunking:** Dividing lengthy documents into smaller, coherent segments based on logical boundaries, such as sections, paragraphs, sentences or clauses, rather than arbitrary character or token limits. This preserves the relationship between relevant information while allowing content to fit within the model's context window.
- **Contextual enrichment:** Adding concise metadata to each chunk, such as the document title, section heading or source, to retain essential context after a document is divided.
- **Document summarisation:** Using LLM-based or abstractive summarisation methods to distil lengthy documents into their key points before adding them to a training dataset or vector database.

Privacy Protection

R T Banks also removed, obscured or replaced sensitive information before using data for model training. Common practices included:

- **Data masking:** Identifying and masking personally identifiable information (PII) while retaining sufficient data utility for model training and testing.

- **Tokenisation:** Replacing sensitive data elements with non-sensitive tokens that have no inherent or exploitable meaning. The original data and the token mapping are stored separately in a secure token vault.
- **Pseudonymisation:** Replacing direct identifiers with realistic but artificial alternatives, such as substituting customer names with randomly generated names. This preserves the structure and context of the data while reducing the ability to identify individuals.
- **Synthetic data generation:** Using GenA.I. to create artificial datasets that replicate the statistical characteristics and relationships of real data without containing actual customer information.

Document and Image Pre-processing for Multimodal LLMs

R T Banks also process significant volumes of unstructured data, including PDFs, scanned forms, images and other documents. Pre-processing is necessary to extract structured and relevant information, improve input quality and protect sensitive information, even where multimodal LLMs can process images directly. Common practices included:

- **Text extraction and layout analysis:** Using optical character recognition (OCR) to convert scanned documents and image-based text into machine-readable content, while preserving relevant layout and document structure where possible.
- **Relevance framing:** Cropping or otherwise processing images to focus the model on relevant regions and reduce visual noise.
- **Image anonymisation and redaction:** Detecting sensitive visual information, such as faces, identity numbers and addresses,

and obscuring the relevant regions through blurring or blacking out.

- **Structured content extraction:** Identifying

key fields, tables, clauses and other relevant content within documents and images, so that the information can be used effectively by downstream models.

2nd cohort

Case Study: Representative and Synthetic Data Generation for a GenA.I. Solution for AML Investigations R T

Background

A bank utilised the advanced natural language understanding and generation capabilities of LLMs to support Anti-Money Laundering (AML) investigations. Given the sensitivity of personally identifiable information (PII) and transaction-monitoring data and the need for reliable, explainable outputs, data preparation was treated as a core control rather than a purely technical exercise. The bank adopted the following approach to creating a representative, high-quality and privacy-protected dataset for model fine-tuning.

Approach

1. Select representative investigation cases:

The project team curated an initial dataset from the bank's transaction-monitoring system, working closely with experienced AML compliance professionals. Cases were selected to reflect the risk profile encountered in practice, including approximately 20% low-risk, 70% medium-risk and 10% high-risk cases, together with complex and edge cases. The dataset covered typologies such as rapid movement of funds, high-risk jurisdictions, unusual transaction velocity and deviations from customers' historical profiles. Non-suspicious cases were also included to provide negative examples and help reduce false positives.

2. Clean and de-identify sensitive data:

All selected cases underwent rigorous cleaning and de-identification before use. Personally identifiable information, including names, account numbers, identity-document details, exact addresses and precise timestamps, was removed or replaced with generalised tokens or randomised values. This preserved the underlying transaction patterns, relationships between counterparties and sequence of events, while irreversibly obscuring customer identities.

3. Standardise case information and enrich decision rationales:

Sanitised investigation narratives were converted from unstructured notes into standardised, machine-readable formats, with common activity categories and reason codes. In a subsequent refinement phase, the team added a "Decision Reason List" to explicitly connect case facts, suspicious behaviours, relevant AML red flags and the final investigative conclusion. This strengthened the explainability of the training data and helped reduce the risk of unsupported or

hallucinatory model outputs.

4. Generate and validate synthetic cases:

To expand the training corpus without exposing additional live customer data, the team developed a multi-stage synthetic-data generation process. Rather than simply masking or paraphrasing historical records, the process abstracted the underlying behavioural patterns of genuine AML cases and used them to construct new, artificial scenarios. This enabled the dataset to grow from approximately 100 sanitised historical cases to around 600 cases suitable for Low-Rank Adaptation (LoRA) fine-tuning.

5. Apply human oversight and quality assurance:

Before synthetic cases were included in the final training dataset, AML subject-matter experts reviewed randomised samples to assess plausibility, policy alignment and data quality. Findings from this review were fed back into the generation prompts and rules, creating a continuous improvement loop before deployment.

Outcome

The approach established a training dataset that balanced operational relevance, privacy protection and scalability. It reduced dependence on live customer records, broadened coverage of infrequent but material AML risks, and embedded clear investigative rationales into the training material. Most importantly, expert validation ensured that the final dataset remained consistent with the bank's transaction-monitoring policies and real-world AML investigation practices.

2.3 Data Augmentation

R T While consistent formats and terminology help standardise model outputs, exposure to diverse scenarios and user inputs is also essential to improve model adaptability. Some participating banks therefore used data-augmentation techniques to expand the scale and diversity of their training datasets, including through A.I. agents.

Data augmentation may also involve synthesising likely user inputs from existing knowledge sources. For example, potential customer enquiries can be derived from product advertisements, brochures and other customer-facing materials. Multiple specialised agents can

be coordinated to emulate different customer personas, generate question-and-answer pairs, verify their consistency with source materials, and identify gaps in coverage. Subject to appropriate evaluation, governance and quality-assurance controls, such synthetic data may be incorporated into training datasets to improve their coverage and diversity.

Various data augmentation techniques can be adopted for text, audio and image data. Data augmentation creates modified versions of existing training examples to improve model robustness and reduce overfitting.

Text Data Augmentation

R T

- Word or sentence shuffling: Randomly rearrange the order of words or sentences while preserving as much meaning as possible.
- Word replacement: Replace selected words with synonyms or contextually similar alternatives.
- Syntax-tree manipulation: Rephrase a sentence by modifying its grammatical structure while preserving its original meaning.
- Random word insertion: Insert relevant words at random positions in a sentence.
- Random word deletion: Randomly remove non-essential words from a sentence.
- Back translation: Translate text into another language and then translate it back to generate a paraphrased version.
- Voice cloning: Generate speech from the text using a cloned or synthetic version of a target speaker's voice.

Audio Data Augmentation

R T

- Noise injection: Add background noise to audio samples to help the model perform better in noisy environments.

- Time shifting: Shift the audio left or right by a random number of seconds.
- Speed adjustment: Speed up or slow down the audio by a fixed or random rate.
- Pitch shifting: Randomly raise or lower the pitch of an audio sample without substantially changing its duration.
- Volume adjustment: Randomly increase or decrease the audio volume to improve robustness to different recording conditions.

Image Data Augmentation

R T

- Geometric transformations: Randomly flip, crop, rotate, translate, stretch, or zoom images. Apply combinations of transformations carefully, as excessive changes can alter the label-relevant content of an image.
- Colour-space transformations: Randomly adjust RGB channels, brightness, contrast, saturation, or hue.
- Kernel filters: Randomly sharpen or blur images to improve robustness to differences in image quality.
- Random erasing: Mask or remove a randomly selected region of an image, encouraging the model to rely on broader visual features.
- Image mixing: Combine two or more images and their labels.

2nd cohort

Case Study: Dataset Construction for a Cantonese A.I. Voicebot R T

Background

A bank's call centres operate in a linguistically complex environment in Hong Kong, where customers commonly switch between Cantonese and English within the same conversation. At the same time, calls may be affected by background noise, varied customer accents and recording environments.

The objective was to construct a knowledge base for development of a voicebot to enable speech

recognition, semantic understanding and natural human-like voice interaction.

Approach

1. Selection of High-Frequency Service Scenarios

The bank engaged frontline call centre staff to identify customer-service scenarios that occurred frequently. The selected scenarios included credit card application, payment-method enquiries, credit-point checks, contact-information updates and discount enquiries.

For each scenario, standard business boundaries were defined. These boundaries set out the intended scope of customer questions, acceptable service responses, terminology and escalation limits. This ensured that the resulting dataset was relevant to real banking operations while remaining focused on approved and standardised service interactions.

2. Dialogue Script Generation

Large language models were used to generate structured dialogue scripts in batches. The scripts were designed to cover typical customer questions and follow-up questions, standard customer-service replies, Cantonese and English banking terminology, credit-card and payment-related product terms, Hong Kong-style Cantonese-English code-switching, variations in wording, sentence structure and dialogue flow.

The use of structured script generation enabled the team to create a broad and consistent set of test materials efficiently. Scripts were reviewed against defined business boundaries before being used for audio generation.

3. Text-to-speech Generation

Internal staff members' real voices were used as acoustic references, subject to applicable consent, privacy and internal data-governance requirements. The objective was not to reproduce individual identities, but to capture speech characteristics relevant to Hong Kong banking-service interactions, such as speaking pace, pronunciation patterns and tone. Text-to-speech (TTS) tools, including **Qwen3-TTS**, **CosyVoice3**, and **edge-tts**, were used to generate intelligible and natural-sounding Cantonese service dialogues (Table 2).

4. Audio Data Augmentation

To assess the voicebot's performance under realistic Hong Kong banking-service conditions, the dataset was augmented for different scenarios:

- **Noise volume:** Signal-to-noise ratio (SNR) levels of 20 dB, 10 dB, 0 dB and -5 dB were applied to simulate environments ranging from quiet to extremely noisy. These levels corresponded approximately to noise proportions of 1%, 10%, 50% and 75%, respectively.
- **Real-scenario noise:** A noise library was incorporated to reproduce common indoor and

- outdoor service environments in Hong Kong. Noise types included car beeps, rainfall, teahouse chatter, street noise, MTR sounds, crossroads, exhibition venues and supermarket background noise.
- **Speaker diversity:** The audio datasets were adapted to cover speakers with different age groups, genders, speaking styles and regional accents.

Outcome:

The bank created a tailored audio dataset for a Cantonese A.I. voicebot. The framework can be extended over time by adding new banking scenarios, customer speech patterns, regional accents, emerging TTS technologies and newly observed fraud typologies. This enabled the bank to develop an adaptive A.I. voicebot tailored for Hong Kong banking scenarios.

Feature	Qwen3-TTS	CosyVoice3	edge-tts
Open-Weight or Proprietary	Open-weight	Open-weight	Proprietary
Model Size	0.6B and 1.7B	0.5B	N/A (Lightweight client wrapper)
Hosting Model	Self-hosted	Self-hosted	Rely on Microsoft's hosted cloud architecture
Hardware Requirement	High (~8GB VRAM NVIDIA GPU)	High (~8GB VRAM NVIDIA GPU)	Low (Runs on any CPU)
Primary Strength	Voice cloning (requires a 3-second audio sample) and voice design (generating a unique voice simply by typing a natural language description)	Zero-shot voice cloning and expression control (emotion, style, rhythm, and prosody)	Zero computational overhead

Table 2 - Examples of Text-to-Speech (TTS) Tools

2.4 Data Quality Check

R T A comprehensive data quality review is essential to identify deficiencies and ensure datasets meet required standards. When preparing training data, participating banks assessed both common scenarios and exceptional cases for consistency, addressing missing data by identifying optional or frequently

absent fields and applying suitable treatments, such as imputation, labelling values as “Unknown”, or excluding consistently incomplete non-essential fields. Datasets were checked manually or automatically for compliance with the unified schema. The process was also documented for tracking, including original format differences, mapping rules, and tools or

scripts used, to support reproducibility and future enhancements.

For multi-agent A.I. systems, some participating banks adopted real-time validation for data quality check to verify and correct data before it impacts downstream A.I. agents. Various evaluation techniques, such as hierarchical tracing, deterministic code checks and LLM-as-a-judge, were deployed across the runtime pipeline to monitor orchestration, handoffs, and tool usage:

- **Instrument Execution Traces:** Map decision flows hierarchically from the root orchestrator down to individual worker agents using observability layers.
- **Implement Dual-Layer Guardrails:** The first layer, code checks (Deterministic) run fast, low-cost Python/TypeScript assertions inline to verify strict schemas, JSON formatting, token limits, and forbidden words instantly. The second layer, LLM-as-a-Judge evaluations (Non-deterministic) involve a faster, secondary LLM or dedicated evaluator package.
- **Validate Agent Handoffs and Tool Calls:** Intercept agent-to-agent (A2A) communication to confirm that parameter passing, tool selection accuracy, and declared capabilities match the orchestrator's central plan.
- **Set Up Automated Circuit Breakers:** Define explicit performance thresholds or negative assertions (e.g., safety checks, bias detection) that halt execution, trigger fallback logic, or alert designated reviewers when drift or degradation occurs.

2nd cohort

Case Study: Image Dataset Curation for a GenA.I.-Enabled eKYC Fraud Detection

Solution **R** **T**

Background

The bank sought to strengthen its electronic Know Your Customer (eKYC) fraud-detection capability during account opening, particularly for risks involving fraudulent identity documents, manipulated facial images, deepfakes, and potentially coordinated mule-account activity. A primary challenge was the limited volume of fraud cases available for model development and testing. Historical cases may not fully represent the range of variations encountered in a live operating environment. This can constrain the performance, coverage and adaptability of fraud-detection models. To address this challenge, the bank implemented a controlled data preparation strategy to expand the available training data using high-level characteristics learned from historical cases, without reproducing or disclosing specific fraud methods, case details or sensitive patterns.

Approach

1. Risk Pattern Learning

The bank reviewed the historical risk cases, covering fraudulent ID and Deepfake, to identify high-level statistical, visual and behavioural characteristics relevant to each fraud category. The analysis focused on general attributes needed to distinguish suspicious submissions from legitimate ones.

No individual customer information, actual case content or specific attack methods was used as a direct template for the generated data.

2. Primary Image Data Collection

A dataset containing approximately 30,000 synthetic facial-recognition images were collected from publicly available datasets, including Labeled Faces in the Wild (LFW), CelebA. These images provided a controlled evaluation dataset for testing image similarity, semantic search, recall, and clustering performance across diverse facial and image conditions.

3. Synthetic Case Generation

GenA.I. models were used to create synthetic samples reflecting the general characteristics of the relevant risk categories. The generation process was governed by predefined controls to produce a broad and balanced range of scenarios. Parameters such as submission conditions, document categories, image characteristics and risk intensity were varied systematically. This enabled the bank to improve dataset diversity while avoiding over-reliance on a small number of historical cases.

4. Data Masking for Privacy Protection

Before being uploaded into the testing environment, the images were processed using BgRemover, an A.I.-based masking tool. Sensitive information was removed or obfuscated to reduce privacy risk while retaining the visual characteristics needed for controlled model evaluation.

5. Vector Embedding for Image

The prepared images were converted into vector embeddings using CLIP, SigLIP, and Qwen3 models. These embeddings convert images into numerical vector representations, allowing images to be compared based on their visual and semantic similarity.

