Large Language Models in FP&A: An Architecture for Grounded Variance Commentary and Driver-Based Narrative Generation

RAHUL RAO JUVVADI

Abstract: *FP&A organisations spend a disproportionate share of each close cycle producing variance commentary, flux narrative, and board-pack discussion that explain how reported results moved against budget, forecast, and prior period. This paper describes an enterprise architecture that pairs a deterministic numerical engine with retrieval-augmented generation, an explicit driver-relationship graph, and a human-in-the-loop review step to produce this narrative without offloading arithmetic or attribution to a general-purpose large language model. In a three-business-unit pilot, commentary cycle time fell from roughly six hours to under one hour per unit per cycle; numerical accuracy ranged from 97.9% to 98.7% against source-of-truth ledger figures; and analyst reviewer ratings ranged from 4.5 to 4.7 on a five-point scale. The architecture is designed for SOX-controlled environments and runs against ERP and EPM systems already in place at most enterprises.*

Keywords: Large Language Models; Variance Commentary; Financial Planning And Analysis; Retrieval-Augmented Generation; Explainable Ai.

Rahul Rao Juvvadi (rahulraoj99@gmail.com) CPA, CGMA, Principal Software Engineer, Microsoft,ORCID: https://orcid.org/0009-0005-0850-8364

Introduction

FP&A sits between the close, the forecast, and the management narrative that explains both. Each cycle produces a familiar set of artefacts: variance bridges that walk last period's results to this period's, flux commentary against budget and prior year at the account and cost-center level, MD&A inputs that feed external reporting, opex flash narratives for the operating review, and board-pack discussion that compresses everything into a small number of paragraphs. Most of this is written by hand. Analysts assemble figures from the ERP and the planning system, identify which variances clear materiality, and draft prose that explains the movement. The work is repetitive, structurally similar from quarter to quarter, and consumes time that would be better spent on forecasting and decision support.

Large language models are an obvious tool for this work. They are also, deployed naively, a poor one. A general-purpose model pointed at financial data will produce confident commentary anchored to invented numbers, misattribute movement to drivers the business does not have, and silently drift away from the organisation's disclosure language between paragraphs. None of this is acceptable in a SOX-controlled reporting environment.

This paper describes an architecture for variance commentary that keeps the arithmetic deterministic, retrieves grounding from approved historical commentary, attributes movement to operational drivers via an explicit graph, and routes every draft through human review before publication. We report on a pilot deployment across three business units and discuss the governance posture the architecture supports.

Related Work

A. Large Language Models for Financial Reporting

The literature on AI in accounting and finance [1]–[3] frames the opportunity: NLP and machine learning are useful for reporting efficiency, financial analysis, and anomaly detection. The other closely related example to the FP&A use case is automated market commentary, which Zhu et al. [4] show can be achieved through an LLM-based pipeline that summarizes thousands of financial time series at scale while satisfying compliance requirements. That result is directly relevant to automating variance commentary, because the inputs have a similar structure and the outputs are subject to similar regulatory constraints.

Domain adaptation is also crucial, as Huang et al.'s FinBERT [5] outperforms general-purpose transformers and dictionary-based approaches on financial sentiment and language tasks because financial terms like "exposure," "leverage," "absorption," and "mix" have meanings in finance that general models commonly misinterpret. Follow-on work using GPT-4 and ChatGPT for sentiment analysis, ESG interpretation, and financial statement review [6], [7] consistently highlights three main limitations: they sometimes hallucinate numerical claims, their reasoning steps are not transparent, and they raise governance challenges in regulated settings. Moreover, Li et al. [8] achieve more than 96% accuracy in extracting structured financial fields from unstructured PDF disclosures using an LLM, which is a strong precedent

for what a grounded LLM can reliably read from documents, and a weaker precedent for what an ungrounded model should be allowed to generate on its own.

B. Retrieval-Augmented Generation and Grounded AI

The main problem we must solve is hallucination, and retrieval is the main lever we pull to mitigate it. Gao et al. [9] survey three large classes of RAG systems — naive, advanced, and modular — and conclude that retrieval, augmentation, and evaluation are the three components that need to be engineered carefully, because our system belongs to the modular class. We explicitly instantiate the retrieval layer, we construct prompts with a templated, version-controlled assembly process, and we run an evaluation harness against held-out, pre-approved commentary from previous quarters.

We are also influenced by research on other narrative-generation systems, such as FABULA [10], which structures intelligence reports around an event–plot graph in order to enhance semantic relevance and reduce redundant evidence; moreover, the closest FP&A analogue is the driver graph, which structures variance attribution around the operational events that produce financial outcomes. Furthermore, GROVE [11] shows that complex narratives are much improved when they are grounded in human-written exemplars, which is directly analogous to using approved historical commentary as our retrieval corpus. READSUM [12] demonstrates that fusing retrieved summaries with transformer generation improves keyword coverage and contextual quality in code summarisation, a structurally similar task. CoordNet [13] is cited here as evidence that unified neural architectures can carry across heterogeneous analytical and reporting workloads.

