SCUOLA DI DOTTORATO
UNIVERSITÀ DEGLI STUDI DI MILANO-BICOCCA

Department of **Economics, Management and Statistics DEMS**

Ph. D. program in **PhD in Economics, Statistics and Data Science (ECOSTATDATA)**
XXXVII cycle
Curriculum of **Big Data & Analytics for Business**

Transparent and Reliable AI for the Financial Domain

Miola Arianna
Registration number: 883921

Tutor: **Manera Matteo**
Supervisor: **Panisson André, Perotti Alan**

Coordinator: **Manera Matteo**

Academic Year **2024/2025**

Abstract

Trust is the foundational currency of finance, yet its absence remains a major obstacle to the adoption of artificial intelligence across the sector. As machine learning and large language models increasingly underpin decision making, from credit scoring and risk assessment to regulatory reporting, financial institutions face the challenge of deploying systems that are not only accurate but also explainable, auditable, and reproducible. This thesis addresses the dual scientific and ethical problem of designing AI that is transparent and reliable under high-stakes operational conditions.

The proposed research develops a structured framework, termed the transparency stack, composed of three complementary pillars: Explainable AI (XAI), Grounded Generation, and Mechanistic Interpretability. XAI establishes quantitative metrics, such as Effective Compactness, Rank Quality Index, and Stability, to evaluate explanation faithfulness and reproducibility across financial models. Grounded Generation extends output transparency through verifiable generative reasoning, integrating document retrieval to ground LLM responses in verifiable sources and reduce hallucination. Mechanistic Interpretability advances internal transparency by probing the computational circuits of transformer architectures, revealing how reasoning-like behaviors emerge within model parameters.

Together, these pillars define a methodological continuum linking interpretability, verifiability, and mechanistic understanding. The thesis unites transparency and reliability as interdependent constructs: transparency enables epistemic access to AI reasoning and evidence, while reliability ensures these properties remain stable across contexts and perturbations. By bridging empirical evaluation, grounded generation, and mechanistic analysis, this work advances the development of Transparent and Reliable AI, a paradigm for financial systems that meet regulatory, ethical, and scientific standards of trust.

List of publications

- "Explainability, Quantified: Benchmarking XAI Techniques" [103], conducted in collaboration with Alan Perotti, Claudio Borile, Francesco Paolo Nerini, and André Panisson, CENTAI Institute (Center for Artificial Intelligence), Turin, Italy; and Paolo Baracco, Anti Financial Crime Digital Hub, Turin, Italy. The work was presented at the International Conference on Explainable Artificial Intelligence (XAI), 2024, and published as a chapter in "Explainable Artificial Intelligence" (Springer Nature Switzerland, pp. 421–444).

- "Keyword-based Annotation of Visually-Rich Document Content for Trend and Risk Analysis Using Large Language Models" [33], conducted in collaboration with Giuseppe Gallipoli, Simone Papicchio, Lorenzo Vaiani, Luca Cagliero, Politecnico di Torino, Turin, Italy; and Daniele Borghi, Intesa Sanpaolo Innovation Center, Turin, Italy. The work was presented at the Joint Workshop of the 7th Financial Technology and Natural Language Processing (FinNLP), the 5th Knowledge Discovery from Unstructured Data in Financial Services (KDF), and the 4th Workshop on Economics and Natural Language Processing (ECONLP), Turin, Italy, May 2024, and published in the Proceedings.

- "Retrieval Augmented Generation of Summarized Answers on Visually-Rich Documents for Trend and Risk Analysis" [32], conducted in collaboration with Giuseppe Gallipoli, Luca Cagliero, Alessandro Mosca, Politecnico di Torino, Turin, Italy; and Daniele Borghi, Intesa Sanpaolo Innovation Center, Turin, Italy. The work was presented and discussed at the 9th International Workshop on Data Analytics solutions for Real-LIfe APplications (DARLI-AP), co-located with the EDBT/ICDT 2025 Joint Conference, Barcelona, Spain, 2025, and published in CEUR Workshop Proceedings, Vol-3946.

Contents

Chapter 1

Introduction

Trust is the currency of finance and its absence is one the primary barriers to the wider adoption of Artificial Intelligence (AI) [56]. As AI is increasingly applied across the financial sector, Machine Learning (ML) and Natural Language Processing (NLP) underpin credit scoring, portfolio optimization, fraud detection, regulatory compliance, and the analysis of vast quantities of unstructured financial data [38]. Yet confidence in these systems remains fragile: in high-stakes settings, institutions require not only accurate outcomes but also explanations that are auditable and reproducible. Without such guarantees, organizations hesitate to scale deployments beyond pilots, slowing innovation and constraining future adoption [79].

This tension between performance and transparency lies at the heart of a growing research and policy debate. AI models, particularly deep learning architectures and Large Language Model (LLM), exhibit remarkable predictive and generative capabilities, but they also operate as opaque, so-called black boxes. Their internal logic, shaped by millions or billions of parameters, resists straightforward human interpretation. When applied to financial contexts, such opacity introduces significant risks: unexplainable decisions may violate regulatory frameworks; untraceable information flows may lead to misinformation or manipulation; and unreproducible behaviours undermine the reliability of analytical pipelines.

Therefore, the challenge facing contemporary AI research is not solely how to make models more powerful, but how to make them **transparent and reliable**, in other words, understandable, grounded, and trustworthy. The scientific and ethical question this thesis addresses is thus: how can we design, evaluate, and interpret AI systems that reason transparently and behave reliably, particularly in high-stakes financial environments?

1.1 The Transparency and Reliability Problem in Financial AI

In the financial domain, the requirements for model transparency are uniquely stringent. Regulatory bodies such as the European Banking Authority (EBA), the European Central Bank (ECB), and the upcoming European Union Artificial Intelligence Act (EU AI Act) have emphasized principles of explainability, accountability, and auditability [26, 27]. Institutions must be able to provide justifiable explanations for automated decisions, demonstrate control over data provenance, and

ensure model stability under changing market conditions.

However, current AI practices expose a gap between regulatory expectations and technical realities. On the one hand, financial stakeholders expect interpretability, the ability to trace outputs back to their logical or evidential foundations. On the other hand, many state-of-the-art models are inherently non-transparent: their internal representations are high-dimensional and distributed, and their outputs often rely on statistical correlations rather than interpretable causal mechanisms.

This creates three key problems:

1. **Opaque reasoning:** Models provide correct outputs without revealing why. This is especially problematic for credit risk scoring or compliance auditing, where a justification is often legally required.

2. **Unverifiable generation:** LLM-based financial assistants or chatbots may produce seemingly plausible statements that are not supported by verifiable documents, leading to "hallucinated" or misleading content.

3. **Incomprehensible mechanisms:** Even when models behave as expected, their internal computations remain largely inscrutable, limiting our capacity to diagnose or prevent systemic errors.

These challenges underscore the need for a comprehensive framework that not only interprets AI decisions post hoc but also establishes verifiable and mechanistic transparency as intrinsic properties of the system.

1.2 The Three Pillars of Transparent and Reliable AI

To address these intertwined challenges, this thesis develops a research agenda based on three complementary pillars: **Explainable AI (XAI)**, **Grounded Generation**, and **Mechanistic Interpretability**. Each pillar represents a distinct level of the transparency problem: from outputs to reasoning processes, to internal representations; and together they define a continuum of interpretability.

1.2.1 Explainable AI (XAI): Quantifying Explainability

Explainable AI seeks to make model behavior intelligible to humans [23, 67]. In the financial context, XAI methods enable analysts, auditors, and regulators to understand why a prediction or decision was made, for instance, which features most strongly influenced a credit approval, or which factors drove an anomaly detection alert.

While many explanation methods exist (e.g., SHAP, LIME, Integrated Gradients), their outputs are often difficult to compare or evaluate [75, 110]. Explanations may vary under small perturbations, exhibit bias toward certain features, or trade off faithfulness for simplicity. Thus, a key open problem is how to measure explanation quality objectively.

The first part of this thesis introduces quantitative metrics to evaluate explanation properties across models and tasks. These metrics, including Effective Compactness, Rank Quality Index, and Stability, capture different dimensions of faithfulness, reproducibility, and usability. Applied to a large-scale benchmark of financial datasets, they reveal systematic trade-offs between interpretability and performance, providing a quantitative foundation for auditing explainability in practice.

In this way, XAI constitutes the first layer of the transparency stack: it addresses what the model communicates through explanations, establishing measurable standards for interpretability.

1.2.2 Grounded Generation: Verifiable AI Outputs

With the rise of LLMs, financial institutions increasingly employ generative AI for document analysis, reporting, and question answeringe [58]. However, generative models often suffer from hallucination, that is, the production of text not grounded in factual evidence [54]. In finance, where precision and traceability are paramount, such hallucinations are unacceptable.

Grounded generation implemented through Retrieval-Augmented Generation (RAG) addresses this issue by integrating information retrieval with generation [63]. Before responding, the model retrieves relevant documents from a trusted corpus, such as financial reports, regulations, or market filings. The generated answer is then conditioned on these documents, allowing human auditors to verify the source of each statement.

The second part of this thesis investigates grounded generation for financial data, including visually rich documents that combine text, tables, and charts. It introduces methods for document annotation, evidence attribution, retrieval, and summarisation that improve both retrieval accuracy and answer faithfulness. Retrieval-Augmented Generation (RAG) is employed as a principal method to condition outputs on audited sources, with domain-adapted retrievers and summarisation steps tailored to financial corpora. Empirical results show that grounded generation substantially reduces hallucination rates and increases factual precision, enabling verifiable AI-assisted financial analysis.

Grounded generation thus forms the second layer of transparency: it links model outputs to external evidence, turning black box generation into traceable reasoning that can be independently verified.

1.2.3 Mechanistic Interpretability: Understanding the Model's Inner Workings

While XAI and grounded generation focus on external transparency, explaining or grounding outputs, **mechanistic interpretability** aims for internal transparency: understanding how neural networks implement specific computations [93, 89]. This emerging field investigates whether we can identify interpretable circuits, neurons, or subspaces corresponding to meaningful concepts, operations, or reasoning patterns.

In the financial domain, such understanding could allow us to trace how models perform numerical comparisons, parse tabular data, or reason over symbolic rules. The third part of this thesis explores experimental techniques for probing transformer-based models, including logit lens and activation patching. These methods help identify mechanistic structures responsible for reasoning-like behavior, such as induction heads.

Mechanistic interpretability provides the final layer of transparency, extending reliability from the level of outputs to the architecture itself. It enables researchers and practitioners to not only test what a model predicts or generates, but to understand how it reaches those outcomes, a prerequisite for truly trustworthy AI.

1.3 The Transparency Stack

This thesis treats transparency and reliability as interdependent foundations for trustworthy financial AI. Transparency denotes the extent to which a system's reasoning, evidence, and mechanisms are intelligible and traceable to human stakeholders, encompassing algorithmic interpretability, procedural clarity, and epistemic traceability [36, 84]. Reliability denotes the stability, reproducibility, and faithfulness of those transparent properties across time, context, and perturbations, turning transparency from a descriptive into a normative standard [52, 3].

The research follows a single methodological continuum that links outputs, evidence, and mechanisms.

The three pillars XAI, Grounded Generation, and Mechanistic Interpretability are not isolated methodologies but components of a **transparency stack**. Each layer addresses a different dimension of epistemic access:

- **Explainable AI (XAI)** provides empirical transparency: understanding model outputs through measurable explanations.
- **Grounded Generation** ensures referential transparency: grounding outputs in verifiable evidence.
- **Mechanistic Interpretability** delivers causal transparency: uncovering the inner computational structures that produce behavior.

This layered view echoes the architecture of AI itself, from inputs and outputs to internal representations. It also establishes a methodological continuity: explainability quantifies what is seen, grounded generation validates what is said, and mechanistic interpretability explains what is done internally. Together, these levels form a coherent framework for transparent and reliable AI.

Other dimensions of trustworthy AI in finance fall outside the perimeter of this thesis. These include fairness and non-discrimination; privacy and data protection; robustness and adversarial resilience; model risk management and governance; and sustainability and computational efficiency. These aspects are acknowledged where relevant, but their technical development and evaluation are not treated as core contributions.

1.4 Research Questions and Contributions

This thesis addresses the overarching research question:
How can AI systems be designed, evaluated, and interpreted to ensure transparency and reliability in high-stakes financial environments?

This question is decomposed into three specific research questions corresponding to each pillar of the transparency stack:

- **RQ1 (XAI):** How can we quantitatively evaluate the faithfulness, stability, and usability of explanation methods in financial prediction tasks?
- **RQ2 (Grounded Generation):** How can generative AI systems for financial document analysis be grounded in verifiable sources to reduce hallucinations and improve reliability?

- **RQ3 (Grounded Generation):** How can retrieval-augmented generation pipelines be adapted to the structural and multimodal characteristics of financial documents?
- **RQ4 (Mechanistic Interpretability):** What internal mechanisms enable transformer-based models to perform structured reasoning tasks relevant to financial data, such as table lookup and numerical reasoning?

The main contributions of this thesis are:

- Quantitative metrics for evaluating explanation quality, including Effective Compactness and Rank Quality Index, validated on financial use cases and presented in Chapter 3.
- An extensive benchmarking and guideline framework for selecting XAI methods in fraud detection and credit scoring tasks, described in Chapter 3.
- A retrieval-augmented generation pipeline tailored to financial documents, supporting referential transparency and reducing hallucinations in LLM outputs, developed across Chapters 4, 5, and 6.
- Mechanistic analyses of transformer models that provide empirical evidence of how specific internal components support table lookup and structured reasoning, reported in Chapter 7.
- A benchmark dataset and evaluation setup for question answering over financial documents, introduced in Chapter 6.
- A conceptual framework for transparency in financial AI, the transparency stack, integrating Explainable AI, Grounded Generation, and Mechanistic Interpretability, articulated in this Introduction and consolidated in Chapter 2.

1.5 Structure of the Thesis

The remainder of this work is organised as follows:

- **Chapter 2** provides the theoretical and technical background for the three pillars of this thesis, introducing foundational concepts in explainability, retrieval-augmented generation, and mechanistic interpretability.
- **Part I** (Chapter 3) introduces a quantitative framework for evaluating explainability methods, establishing metrics that operationalise faithfulness and stability across financial prediction tasks.
- **Part II** (Chapters 4, 5, 6) explores Grounded Generation for financial question answering, presenting novel architectures and benchmarks that ensure output traceability and faithfulness to source documents.
- **Part III** (Chapter 7) investigates Mechanistic Interpretability in transformer models, demonstrating how internal circuits can be identified and analysed to reveal the model's computational mechanisms.

In summary, this thesis argues that transparency must evolve from a rhetorical demand into a scientifically measurable property, and reliability must be understood as the stability of that transparency across the full lifecycle of AI systems. By bridging empirical evaluation, grounded reasoning, and mechanistic analysis, this work seeks to contribute to the creation of AI systems whose decisions, and decision-making processes, can be trusted.

Chapter 2

Background

The evolution of Artificial Intelligence (AI) has been marked by a dual pursuit: achieving ever-higher performance while maintaining the interpretability necessary for human understanding. In regulated and high-stakes environments such as finance, interpretability is not optional but foundational. The capacity to understand, verify, and control model behavior determines not only institutional trust but also legal compliance and ethical accountability. This chapter provides the theoretical and methodological background for the three pillars of this thesis: **Explainable AI (XAI)**, **Grounded generation**, and **Mechanistic Interpretability**; which together define a multi-layered framework for transparent and reliable AI.

2.1 Explainable AI (XAI)

This section provides the foundational understanding of XAI necessary for appreciating the technical contributions and experimental evaluations presented in the chapter 3 of this thesis.

2.1.1 What XAI is and why it exists

Machine Learning (ML) systems often behave as opaque "black boxes", making their internal decision processes difficult to follow. Explainable Artificial Intelligence (XAI) augments model outputs with human-interpretable information so that practitioners and domain experts can understand, trust, debug, and govern ML systems, especially in high-stakes domains such as finance. More broadly, XAI denotes the body of methods and principles that make complex models intelligible to human stakeholders [23, 36, 84]. Its rise reflects a persistent tension between the growing accuracy of modern ML (especially deep learning) and the opacity of their internal reasoning, often framed as a tension between accuracy and interpretability [67]. XAI encompasses three common axes: (i) **ante-hoc** versus **post-hoc** (intrinsically interpretable models versus explanations of trained black boxes), (ii) **local** versus **global** (per-instance versus model-wide), and (iii) **model-aware** versus **model-agnostic** (leveraging internals such as gradients versus treating the model purely via queries). In practice, most explainers are post-hoc and local, while model-aware and model-agnostic families are both well populated. Explanation forms vary considerably, notably **feature attributions**, **counterfactuals**, and **rules**; methods target different data types (tabular, images,

text, and graphs) and machine learning tasks [11, 46]. Chapter 3 focuses specifically on **post-hoc**, **local** explanations applied to trained black-box models across tabular classification, regression, and anomaly detection tasks.

2.1.2 Explainability in Financial Applications

In the financial sector, the demand for explainability arises not only from ethical considerations but also from a complex network of legal, regulatory, and operational requirements. Unlike many other application domains, financial decision-making directly affects individuals' access to credit, insurance, and investment opportunities, thereby raising concerns of fairness, accountability, and consumer protection. Consequently, financial institutions are expected to justify and document the logic behind algorithmic decisions, particularly when these decisions influence rights, obligations, or market positions [12, 124].

Regulatory Imperatives. The European Union's **General Data Protection Regulation (GDPR) (Article 22)** grants individuals the right not to be subject to decisions based solely on automated processing when such decisions have legal or similarly significant effects. This provision implicitly embeds an explainability requirement: affected individuals must be able to receive "meaningful information about the logic involved" in automated decision-making. In practice, this entails providing clear and intelligible explanations of the main factors influencing algorithmic outcomes, such as credit approval or risk assessment.

The forthcoming **EU AI Act** extends these obligations by explicitly classifying AI systems used in finance, including credit scoring, portfolio management, and fraud detection, as high-risk applications. For such systems, the Act mandates requirements related to transparency, robustness, and human oversight [27]. Specifically, Article 13 of the Act stipulates that high-risk AI systems must be "sufficiently transparent to enable users to interpret the system's output and use it appropriately." This legal framing transforms interpretability from a technical desideratum into a regulatory obligation.

Financial supervisory authorities have adopted similar positions. The **European Banking Authority (EBA)** emphasizes transparency and auditability as key principles in its Report on Big Data and Advanced Analytics [26], recommending that model developers document data provenance, algorithmic design, and explanation processes throughout the model lifecycle. Likewise, the **European Central Bank (ECB)** highlights the importance of explainability for model validation and stress testing in its supervisory expectations for banks using AI-driven risk models. Together, these frameworks create a multi-level regulatory environment where explainability is central to compliance, governance, and public accountability.

Ethical and Societal Dimensions. Beyond compliance, explainability in financial AI is grounded in broader ethical concerns. Decisions affecting access to credit, insurance, or investment inherently carry distributive consequences. Uninterpretable models risk reinforcing systemic biases in datasets, leading to unfair or discriminatory outcomes [6]. Transparent explanations enable both internal and external stakeholders to detect, question, and correct such biases. They also facilitate public trust, which is essential in maintaining confidence in financial institutions and markets. From this perspective, explainability functions as an ethical safeguard that complements traditional risk management frameworks.

Operational and Strategic Motivations. Explainability is also instrumental from an operational standpoint. Financial institutions operate under strict model risk management standards, such as the **Basel III** framework and the Principles for the Effective Management and Supervision of Model Risk issued by the Basel Committee on Banking Supervision (BCBS). These guidelines require that model performance, assumptions, and limitations be continuously monitored and documented. In this context, explainable AI supports:

- **Model validation**, by enabling quantitative assessment of feature contributions and sensitivity analysis;
- **Auditability**, by providing traceable evidence of decision logic for internal and external review;
- **Operational resilience**, by allowing early detection of model drift, instability, or concept shift;
- **Knowledge transfer**, by facilitating communication between data scientists, risk officers, and regulators.

Hence, explainability operates not only as a compliance mechanism but also as a driver of institutional efficiency and risk governance. By making model behavior interpretable, organizations can better align AI-driven processes with existing financial regulation and human expertise.

Quantitative Explainability as a Compliance Enabler. Despite these imperatives, most financial institutions still rely on heuristic or qualitative assessments of model interpretability. Explanations are often generated ad hoc and lack standardized evaluation criteria, making them insufficient for formal auditing. This thesis addresses this gap by introducing a set of quantitative metrics: Effective Compactness (EC), Rank Quality Index (RQI), Stability (STB), and Time (T); that enable objective measurement of explanation quality. By operationalizing transparency in measurable terms, these metrics provide a methodological bridge between regulatory principles and technical implementation.

2.1.3 Technical Overview of Explainable AI Methods

This section provides a comprehensive overview of key explainable AI (XAI) methods, organized by explanation form. These methods are foundational in explainable AI research, enabling understanding of complex ML models through diverse algorithmic paradigms, and they are the methods systematically evaluated in Chapter 3. Before detailing specific algorithms, it is useful to fix terminology around explanation forms. Table 2.1 summarizes three canonical forms for tabular settings: rule-based, attribution, and counterfactuals, with simple, concrete examples. Rules express sufficient conditions that "anchor" a prediction; attributions quantify per-feature contributions for a single instance; counterfactuals propose minimally changed alternatives that flip the outcome. The remainder of this section follows this taxonomy: attribution methods are presented first (e.g., LIME, SHAP variants, gradient-based approaches), followed by rule-based explanations (Anchors), and then counterfactuals (MACE).

Method (abbr.)	Explanation / Example (Tabular data)
Rule-Based (RB)	A set of premises that a record must satisfy to meet the rule's consequence. Example: $r:$ Education $\leq$ College $\Rightarrow$ Income $\leq$ 50k.
Attribution	A vector containing one value per feature; each value indicates the feature's importance for the classification.
Counterfactuals (CF)	The user is provided with examples similar to the input query but with a different class prediction. Example: $q:$ Education $<$ College $\Rightarrow$ Income $\leq$ 50k; $c:$ Education $\geq$ Master $\Rightarrow$ Income $\geq$ 50k.

Table 2.1: Examples of explanations divided for different explanation forms.[11]

Attribution Methods

LIME [110] (Local Interpretable Model-Agnostic Explanations) locally approximates a black-box model using an interpretable surrogate model. By generating perturbed samples near a point of interest and fitting a weighted interpretable model, LIME captures local behavior, optimizing:

$$\arg\min_{g \in G} L(f, g, \pi_x) + \Omega(g)$$

where:

- f is the original complex model.
- g is the interpretable model (e.g., a linear model).
- π_x is the locality measure, assigning weights to perturbed samples based on their proximity to the instance x.
- L is the loss function measuring the fidelity of g in approximating f.
- $\Omega(g)$ is a regularization term ensuring the simplicity of the interpretable model.

Advantages. LIME is model-agnostic, meaning it can be applied to any predictive model regardless of architecture, which makes it highly versatile [110]. It provides local interpretability by producing explanations specific to individual predictions, aiding the understanding of complex model decisions. It is also flexible across data modalities, including tabular data, images, and text.
Limitations. Despite its strengths, LIME can be unstable, with explanations varying under different perturbation seeds or sampling, reducing reliability. It depends heavily on synthetic perturbations that may not reflect the true data distribution, and it can be computationally expensive on large datasets or complex models due to the many model evaluations on perturbed samples [46].

SHAP [75] (SHapley Additive exPlanations) is a unified framework for interpreting machine learning models by computing feature attributions based on cooperative game theory. It provides a mathematically rigorous approach to explain individual predictions by quantifying each feature's contribution to the difference between the current prediction and the expected model output [75].

SHAP is grounded in Shapley values from cooperative game theory, where the "game" is the prediction task and "players" are the input features. The core principle is to fairly distribute the prediction among features based on their marginal contributions across all possible coalitions [75].

SHAP represents explanations as an additive feature attribution method, expressed as a linear model:

$$g(\mathbf{z}') = \phi_0 + \sum_{j=1}^{M} \phi_j z'_j$$

where:

- g is the explanation model,
- $\mathbf{z}' = (z'_1, \ldots, z'_M)^T \in \{0,1\}^M$ is the coalition vector (1 = feature present, 0 = feature absent),
- M is the maximum coalition size,
- $\phi_j \in \mathbb{R}$ is the SHAP value (feature attribution) for feature j,
- ϕ_0 represents the baseline/expected model output $\mathbb{E}[\hat{f}(X)]$.

The SHAP values are computed using the classical Shapley value formula:

$$\phi_j = \sum_{S \subseteq F \setminus \{j\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} \left[f_x(S \cup \{j\}) - f_x(S) \right]$$

where:

- F is the set of all features,
- S is a subset of features not including feature j,
- $f_x(S)$ is the expected model output given features in set S.

SHAP satisfies three fundamental properties that uniquely determine additive feature attributions [75, 86]:

1. **Local Accuracy (Efficiency):**

   $$\hat{f}(\mathbf{x}) = g(\mathbf{x}') = \phi_0 + \sum_{j=1}^{M} \phi_j x'_j$$

   The explanation model matches the original model output when all features are present.

2. **Missingness:**

   $$x'_j = 0 \Rightarrow \phi_j = 0$$

   Missing features receive zero attribution.

3. **Consistency:** If a feature's marginal contribution increases across all coalitions when moving from model f to model f', then its SHAP value should also increase or remain the same.

Specialized algorithms for computing SHAP values include KernelSHAP, TreeSHAP, and DeepSHAP, each tailored to different model types and computational considerations:

KernelSHAP [75] is a model-agnostic method, meaning it can be used with any machine learning model. It approximates SHAP values using a weighted linear regression on perturbed samples of the input data, where features are systematically included or excluded to estimate their contribution to the prediction. While flexible, KernelSHAP can be computationally expensive, especially for models with many features, because it requires many model evaluations to estimate the values accurately.

TreeSHAP [76] is a specialized algorithm for tree-based models that computes exact SHAP values efficiently [76, 74]. Instead of evaluating all 2^M possible coalitions, TreeSHAP leverages the tree structure to compute expectations over feature subsets recursively. TreeSHAP uses dynamic programming to efficiently compute the contribution of each feature by traversing the decision paths in the tree ensemble, resulting in polynomial time complexity $O(TLD^2)$ where T is the number of trees, L is the number of leaves, and D is the maximum depth of the trees.

TreeSHAP supports two variants:

- **Interventional TreeSHAP (marginal):** Computes Shapley values under an interventional distribution that breaks feature dependencies, taking expectations over marginal feature values from a background dataset. This is "true to the model": features never used by the trees receive zero attribution (satisfies the Dummy axiom), and importance reflects the model's response to controlled interventions. Caveat: it may evaluate off-manifold feature combinations, which can be less intuitive in highly dependent data [14]. In practice, it requires a background dataset (or synthetic baselines) to take expectations.
- **Path-dependent TreeSHAP (observational):** Uses conditional expectations along tree paths, preserving empirical correlations. This is "true to the data" and stays on-manifold, but can assign credit to proxy/correlated features (including ones unused by the model), potentially violating Dummy and making attributions dataset-dependent [14]. A practical advantage is that it can operate without an external background dataset by relying on the tree's internal training statistics.

DeepSHAP [15] generalizes SHAP to neural networks by combining Shapley value theory with DeepLIFT's layerwise relevance propagation rules. It backpropagates attributions across layers relative to a reference (baseline) input by comparing each neuron's activation to its baseline activation, composing local contributions to approximate global Shapley values[15, 116]. In practice, DeepSHAP uses a small background set to estimate expectations and scales efficiently on deep architectures, avoiding the heavy sampling of model-agnostic methods like KernelSHAP.

Integrated Gradients [120] (IG) core idea is to measure how much each input feature contributes to the model's output by accumulating gradients along a straight path from a baseline input (such as a black image or zero vector) to the actual input. This approach addresses the limitations of simple gradient methods, which can be misleading due to issues like gradient saturation (where gradients become very small and uninformative near the input). For each feature i, the integrated gradient is defined as:

$$\mathrm{IG}_i(x) = (x_i - x_i') \int_0^1 \frac{\partial F(x' + \alpha(x - x'))}{\partial x_i} \, d\alpha$$

where x is the input to be explained, x' is the baseline input (e.g., all zeros), F is the model prediction function, and α is a scaling parameter[120].

IG can be applied to any differentiable model and it satisfies desirable properties such as completeness, meaning the attributions sum up to the difference between the model's output for the actual input and the baseline.

Some limitations are that the results are sensitive to the choice of baseline input: an inappropriate baseline may yield misleading attributions. Furthermore, IG requires multiple model evaluations for different interpolated inputs, making it computationally expensive for large models or high-dimensional data [46].

Saliency [117] is one of the simplest and most intuitive post-hoc interpretation techniques for neural networks, particularly in the context of deep learning models [117]. The main idea is to use the gradient of the model's output with respect to the input features to identify which parts of the input are most influential in the model's prediction.

The core idea is to use the gradient of the model's output with respect to the input features to identify important regions. Mathematically, given a neural network $f(x)$ and an input x, the saliency map $S(x)$ for class c is defined as

$$S(x) = \left| \frac{\partial f_c(x)}{\partial x} \right|,$$

where:

- $f_c(x)$ is the output of the model for class c;
- $\frac{\partial f_c(x)}{\partial x}$ is the gradient of the output with respect to the input x.

This gradient indicates how sensitive the output $f_c(x)$ is to changes in each input feature x_i. Features with higher gradient magnitudes are deemed more important for the prediction.

Saliency maps are easy to compute and understand, providing immediate insights into model behavior. Moreover, they can be applied to any differentiable model. However, saliency maps can be noisy and may highlight irrelevant features, especially in high-dimensional inputs. They also suffer from gradient saturation issues, if the model's output is saturated, the gradient values may be close to zero, making the saliency map less informative [46].

Input×Gradient [116] is a simple feature attribution method where the importance of each input feature is estimated by multiplying the value of the input feature by the gradient of the model's output with respect to that input [116]. For a model F and input x, the attribution for feature i is:

$$\text{Attribution}_i = x_i \cdot \frac{\partial F(x)}{\partial x_i}$$

This method highlights features that both have a large value and to which the model is sensitive (as indicated by the gradient). It is computationally efficient and provides a first-order approximation of feature importance, but can be limited by issues like gradient saturation (as other gradient-based methods like Saliency).

Deconvolution [137] (often called "deconvnet" or "deconvolutional network") is a visualization technique introduced in [137]. It is designed to help interpret the internal representations learned by convolutional neural networks (CNNs). The core idea is to project feature activations from a given layer of a trained CNN back into the input pixel space, effectively "inverting" the operations of the network. This is done by reversing the sequence of operations (such as convolution, pooling, and non-linearities) to reconstruct an approximate image that highlights which parts of the input most strongly activate a particular feature map or neuron. Although originally proposed for convolutional architectures, the deconvnet backward rule constitutes a local modification of backpropagation at ReLU gates and does not rely on convolutional structure; consequently, the same rule is applied in Chapter 3 to the ReLU-MLP to obtain explanations for tabular inputs and to provide a consistent gradient-based baseline across architectures.

Occlusion [137] systematically masks small patches of the input and observes prediction changes, producing saliency maps highlighting influential elements. This approach is model-agnostic but can be computationally expensive[137].

COIN [69] (Contextual Outlier Interpretation) is a model-agnostic framework proposed to provide interpretable explanations for outliers detected by anomaly detection systems. COIN explains why the instance was considered an outlier, examining its features and its relationship to surrounding (normal) data points.

Operationally, for each detected outlier x^*, COIN first identifies a local context by retrieving its nearest normal neighbors, thereby capturing the immediate data distribution against which x^* should be compared. Within this context, it learns a simple and interpretable decision boundary, typically a sparse linear classifier or a shallow decision tree, that separates the outlier from its neighbors. The learned coefficients or split conditions directly reveal which features are most abnormal for that specific instance. When the neighborhood is heterogeneous, COIN decomposes the context into multiple subclusters and analyzes each subregion separately, producing fine-grained explanations that respect local structure rather than imposing a single global rationale. Finally, it assigns an outlierness score grounded in margins of separation and supports refinement with prior domain knowledge by reweighting or constraining attributes [69].

ACE [141] (Anomaly Contribution Explainer) is a model-agnostic method to explain anomaly scores generated by anomaly detection models. It was developed particularly for cyber-security applications. The primary goal of ACE is to provide feature-level attributions, showing which features contribute most to an instance being flagged as an anomaly. Concretely, ACE builds a local linear surrogate around the queried anomalous instance. It samples small perturbations ("neighbors") of the anomalous instance, queries the black-box detector to obtain the corresponding anomaly scores, and fits a weighted linear regression to mimic the local behavior of the black-box model. The fitted coefficients are interpreted as feature contributions. To report only supporting evidence for abnormality, ACE applies a softplus transformation to each weight, $\text{softplus}(w_j) = \log\big(1 + e^{w_j}\big)$, which suppresses negative values and retains positive contributions. Finally, these transformed contributions are normalised to express each feature's relative share of the overall anomaly score for the instance.

DIFFI [13] (Depth-based Isolation Forest Feature Importance) is a model-specific interpretability method developed to explain the predictions made by Isolation Forest (IF) models. It leverages

the Isolation Forest (IF) structure, where trees isolate points by recursively splitting on randomly selected features. In the global setting, DIFFI aggregates evidence across all trees and assigns higher importance to features that repeatedly contribute to isolating anomalous points near the root and that create strong child-node imbalances. To quantify how "isolating" a split is, DIFFI introduces the Induced Imbalance Coefficient (IIC): a split has a high IIC when one child receives very few samples (ideally a single instance), exactly the behaviour IF exploits to single out rare outliers. A feature's global score is then accumulated over its occurrences in such valuable splits, downweighted with depth (shallow splits count more than deep ones) and normalized to correct for IF's random feature subsampling and differing opportunities of selection. This yields a comparable, model-specific global ranking of attributes that most effectively separate outliers from inliers in the forest [13].

Local-DIFFI applies the same logic to a single prediction. For a given instance, it traces the path the instance follows in every tree and accumulates per-feature contributions for the splits encountered along those paths, giving more credit when the instance becomes isolated at a shallow depth and when the intervening splits have high IIC in that local context. Summing over trees produces a per-instance importance vector that explains why that particular point is deemed anomalous, in terms of the features that most directly drove its early isolation in the ensemble [13].

Rule-based Explanations

Anchors [111] is a model-agnostic method designed to provide human-interpretable rules for individual predictions of complex machine learning models. The core idea is to identify "anchor" conditions, sets of feature constraints, such that, if these conditions are met, the model's prediction is highly likely to remain unchanged, even if other features vary. An anchor is a rule of the form: "If these features take these values, then the model will predict this outcome with high probability, regardless of the values of other features." To obtain the anchors, anchor perturbs the instance x obtaining a set of synthetic records employed to extract anchors with precision above a user-defined threshold. First, since the synthetic generation of the dataset may lead to a massive number of samples anchor exploits a multi-armed bandit algorithm. Second, since the number of all possible anchors is exponential anchor uses a bottom-up approach and a beam search [111, 11, 86].

Counterfactual Explanations

MACE [134] (Model-Agnostic Counterfactual Explanation) is a framework that generates counterfactual explanations. The method aims to explain model decisions by identifying minimal feature changes that would alter a prediction, such as "if income were above $50,000, the loan would have been approved."

MACE is designed to work with both differentiable and non-differentiable models (e.g., XGBoost or random forests) and to handle categorical features effectively without assuming continuity. It achieves this through a four-stage pipeline:

1. Counterfactual feature selection: identifies influential features and possible value changes using k-nearest neighbor search.

2. Counterfactual feature optimization: uses a reinforcement learning (RL)-based algorithm (REINFORCE) to generate candidate counterfactuals that optimize validity and sparsity.

3. Counterfactual example selection: chooses diverse and proximate counterfactuals to ensure variety and interpretability.
4. Continuous feature fine-tuning: applies a gradientless descent method to refine numerical feature values for better proximity and realism.

The diversity of these methods illustrates the nuanced nature of interpretability, addressing transparency needs across models, input types, and application domains.

Table 2.2: Classification of Explainable AI Methods by Key Taxonomy and Task Type

Method	**Model-aware or Agnostic**	**Explanation Form**	**Task Type**
Anchors	Agnostic	Rules	Classification
MACE	Agnostic	Counterfactuals	Classification
LIME	Agnostic	Attribution	Classification, Regression
KernelSHAP	Agnostic	Attribution	Classification, Regression, Anomaly detection
TreeSHAP	Aware (Tree)	Attribution	Classification, Regression
DeepSHAP	Aware (NN)	Attribution	Classification, Regression
Integrated Gradients	Aware (NN)	Attribution	Classification, Regression
Saliency	Aware (NN)	Attribution	Classification
Input×Gradient	Aware (NN)	Attribution	Classification
Deconvolution	Aware (NN)	Attribution	Classification
Occlusion	Agnostic	Attribution	Classification
ACE	Agnostic	Attribution	Anomaly detection
DIFFI	Aware (IsoForest)	Attribution	Anomaly detection
COIN	Agnostic	Attribution	Anomaly detection

Table 2.2 summarizes these methods along key taxonomy dimensions and their supported task types.

2.1.4 Why evaluating explainability methods is essential

Despite the proliferation of XAI techniques, there is no universal agreement on what constitutes a "good" explanation. Many works enumerate desirable properties (faithfulness, stability, simplicity) [91], yet operational definitions that practitioners can compute are often missing. In practice, engineers must choose among many explainers without quantitative guidance, risking degraded trust, misleading narratives, or unacceptable latency. In regulated domains, explainability also intersects with compliance and risk management (e.g., transparency and accountability requirements), further motivating standardized, quantitative evaluation [118].

Existing libraries and benchmarks (often classification- or image-centric) provide valuable pieces but typically constrain comparisons to a single task or explanation family [85]. Chapter 3 contributes a unified, minimal metric set and a cross-task, cross-explanation benchmark focused on

tabular settings central to finance, along with concrete practitioner guidelines (see the 10-step checklist in Chapter 3).

2.1.5 Related Work

The preceding sections established that XAI methods vary widely in their explanation forms, model assumptions, and task applicability. This diversity raises a natural question: how should these methods be compared and selected in practice? This section reviews prior work on evaluating explainability, focusing on approaches relevant to tabular modelling and financial applications. The literature broadly divides into two strands: conceptual frameworks that articulate desiderata for explanations, and practical toolkits that implement quantitative tests for selected explanation families.

Conceptual Frameworks and Desiderata Several surveys advocate for metric-based systematisation of XAI. Longo et al. [72] observe that "what is missing is a set of evaluation metrics for explainability that are generally applicable across studies, contexts, and settings". Arrieta et al. [6] call for "further efforts towards new proposals to evaluate the performance of XAI techniques, as well as comparison methodologies among XAI approaches that allow contrasting them quantitatively under different application context, models and purposes". Saeed et al. [112] similarly assert that "further progress is needed towards evaluating XAI techniques' performance and establishing objective metrics for evaluating XAI approaches in different contexts, models, and applications". Hooker et al. [45] emphasise the importance of building "a framework to empirically validate the relative merits and reliability of these [explainability] methods".

In response, multiple studies have proposed lists of desirable explanation properties. Nauta et al. [91] identify twelve "co-*" properties (Correctness, Completeness, Consistency, Continuity, Contrastivity, Covariate complexity, Compactness, Composition, Confidence, Context, Coherence, Controllability). Zhou et al. [143] outline key properties and their relation to evaluation metrics, while Lopes et al. [73] propose Clarity, Broadness, Simplicity, Completeness, and Soundness as computer-centred evaluation criteria. Although these frameworks contextualise explanations from multiple perspectives, they rarely provide operational implementations, thereby limiting their practical utility. Other work focuses on human-centred evaluation, examining how explanation complexity affects users' ability to learn, understand, and trust AI systems [139].