6. Quality Assurance and Filtering

All generated and prepared samples were subjected to quality-assurance screening and validation before being included in training, evaluation or testing datasets. Samples that did not meet the required quality, relevance or diversity thresholds were excluded.

Outcome

This data preparation strategy provided a scalable method for addressing the limited availability of confirmed fraud cases. It enabled the bank to establish a larger, more balanced and quality-controlled dataset for the development of a more resilient eKYC fraud-detection solution while maintaining appropriate controls over sensitive case information and fraud-related intelligence.

2.5 Continuous Data Updates

R T The participating banks implemented various techniques, including automated or semi-automated closed-loop data pipeline, to keep the A.I. agents aligned with evolving regulatory, market, product, and customer contexts. The pipeline continuously ingested new information from approved sources, such as regulatory updates, customer-feedback channels, and product-information repositories, and applied consistent cleansing and standardisation procedures before it is used for training (Figure 4).

This is particularly important for open-world tasks, where A.I. agents could combine search, tool use, and visual reasoning across changing text and image environments. Traditional training data is often generated through fixed processes and can quickly become insufficient as the A.I. agent's capabilities develop.

- **Data ingestion and pre-processing:** Regularly collect approved new data and process it through established cleansing, validation, and standardisation procedures.
- **Forward data-generation pipeline:** Generate initial task seeds, explore relevant web and visual evidence, construct multimodal evidence graphs, and curate verifiable training tasks.
- **Adaptive data evolution:** Use identified failure patterns to adjust the next training epoch's data-generation configuration, focusing on knowledge gaps and capabilities the current policy has yet to master.
- **Periodic model retraining:** Retrain the model on the refreshed and expanded dataset to prevent performance degradation and maintain alignment with current language, regulations, market conditions, and task requirements.

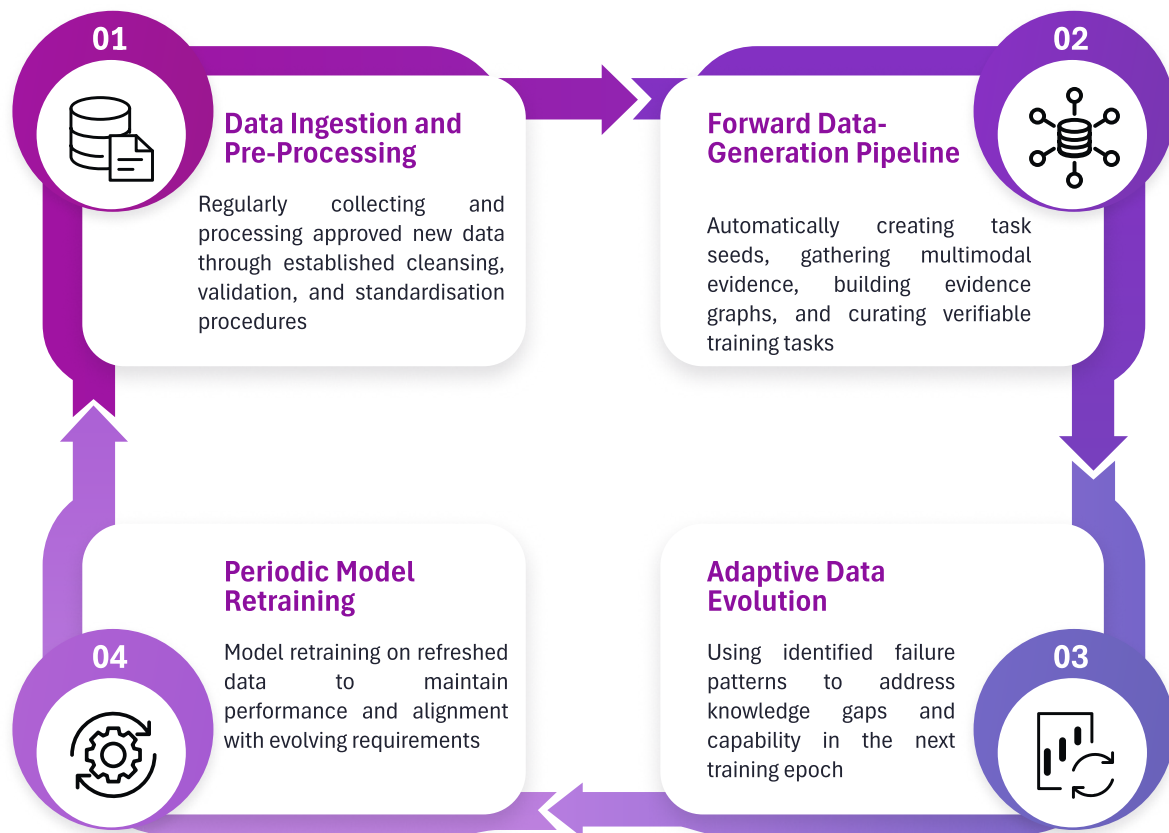


Figure 4 – Continuous Data Updates Techniques

3. Approaches to solution design

3.1 Model Selection

R T Model selection materially affects solution quality, latency, resource requirements and fine-tuning effort. Experience from the first cohort of the GenA.I. Sandbox showed that model choice is not a one-time decision. Participating banks evaluated multiple models against use-case-specific criteria, including task performance, reasoning, multilingual and multimodal capabilities, model openness, licensing, context window, compatibility, infrastructure requirements and ecosystem support. The results indicated that fit-for-purpose performance depends more on aligning model characteristics with the use case, available data and required level of operational control than on selecting the largest or most complex model.

Building on this experience, participants in the second cohort shifted their focus from capability testing towards secure and reliable

implementation, with greater attention to risk controls. A number of use cases adopted multi-agent designs in which work was divided among specialised agents powered by the same or different models. This approach enabled complex workflows to be decomposed, validated and governed, making multi-agent design an important architectural approach for tool-driven A.I. workflows.

Multi-agent Design: How the Workflow is Split

A R T A multi-agent design breaks down an end-to-end process into distinct roles, each assigned a specific task, equipped with a set of allowed tools, and operating within a defined scope. Typical roles in the second cohort use cases include task routing, retrieval or extraction, as well as domain reasoning, drafting, and quality-assurance. Processes like risk monitoring, fraud investigation, and customer service are well-suited for this approach, as their workflows naturally involve multiple steps, systems, and controls. Splitting the work enables human check point or rule gate to sit between agents and track the tasks performed by the agent to enhance the traceability.

2nd cohort

Case Study – Building a Multi-Agent RM Assistant **A R T**

Overview

In the second cohort of use cases, several banks focused on A.I. agents designed to support internal operations. One bank sought to improve the productivity of Relationship Managers (RMs) while maintaining strict controls over client confidentiality, suitability, and regulatory compliance. Rather than deploying a single general-purpose chatbot, the bank designed a multi-agent RM assistant in which specialised A.I. agents performed discrete tasks under a central orchestration and control layer.

Solution Design

1. Fix what the agents cannot do: The agents provide decision support only and will not give final

advice, determine suitability, approve a customer communication or execute a transaction. A recommendation, a draft communication and a pre-filled form all stop with the RM or other authorised personnel. If the information is missing, ambiguous or out of scope, the flow asks for clarification, returns a controlled response or escalates. It does not fill the gap.

2. Split by function, and keep each agent inside a boundary: The workflow is split by the nature of the request.

- **Orchestrator** reads the request, analyses the intent, and chooses the path.
- **Client-insight agent** retrieves and structures customer information.
- **Knowledge retrieval** agent retrieves approved knowledge source and policy material.
- **Communication** agent assesses tone and cues and drafts guidance.
- **Recommendation** agent matches that context to product options and records a rationale.
- **Compliance** agent reviews outputs for unsupported claims or risky wording, validates intent alignment, and returns pass, revise, block or escalate.
- **Action** agent prepares a pre-filled draft or other action-support material. An agent does not take another agent's source or action. Product, policy and customer outputs are grounded in an approved source.

3. Route by the request, not by a free plan: Orchestration is designed to be a controlled chat flow, rather than an open planning loop. The orchestrator selects a predefined path based on the detected intent, calls the agents on that path, and consolidates the outputs. Conditional routing, session variables and structured tool calls govern the sequence. A factual product enquiry can be answered by retrieval from an approved source. A customer-specific recommendation takes the longer path: customer information, product knowledge, a marked recommendation, communication support, then compliance review. Routine steps stay on the fixed path. A step that needs context or a risk check may use a more capable model. The record shows the path, the agents called, the sources used and the review outcome.

Example - Meeting Preparation Workflow for Client A

An RM asked the assistant:

“Prepare me for tomorrow’s meeting with Client A. Summarise recent portfolio movements, outstanding service requests and relevant market updates, and suggest an agenda.”

- **Orchestrator:** Validated the RM’s entitlement to access Client A’s records and routed the request to specialised agents.
- **Client Insight:** Retrieved the client’s profile, current holdings, and recent transactions to

summarise portfolio movements.

- **Knowledge Retrieval:** Accessed approved research, market updates, and relevant business information to support the discussion.
- **Communication:** Interpreted recent customer cues and drafted appropriate response strategies for potential discussion points.
- **Recommendation:** Prepared relevant deposit product options and supporting rationale based on the client's profile, marking them for RM consideration rather than final recommendation.
- **Compliance Review:** Checked all generated content against SOPs, internal guidelines, and PDPO requirements, flagging any compliance risks or missing disclosures.
- **Action Support:** Pre-filled application form fields for any potential next steps, preparing them for RM or authorised staff review.

The draft meeting materials will be sent to RM for further review before meeting with the client.

Outcome

The split saved work that a single model would have repeated on every request. A short enquiry called only the agents it needed, and each agent could be configured with the model suited to that step. Responsibility sat with a named agent: who retrieved, who drafted, who reviewed. The path, the agents called and the sources used were recorded, so the workflow could be traced. The RM remained responsible for anything sent, lodged or treated as advice.

Multi-model Stacks: What Sits Under Each Role

R A multi-model stack uses more than one foundation model inside the same use case. This is not a necessary consequence of having several agents. The same model can play every role through different system prompts; that is multi-agent, single-model. In a number of the second cohort use cases, worker agents that classify, extract or draft at volume are commonly backed by smaller, task-adapted models. Planner, teacher or judge agents more often use a larger open checkpoint or a closed-source service.

Model Hosting: Where Inference Runs

R Hosting is the third decision: on whose

infrastructure the model produces a next token, and therefore where prompts, retrieved context and completions travel. It is independent of whether the system is multi-agent or multi-model. The same open-weight checkpoint can be hosted in the bank's own environment, in a designated private cloud, at Cyberport's AISC, or behind a vendor's managed API. A closed-source model can usually be consumed only as a service because the bank does not host the weights.

Model Selection: Fit for Purpose

R In the second cohort, apart from the selection criteria mentioned above, some participants applied model selection criteria

tailored to each role's specific needs, given that each role has different responsibilities, data sensitivities, and operating patterns. It is noted that **more than 85% of use cases leveraged one or more open-weight models in the end-to-end workflow.**

Smaller open-weight model for high-volume, bounded and repeatable document or case-processing tasks: A smaller open-weight model deployed on bank-owned GPUs may provide higher throughput per GPU-hour than a larger general-purpose model, provided it meets the required quality standard. Throughput can be improved through inference optimisation techniques such as model quantisation and request batching, subject to memory, latency and accuracy trade-offs. A bank tested the use of InvestLM⁵ (32B parameters), and achieved comparable performance to a larger model (284B parameters), with about 90% less computational resources and over 99% system-level accuracy.

For well-defined workflows, systems that orchestrate smaller specialised models may achieve better end-to-end accuracy, latency or cost than a single larger model. The outcomes depend on the task design, models, orchestration and control framework.

To address sensitive-data protection concerns, an open-weight model can be deployed on bank-controlled infrastructure. With an appropriately configured on-premises or private-cloud deployment, customer text, documents and call recordings can remain within the bank's security perimeter rather than being transmitted to an external model provider.

Closed models as a service for reasoning and long-context workflows: Closed-source foundation models can be consumed through APIs or managed cloud services where a use case requires strong general reasoning performance, sophisticated tool use, or the ability to process and synthesise lengthy documents. This can be useful for workloads such as analysing complex policy material, comparing multiple documents, preparing research summaries, or coordinating multi-step workflows in which the model retrieves information and invokes approved tools.

An API-based approach lets the bank use a provider's hosted inference infrastructure instead of deploying and operating a comparable model on its own premises. This can reduce internal GPU demand and related model-serving, scaling, patching, and optimisation work, while providing quicker access to improvements in model capability and context-window size. These closed models are usually accessed via an external-service and data-sharing arrangement. The bank can retain source documents and retrieval systems internally, send only the passages needed for a request, redact or tokenise sensitive fields where feasible, and exclude unnecessary conversation history or operational data from prompts.

3.2 Optimisation Strategies

R T A number of common optimisation strategies (Figure 5) were employed by participating banks, which proved highly effective in enhancing the performance of their GenA.I. solutions. Through rigorous experimentation, these strategies were identified to capably improve model accuracy and overall effectiveness for different operational needs.

⁵ InvestLM: Generative AI for Finance and Investment, The Hong Kong University of Science and Technology School of Business and Management. <https://investlm.hkust.edu.hk/>

A number of use cases have shown that compact, fine-tuned language models can be highly effective solutions. By leveraging **knowledge distillation** from larger models and supplementing smaller models with targeted **data augmentation** and domain-specific **chain-of-thought annotation**, these smaller models can also closely match the analytical capabilities of their full-parameter counterparts.

Moreover, with the right training and evaluation approach, compact models can even surpass larger ones, delivering faster results, reduced resource requirements, and streamlined retraining. For instance, several

banks have successfully implemented various strategies, such as **supervised fine-tuning (SFT)**, to concentrate on specific domain knowledge and achieve better performance. These techniques were enhanced with RAG using vector databases, grounding responses in a knowledge repository to enhance accuracy and relevance. These optimisation measures improved both precision and accuracy by reducing irrelevant or incorrect responses and grounding outputs in authoritative, domain-specific information. Model optimisation also enhanced response consistency, reduced hallucinations, and lowered false-positive and false-negative rates.