C. Explainability, Causality, and Trust

XAI is no longer optional in regulated finance [14]–[16]. The reviews show a steady shift in practice toward interpretable models or post-hoc explanation tooling, driven as much by audit, regulator, and risk-management requirements as by user preference. Valensky and Mohaghegh [17] confirm the user-side observation: interpretable outputs raise trust, which matters when the consumer of the output is a CFO or audit committee.

Causal reasoning is the next layer. Zhang et al. [18] demonstrate LLM-assisted causal model auditing — relevant here because variance commentary is, at its core, a causal claim: this movement happened because of these operational events. García-Méndez et al. [19] address temporal relevance in financial news using topic modelling, which connects to the FP&A requirement that commentary tie movements to the right reporting period rather than smuggling in cross-period context. Tantithamthavorn et al. [20] argue more generally that explainable AI supports productivity in repetitive analytical work — a claim we test directly in our pilot.

Grounded Llm Architecture For Fp&A

A. FP&A Narrative Challenges

The artefacts an FP&A team produces in a single close cycle are well-defined: a variance bridge that decomposes the period-on-period and budget-to-actual walk into price, volume, and mix; flux commentary against budget and prior year at the account and cost-center level; MD&A

inputs that feed external reporting; opex flash narratives for the operating review; and board-pack discussion that compresses everything into a small number of paragraphs. Each requires the same underlying analytical work — pull actuals, tie to source, identify what cleared materiality, assemble the explanation — and each is written largely by hand.

The fundamental operational problem is data fragmentation, because actuals are in the ERP, plans and forecasts are in the EPM tool, and the operational drivers are in the transactional systems that finance do not control. Before anyone can even start on the narrative, the analysts have to bring the sources together and reconcile them, and then every cycle, the commentary itself is written from scratch, so that two business units can describe the same type of movement in entirely different vocabularies.

Public LLMs do not fix this, because if you ask a general purpose model to write variance commentary, it will blithely generate percentages, attribute movements to drivers that do not exist, and drift away from the organisation's disclosure style - sometimes within a paragraph. In a SOX environment, the risk is clear, therefore, if the published narrative does not tie back to the underlying deterministic numbers, you have a control failure. The architecture that we describe below is designed to avoid that by cleanly separating the arithmetic from the language. Variance percentages and similar figures are computed in a deterministic numerical layer:

$$Variance\ \% = (Actual - Budget) / Budget \times 100 \quad (1)$$

and never produced by the language model.

B. Domain-Adapted Financial Language Models

Domain-adapted financial language models — FinBERT [5] being the most cited example — extend the value of LLMs in this setting. Pretraining or fine-tuning on financial corpora gives the model the right reading of terms like margin compression, operating leverage, cost absorption, and revenue-mix impact, all of which a general model will either mishandle or read as common-English idioms. In our pipeline FinBERT and similar models are not the generation layer — that role is filled by a general-purpose LLM — but they are useful upstream: classifying retrieved commentary by topic, tagging exemplars for retrieval filtering, and flagging anomalous language in drafts. The downstream LLM benefits from working over inputs that have already been categorised correctly.

Retrieval grounds the generation step. The LLM does not produce attribution or pull out partial sentences from its training, and it produces text from a structured input: the deterministic variance pack, retrieved approved comments, and part of the driver graph, all within disclosure rules. This means that every sentence has a source that an auditor can trace, because every draft comes with an audit trail: which numbers it used, which examples it found, which template and version created it, and which reviewer approved it. Explainability and control are built-in, not bolted-on, therefore every draft is thoroughly reviewed, and the reviewer approval is the final check, which is mandatory for every narrative leaving the system..

C. Proposed Grounded Architecture

The architecture (Fig. 1) has five components: structured financial data sources, a deterministic numerical engine, a driver-relationship graph, a vector store of approved historical commentary, and an LLM generation layer wrapped in human review.