Practical Toolkits and Benchmarks Complementing conceptual work, several open-source frameworks operationalise XAI evaluation through quantitative metrics and standardised datasets.

OpenXAI [1] provides a framework for evaluating and benchmarking post-hoc explanation methods, offering quantitative metrics, a synthetic data generator, diverse datasets, and pre-trained models. It standardises assessment of faithfulness, stability, and fairness but focuses primarily on classification tasks.

XAI-Bench [70] introduces a suite of synthetic datasets and a library for benchmarking feature attribution algorithms, emphasising parametrised data simulation to approximate real-world conditions. However, XAI-Bench is limited to classification tasks and attribution-based explanations.

Quantus [43] is an actively maintained project organising metrics into six categories (Faithfulness, Robustness, Localisation, Complexity, Randomisation). It supports attribution-based techniques but focuses predominantly on image classification.

Gaps and Motivation for This Work The principal limitation of existing benchmarks is their restriction to a single ML task or explanation type, preventing broader comparison among explainers [60]. Furthermore, the proliferation of proposed properties may create information overload, requiring practitioners to navigate numerous metrics without clear guidance on prioritisation.

Chapter 3 addresses these gaps by introducing a unified, minimal metric set (Effective Compactness, Rank Quality Index, Stability, and Time) and a cross-task benchmark spanning classification, regression, and anomaly detection on tabular data. This approach complements existing resources by enabling systematic evaluation across explanation forms and task types, while providing actionable guidance for practitioners in regulated domains such as finance.

2.2 Grounded Generation

This section introduces grounded generation as a design principle for controllable, auditable generative systems, and situates RAG as a canonical realisation within this broader class. This section provides the foundational understanding necessary for appreciating the technical contributions and experimental evaluations presented in Chapters 4, 5, and 6.

Large Language Models (LLMs) have demonstrated remarkable generative capabilities, but their outputs often lack factual reliability when not directly connected to verifiable evidence. Grounded Generation refers to the paradigm in which a language model produces text explicitly conditioned on external, trustworthy sources, such as documents, databases, or structured knowledge, rather than relying solely on its internal parametric memory. The term grounding thus denotes the process of anchoring generated content in external reality, ensuring that model responses can be verified, traced, and attributed to concrete sources.

Grounded generation can be viewed as a fundamental shift from purely probabilistic text generation toward evidence-based reasoning. In this framework, the model is not only trained to produce fluent language but also constrained to align its statements with retrieved or provided data. This mechanism provides a bridge between semantic understanding and epistemic responsibility, two essential conditions for deploying LLMs in high-stakes domains such as finance.

As the next chapters will illustrate, grounded generation serves as the conceptual foundation for Retrieval-Augmented Generation (RAG), a specific implementation that operationalizes grounding via information retrieval systems. While grounded generation defines the principle of factually aligned generation, RAG defines the methodological realization of that principle through retrieval, embedding, and reasoning modules.

2.2.1 Retrieval Augmented Generation (RAG)

RAG arises at the intersection of information retrieval and conditional text generation. Large language models exhibit strong fluency and emerging reasoning, yet they may produce confident statements without evidential support, a behaviour often labelled hallucination [54]. In finance, where precision, verifiability, and compliance are non-negotiable, this risk is material.

As a grounded generation method, RAG links every answer to retrievable, human-verifiable sources. Transparency follows from traceability: the system surfaces the specific passages that informed a response, enabling auditors to inspect evidence and reproduce outcomes. Operationally,

RAG also decouples knowledge from parameters: updating the index or corpus refreshes model knowledge without fine-tuning.

What RAG is and why it exists RAG enhances an LLM by conditioning on context retrieved from an external knowledge base [63]. Given a query, a retriever selects candidate passages, and the generator produces an answer using the query with those passages. This design improves source transparency and accountability by exposing which documents support the output, and it mitigates temporal staleness by allowing immediate incorporation of new material through reindexing. These properties are well matched to regulated financial workloads that rely on extensive, frequently updated documentation.

Core RAG Architecture

The standard RAG architecture comprises three fundamental components that work in concert to deliver enhanced information processing capabilities: knowledge base, retrieval system, and generation model [29, 34]:

Knowledge Base and Indexing The knowledge base serves as the external repository of information that augments the LLM's internal knowledge. This can include structured databases, document collections, or specialized domain-specific corpora. The knowledge base is indexed using various embedding techniques to enable efficient semantic search and retrieval operations.

Retrieval System The retrieval component is responsible for identifying and extracting relevant information from the knowledge base based on user queries. Modern retrieval systems employ both dense and sparse embedding approaches to capture different aspects of semantic similarity:

- **Dense Retrieval**: Utilizes neural embedding models that represent text as high-dimensional vectors, enabling semantic similarity matching even when exact keyword overlap is absent.
- **Sparse Retrieval**: Employs traditional information retrieval methods such as BM25 and Term Frequency–Inverse Document Frequency (TF-IDF) that excel at exact keyword matching and terminology-based searches.
- **Hybrid Retrieval**: Combines dense and sparse approaches using techniques like Reciprocal Rank Fusion (RRF) to leverage the strengths of both methodologies.

Generation Model The generation component comprises an LLM that synthesises retrieved information with the original query to produce contextually informed responses. The generator may be implemented using various architectural approaches, including encoder-decoder models or decoder-only architectures accessed via API endpoints. Chapter 6 experimentally investigates the performance of instruction-tuned LLMs (e.g., gpt-4o) compared with reasoning LLMs (e.g., OpenAI o1) within the generator component.

Base LLMs are autoregressive next-token predictors: they generate text left-to-right by estimating the next token conditioned on the previous context, learned from massive corpora via the next-token objective.

Instruction-tuned LLMs retain this autoregressive core but, in the assistant setting, are obtained by post-training a pre-trained base model with Supervised Fine-Tuning (SFT) and Reinforcement Learning from Human Feedback (RLHF) on conversational data. Training and deployment typically proceed as follows (see Figure 2.1 for SFT and RLHF):

1. **Pre-training.** The model is trained on vast corpora of filtered internet text (often trillions of tokens, e.g., ~15 trillion) to minimize next-token prediction loss. The resulting base model is a general-purpose sequence predictor that can be viewed as a probabilistic, lossy compressor of its training distribution (an "internet document simulator") [102].
2. **Supervised Fine-Tuning (SFT).** The base model is fine-tuned on curated human–assistant dialogues that specify desired behaviors (helpful, truthful, harmless), teaching instruction following, tone, and format [98].
3. **Reinforcement Learning from Human Feedback (RLHF).** This is the step that follows SFT. For each prompt, humans rank multiple model-generated responses from best to worst. These preference rankings are used to train a reward model $R_\phi(x, y)$ that scores candidate responses. The policy is then optimized to produce high-reward outputs while remaining close to the SFT policy (e.g., Proximal Policy Optimization (PPO) with a Kullback–Leibler (KL) divergence penalty). RLHF is especially valuable for subjective or open-ended tasks (e.g., creative writing, summarization, safe/helpful dialogue) where no single correct answer exists, yielding assistants that better imitate expert human labelers and behave more interactively [98].

Reasoning models ("thinking" models) extend instruction-tuned assistants with a third training stage based on Reinforcement Learning (RL). After pre-training and SFT, the policy is further optimized on verifiable, correctness-centric tasks (e.g., mathematics, programming, formal logic), encouraging the discovery of reliable multi-step token sequences that actually solve problems, rather than merely imitating expert style. Empirically, this yields emergent deliberative behaviors such as planning, hypothesis testing, backtracking, and revision, consistent with a move from fast, pattern-matching System 1 to deliberate System 2 reasoning [64].

Representative systems include OpenAI's o1 and o3 families and DeepSeek-R1 [122], which exemplify RL-driven training and variable test-time compute. In aggregate, such models typically deliver substantial gains in mathematical and logical reasoning (including multi-step proofs and symbolic manipulation), notable improvements in code generation and verification via test-based rewards and verifier-guided search, and better performance on planning and constraint satisfaction where explicit search and reflection reduce brittle shortcutting.

Practically, relative to classical assistants, these models yield enhanced logical consistency and faithfulness to verifiable objectives, stronger error detection and self-correction with improved robustness on tail cases, and a more systematic, stepwise problem-solving approach rather than surface-level pattern matching.

In this thesis, the term "reasoning model" denotes LLMs that perform this RL stage on verifiable tasks and support test-time compute scaling; they are compared against SFT assistants both conceptually (Table 2.3) and empirically in downstream evaluations.

Mathematical Formulation of RAG

Core Probabilistic Model The generation process in RAG can be formally expressed as modeling the **conditional probability distribution** [115]:

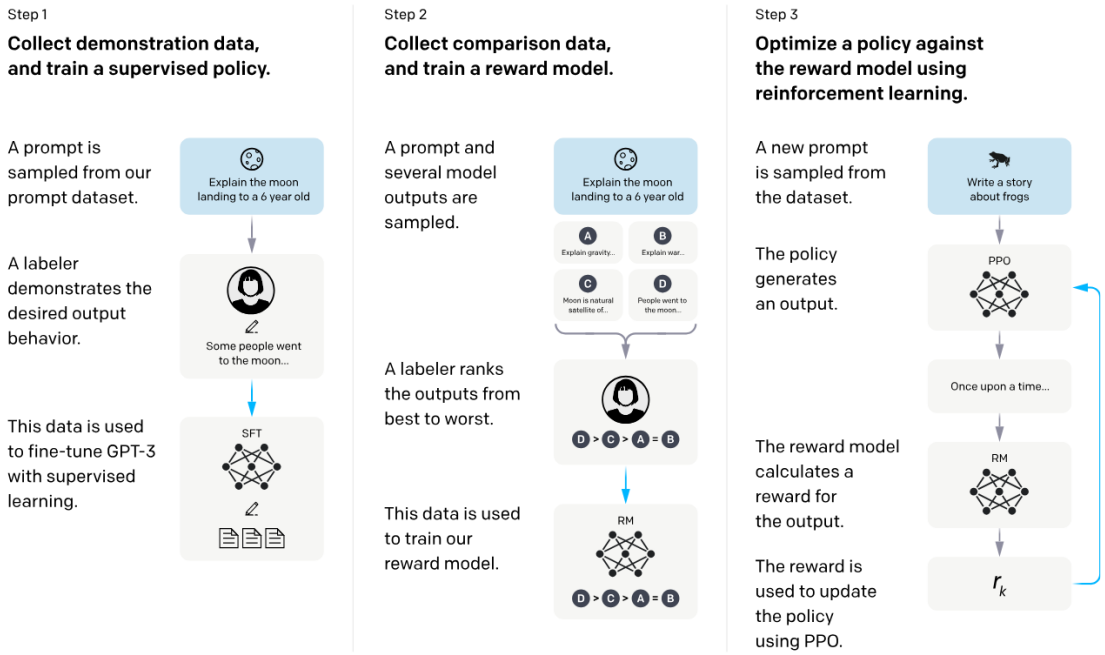


Figure 2.1: A diagram illustrating the three steps of our method: (1) Supervised Fine-Tuning (SFT), (2) Reward Model (RM) training, and (3) reinforcement learning via Proximal Policy Optimisation (PPO) on this reward model. Figure from [98].

$$P(y \mid x) = \sum_{d \epsilon \mathcal{C}} P(y \mid x, d) \cdot P(d \mid x) \tag{2.1}$$

where:

- x represents the input query or prompt
- d denotes a retrieved document from the corpus $\mathcal{C}$
- y is the generated response

In practical implementations, this summation is approximated by retrieving the top-k most relevant documents $d_1, d_2, \ldots, d_k$, yielding the **approximated formulation**:

$$P(y \mid x) \approx \sum_{i=1}^{k} P(y \mid x, d_i) \cdot P(d_i \mid x) \tag{2.2}$$

where $d_1, d_2, \ldots, d_k$ are the top-k documents retrieved for x.

This top-k approximation is necessary because summing over the entire corpus $\mathcal{C}$ is computationally infeasible for large document collections; restricting to the most relevant k documents balances efficiency and retrieval quality.

Feature	**Instruction-tuned LLM (SFT Assistant Model)**	**Reasoning Model (RL Thinking Model)**
Training Stage	Follows pre-training; tuned with Supervised Fine-Tuning (SFT).	Follows SFT; tuned with Reinforcement Learning (RL).
Training Data Source	Curated datasets of conversations demonstrating ideal expert responses.	A large collection of verifiable practice problems (e.g., math and code).
Training Goal	Imitate human expert behavior and adopt the persona of an assistant.	Discover reliable token sequences that maximize the chance of getting the correct final answer.
Internal Process	Acts as a straightforward statistical imitation of a human labeler.	Exhibits emergent cognitive strategies and "chains of thought" by distributing reasoning across many tokens.
Cognitive Mode (System 1 vs System 2) [64]	Fast, intuitive System 1: direct token-by-token generation; limited deliberation; performance driven by pattern recognition.	Slow, deliberate System 2: step-by-step reasoning; variable test-time compute; self-correction; explicit chain-of-thought.
Output Style	Provides a mimicry of a well-written expert solution.	Often involves a visible "thinking process" (e.g., re-evaluating steps, trying different perspectives) before generating the final answer.
Ideal Use Case	General questions, creative writing, simple knowledge retrieval, and dialogue.	Difficult problems in math, coding, and logical reasoning, where they yield higher accuracy.

Table 2.3: Comparison between a standard SFT assistant LLM and a reasoning (RL thinking) model.

This formulation highlights that RAG systems rely on two key probabilistic components: retrieval and generation.

The RAG framework decomposes into two fundamental probability estimations:

Retrieval Probability: $P(d_i \mid x)$ represents the relevance score of document d_i given input x, typically derived from similarity metrics or reranking mechanisms.

Generation Probability: $P(y \mid x, d_i)$ models the probability of generating output y conditioned on both the input x and retrieved document d_i, handled by the language model.

Vector Space Mathematics The knowledge base transformation involves converting textual content into high-dimensional vector representations through **embedding functions**:

$$e_{\text{doc}} = f_{\text{embed}}(\text{document}) \tag{2.3}$$

$$e_{\text{query}} = f_{\text{embed}}(\text{query}) \tag{2.4}$$

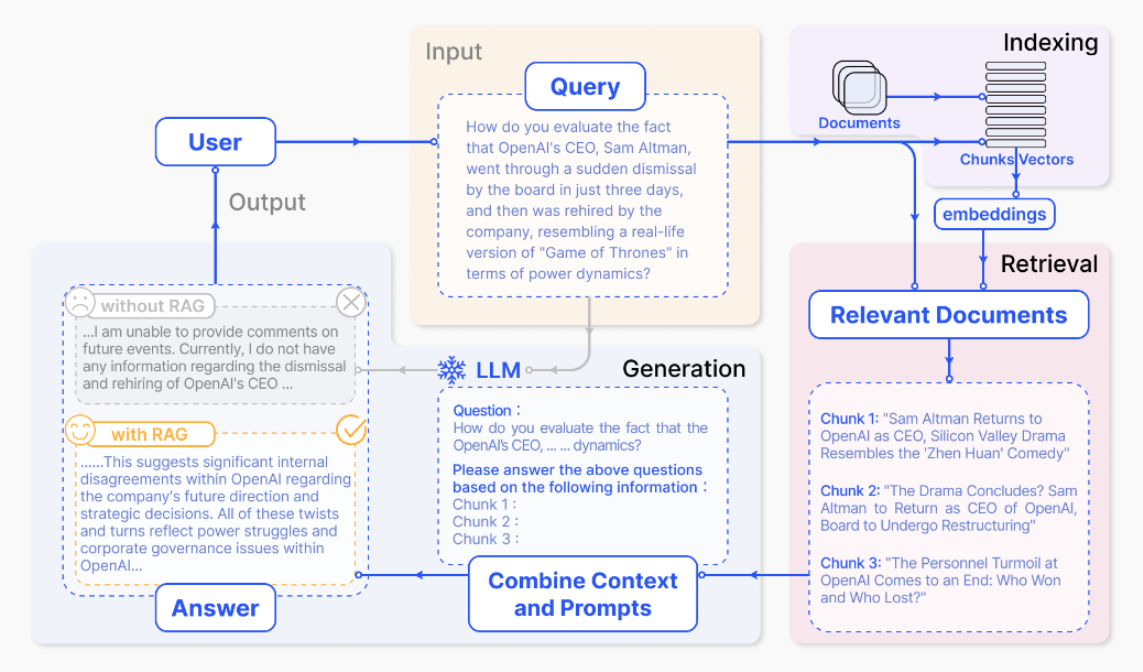


Figure 2.2: A representative instance of the RAG process applied to question answering. It mainly consists of 3 steps. 1) Indexing. Documents are split into chunks, encoded into vectors, and stored in a vector database. 2) Retrieval. Retrieve the top-k chunks most relevant to the question based on semantic similarity. 3) Generation. Input the original question and the retrieved chunks together into LLM to generate the final answer. Figure from [34].

where f_{embed} represents the embedding model that maps text to $\mathbb{R}^d$ space.
The retrieval mechanism relies on **cosine similarity** as the primary similarity metric:

$$\text{sim}(e_{\text{query}}, e_{\text{doc}}) = \frac{e_{\text{query}} \cdot e_{\text{doc}}}{\|e_{\text{query}}\| \|e_{\text{doc}}\|} \tag{2.5}$$

For normalized vectors A and B, this simplifies to the **dot product**:

$$\text{sim}(A, B) = A \cdot B = \sum_{i=1}^{d} A_i \cdot B_i \tag{2.6}$$

The cosine similarity yields values in the range $[-1, 1]$, where:

- 1 indicates semantically identical vectors
- 0 indicates no semantic relationship
- -1 indicates semantically opposite vectors

The retrieval system implements **k-nearest neighbor search** to identify the most relevant documents:

$$\mathcal{D}_{\text{retrieved}} = \arg \max_{d_1,\ldots,d_k \in \mathcal{C}} \sum_{i=1}^{k} \text{sim}(e_{\text{query}}, e_{d_i}) \tag{2.7}$$

subject to the constraint that $|\mathcal{D}_{\text{retrieved}}| = k$.

Advanced RAG Techniques

Contextual Chunk Enrichment Contextual chunk enrichment represents an advancement in retrieval-augmented generation (RAG) systems designed to address the critical problem of context loss when documents are segmented into smaller processing units. This technique fundamentally transforms how information retrieval systems maintain semantic coherence while optimizing computational efficiency [4] .

Contextual chunk enrichment refers to the process of enhancing individual document chunks by augmenting them with additional contextual information from their surrounding text or parent document before embedding and indexing. Unlike traditional chunking methods that create isolated text segments, this approach ensures each chunk retains meaningful connections to its broader documentary context.

The methodology operates on the principle that smaller units of data are enriched with additional context to improve retrieval precision while maintaining semantic integrity. This addresses a fundamental limitation in conventional RAG systems where document fragmentation often results in ambiguous or incomplete information retrieval.

For example, an original chunk stating "The company's revenue grew by 3% over the previous quarter" would be transformed into: "This chunk is from an SEC filing on ACME corp's performance in Q2 2023; the previous quarter's revenue was $314 million. The company's revenue grew by 3% over the previous quarter".

The contextualization process leverages large language models to automatically generate chunk-specific context. A typical implementation uses prompts that instruct the model to provide concise, chunk-specific context explaining the chunk within the overall document framework. The resulting contextual text, usually 50-100 tokens, is prepended to the original chunk before processing.

This technique was implemented in Chapter 6 using the GPT-4.1 model (released in April 2025), which provides a context window exceeding one million tokens (specifically, 1,047,576), thereby allowing extremely large documents to be entered into the prompt. However, in some cases this was not enough to contain the entire document, as some exceeded a thousand pages. To overcome this limitation, the code was structured to allow the selection of a subset of the original document, called source content, which includes the chunk in question and its relevant sections. The central logic for determining which portions of the document to include as source content is based on the `is_source_content_similar` function, which evaluates whether two chunks belong to the same conceptual section. This function analyses the heading hierarchies of the chunks, where the headings are represented as dictionaries that associate each level with its respective title. The implemented algorithm goes beyond a simple comparison of equality, also recognising when a set of headings constitutes a subset of another, a common situation when chunks represent different levels of detail within the same section of the document. The comparison begins by checking whether the difference between the depths of the headings exceeds the `max_level_difference` threshold, which provides a quick exclusion of clearly unrelated sections. Next, a "sharing" measure is calculated by identifying the longest prefix of common headings between the two chunks. The intuition behind this approach is that chunks with headings such as "Chapter 1 ¿ Section A" and "Chapter 1 ¿ Section A ¿

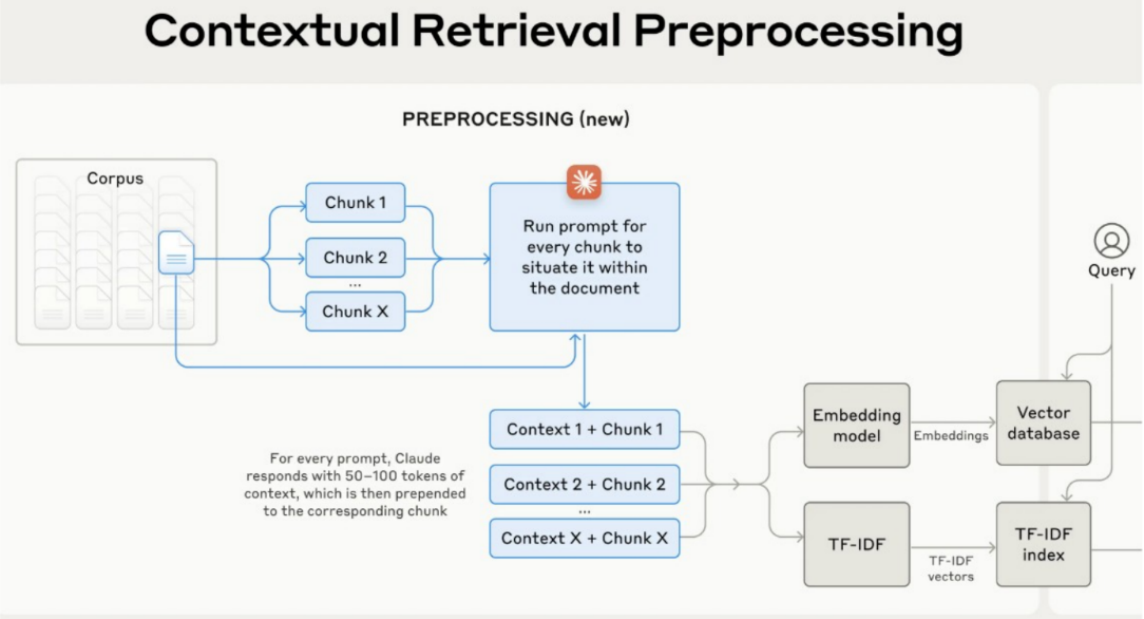


Figure 2.3: Contextual Retrieval Preprocessing. Figure from [4].

Subsection 1" should be considered similar, as they belong to the same conceptual area while representing different levels of detail. The final decision on similarity takes into account whether the gap between the broader heading structure and the shared portion falls within the acceptable limits set by the `max_level_difference` parameter.

The `max_level_difference` value therefore controls the maximum acceptable difference in the heading hierarchy: the higher the parameter, the greater the difference that will be tolerated. Its value ranges from 1 to 6; in the maximum case (6), the entire document is considered source content.

When invoking the API to generate the context of a chunk, the length of the prompt is first checked using the OpenAI `tiktoken` library. If the prompt is too long compared to the GPT-4.1 context window, the `max_level_difference` parameter is progressively reduced until a source content is found that, once included in the prompt, ensures compliance with the maximum number of tokens allowed by the model.

Reranking Reranking represents a critical optimization component within Retrieval-Augmented Generation (RAG) systems, addressing fundamental limitations of initial retrieval mechanisms [106, 131]. In traditional RAG architectures, the retrieval phase employs fast, scalable methods such as embedding-based similarity search or keyword matching to identify potentially relevant documents from large corpora [123]. However, these initial retrieval approaches often prioritize efficiency over precision, resulting in the selection of documents that may contain semantic noise or fail to capture nuanced query-document relationships [106, 131].

Reranking introduces a two-stage retrieval paradigm that fundamentally transforms the RAG pipeline architecture. The enhanced workflow proceeds as follows:

- **Stage 1: Candidate Retrieval**: A computationally efficient retriever (typically embedding-

based or BM25) rapidly identifies a larger set of candidate documents (e.g., 25–100 documents) from the knowledge base, prioritizing recall over precision [119, 123].

- **Stage 2: Precision Reranking**: A sophisticated reranking model evaluates the semantic relevance of each candidate document relative to the query, producing refined relevance scores that enable reordering and filtering to select the most contextually appropriate documents (typically 3–5 documents) for generation [106, 131].

This architectural separation enables systems to balance computational efficiency with retrieval quality, as the expensive deep semantic analysis is applied only to a manageable subset of pre-filtered candidates rather than the entire corpus [34].

The primary objectives of reranking in RAG include:

- Improving retrieval precision by identifying and prioritizing the most relevant documents from the initial candidate pool [119, 131, 106]
- Filtering out irrelevant or misleading content that could negatively impact generation quality [131, 101]
- Enhancing answer accuracy by providing the generation model with higher-quality contextual information [131, 106, 101]

Evaluation Metrics for RAG

Retrieval Evaluation: NDCG@k The retrieval phase is typically evaluated using the Normalised Discounted Cumulative Gain at k (NDCG@k) metric [53, 28], where k can vary (e.g., 1, 10, 20). Here, k is the cutoff in the ranked list: only the top-k retrieved chunks contribute to DCG@k and NDCG@k (items ranked below k are ignored). In our scenario, for each query submitted to the RAG pipeline, k chunks are returned. The goal is for at least one of these chunks to contain the ground truth (when it is actually present) and for the most relevant chunks to be positioned at the top of the returned list. In this context, NDCG@k compares the ranking returned by the system with the ideal ranking, in which all relevant items occupy the highest positions.

Practically, k is chosen to match the number of chunks passed to the generator (top-k retrieval). Small k emphasizes early precision and lower latency; larger k increases recall but may dilute early ranking quality and raise context costs. Typical choices are $k \in \{1, 5, 10, 20\}$; for strict extractive QA one often reports NDCG@1, while for longer answers NDCG@10 or NDCG@20 is common. In two-stage retrieval (retrieve M candidates, then rerank), evaluation at k assumes $k \leq M$.

The NDCG@k calculation is obtained by dividing the DCG@k by the IDCG@k (Ideal DCG@k), which represents the maximum score obtainable by reordering the chunks in an ideal way for the query in question.

$$\text{NDCG@}k \;=\; \frac{\text{DCG@}k}{\text{IDCG@}k} \tag{2.8}$$

Let's take a closer look at the components of this formula.

- **k:** Indicates the number of items considered in the ranking (cutoff). If the returned list contains more than k items, only the first k are considered, both for the actual and ideal rankings.

- **Relevance score:** Each item in the list has a relevance score that reflects how important it is to appear at the top. In many practical cases, relevance is determined at runtime based on the ground truth associated with the query. For example, if a chunk contains the correct answer, it will receive a positive relevance score (e.g., 1), while irrelevant chunks will have a score of 0. In general, NDCG also allows multiple relevance scores (e.g., $0, 1, 2, \ldots$), so it is suitable for scenarios with different degrees of relevance.

- **Cumulative Gain (CG):** Cumulative gain is the sum of the relevance scores of the chunks in the top k places of the returned list:

$$\text{CG@}k = \sum_{i=1}^{k} G_i$$

  where G_i is the relevance score of the element in position i. However, CG does not take into account the order of the elements, so a list that contains all the relevant elements but places them at the bottom would have the same CG as one that puts them at the top.

- **Discounted Cumulative Gain (DCG):** To penalise relevant results that are found in low positions, each relevance score is divided by a logarithmic factor that increases with the position in the ranking:

$$\text{DCG@}k = \sum_{i=1}^{k} \frac{G_i}{\log_2(i+1)}$$

  The first element is not penalised (since $\log_2(1+1) = 1$), while the second is divided by approximately 1.585, and so on. In this way, the relevant results at the top contribute more to the total score.

  It is important to note that the absolute values of DCG depend on both the number of items evaluated and the distribution of relevance scores. As a result, DCG is not directly comparable between rankings of different lengths or relevance distributions.

- **Ideal DCG (IDCG):** This represents the maximum DCG obtainable for the same query and the same set of relevance scores, obtained by sorting all items in descending order of relevance.

- **Normalised DCG (NDCG):** NDCG is therefore the ratio between the DCG actually obtained and the ideal value:

$$\text{NDCG@}k = \frac{\text{DCG@}k}{\text{IDCG@}k}$$

The result is a value between 0 and 1, where 1 represents a perfect ranking (all relevant items are in the highest positions) and 0 represents the absence of relevance in the first k results. In general, the higher the NDCG, the better the quality of the ranking compared to the ground truth. Like any metric, NDCG has advantages and disadvantages. Among its advantages are:

- **Position sensitivity:** NDCG values relevant results positioned at the top, making it suitable for scenarios where the priority is to show the most important information immediately. This distinguishes it from metrics such as Precision and Recall, which do not consider order.

- **Flexibility on relevance scores:** It can handle binary or numerical relevance scores, making it suitable for both tasks with simple answers (presence/absence) and more nuanced scenarios with gradations of relevance.
- **Normalisation:** Normalisation with respect to IDCG allows for fair comparisons even between lists of different lengths or relevance distributions.

Meanwhile, the disadvantages:

- **Interpretability:** The use of the logarithmic factor makes the metric less intuitive than Precision or Recall, and intermediate values are not always immediately interpretable.

Generation phase: Accuracy In Chapter 6, the generation phase consists in providing the prompt with: (i) an initial section that explains the role of the model and its context, (ii) the query, and (iii) the retrieved chunks from which to extract the numerical answer, if present; otherwise, the model should return a string explicitly stating that the answer is not present in the provided context.

The performance of the model is measured using the Accuracy metric: for each evaluated row, a score of 1 is assigned if the ground truth equals the obtained result, or if both are null; otherwise a score of 0 is assigned. Predictions that differ from the ground truth by an absolute margin not exceeding 10^{-2} are treated as equal (for example, 10.01 and 9.99 are both considered equal to 10.00). The overall score is the arithmetic mean across all evaluated queries:

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{1}\{\hat{y}_i \approx y_i\} \tag{2.9}$$

where N is the number of evaluated queries, $\hat{y}_i$ is the predicted value, y_i is the ground truth, and $\mathbf{1}(\cdot)$ is the indicator function. The relation $\hat{y}_i \approx y_i$ holds when both values are exactly equal, both are null, or $|\hat{y}_i - y_i| \leq 10^{-2}$.

2.2.2 Related Work

Having established the conceptual and technical foundations of RAG systems, including their architecture, probabilistic formulation, and evaluation metrics, this subsection reviews existing datasets and benchmarks for financial question answering that are pertinent to retrieval-augmented generation. The subsequent analysis examines resources that pair questions with document-grounded answers and, where available, supporting evidence, drawn from financial disclosures. The comparison emphasises four axes: (i) regulatory and geographic scope, (ii) document modality and structure, (iii) unit of analysis, task formulation, and reasoning type, and (iv) licensing and openness. This framing contextualises the dataset released with this thesis, which targets cross-institution evaluation on European regulatory documents and addresses gaps identified in the existing benchmarking landscape outlined below.

Existing Financial QA Datasets Specialised benchmarks for financial question answering (QA) have advanced research in this domain. However, existing datasets each capture only a fragment of the broader landscape. The benchmark proposed is designed to complement these resources through practitioner-driven construction that reflects real financial workflows, while addressing

gaps in regulatory coverage, document diversity, and cross-institutional analysis. The following subsection reviews the principal financial QA datasets and situates the present contribution within the broader benchmarking ecosystem.

FinDER[19] marks a substantial step forward in financial QA benchmarking, providing 5,703 expert-annotated queries based on real-world financial search tasks. Designed to emulate the ambiguous and abbreviation-rich queries of finance professionals, FinDER challenges models to retrieve relevant information from large document collections. However, its scope is limited to U.S. public companies' annual reports. This constraint restricts generalisation to international or multi-regulatory banking contexts.

FinanceBench [51], comprises approximately 10,000 questions collected from filings of 40 U.S. public companies spanning the period 2015–2023. Each question is paired with an answer and the supporting context extracted from financial reports such as 10-K and 10-Q filings. Although comprehensive for single-company analysis, FinanceBench was not conceived for comparative studies across institutions, a critical capability for retrieval-augmented generation (RAG) systems employed in financial benchmarking and regulatory oversight. Furthermore, only a subset of 150 examples is publicly available, constraining its accessibility for open research.

Complementing these efforts, the **RAG Benchmark (Apple-10K-2022)** [65] provides a focused dataset constructed from Apple's 10-K SEC filings for 2022. It consists of 100 query–response pairs with accompanying contextual information, serving as a targeted testbed for evaluating RAG models on single-institution financial disclosures.

Two additional datasets, **TAT-QA** [144] and **FinQA** [17], extend the focus toward hybrid and numerical reasoning. TAT-QA contains approximately 2,800 hybrid contexts combining semi-structured tables and textual paragraphs, associated with roughly 16,500 financial questions. This design enables the evaluation of models that integrate textual and tabular data in complex reasoning tasks. FinQA, in turn, comprises around 2,800 financial reports yielding approximately 8,000 question–answer pairs, specifically targeting numerical reasoning over both structured and unstructured texts. Together, these datasets have been pivotal in advancing research on hybrid financial reasoning.

Building upon FinQA, **ConvFinQA** [18] introduces a conversational dimension to financial QA. It consists of 3,892 dialogues encompassing 14,115 questions, drawn from 10-K and 10-Q company reports. The dataset includes both simple conversations, generated by decomposing single multi-hop questions, and hybrid conversations, obtained by merging multiple reasoning chains. Each interaction was crafted by annotators with financial expertise, thereby enhancing realism and supporting the study of multi-turn reasoning in financial contexts.

Finally, **HC3 Finance** [42], a specialised subset of the broader HC3 dataset, evaluates model performance through direct comparison between human-generated answers and those produced by ChatGPT. This resource provides insights into the relative strengths and weaknesses of large language models when confronted with domain-specific financial questions. Table 2.4 presents a comparison of the dataset with existing financial and reasoning-oriented benchmarks, highlighting distinctive characteristics and contributions of each resource.

Dataset's Contribution Existing financial QA benchmarks exhibit several limitations: they predominantly focus on U.S. regulation, omit European banking standards such as Basel III Pillar 3 disclosures, and favour single-institution analysis rather than cross-bank comparisons, thereby reducing utility for analysts and regulators. Many benchmarks prioritise academic concerns over professional workflows, which diminishes ecological validity. The present benchmark addresses

Table 2.4: Comparison of financial and reasoning-oriented datasets used in retrieval and question answering tasks.

Dataset	**Description**	**Publication year**	**Number of elements**	**Open-source**
Ours	The dataset consists of 999 rows concerning financial indicators (2019–2023) with ground-truth values.	2026	999 lines, 209 documents	Y
FinDER	Expert-annotated dataset for retrieval-augmented generation in finance: query + evidence + answer triplets drawn from real-world financial queries with domain jargon, abbreviations, and ambiguous search behavior.	2025	5,703	Y
RAG benchmark (Apple-10K-2022)	Benchmark for Retrieval-Augmented Generation in finance: 100 query–response–context triples from Apple's 2022 10-K filing.	2024	100	Y
FinanceBench	Open-book financial QA benchmark: 10,231 questions about publicly traded companies with answers and supporting evidence.	2023	10,231	N (only 150 examples public)
TAT-QA	Tabular And Textual QA over hybrid contexts (text + tables) in financial reports.	2021	16,552 questions across 2,757 hybrid contexts	Y
FinQA	Numerical reasoning over financial reports combining structured tables and unstructured text with annotated reasoning programs.	2021	8,281 QA pairs across 2,789 financial reports	Y
ConvFinQA	Conversational financial QA with multi-turn dialogues requiring numerical reasoning over company reports.	2022	3,037 train / 421 dev / 434 test (conversation level) 11,104 train / 1,490 dev / 1,521 test (turn level)	Y
HC3 Finance	Human–ChatGPT comparison corpus in finance with paired human and ChatGPT answers to finance-related questions.	2023	48,644	Y

these shortcomings by incorporating 89 Pillar 3 reports and 120 annual reports from 24 European banks, enabling systematic, multi-institutional evaluation of RAG systems on regulatory compliance documents. Constructed in collaboration with financial practitioners, the benchmark targets ten standardised indicators (for example, the CET1 ratio and asset-class breakdowns).

Rather than supplanting existing resources, the benchmark complements prior work by filling a specific methodological gap and facilitates:

- cross-institutional analyses for competitive and regulatory benchmarking;
- assessment of regulatory compliance within the European banking context;
- systematic and reproducible extraction of standardised metrics for professional workflows.

This positioning establishes the benchmark as a substantive contribution to the research ecosystem for robust and comprehensive evaluation of RAG methods in financial question answering.

2.3 Mechanistic Interpretability

2.3.1 What is Mechanistic Interpretability

Definition 2.3.1 (Mechanistic Interpretability)**.** *The field of study of reverse engineering neural networks from the learned weights down to human-interpretable algorithms. Analogous to reverse engineering a compiled program binary back to source code [87].*

Mechanistic interpretability seeks to reverse engineer trained neural networks by identifying the internal features and circuits that implement task-relevant computations. It treats a model's parameters as an executable artefact that can be analysed to recover human-understandable algorithms, much as one would decompile a binary program back to source-level structure [87]. The basic units of analysis are features, that is directions or neurons encoding concepts, and circuits, that is connected sets of components whose interactions causally produce specific behaviours. Central phenomena include superposition, where more features are represented than dimensions, and polysemanticity, where units load on multiple unrelated features.

This approach differs from black-box explainability, which focuses on post hoc descriptions of outputs. Mechanistic interpretability aims at causal accounts of localised computation inside the network, asking which attention heads, MLP neurons, or pathways are necessary and sufficient for a behaviour. In practice, it combines descriptive probes with intervention-based tests to support falsifiable claims about how information is represented, routed, and read out.

The motivation in finance is practical as well as scientific. Regulated deployments require auditable evidence of how decisions arise, not only narratives about why a prediction might be plausible. Mechanistic analyses can expose sources of bias, clarify failure modes, and support targeted edits or controls to reduce model risk. Recent studies have begun extending these methods to financial applications, exploring how circuit-level analysis and attribution can improve transparency and compliance in large language models used by financial institutions [39, 121]. For example, Golgoon et al. (2024) present domain-specific methods for analysing large language models in financial services, demonstrating empirical techniques for revealing model internals relevant to auditing and risk assessment [39], while Tatsat & Shater (2025) critically examine the interpretability of LLMs on financial tasks, offering frameworks and evaluation criteria for moving beyond black-box analysis [121]. This aligns with governance needs for documented, verifiable mechanisms that can be tested and monitored over time [8].