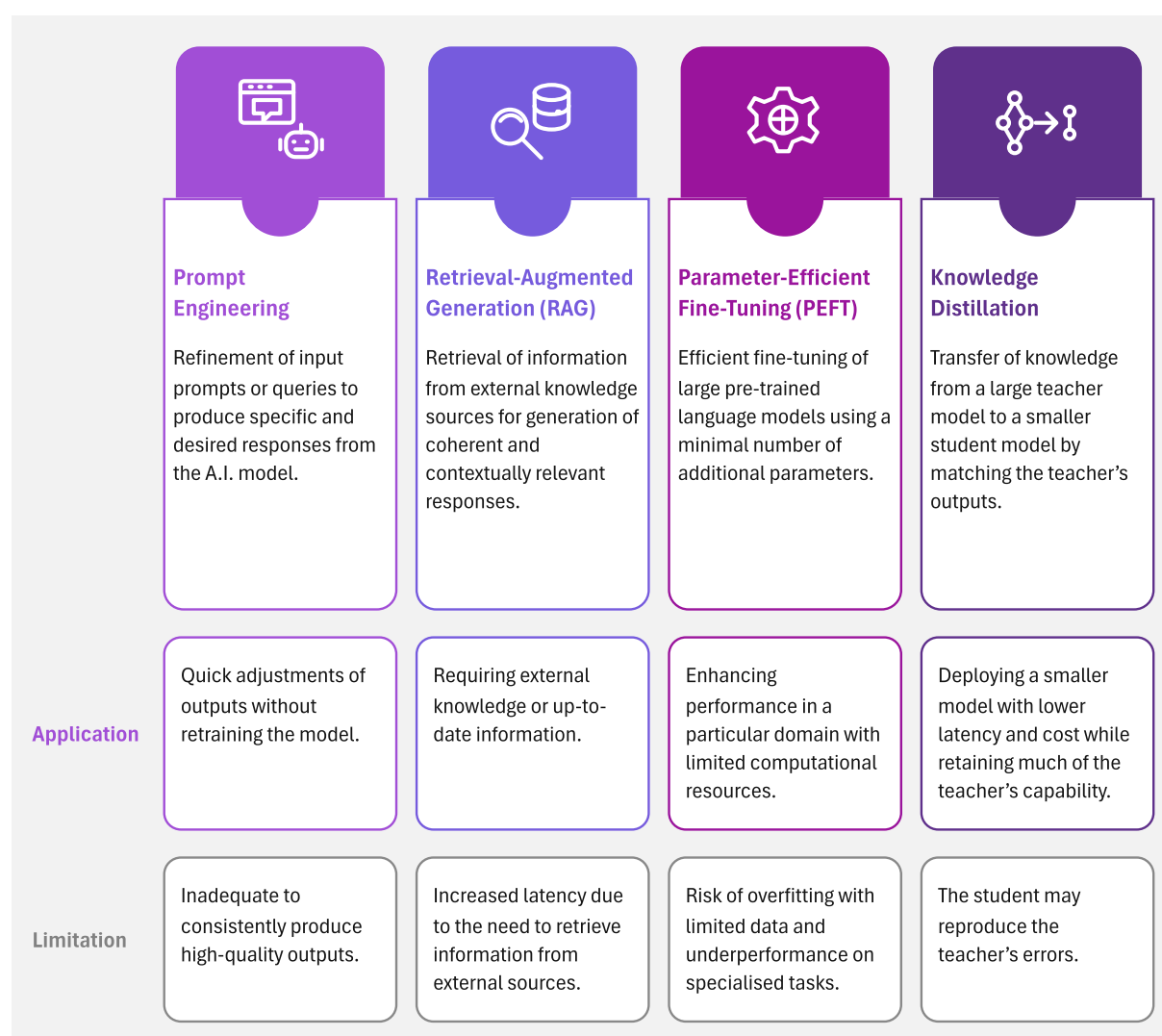


Figure 5 – Key Optimisation Strategies Adopted by Participating Banks in the GenA.I. Sandbox

3.2.1 Prompt Engineering

R T Prompt engineering is a fundamental yet crucial aspect of optimising GenA.I. outputs. This process involves the deliberate design, crafting, and refinement of input prompts or queries to produce specific and desired responses. The primary objective of prompt engineering is to generate high-quality outputs that are informative, relevant, and tailored to the requirements of the task at hand.

- **Zero-Shot Prompting:** presenting a prompt that is directly related to the task without providing any additional context or reference data.
- **Few-Shot Prompting:** a number of examples of the desired output are provided alongside the prompt, enabling the model to learn and adapt.
- **Chain-of-Thought Prompting:** decomposing complex tasks into a series of intermediate steps, where each step builds upon the previous one to produce a final response.

Several banks have successfully leveraged few-shot prompting, with some employing a structured instructional template that outlines clear instructions, input, and expected output, which is continuously refined throughout the project to ensure optimal results. Additionally, some banks incorporated in-scope questions with desired answers, as well as out-of-scope questions with valid responses, to further enhance the effectiveness of the prompting process.

Well-designed prompts can substantially enhance the accuracy, relevance, and coherence of the generated output, thereby improving

overall output quality. Moreover, prompt engineering can increase model efficiency by generating relevant responses rather than expending resources on unnecessary computations. It also allows for a more straightforward way to customise and adapt, as developers or even business users can tailor the prompts to specific use cases, applications, or domains without requiring advanced coding or engineering skills.

However, prompt engineering is not without its limitations. One of the primary challenges is the sensitivity to minor alterations, which can significantly impact the output and make it difficult to design robust and reliable prompts. Understanding why a particular prompt yields the desired output on one occasion but not another can be equally challenging, hindering the debugging and refinement of the prompt engineering process.

Banks generally found that sole reliance on prompt strategies was inadequate to consistently produce high-quality outputs and had to be combined with other strategies, such as Retrieval-Augmented Generation and supervised fine-tuning, to produce accurate, relevant, and structured outputs.

2nd cohort

Case Study - Optimising Trade Finance Document Processing with Prompt Engineering**R T****Overview**

This case study examines the implementation of advanced prompt engineering techniques for a bank trade finance process. The objective was to automate the extraction of structured data from complex trade documents (e.g., Commercial Invoices, Packing Lists, Bills of Lading) using LLMs. By applying specific best practices, the bank significantly improved extraction accuracy, reduced hallucinations, and ensured seamless integration with downstream API systems.

Challenges

Trade finance involves high-volume, multi-lingual documents like Commercial Invoices and Bills of Lading. Traditional OCR and rule-based parsing missed fields when the layout changed. Left unconstrained, the model summarised, inferred missing values and wrapped the answer in explanation, so the output could not be passed to an API. The task was to make it extract, not converse.

Solution Design

The bank fixed the prompt in four steps.

- 1. State the job:** The instruction named the task as extraction and disallowed summary, estimation and inference. English and Chinese documents were both in scope. A role label was added, but it was not the control. The control was the task boundary.
- 2. Lock the fields:** The prompt listed the fields to be returned, including document type, party role and invoice number, date and amount. Document type and role were drawn from closed lists, so the model could not invent a synonym.
- 3. Define a missing value:** If a field was not on the page, the model had to return null. It was told not to fill the gap from context or from a typical invoice.
- 4. Fix the output:** An exact JSON schema was given. The model was told to return raw JSON only, with no markdown and no explanation, so the response could be passed to an API.

Outcome

The results demonstrate measurable performance improvements without model fine-tuning, with prompt enhancement delivering relative gains of approximately 20% to 80% across key evaluation metrics including precision, recall and F1-Score.

3.2.2 Retrieval-Augmented Generation (RAG)

R T Standard LLMs, i.e. foundation model, lack access to real-time information and domain-specific knowledge. To overcome these limitations, some banks applied the RAG techniques, integrating the benefits of retrieval-based systems and generative models. This approach enabled the retrieval of relevant information from external knowledge sources and facilitated the generation of more coherent and contextually relevant responses.

With RAG techniques, the models can retrieve and provide current information, thus minimising the risk of generating inaccurate outputs. By grounding their responses on retrieved data, the participating banks were able to mitigate the issue of hallucination. However, the RAG approach also comes with certain trade-offs, in terms of increased latency due to the need to retrieve information from external sources, which can be a concern in applications where response time is critical.

3.2.3 Parameter-Efficient Fine-Tuning (PEFT)

R Parameter-Efficient Fine-Tuning (PEFT) enables the efficient fine-tuning of large pre-trained language models using a minimal number of additional parameters. By selectively updating only a fraction of the model's parameters, PEFT diverges from traditional fine-tuning methods that require updating the entire model. This targeted approach facilitates rapid adaptation of large language models to new tasks or domains, while minimising the need for substantial computational power and extensive task-specific training data.

Notably, several participating banks in the GenA.I. Sandbox successfully leveraged the PEFT technique, with many opting for Low-Rank

Adaptation (LoRA). The adoption of LoRA yielded significant gains in training efficiency for these banks. This technique demonstrated substantial improvements in training efficiency, achieving an optimal balance between model performance and resource utilisation. The method substantially reduced the number of trainable parameters, resulting in lower memory requirements and computational costs, making it a pragmatic solution for banks with limited resources. One participating bank has demonstrated significant reduction in hardware requirements and marginal accuracy loss below 3% when comparing a well-structured LoRA to full finetuning.

Moreover, some banks took a step further by utilising Quantised Low-Rank Adaptation (QLoRA) to achieve fine-tuning with quantisation. The quantisation can largely maintain model performance while reducing memory usage, enabling larger batch sizes and faster iterations.

3.2.4 Knowledge Distillation

R Knowledge distillation sits naturally alongside prompt engineering, RAG and PEFT as a way to **improve fit-for-purpose performance without putting the largest available model into every role**. The first cohort already used a related pattern: a larger model produced structured chain-of-thought annotations, which were then used to adapt a smaller model to a bounded task. Taking this as the starting point, banks are further exploring how the transfer is done.

- **The Teacher-Student Dynamic.** Distillation has been used to transfer accepted behaviour from a stronger model to a smaller one. Unlike prompt engineering, the student model's weights are updated. Moreover, unlike PEFT, a

separate teacher model is involved, and that teacher model does not have to remain in the live path once the student model meets the bar defined by the bank.

- **Transfer Techniques.** In the first cohort, most cases are primarily used teacher-written demonstrations for the student to imitate. This remains the preferred approach for short, well-labelled tasks. As observed through the second cohort, practice has also included **scoring the student's own outputs**. This method is particularly effective when the student is

expected to extend its own previous text, such as in longer drafting processes or agentic tool-use scenarios.

- **Evaluation and Governance.** Where distillation has been used, acceptance has been based on the bank's own evaluation set rather than on how closely the student resembles the teacher. A smaller copy of an incorrect answer remains incorrect. Items that change frequently have more often been left in retrieval than written into the distilled weights.

2nd cohort

Case Study - Distilling a banking-knowledge model R

Overview

A participating bank ran a three-stage teacher-student distillation on a fixed set of banking-knowledge questions. The larger model is costly to call at volume and load everyday. Distillation transfers the teacher's behaviour on these questions to the student, so that later testing can run on the student alone.

Solution Design

The teacher was Qwen3-235B-A22B, a quantised mixture-of-experts model, used only for distillation and for CFLUE benchmarking. The student was Qwen3-32B. Distillation was run on the offline track. Teacher logits were compressed to the top 1,000 tokens to manage storage, and the student was trained with a combined distillation and supervised-fine-tuning loss. Training used four H800 GPUs for the teacher and one to three for the student. The engineering stack included token alignment, FSDP and Flash Attention. The test set was the 3,864-question CFLUE benchmark.

Outcome

After distillation, the student scored 75.1% on the full set and the teacher 74.8%. **The student model (Qwen3-32B) has successfully outperformed the teacher (Qwen3-235B) model.** Against the student's own base, the gain was 2.82 percentage points. On this set, the smaller student was ahead of the teacher overall.

3.3 Validation and Evaluation

3.3.1 Model Validation

R All participating banks conducted extensive trials to validate various setup options their A.I. solutions for their suitability. The following key approaches were adopted during the trials:

Prompt Structure Validation: Testing the system with a variety of scenarios, including ambiguous or challenging queries, to evaluate and enhance the clarity and effectiveness of the system-level instructions.

Deliberately crafting inputs designed to mislead or deceive the model, and identifying potential vulnerabilities and weaknesses in the system, such as its susceptibility to biased or misleading data.

Example:

Create a set of test scenarios that simulate different user interactions as well as abnormal cases, such as:

- Scenario 1: User asks a simple question (e.g. "What benefits do you offer for the Platinum Credit Card?")
- Scenario 2: User asks a follow-up question (e.g. "How can I apply for it?")
- Scenario 3: User provides incorrect or ambiguous information (e.g. "How can I apply for a Credit Card issued by another bank.")

Hyperparameter Validation: Rigorously testing the effectiveness of different hyperparameters with multiple combinations to pinpoint the most successful approaches and optimise performance.

Base Model Validation: Establishing A/B testing frameworks to facilitate selection of suitable foundation models, taking into consideration the models' differences in architecture, parameter sizes, language capabilities etc. Specifically, some banks have adopted structured language evaluation metrics to assess multilingual performance.

Example:

A set of adversarial examples can be used to test the model's robustness, such as:

- Adding noise to the input text (e.g. typos, misspellings)
- Creating input text that is intentionally ambiguous or contradictory (e.g. "I hate this product, but it's amazing.")
- Using adversarial attacks such as gradient-based attacks to generate input text that is designed to mislead the model

Validation Iteration:

Large language models can typically be validated on an iterative process:

- Apply a validation dataset after training the model on training dataset (early stop or smaller training dataset can be considered for more efficient validation cycles)
- Assessing the model's performance based on the selected metrics
- Using the validation results to adjust the model's hyperparameters (e.g. learning rate, batch size) and re-train the model

- Repeating this process for multiple iterations, monitoring the model's performance and adjusting the hyperparameters as needed

Example:

The performance of two (or more) different language models on a sentiment analysis task can be evaluated following the steps below:

- Train both models on the same dataset and evaluate their performance on a held-out test set using metrics such as accuracy, precision, and recall.
- Evaluate the models' performance on out-of-domain data, such as text from a different genre or style.

Compare the models' computational requirements, such as training time, memory usage, and inference speed.

Customisation Approach Validation: Applying different customisation approaches, such as full finetuning and LoRA, over similar validation iterations to identify techniques that are more suitable for the use cases. During the validation process, other customisation techniques such as RAG can also be adopted to aid the selection of the most suitable technique for a given scenario, where applicable. For customisation with RAG, different augmentation configurations can also be validated, such as embedding models, chunk sizes, retrieval settings, etc.

The first-cohort report set out examples of model validation, including prompt and hyperparameter testing, conventional generation metrics, human review, and LLM-as-judge scoring. Since then, it

has become clearer that these methods were designed primarily to assess a completed piece of text.

Participating banks that have advanced their evaluation practices are therefore assessing both the process and the final output. This can include checking whether the appropriate source was included in the context, whether a schema or policy requirement was satisfied, and whether the same case can be reproduced under a fixed configuration. Public benchmarks remain useful reference points, but they do not replace the bank's own use cases and acceptance criteria.

3.3.2 Model Evaluation

A R Model Evaluation is a critical process for assessing the performance and effectiveness of GenA.I. models to ensure they meet the desired standards of reliability, accuracy, and ethical deployment. Participating banks adopted a structured approach to evaluating their GenA.I. systems, leveraging a range of metrics to assess the performance:

Standard and Specialised Evaluation Metrics: For classification-related tasks, standard evaluation metrics such as precision, recall, F1-score, and confusion matrix were commonly employed. Additionally, some banks utilised specialised metrics, such as the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) score for text summarisation tasks, the Bilingual Evaluation Understudy (BLEU) score for machine translation tasks, and Natural Language Processing-based metrics, such as BERTScore, for question-and-answer tasks.

Types of Evaluation Metrics:

Precision

Precision measures the accuracy of positive predictions, which indicates the percentage of actual positive out of all items the model identified as positive.

Recall

Recall measures the completeness of positive predictions, which indicates the percentage of positive instances that the model correctly identifies out of all actual positive.

F1 Score

$F1 = 2 * (Precision * Recall) / (Precision + Recall)$. This score ranges from 0 to 1, with 1 being the best possible score. For LLMs, the F1 Score is particularly useful in tasks with binary or categorical outputs, such as text classification and named entity recognition.

Confusion Matrix

Confusion matrix is a table showing the number of correct and incorrect predictions. For binary classification, it includes:

		Actual	
		Positive	Negative
Predicted	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Recall-Oriented Understudy for Gisting Evaluation (ROUGE)

ROUGE score measures the similarity between machine-generated text and human-written references, focusing on word accuracy and sequence structure, ranging from 0 to 1, with higher scores indicating better performance.

Bilingual Evaluation Understudy (BLEU)

BLEU score evaluates how well a generated summary captures the key information and structure of a human-written reference. Both metrics are scored on a decimal scale from 0 to 1, with higher scores indicating better performance.

BERTScore

BERTScore evaluates the similarity of the model outputs and the corresponding reference answers, leveraging token embeddings to allow quantitative evaluation measurements. This approach provides flexibility for models to paraphrase outputs as long as the semantic meanings are preserved.

Neutrality

Neutrality assesses whether model outputs present information objectively and avoid unjustified preference towards particular viewpoints, entities or groups. It can be evaluated through human review or predefined criteria, including balanced framing, factual language and consistent treatment of comparable subjects. As there is no universal neutrality metric, the evaluation methodology and scoring scale should be clearly specified.

Multifaceted Evaluation Metrics: In evaluating the outcomes of GenA.I., several banks reported employing a multifaceted approach, incorporating both quantitative and qualitative metrics to gain a comprehensive understanding of system performance. This holistic evaluation framework enabled banks to assess the effectiveness of their GenA.I. systems from multiple angles, ensuring they could meet the evolving needs of the business.

Human Oversight in Evaluation: Some banks also emphasised the importance of human feedback to further refine their GenA.I. systems. By designing feedback mechanisms that allow users to assess the accuracy and performance of the system, the banks can gather valuable insights and ensure that their GenA.I. systems remain aligned with business objectives. This ongoing human oversight supports the identification of improvement opportunities and potential biases, and enables the continuous optimisation of GenA.I. systems.

User Testing Feedback: Provides direct evaluation and insights from internal testers who simulate real-client interactions, offering a first hand assessment of the system's performance.

Continuous Evaluation: Establishing continuous re-evaluation pipelines based on a curated set of multifaceted evaluation metrics to ensure model performance remains up-to-date with changing market conditions.

The evaluation methods observed among participating banks are summarised in Figure 6.



01 Standard and Specialised Metrics

Precision, recall, F1-score and confusion matrix for classification. ROUGE for summarisation, BLEU for translation, and BERTScore for question-and-answer tasks.



02 Multifaceted Evaluation

Quantitative and qualitative metrics used together, to assess performance from multiple angles and meet evolving business needs.



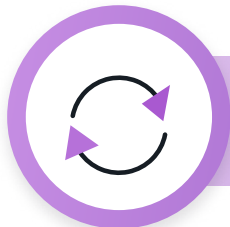
03 Human Oversight in Evaluation

User feedback on accuracy and performance to refine systems, address bias, and stay aligned with business objectives.



04 User Testing Feedback

Simulation of real-client interactions performed by internal testers, providing a first-hand assessment of system performance.



05 Continuous Evaluation

Ongoing re-evaluation pipelines on a curated metric set, so performance stays current with changing market conditions.

Figure 6 – Model Evaluation Methods

Examples of Human Oversight in Evaluation

1. Evaluation with User Feedback

A participating bank conducted evaluation on the GenA.I. model for generating investment market commentary videos based on visual fidelity, motion accuracy, lip-sync precision, and linguistic accuracy across languages, as well as user feedback.

To evaluate user experience, the bank designed a survey to collect feedback from members of an

insight community consisting of the bank's customers who voluntarily participated in pre-launch testing. Survey participants were asked to rate different aspects such as smoothness, naturalness, and overall satisfaction on the interactions with the applications.

2. Integration and Refining with Human Review

Another bank experimented using GenA.I. to handle AML/KYC relevance classification tasks. The model determines whether the matched information is referring to the same entity with three possible results:

- Relevant: The matched information is referring to the same entity.
- Irrelevant: The matched information indicating a different entity.
- Not Sure: Information is insufficient to make a decision.

However, the prediction of "Not Sure" cases showed a low precision due to insufficient training data of ambiguous inputs. To address this issue, the bank introduced a probabilistic uncertainty threshold mechanism, allowing the model to flag ambiguous cases for human review to reduce false positives. Additionally, a feedback loop was established where flagged cases were reviewed by compliance experts, and their decisions were used to further refine the model.

1st cohort

Case Study – Validation and Evaluation of Fine-tuning Process for Anti-Money Laundering (AML) Applications R

A participating bank experimented on the fine-tuning process of the Qwen3-32B model for Anti-Money Laundering (AML) applications in three iterative cycles, each focusing on different aspects of model improvement. The quantitative measures used to evaluate the model's performance were BLEU and ROUGE scores, which assess the quality of generated text.

Cycle 1: Baseline Assessment

The baseline model without fine-tuning demonstrated some structural similarity to manual Suspicious Transaction Reports (STRs) but lacked linguistic precision and semantic depth. BLEU and ROUGE evaluations were not conducted due to insufficient quality.

Cycle 2: Prompt Engineering Enhancement

Through enhanced prompt engineering, including the use of few-shot examples, there were improvements in semantic and structural alignment, with better consistency in tone and AML terminology, but linguistic precision and formatting remained inadequate. BLEU and ROUGE scores were both below 0.02, indicating poor performance.

Cycle 3: Supervised Fine-Tuning

After applying supervised fine-tuning using 1,000 narrative samples with annotated fund flows, behavioural patterns and risk assessments, the model demonstrated strong linguistic and structural alignment with expert-written reports. It achieved both BLEU and ROUGE scores above 0.86, indicating excellent performance.

Conclusion:

The quantitative measures highlight the effectiveness of supervised fine-tuning in improving the model's narrative generation capabilities. The significant increase in BLEU and ROUGE scores from Cycle 2 to Cycle 3 (from below 0.02 to above 0.86) demonstrates the substantial enhancement in the model's performance, indicating that the fine-tuned model is a viable tool for augmenting investigative workflows.

3.4 A.I. Agents

A R T While GenA.I. focuses on creating original content from user prompts, A.I. agents possess the autonomy to reason, decide, and act. It independently pursues complex, multi-step goals, requiring only minimal supervision to achieve desired outcomes. Recognising the operational value of task-specific automation, several banks are initiating their A.I. transformation by deploying A.I. agents as one of the components of their systems. Rather than jumping directly to fully autonomous systems, these banks are leveraging the multi-functional autonomy of A.I. agents to handle complex, multi-step workflows. This approach combines the creative capabilities of GenA.I. with the operational autonomy of A.I. agents to achieve more comprehensive outcomes.

Single-agent

A R T A small number of banks have tested single-agent solutions, demonstrating promising logical reasoning capabilities in a relatively monolithic setup. However, the resulting business improvements were constrained. To

address this, these banks are investigating multi-agent architectures to further enhance performance. A bank noted that while a single-agent RAG approach facilitates access to relevant data, it integrates retrieval, reasoning, drafting, and validation into a single, less transparent generation process.

Multi-agent

A R T Instead of relying on a single general-purpose model to handle the entire workflow spanning information retrieval, communication support, compliance review, and action preparation, multi-agent solutions distribute these distinct tasks among specialised agents. A bank conducted trials on transaction monitoring tasks, revealing that accuracy improved when investigations were decomposed into smaller, well-defined tasks, such as incoming and outgoing transactions, counterparty analysis, and transaction summaries. This modular approach minimised ambiguity and enhanced the traceability of the individual report sections.

3.4.1 Advantages of Multi-agent Systems

A R T Multi-agent systems can enhance the design and operation of GenA.I. applications by dividing complex processes into specialised, governed and scalable components. The key advantages observed are summarised in Figure 7 and discussed below.

Enhanced Modularity via Functional Specialisation

Multi-agent systems decompose complex workflows into distinct, specialised components, such as intent routing, customer insight generation, knowledge retrieval, personalised recommendations, compliance verification, action execution, and auditability. By assigning each agent a defined business or control function, the system ensures precise task separation, allowing for greater accuracy, modularity, and efficient handling of specific stages within the workflow.

Enhanced Governance and Control

Multi-agent systems' modular structure simplifies prompt governance by allowing teams to refine specific functional areas without rewriting entire workflows. Furthermore, it improves reliability by grounding all outputs in approved knowledge sources and structured data, thereby maintaining compliance and accuracy.

Structured Workflow Management

Multi-agent systems enable precise control over the workflow by directing requests through predefined paths based on intent, context, and processing requirements. This structured routing allows for the integration of critical checkpoints, such as intent-alignment validation, document completeness checks and material confirmation, before any draft generation occurs, resulting in higher accuracy and compliance at every stage.

Preservation of Human Judgment and Full Traceability

Multi-agent systems support appropriate human

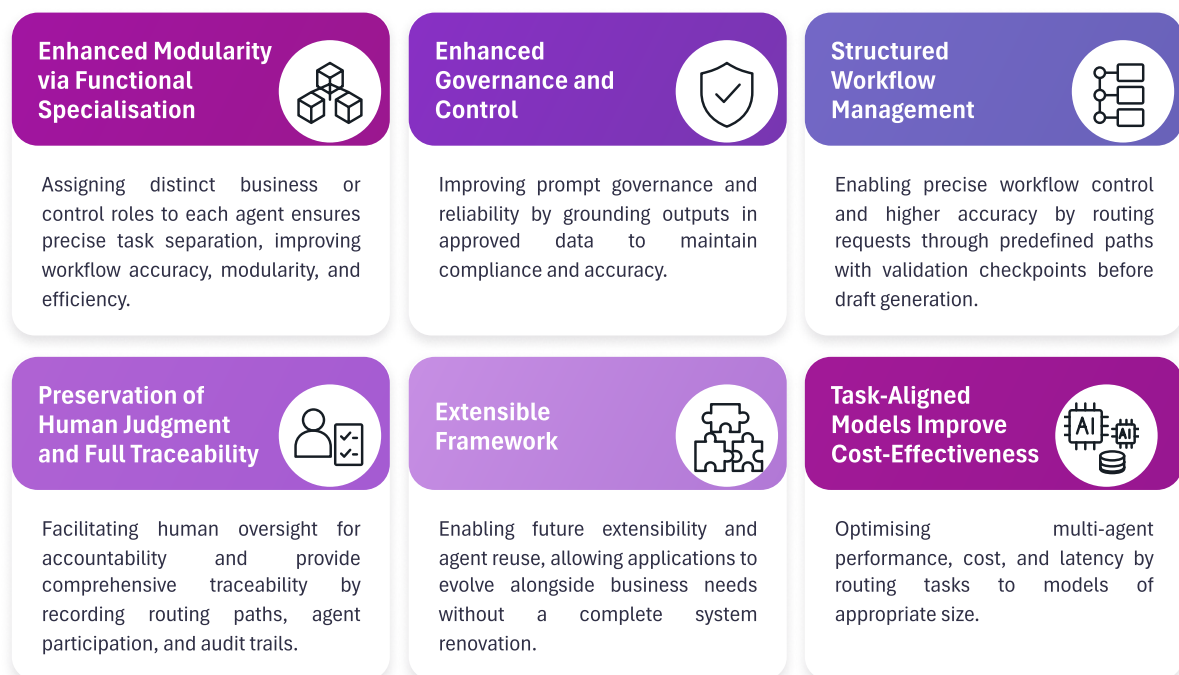


Figure 7 – Advantages of Multi-agent Systems

oversight and accountability by requiring authorised personnel to validate critical decisions, and operational inputs. This architecture also enables comprehensive traceability, where workflow records the routing path, participating agents, retrieved information, and final review outcomes to maintain a clear audit trail.

Extensible Framework

The system's modular architecture facilitates future extensibility, enabling the addition of new agents based on defined business value, data integrity, and governance arrangements. Such architecture also enables the reuse of common-purpose agents across different workflows, improving performance consistency. This flexibility allows applications to evolve alongside emerging business needs without requiring a complete system renovation.

Task-Aligned Models Improve Cost-effectiveness

Multi-agent architectures support model tiering, ensuring that only complex tasks utilise large, reasoning-heavy models. By assigning lightweight tasks to smaller models, banks achieve gains in cost-efficiency and reduced latency without compromising performance on critical, compliance-sensitive workflows.

3.4.2 Challenges & Limitations

A R T Not all A.I. applications require an agentic architecture as their inherent complexity may be disproportionate to the requirements of simpler use cases. Multi-agent frameworks introduce distinct operational and technical challenges that may render them unsuitable for specific banking solutions, necessitating a careful evaluation of their feasibility before

adoption.

Agent Coordination Complexity

While multi-agent orchestration enhances workflow control, it simultaneously introduces additional operational complexity. Unlike single-prompt solutions, this approach demands rigorous design of routing logic, workflow sequencing, prompt boundaries, intermediate output management, and error handling. Managing long-context capabilities extends beyond mere model selection, it requires a holistic strategy encompassing document chunking, retrieval filtering, prompt compression, summarisation of supporting documents, and precise control over the volume of information passed to each generation step.

Overhead of Agent-to-Agent Interaction

Inter-agent dependencies add another layer of intricacy. The cumulative effect of individual agent invocations introduces latency into the end-to-end response time. Specifically, sequential agent chains exhibit linear latency growth, necessitating a trade-off between the enhanced accuracy of multi-agent consultation and the imperative for rapid response. As such, optimising the execution model, whether sequential or parallel, represented a significant design consideration.

State Consistency

Maintaining contextual consistency among parallel agents is a complex challenge. For instance, when multiple agents simultaneously access and modify client-specific data within the same request cycle, ensuring that downstream agents receive a coherent and unified view necessitates rigorous state management.

Debugging and Maintenance

The complexity of multi-agent pipelines necessitates advanced logging and observability frameworks to identify root causes that span agent boundaries. While single-model systems allow for straightforward error attribution to individual components, multi-agent environments present unique challenges, as errors may originate from the intricate interplay between multiple agents.