Within the transparency stack adopted in this thesis, mechanistic interpretability provides causal transparency that complements empirical transparency from XAI and referential transparency from grounded generation. However, it is an emerging field: methods, tooling, and standards of evidence

are not yet mature, results can be brittle across prompts, datasets, and models, and many claims rely on toy tasks with uncertain external validity. The thesis therefore adopts a pragmatic objective to begin studying this area and to contribute initial, domain-relevant evidence by: (i) defining controlled tasks, metrics, and intervention protocols for component-level analysis, and (ii) conducting empirical case studies that test transfer from simplified circuits to realistic financial artefacts. The underlying premise is that, if mechanistic accounts scale and stabilise, they could deliver practically useful outcomes, including auditable causal explanations, targeted model edits, and stronger model risk controls. This chapter positions the work as a first step toward that goal, clarifying potential, limits, and requirements for maturity, while acknowledging open challenges such as superposition, robustness under distribution shift, and shared evaluation protocols for circuit claims.

2.3.2 Transformers as computational systems

A transformer can be viewed as a computational system that maintains a shared state, the residual stream, while a succession of components read this state and write incremental updates to it. This read–write perspective is central to mechanistic analysis: each layer decomposes into an attention sublayer that routes information between positions, and an MLP sublayer that applies local feature transformations, with both writing back to the same residual channel (see Figure 2.4). Recent mechanistic studies [82], demonstrate that the MLP acts as a local key-value store for factual knowledge: it processes a subject-specific 'key' to retrieve and inject the appropriate factual 'value' into the residual stream, enabling efficient and precise recall and modification of factual associations. Thus, the MLP is not merely a nonlinear processor, but also a locus for storing and editing knowledge, as knowledge tuples can be traced to and altered within distinct MLP modules. The final prediction is obtained by a linear readout (the unembedding), which allows component-wise contributions to be projected directly into vocabulary space.

The attention mechanism maps tokens to queries (Q), keys (K), and values (V) and computes context-dependent mixtures of value vectors. A standard scaled dot-product attention head is [127]

$$\mathrm{Attn}(Q, K, V) \;=\; \mathrm{softmax}\!\left(\frac{QK^{\top}}{\sqrt{d_{\mathrm{attn}}}}\right)V, \tag{2.10}$$

where d_{attn} is the query/key dimensionality. Multi-head attention runs several heads in parallel and concatenates their outputs before a learned projection.

Other core components include:

- **Token embeddings**: learned vector representations of vocabulary items (tokens).
- **Positional embeddings**: inject order information, e.g., sinusoidal encodings added to embeddings [127].
- **Add & Norm layers**: residual connections and layer normalization around sublayers are used to help with the vanishing gradient problem during training, and to make the training faster and more stable.
- **Unembedding**: maps hidden states back to a distribution over tokens.

An intuitive analogy, taken from McDougall [80], is helpful. Imagine a line of people, one per token position, each holding a token and maintaining a running note that represents the residual state. Each person tries to guess the token held by the person in front. Attention heads correspond to

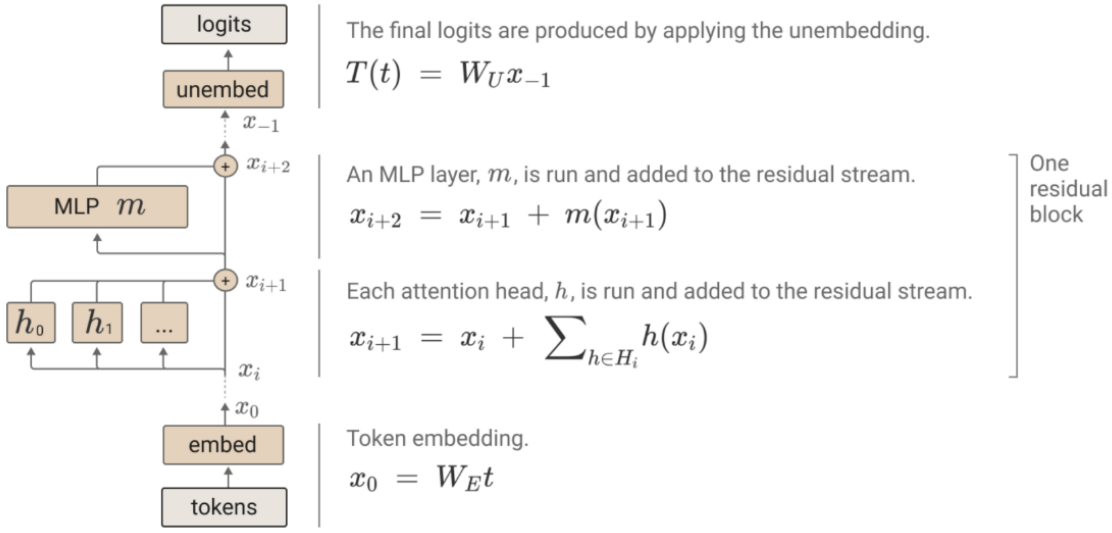


Figure 2.4: Transformer architecture visualised as a computational system with a shared residual stream. Figure from [25]

questions that any person may ask of everyone standing behind them: queries are the questions, keys determine who answers, and values determine what information is passed back to the original asker. The MLP represents the person's local processing, in which the current note is inspected, useful features are detected, and a further update is written (see Figure 2.5).

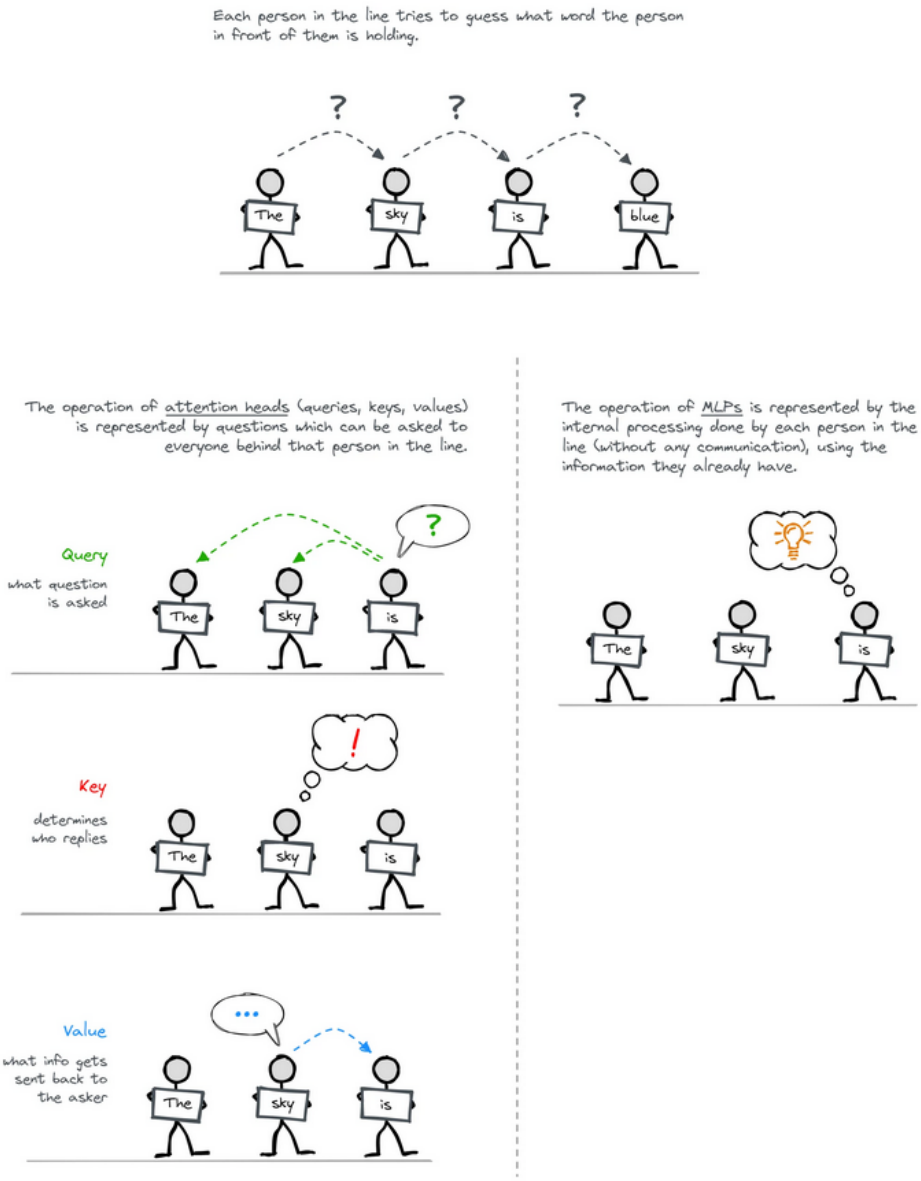


Figure 2.5: Analogy for transformer computation as read and write over a shared residual stream. Figure from [80].

Layer by layer, these edits accumulate in the residual stream. Since the final decision is a linear readout of the note, parts of the decision can be attributed to the specific edits that produced them. This operational picture supports the circuit-level analyses used in this thesis, including head-wise

attribution, activation patching, and path patching within and across layers.

Building on prior work [25], we posit that each attention head can be decomposed into two distinct circuits, as illustrated in Figure 2.6:

- The **Query-Key (QK) circuit** determines the routing of information, specifying where to move information to and from. This circuit answers the question of *where to look* in the input sequence. For a given attention head h, this mechanism is mathematically represented by the matrix product:
$$W_E^\top (W_Q^h)^\top W_K^h W_E,$$
where W_E denotes the token embedding matrix, and W_Q^h and W_K^h represent the query and key matrices, respectively.
- The **Output Value (OV) circuit** governs the content of the information being moved. This circuit determines *what is moved* to the residual stream. For the same attention head h, this circuit is captured by the matrix product:
$$W_U W_O^h W_V^h W_E,$$
where W_U is the unembedding matrix, W_O^h represents the output weights of the attention head, and W_V^h is the value matrix.

In this formulation, the embedding and unembedding processes are encapsulated by W_E and W_U, respectively, while the internal operations of the attention head are mediated by its query, key, value, and output matrices. A detailed analysis of these circuits, specifically for transformers with no more than two layers and without MLP sublayers, is provided in [25].

2.3.3 Circuit discovery

This subsection outlines principles and tools for discovering circuits in transformer models, with emphasis on controlled algorithmic tasks that support causal analysis [21]. Circuit formation has been used to account for grokking, the delayed transition from overfitting to generalisation after extended training, in small transformers trained on arithmetic, and to contextualise putative emergent abilities in larger language models [89, 47].

The approach followed here adopts the methodology of [130], which uses algorithmic tasks to elicit identifiable computations that carry over to applied settings. The focal task is Indirect Object Identification (IOI) evaluated on GPT-2 Small [107], a 12-layer model with approximately 80 million parameters. Given a sentence such as "When Mary and John went to the store, John gave a drink to ...", the correct continuation is "Mary", not "John". In [130], a computational subgraph comprising 26 attention heads is identified as responsible for the IOI behaviour. These heads are grouped into seven functional categories, including "duplicate token heads" that detect repeated names.

Synthetic IOI instances are generated from templated prompts [130, 129]; for example:

- "When [B] and [A] got a [OBJECT] at the [PLACE], [B] decided to give the [OBJECT] to [A]."

Names are sampled from a curated list of English first names; places and objects are drawn from hand-crafted sets of common terms. The model is evaluated on this prompted dataset, and its IOI circuitry is analysed mechanistically.

The discovery workflow combines contrastive readouts with causal interventions [81]:

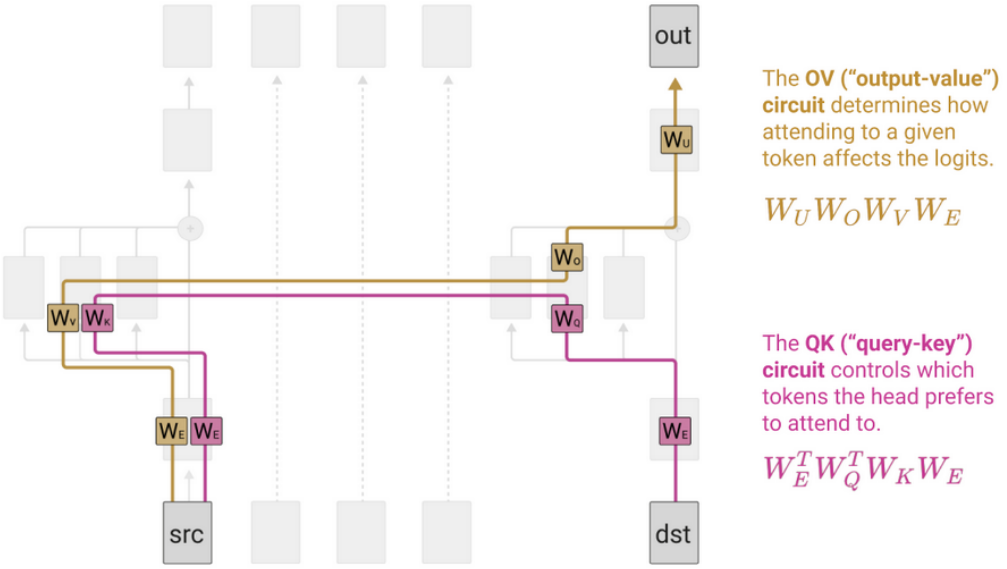


Figure 2.6: Figure from [25].

- **Contrastive logit metric.** Next-token probabilities arise from applying softmax to logits. A sensitive readout for IOI is the logit difference between the correct and contrast names, for example, logit(`Mary`) – logit(`John`).
- **Direct logit attribution.** To quantify how layers and heads write into the residual stream, one projects intermediate residuals into the unembedding and computes the logit difference "as if" subsequent layers were absent. This attributes contribution to specific components and is implemented in TransformerLens [88, 81].
- **Activation patching.** Two runs are constructed: a clean run that produces the correct answer, and a corrupted run that does not. In denoising, an activation from the clean run replaces the corresponding activation in the corrupted run; the improvement in the contrastive logit indicates causal importance (see Figure 2.7). In noising, the direction is reversed (corrupted into clean) to test necessity. Operationally, noising studies whether a component is necessary to maintain performance, whereas denoising studies whether it is sufficient to restore performance. Patching can target residual stream nodes (pre-, mid-, or post-layer), attention head outputs, MLP outputs, or decomposed head components (queries, keys, values, or attention patterns) [81, 82].
- **Path patching.** Rather than patching node activations, path patching intervenes on interactions between components. Treating nodes as attention heads and edges as couplings mediated by the residual stream, it replaces the signals carried along hypothesised head-to-head routes from source tokens to the answer position, thereby attributing behaviour to specific communication chains (see Figure **??**). For IOI, a common corruption removes duplicate-name evidence by randomising the agent names, for example, "When [X] and [Y] went to the store,

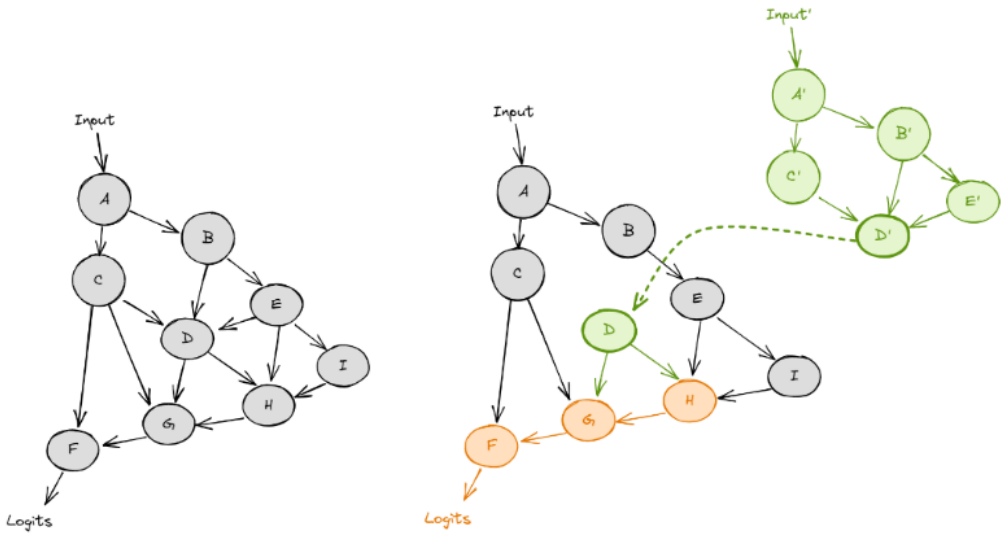


Figure 2.7: Activation patching: the clean run is captured on the left. On the right, an activation from the corrupted run is patched in the corresponding one from the clean run. Figure from [81].

> [Z] gave a drink to ...", while holding other tokens fixed. Restoring only a candidate path from the clean run tests whether that route is sufficient for correct behaviour [81].

To assess whether a discovered circuit is a reliable explanation for the model's behaviour, [130] proposes three criteria:

- **Faithfulness**: the extracted circuit achieves task performance comparable to the full model when run with appropriate scaffolding.
- **Completeness**: the circuit contains all nodes the model uses to perform the task.
- **Minimality**: all nodes in the circuit are relevant to the task.

These criteria guide the iterative process of proposing, testing, and refining circuits from attribution through to patching-based validation.

2.3.4 Superposition and polysemanticity

Superposition refers to the regime in which a model represents more than n distinct features using an n-dimensional activation space. Features continue to correspond to directions, but the set of interpretable directions is larger than the dimensionality of the space. This overcomplete set of directions is sometimes described as an overcomplete dictionary [25]. It is not a basis in the linear algebraic sense because it is not linearly independent. A practical intuition is that the network is compressing many meaningful features into fewer dimensions, thereby simulating a larger model with limited capacity.

An immediate consequence is that there does not exist a global, neuron-aligned basis in which every salient feature corresponds to a single unit. The features-as-neurons view cannot hold exactly

when superposition is present. Interference between features then becomes context dependent: the same neuron may contribute to different features in different inputs as the local superposition is resolved by the surrounding computation.

Neuron polysemanticity denotes the observation that a single neuron can respond to multiple seemingly unrelated features. Empirically, such neurons may activate on disparate clusters, for example, visual neurons responding both to images of dice and to portraits of poets [37]. Superposition implies polysemanticity because more features than neurons must share units. The converse does not necessarily hold: a model can exhibit polysemantic neurons while still admitting an interpretable feature basis that is rotated relative to the standard neuron basis. In such cases, polysemanticity arises from using a non-standard, but interpretable, set of directions rather than from capacity-driven superposition.

For analysis, these phenomena motivate methods that operate on directions and circuits rather than on single neurons, and they encourage the search for sparser, more monosemantic representations when possible. In the present work, claims at neuron level are treated with caution, and directional or path-wise attributions are preferred when assessing causal roles within circuits.

> *In my (extremely biased!) opinion, mech interp is a very exciting subfield of alignment. Currently our models are inscrutable black boxes! If we can really understand what models are thinking, and why they do what they do, then I feel much happier about living in a world with human level and beyond models, and it seems far easier to align them.*
>
> — Neel Nanda, Google DeepMind [87]

2.4 Unifying Framework

In summary, this chapter has outlined the theoretical foundations underpinning this thesis. Explainable AI provides measurable interpretability at the model-output level; Grounded Generation extends transparency to generative reasoning; and Mechanistic Interpretability anchors these perspectives in the causal structure of the models themselves. The following chapters build upon this background to present novel contributions within each of these dimensions, uniting them under a single objective: to make AI systems in the financial domain not only powerful but also comprehensible, verifiable, and trustworthy.

Chapter 3

Explainability, Quantified: Benchmarking XAI techniques

This chapter builds on the study "Explainability, Quantified: Benchmarking XAI Techniques" [103], conducted in collaboration with Alan Perotti, Claudio Borile, Francesco Paolo Nerini, and André Panisson, CENTAI Institute (Center for Artificial Intelligence), Turin, Italy; and Paolo Baracco, Anti Financial Crime Digital Hub, Turin, Italy. The work was presented at the International Conference on Explainable Artificial Intelligence (XAI), 2024, and published as a chapter in "Explainable Artificial Intelligence" (Springer Nature Switzerland, pp. 421–444). Minor adaptations and extensions have been made to ensure coherence with the overall thesis.

This chapter addresses the first pillar of the transparency stack, namely Explainable AI, by operationalising explanation quality through a small set of task-agnostic metrics and a large-scale benchmark. The goal is to move explanation assessment from ad hoc, method-specific claims to measurable criteria that support auditing, comparison, and deployment decisions in financial settings.

The chapter formulates four requirements that explanations should satisfy in practice: faithfulness to the model under explanation, parsimony compatible with human scrutiny, algorithmic reproducibility, and computational tractability. These are instantiated as Effective Compactness, Rank Quality Index, and Stability. The metrics are designed to compare heterogeneous explanation families, and to quantify trade-offs that practitioners face when selecting methods for production pipelines.

A controlled experimental protocol evaluates these metrics across tabular classification, regression, and anomaly detection. The study spans widely used models, including gradient-boosted trees, multi-layer perceptrons, isolation forests, one-class support vector machines, and autoencoders. It covers attribution, counterfactual, and rule-based explainers, with deletion-curve-based evaluation and repeated trials to characterise variance. The resulting grid yields a comprehensive, model- and task-level comparison that reflects real operational constraints.

The contributions are threefold. First, a minimal, implementable metric set is proposed that is applicable across explanation types and predictive tasks. Second, an extensive benchmark is provided that exposes consistent trade-offs between accuracy, parsimony, stability, and latency,

thereby informing explainer choice under regulatory and service constraints. Third, the analysis establishes a quantitative foundation for the subsequent chapters: grounded generation extends output transparency through evidential linkage, while mechanistic interpretability advances causal transparency by probing internal circuits. Together, these chapters build the case for transparent and reliable AI in finance.

The remainder of the chapter introduces the metrics, describes the benchmark design, reports the empirical results, and discusses implications for model auditing and deployment.

3.1 Metrics for XAI

In this section, three metrics are defined that form a minimal yet complete framework for quantitative explanation evaluation, covering the main properties that are desirable for an effective explanation method: adherence to the inner workings of the explained black-box, comprehensibility for the human being, and algorithmic reproducibility.

While the notion of evaluating explanation quality has been widely discussed in the XAI literature, there is no shared or standardized set of quantitative metrics applicable across explanation types and machine learning tasks. The metrics introduced in this section combine both original contributions and principled adaptations of existing ideas. Where relevant, connections to prior work are explicitly discussed, and the specific novelty of each metric is clarified.

After introducing the property that each metric captures, a description of how they are tailored to different ML tasks and for different explanation types is provided; the fourth criterion considered in the experimental section is execution time, but due to its trivial implementation, it is not discussed further.

Effective Compactness (EC) measures the length and complexity of the produced explanations. While some explainers can generate succinct explanations involving one or two features, others might produce explanations that involve dozens of elements. Clearly, the explanation's compactness correlates with the cognitive load required for a human to understand it. The concept of compactness is often suggested in the literature and fairly easy to measure; for instance, by computing the number of features with non-zero attributions or the number of features with modified values in a counterfactual. However, the compactness of an explanation should be paired with its impact on the black-box: if an explanation highlights only a few features, but altering them does not change the model's output, then the amount of information included in an explanation that is able to impact the outcome of the black-box model for the ML task at hand should be measured.

Rank Quality Index (RQI) refers to the ability of the explainer to capture the internal decision process of the black-box ML model. The commonly adopted metrics of fidelity/faithfulness are conceptually very similar, but the literature does not offer a standard formulation of this metric. Furthermore, some fidelity definitions are quite narrow in scope; for instance, the definition of Zhang et al. [139] can be applied only to explainers based on surrogate models. Instead, the focus is placed on the explanation-induced feature ranking, measuring its impact on the model's behaviour compared to random baselines.

Stability (STB) refers to the ability of the explainer to always reproduce the same result, given the same model and input. It is therefore a proxy for algorithmic reproducibility: can a single explanation be trusted, or should the explainer be queried over and over in order to average out some fluctuations in the output? Some explainers are deterministic, while others use stochastic sampling or approximations and therefore might produce noisy explanations.

3.1.1 Background values and Deletion Curves

In the absence of a given ground truth for direct comparison, one reasonable way to assess the quality of an attribution-based explanation, it can be useful, for a specific data-point and its relative local explanation, to check if the removal of features with high scores causes a sensible change in the output of the model, while the removal of features with low scores has almost no impact on the model's output. Despite the simplicity and effectiveness of this approach, the removal of features is not possible once the ML algorithm is trained. Some explainers exploit a workaround to overcome this limitation: instead of removing a feature, its actual value is replaced by a 'default' one, called **background value** (or, sometimes, baseline value). SHAP [75], for instance, is based on this concept, and a background dataset has to be explicitly provided. Background values are typically representative values (such as medians or centroids) or obtained by sampling. It must be noted that the specific choice of background values can have a great impact and must be chosen carefully, and in particular it depends on the ML task at hand; different scenarios are defined below. Background values are used in conjunction with another fundamental construct of this work: deletion curves.

The **deletion curve**, introduced first in [104], is defined as the curve obtained by deleting (or perturbing) the features of the input data point, one by one according to their attribution importance, and measuring for each new input the change in output of the predictive model. As stated before, the concept of feature removal is implemented as a replacement with a background value. This process is sometimes called masking.

As a simple guided example, consider a binary classification task of loan approval (yes/no) on tabular data with four features: name, age, income, and hobby. Consider a data-point $DP = [John, 38, \$40.000, rugby]$ and a trained black-box model f, which outputs a probabilistic estimation of the positive class, such that $f(DP) = 0.9$. For this example, let a background data-point be $BG = [Mark, 60, \$50.000, golf]$, with $f(BG) = 0.2$. Suppose that an explanation for the verdict $f(DP) = 0.9$ is the set of attributions name:0, age:0.1, income:0.4, hobby:0.05. The induced (decreasing) order is $income > age > hobby > name$, and for each step, a progressively masked data-point will be created. In each case, the black-box model f will be queried, thus creating the transition from $f(DP) = 0.9$ to $f(BG) = 0.2$. The progressively masked data-points are displayed in Table 3.1 and Figure 3.1. It is clear that if an explanation is able to highlight relevant features, the deletion curve will show an immediate steep drop, since few maskings are enough to cause the model to change output. Conversely, a sub-optimal explanation that (wrongly) highlights irrelevant features will produce a deletion curve which starts flat and only decreases later on. It is noted that both curves would have $f(DP)$ as the starting point and $f(BG)$ as the ending point. The area under the deletion curve (Area Under the Deletion Curve (AUDC)) can be used as a proxy for attribution-based explanation quality: better explanations produce quickly dropping curves with small AUDC.

3.1.2 Effective Compactness (EC)

Effective Compactness EC) is introduced as a novel metric in this work. While the general concept of compactness has been informally discussed in prior literature, typically as the number of features involved in an explanation, EC is the first metric, to our knowledge, that explicitly couples compactness with causal impact on the black-box model through interventional deletion curves. EC measures the amount of information, included in an explanation, that is actually able to impact the outcome of the black-box model. The nature of the amount of information depends on the explanation type, while impacting the outcome of a black-box depends on the ML task at hand. The EC metric satisfies the following properties:

	DP	step1	step2	step3	step4=BG
name	John	John	John	John	**Mark**
age	38	38	**60**	60	60
income	$40K	**$60K**	$60K	$60K	$60K
sport	rugby	rugby	rugby	**golf**	golf
loan?	0.9	0.4	0.4	0.25	0.2

Table 3.1: Deletion curve (tabular)

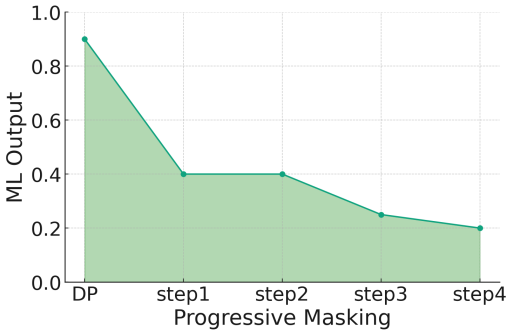


Figure 3.1: Deletion curve (visual)

1. Its minimum value is 1, as at least one interventional step on the data-point is necessary to impact the black-box.

2. Similarly, its maximum value is the feature number K: in the worst case, not even acting on all features changes the output of the black-box.

3. Lower values are to be preferred: an explanation with low EC is able to correctly identify the minimum set of features that drive a ML model's outcome.

EC for attributions

When dealing with attribution-based explanations, the implementation of EC is based on deletion curves. In all cases, EC corresponds to the number of features that need to be masked in order to impact the black-box.

For binary classification tasks, the two BG values are the medoids of the two classes. For instance, in the loan example, each loan=yes data-point will be explained using the loan=no medoid as BG, and vice versa. This guarantees that the deletion curve always crosses the decision boundary, as its starting and final data-points have opposite labels. The EC in this case corresponds to the number of replacements that are required in order to cross the decision boundary.

This framing can be easily extended to multi-class classification, where for each data-point the EC score is the minimum value across all other classes. For instance, if a data-points belongs to class C_i, for each class C_j $(j \neq i)$ the medoid will be selected as BG_j, and the score EC_j computed as described above; the lowest EC_j will be selected as EC. Intuitively, this corresponds to the minimum number of changes which is required for the model to change its prediction about the data-point from the current class to any other class.

For regression on tabular data, thresholds of one standard deviation above and below the predicted values are defined, and the resulting partition of the datasets is treated as a multiclass classification. Note that the number of classes might be three (for data-points where the predicted value is not extreme) or two (for instance, if the predicted value is high and there are no other points with a predicted value which is a standard deviation above). This is treated as a default value for the threshold parameter, but this can be set to a domain-dependent value if needed.

For anomaly detection tasks, EC is computed following a similar logic, with one caveat: if the data-point is classified as anomalous, the medoid of the 'normal' class will be used as BG, and only

one deletion curve will be computed. Conversely, if the data-point is classified as normal, every anomaly will be treated as the medoid-BG of itself, and for each case, a different deletion curve will be computed; as for the multiclass scenario, the lowest value will be selected as EC.

EC for counterfactuals

For counterfactual explanations, it is verified whether the counterfactual belongs to an admissible class. If this test succeeds, the EC is given by the number of modified features in the proposed counterfactual; if the test fails, and the counterfactual candidate is not actually a counter-factual, the number of features is assigned as the EC score, as a default 'failure' value.

EC for rules

When the explanation is provided as rules, it is checked whether the data-point satisfies all rules. As with the counterfactual case above, if the test fails, the number of features is assigned as the EC score. If the set of rules is valid, the number of rule antecedents is assigned as the EC value.

3.1.3 Rank Quality Index (RQI)

Rank Quality Index (RQI) builds on the general notion of fidelity or faithfulness introduced in previous work, but proposes a task-agnostic and explainer-agnostic formulation. Unlike existing fidelity metrics, which are often limited to surrogate-based explainers or specific model classes, RQI introduces a comparative framework based on random baselines, making it applicable to attributions, counterfactuals, and rule-based explanations. RQI refers to the ability of the explainer to effectively capture the inner workings of the model to be explained, when compared with random alternatives. In all its possible implementations, the RQI metric satisfies these properties:

1. It is bounded in $[0, 1]$.
2. Higher values are preferable.

The main idea is a systematic comparison of the performance of the explanation for a considered data-point with what would be obtained by choosing random explanations. How to compare them and the choice of random explanations vary depending on the specific explanation type.

RQI for attributions

For attributions, the RQI implementation is based on deletion curves. The deletion curve induced by the explanation to be evaluated is computed. Several random deletion curves are also created, based on randomly generated feature rankings. It is recalled that a good property of deletion curves is to have a small AUDC. A failure is considered whenever a point of a random deletion curve falls below the "real" deletion curve under scrutiny. The RQI is computed as the percentage of successes. For instance, in Figure 3.2, in three cases (red dots) the random explanations produced deletion curves that went below the main deletion curve. The RQI is $9/12 = 0.75$.

A perfect deletion curve would be below any random curve, thus obtaining a maximal RQI of 1. Conversely, the worst possible explanation induces a deletion curve which is always above all random ones and corresponds to a RQI of 0.

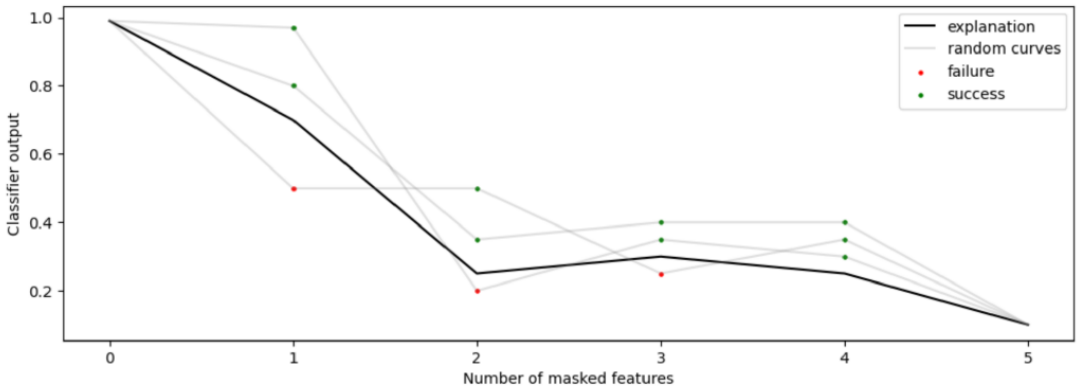


Figure 3.2: Deletion curves, successes and failures.

RQI for counterfactuals

In this scenario, deletion curves are not used. The edit distance between two data-points can be informally defined as the number of features whose values differ; let D be the edit distance between the original data-point and the counterfactual produced as explanation.

A number of data-points belonging to another class (with respect to the original data-point) are sampled, and data-points with an edit distance greater than D are considered successes. Conversely, sampled data-points with smaller distances are considered failures. As for the previous case, the RQI is computed as the $success/(failure+success)$ ratio. The intuition here is to measure whether the explainer is able to produce counterfactuals that are closer to the initial data-point with respect to other existing data-points.

RQI for rules

One crucial limitation of explanations as rules is that they are inherently binary in assigning importances to features. That is, either a feature is involved in the explanation or not, and consequently it appears that only the features appearing in the rule are important while all the others are completely irrelevant. For this reason, given the explanation provided by the explainer, the values of the features involved in the explanation are clamped, and the values of others are substituted with those of randomly chosen data-points. The RQI score is given by the ratio of generated data-points that keep the same predicted label as the original data-point over all generated data-points. The rationale behind this is that if the rule really captures the relevant features, then changing the remaining features should have no impact in the black-box prediction.

3.1.4 Stability (STB)

Stability, as defined in this work, is an operational metric that formalizes the intuitive notion of explanation reproducibility discussed in prior literature. While the importance of stable explanations has been previously acknowledged, this thesis provides a concrete and unified quantitative definition of stability across different explanation types, including attributions, counterfactuals,

and rules. Given a data-point, stability is defined as the complement to one of the weighted variability around the reference values (e.g., mean) of the normalised explanations obtained by calling the explainer multiple times for the same data-point. The stability metric satisfies the following properties:

1. It is bounded in $[0, 1]$.
2. Higher values are preferable.
3. Deterministic explainers have a stability equal to 1.

It can be noted that the definition above still requires some clarification for terms such as "variability" and "reference". These operational definitions depend on the input data and explanation type.

STB for attributions

To measure stability, the same data-point (table row) is explained multiple times, yielding a scoring matrix; for statistical relevance, the experiment is repeated across numerous data-points, resulting in a score tensor A of dimension $N \times D \times F$, where N represents the number of trials for each explanation, D is the number of data-points, and F denotes the dimensionality of the feature vector. First the tensor score is normalised across data-points

$$\tilde{A} = \frac{A_x}{\max(A_x)} \quad \in \mathbb{R}^{N \times F} \quad \forall x \tag{3.1}$$

Then, the main quantities are computed; that is, the mean of the absolute value and the standard deviation of the normalised scores across trials

$$M = \langle |\tilde{A}| \rangle_{tr}, \quad \Sigma = \sigma_{tr}(\tilde{A}) \tag{3.2}$$

The instability score for each data-point is given by the weighted mean of the standard deviations across features

$$IS_x = \frac{1}{F} \sum_i M_i \cdot \Sigma_i^T \tag{3.3}$$

Finally, the stability score for attributions on tabular data is given by the complement to one of the average of IS_x over all data-point:

$$\text{stability} = 1 - \langle IS \rangle_x \quad \in [0, 1] \tag{3.4}$$

STB for counterfactuals

While feature attributions typically involve numerical scores, counterfactual explanations are data-points that may include categorical variables.

For numerical features, the instability is computed using the same procedure as for attributions, while for categorical variables a notion of variability is used [2], where frequency histograms f are first computed for each feature across trials, and the instability is given by

$$IS_x^c = 1 - \|f\|_x \,, \tag{3.5}$$

where the apex c refers to categorical features. At this point, the stability score for counterfactuals on tabular data is computed as

$$\text{stability} = 1 - \langle IS \rangle_x \quad \in [0, 1] \tag{3.6}$$

STB for rules

For rules on tabular data, the stability score is given by computing the complement to one of the normalized standard deviations of the length of the provided rules.

In summary, Effective Compactness represents a novel contribution, while Rank Quality Index and Stability provide principled generalizations and formalizations of existing evaluation concepts. Together, these metrics form a minimal yet comprehensive framework for quantitative explanation assessment across heterogeneous XAI methods.

3.2 Experimental Setup

In this section, an overview of the experimental setup used to evaluate the explainers is provided. Each experiment was conducted over a specific valorisation of dataset, model, and explainer (plus the metrics). The focus is on three ML tasks on tabular data: binary classification, regression, and anomaly detection.

Datasets that were publicly available and provided an overall heterogeneous variety of features and data-points were preferred. Datasets exhibiting categorical features were also favoured. As further criteria, datasets with no missing values and at least one hundred points were chosen. The datasets were randomly split with a seed into training (70% of the dataset) and testing (30%) sets. During training, the training split was further divided through a random stratified split into training and validation splits to mitigate issues due to class imbalance. Five datasets were selected for each task:

- Classification: adult [9], credit [135], eye [113], german [44], heloc[1]
- Regression: abalone [90], bike [30], california [57], diabetes [24], diamonds [40]
- Anomaly Detection: AID362 [100], bank [100], chess [100], glass [35], U2R [100]

To select the models, the popularity of model types in each task and the requirements of common model-aware explainers, such as TreeSHAP and Integrated Gradients (IG), were considered, resulting in the following models:

- Binary classification: MLP (multi-layer perceptron) and Light Gradient Boosting Machine (LGBM)(tree-based, gradient-boosting model).
- Regression: MLP and LGBM with regression heads.
- Anomaly Detection: Isolation Forest (IF), one-class Support Vector Machine (Support Vector Machine (SVM)) and Autoencoder (AE).

Each model was trained on all five datasets selected for the task, and the results were compared to baseline performances. Model performance was evaluated against reference baselines reported in prior benchmark studies and publicly available implementations, which served as reference points to verify that the trained models achieved competitive predictive performance before the explainability analysis was conducted. A summary of task performance across datasets and model types

[1] `https://community.fico.com/s/explainable-machine-learning-challenge`

is provided in Tables 3.2, 3.3, and 3.4. The final number of models to explain was 35 (15 models for anomaly detection, 10 models each for binary classification and regression). A hyperparameter optimisation was performed for each of the models, using a k-fold cross-validation strategy on the training data. Several explainers were applied to the trained models, and the quality of their explanations was evaluated through the metrics. If the explainer had any parameter to be selected, the default or suggested setting was followed, unless otherwise specified in the following. For every dataset-model-explainer tuple, 100 points of the test set were explained, chosen at random regardless of the model prediction. The computational time was tracked, and the EC and RQI of each explanation were measured. Each explanation was then repeated 100 times to evaluate the stability. Taking these multiplicative factors into account, the final complete benchmarking grid encompassed roughly 1.6 million explanations, with four metric values measured in each case.

3.2.1 Binary classification

Datasets

The selected classification datasets cover a range of standard financial and behavioral prediction tasks. The Adult dataset concerns income prediction from census data, while German and Credit address credit risk and default prediction using demographic and financial attributes. The Eye dataset focuses on eye movement classification, providing a non-financial but feature-rich benchmark with categorical variables. HELOC models credit risk using real-world credit bureau data and is the only dataset composed exclusively of numerical features.

Table 3.2: Datasets for binary classification. Types of features correspond to (C) categorical, (I) integer, (R) real.

Dataset	**Target**	**class percentages**	**number of points**	**number of features**	**types of features**	**LGBM F1-Score**	**MLP F1-Score**
adult	Census prediction	76%-24%	50,000	14	C, I	0.7117	0.6709
credit	Credit card default	50%-50%	13,272	21	C, R	0.6865	0.6769
eye	Eye movements	50%-50%	7,608	23	C, N	0.6412	0.6053
german	Credit risk	70%-30%	1,000	20	C, I	0.6057	0.5424
heloc	Credit risk	48%-52%	10,500	23	I	0.7487	0.7261

ML Models

For both models, a randomised grid-search is performed using the scikit-learn API (Application Programming Interface (API)). For the LightGBM models, the number of leaves, maximum depth of the trees, learning rate and loss regularisation terms are varied; while for the MLP, the number of hidden layers, the number of hidden channels and the batch size are considered, along with the learning rate and the weight decay regularisation. The MLPs are trained for a maximum number of 300 epochs with an early stopping criterion if the loss does not change more than 10^{-4} for 10

epochs. The models which minimise the F1-score in training are selected. The models are then retrained with the full training data, and the F1-score is measured on a test set, as presented in Table 3.2. No preprocessing is applied for the LightGBM models, while for the MLP a one-hot encoding is performed for the categorical variables and a quantile normalisation for the numerical variables.

The performances over the datasets "eye" and "credit" can be compared with the results shown in Grinsztajn et al., 2022 [40], a benchmark of various models over tabular datasets with categorical features, from which these datasets were selected. The "adult" and "german" datasets are instead collected from the UCI ML Repository website,[2] where baseline performances are reported. Results on the "heloc" dataset can be compared with those obtained by the winners of the original challenge in which the dataset was proposed.[6]
Overall, the accuracy of the models is comparable with all the baselines considered.
Benchmark values from these external sources are not reproduced as additional columns in the summary tables, since doing so would unnecessarily increase their complexity and hinder readability. Instead, they are used as external reference points to validate that the trained models attain competitive predictive performance prior to the explainability analysis.

Explainers

KernelSHAP [75], Local Interpretable Model-agnostic Explanations (LIME) [110], Anchors [111] and MACE [134] are applied to both MLP and LightGBM. For LightGBM, TreeSHAP [76] is also used, while for MLP, Deconvolution [137], DeepSHAP [15], Integrated-Gradients [120], InputXGradient [116], Occlusion [137], and Saliency [117] are included. This selection of explainers allows the production of explanations in the shape of rules (Anchors), counterfactuals (MACE), and attributions (all the others); thus, several implementations of the metrics are leveraged.

In general, algorithms such as SHAP or LIME rely on sampling from the training dataset, either to construct background datasets used in deletion curves or to fit local surrogate models. Consequently, the proposed metrics are indirectly affected by both the size and the quality of these sampled sets. If the sample is too small, estimates of background values and local feature effects become noisy, which in turn produces higher-variance deletion curves and can inflate or deflate Effective Compactness and Rank Quality Index, as well as slightly lowering Stability for sampling-based explainers. If the sample is large but unrepresentative (for instance, due to class imbalance or covariate shift), the background distribution no longer reflects the operational data, and the metrics may systematically favour or penalise specific explainers in a way that reflects sampling artefacts rather than genuine explanation quality. To mitigate these effects, stratified sampling from the training data was adopted wherever possible, and the number and construction of background samples were kept fixed within each experimental block so that metric comparisons remain meaningful across explainers.

3.2.2 Regression

Datasets

The regression datasets span diverse predictive tasks and data characteristics. Abalone and Diamonds involve value estimation from mixed categorical and numerical features, while Bike models demand forecasting from temporal and contextual variables. California focuses on house price

[2] `https://archive.ics.uci.edu/`

prediction from purely numerical census features, and Diabetes predicts disease progression from clinical measurements.

The datasets "abalone", "bike" and "diamonds" are also taken from Grinsztajn et al., 2022 [40], enabling direct comparison with their reported results. The "california" and "diabetes" datasets are instead widely employed as examples due to their inclusion in the package scikit-learn: model performance is compared with the tutorials provided by SHAP[3] and Captum[4], along with the performances obtained by users on Kaggle. All these datasets, except for "california", have categorical and numerical features as well.

ML models

As in the classification case, an analogous randomised grid search was performed over the same hyperparameters, optimising the root mean square error (Root Mean Square Error (RMSE)) instead. The resulting models achieved performance consistent with published baselines, and the test set R^2 values are reported in Table 3.3. For the MLP models used for regression, the same preprocessing as before was followed: a one-hot encoding for the categorical variables and a quantile normalisation for the numerical variables.

Table 3.3: Datasets for regression. Types of features correspond to (C) categorical, (I) integer, (R) real.

Dataset	**Target**	**number of points**	**number of features**	**types of features**	**LGBM R^2**	**MLP R^2**
abalone	Abalone age	4,177	8	C, I	0.5417	0.5541
bike	Bike sharing demand	17,379	11	C, N	0.9461	0.9404
california	House value	20,640	8	R	0.8437	0.8122
diabetes	Disease progression	442	10	I, R	0.5285	0.4428
diamonds	Diamonds price	53,940	23	C, N	0.9927	0.9907

Explainers

For this task, only attribution-based explainers are used: SHAP kernels (KernelSHAP [75], TreeSHAP [76], DeepSHAP [15]), LIME [110], and Integrated-Gradients [120]. The default synthetic thresholds (one standard deviation above or below the regression output of each data-point) are used as a surrogate of the decision boundary.

[3] https://shap.readthedocs.io/en/latest/example_notebooks/tabular_examples/model_agnostic/Diabetes%20regression.html

[4] https://captum.ai/tutorials/House_Prices_Regression_Interpret

3.2.3 Anomaly Detection

Datasets

The anomaly detection datasets represent heterogeneous real-world scenarios with varying anomaly rates. AID362 and Bank involve behavioral and transactional anomalies, Chess models rare game configurations, and U2R focuses on network intrusion detection. Glass provides a small numerical benchmark for material classification with a comparatively higher anomaly ratio.

All of the datasets, with the exception of "glass", are coming from ADRepository. The performance of these datasets can be compared with the state-of-the-art performances reported in Pang et al., 2021 [99]. The glass dataset is instead collected from UCI ML Repository and the performance can be compared with the baseline provided on the platform and with the isolation forest used for the experiments in Carletti et al., 2023 [13]. With the exception of "glass", all the dataset presents categorical features, with "bank" and "U2R" providing no numerical features.

ML models

In this case, a lighter grid search is performed for the hyperparameters of the isolation forests and the one-class support vector machines by comparing the area under the Receiver Operating Characteristic, Area Under the Curve (ROC AUC). For the isolation forests, the number of estimators and the maximum number of samples for each tree are varied. For the one-class support vector machine, the Radial Basis Function (RBF) kernel is selected and only the gamma factor is varied.

In the case of the autoencoders, the optimization is performed with respect to the reconstruction loss of the training set. Couples of encoders and decoders of 4 layers each (3 in the case of the glass and chess datasets) are used, and the bottleneck dimension, the batch size, the learning rate, the weight decay regularization and the number of epochs are randomly varied. The isolation forests require no preprocessing, while a scaling is performed for numerical variables and a one-hot encoding for categorical ones in both one-class support vector machines and autoencoders. Performance is assessed using the ROC AUC on the test sets, as reported in Table 3.4. Isolation forests and autoencoders achieve results in line with published baselines, while one-class support vector machines systematically underperform.

Table 3.4: Datasets for anomaly detection. Types of features correspond to (C) categorical, (I) integer, (R) real.

Dataset	**Target**	**number of points**	**number of anomalies (% to total)**	**number of features**	**types of features**	**IF ROC AUC**	**SVM ROC AUC**	**AE ROC AUC**
AID362	Chemical inactivity	4,279	60 (1.4%)	114	C, I	0.6627	0.5365	0.6227
bank	Deposit subscription	41,188	4640 (11%)	10	C	0.6063	0.6102	0.6632
chess	Game result	28,056	27 (0.1%)	6	C, I	0.9435	0.5535	0.8212
glass	Glass recognition	214	51 (23%)	9	R	0.9320	0.7347	0.7693
U2R	Network attack	60,821	228 (0.4%)	6	C	0.9808	0.8834	0.9646

Explainers

Since the number of anomalies is not always large enough, a set of 50 inliers and the maximum number of available outliers in the test set, up to 50 other points, are explained for each dataset. Specifically, 100 points were evaluated in total for bank and U2R, 74 for AID362, 65 for glass and 55 for chess.

There are not many explainers for anomaly detection, and several openly available repositories had to be discarded due to a lack of maintenance or general malfunctions. Three task-specific explainers were selected: ACE [141], DIFFI [13], and COIN [69]. KernelSHAP and LIME were also included as model agnostic explainers. Both are explainers for supervised ML models, while anomaly detection is an unsupervised task. However, a trained anomaly detector behaves like a regressor, as it predicts a continuous-valued anomaly score, and then proceeds to classify a data-point as anomalous if the anomaly score is over a threshold. Leveraging this, KernelSHAP and LIME can be applied on all three anomaly detection models. The metrics are measured as if the task is a multiclass classification, with one normal class and up to 50 anomalous classes.

3.2.4 Preprocessing and attributions

One of the most common approaches for treating categorical variables in neural networks is one-hot encoding, which is also adopted in this study. This preprocessing is non-differentiable and causes gradient-based explainers to lose information about the mapping from the encoded variables to the original features. In general, for some explainers, there is no principled way to aggregate the encoded features to obtain attributions for the original features. The most commonly used approach of summing the attributions of the encoded features derived from a single categorical feature in the original table is used here. This choice is coherent with the properties of DeepSHAP and Integrated Gradients, which allow the summation of the attributions of a subset of variables to obtain their aggregated importance.

3.3 Results

The results of the benchmark are presented in this section. Different tables are displayed for each combination of metric and task: Tables 3.5 3.6 3.7 for effective compactness; Tables 3.8 3.9 3.10 for rank quality index; Tables 3.11 3.12 3.13 for stability; Tables 3.14 3.15 3.16 for time.

For effective compactness and Rank Quality Index (RQI), all metric entries report the mean and the standard deviation across the data-points explained for each dataset. For stability (STB), the metric itself is a standard deviation across data-points and trials. For time, the average time to produce one explanation is reported. In all tables, for each combination of model and dataset, the best explainer(s) are highlighted in bold, and the second best explainer(s) with an underline. It is indicated when explainers produce counterfactual (cf) or rules (r) as explanations.

COIN proved to be extremely slow, providing results in a reasonable amount of time for the experiments only on the glass dataset (as can be noted in Table 3.16). In other cases, producing a single explanation required up to several hours. The 100 COIN trials were therefore not performed for the other datasets (AID362, bank, chess and U2R); as a result, standard deviation errors are not reported, and stability could not be computed; these cases are highlighted with an asterisk (*) in the tables. ACE, instead, often produced NaN values as explanations using the original implementation. The decision was made to reduce the default learning rate of the method from

0.01 to 0.0005, while the loss regularization parameter "beta" of the KL term was reduced from 50 to 1. By doing so, explanations were produced in all scenarios, with one exception: one-class SVM for dataset bank. This special case is reported in the tables with "NE" (no-explanation).

The first observation that arises from the results is that no single explainer consistently dominates across all metrics, datasets, and model types. While certain methods perform well on specific combinations, none achieves uniformly superior scores for effective compactness, rank quality index,

Table 3.5: Effective Compactness scores for the classification task.

Model	Explainer	adult	credit	eye	german	heloc
lgbm	Anchors (r)	3.5 ± 2.6	2.8 ± 2.0	10.3 ± 4.5	4.4 ± 3.1	3.0 ± 1.0
	MACE (cf)	**1.5 ± 0.8**	1.8 ± 1.4	**1.9 ± 1.2**	**1.9 ± 1.2**	2.3 ± 1.4
	LIME	4.5 ± 4.5	1.6 ± 1.3	2.6 ± 2.4	3.5 ± 3.2	2.5 ± 2.5
	KernelSHAP	4.2 ± 3.7	1.5 ± 1.2	1.9 ± 1.4	3.0 ± 2.7	**1.9 ± 1.2**
	TreeSHAP	5.0 ± 4.0	**1.4 ± 1.1**	2.0 ± 1.7	3.5 ± 3.4	2.2 ± 1.7
mlp	Anchors (r)	3.4 ± 2.0	3.4 ± 2.6	3.6 ± 1.7	4.5 ± 2.3	2.6 ± 0.7
	MACE (cf)	**1.8 ± 1.0**	**2.2 ± 1.8**	**2.0 ± 1.7**	**1.9 ± 1.1**	3.0 ± 1.6
	LIME	3.5 ± 3.1	2.6 ± 2.3	3.8 ± 5.2	6.2 ± 6.1	2.7 ± 1.7
	KernelSHAP	2.4 ± 1.4	2.2 ± 1.9	2.1 ± 1.4	2.5 ± 2.0	**2.3 ± 1.3**
	DeepSHAP	2.3 ± 1.2	2.8 ± 3.6	2.2 ± 1.4	2.5 ± 2.4	2.4 ± 1.4
	Integrated-Gradients	3.5 ± 3.2	4.5 ± 4.2	2.8 ± 3.2	2.2 ± 2.0	2.5 ± 1.5
	Saliency	7.7 ± 4.0	6.9 ± 5.6	4.7 ± 4.8	5.5 ± 5.0	3.6 ± 2.5
	InputXGradient	4.4 ± 3.8	4.8 ± 4.8	3.9 ± 5.4	2.1 ± 1.6	2.6 ± 1.7
	Deconvolution	3.8 ± 3.2	6.6 ± 4.7	13.0 ± 7.0	9.1 ± 6.3	12.5 ± 8.2
	Occlusion	3.9 ± 3.5	5.2 ± 5.4	3.2 ± 4.6	2.2 ± 2.2	2.6 ± 1.7

Table 3.6: Effective Compactness scores for the regression task.

Model	Explainer	abalone	bike	california	diabetes	diamonds
lgbm	LIME	2.1 ± 2.3	**3.5 ± 3.9**	3.3 ± 2.7	**3.4 ± 3.5**	3.2 ± 3.1
	KernelSHAP	**2.1 ± 2.2**	3.7 ± 4.0	**3.2 ± 2.7**	3.6 ± 3.4	**3.0 ± 3.0**
	TreeSHAP	**2.1 ± 2.2**	3.7 ± 4.0	**3.2 ± 2.7**	3.6 ± 3.5	**3.0 ± 3.0**
mlp	LIME	**1.2 ± 0.9**	3.9 ± 3.8	2.5 ± 2.4	3.7 ± 3.2	3.0 ± 2.6
	KernelSHAP	1.2 ± 1.0	**3.5 ± 3.7**	2.4 ± 2.3	3.6 ± 3.4	2.8 ± 2.5
	DeepSHAP	**1.2 ± 0.9**	**3.5 ± 3.7**	2.5 ± 2.4	**3.5 ± 3.4**	**2.8 ± 2.4**
	Integrated-Gradients	1.3 ± 0.8	3.7 ± 3.9	**2.2 ± 2.1**	3.7 ± 3.4	2.8 ± 2.5

Table 3.7: Effective Compactness scores for the anomaly detection task.

Model	Explainer	glass	AID362	bank	chess	U2R
if	LIME	3.3 ± 2.2	31.5 ± 18.0	**2.5 ± 2.9**	2.5 ± 1.2	2.2 ± 1.4
	KernelSHAP	3.2 ± 2.3	**14.6 ± 11.2**	**2.5 ± 2.9**	**2.2 ± 1.1**	**1.2 ± 0.7**
	ACE	2.7 ± 1.1	56.0 ± 13.0	3.6 ± 1.4	4.5 ± 0.9	2.9 ± 0.8
	DIFFI	**2.5 ± 1.1**	48.5 ± 15.5	2.7 ± 1.3	3.7 ± 1.2	2.5 ± 0.6
	COIN	2.5 ± 1.2	52.4*	2.0*	5.0*	2.0*
svm	LIME	2.9 ± 2.8	**11.7 ± 33.1**	**2.8 ± 3.1**	**1.3 ± 1.0**	**4.1 ± 2.4**
	KernelSHAP	3.2 ± 3.4	**11.7 ± 33.1**	**2.8 ± 3.1**	1.3 ± 1.1	**4.1 ± 2.4**
	ACE	**1.7 ± 1.4**	53.4 ± 48.1	NE	2.0 ± 1.3	4.7 ± 1.4
	COIN	2.2 ± 1.9	11.0*	3.0*	4.0*	4.0*
ae	LIME	2.7 ± 3.2	26.0 ± 19.5	2.9 ± 2.9	1.4 ± 1.1	4.1 ± 1.1
	KernelSHAP	2.8 ± 3.3	**13.3 ± 20.2**	**2.7 ± 2.8**	**1.3 ± 0.8**	4.4 ± 0.7
	ACE	**2.0 ± 2.0**	31.9 ± 22.4	2.9 ± 2.0	3.5 ± 1.5	**3.6 ± 1.6**
	COIN	2.1 ± 1.6	50.2*	5.0*	4.0*	3.0*

Table 3.8: Rank Quality Index scores for the classification task.

Model	Explainer	adult	credit	eye	german	heloc
lgbm	Anchors (r)	0.93 ± 0.08	0.85 ± 0.12	0.90 ± 0.12	0.93 ± 0.06	0.81 ± 0.15
	MACE (cf)	**0.99 ± 0.04**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.01**
	LIME	0.80 ± 0.19	0.92 ± 0.06	0.78 ± 0.12	0.84 ± 0.09	0.90 ± 0.11
	KernelSHAP	0.81 ± 0.13	0.92 ± 0.08	0.84 ± 0.12	0.88 ± 0.08	0.96 ± 0.06
	TreeSHAP	0.78 ± 0.15	0.93 ± 0.07	0.85 ± 0.11	0.80 ± 0.10	0.94 ± 0.07
mlp	Anchors (r)	0.92 ± 0.10	0.81 ± 0.12	0.93 ± 0.08	0.90 ± 0.04	0.80 ± 0.13
	MACE (cf)	**0.99 ± 0.03**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.01**
	LIME	0.82 ± 0.12	0.89 ± 0.07	0.87 ± 0.14	0.71 ± 0.17	0.93 ± 0.05
	KernelSHAP	0.93 ± 0.07	0.88 ± 0.07	0.93 ± 0.06	0.92 ± 0.08	0.94 ± 0.04
	DeepSHAP	0.92 ± 0.08	0.87 ± 0.07	0.92 ± 0.06	0.92 ± 0.09	0.94 ± 0.05
	Integrated-Gradients	0.82 ± 0.19	0.78 ± 0.14	0.86 ± 0.14	0.88 ± 0.09	0.95 ± 0.05
	Saliency	0.50 ± 0.17	0.63 ± 0.20	0.67 ± 0.17	0.64 ± 0.17	0.82 ± 0.15
	InputXGradient	0.75 ± 0.19	0.75 ± 0.12	0.85 ± 0.16	0.86 ± 0.09	0.93 ± 0.06
	Deconvolution	0.73 ± 0.18	0.72 ± 0.16	0.39 ± 0.20	0.56 ± 0.19	0.50 ± 0.27
	Occlusion	0.76 ± 0.19	0.76 ± 0.12	0.87 ± 0.14	0.87 ± 0.09	0.92 ± 0.05

Table 3.9: Rank Quality Index scores for the regression task.

Model	Explainer	abalone	bike	california	diabetes	diamonds
lgbm	LIME	**0.96 ± 0.08**	0.90 ± 0.16	0.96 ± 0.07	**0.98 ± 0.03**	**0.97 ± 0.05**
	KernelSHAP	0.94 ± 0.08	**0.92 ± 0.11**	**0.96 ± 0.06**	0.97 ± 0.04	0.97 ± 0.07
	TreeSHAP	0.94 ± 0.08	**0.92 ± 0.11**	**0.96 ± 0.06**	**0.98 ± 0.03**	0.97 ± 0.07
mlp	LIME	0.94 ± 0.07	0.85 ± 0.16	0.91 ± 0.11	0.88 ± 0.14	0.95 ± 0.07
	KernelSHAP	0.95 ± 0.07	0.89 ± 0.13	**0.92 ± 0.09**	**0.97 ± 0.04**	**0.97 ± 0.04**
	DeepSHAP	**0.95 ± 0.06**	**0.93 ± 0.09**	0.91 ± 0.10	0.96 ± 0.05	**0.97 ± 0.04**
	Integrated-Gradients	0.92 ± 0.08	0.90 ± 0.12	0.88 ± 0.13	0.90 ± 0.13	0.94 ± 0.10

Table 3.10: Rank Quality Index scores for the anomaly detection task.

Model	Explainer	glass	AID362	bank	chess	U2R
if	LIME	0.72 ± 0.33	**0.98 ± 0.11**	**0.87 ± 0.29**	0.93 ± 0.12	0.90 ± 0.11
	KernelSHAP	0.74 ± 0.35	**0.98 ± 0.11**	0.87 ± 0.30	**0.95 ± 0.12**	**0.97 ± 0.09**
	ACE	0.80 ± 0.15	0.57 ± 0.18	0.76 ± 0.15	0.64 ± 0.18	0.88 ± 0.13
	DIFFI	0.85 ± 0.11	0.68 ± 0.20	0.83 ± 0.11	0.83 ± 0.11	0.95 ± 0.05
	COIN	**0.89 ± 0.12**	0.74*	0.83*	0.46*	0.97*
svm	LIME	0.71 ± 0.34	**0.95 ± 0.14**	**0.85 ± 0.32**	**0.95 ± 0.19**	0.62 ± 0.31
	KernelSHAP	0.71 ± 0.41	**0.95 ± 0.14**	**0.85 ± 0.32**	0.94 ± 0.20	**0.63 ± 0.30**
	ACE	**0.81 ± 0.14**	0.80 ± 0.22	NE	0.89 ± 0.18	0.61 ± 0.25
	COIN	0.80 ± 0.19	0.99*	0.83*	0.88*	0.81*
ae	LIME	0.74 ± 0.32	0.88 ± 0.18	0.80 ± 0.31	0.86 ± 0.14	**0.65 ± 0.23**
	KernelSHAP	**0.75 ± 0.37**	**0.95 ± 0.18**	**0.81 ± 0.34**	**0.96 ± 0.06**	0.62 ± 0.16
	ACE	0.74 ± 0.14	0.84 ± 0.14	0.79 ± 0.21	0.76 ± 0.13	0.62 ± 0.23
	COIN	0.71 ± 0.12	0.76*	0.92*	0.73*	0.80*

stability, and execution time simultaneously. For instance, SHAP-based methods tend to excel in stability and faithfulness but may incur higher computational costs, whereas gradient-based explainers offer speed advantages but often underperform on rank quality. This variability supports the claim that explainers should be selected based on the specific requirements of the task at hand (e.g., favouring effective compactness at the expense of time, or prioritising stability for regulatory compliance). The only notable exception is MACE, which very often outperforms all the other

Table 3.11: Stability scores for the classification task.

Model	Explainer	adult	credit	eye	german	heloc
lgbm	Anchors (r)	0.891	0.983	0.910	0.893	0.944
	MACE (cf)	0.968	0.973	0.968	0.945	0.991
	LIME	**0.995**	0.987	0.960	0.983	0.988
	KernelSHAP	0.985	**0.994**	**0.991**	0.992	**0.994**
	TreeSHAP	0.988	**0.994**	**0.991**	**0.993**	0.993
mlp	Anchors (r)	0.891	0.965	0.953	0.896	0.952
	MACE (cf)	0.958	0.973	0.976	0.927	0.982
	LIME	0.997	0.988	0.992	0.990	0.986
	KernelSHAP	0.985	0.993	0.993	0.991	0.993
	DeepSHAP	0.985	0.993	0.993	0.990	0.991
	Integrated-Gradients	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
	Saliency	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
	InputXGradient	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
	Deconvolution	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
	Occlusion	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**

Table 3.12: Stability scores for the regression task.

Model	Explainer	abalone	bike	california	diabetes	diamonds
lgbm	LIME	0.981	0.983	0.980	0.987	0.987
	KernelSHAP	**0.984**	**0.989**	**0.986**	0.984	**0.989**
	TreeSHAP	**0.984**	**0.989**	**0.986**	**0.990**	**0.989**
mlp	LIME	0.977	0.981	0.981	0.996	0.983
	KernelSHAP	0.982	0.990	0.985	0.984	0.984
	DeepSHAP	0.981	0.988	0.983	0.982	0.985
	Integrated-Gradients	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**

Table 3.13: Stability scores for the anomaly detection task.

Model	Explainer	glass	AID362	bank	chess	U2R
if	LIME	0.985	0.974	0.988	0.957	0.957
	KernelSHAP	0.982	0.993	0.974	0.976	0.988
	ACE	0.964	0.994	0.96	0.947	0.968
	DIFFI	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
	COIN	0.995	*	*	*	*
svm	LIME	0.951	0.991	**0.988**	0.985	0.980
	KernelSHAP	0.975	**0.999**	0.980	0.976	0.984
	ACE	0.960	0.993	NE	**0.987**	**0.988**
	COIN	**0.990**	*	*	*	*
ae	LIME	**0.990**	0.972	**0.983**	0.969	**0.981**
	KernelSHAP	0.980	**0.993**	0.974	**0.970**	0.977
	ACE	0.960	0.992	0.954	0.943	0.960
	COIN	0.993	*	*	*	*

classification explainers both in effective compactness and rank quality index. It lacks however in stability and time, as it is often slower and more affected by stochasticity than the attributions methods.

LIME and SHAP both perform well and, except for the classification task, are also very often comparable. In fact, for classification, SHAP tend to provide better results in all the metrics. This is especially true when accounting for its model-aware implementations: for example, KernelSHAP

Table 3.14: Average execution time (in seconds) for the classification task.

Model	Explainer	adult	credit	eye	german	heloc
lgbm	Anchors (r)	3.8201	9.3388	21.1223	3.6820	10.2143
	MACE (cf)	0.1998	0.4982	0.5157	0.3681	0.5595
	LIME	0.0112	0.0148	0.0179	0.0133	0.0217
	KernelSHAP	0.8346	0.7873	1.3785	0.7840	1.2840
	TreeSHAP	**0.0032**	**0.0020**	**0.0140**	**0.0012**	**0.0076**
mlp	Anchors (r)	22.5568	26.1702	17.7036	7.7750	21.6247
	MACE (cf)	1.2197	1.1235	1.2713	0.6018	1.2504
	LIME	0.2364	0.0517	0.0251	0.0186	0.0324
	KernelSHAP	55.5335	8.8678	1.8452	1.3800	1.5163
	DeepSHAP	0.1250	0.0190	0.0081	0.0026	0.0057
	Integrated-Gradients	0.0261	0.0033	0.0008	0.0007	0.0007
	Saliency	**0.0006**	**0.0002**	**0.0001**	**0.0001**	**0.0001**
	InputXGradient	**0.0006**	**0.0002**	**0.0001**	**0.0001**	**0.0001**
	Deconvolution	0.0007	**0.0002**	**0.0001**	**0.0001**	**0.0001**
	Occlusion	0.0245	0.0008	0.0003	0.0003	0.0003

Table 3.15: Average execution time (in seconds) for the regression task.

Model	Explainer	abalone	bike	california	diabetes	diamonds
lgbm	LIME	0.0080	0.0111	0.0114	0.0099	0.0118
	KernelSHAP	0.1579	1.3664	0.1824	0.4810	0.4089
	TreeSHAP	**0.0024**	**0.0092**	**0.0060**	**0.0016**	**0.0166**
mlp	LIME	0.0154	0.0131	0.0033	0.0257	0.3026
	KernelSHAP	0.1421	1.2096	0.7558	2.0956	14.4160
	DeepSHAP	0.0051	0.0042	0.0146	0.0127	0.1333
	Integrated-Gradients	**0.0011**	**0.0008**	**0.0033**	**0.0018**	**0.0323**

Table 3.16: Average execution time (in seconds) for the anomaly detection task.

Model	Explainer	glass	AID362	bank	chess	U2R
if	LIME	0.4676	0.6545	0.4546	0.7465	0.4116
	KernelSHAP	4.3165	88.4451	10.7356	1.8215	0.2653
	ACE	0.3786	0.3056	**0.1860**	**0.4686**	0.4764
	DIFFI	**0.0633**	**0.1794**	1.3023	5.7222	**0.0928**
	COIN	4.2136	65.7728*	5517.0394*	2479.7800*	18873.1220*
svm	LIME	**0.0120**	**0.0194**	**0.3588**	**0.0573**	0.0344
	KernelSHAP	0.2841	1.3515	32.4892	0.1525	**0.0200**
	ACE	0.2785	0.1034	NE	0.8294	49.1368
	COIN	8.0503	68.8571*	5406.9440*	2659.0604*	15503.0285*
ae	LIME	**0.0346**	**0.1176**	**0.0466**	**0.0503**	0.0503
	KernelSHAP	0.8784	56.5848	2.8444	0.0994	**0.0399**
	ACE	392.9724	0.2699	0.1159	0.1502	0.1243
	COIN	6.4690	68.3407*	5335.9604*	2439.5188*	18740.9937*

is usually slower than LIME in providing explanations, but DeepSHAP and TreeSHAP provide similar performances with reduced time requirements.

Indeed, the variants of SHAP tend to perform similarly. In the regression task, TreeSHAP and KernelSHAP provide the same explanations. This is also due to the "interventional" default setting of TreeSHAP, which makes its behaviour closer to the original implementation of SHAP. The main, important difference between the two in this case is that the model-aware implementation is

significantly faster in providing explanations.

In general, model-aware and model-agnostic explainers are not so different in the quality of their explanations. Saliency and Deconvolution are often underperforming compared to other methods, but they were not originally conceived for tabular datasets. This is also true for the other model-aware explainers for neural networks, but they provide overall reasonable performances, although rarely better than other explainers. They are also especially valuable for providing stable and significantly faster explanations, since, except DeepSHAP, they do not rely on any randomization. Model-agnostic are often easier to include in a ML pipeline, since they can access a ML model's API and not its code; since this additional requirement of model-aware explainers is not balanced by highly-scoring explanation, model-agnostic explainers seems to be at an advantage.

The time required for producing an explanation, often overlooked in benchmarking approaches, can vary significantly, up to entirely different orders of magnitude, across different dataset-model-explainer combinations. This constitutes a crucial point in the explainer selection process, as a slow explainer could congest online services and render an explainable pipeline unusable from a user interaction perspective.

3.4 Discussion

This chapter has established a quantitative framework for evaluating explanation quality in machine learning models, addressing the empirical transparency required for trustworthy financial AI. By introducing a minimal set of metrics—Effective Compactness, Rank Quality Index, and Stability—and applying them across a diverse benchmark of models and tasks, the work transforms explainability from an aspirational property into a measurable standard. The results demonstrate that explanation quality is inherently multidimensional, with trade-offs between faithfulness, parsimony, reproducibility, and computational cost that must be balanced according to operational and regulatory constraints.

Within the broader thesis narrative, these findings form the empirical foundation of the transparency stack outlined in the introduction. The metrics and benchmarking approach provide practitioners and auditors with tools to assess and compare explainers in a principled manner, supporting the deployment of interpretable AI in high-stakes financial environments.

The adoption of standardised evaluation metrics is crucial in advancing the field of XAI, especially in light of the newly approved European AI ACT [5]. It is recommended that the assessment of newly proposed explainers against established benchmarking frameworks and metrics becomes a common practice.

A pragmatic recommendation is to integrate the XAI component into the ML development lifecycle from an early stage. The following guidelines offer a structured approach for practitioners:

1. **Define the problem scope.** Specify the ML task (classification, regression, or anomaly detection) and input data type (tabular, image, or text). If the model architecture is predetermined, note this constraint. Identify whether local or global explanations are required and whether particular explanation types (attributions, counterfactuals, or rules) are mandated by stakeholders or regulations.

2. **Survey available explainers.** Consult benchmark results (such as those presented in this chapter) to identify candidate explainers compatible with the task and model class. Prioritise

[5] https://digital-strategy.ec.europa.eu/en/policies/regulatory-framework-ai

methods that satisfy operational constraints: for instance, latency-sensitive applications may exclude slow sampling-based explainers, whereas audit requirements may favour high-stability methods.

3. **Finalise model and explainer selection.** If the model was not fixed in step 1, select an architecture that admits high-scoring explainers. Where multiple explainers are viable, retain at least two to enable cross-validation of explanations during development.

4. **Review technical requirements.** Examine installation dependencies, input format expectations (e.g., handling of categorical variables, one-hot encoding conventions), and API compatibility. Address known pitfalls early: for example, some explainers require models to expose specific methods or to be wrapped in particular classes.

5. **Configure a unified environment.** Establish a Python environment that satisfies the dependencies of both the ML model and the selected explainer(s). Version conflicts between XAI libraries and deep learning frameworks are common and should be resolved before training.

6. **Train the model.** Develop and train the ML model in accordance with the constraints identified in step 4. Ensure that any preprocessing (e.g., scaling, encoding) is applied consistently during both training and explanation generation.

7. **Generate explanations.** Produce explanations for a representative sample of data points, including both correctly and incorrectly predicted instances, as well as edge cases relevant to the domain.

8. **Evaluate explanation quality.** Apply the proposed metrics (Effective Compactness, Rank Quality Index, Stability) to the generated explanations. Compare scores against benchmark values to verify that the explainer performs as expected on the specific dataset and model.

9. **Use explanations for debugging.** Treat explanation quality as a diagnostic signal. Investigate discrepancies between benchmark and observed scores, or systematic differences between explanations for correct versus misclassified predictions, to identify potential issues in the model or data pipeline.

10. **Integrate explanations into production.** Once validated, produce explanations alongside ML predictions as standard practice. Document the explainer configuration and metric scores for auditability and regulatory compliance.

Finally, work is underway to extend the metrics to other data types: images, text, and graphs. While also in this case most explainers produce attribution scores, the challenge is to extend the metrics to encompass peculiar explanation types, such as subgraphs and walks for graphs, and prototypes for images.

This empirical layer is complemented in subsequent chapters by referential transparency, where the focus shifts from explanation of predictions to the traceability of generated outputs. Chapter 4 builds on this foundation by developing grounded generation methods for annotating and retrieving evidence from visually rich financial documents, thereby linking model outputs to verifiable sources and advancing the transparency stack towards comprehensive, auditable AI reasoning.

Chapter 4

Keyword-based Annotation of Visually-Rich Document Content for Trend and Risk Analysis using Large Language Models

This chapter builds on the study "Keyword-based Annotation of Visually-Rich Document Content for Trend and Risk Analysis Using Large Language Models" [33], conducted in collaboration with Giuseppe Gallipoli, Simone Papicchio, Lorenzo Vaiani, Luca Cagliero, Politecnico di Torino, Turin, Italy; and Daniele Borghi, Intesa Sanpaolo Innovation Center, Turin, Italy. The work was presented at the Joint Workshop of the 7th Financial Technology and Natural Language Processing (FinNLP), the 5th Knowledge Discovery from Unstructured Data in Financial Services (KDF), and the 4th Workshop on Economics and Natural Language Processing (ECONLP), Turin, Italy, May 2024, and published in the Proceedings. Minor adaptations and extensions have been introduced for coherence within this thesis.

This chapter develops the second pillar of the transparency stack, namely grounded generation, in the context of visually rich financial documents. The objective is to transform heterogeneous pages of text, tables, and images into structured, auditable annotations that make subsequent retrieval and analysis traceable. In the thesis narrative, this work supplies the referential layer that links model outputs to verifiable evidence, preparing the ground for retrieval-augmented pipelines introduced later in Part II.

Methodologically, the chapter presents a practical pipeline for keyword-based annotation under real operational constraints. Given a facet of interest, it generates candidate keywords and short descriptions, segments documents into textual paragraphs, tables, and images, and produces textual equivalents where direct extraction is unreliable. Elements and keyword descriptions are embedded for semantic comparison, and keywords are assigned by unsupervised similarity, with optional large language model prompting for re-ranking. Care is taken to handle layout variability and duplication, and to preserve provenance so that each annotation remains inspectable.

The empirical study evaluates the pipeline on bilingual, in-domain collections curated with Intesa

Sanpaolo and Intesa Sanpaolo Innovation Center. Annotation quality is assessed with information retrieval metrics, and description quality is assessed both automatically and through human expert judgement. The analysis compares open-source and proprietary model families in zero-shot and few-shot settings. Proprietary embeddings yield the strongest semantic matching; open-source models are competitive for keyword generation; GPT-4 produces higher quality descriptions. These findings inform design choices for reliable, traceable annotation in production workflows.

Within the broader thesis, the annotated corpora and the lessons from this study support the construction of Retrieval-Augmented Generation systems that meet referential transparency requirements. The next chapters build on this foundation by strengthening evidence attribution, retrieval, and summarisation, and by evaluating answer faithfulness on financial question answering tasks.

4.1 Background and motivation

Understanding and exploring the content of visually-rich documents, such as Portable Document Format (PDF) files and scanned documents, is of primary importance for trend and risk analysts in the banking and finance sectors. As these documents exhibit variable layouts and content, comprising text, images, and tables, their thorough interpretation requires advanced multimodal learning capabilities.

This study aims to enhance the research and analysis capabilities of a primary Italian financial institution, with a focus on emerging trends within national and international contexts. Strengthening these functions is crucial for the bank's strategic positioning and for delivering value-added services to customers. Partial automation of the research process enables the inclusion of a broader range of data sources that were previously untapped due to operational constraints. Given the relentless flow of information in today's environment, this constitutes a strategic step towards expanded informational access and a stronger ability to adapt proactively to market evolution.

A financial document annotator is introduced for bank analysts, relying on multimodal Large Language Models (LLMs). Given a topic of interest (hereafter denoted by *facet*), a list of facet-related keywords is generated, together with the corresponding textual descriptions and high-dimensional vector representations. In parallel, multimodal document content is partitioned into textual paragraphs, images, and tabular elements, and processed to obtain embeddings of their textual equivalents. The annotation process is then treated as a keyword retrieval task over document elements, driven by textual semantic similarity. An extensive empirical analysis, supported by a bilingual testing collection and expert validation of keyword descriptions, provides an in-depth performance comparison between the open-source Meta AI Llama2 and the proprietary OpenAI GPT-4 models.

4.2 Problem statement

Given a set of multi-page financial documents $\mathcal{D}$ and a set of facets $\mathcal{F}$ describing the topics of interest, the objectives are threefold:

1. **Keyword generation and description**. Generate for each facet $f_i \in \mathcal{F}$ a set of keywords $k_j \in \mathcal{K}^i$ related to f_i. Next, annotate each keyword k_j with a free-text description $descr(k_j)$ summarising its general meaning.