3.4.3 Agent Intent

R **Agent intent** refers to the clearly defined objective, responsibilities and permitted scope of an A.I. agent within a system. It guides how an agent interprets requests, selects actions, uses tools and interacts with other agents. Defining agent intent helps ensure that each agent operates within appropriate functional and control boundaries, maintaining system reliability, regulatory compliance and a consistent user experience. By assigning distinct intents to specialised agents, banks can decompose complex workflows into manageable functions while

tailoring the overall design to their operational, risk and customer-service requirements.

While banks select and combine agents based on their specific operational needs, certain agent types have a particularly significant impact on the overall system effectiveness. For example, the **Orchestrator Agent** serves as the central coordination layer. It interprets user requests, translates intent, selects appropriate workflow paths, invokes relevant agents and tools, and consolidates outputs. It offers flexibility by dynamically selecting agents, determining execution sequences, and revising plans.

The orchestrator can also enforce control through intent-based routing. Simple enquiries may follow streamlined workflows, whereas action-oriented or compliance-sensitive requests can be directed through enhanced validation, approval or control layers. Well-defined agent intents also help prevent fragmented or duplicated activities across systems, enabling agents to collaborate within clear functional boundaries and support the efficient, cohesive completion of complex tasks.

2nd cohort

Case Study - Architectural Trade-offs: Operational Reliability vs. Conversational Naturalness **R**

A bank evaluated two distinct architectural designs for A.I. agents, identifying workflow orchestration and autonomous orchestration as the key approaches. Recognising that the general capabilities of pre-trained models were sufficient for the specific business scenario, both approaches did not involve fine-tuning LLM. The team relied on prompt engineering, parameter tuning, and architectural design to drive behaviour adaptation.

1. Approach A: Determinism via Code-Level Control

Approach A leverages the Dify low-code platform to implement a 'strong control' paradigm, restricting the LLM's behaviour to a minimal necessary scope, such as intent recognition and parameter extraction. This architecture ensures determinism at the code layer rather than relying on

the model layer.

- **Mechanism:** The system uses a Switch-Case control flow similar to traditional code, forcing the model to choose between mutually exclusive logical paths. This approach helps avoid the ambiguity and multiple meanings often found in natural language.
- **Strengths:**
 - **High Reliability:** Achieves high task completion rates under standard input scenarios.
 - **Predictability:** Offers clear fault boundaries and predictable behaviour, making it the most stable baseline for the bank's current operations.
 - **Robust Fallback:** Effectively manages errors through code-level strong validation and mandatory confirmation cards.
- **Limitations:**
 - **Rigid Interaction:** The fixed templates expose flaws in multi-turn interactions. Due to extraction omissions, users are sometimes re-asked for information they have already provided, which can lead to a more frustrating user experience.

2. Approach B: Autonomy via Guardrail-Based Constraints

Approach B leverages the LangGraph Agent framework to implement a 'light constraint' paradigm. It prioritises user experience and autonomy through flexible prompting and agent collaboration, utilising multi-layered guardrails as a fallback mechanism.

- **Mechanism:** The system adopts a layered collaborative architecture comprising seven specialised agents. Since different tasks impose vastly different demands on model capabilities, the architecture is designed to handle complex requirements; for instance, risk detection necessitates rigorous compliance judgment.
- **Strengths:**
 - **Superior User Experience:** Demonstrates high fluency in conversational naturalness, effectively handling multi-intent queries.
 - **Flexibility:** Better adapts to complex, unstructured user inputs compared to rigid templates.
- **Limitations:**
 - **Complex Error Patterns:** Exhibits more sophisticated failures, such as:
 - RAG mismatches.
 - Function Calling boundary failures and incomplete state machine rollbacks.
 - **Higher Maintenance Cost:** While errors are recoverable, they are more costly to diagnose

and fix, requiring robust observability infrastructure.

Conclusion

The choice between Approach A and Approach B represents a fundamental trade-off between operational stability and user experience. Approach A serves as the safe, stable baseline for critical transactions, while Approach B offers a superior conversational experience at the cost of increased complexity in error management and monitoring.

3.4.4 Governance on A.I. Agents

A R T A.I. agents operate through decomposed processes, intent-dependent behaviours, and active interaction with enterprise systems. This dynamic nature means that traditional governance models designed for static, conventional A.I. models might be insufficient. Effective governance may extend beyond validating model outputs to actively controlling agent actions, monitoring execution, and enforcing accountability. This requires a dual focus on establishing strict **Agent Boundaries** to define operational scope and implementing robust **Agent Intervention** mechanisms to manage exceptions, validate quality, and ensure human oversight.

Agent Boundary: Defining Scope and Operational Limits

A R T To mitigate risk, banks can clearly define agent scopes, establish system boundaries, and enforce rigorous pre-deployment validation aligned to the agent intent. A robust governance model requires strict policy controls that explicitly delineate what an agent can and cannot do.

1. Strict Role Definition and Prohibitions

Agents can be programmed to operate within narrow, well-defined task boundaries. For instance, one bank has established that its

agents serve strictly in a **decision-support capacity**. They are explicitly prohibited from independently providing final advice, determining product suitability, approving customer communications, or executing transactions. All recommendations, communications, and action-support outputs generated by these agents require mandatory human review before finalisation.

2. System Prompts and Abstention Conditions

Governance is reinforced through role-specific system prompts that explicitly define input and output requirements, source-grounding rules, language constraints, and context handling. Crucially, these prompts include **abstention conditions**. Agents are programmed to handle uncertainty appropriately, when information is unavailable, ambiguous, or out of scope, the workflow is designed to request clarification, return a controlled response, or escalate for review. Agents are not expected to infer unsupported facts, product terms, or regulatory conclusions.

3. The Three Lines of Defence in Boundary Setting

As agentic A.I. evolves, the three lines of defence may proactively adapt to enforce these boundaries:

- **First Line:** Takes greater responsibility for

validating that solutions operate within intended boundaries, embedding intent-anchored accountability into the development process.

- **Second Line:** Oversees the framework that shapes agent decision-making. This involves setting governance standards, defining control boundaries, and monitoring expectations for how agents operate within those limits.
- **Third Line:** Assesses the control framework to confirm it is appropriately designed and operates effectively. This requires rigorous auditing of orchestration compliance and the ability to demonstrate full traceability, specifically, which agent acted, under whose delegation, with what intent, and within what conduct limits.

Agent Intervention: Validation, Oversight, and Human Accountability

A R T While boundaries define the "safe zone" for agents, intervention mechanisms manage deviations, ensure quality, and provide necessary human oversight. This involves both automated "A.I. vs. A.I." checks and structured human intervention protocols.

1. Evaluator Agents and "A.I. vs. A.I." Strategies

Several banks have adopted an 'A.I. vs. A.I.' strategy, deploying secondary evaluator or checker agents to validate outputs from primary agents prior to release. These agents review A.I.-generated content from upstream workflows, scoring them on content accuracy, completeness, compliance, logical consistency, and prompt injection risks.

- **Feedback Loops:** When an evaluator

agent scores a report below a defined threshold, it triggers specific feedback mechanisms, such as initiating a retry loop or escalating the issue for human review.

- **Heterogeneous Inspection:** A bank observed that employing a heterogeneous inspection model (a checker agent from a different model family than the primary agent) is more effective than a homogeneous model for detecting logical mismatches and implicit violation expressions.
- **Performance Impact:** One bank integrated an external guardrail layer with internal checking agent to block prompt injection, filter sensitive wording, validate rewritten queries, and check for improper content. This comprehensive approach significantly reduced misinformation. This approach improved misinformation detection accuracy from 81.6% to 92.1% during end-to-end red-teaming exercises, effectively mitigating hallucination risks.

2. Multi-Party Checker Mechanisms

To enhance governance and counteract potential biases, some banks have implemented multi-party checker mechanisms. For example, one bank integrated one internal agent with two external agents from different domains. The integration of these third-party agents serves to provide diverse perspectives, enhancing the detection of nuanced errors or biases that a single internal model might overlook.

3. Human Oversight

Banks recognise the importance of maintaining

human oversight, particularly before critical actions are executed. However, traditional human-in-the-loop mechanisms often fail due to passive reviewer behaviour, such as over-trust or "rubber-stamping". To prevent this, modern workflows empower reviewers through transparency:

- **Evidence-Based Review:** Workflows provide transparency by displaying evidence, rationale, policy basis, data lineage, and escalation options. This visibility empowers reviewers to effectively challenge recommendations without becoming overwhelmed.
- **Explainability and Traceability:** To support effective intervention, banks have implemented multi-layered control frameworks that embed source citations within generated responses, explicitly linking outputs to specific documents, database records, and policy references. For product recommendations, the system provides human-readable explanations derived exclusively from verified data.
- **Audit Trails:** The system captures and stores a comprehensive log of every interaction between users and agents. These audit trails are readily accessible to compliance, risk, and audit teams, facilitating thorough investigation and review. This ensures that every intervention, escalation, or approval is fully traceable and subject to independent challenge.

4. Task-level Oversight

In complex processes such as transaction monitoring, a bank uses specialised A.I. agents

for discrete tasks, including evidence collection, alert analysis, and narrative drafting. By implementing step-by-step governance with continuous evidence recording and A.I. checks at the task-level, organisations ensure rigorous validation. This method highlights a key principle in A.I. agent design, where the procedural validity of the decision path is just as important as the final result.

4. Addressing Key Challenges and Risks

4.1 Ongoing Risks of GenA.I.

4.1.1 Hallucination and Inaccuracy

R T A primary challenge in the adoption of GenA.I. is hallucination, where a model produces plausible-sounding but inaccurate, misleading or fabricated information. Such outputs can undermine credibility, create operational risk and compromise the reliability of solutions built on GenA.I..

To mitigate this risk, participating banks adopted a combination of controls across prompting, knowledge retrieval, validation and human oversight. No individual control can eliminate hallucination. Rather, reliability depends on a multi-layered approach proportionate to the use case, potential customer impact and materiality of decisions.

Advanced Prompt Engineering

While not a complete solution, sophisticated prompt engineering serves as the first and an agile line of defence against model hallucination. By strategically structuring instructions, banks can significantly constrain a model's generative

scope, forcing its outputs to be more factual, relevant, and verifiable.

The following advanced prompt engineering techniques were effectively employed by the participating banks to reduce factual inaccuracies:

Self-Verification Prompts

This technique instructs the model to check whether its proposed answer is supported by the provided information before returning it.

- **Prompt:** “Answer the client’s question using only the provided factsheet. Before finalising your response, confirm that each material statement is supported by the factsheet. List the supporting sentences or sections.”
- **Control:** The system may return an “insufficient information” response where no supporting information is available.

Grounded Reasoning and Verifiable Calculation Traces

For calculations or rule-based decisions, this method requires the model to provide a concise, auditable record of the applicable rules, input values, calculation steps and source references.

- **Prompt:** “Using only the approved account data and credit-policy rules provided, calculate the available credit. Return the applicable rule identifiers, input values, calculation result and source references. If any required input or rule is unavailable, return INSUFFICIENT_EVIDENCE.”
- **Control:** Where feasible, calculations could be independently validated through

deterministic application logic or a rules engine.

Structured Output Mandates

Requiring the model to output in a strict, machine-parsable format like JSON or XML with predefined fields reduces discursive and unverified text. (e.g. final_answer, confidence_score, source_references).

- **Prompt:** “Analyse the customer’s transaction description and classify its merchant category. Return the result in the required JSON format. If confidence is not high, return a null category and explain the missing or conflicting evidence. { *"category": "<category_or_null>,"* *"confidence": "high/medium/low",* *"evidence_references": ["<source_reference>"],* *"rationale": "<concise explanation>"}*”
- **Control:** Outputs should be validated against a schema, permitted values and applicable business rules. Outputs that fail validation should be rejected or routed for review.

Contextual Bounding and Explicit Prohibition

This involves clearly demarcating the source of truth and explicitly forbidding the model from using its internal knowledge, thereby strictly limiting its response domain.

- **Prompt:** “You are a support bot for Bank A. You may use only the product information provided below. If the answer is not contained in the provided text, respond: ‘I am sorry, I cannot find that information in my current resources. Please contact customer service at 852-XXXX-XXXX for assistance.’ Do not

speculate or use knowledge outside the provided text.”

- **Control:** This establishes a clear instructional boundary and reduces the risk of the model combining source content with pre-trained knowledge to generate unsupported answers.

Limitations

In practice, the participating banks found that combining these techniques yielded better results than applying any single technique in isolation. For example, contextual bounding and source-linked, verifiable calculation traces

can be combined with structured outputs and defined abstention conditions. However, prompt engineering is a mitigation rather than a guarantee of factual accuracy. For high-stakes applications, participating banks recognised the need for a multi-layered approach incorporating RAG from governed and current sources, output and calculation validation, independent evidence checks where appropriate, and human review or escalation procedures. Fine-tuning may further improve task adherence, domain language and output consistency, but should not be relied upon as the primary mechanism for factual grounding or maintaining current information.

1st cohort

2nd cohort

Examples of Prompt Engineering in GenA.I. Models R T

Prompt engineering is the practice of designing instructions, context and output requirements that guide GenA.I. models towards intended behaviours. It can support adherence to operational rules, improve consistency and reduce unsupported outputs.

1. Embedding Operational Guidelines into Prompts

Bank A employed prompt engineering to support operational compliance in its website chatbot. The prompt included instructions on using approved knowledge sources, mapping services, defining out-of-scope conditions, applying grounding requirements, and maintaining an appropriate tone and format.

The following guidelines were explicitly provided:

- Require retrieval from an approved datastore before answering knowledge-based queries.
- Use an approved service and product catalogue to identify relevant offerings.
- Define precise conditions under which a query should be treated as out of scope.
- Specify the required tone, language and response format.
- Require source references for material factual statements.
- Require the model to abstain or escalate where approved sources do not support an answer.

2. Using Explicit, Prohibitions and Escalation Requirements

Small changes in instruction wording can materially affect a GenA.I. model's adherence to

operational requirements. Participating banks observed that explicit, structured and mandatory wording was generally followed more reliably than broad or permissive instructions.

Instruction Hierarchy

Where instructions overlap or conflict, the model generally applies them in the following order:

- Prohibitions on unsupported information and requirements to use approved sources.
- Scope restrictions and mandatory escalation rules.
- Task-specific analytical instructions and output requirements.

If the available information is insufficient, instructions conflict, or the required task cannot be completed within the approved source boundary, the model should abstain or indicate that review is required.

Mandatory Task Enforcement

Bank B standardised stronger prompt patterns across its agent prompts and supplemented them with workflow-level controls. For example, an agent was restricted to handling questions related to approved customer news summaries, with other requests routed to a defined fallback response.

Example: Article-date relevance check

Critical Rule — Mandatory Date Validation

For each news article, identify:

- The publication date.
- Any date stated in the article as the date of the reported event or development.

Compare the dates and assess whether the article remains relevant to the requested reporting period. Do not assume that the publication date and event date must be identical.

If the event date is unavailable, ambiguous, materially earlier than the publication date, or inconsistent with the requested reporting period, return REVIEW_REQUIRED. Do not infer or invent a date.