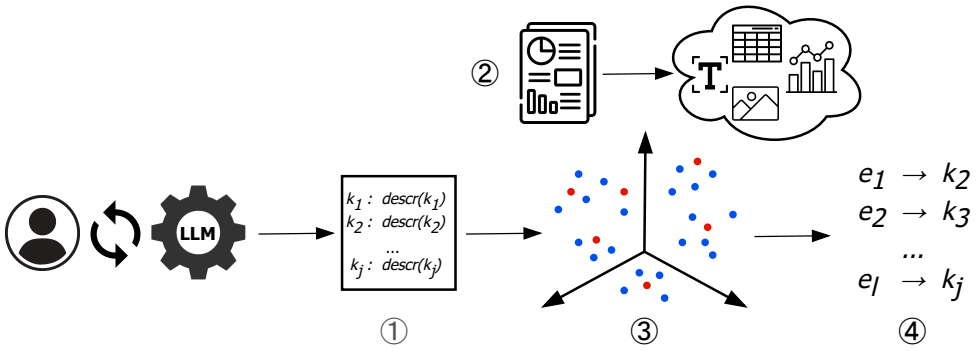


Figure 4.1: The figure illustrates the main steps of the proposed method: (1) keyword and description generation; (2) document preprocessing; (3) document element and keyword description encoding; and (4) keyword-based content annotation. In step (3), blue and red dots represent the embedding representations of document elements and keyword descriptions, respectively.

2. **Captioning of non-textual document elements**. Produce textual descriptions of multimedia document elements $e_l \in \mathcal{E}^m$, where an arbitrary element e_l in a document $d_m \in \mathcal{D}$ can be either an image, a table, or a textual paragraph.

3. **Keyword-based content annotation**. For each element e_l, retrieve the keywords k_j that are most relevant to e_l.

The goal is to compare the performance of LLMs, in zero-shot or few-shot learning, to address all the above-mentioned tasks. Hereafter, Llama2 [125] (Large Language Model Meta AI (Llama)) or its Italian version Camoscio [114]) is considered as the representative open-source model and GPT-4 [94] (Generative Pretrained Transformer (GPT)) as the representative proprietary model.

4.3 Proposed approach

The main steps of the method are described below. A sketch of the proposed pipeline is displayed in Figure 4.1.

4.3.1 Generation of keywords and keyword descriptions

Given a user-provided facet name f_i, the LLM is used to automatically generate a set of related keywords k_j as well as the corresponding free-text descriptions $descr(k_j)$.

The following settings are explored:

- *Zero-Shot learning – Cold Start Setting*: The LLM is prompted with the facet name only. It is assumed that neither facet-relevant keywords nor examples of textual descriptions is available.

- *Few-Shot learning – Cold Start Setting*: The LLM is prompted with the facet name and h examples chosen randomly from keywords and their corresponding descriptions, previously provided by the domain expert. In this case, some examples of keyword descriptions are

available, but no facet-related keyword is necessarily known, since the selected examples are not required to be related to the input facet.

- *Few-shot learning – Additional Keyword Recommendation*: The LLM is prompted with the facet name and h examples of facet-related keywords and their corresponding descriptions. In this case, the examples are not chosen randomly but shortlisted by a human expert (e.g., by validating a previous output).

In few-shot learning settings, the input examples are chosen so that they do not overlap with the keyword currently being prompted.

The output of this step is then used in the keyword-based content annotation stage.

4.3.2 Document pre-processing

To process the input PDF documents, three main element types are extracted: (i) textual paragraphs (e.g., titles, sections, subsections), (ii) visual items (e.g., images, sketches of architectures/processes/pipelines, iconography, graphical examples), and (iii) tables.

Textual paragraphs and tables are extracted from PDF documents using the proprietary Document Intelligence service provided by the Azure AI platform [5]. For visual and textual content extraction, the following challenges arise:

- *Slide extraction*: Some input documents consist of slide presentations, which are unsuitable for text and image extraction using standard content extraction tools. To address this issue, textual explanations of slide content are generated opportunistically using the Multimodal Large Language Model GPT-4 Vision [94]. Specifically, an ad hoc Convolutional Neural Network (CNN) is trained to automatically detect the presence of presentation slides on a PDF document page. If a page is classified as a *slide*, the input is processed directly by the Multimodal LLM.

- *Paragraph length*: Some extracted textual elements contain few words, likely due to misalignment of PDF content. To mitigate this, textual elements consisting of fewer than four words are discarded.

- *Redundant table content*: The textual content within table cells sometimes appears incorrectly twice, in separate tabular and textual elements. During table extraction, potential overlaps between the table bounding box and text positions are detected early. Duplicated text is then disregarded whenever it is not deemed relevant.

- *Irrelevant images*: The image detector module also recognises irrelevant visual items such as banners or graphical separators. Boundary regions of each document page (e.g., the bottom of the page) are defined, and all images placed in those regions are ignored, as they are unlikely to convey informative content. To prune irrelevant content, the following filters are applied to all visual elements: (1) *Minimum image size*: visual elements containing fewer than 150 pixels are dropped; (2) *Minimum height-width ratio*: visual elements whose absolute ratio is above 500

4.3.3 Keyword-based content annotation

For each document element e_l within each multi-page financial document $d_m \in \mathcal{D}$, the keywords k_j most relevant to e_l are retrieved. Specifically, a ranked list $k^1_{e_l,d_m}, ..., k^K_{e_l,d_m}$ of the top-K keywords assigned to e_l is returned. Note that the assigned keywords may refer to any facet, and the retrieved list can be empty.

The retrieval of keywords for text-only content is conducted in an unsupervised fashion using various textual similarity approaches, including both syntax-oriented and semantic-oriented methods. For each document element e_l, the K keywords whose textual descriptions are most similar to e_l are assigned according to the following measures:

- Syntactic similarities: (1) **ROUGE-1/2/L F1-Score** [66] measures syntactic overlap in terms of common unigrams, bigrams, or longest matching subsequence; (2) the **Levenshtein**, **Jaro**, and **Jaro-Winkler edit distances** measure the number of character-level operations needed to transform one piece of text into another.
- Semantic similarity: **SentenceBERT** [109] and **proprietary embeddings**, used to compare document elements and keyword descriptions via cosine similarity.

Additionally, GPT-4 is prompted with both the document element to be labelled and all possible keywords, requesting the assignment of the K most pertinent ones.

For simplicity, Figure 4.1 displays document elements and keyword descriptions as embedding representations in a latent space. However, the syntactic similarity and prompting approaches described above are also considered.

4.4 Experimental evaluation

Experiments were run on a machine equipped with a single NVIDIA® RTX A6000 48GB GPU. Standard Python libraries were used to compute syntactic similarity measures, whereas semantic similarity employed the SentenceBERT `paraphrase-MiniLM-L6-v2` model and `text-embedding-ada-002` as the proprietary OpenAI model. Llama2-Chat 7B with 16-bit quantisation was employed. GPT-4 (`gpt-4-0613`), GPT-4 Vision (`gpt-4-1106-vision-preview`), and `text-embedding-ada-002` were accessed through the OpenAI API.

Dataset. Business Units provided the following two in-domain datasets: (1) **Information and Communication Technology (ICT) Risk Analysis**, consisting of 11 documents and annotated with 2 facets and 25 keywords. It contains 991 textual elements, 13 images, and 15 tables. (2) **Trend Analysis**, consisting of 4 documents, annotated with 1 facet and 12 keywords, and including 69 images. Most images are presentation slides, which were handled by the LLM to obtain textual reformulations. Additional facets and keywords, along with their corresponding descriptions (92 overall), were also available, although not used for element annotation.

Evaluation Metrics. The efficacy of element annotation was evaluated using the following information retrieval metrics [78]:

- **Precision at K (P@K)**: percentage of returned keywords that occur in the expected keyword list.

- **Recall at K (R@K)**: percentage of expected keywords that occur in the returned keyword list.
- **Mean Reciprocal Rank (MRR)**: mean of the multiplicative inverse of the rank of the first correctly assigned keyword.

where K is the number of keywords retrieved that are considered. The rank order is based on the similarity score used to retrieve the keywords.

Keyword and description generation were assessed by comparing the produced and expected outcomes using established sequence-to-sequence metrics, i.e., ROUGE-1/2/L (R1/2/L) F1 score [66] for syntactic similarity and BERTScore (BS) F1 score [140] for semantic similarity.

Prompt description. The prompts for keyword and description generation were defined through a small number of iterative trials with domain experts. The design objective was to obtain concise, domain-relevant outputs with a stable structure that could be parsed automatically. For both models, the prompts explicitly specified the task (keyword generation or description), the target domain, the output language, and the desired output format (plain lists or short paragraphs), while keeping instructions as short as possible to reduce prompt sensitivity.

For keyword generation, we adopted a minimal, imperative style that foregrounds the facet name and the number of requested keywords:

Keyword generation: *The [K] most relevant keywords for the [FACET] domain are:*

where [K] and [FACET] are replaced with the desired number of keywords and the facet name of interest, respectively. In the few shot settings, this template is preceded by a short block of input–output examples of the form *Facet: [FACET_EXAMPLE]; Keywords: [KEYWORD_1], [KEYWORD_2], ...*, selected from expert-provided annotations. Examples are chosen to avoid overlap with the facet currently being evaluated, to limit lexical priming and data leakage.

For description generation, the prompt was designed to elicit short, self-contained explanations that resemble glossary entries:

Description generation: *Explain in a few lines the word between the quotation marks: "[KEYWORD]"*

where [KEYWORD] is replaced with the keyword for which to generate a description. In the few shot setting, one to three examples of the form *"[KEYWORD_EXAMPLE]": [DESCRIPTION_EXAMPLE]* are prepended to the prompt. For Italian experiments, the same templates were used with literal translations and explicit specification of the Italian language in the instruction. For all models, generation parameters were kept fixed across runs (temperature, maximum length, and nucleus sampling) to reduce stochastic variability and to make quantitative comparisons reproducible.

4.4.1 Results on content annotation

Textual semantic similarity based on contextual embeddings and LLM prompting achieved very promising results (MMR above 0.7) and outperformed both syntactic measures and edit distances (see Table 4.1). System precision decreased as the number K of retrieved keywords increased, whereas recall exhibited the opposite trend (see Figure 4.2). Similarity based on OpenAI embeddings performed best; for $K = 3$, P@K > 40% and R@K > 50% on both ICT Risk and Trend.

Similarity measure	ICT Risk Analysis	Trend Analysis
R1	0.458	0.300
R2	0.367	0.279
RL	0.472	0.258
Levenshtein	0.347	0.247
Jaro	0.483	0.249
Jaro-Winkler	0.483	0.249
SentenceBERT	0.658	0.430
embedding-ada-002	**0.779**	**0.610**
GPT-4	0.729	0.500

Table 4.1: Mean Reciprocal Ranks.

	K = unspecified		K = 3		K = 5		K = 10		K = 20	
	GPT-4	Llama2	GPT-4	Llama2	GPT-4	Llama2	GPT-4	Llama2	GPT-4	Llama2
RL	0.051	**0.066**	**0.070**	0.062	0.058	**0.062**	0.057	**0.065**	0.058	**0.060**
BS	**0.860**	0.859	0.862	**0.864**	0.861	**0.863**	**0.865**	0.860	**0.861**	0.857
P@K	0.771	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.867	**0.944**	0.833	**0.894**
R@K	**0.447**	0.296	**0.133**	**0.133**	**0.221**	**0.221**	0.375	**0.420**	0.721	**0.783**

Table 4: Evaluation of keyword generation for varying K. ICT Risk Analysis dataset. English language.

4.4.2 Results on keyword and description generation

Tables 2 and 4 summarise system performance on keyword description and keyword generation tasks, respectively. Due to space constraints, only outcomes on a single dataset, i.e., ICT Risk, are reported for both languages.

- *Proprietary vs. open-source LLM*: The proprietary model GPT-4 outperformed open-source models (Llama2/Camoscio) on keyword description generation for both tested languages (e.g., +33% ROUGE-1 on Italian documents). Conversely, open-source LLMs were highly competitive on keyword generation, likely because training examples of smaller models are more focused on specific domains, such as finance. This trend is confirmed by the results on Italian documents (not shown here due to space constraints).
- *Italian vs. English*: Both LLMs performed better on English than on Italian text. The gap in performance was more evident for the open-source LLMs, e.g., ROUGE-1 for description generation 0.31 Italian vs. 0.39 English.
- *In-context learning*: Prompting LLMs with few training examples (from 3 to 5) was beneficial for both keyword generation and description generation. Few-shot learning proved more beneficial for open-source LLMs because of their lower pre-trained model complexity.

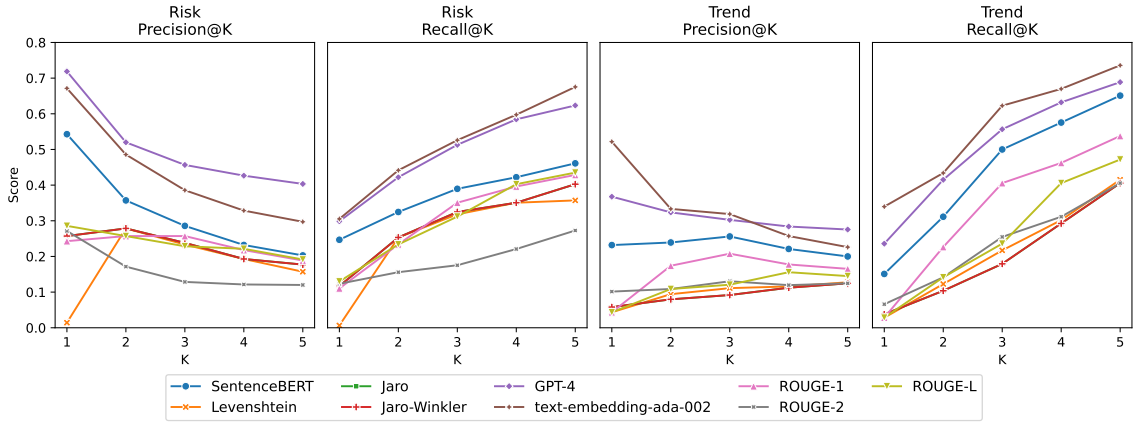


Figure 4.2: Precision@K and Recall@K values of different similarity measures on the ICT Risk Analysis (left) and Trend Analysis (right) datasets. English language.

4.4.3 Human evaluation

Each generated description, for both Italian and English languages, was annotated by five domain experts using a 5-point Likert scale based on five criteria [49]: (1) **Usefulness** (effectiveness in conveying key information); (2) **Coherence** (logical and semantic coherence); (3) **Non-Redundancy** (conciseness); (4) **Grammaticality** (linguistic correctness); and (5) **Overall Quality** (holistic evaluation of the generated description).

Results (see Table 3) were satisfactory and aligned with the quantitative outcomes (see Section 4.4.2). The perceived quality of Italian-written descriptions was lower than that of English ones, likely due to more limited model capabilities on languages other than English.

4.4.4 Qualitative examples

To illustrate the proposed approach, examples of outputs of the different steps are provided below.

Considering the ICT Risk Analysis dataset, one of the keywords associated with the *cyber risk* facet is *third-party risk*.

Reference description: *It refers to the potential risks or threats to an organization arising from relationships with third parties, such as suppliers, business partners, or external contractors. These risks* [...]

Generated description: *It is the risk that arises from the use of third-party vendors, suppliers, or partners that provide goods or services to an organization. Third-party risk can include a wide range of* [...]

Document element: *The image presents* [...] *in the context of retail banking leaders.* [...] *security providers aim to protect company, payment, card, and consumer data* [...] *the importance of various data privacy and security measures and where they stand in terms of industry focus and market trends.*

Target keywords: *third-party risk*, *regulation*
Assigned keywords: *third-party risk*, *regulation*, *compliance*

	Italian		English	
	Camoscio	GPT-4	Llama2	GPT-4
R1	0.310	**0.413**	0.394	**0.437**
R2	0.082	**0.169**	0.131	**0.150**
RL	0.208	**0.279**	0.254	**0.284**
BS	0.719	**0.773**	0.760	**0.902**

Table 2: Evaluation of keyword description generation performance. ICT Risk Analysis dataset.

	Italian	English
Usefulness	**4.38**±1.60	4.33±1.78
Coherence	4.52±0.93	**4.62**±1.15
Non-Redundancy	4.38±1.12	**4.52**±1.11
Grammaticality	4.60±0.73	**4.81**±1.20
Overall Quality	4.33±1.37	**4.34**±1.66

Table 3: Human evaluation of keyword descriptions. ICT Risk Analysis dataset.

4.4.5 Qualitative examples

To illustrate the behaviour of the proposed pipeline, this subsection reports qualitative examples covering both successful and less successful cases of annotation.

Considering the ICT Risk Analysis dataset, one of the keywords associated with the *cyber risk* facet is *third-party risk*.

Reference description: *It refers to the potential risks or threats to an organization arising from relationships with third parties, such as suppliers, business partners, or external contractors. These risks* [...]

Generated description: *It is the risk that arises from the use of third-party vendors, suppliers, or partners that provide goods or services to an organization. Third-party risk can include a wide range of* [...]

Document element: *The image presents* [...] *in the context of retail banking leaders.* [...] *security providers aim to protect company, payment, card, and consumer data* [...] *the importance of various data privacy and security measures and where they stand in terms of industry focus and market trends.*

Target keywords: *third-party risk*, *regulation*
Assigned keywords: *third-party risk*, *regulation*, *compliance*

In this example the generated description is close in content and style to the expert-written one, and the annotation step correctly recovers both target keywords. The additional keyword *compliance* reflects a broader regulatory framing of the same passage. Domain experts regarded this as acceptable enrichment rather than a genuine error, since governance and compliance aspects are discussed explicitly in the surrounding text.

The pipeline produces informative, domain-appropriate annotations in many practical situations, while still exhibiting characteristic failure modes linked to facet ambiguity, overlapping keyword definitions, and language coverage.

4.5 Discussion

The empirical evaluation of the automatic pipeline for annotating visually-rich financial documents yields several practical insights: (1) *Semantic similarity*: proprietary embeddings outperform open-source solutions for both Italian and English text; (2) *Keyword generation*: open-source LLMs perform as good as or even better than GPT-4 in zero-shot and few-shot learning settings on the tested documents, likely due to a higher in-domain specialization; (3) *Description generation*: GPT-4 performs best, while open-source LLMs perform reasonably well. Human feedback is in line with quantitative results based on established performance metrics.

Possible directions for future research include integrating the proposed method within a Retrieval Augmented Generation system and exploring zero-shot document classification by leveraging additional keywords that were not used for annotation. Another avenue is to evaluate the capabilities of other multimodal LLMs (e.g., LLaVA [68]) for generating textual descriptions of multimedia document elements.

- **Text-only processing.** This study focuses on textual content, whether originally textual or converted from visual formats. Visual and tabular content is not embedded directly. Extending the approach to handle native visual and tabular inputs would enable the adoption of state-of-the-art multimodal learning techniques.
- **Limited robustness to document layout variety.** Document structures vary substantially within and across domains, which may introduce inconsistencies in pre-processing and content extraction. Such variability can lead to suboptimal results in the subsequent annotation phase. The pre-processing and extraction pipeline will be refined as new document layout understanding models become available.
- **Limited scope of Multimodal LLM reasoning.** When text cannot be reliably extracted, in particular from presentation slides, GPT-4 Vision is used to generate textual explanations. The generated text may omit useful content or introduce inaccuracies; however, manual inspection suggests that the resulting explanations are adequate for the present purposes. A more systematic assessment of multimodal LLM capabilities on visually rich, domain-specific content is planned.

This chapter established the referential layer of the transparency stack for visually rich financial documents. It presented a keyword-based annotation pipeline that converts heterogeneous pages into traceable links between document elements and facet-specific keywords, with provenance preserved for audit. In the thesis narrative, these annotations realise grounded generation, since outputs can be tied to verifiable evidence within the source documents.

The annotated corpora, together with the element-level provenance signals, provide the foundation for Retrieval Augmented Generation pipelines developed in the next chapter. Chapter 5 builds directly on this groundwork to study answer synthesis over retrieved passages, comparing classical RAG with explicit summarisation strategies, and aligning referential transparency with concise, well-attributed answers.

Chapter 5

Retrieval Augmented Generation of Summarized Answers on Visually-Rich Documents for Trend and Risk Analysis

This chapter builds on the study "Retrieval Augmented Generation of Summarized Answers on Visually-Rich Documents for Trend and Risk Analysis" [32], conducted in collaboration with Giuseppe Gallipoli, Luca Cagliero, Alessandro Mosca, Politecnico di Torino, Turin, Italy; and Daniele Borghi, Intesa Sanpaolo Innovation Center, Turin, Italy. The work was presented and discussed at the 9th International Workshop on Data Analytics solutions for Real-LIfe APplications (DARLI-AP), co-located with the EDBT/ICDT 2025 Joint Conference, Barcelona, Spain, 2025, and published in CEUR Workshop Proceedings, Vol-3946. Minor adaptations and extensions have been introduced for coherence within this thesis.

This chapter advances the second pillar of the transparency stack, grounded generation, by studying Retrieval Augmented Generation on visually rich documents with an explicit focus on answer synthesis. Financial Trend and Risk Analysis relies on documents that mix text, tables, and figures; answers drawn from such sources are often verbose and repetitive, which undermines usability and obscures attribution. Building on the evidence-centred annotation workflow in Chapter 4, the present work examines how to produce concise answers while preserving traceability to the retrieved passages.

The contribution is a comparative evaluation of three strategies for producing synthesised answers from retrieved evidence: classical RAG without explicit compression, direct summarisation of the retrieved passages, and a cascade that summarises the answer produced by classical RAG. The pipeline segments and describes multimodal elements to enable retrieval over textual equivalents, then applies alternative summarisation modules, including large language models and traditional sequence-to-sequence systems. Evaluation combines automatic metrics that capture lexical overlap, semantic alignment, and keyword fidelity with a human assessment of grammaticality, usefulness, coherence, and non-redundancy.

The results indicate that direct summarisation of retrieved passages yields the most concise and well-attributed answers, with proprietary large language models outperforming open-source models and traditional summarisers. Cascading a summariser on top of a classical RAG answer is consistently detrimental; compressing the retrieved evidence is more effective than compressing an already generated response. These findings refine the design space for grounded generation in financial settings, linking referential transparency to practical answer quality.

Within the thesis narrative, this chapter connects the evidential groundwork of Chapter 4 with subsequent analysis of retrieval and answer faithfulness. The insights reported here inform prompt design, module choice, and evaluation protocols for RAG pipelines that must balance concision, fidelity to sources, and operational constraints.

5.1 Background and motivation

Visually-Rich Documents (VRDs) are types of documents that are commonly used in the banking sector to perform Trend and Risk Analysis (TRA). They consist of visual and textual elements such as charts, diagrams, textual paragraphs, and tables. Multimodal elements collectively refer to semantic entities whose identification, comprehension, and elaboration are crucial to solve advanced reasoning tasks such as Visual Question Answering [132], Entity Linking [16], and Key Information Extraction [22].

In the banking sector, analysts of TRA units often need to query financial VRDs to gain insights into the latest advancements in economic and technological fields. To support this time-consuming activity, the use of Large Language Models has become increasingly appealing [61]. Specifically, Retrieval Augmented Generation (RAG) systems combine the effectiveness of Information Retrieval modules, which extract passages relevant to the analyst-generated question, with the generative capabilities of LLMs [29]. Existing RAG applications to financial documents mainly focus on textual reports [136, 142], with limited research devoted to multimodal sources [33, 133], which are, however, of major interest for TRA banking units. Although RAG answers produced by LLMs are expected to be relevant to the input question, their conciseness and non-redundancy are usually not guaranteed by design. However, especially when dealing with financial VRDs, the multimodal content and its textual reformulations are often characterized by a fairly high level of verbosity, making the generated answers not sufficiently focused.

As a response to this issue, a RAG system was designed and implemented, and then tested on financial VRDs provided by the separate TRA units of a primary banking institution. Subsequently, given the textual passages shortlisted by the multimodal retrieval step, the level of synthesis of the RAG outputs produced by three alternative strategies is compared: (S1) **Classical RAG**: The LLM is prompted with the content of the retrieved passages without explicitly enforcing any summarization constraints; (S2) **Summarization**: The retrieved passages are summarized by an ad hoc summarization module; (S3) **Cascade of RAG and Summarization**: The output of S1 is summarized by an ad hoc summarization module.

The summarization performance achieved by the above-mentioned strategies S1–S3 is compared against a human-generated ground truth to address the following research questions:

(Q1) *Are LLMs effective in summarizing TRA document passages?*

(Q2) *To what extent are RAG outputs less similar than summarizers' outputs to ground truth summaries?*

(Q3) *Is it beneficial to apply text summarization on top of the Classical RAG answers?*

5.2 Problem statement

Given a set of Visually-Rich Documents $\mathcal{D}$ related to Trend, Innovation, and Risk Analysis in the banking sector and a textual question $t \in \mathcal{T}$ on $\mathcal{D}$, the RAG system returns the answers to each question t by performing the following steps: (1) *Document chunking and encoding*: it recognizes and splits the visual and textual elements in the documents' content, generates alternative textual descriptions of the visual elements, and encodes the text corresponding to each element separately; (2) *Passage retrieval*: it encodes the question t and retrieves the top-k passages P^t from $\mathcal{D}$ that are most relevant to t; (3) *LLM prompting*: it prompts the LLM with both the question t and the retrieved passages P^t. Note that Classical RAG prompts are designed for Question Answering and do not include any explicit summarization step.

The analysis focuses on the level of synthesis of the RAG output RAG_t^P. Specifically, given an input question $t \in T$, the corresponding passages P^t and the (human-generated) ground truth summary GT_P^t of P^t are compared with the following outputs:

1. **Classical RAG**: The final output of the RAG, denoted by RAG_t^P;
2. **Summarizer**: The output of an external summarizer that takes as input P^t, denoted by S_P^t;
3. **Cascade of RAG+Summarizer**: The output of an external summarizer that takes as input the RAG output RAG_t^P, denoted by $S\text{-}RAG_t^P$.

The diagram in Figure 5.1 illustrates the scenario under analysis, where similarities between retrieved passages and outputs are depicted using blue dashed lines, and similarities between outputs and ground truth summaries are depicted using red dashed lines. The summarizer module is not necessarily integrated into the RAG system, as various summarization approaches and models are explored, including abstractive summarization (using both LLMs and non-LLM models) and hybrid strategies combining extractive and abstractive methods.

5.3 Settings of RAG and Summarizers

The RAG system is implemented using the LangChain framework.

Document chunking and encoding The VRD elements are detected using the proprietary Document Intelligence service provided by the Azure AI platform [83], and alternative textual descriptions of visual contents are generated using the Multimodal LLM GPT-4o [95]. To encode the VRD elements, the OpenAI `text-embedding-ada-002` embedding model is used.

Passage retrieval Passages are retrieved via textual semantic similarity. Specifically, the textual content associated with the k elements in $\mathcal{D}$ whose embeddings maximize the cosine similarity with the t's encoding is retrieved.

LLM prompting The proprietary LLM GPT-4o is considered, and the following prompt is used:

> You are a virtual assistant that can do Q&A. Try to answer without using bullet points. Given the following context, write a text that highlights the topics discussed in the question. If any of the context elements are not useful, ignore them. If you don't know

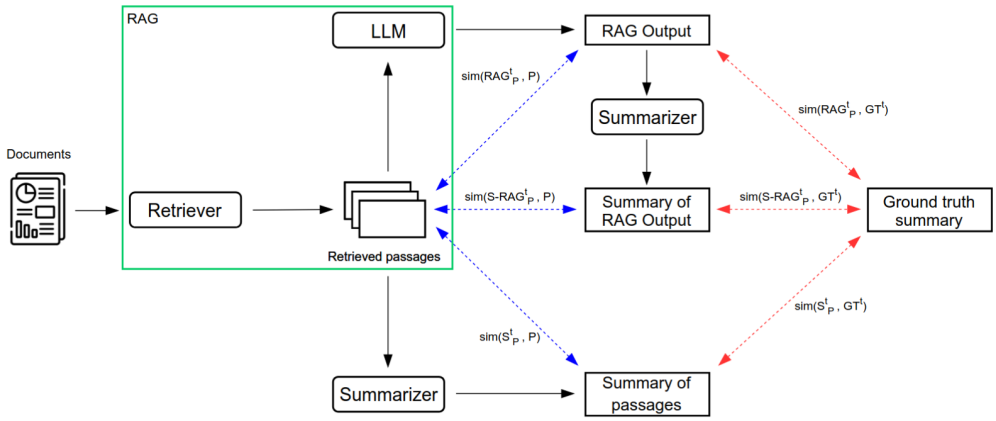


Figure 5.1: Sketch of our research scenario. Given a RAG system and an external summarizer, the similarities between the outputs of the RAG system and the summarizer are analysed with both the retrieved passages (shown as blue dashed lines) and the ground truth summaries (shown as red dashed lines).

> the answer, just say you don't know, don't try to invent an answer, but say that the documents you have can't satisfy the request.
> [context]
> [question]

where `[context]` and `[question]` are the retrieved passages and the current question, respectively.

External summarizers Experiments are conducted with traditional Transformer-based models, i.e., LED [10], which is suited to long documents, PEGASUS [138], BART [62], and T5 [108], three open-source LLMs, i.e., Llama3-Instruct 8B [71], Zephyr 7B [126], and Mistral-Instruct 7B [55] and one proprietary LLM, i.e., GPT-4o [95]. For LLM-based summarization, the following prompt is used:

> Summarize the following text.
> Focus on the topic of [keyword]: [to_summarize]

where `[keyword]` is replaced with the question expressed as a keyword and `[to_summarize]` with the corresponding retrieved passages to summarize. Two hybrid strategies combining extractive summarization using graph-based (TextRank, LexRank) or clustering (K-Means) methods with an LLM-based generative step are also tested.

5.4 Strategies for similarity computation

The pairwise textual similarities between passages, LLM answers, and summaries are evaluated using the following strategies:

Syntactic similarity The ROUGE-1/2/L (R1/2/L) F1-score is computed [66], indicating the unit overlap between the generated text and the ground truth in terms of unigrams, bigrams or longest common subsequence.

Semantic similarity The BERTScore (BS) F1-score is employed [140], leveraging BERT to compare the contextualised embeddings of the generated text and the ground truth. BS is used as a soft indicator of semantic relatedness rather than exact meaning preservation: distributional embeddings may assign relatively high scores to sentences that differ in polarity or contain antonyms (for example, "risk increased" vs. "risk decreased"), because they capture contextual similarity more readily than logical opposition. For this reason, BS is always interpreted jointly with ROUGE, keyword F1, and human judgements when assessing the quality and factual consistency of generated summaries.

Keyword-based similarity KeyBERT [41] is adopted to extract keywords from both the generated text and the ground truth, and then the corresponding F1-score is computed.

LLM-based similarity GPT-4o is prompted with the generated texts and the ground truth summary, asking it to identify which model's answer is better.

5.5 Experimental results

Open-source models are accessed via the Hugging Face Transformers library, and the proprietary GPT-4o (gpt-4o-2024-05-13) model [95] is accessed using the OpenAI API. Experiments were conducted on a machine equipped with an Intel® Core™ i9-10980XE CPU, 1 × NVIDIA® RTX A6000 48GB GPU, and 128 GB of RAM, running Ubuntu 22.04 LTS.

Datasets Three proprietary collections of English VRDs were analysed, provided by the following TRA units of a leading banking institution: (1) ICT Risk: 10 documents related to cyber risk, Distributed Ledger Technology, and AI in the ICT Risk area, containing 2800 textual elements and 45 visual ones; (2) Innovation: 3 documents related to embedded finance/insurance, digital players, and Digital Wealth Management, containing 82 textual elements and 32 visual ones; (3) Trend: 5 documents related to specific technologies and technological fields such as hydrogen economy, mainly containing visual elements (232).

TRA units' experts were asked to generate questions corresponding to distinct keywords (72 for ICT Risk, 19 for Innovation, and 11 for Trend). For each question, ground truth summaries were manually annotated by at least 3 units' experts.

Human evaluation of generated summaries To address Q1, a human validation of the passage summaries generated by the best-performing open-source LLM, according to the automatic evaluation metrics, and GPT-4o was conducted. TRA units' experts evaluated each output as Very Bad, Bad, Moderate, Good, or Very good, according to the following facets: Grammaticality, Usefulness, Coherence, Non-Redundancy, and Overall Quality [50]. Across the three datasets (72 ICT Risk, 19 Innovation, and 11 Trend questions), each system produced one summary per question, resulting in 102 summaries evaluated for each system. Multiple experts contributed ratings. In line with common practice in NLP, the 5-point Likert responses are treated as approximately interval-scaled and summarised using mean and standard deviation in Table 5.1; these values are used descriptively to compare systems, and we additionally inspected the full distribution of ratings (not reported here) to confirm that the observed ordering is robust. The results reported in Table5.1 highlight GPT-4o's superior summarization capabilities and, conversely, the limitations of open-source summarizers on TRA-related VRDs. Notably, GPT-4o demonstrates significant im-

Table 5.1: Human evaluation of summaries generated by the best-performing open-source LLM and GPT-4o. Bold denotes the best score for each metric.

	ICT Risk Analysis		Innovation Analysis		Trend Analysis	
	Llama3-Instruct	GPT-4o	Zephyr	GPT-4o	Zephyr	GPT-4o
Grammaticality	4.08±1.21	4.44±0.67	4.35±0.53	**4.53±0.45**	4.06±0.91	4.23±0.75
Usefulness	3.23±1.64	3.40±1.34	3.20±1.72	3.40±1.79	2.99±1.65	**3.76±1.56**
Coherence	3.55±1.19	3.75±1.34	3.70±0.71	3.87±1.10	3.66±1.14	**4.01±1.21**
Non-Redundancy	3.00±2.58	**4.15±1.35**	3.78±0.76	3.91±1.13	3.44±1.74	3.90±1.20
Overall Quality	3.02±1.64	**3.59±1.38**	3.17±1.55	3.36±1.78	2.96±1.71	3.56±1.62

provements in both Non-Redundancy (e.g., 3.00 vs. 4.15 in ICT Risk Analysis) and Overall Quality (e.g., 2.96 vs. 3.56 in Trend Analysis) criteria.

Comparison between RAG and summarizers' outputs with respect to ground truth summaries To address Q2, the similarities between the ground truth summaries and the outputs are evaluated for: (1) the classical RAG (see line *Output of Classical RAG* in Table5.2), and (2) the best configurations for each dataset of the different summarizers employed (see lines *GPT-4o, Llama3-Instruct, Zephyr, LED large, BART large*, and hybrid strategies in Table5.2). The results indicate that GPT-4o outperforms Classical RAG in terms of coherence with the ground truth summaries, whereas all the other summarizers, including the open-source LLMs and hybrid approaches, generally perform on par with or even worse than Classical RAG.

Effect of cascading RAG and Summarizer To address Q3, the line *Output of Cascade RAG+Summarizer* in Table5.2 reports the summarization performance of the approach based on applying summarization on top of the RAG output. In this case, GPT-4o is consistently used as the best-performing model for summarization. The comparison with Classical RAG indicates that cascading is never beneficial, even when employing the most effective summarizer, consistently resulting in performance degradation. These findings suggest that, since RAG answers are not specifically designed for the summarization task, concise answers condensing the relevant retrieved information can be produced more effectively by applying an explicit summarization step directly to the retrieved passages. Notably, summarizing the RAG output generated in the previous step fails to achieve the same quality as a direct summarization step, leading to even lower performance.

LLM-based similarity An A/B test is carried out using GPT-as-an-expert to compare Classical RAG against summarizers' outputs (see the left-hand side bars of the plots in Figure5.2) and Cascade RAG+Summarizer against summarizers' outputs (see the right-hand side bars). GPT-4o, the best-performing open-source LLM for each dataset, and the best-performing traditional Transformer-based model (non-LLM) for each dataset are considered as summarizers in the comparison. The results align with the automatic evaluation: GPT-4o as a summarizer outperforms both Classical RAG and Cascade RAG+Summarizer (>80% vs. <20%), open-source LLMs perform comparably with them (both around 50%), whereas non-LLM summarizers demonstrate worse performance (<20% vs. >80%).

Comparison between RAG and summarizers' outputs with respect to retrieved passages Instead of evaluating the similarities between the generated outputs and the ground truth summaries (see Table5.2), in this analysis the retrieved passages are considered as references. Specifically, Table5.3 reports the results of the comparisons between the retrieved passages and the outputs of (1) the classical RAG (see line *Output of Classical RAG* in Table5.2), (2) the

Table 5.2: Similarity results between RAG and summarizers' outputs with the ground truth summaries. Bold denotes the best score for each metric. * and † denote results for which $p < 0.05$ with respect to the outputs of Classical and Cascade RAG+Summarizer.

dataset	model	R1	R2	RL	BS	keyword F1	Δ token
ICT Risk Analysis	GPT-4o	**31.2***†	**12.2***†	**18.1***†	**85.2***†	10.0*	501†
	Llama3-Instruct	29.5†	9.4	16.9†	83.0*	7.4	160*†
	LED large	22.1*	11.1	15.7	81.2*†	**12.0***†	492†
	TextRank + Llama3-Instruct	29.9†	9.1	17.2†	82.8*	8.4†	170†
	K-Means + GPT-4o	23.8*	9.0	13.4*	85.0*	7.1	629*†
	Output of Classical RAG	29.1	9.0	16.2	84.4	7.1	504
	Output of Cascade RAG+Summarizer	24.6	7.4	13.9	84.2	5.6	573
Innovation Analysis	GPT-4o	**33.6***†	14.4†	21.2	**86.0***†	10.5†	-67
	Zephyr	29.7	9.9	17.0†	84.4	15.0†	68
	BART large	26.5*	**18.7**†	**23.7**†	81.5	17.5	175
	LexRank + Mistral-Instruct	32.4†	14.5†	15.4*	85.0	**20.0***†	65
	TextRank + GPT-4o	28.1	7.8	16.2*	85.4	7.5	124
	Output of Classical RAG	31.2	13.7	22.4	84.8	10.0	-34
	Output of Cascade RAG+Summarizer	26.5	5.4	14.9	83.9	5.0	39
Trend Analysis	GPT-4o	**42.8***†	**20.6***†	**25.1***†	**87.0***†	**18.5***†	245†
	Zephyr	31.5†	13.1†	18.3†	85.8†	18.1*†	538
	LED large	31.0†	16.8*†	20.4†	85.5	12.7	559
	LexRank + Llama3-Instruct	36.3†	16.7	20.1†	85.8	15.7*	-98*†
	TextRank + GPT-4o	26.7*	10.7	15.9	86.0†	14.4	621*†
	Output of Classical RAG	31.4	9.0	16.3	85.1	13.8	515
	Output of Cascade RAG+Summarizer	24.3	6.4	12.9	84.3	10.0	609

cascade of RAG and summarizer (see line *Output of Cascade RAG+Summarizer*), and (3) the best configurations for each dataset of the different summarizers tested (see lines *GPT-4o, Llama3-Instruct, Zephyr, LED large, BART large*, and hybrid strategies).