Return the result in the required format: publication date, event date, relevance assessment and reason.

This design makes the required task, decision criteria, expected output and escalation path explicit.

Example: Scope restriction

Critical Rule — Scope Restriction

Answer only questions relating to approved customer news summaries. If the request is outside this scope, respond: “I can assist only with approved customer news summaries. Please use the appropriate channel for other requests.” Do not answer the out-of-scope question.

Explicit Prohibitions and Abstention

Bank B also found that explicit prohibitions may reduce unsupported generation, particularly where

source information is limited or unavailable.

Example: Company brief with limited public information

Critical Rule — No Unsupported Information

You may use only the approved sources provided. Do not invent, estimate or infer any fact, name, number or detail.

Prepare a company brief for XXX Logistics, including its 2025 revenue, current chief executive officer and number of warehouses operated in Asia.

For every material statement, provide a source reference. If an item cannot be verified from the approved sources, state INFORMATION NOT AVAILABLE IN APPROVED SOURCES. Do not provide an estimate or substitute information.

This approach defines the permitted source boundary, prohibited behaviours, evidence requirement and required fallback response.

4. Providing descriptive and detailed instructions

Bank C focused on reducing hallucination in their LLMs for detecting phishing emails. They crafted a detailed prompt instructing the model to analyse specific email features, such as suspicious URLs and urgent language. This approach guided the model towards more accurate classifications.

The example of prompt is provided below:

Instruction

You are a helpful assistant trained to classify emails as either "Phishing" or "Legitimate". A phishing email typically has certain characteristics, such as suspicious links, urgent requests for personal information, poor grammar, and mismatched email addresses. Given the email content, your task is to determine whether it is a phishing attempt or a legitimate communication.

Analyse the email content for the following features:

Presence of suspicious or mismatched URLs.

Requests for personal or financial information.

Urgent language or threats of account suspension.

Poor grammar or spelling errors.

Inconsistencies in sender's email address and domain.

Consider the context and tone of the email. Phishing emails often create a sense of urgency or fear.

Use the identified features to classify the email as either "Phishing" or "Legitimate".

5. Defining Input and Output Schemas

Schema-based prompting refers to defining a structured data format in the prompts that the inputs or outputs should adhere to, including required fields, acceptable value ranges and preferred format, which can enhance consistency and reliability in the outputs. Here are some examples:

Bank D adopted **canonical representation** for categorical and free-text variables, which involves transforming the text into numerical or vectorised forms that can be read by A.I. models. For

example, the transaction types of “Wire Transfer” and “Money Transfer” were standardised under the same term “Wire Transfer” to ensure uniformity in interpretation and reporting.

Bank D included clear guidelines of the response format in the prompt instructions, requiring the model to **quote the title of the source document** at the beginning of its responses. Additionally, a reference source button is provided for users, allowing them to easily view the exact paragraph of the source from which the answer was extracted.

Source-document titles are supplemented by a stable document identifier, version or effective date, and a precise source location. This enables users and reviewers to verify that a response is supported by an authoritative and current source.

Example output schema:

```
{
  "answer": "<concise_answer_or_INSUFFICIENT_EVIDENCE>",
  "confidence": "high/medium/low",
  "source_references": [
    {
      "document_id": "<approved_identifier>",
      "document_title": "<title>",
      "version_or_effective_date": "<value>",
      "source_location": "<section_or_paragraph>"
    }
  ],
  "escalation_required": false
}
```

Bank E used a **standardised training data format** with an instruction for input and output to embed desired behaviours into the model. This approach ensured consistent and reliable responses, providing a solid foundation for customer interactions. The example of prompt is provided below:

Instruction

You are a professional and helpful customer service assistant. Your task is to provide accurate, concise, and neutral information based exclusively on the knowledge you have been trained on. Do not speculate or offer financial advice. Respond in the language of the user's query.

Input

What are the benefits of the ABC Platinum credit card?

Output

The ABC Platinum Card offers several benefits, including cash rebates on eligible spending and exclusive merchant discounts. For a detailed list, you may refer to our official product page.

Inference Parameter Tuning

A critical technical approach to be combined with other techniques in reducing hallucination involves the precise calibration of inference-time parameters, which directly control the model's sampling behaviour during text generation. Strategic configuration of these parameters constrains the model's creativity, forcing it to prioritise high-probability, factual outputs aligned with its training and provided context.

Key Parameters

Temperature: This parameter influences the model's output by adjusting its level of randomness. As temperature increases, the model's responses become more creative and diverse. Conversely, a lower temperature sharpens the probability distribution, making the model favour the most likely tokens. This significantly reduces the chance of selecting irrelevant or fabricated tokens. However, when the temperature is set to 0, the model becomes highly deterministic, resulting in the consistent selection of the most probable word at each step.

Top-p and top-k: These two parameters are used

to control the output by limiting the number of possible next words. Top-k restricts the model to consider only a fixed number (k) of the most likely next words, while top-p considers all words with a cumulative probability up to certain threshold (p). Lower top-p or top-k values reduce the variety of generated text by focusing on the model's top predictions, leading to more focused and less diverse outputs.

Repetition Penalty: This parameter can be increased to discourage the model from repeating phrases or ideas, which can sometimes be a symptom of a failing or "stuck" generation loop that may precede hallucination. Tuning this helps maintain coherent and novel output within the constrained framework set by temperature and top-p/k.

Finetuning Tactics

Model finetuning in general is able to make a model more relevant by incorporating the knowledge from the training data into the model. A bank revealed in the finetuning process that certain banking-specific details in the training datasets, such as product and service names, can negatively influence the model, resulting in unexpected accuracy issues. A potential reason is that some of the conventions used in those particular data fields in the training dataset may not be linguistically intuitive to the base model. The bank has adopted a placeholder-tagging technique to mitigate this issue. The placeholder-tagged outputs allow the bank to integrate information from real-time database to avoid the need for the model to handle certain names and values that are more prone to confusion, thereby ascertaining accuracy.

Hybrid Fine-tuning Approach

A hybrid fine-tuning approach refers to combining

Insights:

To mitigate potential hallucination risks, a participating bank configured the GenA.I. model to operate with minimal "creativity" by controlling model inferencing parameters such as lower temperature (e.g. 0.1 - 0.3), top-p (e.g. 0.4 - 0.6) and top-k (e.g. 10 - 30) to reduce randomness and the likelihood of generating inaccurate or fabricated responses, especially when providing real-time responses to product inquiries from customers.

model fine-tuning with other techniques such as RAG or data augmentation. This approach has shown promise in reducing hallucination and improving the overall performance of the model.

Fine-tuning offers strong domain-specific performance with quick responses. However, it relies on a static snapshot of knowledge, and may lead to outdated responses or hallucinations where the model confabulates answers to queries beyond its training data. On the other hand, RAG provides the advantage of dynamically updating knowledge, effectively grounding responses and reducing fabrications. However, without fine-tuning, the base model may lack the specialised domain understanding to optimally interpret retrieved information, potentially leading to incoherent responses.

Combining both approaches (e.g. a fine-tuned model serving as the primary reasoning engine, augmented by a RAG system) allows for the integration of effectively addressing dynamic information requirements and enhancing conversational coherence.

4.1.2 Bias and Fairness

A T When A.I. algorithms or training data contain inherent biases, the application may generate unfair or discriminatory outputs when processing customers' input related to protected characteristics, such as age, race, or gender. In particular, word embedding in natural language processing can perpetuate biases.

Neutrality is treated as a core component of the validation framework. Banks in the second cohort tested for scorer bias by analysing whether scores followed the answer's quality rather than the scorer's identity. Key checks included controlling for positional bias by swapping item order, mitigating length bias by

comparing short versus long answers, and conducting rigorous human review. Models that failed to align with human reviewers, despite being internally consistent, were flagged as biased and excluded from the final evaluation.

Participating banks in the first and second cohorts implemented below mitigation strategies to minimise biases and maintain fairness:

Pre-processing with targeted word filtering: A pre-processing layer can be implemented to preprocess user prompts, detecting and filtering out language that is likely to elicit biased or stereotypical responses from the LLM, including a predefined list of words, phrases, and concepts.

Structured prompting with a neutral persona: By setting the conversation with a prompt that explicitly instructs the LLM to remain neutral, objective, and unbiased, e.g. with explicit directives such as: "You are a customer advisor who must provide objective, equitable advice. Do not make assumptions or recommendations based on a user's perceived age, gender, or background. Focus solely on the stated queries." This constrains the model's responses, mitigating inherent biases and promoting a consistent, impartial tone throughout the interaction.

Balanced seed dataset curation: A pre-processing layer is implemented to curate a highly representative seed dataset, ensuring a balanced distribution across various critical risk tiers and demographic segments. This manual selection process prevents the over-representation of certain historical transaction types in the training data, thereby mitigating historical and representational bias by establishing a neutral, equitable foundation for model learning.

Controlled synthetic augmentation: Controlled

synthetic generation is employed to safely introduce complex edge cases and underrepresented scenarios that are scarce in real-world data. By algorithmically expanding the dataset with these diverse

examples, the model addresses sampling bias and enhances its ability to identify evolving patterns, ensuring robust performance on novel or atypical transactions without relying on static historical norms.

1st cohort

Case study – Bias Management in Chatbot A T

Bias in GenA.I. models can arise from the selection and processing of training data and algorithms, leading to unfair outputs. Addressing bias requires continuous effort throughout the model's lifecycle. This involves curating diverse datasets, conducting regular audits, using fairness-aware algorithms, engaging users for feedback, maintaining transparency, fostering collaboration, and adhering to ethical guidelines. A participating bank in the GenA.I. Sandbox integrated bias assessment into their testing process, demonstrating practical bias assessment.

1. Data-Level: Tackling the Data Source

The bank identified sensitive attributes including gender, race, age, nationality, and income that could lead to bias. To ensure diverse and representative datasets, they proportionally represented all relevant groups across these attributes. Additionally, the bank defined "privileged", which historically received more favourable treatment, and "non-privileged" groups for each attribute to identify potential biases.

2. Model-Level: Building Fairness In

Before testing, the bank created two types of questions: (1) neutral questions and (2) paired questions, each consisting of two versions that are identical except for one demographic attribute. These questions were then used to query the chatbot one by one, and the responses were recorded. Before conducting the evaluation, any demographic attributes in the chatbot's responses were replaced with placeholders to ensure the evaluation was free from demographic biases.

3. Bias Assessment

After the model generated results, the bank conducted assessments for more than 500 results across below categories:

- (1) Traditional NLP similarity analysis: This analysis assessed statistically significant differences across demographics by calculating ROUGE-L and BLEU scores for masked answers. The scoring system ranged from 0 to 5, with higher scores indicating greater similarity.
- (2) Embedding similarity: This involved computing sentence embeddings and calculating cosine similarity.
- (3) Clarity and conciseness testing: This test evaluated whether the answers provided clear and

concise responses using LLMs.

(4) Sentiment/ Tone: classify each answer's sentiment or tone using an LLM.

4. Bias Mitigation

The bank outlined two methods to be implemented to mitigate bias: (1) Prompt engineering with a multilingual persona, which guides the model's behaviour by crafting a system prompt that defines an unbiased, objective character for the A.I. to embody; (2) preprocessing with target word filtering, which involves programmatically scanning and modifying user input queries before they are sent to the Chatbot model.

Addressing bias is an iterative process. As societal norms evolve and new data is collected, models require re-assessment and updates to maintain standards of fairness and responsibility.

4.1.3 Security and Privacy

4.1.3.1 Security

R LLMs and other GenA.I. systems present security risks requiring specialised controls. These include prompt injection, sensitive-information disclosure, improper output handling and misinformation. They

may also include indirect prompt injection, unsafe tool use and third-party supply-chain risks, particularly where GenA.I. systems retrieve external content or connect to internal systems.

Participating banks from both cohorts used the mitigation techniques below to address these risks.

Risks	Mitigation Strategies
Prompt injection — Malicious actors craft inputs designed to override system instructions, potentially manipulating the model into performing unauthorised actions. This may also arise indirectly from retrieved documents, webpages, emails or attachments containing malicious instructions.	- Implement monitoring and detection systems for prompt-injection attacks. - Reject queries that are out of scope. - Implement strict separation between system instructions, retrieved context and user input to prevent instruction override. - Deploy secondary classifier models to detect and block potentially malicious prompts before processing by the primary LLM. - Establish dynamic rules that evaluate prompt intent and context rather than relying solely on keyword blocking. - Treat retrieved content as untrusted data, rather than as instructions to be followed. - Use approved retrieval sources, provenance controls and content-sanitisation measures.

Risks	Mitigation Strategies
Sensitive-information disclosure — Models may inadvertently reveal proprietary algorithms, customer data or internal processes through direct extraction, retrieval or inference attacks.	- Implement tokenisation, masking or de-identification to pre-process and sanitise sensitive information. - Restrict model access to sensitive data according to the principle of least privilege. - Implement strict access controls and tenant segregation for vector databases and knowledge stores. - Encrypt sensitive data in transit and at rest. - Incorporate statistical noise during training, where appropriate, to reduce the risk of memorising individual data points while maintaining aggregate model utility. - Deploy automated scanners to detect and redact potentially sensitive information, such as account numbers and personal identifiers, before outputs are delivered to users. - Conduct testing for unauthorised retrieval, data leakage and cross-user or cross-tenant access.
Improper output handling and unsafe tool use — Model outputs may be malformed, malicious or unauthorised. Where the model connects to backend functions or tools, this may result in inappropriate data retrieval, customer communications, transactions or system changes.	- Apply proper input validation to model outputs before passing them to backend functions. - Implement robust logging and monitoring systems to detect unusual patterns in LLM outputs. - Enforce strict output validation against predefined schemas, rejecting responses that deviate from expected formats. - Validate proposed tool calls, parameters, identities, permissions and business rules in the backend before execution. - Restrict tool permissions to the minimum necessary, and require user confirmation or human approval for material or irreversible actions.
Misinformation and information-integrity failures — Models may generate unsupported, outdated or misleading information, including inaccurate citations or conclusions not supported by retrieved sources.	- Specify the exact conditions that determine when a query falls outside scope. - Maintain detailed audit trails linking model responses to their source materials, facilitating rapid verification and accountability. - Use approved, current and version-controlled source repositories. - Require source references for material factual statements and validate whether the cited evidence supports the relevant claim. - Define abstention and escalation procedures where information is unavailable, conflicting or outside scope. - Do not rely solely on model self-reported confidence scores. Use source quality, retrieval quality, rule-based checks and human review to determine whether escalation is required.

Risks	Mitigation Strategies
Model, data and supplier supply-chain risk — Risks may arise from third-party models, model updates, fine-tuning data, open-source components, plug-ins, embedding models and external knowledge sources.	- Maintain an inventory of models, dependencies, data sources and service providers. - Assess material model, system and knowledge-base changes through regression, security and safety testing before deployment. - Control and review fine-tuning datasets and knowledge-base updates. - Maintain rollback, fallback and incident-response arrangements.

4.1.3.2. Privacy

T Data privacy is another crucial area in the development of GenA.I. models to ensure protection of sensitive information, prevention of bias and discrimination, compliance with legal requirements, as well as maintenance of user trust. To address privacy concerns, the following mitigation methods were implemented by the participating banks:

Intelligent Data Masking: Participating banks implemented automated data masking to anonymise Personally Identifiable Information (PII) such as the customer's name, Hong Kong Identity Card (HKID) number, account number, phone number, etc. To streamline the data masking process, a bank used Named Entity Recognition (NER), which is a natural language processing technique that can automatically identify and categorise key elements and other important concepts. This approach enabled efficient processing of large text corpora while maintaining privacy.

Synthetic Data Generation: Several banks replaced sensitive information with randomly generated or synthetic data. Additionally, differential privacy technique was applied to the synthetic data by introducing controlled randomness or noise into the dataset to further protect individual privacy.

Data Minimisation: During preparation stage, participating banks adhered to strict data minimisation principles, collecting only essential data fields required for specific model objective.

Temporal and Access Control: Banks implemented role-based access controls to knowledgebase as well as limited retention periods to provide assurance that models and personnel only accessed sensitive data when necessary, following the principle of least privilege.

Monitoring: Participating banks also implemented logging mechanisms with detailed audit trails of all data access and processing activities, enabling comprehensive monitoring.

4.1.4 Explainability and Transparency

T Participating banks implemented practical measures to make GenA.I.-supported outputs traceable, reviewable and suitable for control functions.

These measures focused on three elements:

- **Evidence:** the source data, documents or policies supporting an output.
- **Decision rationale:** the relevant criteria, calculations and factors applied.
- **Execution trace:** the sequence of agent

actions, retrievals, validations and outputs leading to the final response.

Together, these measures enable users to verify material outputs, investigate exceptions and identify where escalation or human review is required.

This approach enables users to review and validate the factors, calculations and source evidence supporting a preliminary assessment, and to escalate cases where information is missing, inconsistent or outside defined policy criteria.

2nd cohort

Case Study: Evidence-Based Decision Rationale for Property Mortgage Loan (PML)

Credit Assessment Assistant T

Bank A fine-tuned a GenA.I. model to support the Property Mortgage Loan (PML) Credit Assessment Assistant for assessing property mortgage loan applications. The use case was designed to produce a concise and auditable decision rationale.

The rationale includes the relevant eligibility criteria, input data, calculations, source references and any missing or adverse information.

Illustrative output:

Assessment item	Rationale and evidence
Income requirement	The applicant's verified income meets the minimum policy requirement. Source: Approved income-verification record and applicable credit policy.
Debt-service ratio	The debt-service ratio is within the applicable policy threshold. Source: Approved calculation rule and verified financial inputs.
Transaction-history review	No relevant red flags were identified in the available transaction-history data. Source: Transaction-monitoring record dated [date].

1st cohort

Source Citation in RAG-Powered Chatbots T

For customer-service chatbots answering product questions, participating banks used RAG. The systems were designed to provide source references for material factual statements, enabling customers and staff to review the authoritative information supporting an answer.

Illustrative example:

User: What is the early-withdrawal penalty for a 12-month fixed deposit?

Chatbot: According to Section 4.2 of the Fixed Deposit Terms and Conditions (version 2024-11), early withdrawal of a 12-month fixed deposit results in forfeiture of 60 days of interest.

[Source: Document ID BA-TC-2024, version 2024-11, paragraph 12.]

This supports verifiability and allows users to assess whether an answer is grounded in approved documentation.

Role-Based Multi-Agent Workflows

Participating banks explored multi-agent designs in which complex tasks were divided among agents with defined responsibilities, such as information retrieval, analysis, validation and final-response generation.

This design supports task-level transparency by enabling users and control functions to trace:

- Which agent performed each task.
- The inputs, outputs and tool calls associated with each step.
- The source documents or data retrieved.
- The validation checks performed.
- The point at which an error, exception or unsupported conclusion arose.

Each agent's permissions and permitted tools should be limited to its defined role. Hand-offs between agents should be structured, logged and validated.

4.2 Emerging Risks in More Autonomous A.I. Systems

A R A.I. agents expose the system to more novel risk factors such as incorrect routing, duplicated processing, increased workflow latency, and inconsistent hand-offs between

components. The efficacy of an A.I. agent framework extends beyond the quality of its final outputs. It may also be measured by routing accuracy, workflow reliability, end-to-end traceability, and the overall effectiveness of its governance controls.

Banks are addressing these risks through targeted mitigation strategies across various critical operational dimensions. Crucially, the strictness of these controls is not uniform. Instead, agent actions and the corresponding safeguards are calibrated to align with each bank's specific **risk appetite**.

Autonomy: Routing and Orchestration Integrity

Autonomy describes the extent to which an agent can self-direct its decision-making and action execution without human involvement. High autonomy increases efficiency but raises the risk of "misrouting", where an agent incorrectly identifies its role or fails to trigger necessary action flows. To mitigate this, banks refine their routing rules and implement rigorous intent checks before delegating tasks. This ensures that actions are triggered only when confidence thresholds are met. Banks with a lower risk appetite can enforce stricter confidence thresholds and require human confirmation for multi-intent actions.

Explainability and Traceability Measures

Control measure	Implementation description
Source traceability and evidence	Material factual claims include citations to specific documents, database records or policy references. Users can view the relevant source material and independently verify the claim.
Controlled rationale generation	The system generates concise, human-readable rationales based only on verified input data, approved knowledge sources and defined criteria. The rationale identifies key supporting factors, limitations, exceptions and missing information.
Execution and audit trail	The system records user prompts, agent invocations, timestamps, inputs, outputs, tool calls, retrieved documents, validation and compliance-check results, and the final delivered response. Authorised compliance, risk and audit personnel can use these records for review and investigation.
Access and retention controls	Explainability records and audit logs are access-controlled, protected against unauthorised alteration, retained according to defined periods, and reviewed to avoid unnecessary retention of personal or confidential.

Intent Alignment: Retrieval & Grounding

Intent Alignment is defined by the strict adherence of an agent's output to established business logic, factual accuracy, and regulatory requirements. Misalignment occurs when agents hallucinate details or provide ungrounded claims, deviating from the intended business goal. Banks address this by enhancing metadata structures and enforcing strict retrieval validation protocols. A common mitigation strategy is the implementation of a "no source, no claim" policy, where agents are prohibited from generating assertions without verifiable citations.

Additionally, systems are configured to explicitly display sources, allowing human reviewers to validate the grounding of the agent's output. Based on the risk level of specific tasks, banks may mandate the outputs be grounded in pre-approved, high-confidence datasets, rejecting any response that lacks explicit source links.

Cascading Failures: Context and Customer Handling

Cascading Failures arise when a breakdown in one component, like context loss, ripples through the workflow, leading to erroneous results or total failure in subsequent agents. In multi-agent systems, weak session control can lead to a breakdown in the entire interaction chain. To counter these risks, banks strengthen memory management mechanisms and enhance language control protocols. Furthermore, they implement specialised customer search logic to ensure that agent responses are personalised and consistent with historical interactions and specific user profiles. For tasks prioritising data privacy and accuracy, context handling may be restricted to anonymised or aggregated data, limiting personalisation.

Control: Output and Compliance

Control encompasses the final oversight mechanisms, including filtering, sanitisation, and audit trails. This dimension addresses the exposure of sensitive reasoning traces or field-mapping errors that could lead to regulatory breaches. Banks mitigate these by expanding compliance testing frameworks and sanitising final outputs before they reach the user. This includes validating form fields to ensure data integrity and applying strict filters to prevent the leakage of internal reasoning processes or unverified data. Low-risk-appetite banks may implement exhaustive, rule-based filtering that blocks any output containing ambiguous language or unverified claims.

Ultimately, the implementation of these mitigation strategies is not one-size-fits-all. The specific controls, thresholds, and human oversight arrangements should be calibrated to align with each bank's **risk appetite**. For banks with a lower risk tolerance for a specific function, this may mean stricter routing rules, more frequent human interventions, and conservative data sourcing. Conversely, banks with a higher tolerance may rely more heavily on automated validation layers and broader data access to drive efficiency. Therefore, effective governance requires a dynamic approach where agent actions and safeguards are continuously adjusted to balance innovation with the specific risk profile of the organisation.

4.3 A.I. vs. A.I.: An Evolving Security Dynamic

A R As A.I. is increasingly used to process information and support banking activities, it may become difficult to rely solely on manual review and monitoring. The speed, scale and evolving nature of A.I.-enabled applications call for

controls that are scalable, continuous and, where appropriate, embedded in relevant workflows.

To keep pace with these developments, the GenA.I. Sandbox is exploring an **“A.I. vs. A.I.”** approach. This refers to the use of one A.I. system to test, evaluate, challenge, monitor or validate the outputs and behaviour of another A.I. system. Such tools may help banks assess the quality, consistency, robustness and compliance of A.I. outputs, and identify potential risks at an earlier stage. As use cases evolve beyond generating responses to performing information retrieval, analysis and workflow support, validation should extend beyond output quality. Banks may also consider whether an A.I. system operates within its intended scope, follows applicable policies and instructions, handles sensitive information appropriately, and produces outputs that are sufficiently accurate, reliable and explainable for the relevant use case.

A.I.-enabled validation may reinforce controls across the three lines of defence by supporting the timely detection, escalation and investigation of anomalies or control failures. It should complement, rather than replace, appropriate human oversight, accountability and independent review.

Examples of A.I. vs. A.I. validation approaches include:

- **LLM-as-a-Judge:** A separately configured or specialised large language model assesses the outputs of a target model against predefined evaluation criteria, such as accuracy, relevance, completeness and adherence to policies or procedural requirements.
- **Adversarial testing or red teaming:** An A.I. tool generates challenging prompts, edge cases, misleading inputs or simulated attack scenarios

to test whether a target A.I. system may produce inaccurate, harmful or non-compliant outputs, or fail to follow established controls.

- **Automated guardrails:** Validator models, classifiers or rule-based controls screen user inputs and model outputs for specified risks, such as harmful content, possible confidential-data leakage, policy breaches or unsupported claims. These controls may operate before an input is processed, before an output is presented to a user, or at other defined control points. When developing more comprehensive agent

harness, domain-specific risk management policies and procedures may be integrated to enable more dynamic guardrails that are aligned to the bank's risk management framework.

- **Cross-model comparison:** Different models independently assess the same output, question or dataset. Material differences in their assessments may be flagged for further investigation. Agreement between models may provide an additional validation signal, but should not be treated as conclusive evidence of correctness.

2nd cohort

Case Study: Implementing A.I. vs. A.I. Validation **A** **R**

To mitigate model-specific errors and enhance output reliability, a bank has adopted an **A.I. vs. A.I. validation framework** during its testing and evaluation phases. This approach executes identical document processing tasks across **multiple independent A.I. models**, designating one as the primary processor while others serve as independent validators.

By cross-referencing these outputs, the Bank establishes a dynamic confidence score:

- **High Confidence:** Achieved when multiple models produce substantially similar results, confirming accuracy and completeness.
- **Low Confidence/Flagged:** Triggered when significant discrepancies arise, prompting further review.

This cross-model verification acts as a robust quality assurance control, effectively reducing model bias, minimising omissions, and supporting more reliable, data-driven decision-making.

2nd cohort

Case study: Dual-Chain Quality Evaluation **A** **R**

A participating bank uses two separate AI review pipelines to reduce the risk of unsafe automated approvals.

1. More models in one pipeline do not necessarily improve safety

The project tested review pipelines using between two and six A.I. “judge” models in sequence. The results showed that using more than two models did not consistently improve performance. Similar models often struggle with the same difficult cases and may share similar blind spots.

More opinions therefore do not necessarily produce a safer decision if those opinions are not

sufficiently independent.

2. A single pipeline may still pose unacceptable approval risk

A single pipeline may have an error rate of around 10% among cases it automatically approves. That is, approximately 10 in every 100 approved cases may be inappropriate approvals. For high-risk transactions, documents or customer cases, even one incorrect approval may be unacceptable. Making the pipeline stricter can reduce this risk, but may also escalate many more cases to human reviewers and reduce the value of automation.

3. Solution: require two independent pipelines to agree

Rather than relying on one stricter pipeline, the system runs two independent A.I. review chains in parallel.

Case → A.I. Chain A $\neg$

→ Automate only if both pass / agree

Case → A.I. Chain B $\neg$

Otherwise → Escalate to a human reviewer

A case is automatically approved only if both chains independently approve it. If either chain identifies a concern, is uncertain, or disagrees with the other chain, the case is escalated to a human reviewer.

4. How the control reduces false approvals

The relevant measure is the chance that a genuinely bad case is wrongly approved:

$$P(\text{approved} \mid \text{bad case})$$

If Chain A wrongly approves a bad case with probability α_A , and Chain B with probability α_B , then, if their errors are conditionally independent:

$$P(\text{both approve} \mid \text{bad case}) = \alpha_A \times \alpha_B$$

For example, if each chain wrongly approves 10% of bad cases, the estimated probability that a bad case passes both chains is:

$$10\% \times 10\% = 1\%$$

This is a reduction in the **false-approval rate for bad cases**. It is not automatically the error rate among all cases jointly approved. That latter measure also depends on the share of bad cases in the population and how often good cases are approved.

5. Validation is essential

The bank validated the two-chain design using representative banking cases and report both:

- the false-approval rate: $P(\text{both approve} \mid \text{bad case})$; and
- the error rate among jointly approved cases: $P(\text{bad case} \mid \text{both approve})$

The expected benefit depends on the two pipelines making sufficiently different errors. This should be tested rather than assumed, including through different models, prompts, decision logic, evaluation data and known failure scenarios.

4.4 Adaptive Guardrails

R T Adaptive guardrails represent an evolution in A.I. risk management, moving beyond rigid, zero-tolerance blocking mechanisms towards dynamic, context-aware supervision. Rather than treating all potential risks as binary threats requiring immediate interruption, they process real-time risk signals to distinguish legitimate use from malicious misuse. When implemented in the more advanced form of agent harness, this enables organisations to support neutral or beneficial A.I. interactions while maintaining appropriate safeguards that aligns to the organisations' risk management policies and procedures.

For example, a bank has applied this approach to calls handled through its contact centre. By assessing confidence levels and contextual factors, rather than relying on a simple on-or-off decision, adaptive guardrails can reduce the operational bottlenecks and user-experience disruption associated with static security models. This helps ensure that controls are proportionate to the level of risk.

The bank uses a tiered response protocol that aligns the level of intervention with transaction sensitivity. For conversational or public-information enquiries that do not involve

sensitive data, the system applies standard safety guardrails and control measures without interrupting the interaction, thereby preserving service continuity. For lower-risk enquiries involving personal account data, the system requires dynamic authentication and restricts responses to pre-approved content repositories. For critical scenarios, such as threats to asset security or high-value regulated transactions, it escalates cases to human agents and activates automated protective measures, such as account freezing. This structured approach supports seamless handling of legitimate customer interactions while enabling prompt intervention in security-critical events, balancing robust compliance with a positive user experience.