In most cases, GPT-4o outperforms other approaches, including open-source LLMs, traditional Transformer-based models, and hybrid strategies, achieving significantly higher performance compared to both Classical and Cascade RAG+Summarizer outputs. Consistent with the previous analysis, applying an additional summarization step on top of the RAG output proves detrimental, leading to lower scores across all metrics. The results align with those obtained using ground truth summaries as references. A key aspect is that higher similarity with respect to the retrieved passages indicates better attribution of the generated text to the source passages, which is particularly relevant in TRA domains, where maintaining a high level of accountability to the document sources is critical. In conclusion, the results indicate that GPT-4o, when used as a summarizer directly applied to the retrieved passages, excels at generating summaries that align well with both the ground truth summaries and the source documents. In contrast, the other summarizers, and in particular the two types of RAG outputs considered, demonstrate lower performance.

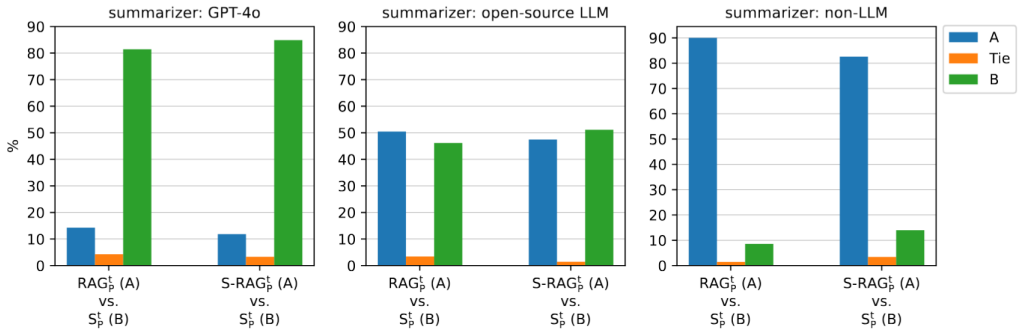


Figure 5.2: A/B tests using GPT-as-an-expert between (1) Left-hand side bars: Classical RAG output RAG_t^P (A) vs. summary output S_P^t (B); (2) Right-hand side bars: Cascade RAG+Summarizer output $S\text{-}RAG_t^P$ (A) vs. summary output S_P^t (B).

5.6 Discussion

The analysis examined methods for summarising passages retrieved by a RAG system indexing financial VRDs for Trend and Risk Analysis in the banking sector. In absolute terms, the ROUGE, BERTScore, and keyword F1 values reported in Tables 5.2 and 5.3, as well as the mean human ratings in Table 5.1, may appear moderate rather than high. This behaviour is consistent with the difficulty of the task, where short abstractive summaries must condense information from long, multimodal, domain-specific documents; even human-written references exhibit limited lexical overlap with system outputs, and experts apply a stringent rubric that penalises residual redundancy and missing nuance. For this reason, the discussion focuses on relative differences between systems under a fixed evaluation protocol, which still reveal consistent gains for GPT-4o over Classical RAG and the other summarisers across datasets and metrics. In interpreting the results, BERTScore is treated as complementary to ROUGE and keyword F1, since embedding-based similarity can overestimate the closeness of summaries that differ in sentiment or contain antonymic formulations; factual coherence is therefore ultimately validated through the human evaluation protocol. Using proprietary LLMs as summarisers improves the level of synthesis of classical RAG outputs, whereas open-source LLMs and traditional summarisers do not yield substantial performance gains given the complexity of multimodal, domain-specific sources. Applying summarisers directly to the retrieved passages is more effective than cascading RAGs with an additional summarisation step. Possible directions for future research include generating summarised answers using RAG systems with different characteristics, evaluating them on established benchmarks, and exploring sequence-to-sequence models specialised for languages other than English [59]. Another avenue is the development of explanation methods that highlight the weaknesses of RAG outputs.

The following limitations of this work are acknowledged:

Open-source LLMs Due to computational constraints, only the 8B-parameter version of Llama3-Instruct was tested. As a future extension, a broader suite of open-source LLMs with varying levels of complexity will be evaluated. Despite their smaller number of parameters, the open-source LLMs considered show fairly good performance, in some cases comparable to that of larger, proprietary models.

Model fine-tuning Currently, neither the LLMs nor the traditional Transformer-based model

Table 5.3: Similarity results between RAG and summarizers' outputs with the retrieved passages. Bold denotes the best score for each metric. * and † denote results for which $p < 0.05$ with respect to the outputs of Classical and Cascade RAG+Summarizer.

dataset	model	R1	R2	RL	BS	keyword F1	Δ token
ICT Risk Analysis	GPT-4o	**45.5***†	**27.2***†	**35.4***†	**88.7***†	**36.6***†	433†
	Llama3-Instruct	37.6	22.4	22.8	84.5	21.1	50†
	LED large	33.2	**28.5***†	20.7†	83.4	32.6*†	473†
	TextRank + Llama3-Instruct	36.8	24.1†	22.9	84.6†	23.1	187†
	K-Means + GPT-4o	36.1†	21.6†	23.8	84.8†	29.3*†	593†
	Output of Classical RAG	33.5	17.8	23.9	86.5	20.8	494
	Output of Cascade RAG+Summarizer	28.2	12.9	19.5	85.9	17.9	557
Innovation Analysis	GPT-4o	**41.8***†	21.8*†	32.7*†	87.9*†	31.0*†	228
	Zephyr	37.6†	22.6†	27.9*†	86.1*†	25.8*†	230
	BART large	39.6†	**34.8***†	**37.2***†	**88.3***†	**42.7***†	338*†
	LexRank + Mistral-Instruct	35.8†	19.3†	20.0	86.7	26.8*†	224
	TextRank + GPT-4o	39.9†	19.7	30.0	87.9*	32.5*†	304
	Output of Classical RAG	31.4	13.6	21.2	85.2	14.5	210
	Output of Cascade RAG+Summarizer	27.1	8.7	17.1	84.9	15.7	256
Trend Analysis	GPT-4o	**42.5***†	**28.9***†	**32.7***†	**89.2***†	**36.7***†	2133*†
	Zephyr	27.7†	21.4†	22.6†	85.8†	25.9*†	2416
	LED large	26.3†	26.1†	18.5†	85.7	27.1†	2407
	LexRank + Llama3-Instruct	40.2*†	21.9†	29.6†	86.4	24.7†	1764*†
	TextRank + GPT-4o	24.6†	11.7	22.1	86.6	27.0	2043†
	Output of Classical RAG	24.5	11.8	15.7	85.8	16.5	2394
	Output of Cascade RAG+Summarizer	18.7	8.2	11.9	85.0	12.8	2471

versions employed are specialised on domain-specific data. A selection of models will be fine-tuned to generate more domain-aware summaries.

RAG architecture For visual elements, the retrieval module currently relies on semantic similarity between textual descriptions of multimodal elements generated using GPT-4o. Future work will explore the use of different multimodal LLMs that also capture layout information (e.g., LayoutLLM [77]) and will test them in combination with various document retrieval strategies.

This chapter operationalised grounded generation for visually rich financial documents, with a focus on answer synthesis within Retrieval Augmented Generation. Within the transparency stack, the results strengthen the second pillar by showing how traceable evidence can be transformed into usable answers without eroding attribution. The evaluation protocol, combining automatic metrics with expert judgement, demonstrates how concision, coherence, and source alignment can be assessed under realistic operational constraints.

Chapter 6 extends this line of work from synthesised answers to end-to-end financial question answering on regulatory reports. It introduces a benchmark of numeric indicators, a multi-stage RAG architecture with contextual enrichment, advanced embeddings, and optional reranking, and a separation of retrieval and generation evaluation using NDCG and accuracy conditioned on ground-truth presence. The transition from passage-level synthesis to audited numeric answers consolidates

grounded generation under deployment constraints, preparing the ground for subsequent analysis of internal mechanisms.

Chapter 6

Enhancing Financial Question Answering Through Retrieval-Augmented Generation: A Novel Benchmark Dataset and Multi-Stage RAG Architecture

This chapter consolidates the second pillar of the transparency stack, grounded generation, by introducing a financial question answering benchmark and a multi-stage Retrieval-Augmented Generation architecture tailored to regulatory reports. Aligned with the thesis narrative, the objective is to strengthen referential transparency under operational constraints, and to quantify reliability through systematic evaluation. Building on the annotation and retrieval insights developed in Chapters 4 and 5, the focus here is end-to-end performance on numeric indicators that are central to risk and capital analysis.

The chapter presents a benchmark of 999 questions across 24 banks and five years, paired with a corpus of annual and Pillar 3 reports. The pipeline integrates hierarchical chunking with contextual enrichment, domain-strong dense embeddings, optional reranking, and controlled generation with both a general-purpose model and a reasoning-oriented model. Evaluation separates retrieval and generation, using NDCG@k for ranking quality and accuracy conditioned on ground-truth presence in the retrieved context for answer quality. Latency and cost are recorded to reflect deployment trade-offs.

Empirically, contextual enrichment combined with advanced embeddings yields substantial gains in retrieval quality, while reranking provides mixed benefits when initial retrieval is already strong. For generation, a reasoning-optimised model markedly improves numeric answer accuracy, although at higher computational cost. These results provide design guidance for financial RAG systems that must balance precision, traceability, and efficiency, and they complete the grounded generation layer that supports the subsequent investigation of internal mechanisms in Chapter 7.

6.1 Background and motivation

The extraction of key indicators such as capital ratios (e.g., CET1), liquidity reserves, or asset and liability classifications requires substantial time, specialist skills, and cognitive effort from analysts and auditors. This process is particularly difficult to scale in large-scale comparative contexts, highlighting the need for automated analysis solutions.

Recent advances in Large Language Models (LLMs) have demonstrated remarkable capabilities across various natural language processing tasks. However, LLMs face significant limitations when applied to specialized financial domains, including the phenomenon of hallucinations—generating plausible but factually incorrect responses—and knowledge cutoffs that prevent access to recent financial data. To address these limitations, Retrieval-Augmented Generation (RAG) frameworks have emerged as a promising solution, In the financial sector, institutions such as banks are required to periodically publish detailed documents illustrating their financial position, capital levels, risk exposure, and liquidity availability. These documents include financial reports prepared following international accounting standards (International Financial Reporting Standards (IFRS) [48] or Generally Accepted Accounting Principles (GAAP) [31]) and Pillar 3 reports prepared following Basel Committee standards [7]. While essential for transparency and regulatory oversight, these reports present significant analysis challenges due to their considerable length (often exceeding hundreds of pages) and complex technical and legal structure.

This work addresses the critical need for effective financial question answering systems and makes the following contributions:

1. A novel benchmark dataset is introduced, containing 999 questions related to key financial indicators across 24 major international banks, spanning 2019–2023.

2. A comprehensive RAG architecture specifically optimized for financial document analysis is presented, incorporating contextual chunk enrichment, hybrid retrieval strategies, and advanced reranking techniques.

3. An extensive experimental evaluation compares multiple retrieval approaches, embedding models, and generation strategies, establishing new performance baselines for financial question answering.

4. Significant performance improvements resulting from domain-specific optimizations are reported, with retrieval accuracy increasing by 38.8% and generation accuracy by 34.4%.

Our research seeks to answer the overarching question:

> Can the performance of Retrieval-Augmented Generation (RAG) systems for financial question answering be improved through (i) enhanced retrieval techniques, specifically changing the embedding model, applying chunk enrichment, and incorporating reranking, and (ii) improved generation techniques, specifically employing a reasoning-capable model?

6.2 Dataset

6.2.1 Dataset Structure

Our benchmark dataset was compiled by the Market and Counterparty Risk IMA Methodologies group at Intesa Sanpaolo Milan, based on data collected annually for benchmarking operations and comparative bank analysis. The dataset contains 999 rows representing financial questions across 10 key financial indicators.

The dataset structure includes the following components:

- **Data**: The name of the indicator of interest (10 data values). The financial indicators covered include: Common Equity Tier 1 (CET1) capital, Day One Profit, Total Additional Valuation Adjustments (AVA), Total Assets, Total Fair Value Level 1, 2, and 3 Assets, Total Fair Value Level 1, 2, and 3 Liabilities.
- **Year**: The year to which the data refers, ranging from 2019 to 2023 (inclusive) (5 years).
- **Bank**: The bank to which the data refers (24 major international banks).
- **Doc Type**: indicates the type of document in which the data appears. It can be a *Report* or a *Pillar 3* document.
- **Query**: The queries submitted to the model. These follow a standardised format:

  > *"What is the consolidated <Data> value in millions for <Bank> for the year <Year>?"*

- **Value**: column, which serves as ground truth and is a decimal number in millions. All ground truths are *"simple"* values, i.e., they are not derived from a combination of other values.
- **Numerical**: column indicates whether the value is present in clear numerical form. For example, sometimes there are indicators whose value is zero (21 cases). These can be indicated in natural language or expressed in tables with dashes (–). The reason this table was added is to better manage the retrieval phase.
- **In table**: column indicates whether the value appears in a table (Y), is present in the document but not in a table (N).
- **Source Document Year**: column specifies the publication year of the document from which the data was extracted. Financial reports commonly include comparative data, presenting figures for the current reporting period alongside those from one or more previous years. Consequently, the value for a specific financial indicator and year (e.g., 2022) might be sourced from a document published in a subsequent year (e.g., the 2023 annual report). This means the **Source Document Year** does not necessarily match the **Year** of the data point, a characteristic that reflects real-world financial analysis practices and introduces a temporal challenge for retrieval systems.

Data	Year	Bank	Doc Type	Query	Value	Numerical	In table	Source Document Year
Total Assets	2023	Banco BPM	Report	What is the Total Assets value in million for Banco BPM for the year 2023?	202131.973	Y	Y	2023

Table 6.1: Example of a row in the dataset

Total number of documents	209	**Total number of chunks**	150437
Mean number of chunks	720	**Max number of chunks**	2345
Mean number of words/chunk	276	**Max number of words/chunk**	19625
Mean number of words/document	198515	**Max number of words/document**	556505

Table 6.2: Detailed statistics of the dataset.

6.2.2 Dataset Statistics

Table 6.2 reports the corpus statistics: 209 documents segmented into 150,437 chunks, averaging 720 chunks per document with a maximum of 2,345. Each chunk contains on average 276 words (maximum 19,625), while documents average 198,515 words with a maximum of 556,505.

6.2.3 Document Corpus

The dataset includes 209 financial documents:

- 120 Annual Financial Reports
- 89 Pillar 3 Reports

These documents span five years (2019-2023) and cover 24 major international banks including both European institutions (Intesa Sanpaolo, UniCredit, BNP Paribas, Deutsche Bank, HSBC) and American banks (JPMorgan Chase, Bank of America, Citigroup, Wells Fargo).

6.2.4 Dataset Characteristics

The dataset is characterized by several inherent challenges that mirror real-world financial analysis. These include the structural complexity of financial documents, which often span hundreds of pages with intricate hierarchies; the prevalence of technical jargon and domain-specific abbreviations; the necessity for precise numerical reasoning to extract and interpret quantitative data, including unit conversion to match the ground truth required in millions; and the diverse presentation formats, with information appearing in tables, embedded within text, or occasionally absent.

6.3 Experimental Setup

6.3.1 RAG Architecture

Retrieval-Augmented Generation (RAG) is an architectural paradigm designed to enhance the accuracy and reliability of Large Language Models (LLMs) by grounding their responses in external knowledge bases. Instead of relying solely on its internal, pre-trained knowledge, a RAG system first retrieves relevant information from a specified corpus of documents and then uses this information to generate a more informed and contextually accurate answer. Our RAG architecture can be deconstructed into a multi-stage pipeline: document ingestion and preprocessing, chunk indexing, retrieval, reranking and generation.

Document Ingestion and Preprocessing

PDF documents were processed using Microsoft Azure Document Intelligence [5] to extract text in Markdown format. A modified version of LangChain's MarkdownHeaderTextSplitter was developed to segment documents according to hierarchical headings, maximising granularity to improve retrieval accuracy. The preprocessing pipeline comprises OCR-based text extraction with layout preservation, Markdown conversion that maintains document structure, hierarchical chunking based on heading levels, page-range tracking for each chunk, and preservation of metadata (bank name, year, document type).

Chunk Indexing

The processed chunks are transformed into numerical representations (embeddings) and stored in a specialized vector database for efficient searching. Two leading dense embedding models were evaluated to assess their impact on retrieval performance within the financial domain. All chunks were indexed in a Qdrant [105] vector store. The selected models were:

- **OpenAI text-embedding-3-large (baseline) [96]**: A powerful, general-purpose model with 3072 dimensions. It serves as baseline due to its strong performance on broad benchmarks like MTEB - Massive Text Embedding Benchmark (64.6%) [97]and its widespread adoption in production RAG systems.
- **VoyageAI voyage-3-large [128]**: A state-of-the-art model specifically optimized for retrieval tasks. It has demonstrated superior performance over other models, including a 9.74% higher NDCG@10 than *text-embedding-3-large* across diverse datasets [128].

Contextual Chunk Enrichment

Traditional RAG systems split documents into small chunks for retrieval and semantic search. However, this splitting often discards important context, making individual chunks hard to understand and leading to weaker retrieval results.

To address this issue, the Contextual Retrieval technique inspired by Anthropic [4] is applied. For each chunk, a concise, synthetic contextual summary (typically 50-100 tokens) is automatically generated using GPT-4.1, which has a 1 million token context window. This summary describes how the chunk fits within the overall document, so that even small document fragments retain sufficient background information to be accurately retrieved and interpreted. The context is then

prepended to the original chunk before embedding or indexing, thereby supplying retrieval models with rich, domain-aware representations.

The enrichment process uses document structure and heading hierarchies to identify the appropriate context for each chunk, even in very large files that exceed the model's context window. By dynamically adjusting parameters (such as the allowed difference in heading levels), the process ensures that each chunk's context remains accurate while respecting computational and token limits.

Reranking

Initial retrieval, based on semantic search, is optimized for speed over a large corpus but may return chunks that are only broadly relevant. Reranking introduces a second, more precise filtering stage to refine these initial results. It employs a more powerful but computationally intensive model, which re-evaluates the top-k relevant chunks by jointly considering the query and each chunk's content. This deep semantic analysis provides a more accurate relevance score than the initial vector similarity search.

The model used in this work is Cohere rerank-3.5 [20], a cross-encoder trained on large-scale data including web search, question-answering, and multilingual corpora. The reranking process initially retrieves 100 chunks, then narrows down to the top-k through relevance scoring.

Generation Models

In the final phase, the top-ranked, context-rich chunks are combined with the original user query and passed as context to a generator LLM. The LLM then synthesizes this information to produce the final answer.

We evaluate two large language models for this phase:

- **GPT-4o**: A highly capable, general-purpose multimodal model used as a baseline. It features a 128,000-token context window and is known for its strong performance across a wide range of tasks.
- **GPT-o1-high**: A model specifically optimized for complex reasoning tasks. It provides a 200,000-token context window and is configured to use its maximum reasoning effort, prioritizing accuracy over response latency. This model is selected to test the hypothesis that enhanced reasoning capabilities improve performance on numerical extraction and interpretation tasks.

6.3.2 Evaluation Metrics

Retrieval Phase

Retrieval quality is evaluated using the Normalised Discounted Cumulative Gain at k (NDCG@k) metric [53, 28], which assesses the quality of a ranked list by prioritising relevant items in top positions. Tests were conducted for k values of 1, 10, and 20. NDCG@k is defined as:

$$NDCG@k = \frac{DCG@k}{IDCG@k} \tag{6.1}$$

where DCG@k (Discounted Cumulative Gain (DCG)) is calculated as:

$$DCG@k = \sum_{i=1}^{k} \frac{G_i}{\log_2(i+1)} \tag{6.2}$$

Here, G_i is the relevance score of the chunk at rank i, and $IDCG@k$ is the DCG of an ideal ranking.

Because the dataset does not explicitly label "golden chunks," the relevance score G_i is computed dynamically for each retrieved chunk. A dedicated function searches the chunk text for the numerical ground truth value, accounting for various formatting permutations (e.g., different decimal and thousand separators). The relevance score G_i equals the number of occurrences of the ground truth value within the chunk; chunks not containing the value receive a score of 0. Queries for which the ground truth is not a numerical value are excluded from this evaluation.

The result is a value between 0 and 1, where 1 represents a perfect ranking (all relevant items are in the highest positions) and 0 represents the absence of relevance in the first k results. In general, the higher the NDCG, the better the quality of the ranking compared to the ground truth.

Generation Phase

To separate generator performance from retrieval errors, generation accuracy is evaluated only on queries for which the ground-truth value appears within the top-k retrieved chunks. Performance is measured using accuracy, defined as the proportion of correct answers. An answer is deemed correct if the generated value equals the ground-truth value, if both values are null, or if the absolute difference between the numerical values does not exceed 0.01. The accuracy is calculated as:

$$\text{Accuracy} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}(\hat{y}_i \approx y_i) \tag{6.3}$$

where N is the number of evaluated queries, $\hat{y}_i$ is the predicted value, y_i is the ground truth, and $\mathbb{I}(\cdot)$ is the indicator function that is 1 if the condition is met and 0 otherwise.

6.4 Results

The experimental evaluation is structured in two phases: retrieval and generation. Various configurations are assessed to identify the optimal components for a financial RAG pipeline.

6.4.1 Retrieval Performance

Table 6.3 summarizes the performance of different embedding and retrieval strategies. All values were calculated from a total of 978 queries (999 total minus 21 where the ground truth is not numerical).

The key findings from the retrieval evaluation are:

- **Contextual Enrichment and Advanced Embeddings**: The combination of contextual chunk enrichment and the VoyageAI embedding model yields a dramatic performance increase. This configuration achieved a peak NDCG@10 of 0.710, a 38.8% absolute improvement over the 0.322 NDCG@10 of the OpenAI baseline. This configuration also had the fastest average retrieval time.

Table 6.3: Retrieval results (NDCG@k) for different scenarios. The numbers in parentheses indicate the count of queries for which the ground truth was found in the top-k retrieved chunks. Bold values indicate the best performance for each k.

Scenario	NDCG@1	NDCG@10	NDCG@20	Avg. Time (s)
OpenAI embeddings (baseline)	0.163 (159)	0.322 (524)	0.347 (649)	1.94
OpenAI embeddings + headings	0.166 (162)	0.315 (508)	0.346 (651)	1.44
OpenAI contextualized embeddings	0.248 (242)	0.459 (723)	0.478 (830)	1.79
OpenAI contextualized + reranking	0.332 (324)	0.494 (698)	0.511 (795)	4.30
VoyageAI contextualized embeddings	**0.510** (498)	**0.710** (929)	**0.705** (956)	**1.33**
VoyageAI contextualized + reranking	0.342 (334)	0.498 (700)	0.512 (791)	3.38

- **Reranking Effects**: Reranking produced mixed results. It improved the performance of the OpenAI contextualized embeddings scenario by an average of 5.1% across k values. However, it degraded the performance of the superior VoyageAI configuration by an average of 19.1%, suggesting that reranking may be detrimental when the initial retrieval quality is already high.
- **Impact of k**: Increasing k from 10 to 20 for the best-performing scenario (VoyageAI contextualized) slightly decreased the NDCG score (from 0.710 to 0.705) but increased the number of queries with a retrieved ground truth from 929 to 956 (out of 978). This highlights a trade-off between ranking quality and overall recall.
- **Structural vs. Semantic Enrichment**: Using document headings for enrichment provided negligible benefit over the baseline, indicating that rich semantic context is more critical for retrieval than structural metadata alone.

6.4.2 Generation Performance

Generation performance was evaluated using accuracy, defined as the proportion of generated answers matching the ground truth. To isolate generator capability, only queries for which the ground truth appeared in the retrieved context were considered. Reranking scenarios were therefore excluded from this phase based on the retrieval results.

Table 6.4 presents the accuracy for the GPT-4o (baseline) and GPT-o1-high (reasoning-optimized) models across different retrieval scenarios and k values.

The main insights from the generation phase are:

- **Reasoning Model Superiority**: The reasoning-optimized GPT-o1-high model consistently and significantly outperformed the GPT-4o baseline across all scenarios. The weighted average accuracy for GPT-o1-high was 79.0%, compared to 44.6% for GPT-4o, representing a 34.4% absolute improvement.
- **Optimal Context Size (k)**: For generation, providing a smaller, more focused context (k=1) often yielded the highest accuracy, particularly for the top-performing GPT-o1-high model. This suggests that including more, potentially noisy, chunks can degrade the generator's ability to pinpoint the correct answer, even if the ground truth is present.

Table 6.4: Generation accuracy for different models and retrieval scenarios. The number in parentheses indicates the count of queries evaluated.

Scenario	Model	k=1	k=10	k=20
OpenAI embeddings (baseline)	4o	0.306 (180)	0.402 (545)	0.399 (670)
	o1-high	0.544 (180)	0.747 (545)	0.755 (670)
OpenAI embeddings + headings, prompt + headings	4o	0.399 (183)	0.454 (529)	0.406 (672)
	o1-high	0.623 (183)	0.798 (529)	0.808 (672)
OpenAI contextualized embeddings	4o	0.490 (263)	0.504 (744)	0.497 (851)
	o1-high	0.722 (263)	0.823 (744)	0.812 (851)
VoyageAI contextualized embeddings	4o	0.455 (519)	0.467 (950)	0.436 (977)
	o1-high	0.852 (519)	0.822 (950)	0.811 (977)

- **Performance vs. Latency**: While GPT-o1-high delivered superior accuracy, its average generation time was approximately 20 times longer than GPT-4o (35.0s vs. 1.8s). This highlights a critical trade-off between accuracy and response latency for practical applications.

6.4.3 Error analysis

A qualitative inspection of a stratified sample of failure cases was conducted to characterise typical error patterns in both retrieval and generation. On the retrieval side, most errors arose in settings where the ground-truth value appeared in dense tables alongside several related quantities, sometimes expressed with different units or scaling factors. In these cases, semantically similar chunks referring to the same indicator but a different bank, year, or consolidation perimeter were occasionally ranked above the truly relevant chunk, especially for the baseline embedding configurations. Additional retrieval failures were observed when the value was expressed in natural language or through symbols (for example, dashes indicating zero) rather than as an explicit numeral, which reduced the effectiveness of the value-matching procedure used for relevance scoring. For generation, GPT-4o errors typically involved selecting a figure from the correct table but the wrong cell, often taking the value from an adjacent column or row instead of the target one, confusing current and comparative years, or misinterpreting units when tables mixed millions with other scales. In a smaller number of cases, the model produced plausible but unsupported values when the relevant chunk had not been retrieved. GPT-o1-high substantially reduced these errors, particularly unit and year confusions and adjacent-cell misselections, but did not eliminate them completely, and occasional mistakes still occurred when tables contained multiple candidate values or when formatting made numerical parsing difficult. These observations support the quantitative findings by indicating that residual errors are concentrated in structurally complex cases, and they suggest that further gains are likely to come from table-aware retrieval and parsing and from a more detailed analysis of table lookup mechanisms, which is taken up in the following chapter.

6.5 Discussion

This chapter presents a comprehensive evaluation of RAG systems for financial question answering, introduces a novel benchmark dataset, and demonstrates significant performance improvements resulting from the applied optimizations. The key contributions are:

1. A novel benchmark dataset with 999 expert-validated questions across 10 financial indicators and 24 major banks, providing a rigorous testbed for financial QA systems.
2. Substantial performance improvements through contextual chunk enrichment and advanced embedding models, with retrieval accuracy increasing by 38.8% (NDCG@10: 32.2% to 71.0%).
3. Demonstration of reasoning model superiority, with GPT-o1-high achieving 34.4% higher generation accuracy compared to GPT-4o (79.0% vs 44.6%).
4. Comprehensive evaluation of multiple RAG configurations, revealing that domain-specific optimizations significantly outperform generic approaches.

The findings highlight several important insights for financial RAG system development. Contextual enrichment of document chunks is more valuable than structural enhancements, indicating that semantic understanding supersedes syntactic organisation. Advanced embedding models adapted to the financial domain yield substantial improvements over general-purpose alternatives. Reasoning-optimised language models demonstrate superior performance on tasks requiring precise numerical extraction and interpretation, which are common in financial analysis.

Several limitations affect the current approach:

- **Computational Cost**: Contextual enrichment requires processing entire documents with large language models, introducing significant computational overhead that may be prohibitive for smaller organisations.
- **Reranking Inconsistency**: Although reranking typically improves retrieval performance, the results exhibit mixed outcomes, indicating that the interaction between initial retrieval quality and reranking effectiveness warrants further investigation.
- **Reproducibility Concerns**: The stochastic behaviour of reasoning models, particularly GPT-o1-high, may affect the reproducibility of outcomes, which is critical for financial applications that require audit trails.

Several directions emerge for future research:

- **Efficiency Optimization**: Developing more efficient contextual enrichment methods, potentially using smaller specialized models for context generation or selective enrichment based on chunk importance.
- **Refined Preprocessing**: Enhancing document preprocessing by using cost-effective models to pre-filter chunks, selecting only those containing tables or relevant financial data to improve the efficiency of subsequent pipeline stages.
- **Alternative Reranking Strategies**: Exploring alternative reranking methods, such as using more economical proprietary models or fine-tuning open-weight models, to address the inconsistent performance and high cost of current rerankers.

- **Advanced Generation Models**: Experimenting with newer and more advanced language models in the generation phase to evaluate potential improvements in accuracy and reasoning capabilities.

Our work establishes a foundation for advanced financial document analysis systems, demonstrating that thoughtful application of RAG techniques can significantly improve the accuracy and efficiency of financial question answering. As financial institutions increasingly adopt AI-driven analysis tools, these findings provide crucial guidance for developing robust, reliable, and scalable solutions.

From Financial RAG to Mechanistic Interpretability The development and evaluation of the Retrieval Augmented Generation pipeline addressed the practical problem of extracting verifiable quantitative information from large regulatory documents. By combining contextual chunk enrichment with dense-sparse hybrid indexing and reranking, then pairing retrieval with reasoning oriented large language models, the system produced consistent gains in retrieval precision and answer accuracy on numerical risk indicators, including capital adequacy and liquidity ratios.

Despite these improvements, limitations were observed in the generative stage, including occasional hallucinations, numerical inconsistencies, and a lack of transparency in the internal decision process. These behaviours raised a central research question, namely how transformer models internally retrieve and manipulate structured information that is typical of financial documents, such as tables.

Within the transparency stack, the RAG architecture examined here instantiates the grounded generation layer. It establishes referential transparency by grounding model outputs in retrievable document passages, thereby enabling citation and verification. However, the generative component remains opaque at the computational level. The reasoning processes that select, combine, and transform retrieved information into numerical answers are not directly observable, which limits the interpretability required for high-stakes financial applications.

This observation motivated a shift towards a mechanistic interpretability perspective. A preliminary study was therefore conducted on table lookup tasks, which provide a simplified yet representative abstraction of structured data reasoning in financial QA. The study applied logit attribution, activation patching, and path patching on smaller language models to identify the attention heads and multi layer perceptrons that locate and propagate tabular information. This bridge from applied RAG to circuit level analysis sets the stage for the next chapter, where the internal mechanisms underlying table lookup are examined in detail. That investigation aims to populate the computational transparency layer of the stack, thereby complementing the referential transparency achieved through retrieval with insight into the algorithmic primitives that underlie structured reasoning.

Chapter 7

Mechanistic Interpretability Analysis of Table Lookup

This chapter develops the third pillar of the transparency stack, mechanistic interpretability, by analysing how transformer language models implement table lookup under controlled conditions that mirror financial question answering. In the thesis narrative, the focus shifts from referential transparency in grounded generation to causal transparency, that is, from verifying answers against evidence to explaining the internal computations that produce those answers. The preceding chapter demonstrated that a Retrieval Augmented Generation pipeline can extract risk measures from large scale regulatory documents with improved precision and answer accuracy. Residual hallucinations, inconsistencies in numerical reasoning, and opaque internal decision processes motivate a closer examination of the mechanisms that support structured retrieval and the mapping from queries to numeric values.

Methodologically, the analysis follows standard practice in mechanistic interpretability: begin with a simplified, concrete task, then seek generalisation once mechanisms are identified. Table lookup is adopted as a minimal two dimensional addressing problem that requires binding a column header to a row identifier and propagating the corresponding cell value to the output. Prompts enforce single token headers, row identifiers, and values in a compact Markdown table to stabilise attribution and isolate the computation of interest. The study employs a small open source model to enable exhaustive probing. Direct logit attribution and the logit lens indicate when the correct direction emerges in the residual stream. Activation patching, with matched clean and corrupted prompts along both the column and row axes and a logit difference metric at the answer position, tests the sufficiency of specific components.

The objective is to identify computational pathways, at the level of attention heads and multi layer perceptrons, that support key subroutines, including locating the relevant row and column, composing keys, and propagating the correct value to the output. The contribution is a circuit level account of how a transformer can encode and retrieve tabular associations, linking the applied insights from financial RAG with an explanatory description of the underlying mechanisms. The study is framed as exploratory, oriented toward circuit discovery rather than definitive claims.

7.1 Exploratory analysis

Following [130], a strategy of constructing simple, controlled algorithmic tasks that elicit the computation of interest is adopted. These tasks allow the isolation, probing, and causal validation of mechanisms for table lookup, without confounds from broader linguistic context. The steps presented in [81] for the IOI task are adapted and tailored to understanding financial language models tasks involving table lookup.

7.1.1 Prompt design

The prompt is designed to isolate the table lookup mechanism under controlled conditions while remaining close to financial question answering. A compact Markdown table with a 3×3 value grid (plus an identifier column) minimises sequence length, which makes attribution methods stable and reduces confounds from long-range dependencies. Markdown is also widely represented in pretraining corpora, and its pipe-separated structure provides consistent boundary tokens that help diagnose positional and structural cues.

```
PROMPT FORMAT:
Look at the table below and answer the question.

| Bank   |   IRC |   ECL |   ROE |
|:-------|------:|------:|------:|
| Red    |     4 |     7 |     8 |
| Blue   |     5 |     9 |     3 |
| Gold   |     6 |     1 |     2 |

Question: What is the {} for Bank {}?

Answer:
```

Single-token constraints for headers, row identifiers, and values are essential for mechanistic analysis. They remove subword composition effects that otherwise blur token-level attributions, and they ensure that attention, QK, and OV decompositions can be interpreted at the granularity at which the model actually computes. Choosing financial acronyms (IRC, ECL, ROE) preserves domain semantics relevant to the application, yet remains single-token under Gemma-2b-it. Using simple row identifiers that are single tokens achieves the same goal on the row axis.

The values are restricted to digits 1–9, with no duplicates anywhere in the table. Digits are single tokens, they avoid subword ambiguity, and excluding zero mitigates tokenisation and behavioural idiosyncrasies often seen with 0. Uniqueness is critical for clean versus corrupted comparisons: when generating counterfactuals by changing only the queried row or column, the correct answer necessarily differs, so activation patching metrics are well defined and not attenuated by accidental equality.

The question template explicitly names a column and a bank, which forces the model to bind a column header to a row identifier, then read the corresponding cell. The explicit Answer cue standardises the decoding context, reducing output variability and enabling consistent logit-difference measurements at the answer position. This structure supports direct logit attribution, residual

stream lensing, and per-component patching because the task reduces to selecting a single target token given two disambiguating keys.

Finally, the chosen configuration is the minimal non-trivial setting that requires two-dimensional addressing while remaining short enough for exhaustive causal probing. It also scales: varying table size, format (Markdown versus HTML), semantics of headers, and the presence or absence of duplicates provides a controlled suite of shifts to probe whether the model relies on positional heuristics, surface formatting, or genuine row–column composition. This progression enables both performance characterisation and identification of stable computational pathways across prompt variants.

7.1.2 Model

This study adopts `google/gemma-2b-it` as the primary subject of analysis. Among models supported in `TransformerLens`, it is the model that, while keeping parameter count and internal dimensions tractable for exhaustive attribution and patching, consistently solves the constrained table-lookup prompt. Under identical prompting and decoding, alternative baselines were also evaluated; `gpt2-small`, `gpt2-xl`, and `EleutherAI/gpt-neo-2.7B` failed to answer correctly (0% accuracy), whereas `google/gemma-2b-it` achieved 100% accuracy.

Compared with the `GPT-2 Small` configuration used in [81], Gemma introduces architectural features that complicate a mechanistic interpretability workflow. In particular, Root Mean Square Normalisation (RMSNorm), Rotary Positional Embeddings (RoPE), and Multi-Query Attention (MQA) or Grouped-Query Attention (GQA) each introduce pitfalls for activation patching, QK/OV decomposition, and head-wise interventions. The next subsection details these implications and the mitigations adopted.

Architectural considerations and mitigations

The adoption of Gemma-2b-it introduces three principal architectural features that complicate standard mechanistic interpretability workflows. Each requires explicit mitigation to ensure reliable attribution and causal analysis.

First, `RMSNorm` replaces layer normalization. Unlike LayerNorm, which centres activations to zero mean, RMSNorm only rescales by root-mean-square. This asymmetry means that patching intermediate activations can alter scale in unintended ways, potentially inflating or suppressing logit differences independent of the causal intervention. To mitigate this, the `fold_ln=True` flag is used during model loading, which absorbs normalization weights into surrounding linear layers. This transformation preserves model outputs whilst simplifying the residual stream geometry, rendering activation patching more reliable.

Second, Gemma employs rotary positional embeddings (RoPE) rather than the additive positional encodings used in GPT-2. RoPE encodes position through rotation matrices applied to query and key vectors, making them position-dependent in a multiplicative sense. Consequently, patching Q or K vectors between prompts or positions risks breaking the geometric assumptions underlying attention patterns. The standard flag `refactor_factored_attn_matrices=True`, which orthogonalizes query and key projections, cannot be safely applied with RoPE. Instead, interventions on attention patterns and values are preferred over direct manipulation of Q and K, since attention patterns are position-agnostic once computed.

Third, Gemma uses multi-query attention (MQA), wherein multiple query heads share a single set of key and value heads. This architectural choice reduces parameters and computation but

introduces a dependency: patching value vectors affects all query heads simultaneously, and decomposing the circuit at the level of individual heads becomes ambiguous. To address this, the analysis focuses on head-wise output patching rather than per-head value patching where MQA complicates interpretation. When value-level analysis is necessary, results are interpreted with explicit acknowledgement of this shared structure. Additionally, the `get_act_patch_attn_head_all_pos_every` function, which performs activation patching with head decomposition, was modified to properly handle MQA architectures, ensuring that patching operations on shared key and value heads are correctly attributed across query heads.

The model is loaded via `TransformerLens` using `from_pretrained_no_processing()` to preserve the native architecture. The `fold_ln=True` flag is applied at load time. These choices balance mechanistic tractability against architectural fidelity, enabling reliable causal intervention whilst respecting the model's actual design.

7.1.3 Metric

Next, model performance is quantified using the logit-difference at the answer position. Let z denote the logits over the vocabulary, and let c and i index the correct and contrast tokens, respectively. The metric is

$$\mathrm{LD} = z_c - z_i.$$

Although the model is trained to optimise cross-entropy (log probabilities), softmax is shift-invariant, so the normalising constant cancels in differences:

$$\log p(c) - \log p(i) = \left(z_c - \log \sum_w e^{z_w}\right) - \left(z_i - \log \sum_w e^{z_w}\right) = z_c - z_i = \mathrm{LD}.$$

Hence log-probability difference equals logit difference. Positive values indicate a preference for the correct token, negative values indicate a preference for the contrast token, and zero indicates neutrality. Unless stated otherwise, results are averaged over prompts and evaluated at the first answer token.

The prompts investigated are illustrated in Table 7.1, together with the corresponding logit differences; the incorrect answer is defined as the value in the adjacent column to the right within the same row of the table.

This choice reflects empirical observations on analogous prompts with larger tables, where the model frequently returns the value from the neighbouring cell rather than the correct entry, typically the cell in the next column to the right. Defining the contrast in this way targets the dominant failure mode, and yields a sensitive, interpretable logit-difference signal.

Table 7.1: Logit differences. *Note:* To maintain compactness, the prompts following the first row use the placeholder `TABLE` in place of the full table content, which remains unchanged from the first row.

Prompt	**Correct**	**Incorrect**	**Logit Difference**
Look at the table below and answer the question. Bank | IRC | ECL | ROE Red | 4 | 7 | 8 Blue | 5 | 9 | 3 Gold | 6 | 1 | 2 Question: What is the IRC for Bank Red? Answer:	4	7	11.127
Look at the table below and answer the question. TABLE Question: What is the ECL for Bank Red? Answer:	7	8	12.862
Look at the table below and answer the question. TABLE Question: What is the IRC for Bank Blue? Answer:	5	9	7.173
Look at the table below and answer the question. TABLE Question: What is the ECL for Bank Blue? Answer:	9	3	11.165
Look at the table below and answer the question. TABLE Question: What is the IRC for Bank Gold? Answer:	6	1	6.858
Look at the table below and answer the question. TABLE Question: What is the ECL for Bank Gold? Answer:	1	2	8.407

7.1.4 Logit lens

The logit lens [92] projects intermediate residual stream representations at each layer into vocabulary space via the unembedding, thereby enabling tracking of when, and where, the correct answer emerges in the computational process. It is used here to quantify how each transformer block, and its constituent attention heads, contributes to the logit difference in the residual stream, by decomposing the accumulated signal into layer-wise and head-wise terms, and identifying components that

enhance or inhibit the correct-answer direction. Figure 7.1 reports the layer-wise decomposition across depth.

For layer n, n-pre denotes the residual at the start of the layer, whereas n-mid denotes the residual immediately after the attention sublayer. The vertical separation between n-pre and n-mid measures the attention contribution; the step from n-mid to (n+1)-pre measures the MLP contribution. The sign of the logit difference indicates preference at that point:

- $\text{LogitDiff}^{(\ell)} > 0$: layer ℓ favours the correct answer,
- $\text{LogitDiff}^{(\ell)} < 0$: layer ℓ favours the incorrect answer,
- $\text{LogitDiff}^{(\ell)} = 0$: layer ℓ is neutral.

It is observed that the model contributes negligibly to the task before layer 14. The dominant improvement is produced by the attention sublayer of layer 14, and the cumulative signal degrades slightly after layer 16.

This pattern indicates that layer 14 contains the primary mechanism writing the correct direction into the residual stream, which motivates targeted analysis of the respective roles of the attention and MLP sublayers and the specific attention heads that drive the behaviour.

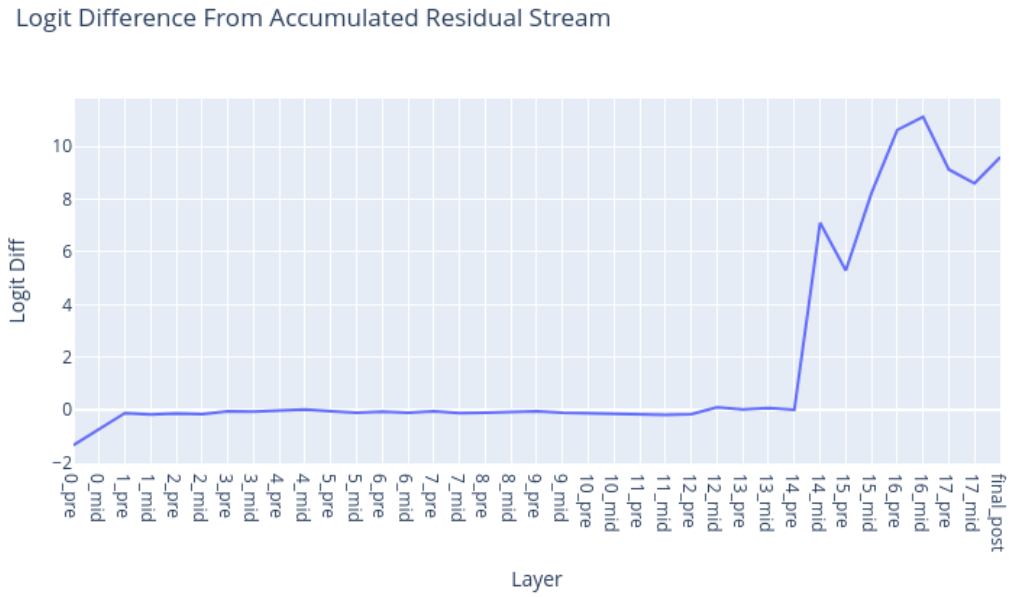


Figure 7.1: Calculated logit difference for the decomposed accumulated residual stream after each layer. n-pre denotes the residual stream at the start of layer n, while n-mid denotes the residual stream after the attention part of layer n.

The logit difference between adjacent residual streams is examined to attribute marginal contributions by sublayer (Figure 7.2). The largest positive steps arise from the attention sublayers of layers 14 and 15, in line with the cumulative analysis above. In contrast to the IOI task [130], where

attention dominates, the MLP sublayers make consequential contributions in this setting. Notably, MLP 14 and MLP 16 reduce the logit difference, indicating inhibitory effects on the correct-answer direction.

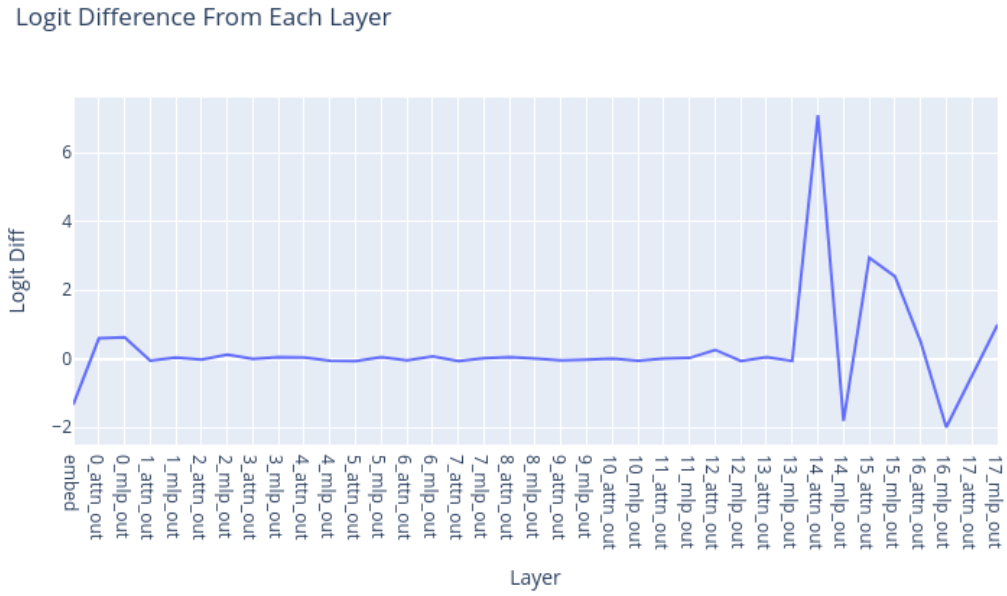


Figure 7.2: Break down of logit differences from each layer between adjacent residual streams.

Next, the output of each attention layer is decomposed into the sum of its constituent heads, permitting head-wise attribution of the logit difference (Figure 7.3). Using the notation l.h to denote head h in layer l, with zero-based indexing, heads 14.0, 14.4, and 15.1 exhibit the largest positive contributions, indicating that these components play a central role in the table lookup mechanism.

7.1.5 Activation Patching

One of the limitations of the direct logit attribution is that it only looks at the very end of the circuit which affects the logits directly [81, 130]. To obtain a more refined understanding, in what follows, we utilise the activation patching technique [82]. In doing so, we design matched pairs of clean and corrupted prompts. The prompt format remains exactly as depicted above. Each clean prompt is a table-lookup question that specifies a column and a bank; its corresponding corrupted prompt alters only the queried column to the adjacent column to the right within the same row, holding all other context fixed. We construct six clean prompts, spanning the first two columns (IRC and ECL) across the three banks, and six corresponding corrupted prompts.

In the clean run, the model assigns higher logits to the correct answer, producing a positive logit difference; in the corrupted run, it favours the contrast answer, producing a negative logit difference. Overall the average clean logit diff is 9.599 and the average corrupted logit diff is -6.026. Activation patching then executes the model on the corrupted prompts and replaces a selected intermediate

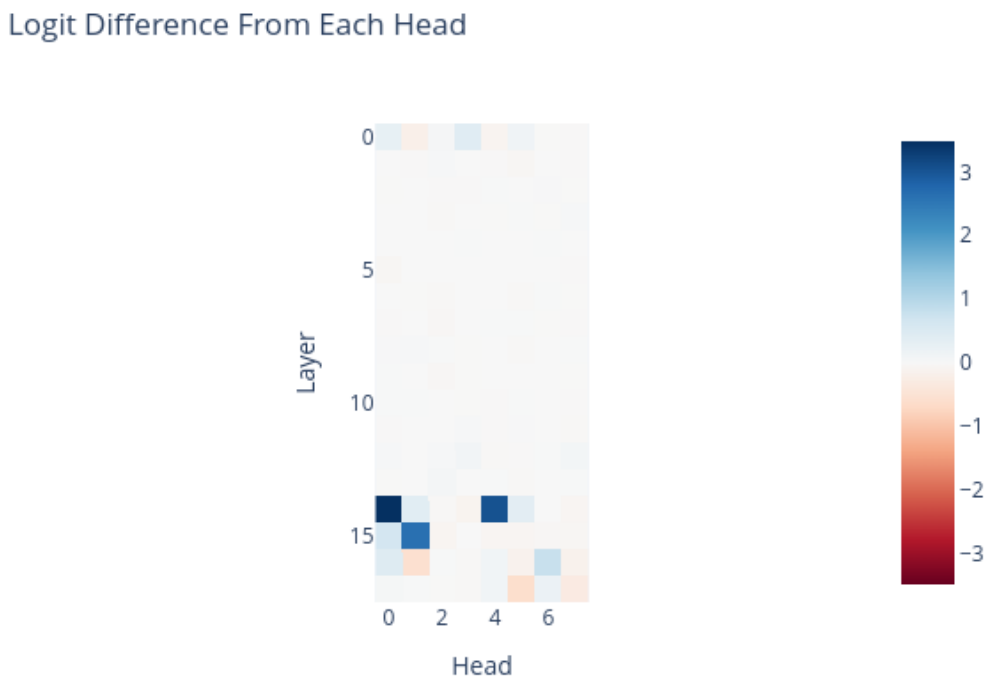


Figure 7.3: Attention heads heat map per layer illustrating logit difference from each head.

activation with its counterpart from the clean run. This denoising intervention (causal tracing) assesses whether that component is sufficient to reinstate clean run behaviour, measured by a shift of the logit difference towards the correct token. Figure 7.4 depicts the results for residual stream patching at the start of each layer. The full prompt spans 91 token positions, but the figure reports a zoom on the final tokens of the prompt, because scores for earlier tokens are consistently zero across layers. A score closer to 0 means the performance is closer to the one obtained on the corrupted input, while a score closer to 1 means that the performance is closer to that of the clean input. We are trying to look for activations that are sufficient to recover the correct response. The patching map indicates: (i) computation is highly localised to the `IRC` and `END` positions, with near zero scores elsewhere; (ii) information needed to distinguish the correct cell from the adjacent right hand cell is first stored at `IRC` and then transferred to `END` by layers 12 to 14, without detours; (iii) patching the residual stream at the correct position nearly exactly recovers clean run performance; (iv) the model is effectively finished after layer 14, and later layers provide only slight positive refinements.

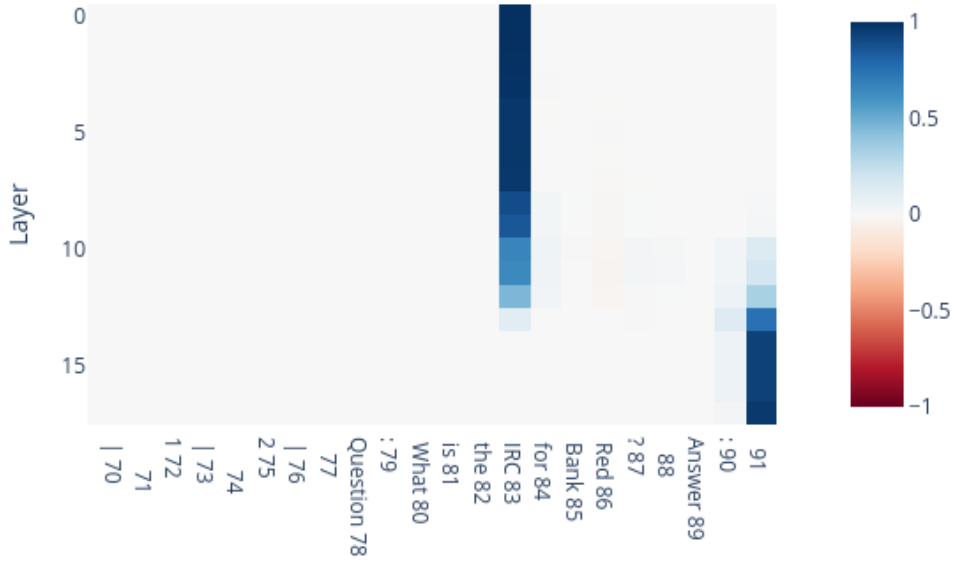


Figure 7.4: Residual stream patching at the onset of each layer across token positions, averaged over six prompts. The y-axis represents the layer number (0 to 17), and the x-axis represents token positions. Note: for reference, tokens and their indices from the first prompt are labelled on the x-axis. In a slight abuse of notation, the plotted differences are averaged over all prompts, while the labels are taken from the first prompt only.

Beyond patching the residual stream at the start of each layer, activation patching can be applied immediately after the attention sublayer and after the MLP sublayer. This approach provides a more granular view of the contributions made by each component. Figure 7.5 presents the results of these interventions. The analysis reveals that several attention layers exert significant influence, with early layers primarily affecting the `IRC` token and later layers impacting the `END`

token. In contrast, the majority of MLP layers contribute minimally, with the exception of MLP 0, which exhibits a pronounced effect. This pattern suggests that the attention mechanism is chiefly responsible for transferring information required for table lookup. At the same time, the results indicate a qualitative difference from the IOI setting, where attention layers account for almost all effective computation [81, 130]. In the present task, a subset of mid to late MLP sublayers measurably alters the logit difference, including both supportive and inhibitory effects observed in the layer-wise analysis. This supports the interpretation that table lookup is not solved purely by routing information via attention. Some processing within MLPs appears necessary, plausibly to compose row and column cues, and to check that the retrieved cell value is plausible given the prompt context.

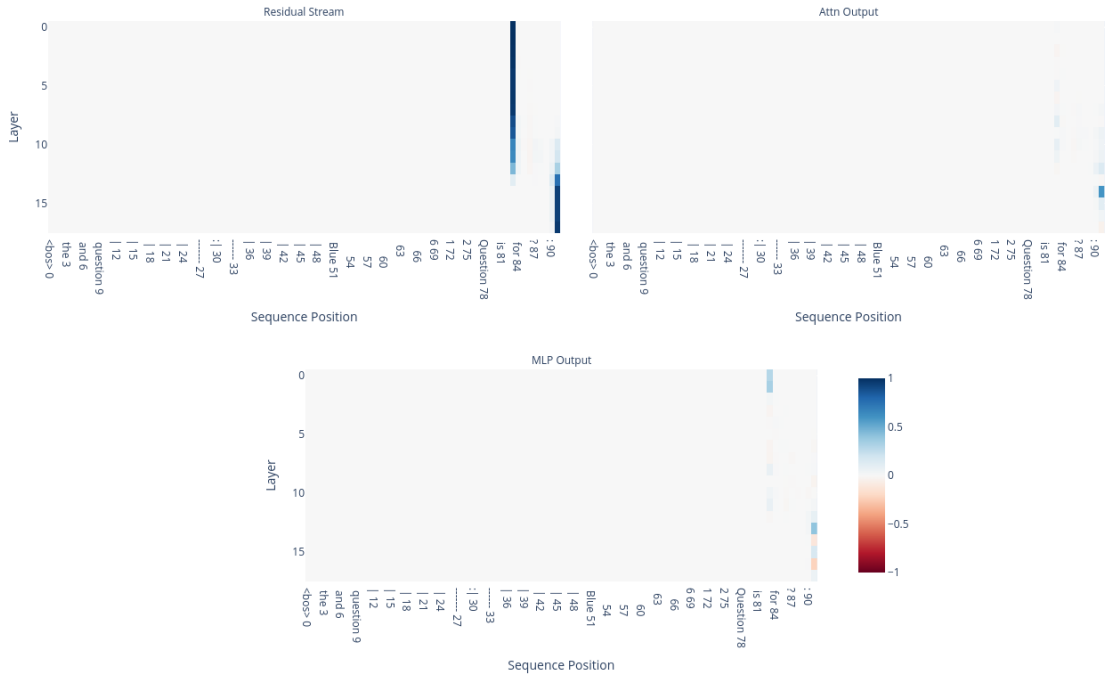
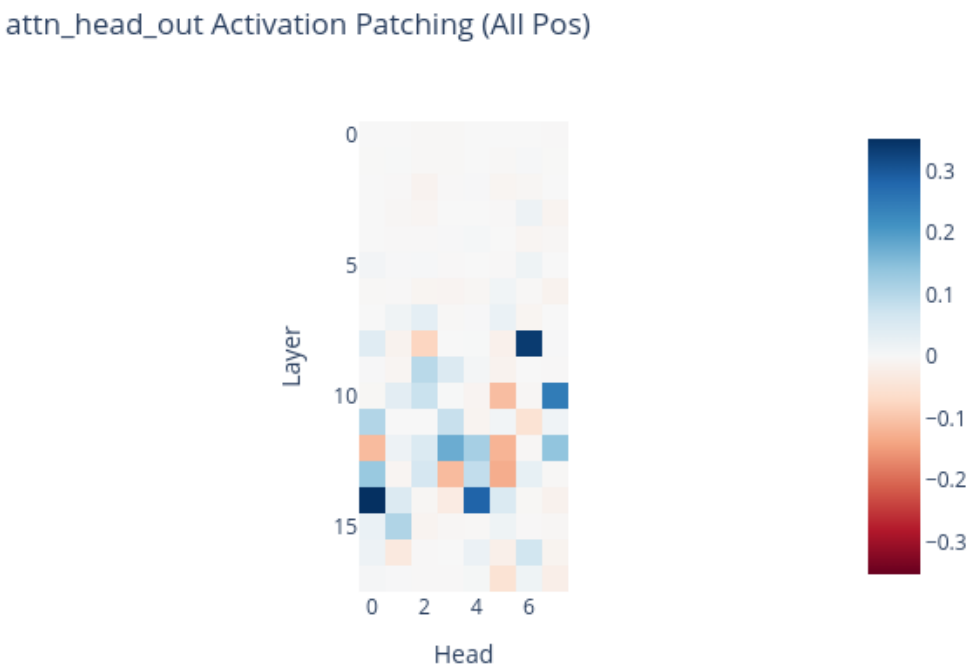


Figure 7.5: Activation patching results for attention and MLP sublayers across layers and token positions, averaged over six prompts. The figure illustrates the relative contribution of each sublayer to restoring clean run behaviour when patched, highlighting the localisation of computation in attention layers and the limited role of most MLPs.

Next, we can further refine our analysis (see Figure 7.6) by patching in on individual attention heads for all layers and tokens. We see that the heads 14.0, and 14.4, from later layers, and the heads 8.6, 10.7, and 12.3, from earlier layers, have very large positive scores.

7.1.6 Validation of early heads

Early-layer attention heads frequently implement distribution-agnostic sequence operations that later layers reuse for task-specific routing. In the present setting, we probe such heads on duplicated




```
Top 10 most positive values:  Top 10 most positive values:
 1. Layer 14, Head 0: 0.3518   1. Layer 14, Head 0: 0.3518
 2. Layer  8, Head 6: 0.3381   2. Layer  8, Head 6: 0.3381
 3. Layer 14, Head 4: 0.2839   3. Layer 14, Head 4: 0.2839
 4. Layer 10, Head 7: 0.2478   4. Layer 10, Head 7: 0.2478
 5. Layer 12, Head 3: 0.1758   5. Layer 12, Head 3: 0.1758
 6. Layer 12, Head 7: 0.1412   6. Layer 12, Head 7: 0.1412
 7. Layer 13, Head 0: 0.1327   7. Layer 13, Head 0: 0.1327
 8. Layer 12, Head 4: 0.1188   8. Layer 12, Head 4: 0.1188
 9. Layer 15, Head 1: 0.1075   9. Layer 15, Head 1: 0.1075
10. Layer 11, Head 0: 0.1051  10. Layer 11, Head 0: 0.1051
```

Figure 7.6: Activation patching results for individual attention heads across layers and token positions, averaged over six prompts. The heatmap highlights heads with substantial positive or negative contributions to restoring clean run behaviour, indicating their respective roles in the table lookup mechanism.

random-token sequences to characterise previous-token, induction, and duplicate-token behaviours. Head 8.6, which appeared as important in both activation-patching analyses above, exhibits the canonical duplicate-token signature: attention from each position in the second copy concentrates on the matching position in the first copy, yielding a near-unity score at offset n. This supports the interpretation that 8.6 functions as a copying mechanism that propagates information across repeated identifiers, a capability that can be recruited when moving table entries to the answer position.

There are three kinds of heads which appear early in the circuit, and these can be validated by inspecting their attention patterns on simple prompt ad hoc: duplicated random sequences of tokens (e.g [A,B,C,A,B,C]).

In this setting, previous-token heads attend to the immediately preceding position; induction heads, in duplicated sequences, connect the second occurrence of a token to the token that followed its first occurrence; duplicate-token heads link a repeated token to its earlier copy.

We can validate them all at the same time, using sequences of n random tokens followed by those same n random tokens repeated. This works as follows:

- Previous-token heads: measure attention with an offset of one (one below the diagonal).
- Induction heads: measure attention with an offset of $n - 1$ (the second instance of a token attends to the token after its first instance).
- Duplicate-token heads: measure attention with an offset of n (a token attends to its previous instance).

In all three cases, if heads score has a high value on these metrics, it is strong evidence that they are working as this type of head.

Note, it is a leaky abstraction to say things like "head X is an induction head", since we are only observing it on a certain distribution. For instance, it is not clear what the role of induction heads and duplicate-token heads is when there are no duplicates (they could in theory do something completely different). In this analysis, head 8.6 consistently exhibits the behaviour of a duplicate-token head (Figure 7.7). When probed on the duplicated random-token sequence, its attention pattern aligns with the canonical signature: each position in the second half of the sequence attends strongly to the corresponding position in the first half, effectively copying information from earlier occurrences. Crucially, this behaviour is not confined to the synthetic validation setting. In the table lookup task, head 8.6 also demonstrates a similar copying mechanism, facilitating the transfer of relevant cell values from their original positions in the table to the answer position. This dual role across both controlled and task-relevant prompts supports the interpretation that head 8.6 implements a general-purpose copying circuit, which the model recruits for structured retrieval in table-based question answering.

7.1.7 Activation patching with row-corrupted prompts

To further probe the mechanisms underlying table lookup, activation patching is extended to a complementary corruption scheme. Instead of altering the queried column, the corrupted prompt now modifies the queried row, selecting the bank immediately below the original target while keeping the column fixed. This approach tests whether the model's internal circuit generalises across both axes of the table and whether the same components are responsible for binding row and column cues.

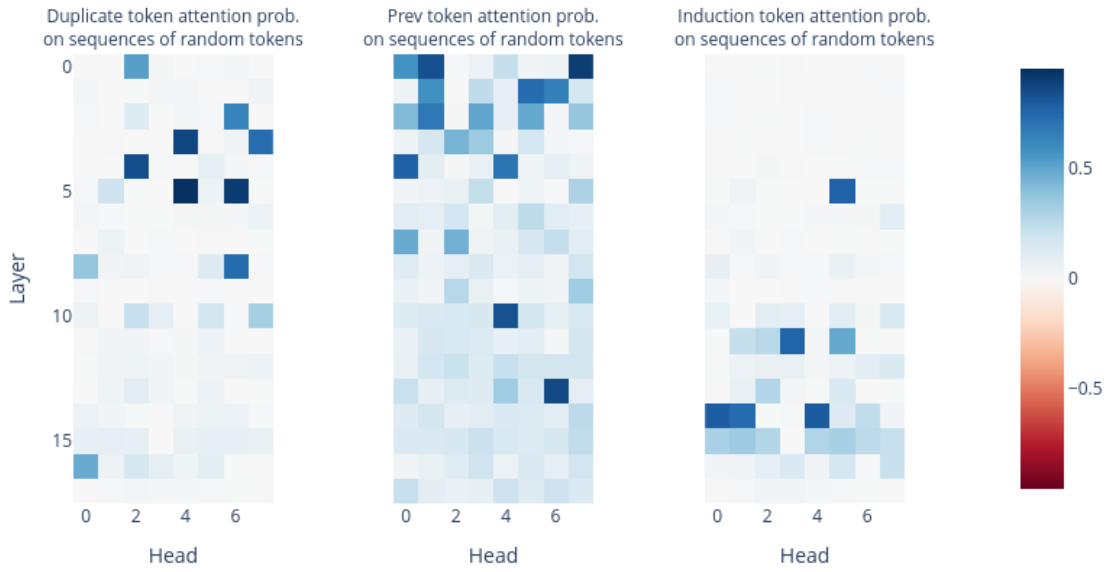


Figure 7.7: Attention-pattern validation for early heads (previous-token, induction, and duplicate-token) on duplicated random sequences.

Matched pairs of clean and row-corrupted prompts are constructed analogously to the column-corrupted setting. For each clean prompt, the corresponding corrupted prompt replaces the bank identifier in the question with the next bank in the table. The answer cell thus shifts vertically, and the correct and contrast tokens are defined accordingly. Analysing the attention pattern at the final token of the prompt reveals that attention is primarily directed to the row identifiers and the column headers. This behaviour, illustrated in Figure 7.8, appears to suggest that head 12.3 may contribute to binding the row and column identifiers to the target cell value during retrieval. Similarly, head

Figure 7.8: Attention pattern for head 12.3, illustrating its role in propagating information during the table lookup task. The figure highlights the specific tokens attended to by this head at the final token of the prompt.

11.0 demonstrates increased importance in the row-corrupted analysis, further highlighting its role in the table lookup mechanism. In contrast, head 8.6, which was prominent in the column-corrupted ranking, appears slightly lower in the ranking for row-corrupted prompts, suggesting a reduced but still significant contribution.

The localisation of computation to the answer and queried bank positions persists, and the same circuit elements are largely recruited to propagate information along the row axis. These findings support the interpretation that the model implements a generalised table lookup mechanism, capable of binding both row and column identifiers to the correct cell value. The observed differences in head-wise contributions across the two corruption schemes provide additional nuance to the causal

attribution of these components to the underlying circuit.

7.1.8 Activation patching for decomposed attention head

To refine localisation, attention heads were decomposed into the two functional subcircuits conventionally associated with attention: the QK circuit, which determines where information is routed, and the OV circuit, which determines what information is communicated. The causal contribution of each component was assessed by activation patching applied to individual subcomponents of a head (see Figure 7.10).

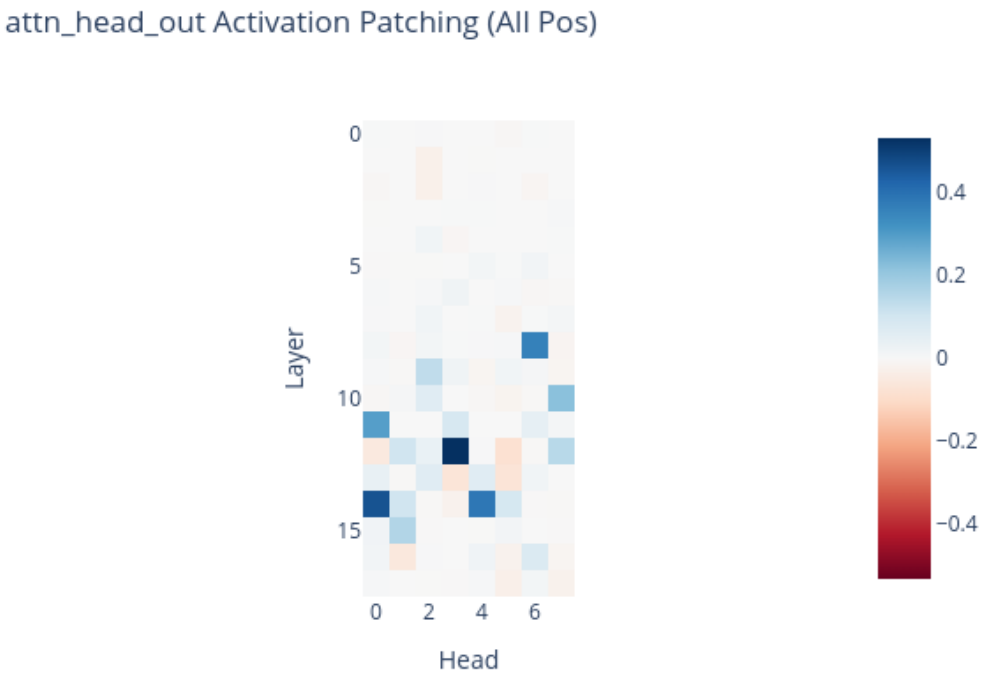
- **Output.** Patching the head output substitutes the final vector written to the residual stream. This probes the combined effect of where the head attends and what it writes.
- **Queries.** Patching query vectors tests how a token requests information, that is, whether the head's interrogative representation is critical for retrieval.
- **Keys.** Patching key vectors examines how a token advertises its content, that is, whether the representation that makes a token findable drives the head's contribution.
- **Values.** Patching value vectors assesses the causal role of the content moved by the head, independent of the attention pattern.
- **Pattern.** Patching the attention weights isolates the effect of where the head looks, while preserving the corrupted run's values.

Each patching mode was applied independently. The resulting interventions provide complementary evidence about whether a head contributes via its attention pattern, via the content it transports, or via the query/key representations that enable matching.

The decomposition yields three findings: (i) heads in layers 14 and 15 contribute primarily via their attention patterns; (ii) heads in layer 12 exert their influence also with their value vectors; and (iii) in later layers, value patching has a larger effect than key or query patching, indicating contributions from the OV pathway at these depths.

7.2 Discussion

This chapter reported an exploratory mechanistic analysis of table lookup in transformer language models in pursuit of causal transparency. Following established practice in mechanistic interpretability, the investigation began with a simplified, concrete task before attempting to generalise. A compact Markdown table with single-token headers, row identifiers, and values elicited two-dimensional addressing while keeping attribution stable. Using `google/gemma-2b-it`, with explicit mitigations for RMSNorm, RoPE, and multi-query attention, the analysis combined logit lensing, direct logit attribution, and activation patching. The results located the principal computation in mid to late layers, with attention sublayers in layers 14 and 15 providing the dominant positive contributions. Head-wise analyses highlighted heads 14.0, 14.4, and 15.1, with early heads such as 8.6, 10.7, and 12.3 supporting routing and copying. Residual, attention, and MLP patching showed highly localised computation at the queried header and answer positions, and indicated that a subset of MLPs exerts both supportive and inhibitory effects. Row-corrupted prompts corroborated the binding of both row and column cues, and validation on duplicated sequences confirmed




```
Top 10 most positive values:  Top 10 most negative values:
 1. Layer 12, Head 3: 0.5315   1. Layer 12, Head 5: -0.0820
 2. Layer 14, Head 0: 0.4629   2. Layer 13, Head 5: -0.0711
 3. Layer 14, Head 4: 0.3838   3. Layer 13, Head 3: -0.0677
 4. Layer  8, Head 6: 0.3611   4. Layer 16, Head 1: -0.0577
 5. Layer 11, Head 0: 0.2953   5. Layer 12, Head 0: -0.0522
 6. Layer 10, Head 7: 0.2218   6. Layer 17, Head 5: -0.0317
 7. Layer 15, Head 1: 0.1585   7. Layer  1, Head 2: -0.0284
 8. Layer 12, Head 7: 0.1478   8. Layer  2, Head 2: -0.0284
 9. Layer  9, Head 2: 0.1357   9. Layer 17, Head 7: -0.0239
10. Layer 14, Head 1: 0.1065  10. Layer 14, Head 3: -0.0224
```

Figure 7.9: Activation patching results for individual attention heads across layers and token positions, averaged over **row-corrupted prompts**. The heatmap highlights heads with substantial positive contributions to restoring clean run behaviour, demonstrating the generalisation of the table lookup circuit across both row and column axes.

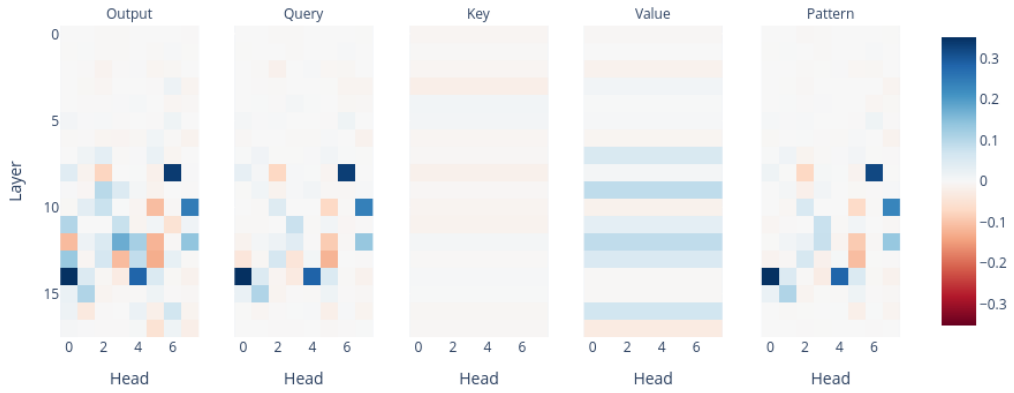


Figure 7.10: Activation patching results for decomposed attention head components (output, query, key, value, and pattern). The figure illustrates the differential impact of patching each component on restoring clean run behaviour. Note that for value and key, the same value is shared across all heads of the layer due to Gemma-2b-it use of multi-query attention (MQA).

head 8.6 as a duplicate-token copying mechanism. Taken together, these findings provide a preliminary circuit-level account of how the model performs table lookup under controlled conditions.

Future directions The next stage will deepen causal claims and test generality:

- Apply path patching to trace end-to-end routes from query tokens to the answer position, isolating pathways that carry the correct signal.
- After identifying a candidate circuit, perform targeted ablation to evaluate whether it is faithful (performs as well as the full model), complete (contains all nodes used for the task), and minimal (excludes nodes irrelevant to the task).
- Replicate the analysis on an alternative open-source model to assess portability of the mechanism across architectures.
- Vary and extend the table setting to probe reliance on positional heuristics versus genuine row–column composition, including table size, format (Markdown versus HTML), and header semantics.

These steps will move the analysis from exploratory evidence toward a rigorous, causally validated account of the table-lookup circuit.

Connection to the thesis narrative Within the overall thesis, the mechanistic analysis developed here abstracts the numeric question answering and table-based retrieval settings studied in Chapters 4–6. The controlled table-lookup task distils the error modes observed in the RAG experiments on regulatory reports, such as near-column confusions in financial tables, into a form that can be probed at circuit level. Conversely, the constraints of those applied studies, including the need for accountable numeric answers and explicit evidence traces, shape the choice of prompts, metrics,

and models in this chapter. Mechanistic findings therefore feed back into the design of grounded financial RAG systems, indicating which internal components are responsible for structured lookup and which may be targets for future constraint or monitoring.

Summary This chapter completed the transparency stack by advancing from referential to causal transparency. Under controlled conditions that mirror financial question answering, the analysis identified circuit components that support two-dimensional addressing in table lookup, linking attention heads and multilayer perceptrons to the subroutines of locating, composing, and propagating the correct value. Together with the evidence-centred annotation and Retrieval-Augmented Generation pipelines developed earlier, these findings connect outputs, evidence, and mechanisms within a single methodological continuum.

The results remain exploratory, yet they provide an operational route from grounded behaviour to internal computation, which is necessary for accountable systems in finance. They also clarify where mechanistic insight can inform design, evaluation, and governance, for example by guiding targeted ablations, constraint mechanisms, or diagnostics for numeric reasoning.

Chapter 8 synthesises the contributions across the three pillars, articulates their implications for trustworthy financial AI, and outlines a research agenda for scalable mechanistic analysis and interpretable retrieval-augmented generation under deployment constraints.

Chapter 8

Conclusions

Transparent and reliable AI for finance requires more than accurate outputs. It requires systems whose behaviour can be understood, whose claims can be traced to evidence, and whose internal computations can be examined. This thesis advanced a unified programme to meet this requirement, articulated as a transparency stack with three complementary layers: Explainable AI for empirical transparency, Grounded Generation for referential transparency, and Mechanistic Interpretability for causal transparency. Read together, these layers define a methodological continuum that links what models output, on what evidence they rely, and how they internally produce their decisions.

A unified vision: the transparency stack The transparency stack is both an organising principle and a practical framework. It treats transparency as a property that spans outputs, evidence, and mechanisms, rather than a single post hoc explanation. The first layer, Explainable AI (XAI), quantifies explanation quality for predictive models. The second layer, Grounded Generation, constrains language models with retrieval and provenance, ensuring that answers are auditable against source documents. The third layer, Mechanistic Interpretability, examines internal components and pathways, aiming to relate behaviours to identifiable circuits. The result is a coherent approach in which interpretability, verification, and causal analysis reinforce one another.

Contributions within a single programme The contributions of the thesis were designed to interlock.

First, a quantitative framework for explainability transformed interpretability into a measurable attribute. Metrics such as Effective Compactness, Rank Quality Index, and Stability were defined and evaluated across classification, regression, and anomaly detection. These measures make explanation faithfulness and reproducibility auditable, providing a defensible basis for model risk management and regulatory assurance.

Second, the thesis extended transparency to generative settings. It introduced an evidence-centred workflow for visually rich financial documents, including keyword-based annotation and provenance-preserving descriptions for text, tables, and figures. Building on this substrate, it developed and evaluated Retrieval-Augmented Generation pipelines that assess answer synthesis, and record citations. The work released a financial QA benchmark and demonstrated that contextual enrichment and domain-strong embeddings improve retrieval quality, while reasoning-oriented generators increase accuracy for numeric extraction under explicit latency and cost constraints.

Third, the thesis initiated a mechanistic account of structured reasoning in language models. Under controlled table lookup tasks that mirror financial question answering, it applied Logit Lens, Direct Logit Attribution, and Activation Patching to identify components necessary for two-dimensional addressing. The analysis linked attention heads and multilayer perceptrons to subroutines such as locating rows and columns, composing keys, and propagating the correct value to the output. Although exploratory, these results establish an operational route from grounded behaviour to internal computation.