4.5 A Multi-Layered Defence Against Deepfake Threats

R T Participating banks observed that the increasing accessibility and quality of A.I.-generated media has reduced the effectiveness of static verification methods. Threat scenarios ranged from face-swapping or reenactment during selfie verification to manipulated identity documents and media-injection attacks designed to alter camera or video feeds before they reached verification systems.

A key insight was that deepfake detection models

were more effective when deployed as part of a multi-layered fraud-control framework, rather than as standalone safeguards. The controls explored included:

- Liveness detection and challenge-response mechanisms.
- Document-authenticity and tamper-detection checks.
- Cross-checking of identity-document and selfie images.
- Detection of duplicate or highly similar identity images across applications.
- Device-integrity, application-attestation and mobile-runtime protections.
- Detection of sideloaded or compromised applications.
- Transaction, behavioural and network-level fraud indicators.
- Manual review of uncertain, high-risk or anomalous cases.

Participating banks also used adversarial testing, including red-team and blue-team exercises, to evaluate detection and response capabilities against deepfake videos, manipulated identity documents, personal-data tampering, replay attacks and coordinated fraudulent account applications. Test results informed the refinement of detection models, fraud rules and response procedures.

In mobile applications, client-side controls such as Runtime Application Self-Protection (RASP), device-integrity checks and sideloaded-application detection were used to help identify media-injection attempts. Banks noted that these controls were most effective when paired

with server-side validation, as client-side protections may be bypassed on compromised devices.

Synthetic and adversarial datasets were used to develop and test deepfake-detection capabilities. The datasets included variation in capture devices, compression levels, lighting conditions, demographic groups and attack techniques. This helped identify performance disparities and robustness gaps, although participating banks recognised that synthetic data alone cannot eliminate bias.

A.I.-driven image clustering also provided a means of identifying suspicious patterns across large volumes of identity-verification images. By generating image embeddings and grouping visually similar images, banks could identify potential duplicate identities, coordinated fraud campaigns or repeated use of manipulated images that may not be apparent through manual review or one-to-one matching.

Banks treated image-clustering results as risk indicators for investigation rather than conclusive evidence of fraud. They also recognised that image embeddings and biometric data require appropriate data-minimisation, access-control, retention, security and human-review measures.

5. Business Outcome

The adoption of GenA.I. and A.I. agents in the second cohort marked a shift from relatively discrete applications towards more complex and business-critical operational workflows. These use cases involved integrating multiple data sources, implementing higher levels of judgment, addressing material risk and control considerations, and ensuring effective guardrails and human oversight. While human-in-the-loop review remained relevant for selected higher-risk activities, the second cohort's experience indicated that requiring review of every individual A.I.-generated output may not be scalable for high-volume or time-sensitive workflows. Banks therefore combined automation with risk-proportionate human oversight arrangements, including defined operating boundaries, exception-based escalation, monitoring and intervention mechanisms.

Testing results from the second cohort demonstrated measurable benefits, not only in terms of time and cost savings, but also in the control effectiveness, decision-making capabilities, operational resilience and customer experience (Figure 8).

Strengthened Risk Management, Anti-Fraud and Financial Crime Controls

Banks operate in an increasingly complex risk environment, where they need to protect their financial soundness, reputation and customers while meeting regulatory obligations. Effective controls against money laundering, terrorist financing, fraud, sanctions breaches and other illicit activity are essential to preventing misuse of the banking system and reducing the risk of

significant losses or disruption.

The second cohort explored GenA.I. and A.I. agent applications in high-impact anti-fraud, risk and control environments. These applications can enhance the speed, scale and quality of fraud detection and risk assessment, enabling earlier intervention, reducing manual effort and allowing specialists to focus on higher-risk and more complex cases.

- Reduced False Acceptance Rate from 10% to below 1% in testing of over 10,000 attack samples, with potential annual fraud-loss savings of millions of Hong Kong dollars.
- Reduced manual risk-review workload by around 60%, enabling staff to focus on higher-risk cases.
- Reduced customer AML risk review time by 66%, from around 60 to 20 minutes per case, while achieving 95% text-extraction accuracy.

Improved Operational Productivity and Decision-Making

Banks face growing volumes of customer, transaction, market and regulatory data. Improving productivity therefore requires more than automating routine tasks; it requires the ability to retrieve, synthesise and analyse information across complex, often unstructured, datasets in a controlled and auditable manner.

The second cohort focused on applications supporting knowledge-intensive and decision-critical processes, including financial analysis, market-risk enquiry and credit-proposal preparation. The results demonstrate that

GenA.I. and A.I. agents can shorten information-processing cycles and support employees in producing more timely, consistent and better-informed outputs, while retaining appropriate human review and accountability.

- Achieved up to 90% automation in financial data extraction, and 50%-60% automation in financial analysis.
- Achieved a 95% reduction in market risk data query time from hours to under two minutes by leveraging a risk analyst agent.
- Reduced drafting time of complete credit proposals by over 30% for simple cases and 60.0% for complex cases.

Enhanced Customer Experience

Customers increasingly expect banking services to be fast, personalised, seamless and accessible across channels. For example, a customer-facing chatbot requires a high standard of factual accuracy, relevance, suitability and

responsiveness, particularly in areas such as wealth management, where customer trust and risk management are essential.

The second cohort examined applications that combine service enhancement with appropriate controls. The findings suggest that GenA.I. and A.I. agents can help banks streamline customer onboarding journeys, provide more responsive support and deliver accurate and relevant information.

- Delivered faster end-to-end customer onboarding, while maintaining risk-control effectiveness.
- A wealth-management chatbot achieved over 99% in factual accuracy, relevance and completeness.
- The chatbot earned strong positive feedback on user friendliness, demonstrating high user acceptance and enabling more convenient, responsive client service.



Figure 8 – Business Outcomes of the GenA.I. Sandbox

6. Conclusion

The second cohort of the GenA.I. Sandbox has demonstrated the potential of **A.I. agents** to support more integrated banking workflows. Building on the first cohort's exploration of GenA.I. capabilities, participating banks tested how A.I. agents can combine approved data sources, tools, business rules and control mechanisms to support activities across customer service, relationship management, financial crime investigation, risk management and operations.

The trials indicate that the value of A.I. agents lies in combining fit-for-purpose models, defined workflows, deterministic controls, dynamic guardrails and evidence-based validation. This can improve the quality, consistency and traceability of A.I.-supported processes, while maintaining appropriate human oversight and accountability.

Practical Insights for A.I. Adoption

The second cohort identified several practical considerations for banks developing and deploying GenA.I. solutions and A.I. agents:

- **Clear control boundaries:** A.I. components should operate within boundaries that are anchored to the intent of the tasks. Such boundaries include defined responsibilities, approved data and tool access, structured inputs and outputs, and appropriate escalation arrangements. This supports traceability, testing and issue isolation.
- **Task-aligned model selection:** Different tasks may require different models. More capable models may be appropriate for complex reasoning, long-context analysis and multilingual content, while smaller or specialised models may support routine classification, extraction, routing and validation more cost-effectively.
- **Deterministic techniques for defined controls:** Rule engines, function calls, calculations, schema validation and predefined thresholds can complement LLM-based reasoning where outcomes need to be consistent, transparent and repeatable.
- **Evidence grounding and validation:** Source tagging, context isolation, retrieval controls, structured outputs and post-generation validation can help mitigate source mixing, unsupported claims and other output-quality risks.
- **Task-based solution design:** Complex processes may benefit from multiple specialised A.I. agents for retrieval, analysis, drafting and review. Simpler use cases may adopt the minimum necessary workflow to manage latency, cost and operational complexity.

Advancing A.I. vs. A.I. Controls

The second cohort also advanced the practical exploration of the “A.I. vs. A.I.” approach. Separate A.I. systems may be used to review, challenge and monitor primary A.I. systems for unsupported claims, missing caveats, inaccurate reasoning, prompt-injection risks and non-compliance with predefined requirements.

Where appropriate, banks may use validation models from different model families, architectures or providers to provide a more independent challenge and reduce common weaknesses affecting both the primary and validation models.

When adopting an “A.I. vs. A.I.” approach across the three lines of defence, banks may explore specific integration of their enterprise risk management frameworks into dynamic guardrails. This approach allows banks to more effectively translate its risk appetite into related A.I. workflows. These arrangements should complement, rather than replace, human accountability and established control functions.

Looking Ahead: Sandbox++

Looking ahead, the first cohort of the upgraded **GenA.I. Sandbox++**⁶ will focus on agentic A.I. applications that explore how A.I. can responsibly take on greater autonomy in end-to-end financial processes. This includes use cases relating to customer onboarding, payments, insurance claims and customer interactions.

GenA.I. Sandbox++ will also further explore **agent harnesses** and related orchestration arrangements that enable more comprehensive agentic A.I. systems to operate within defined

control boundaries. This may include the allocation of specialised roles among individual A.I. agents, controlled access to approved tools and data sources, structured workflow sequencing, logging and traceability, deterministic guardrails, and human escalation for exceptions or higher-risk actions.

Building on the “A.I. vs. A.I.” theme, GenA.I. Sandbox++ will further examine how A.I. can provide dynamic oversight of actions taken by other A.I. applications. This may include using separate and, where appropriate, heterogeneous models to dynamically challenge outputs, identify anomalies, assess adherence to predefined requirements and trigger timely intervention. These controls should complement human accountability and established governance arrangements.

⁶ First cohort of GenA.I. Sandbox++, 27 August 2026. <https://www.hkma.gov.hk/eng/news-and-media/press-releases/2026/08/20260827-3/>

7. Appendix

First cohort of GenA.I. Sandbox and examples of proposed use cases

Participating Banks	Technology Partners
• Bank of China (Hong Kong) Limited • China CITIC Bank International Limited • China Construction Bank (Asia) Corporation Limited • Citibank (Hong Kong) Limited • Dah Sing Bank, Limited • Hang Seng Bank, Limited • Hongkong and Shanghai Banking Corporation Limited • Livi Bank Limited • Societe Generale • Standard Chartered Bank (Hong Kong) Limited	• Aereve • Alibaba Cloud • Beijing Baidu Netcom Science Technology Co., Ltd. • Forms Synttron Information (HK) Limited

Theme	Example Use Cases
Risk Management	• A.I.-assisted financing approval • A.I.-powered AML Suspicious Transaction Reporting (STR) • Enhanced Know Your Customer (KYC) with unstructured data
Anti-Fraud Measures	• Intelligent assistant for fraud investigator • Protection against fraudulent account openings
Customer Experience	• A.I. advisor for financial knowledge and updates • Banking chatbot with A.I.-enhanced interactivity

Second cohort of GenA.I. Sandbox and examples of proposed use cases

Participating Banks	Technology Partners
• Ant Bank (Hong Kong) Limited • Bank of China (Hong Kong) Limited • Bank of Communications (Hong Kong) Limited • Bank of East Asia, Limited • China Construction Bank (Asia) Corporation Limited • Chiyu Banking Corporation Limited • CMB Wing Lung Bank Limited • Dah Sing Bank, Limited • Fubon Bank (Hong Kong) Limited • Hongkong and Shanghai Banking Corporation Limited • Industrial and Commercial Bank of China (Asia) Limited • Industrial Bank Co., Ltd. • Livi Bank Limited • Nanyang Commercial Bank, Limited • Natixis • Ping An Digital Bank (International) Limited • Public Bank (Hong Kong) Limited • Shanghai Commercial Bank Limited • Standard Chartered Bank (Hong Kong) Limited • Tai Sang Bank Limited	• AIFT • Alibaba Cloud • Ant Digital Technologies • China Mobile Hong Kong • CLPS Technology • D2 Intelligence • Dyna.AI • Ernst & Young • FORMS HK • Hong Kong University of Science and Technology • iMBrace • OneConnect Financial Technology • RealAI Technology • Votee AI

Theme	Example Use Cases
Risk Management	• Multi-agent A.I. for credit portfolio management • Intelligent trade finance facilitation • Market risk monitoring and early alert
Anti-Fraud Measures	• Fortified defence against fraudulent facial authentication • Cross-sector and multi-domain fraud detection
Customer Experience	• Investment insights powered by A.I. • Conversational customer service A.I. voicebot • Omni-channel customer relationship management • Inclusive banking with A.I.-aid accessibility tool

Fact Sheets / Supplementary Reports (First Cohort)

Participant	QR Code / Link	Participant	QR Code / Link
Bank of China (Hong Kong) Limited	 Link	China CITIC Bank International Limited	 Link
China Construction Bank (Asia) Corporation Limited	 Link	Citibank (Hong Kong) Limited	 Link
Dah Sing Bank, Limited	 Link	Hang Seng Bank, Limited	 Link
Hongkong and Shanghai Banking Corporation Limited	 Link	Livi Bank Limited	 Link
Standard Chartered Bank (Hong Kong) Limited	 Link		

Note: Any positions or statements made in the above factsheets / supplementary reports as prepared by Sandbox participants do not necessarily represent the views of the HKMA.

Fact Sheets / Supplementary Reports (Second Cohort)

Participant	Theme	QR Code / Link
Ant Bank (Hong Kong) Limited	Anti-Fraud Measures	 Link
Bank of China (Hong Kong) Limited	Customer Experience	 Link
Bank of Communications (Hong Kong) Limited	Anti-Fraud Measures	 Link
China Construction Bank (Asia) Corporation Limited	Customer Experience	 Link
Chiyu Banking Corporation Limited	Risk Management	 Link
CMB Wing Lung Bank Limited	Customer Experience	 Link

Participant	Theme	QR Code / Link
Dah Sing Bank, Limited	Risk Management Anti-Fraud Measures Customer Experience	 Link
Fubon Bank (Hong Kong) Limited	Customer Experience	 Link
Hongkong and Shanghai Banking Corporation Limited	Anti-Fraud Measures Customer Experience	 Link
Industrial and Commercial Bank of China (Asia) Limited	Risk Management	 Link
Industrial Bank Co., Ltd.	Anti-Fraud Measures	 Link
Livi Bank Limited	Anti-Fraud Measures	 Link

Participant	Theme	QR Code / Link
Nanyang Commercial Bank, Limited	Customer Experience	 Link
Natixis	Risk Management	 Link
Ping An Digital Bank (International) Limited	Anti-Fraud Measures	 Link
Public Bank (Hong Kong) Limited	Risk Management	 Link
Shanghai Commercial Bank Limited	Customer Experience	 Link
Standard Chartered Bank (Hong Kong) Limited	Anti-Fraud Measures Customer Experience	 Link

Participant	Theme	QR Code / Link
Tai Sang Bank Limited	Anti-Fraud Measures	 Link

Note: Any positions or statements made in the above factsheets / supplementary reports as prepared by Sandbox participants do not necessarily represent the views of the HKMA.