Synthesis: linking outputs, evidence, and mechanisms Considered jointly, the three layers deliver complementary assurances. XAI quantifies whether explanations faithfully reflect model behaviour. Grounded Generation ensures that answers are tied to verifiable passages, improving factual reliability and auditability on complex, multimodal documents. Mechanistic Interpretability begins to show how those behaviours arise from internal structure, supporting principled diagnostics and targeted interventions. The combined effect is a scientific standard for transparency that spans user-facing explanations, document-level provenance, and circuit-level analysis. This standard aligns with the preface's argument that trust in financial AI depends on intelligibility, traceability, and stability under realistic constraints.

Limitations and a research agenda The work also delineates clear boundaries that motivate a forward agenda consistent with the stack. Other dimensions of trustworthy AI in finance fall outside the perimeter of this thesis. These include fairness and non-discrimination; privacy and data protection; robustness and adversarial resilience; model risk management and governance; and sustainability and computational efficiency. These aspects are acknowledged where relevant, but their technical development and evaluation are not treated as core contributions.

- Scalable mechanistic analysis. Extend circuit discovery and validation to larger, multimodal, and reasoning models through modular instrumentation, hierarchical abstractions, and standardised protocols for patching and ablation, while maintaining interpretive resolution.
- Interpretable Retrieval-Augmented Generation. Trace how retrieved passages influence intermediate representations during generation, linking retrieval events to internal computations, and closing the gap between referential and causal transparency.
- Governance for agentic workflows. Translate mechanistic insights into constraints, monitors, and interventions suitable for planning, memory, and tool use in financial agents, aligning with institutional risk management.
- Human–AI interpretability interfaces. Integrate quantitative metrics and mechanistic findings into practitioner-facing reporting, so that transparency becomes actionable for auditors, compliance teams, and model validators.
- Cross-cutting trustworthiness dimensions. Integrate fairness assessment, privacy-preserving techniques, robustness and adversarial evaluation, model risk governance controls, and sustainability and computational efficiency metrics into the transparency stack, with protocols to quantify trade-offs with accuracy, latency, and transparency.

Closing perspective The thesis set out to reconcile performance with transparency in financial AI. By operationalising explainability metrics, delivering auditable grounded generation on visually rich documents, and initiating a causal account of a core reasoning task, it advances a unified vision in which the three layers of the transparency stack function together. The resulting methodology enables stakeholders to assess what models communicate, verify what they claim against evidence, and probe how they compute. This coherence is the central contribution: a pathway toward AI systems that are not only accurate and efficient, but also comprehensible, verifiable, and governable in the financial domain.

List of Acronyms

AE Autoencoder. 47

AI Artificial Intelligence. 2

API Application Programming Interface. 48

AUDC Area Under the Deletion Curve. 42

AVA Additional Valuation Adjustments. 81

CET1 Common Equity Tier 1. 81

CNN Convolutional Neural Network. 62

DCG Discounted Cumulative Gain. 85

EBA European Banking Authority. 9

EC Effective Compactness. 41, 42

ECB European Central Bank. 9

EU AI Act European Union Artificial Intelligence Act. 2, 9

GAAP Generally Accepted Accounting Principles. 80

GDPR General Data Protection Regulation. 9

GPT Generative Pretrained Transformer. 61

GQA Grouped-Query Attention. 92

ICT Information and Communication Technology. 63

IF Isolation Forest. 47

IFRS International Financial Reporting Standards. 80

IG Integrated Gradients. 47

IOI Indirect Object Identification. 36, 91

LGBM Light Gradient Boosting Machine. 47

LIME Local Interpretable Model-agnostic Explanations. 49

Llama Large Language Model Meta AI. 61

LLM Large Language Model. 2, 20

ML Machine Learning. 2, 41

MQA Multi-Query Attention. 92

MRR Mean Reciprocal Rank. 64

NDCG@k Normalised Discounted Cumulative Gain at k. 84

NLP Natural Language Processing. 2

OV Output Value. 36, 91

P@K Precision at K. 63

PDF Portable Document Format. 60

PPO Proximal Policy Optimisation. 22, 111

QK Query-Key. 36, 91

R@K Recall at K. 64

RAG Retrieval-Augmented Generation. 4, 20

RBF Radial Basis Function. 51

RL Reinforcement Learning. 21

RLHF Reinforcement Learning from Human Feedback. 21

RM Reward Model. 22, 111

RMSE Root Mean Square Error. 50

RMSNorm Root Mean Square Normalisation. 92

ROC AUC Receiver Operating Characteristic, Area Under the Curve. 51

RoPE Rotary Positional Embeddings. 92

RQI Rank Quality Index. 41

RRF Reciprocal Rank Fusion. 20

SFT Supervised Fine-Tuning. 21, 22, 111

STB Stability. 41

SVM Support Vector Machine. 47

TF-IDF Term Frequency–Inverse Document Frequency. 20

TRA Trend and Risk Analysis. 70

VRDs Visually-Rich Documents. 70

XAI Explainable Artificial Intelligence. 8

List of Figures

List of Tables

Acknowledgements

First and foremost, my deepest gratitude goes to my supervisors, André Panisson and Alan Perotti, for their constant support throughout this PhD. Their guidance, reliability, and ability to provide incredibly sharp and thoughtful feedback, especially during the moments when I felt stuck for days, have been truly invaluable. Our weekly meetings were not only productive but also genuinely enriching, and I will miss them greatly.

I would also like to thank Professor Luca Cagliero and his research group for their valuable collaboration, as well as for their support and openness throughout our work together.

I owe a great deal to my managers, Luigi, Laura, Silvia, and Viviana, who believed in me from the very beginning and gave me the opportunity to take on this journey. This experience has helped me grow immensely, both professionally and personally.

I also wish to thank the reviewers for their time, careful reading, and valuable feedback. Their comments have significantly contributed to improving the quality of this thesis.

I am deeply grateful to my family, and to my friends and colleagues, who supported me along the way. This PhD has been as much a personal journey as an academic one, and I truly appreciate everyone who encouraged me, listened to me, and stood by me throughout it.

A special thank you goes to Alessandro, who has always given me the freedom to grow and fulfil myself, even when it meant making sacrifices. My gratitude also extends to Matteo, with whom I hope to share some of the best lessons this journey has taught me. Finally, Pepe, my dog, deserves a mention for always being there when I needed it most.

Bibliography

[1] Agarwal, C., Krishna, S., Saxena, E., Pawelczyk, M., Johnson, N., Puri, I., Zitnik, M., Lakkaraju, H.: Openxai: Towards a transparent evaluation of model explanations. Advances in Neural Information Processing Systems **35**, 15784–15799 (2022)

[2] Allaj, E.: Two simple measures of variability for categorical data. Journal of Applied Statistics **45**(8), 1497–1516 (Jun 2018)

[3] Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., Mané, D.: Concrete problems in ai safety (2016), https://arxiv.org/abs/1606.06565

[4] Anthropic: Introducing contextual retrieval (2024), https://www.anthropic.com/news/contextual-retrieval

[5] Azure, M.: Azure AI Document Intelligence (2024), https://azure.microsoft.com/en-us/products/ai-services/ai-document-intelligence/

[6] Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., Herrera, F.: Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. Information Fusion **58**, 82–115 (2020)

[7] Basel Committee on Banking Supervision: Basel iii: international regulatory framework for banks. Online (2023), https://www.bis.org/bcbs/basel3.htm

[8] Beauty, A.M.: Explainable ai in data-driven finance: balancing algorithmic transparency with operational optimization demands. Int J Adv Res Publ Rev **2**(6), 125–49 (2025)

[9] Becker, B., Kohavi, R.: Adult. UCI Machine Learning Repository (1996)

[10] Beltagy, I., Peters, M.E., Cohan, A.: Longformer: The long-document transformer (2020), https://arxiv.org/abs/2004.05150

[11] Bodria, F., Giannotti, F., Guidotti, R., Naretto, F., Pedreschi, D., Rinzivillo, S.: Benchmarking and survey of explanation methods for black box models. Data Mining and Knowledge Discovery **37**(5), 1719–1778 (2023)

[12] Bussmann, N., Giudici, P., Marinelli, D., Papenbrock, J.: Explainable artificial intelligence for financial services: A survey and evaluation. Frontiers in Artificial Intelligence **3**, 26 (2020)

[13] Carletti, M., Terzi, M., Susto, G.A.: Interpretable anomaly detection with diffi: Depth-based feature importance of isolation forest. Engineering Applications of Artificial Intelligence **119**, 105730 (2023)

[14] Chen, H., Janizek, J.D., Lundberg, S., Lee, S.I.: True to the model or true to the data? (2020), https://arxiv.org/abs/2006.16234

[15] Chen, H., Lundberg, S., Lee, S.I.: Explaining models by propagating shapley values of local components (2019)

[16] Chen, Y.M., Hou, X.T., Lou, D.F., Liao, Z.L., Liu, C.L.: Damgcn: Entity linking in visually rich documents with dependency-aware multimodal graph convolutional network. In: Fink, G.A., Jain, R., Kise, K., Zanibbi, R. (eds.) Document Analysis and Recognition - ICDAR 2023. pp. 33–47. Springer Nature Switzerland, Cham (2023)

[17] Chen, Z., Chen, W., Smiley, C., Shah, S., Borova, I., Langdon, D., Moussa, R., Beane, M., Huang, T.H., Routledge, B., Wang, W.Y.: FinQA: A dataset of numerical reasoning over financial data. In: Moens, M.F., Huang, X., Specia, L., Yih, S.W.t. (eds.) Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. pp. 3697–3711. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic (Nov 2021). https://doi.org/10.18653/v1/2021.emnlp-main.300, https://aclanthology.org/2021.emnlp-main.300/

[18] Chen, Z., Li, S., Smiley, C., Ma, Z., Shah, S., Wang, W.Y.: ConvFinQA: Exploring the chain of numerical reasoning in conversational finance question answering. In: Goldberg, Y., Kozareva, Z., Zhang, Y. (eds.) Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing. pp. 6279–6292. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates (Dec 2022). https://doi.org/10.18653/v1/2022.emnlp-main.421, https://aclanthology.org/2022.emnlp-main.421/

[19] Choi, C., Kwon, J., Ha, J., Choi, H., Kim, C., Lee, Y., yong Sohn, J., Lopez-Lira, A.: Finder: Financial dataset for question answering and evaluating retrieval-augmented generation (2025), https://arxiv.org/abs/2504.15800

[20] Cohere: Introducing rerank 3.5: Precise ai search (2025), https://cohere.com/blog/rerank-3pt5

[21] Conmy, A., Mavor-Parker, A., Lynch, A., Heimersheim, S., Garriga-Alonso, A.: Towards automated circuit discovery for mechanistic interpretability. In: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (eds.) Advances in Neural Information Processing Systems. vol. 36, pp. 16318–16352. Curran Associates, Inc. (2023), https://proceedings.neurips.cc/paper_files/paper/2023/file/34e1dbe95d34d7ebaf99b9bcaeb5b2be-Paper-Conference.pdf

[22] Ding, Y., Vaiani, L., Han, C., Lee, J., Garza, P., Poon, J., Cagliero, L.: 3MVRD: Multimodal multi-task multi-teacher visually-rich form document understanding. In: Ku, L.W., Martins, A., Srikumar, V. (eds.) Findings of the Association for Computational Linguistics: ACL 2024. pp. 15233–15244. Association for Computational Linguistics, Bangkok, Thailand (Aug 2024). https://doi.org/10.18653/v1/2024.findings-acl.903, https://aclanthology.org/2024.findings-acl.903/

[23] Doshi-Velez, F., Kim, B.: Towards a rigorous science of interpretable machine learning (2017), https://arxiv.org/abs/1702.08608

[24] Efron, B., Hastie, T., Johnstone, I., Tibshirani, R.: Least angle regression. The Annals of Statistics **32**(2), 407 – 499 (2004)

[25] Elhage, N., Nanda, N., Olsson, C., Henighan, T., Joseph, N., Mann, B., Askell, A., Bai, Y., Chen, A., Conerly, T., DasSarma, N., Drain, D., Ganguli, D., Hatfield-Dodds, Z., Hernandez, D., Jones, A., Kernion, J., Lovitt, L., Ndousse, K., Amodei, D., Brown, T., Clark, J., Kaplan, J., McCandlish, S., Olah, C.: A mathematical framework for transformer circuits. Transformer Circuits Thread (2021), https://transformer-circuits.pub/2021/framework/index.html

[26] European Banking Authority: Eba report on big data and advanced analytics. Tech. rep., EBA (2021), https://www.eba.europa.eu/

[27] European Union: Regulation of the european parliament and of the council laying down harmonised rules on artificial intelligence (ai act) (2024), https://eur-lex.europa.eu/, official Journal of the European Union, L 180/1, 2024

[28] EvidentlyAI: Normalized discounted cumulative gain (NDCG) explained (2025), https://www.evidentlyai.com/ranking-metrics/ndcg-metric

[29] Fan, W., Ding, Y., Ning, L., Wang, S., Li, H., Yin, D., Chua, T.S., Li, Q.: A survey on rag meeting llms: Towards retrieval-augmented large language models. In: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 6491–6501. KDD '24, Association for Computing Machinery, New York, NY, USA (2024). https://doi.org/10.1145/3637528.3671470, https://doi.org/10.1145/3637528.3671470

[30] Fanaee-T, H., Gama, J.: Event labeling combining ensemble detectors and background knowledge. Progress in Artificial Intelligence **2**, 113–127 (06 2014)

[31] Financial Accounting Standards Board: Welcome to the accounting standards codification. Online, https://asc.fasb.org/Home

[32] Gallipoli, G., Cagliero, L., Mosca, A., Miola, A., Borghi, D.: Retrieval augmented generation of summarized answers on visually-rich documents for trend and risk analysis. In: Proceedings of the Workshops of the EDBT/ICDT 2025 Joint Conference - 9th International Workshop on Data Analytics solutions for Real-LIfe APplications (DARLI-AP). vol. Vol-3946. CEUR Workshop Proceedings (2025), https://ceur-ws.org/Vol-3946/DARLI-AP-6.pdf

[33] Gallipoli, G., Papicchio, S., Vaiani, L., Cagliero, L., Miola, A., Borghi, D.: Keyword-based annotation of visually-rich document content for trend and risk analysis using large language models. In: Chen, C.C., Liu, X., Hahn, U., Nourbakhsh, A., Ma, Z., Smiley, C., Hoste, V., Das, S.R., Li, M., Ghassemi, M., Huang, H.H., Takamura, H., Chen, H.H. (eds.) Proceedings of the Joint Workshop of the 7th Financial Technology and Natural Language Processing, the 5th Knowledge Discovery from Unstructured Data in Financial Services, and the 4th Workshop on Economics and Natural Language Processing. pp. 130–136. Association for Computational Linguistics, Torino, Italia (May 2024), https://aclanthology.org/2024.finnlp-1.13

[34] Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., Wang, H.: Retrieval-augmented generation for large language models: A survey (2024), https://arxiv.org/abs/2312.10997

[35] German, B.: Glass Identification. UCI Machine Learning Repository (1987)

[36] Gilpin, L.H., Bau, D., Yuan, B.Z., Bajwa, A., Specter, M., Kagal, L.: Explaining explanations: An overview of interpretability of machine learning. In: Proceedings of the IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA). pp. 80–89. IEEE (2018)

[37] Goh, G., †, N.C., †, C.V., Carter, S., Petrov, M., Schubert, L., Radford, A., Olah, C.: Multimodal neurons in artificial neural networks. Distill (2021). https://doi.org/10.23915/distill.00030, https://distill.pub/2021/multimodal-neurons

[38] Goldi Soni, M.S.S., Kumari, M.S.: Advancing and transforming finance through artificial intelligence: Development and applications. International Journal for Research in Applied Science & Engineering Technology **13**(9) (9 2025). https://doi.org/https://doi.org/10.22214/ijraset.2025.74271

[39] Golgoon, A., Filom, K., Ravi Kannan, A.: Mechanistic interpretability of large language models with applications to the financial services industry. In: Proceedings of the 5th ACM International Conference on AI in Finance. p. 660–668. ICAIF '24, ACM (Nov 2024). https://doi.org/10.1145/3677052.3698612, http://dx.doi.org/10.1145/3677052.3698612

[40] Grinsztajn, L., Oyallon, E., Varoquaux, G.: Why do tree-based models still outperform deep learning on typical tabular data? In: Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (2022)

[41] Grootendorst, M.: Keybert: Minimal keyword extraction with bert. Zenodo (2020). https://doi.org/10.5281/zenodo.4461265, https://doi.org/10.5281/zenodo.4461265

[42] Guo, B., Zhang, X., Wang, Z., Jiang, M., Nie, J., Ding, Y., Yue, J., Wu, Y.: How close is chatgpt to human experts? comparison corpus, evaluation, and detection (2023), https://arxiv.org/abs/2301.07597

[43] Hedström, A., Weber, L., Krakowczyk, D., Bareeva, D., Motzkus, F., Samek, W., Lapuschkin, S., Höhne, M.M.C.: Quantus: An explainable ai toolkit for responsible evaluation of neural network explanations and beyond. Journal of Machine Learning Research **24**(34), 1–11 (2023)

[44] Hofmann, H.: Statlog (German Credit Data). UCI Machine Learning Repository (1994)

[45] Hooker, S., Erhan, D., Kindermans, P.J., Kim, B.: A benchmark for interpretability methods in deep neural networks. Advances in neural information processing systems **32** (2019)

[46] Hsieh, W., Bi, Z., Jiang, C., Liu, J., Peng, B., Zhang, S., Pan, X., Xu, J., Wang, J., Chen, K., Feng, P., Wen, Y., Song, X., Wang, T., Liu, M., Yang, J., Li, M., Jing, B., Ren, J., Song, J., Tseng, H.M., Zhang, Y., Yan, L.K.Q., Niu, Q., Chen, S., Wang, Y., Liang, C.X.: A comprehensive guide to explainable ai: From classical models to llms (2024), https://arxiv.org/abs/2412.00800

[47] Huang, Y., Hu, S., Han, X., Liu, Z., Sun, M.: Unified view of grokking, double descent and emergent abilities: A perspective from circuits competition (2024), https://arxiv.org/abs/2402.15175

[48] IFRS Foundation: Ifrs accounting standards navigator. Online, https://www.ifrs.org/issued-standards/list-of-standards/

[49] Iskender, N., Polzehl, T., Möller, S.: Reliability of human evaluation for text summarization: Lessons learned and challenges ahead. In: Belz, A., Agarwal, S., Graham, Y., Reiter, E., Shimorina, A. (eds.) Proceedings of the Workshop on Human Evaluation of NLP Systems (HumEval). pp. 86–96. Association for Computational Linguistics, Online (Apr 2021), https://aclanthology.org/2021.humeval-1.10

[50] Iskender, N., Polzehl, T., Möller, S.: Reliability of human evaluation for text summarization: Lessons learned and challenges ahead. In: Belz, A., Agarwal, S., Graham, Y., Reiter, E., Shimorina, A. (eds.) Proceedings of the Workshop on Human Evaluation of NLP Systems (HumEval). pp. 86–96. Association for Computational Linguistics, Online (Apr 2021), https://aclanthology.org/2021.humeval-1.10/

[51] Islam, P., Kannappan, A., Kiela, D., Qian, R., Scherrer, N., Vidgen, B.: Financebench: A new benchmark for financial question answering. CoRR **abs/2311.11944** (2023), https://doi.org/10.48550/arXiv.2311.11944

[52] Jacovi, A., Goldberg, Y.: The role of reliability in model interpretability: Measuring and improving faithfulness across explanations. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL). pp. 5564–5577 (2021)

[53] Järvelin, K., Kekäläinen, J.: Cumulated gain-based evaluation of IR techniques. ACM Transactions on Information Systems **20**(4), 422–446 (2002). https://doi.org/10.1145/582415.582418, https://api.semanticscholar.org/CorpusID:1981391

[54] Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y.J., Madotto, A., Fung, P.: Survey of hallucination in natural language generation. ACM computing surveys **55**(12), 1–38 (2023)

[55] Jiang, A.Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D.S., de las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L.R., Lachaux, M.A., Stock, P., Le Scao, T., Lavril, T., Wang, T., Lacroix, T., Sayed, W.E.: Mistral 7b (2023), https://arxiv.org/abs/2310.06825

[56] Kanaparthi, V.: Ai-based personalization and trust in digital finance (2024), https://arxiv.org/abs/2401.15700

[57] Kelley Pace, R., Barry, R.: Sparse spatial autoregressions. Statistics & Probability Letters **33**(3), 291–297 (1997)

[58] Khan, A.T., Li, S., Cao, X.: Bridging finance and ai: a comprehensive survey of large language models in financial system. Digital Finance **7**(4), 679–701 (2025). https://doi.org/10.1007/s42521-025-00146-3, https://doi.org/10.1007/s42521-025-00146-3

[59] La Quatra, M., Cagliero, L.: Bart-it: An efficient sequence-to-sequence model for italian text summarization. Future Internet **15**(1) (2023). https://doi.org/10.3390/fi15010015, https://www.mdpi.com/1999-5903/15/1/15

[60] Le, P.Q., Nauta, M., Nguyen, V.B., Pathak, S., Schlötterer, J., Seifert, C.: Benchmarking explainable ai - a survey on available toolkits and open challenges. In: Elkind, E. (ed.) Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23. pp. 6665–6673. International Joint Conferences on Artificial Intelligence Organization (8 2023), survey Track

[61] Lee, J., Stevens, N., Han, S.C.: Large language models in finance (finllms). Neural Computing and Applications **37**(30), 24853–24867 (Jan 2025). https://doi.org/10.1007/s00521-024-10495-6, http://dx.doi.org/10.1007/s00521-024-10495-6

[62] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L.: BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: Jurafsky, D., Chai, J., Schluter, N., Tetreault, J. (eds.) Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 7871–7880. Association for Computational Linguistics, Online (Jul 2020). https://doi.org/10.18653/v1/2020.acl-main.703, https://aclanthology.org/2020.acl-main.703/

[63] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.t., Rocktäschel, T., Riedel, S., Kiela, D.: Retrieval-augmented generation for knowledge-intensive nlp tasks. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. NIPS'20, Curran Associates Inc., Red Hook, NY, USA (2020)

[64] Li, Z.Z., Zhang, D., Zhang, M.L., Zhang, J., Liu, Z., Yao, Y., Xu, H., Zheng, J., Wang, P.J., Chen, X., Zhang, Y., Yin, F., Dong, J., Li, Z., Bi, B.L., Mei, L.R., Fang, J., Liang, X., Guo, Z., Song, L., Liu, C.L.: From system 1 to system 2: A survey of reasoning large language models (2025), https://arxiv.org/abs/2502.17419

[65] Lighthouz AI: New RAG benchmark for finance applications: Apple 10k 2022. Blog post (2024), https://eval.lighthouz.ai/blog/rag-benchmark-finance-apple-10K-2022/

[66] Lin, C.Y.: ROUGE: A package for automatic evaluation of summaries. In: Text Summarization Branches Out. pp. 74–81. Association for Computational Linguistics, Barcelona, Spain (Jul 2004), https://aclanthology.org/W04-1013

[67] Lipton, Z.C.: The mythos of model interpretability. In: Queue. vol. 16, pp. 31–57. ACM (2018)

[68] Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. In: Oh, A., Neumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (eds.) Advances in Neural Information Processing Systems. vol. 36, pp. 34892–34916. Curran Associates, Inc. (2023), https://proceedings.neurips.cc/paper_files/paper/2023/file/6dcf277ea32ce3288914faf369fe6de0-Paper-Conference.pdf

[69] Liu, N., Shin, D., Hu, X.: Contextual outlier interpretation. In: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18. pp. 2461–2467. International Joint Conferences on Artificial Intelligence Organization (7 2018)

[70] Liu, Y., Khandagale, S., White, C., Neiswanger, W.: Synthetic benchmarks for scientific research in explainable machine learning. In: Advances in Neural Information Processing Systems Datasets Track (2021)

[71] Llama Team: The llama 3 herd of models (2024), https://arxiv.org/abs/2407.21783

[72] Longo, L., Brcic, M., Cabitza, F., Choi, J., Confalonieri, R., Ser, J.D., Guidotti, R., Hayashi, Y., Herrera, F., Holzinger, A., Jiang, R., Khosravi, H., Lecue, F., Malgieri, G., Páez, A., Samek, W., Schneider, J., Speith, T., Stumpf, S.: Explainable artificial intelligence (xai) 2.0: A manifesto of open challenges and interdisciplinary research directions. Information Fusion **106**, 102301 (2024)

[73] Lopes, P., Silva, E., Braga, C., Oliveira, T., Rosado, L.: Xai systems evaluation: A review of human and computer-centred methods. Applied Sciences **12**(19) (2022)

[74] Lundberg, S.M., Erion, G., Chen, H., DeGrave, A., Prutkin, J.M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., Lee, S.I.: Explainable ai for trees: From local explanations to global understanding (2019), https://arxiv.org/abs/1905.04610

[75] Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017)

[76] Lundberg, S.M., Lee, S.I.: Consistent feature attribution for tree ensembles (2018)

[77] Luo, C., Shen, Y., Zhu, Z., Zheng, Q., Yu, Z., Yao, C.: Layoutllm: Layout instruction tuning with large language models for document understanding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 15630–15640 (2024)

[78] Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, Cambridge, UK (2008), http://nlp.stanford.edu/IR-book/information-retrieval-book.html

[79] Maple, C., Szpruch, L., Epiphaniou, G., Staykova, K., Singh, S., Penwarden, W., Wen, Y., Wang, Z., Hariharan, J., Avramovic, P.: The ai revolution: Opportunities and challenges for the finance sector. Tech. rep., The Alan Turing Institute and Financial Conduct Authority (8 2023). https://doi.org/10.48550/arXiv.2308.16538, https://arxiv.org/abs/2308.16538, report

[80] McDougall, C.: An analogy for understanding transformers (2023), https://www.lesswrong.com/posts/euam65XjigaCJQkcN/an-analogy-for-understanding-transformers

[81] McDougall, C.: Arena 3.0. https://github.com/callummcdougall/ARENA_3.0 (2024), gitHub repository

[82] Meng, K., Bau, D., Andonian, A., Belinkov, Y.: Locating and editing factual associations in gpt. In: Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., Oh, A. (eds.) Advances in Neural Information Processing Systems. vol. 35, pp. 17359–17372. Curran Associates, Inc. (2022), https://proceedings.neurips.cc/paper_files/paper/2022/file/6f1d43d5a82a37e89b0665b33bf3a182-Paper-Conference.pdf

[83] Microsoft Azure: Azure ai document intelligence (2024), https://azure.microsoft.com/en-us/products/ai-services/ai-document-intelligence/

[84] Miller, T.: Explanation in artificial intelligence: Insights from the social sciences. Artificial Intelligence **267**, 1–38 (2019)

[85] Mohsin, M.T., Nasim, N.B.: Explaining the unexplainable: A systematic review of explainable ai in finance (2025), https://arxiv.org/abs/2503.05966

[86] Molnar, C.: Interpretable Machine Learning. 3 edn. (2025), https://christophm.github.io/interpretable-ml-book

[87] Nanda, N.: A comprehensive mechanistic interpretability explainer & glossary. Online (2022), https://neelnanda.io/glossary

[88] Nanda, N., Bloom, J.: Transformerlens. https://github.com/neelnanda-io/TransformerLens (2022), gitHub repository

[89] Nanda, N., Chan, L., Lieberum, T., Smith, J., Steinhardt, J.: Progress measures for grokking via mechanistic interpretability. In: The Eleventh International Conference on Learning Representations (2023), https://openreview.net/forum?id=9XFSbDPmdW

[90] Nash, W., Sellers, T., Talbot, S., Cawthorn, A., Ford, W.: 7he population biology of abalone (_haliotis_ species) in tasmania. i. blacklip abalone (_h. rubra_) from the north coast and islands of bass strait. Sea Fisheries Division, Technical Report No **48** (01 1994)

[91] Nauta, M., Trienes, J., Pathak, S., Nguyen, E., Peters, M., Schmitt, Y., Schlötterer, J., van Keulen, M., Seifert, C.: From anecdotal evidence to quantitative evaluation methods: A systematic review on evaluating explainable ai. ACM Comput. Surv. **55**(13s) (jul 2023)

[92] nostalgebraist: interpreting gpt: the logit lens. Online post, AI Alignment Forum (aug 2020), https://www.alignmentforum.org/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens

[93] Olah, C., Cammarata, N., Schubert, L., Goh, G., Carter, S., et al.: Zoom in: An introduction to circuits. Distill **5**(3), e00024 (2020). https://doi.org/10.23915/distill.00024

[94] OpenAI: GPT-4 technical report. ArXiv **abs/2303.08774** (2023), https://arxiv.org/abs/2303.08774

[95] OpenAI: Gpt-4o system card (2024), https://openai.com/index/gpt-4o-system-card/

[96] OpenAI: text-embedding-3-large (2024), https://platform.openai.com/docs/models/text-embedding-3-large

[97] OpenAI: Obtaining the embeddings. OpenAI Platform Documentation (2025), https://platform.openai.com/docs/guides/embeddings#obtaining-the-embeddings

[98] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P.F., Leike, J., Lowe, R.: Training language models to follow instructions with human feedback. In: Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., Oh, A. (eds.) Advances in Neural Information Processing Systems. vol. 35, pp. 27730–27744. Curran Associates, Inc. (2022), https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf

[99] Pang, G., Cao, L., Chen, L.: Homophily outlier detection in non-IID categorical data. Data Mining and Knowledge Discovery **35**(4), 1163–1224 (Jul 2021)

[100] Pang, G., Shen, C., Cao, L., Hengel, A.V.D.: Deep learning for anomaly detection: A review. ACM Computing Surveys (CSUR) **54**(2), 1–38 (2021)

[101] Park, Y., Lee, C., Yoon, S., et al.: Cofe-rag: A comprehensive full-chain evaluation framework for retrieval-augmented generation with enhanced data diversity. In: arXiv preprint arXiv:2410.12248 (2024), http://arxiv.org/pdf/2410.12248.pdf

[102] Penedo, G., Kydlíček, H., allal, L.B., Lozhkov, A., Mitchell, M., Raffel, C., Von Werra, L., Wolf, T.: The fineweb datasets: Decanting the web for the finest text data at scale. In: Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., Zhang, C. (eds.) Advances in Neural Information Processing Systems. vol. 37, pp. 30811–30849. Curran Associates, Inc. (2024), https://proceedings.neurips.cc/paper_files/paper/2024/file/370df50ccfdf8bde18f8f9c2d9151bda-Paper-Datasets_and_Benchmarks_Track.pdf

[103] Perotti, A., Borile, C., Miola, A., Nerini, F.P., Baracco, P., Panisson, A.: Explainability, quantified: Benchmarking xai techniques. In: Longo, L., Lapuschkin, S., Seifert, C. (eds.) Explainable Artificial Intelligence. pp. 421–444. Springer Nature Switzerland, Cham (2024)

[104] Petsiuk, V., Das, A., Saenko, K.: Rise: Randomized input sampling for explanation of black-box models. In: British Machine Vision Conference (BMVC) (2018)

[105] Qdrant: An introduction to Vector databases (2024), https://qdrant.tech/articles/what-is-a-vector-database/

[106] Rackauckas, Z.: Rag-fusion: A new take on retrieval augmented generation. International Journal on Natural Language Computing **13**(1), 37–47 (Feb 2024). https://doi.org/10.5121/ijnlc.2024.13103, http://dx.doi.org/10.5121/ijnlc.2024.13103

[107] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. OpenAI blog **1**(8), 9 (2019)

[108] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research **21**(140), 1–67 (2020), http://jmlr.org/papers/v21/20-074.html

[109] Reimers, N., Gurevych, I.: Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In: Inui, K., Jiang, J., Ng, V., Wan, X. (eds.) Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 3982–3992. Association for Computational Linguistics, Hong Kong, China (Nov 2019). https://doi.org/10.18653/v1/D19-1410, https://aclanthology.org/D19-1410

[110] Ribeiro, M.T., Singh, S., Guestrin, C.: "why should i trust you?": Explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. p. 1135–1144. KDD '16, Association for Computing Machinery, New York, NY, USA (2016)

[111] Ribeiro, M.T., Singh, S., Guestrin, C.: Anchors: High-precision model-agnostic explanations. Proceedings of the AAAI Conference on Artificial Intelligence **32**(1) (Apr 2018)

[112] Saeed, W., Omlin, C.: Explainable ai (xai): A systematic meta-survey of current challenges and future opportunities. Knowledge-Based Systems **263**, 110273 (2023)

[113] Salojarvi, J., Puolamaki, K., Simola, J., Kovanen, L., Kojo, I., Kaski, S.: Inferring Relevance from Eye Movements: Feature Extraction. Publications in Computer and Information Science (Mar 2005)

[114] Santilli, A., Rodolà, E.: Camoscio: an Italian instruction-tuned LLaMA (2023)

[115] Sharma, C.: Retrieval-augmented generation: A comprehensive survey of architectures, enhancements, and robustness frontiers (2025), https://arxiv.org/abs/2506.00054

[116] Shrikumar, A., Greenside, P., Kundaje, A.: Learning important features through propagating activation differences. In: Precup, D., Teh, Y.W. (eds.) Proceedings of the 34th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 70, pp. 3145–3153. PMLR (06–11 Aug 2017), https://proceedings.mlr.press/v70/shrikumar17a.html

[117] Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: visualising image classification models and saliency maps. pp. 1–8. ICLR (2014)

[118] Sokol, K., Flach, P.: Explainability fact sheets: a framework for systematic assessment of explainable approaches. In: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency. p. 56–67. FAT* '20, ACM (Jan 2020). https://doi.org/10.1145/3351095.3372870, http://dx.doi.org/10.1145/3351095.3372870

[119] de Souza P. Moreira, G., Ak, R., Schifferer, B., Xu, M., Osmulski, R., Oldridge, E.: Enhancing q&a text retrieval with ranking models: Benchmarking, fine-tuning and deploying rerankers for rag (2024), https://arxiv.org/abs/2409.07691

[120] Sundararajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks. In: Precup, D., Teh, Y.W. (eds.) Proceedings of the 34th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 70, pp. 3319–3328. PMLR (06–11 Aug 2017)

[121] Tatsat, H., Shater, A.: Beyond the black box: Interpretability of llms in finance (2025), https://arxiv.org/abs/2505.24650

[122] Team, D.A.: Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning (2025), https://arxiv.org/abs/2501.12948

[123] Thakur, N., Reimers, N., Sani, A., Rücklé, A., Srivastava, A., Gurevych, I.: Beir: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In: Proceedings of the 2021 Conference on Neural Information Processing Systems (NeurIPS) (2021), https://arxiv.org/pdf/2104.08663.pdf

[124] Tjoa, E., Guan, C.: A survey on explainable artificial intelligence (xai): Toward medical, financial and industrial applications. IEEE Transactions on Neural Networks and Learning Systems **32**(11), 4793–4813 (2021)

[125] Touvron, H., et al.: Llama 2: Open foundation and fine-tuned chat models. CoRR **abs/2307.09288** (2023). https://doi.org/10.48550/ARXIV.2307.09288, https://doi.org/10.48550/arXiv.2307.09288

[126] Tunstall, L., Beeching, E.E., Lambert, N., Rajani, N., Rasul, K., Belkada, Y., Huang, S., Werra, L.V., Fourrier, C., Habib, N., Sarrazin, N., Sanseviero, O., Rush, A.M., Wolf, T.: Zephyr: Direct distillation of LM alignment. In: First Conference on Language Modeling (2024), https://openreview.net/forum?id=aKkAwZB6JV

[127] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017), https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[128] VoyageAI: voyage-3-large: the new state-of-the-art general-purpose embedding model (2025), https://blog.voyageai.com/2025/01/07/voyage-3-large/

[129] Wang, K.R., Variengien, A., Conmy, A., Shlegeris, B., Steinhardt, J.: Redwood research easy-transformer. https://github.com/redwoodresearch/Easy-Transformer (2022), gitHub repository

[130] Wang, K.R., Variengien, A., Conmy, A., Shlegeris, B., Steinhardt, J.: Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In: The Eleventh International Conference on Learning Representations (2023), https://openreview.net/forum?id=NpsVSN6o4ul

[131] Wang, Z., Liang, Z., Shao, Z., Ma, Y., Dai, H., Chen, B., Mao, L., Lei, C., Ding, Y., Li, H.: Infogain-rag: Boosting retrieval-augmented generation via document information gain-based reranking and filtering. ArXiv **abs/2509.12765** (2025), https://api.semanticscholar.org/CorpusID:281325755

[132] Xu, Y., Li, M., Cui, L., Huang, S., Wei, F., Zhou, M.: Layoutlm: Pre-training of text and layout for document image understanding. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. p. 1192–1200. KDD '20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3394486.3403172, https://doi.org/10.1145/3394486.3403172

[133] Xue, S., Chen, T., Zhou, F., Dai, Q., Chu, Z., Mei, H.: Famma: A benchmark for financial domain multilingual multimodal question answering (2024), https://arxiv.org/abs/2410.04526

[134] Yang, W., Li, J., Xiong, C., Hoi, S.C.H.: Mace: An efficient model-agnostic framework for counterfactual explanation (2022)

[135] Yeh, I.C., hui Lien, C.: The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. Expert Systems with Applications **36**(2, Part 1), 2473–2480 (2009)

[136] Yepes, A.J., You, Y., Milczek, J., Laverde, S., Li, R.: Financial report chunking for effective retrieval augmented generation (2024), https://arxiv.org/abs/2402.05131

[137] Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) Computer Vision – ECCV 2014. pp. 818–833. Springer International Publishing, Cham (2014)

[138] Zhang, J., Zhao, Y., Saleh, M., Liu, P.: PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In: III, H.D., Singh, A. (eds.) Proceedings of the 37th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 119, pp. 11328–11339. PMLR (13–18 Jul 2020), https://proceedings.mlr.press/v119/zhang20ae.html

[139] Zhang, Q., Hall, M., Johansen, M., Galetic, V., Grange, J., Quintana-Amate, S., Nottle, A., Jones, D.M., Morgan, P.L.: Towards an integrated evaluation framework for xai: An experimental study. Procedia Computer Science **207**, 3884–3893 (2022), knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 26th International Conference KES2022

[140] Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q., Artzi, Y.: BERTScore: Evaluating text generation with BERT. In: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net (2020), https://openreview.net/forum?id=SkeHuCVFDr

[141] Zhang, X., Marwah, M., Lee, I.t., Arlitt, M., Goldwasser, D.: Ace – an anomaly contribution explainer for cyber-security applications. In: 2019 IEEE International Conference on Big Data (Big Data). pp. 1991–2000 (2019)

[142] Zhao, Y., Singh, P., Bhathena, H., Ramos, B., Joshi, A., Gadiyaram, S., Sharma, S.: Optimizing LLM based retrieval augmented generation pipelines in the financial domain. In: Yang, Y., Davani, A., Sil, A., Kumar, A. (eds.) Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track). pp. 279–294. Association for Computational Linguistics, Mexico City, Mexico (Jun 2024). https://doi.org/10.18653/v1/2024.naacl-industry.23, https://aclanthology.org/2024.naacl-industry.23/

[143] Zhou, J., Gandomi, A.H., Chen, F., Holzinger, A.: Evaluating the quality of machine learning explanations: A survey on methods and metrics. Electronics **10**(5), 593 (2021)

[144] Zhu, F., Lei, W., Huang, Y., Wang, C., Zhang, S., Lv, J., Feng, F., Chua, T.S.: TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance. In: Zong, C., Xia, F., Li, W., Navigli, R. (eds.) Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). pp. 3277–3287. Association for Computational Linguistics, Online (Aug 2021). https://doi.org/10.18653/v1/2021.acl-long.254, https://aclanthology.org/2021.acl-long.254/