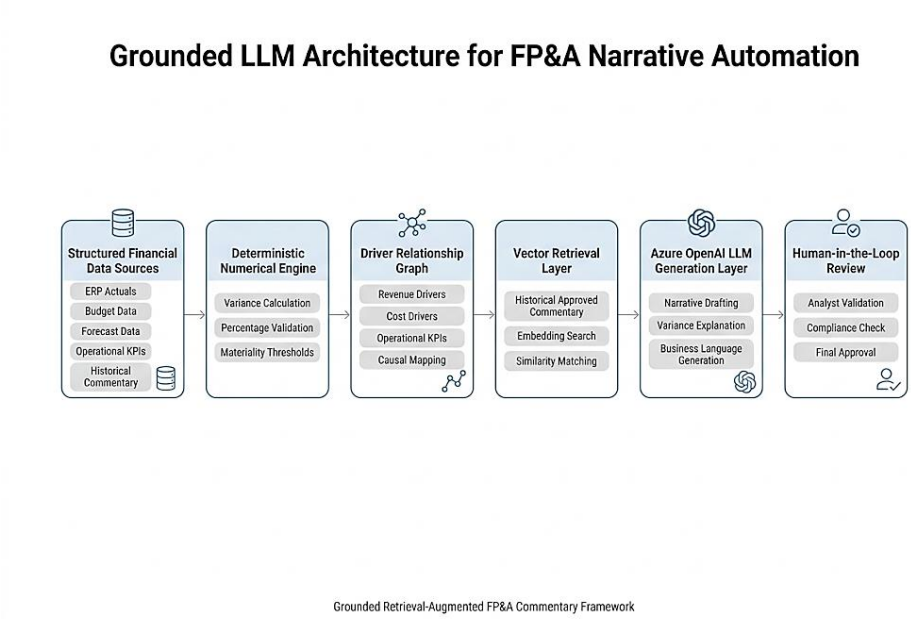

Fig. 1. Grounded LLM architecture for FP&A narrative automation.

Data extraction is easy, and actuals are grabbed from the general ledger (e.g. SAP BSEG and ACDOCA tables in S/4HANA or ECC, GL_BALANCES and related objects in Oracle Fusion or EBS, or the equivalent data structures in NetSuite) and staged into a central warehouse (e.g. Snowflake, BigQuery, Databricks) where the joins can be done cleanly. Plan and forecast come from the EPM platform that the organisation is already using (Workday Adaptive Planning, Anaplan, Oracle Hyperion / EPM, OneStream, etc.), while operational drivers are pulled from the transactional systems that own them (headcount from HRIS - Workday HCM, SAP SuccessFactors, units from MES or order management, ticket or call volumes from service platform). A semantic layer (entity, cost center, account, sub-account, period) is fixed at load time in the warehouse, so every downstream component can assume a consistent, reconciled view.

On top of this, a deterministic numerical engine calculates variances and percentages, ranks them by absolute and relative size, and applies the materiality thresholds that controllership has approved for the period, consequently, the result is the variance pack - a structured payload that becomes the spine of the generated narrative. The LLM never recomputes these figures.

The driver graph encodes the operational-to-financial relationships the business actually has: headcount and overtime drive wage cost; production volume and unit cost drive COGS; shipment volume and fuel cost drive logistics; and so on up to the EBITDA roll. The graph is

maintained by FP&A — not by the LLM and not by the data-science team — because it encodes business judgment about which operational signals attribute to which financial outcomes.

Approved historical commentary is embedded and indexed for retrieval (Section III.D). The retrieval result and the relevant driver subgraph are passed to the generation layer alongside the deterministic variance pack. The output goes to one analyst for review and to a second analyst for approval; nothing publishes without a sign-off.

D. Structured Data Sources and Retrieval

Table I summarises the major structured inputs. The retrieval layer in our reference deployment is built on Azure AI Search (formerly Cognitive Search) using text-embedding-ada-002 to embed approved historical commentary. Open-source equivalents — BGE-large or e5-large with FAISS, Weaviate, or Pinecone as the vector store — substitute one-for-one in deployments outside Azure.

Chunking is paragraph-level with a small overlap, because variance commentary is naturally paragraph-structured: one paragraph per material driver, one for outlook. Each chunk carries metadata for entity, cost center, account hierarchy, reporting period, and disclosure topic. At retrieval time the query is the embedded variance summary, and metadata filters constrain the result to the same entity (or a comparable one), an analogous period — same quarter prior year, prior quarter — and the same account class. Retrieval is hybrid: BM25 over the metadata-tagged text plus dense retrieval over the embeddings, fused at the score level. The top eight chunks pass forward; the rest are dropped without reranking unless the cohort is unusually heterogeneous, in which case a cross-encoder reranker is applied.

TABLE I:MAJOR STRUCTURED INPUTS

Data Source	Description	Granularity	Purpose
ERP Actuals	Posted ledger values from BSEG / ACDOCA, GL_BALANCES, etc.	Account level	Variance calculation
Plan / Forecast	Plan and rolling forecast cubes from Adaptive, Anaplan, or Hyperion	Cost-center level	Benchmark for variance
Driver Metrics	Operational KPIs sourced from HRIS, MES, order management	Business-unit level	Causal attribution
Historical Commentary	Approved prior-period variance narrative	Paragraph level	Retrieval grounding

Narrative Generation Pipeline

A. Four-Stage Pipeline

The pipeline has four stages (Fig. 3): variance decomposition, driver attribution, retrieval-augmented drafting, and human review. The first two are deterministic — the same inputs produce the same outputs, every time. The third is LLM-mediated. The fourth is human. The boundary matters: nothing inside the deterministic stages can be silently rewritten by the language model downstream.

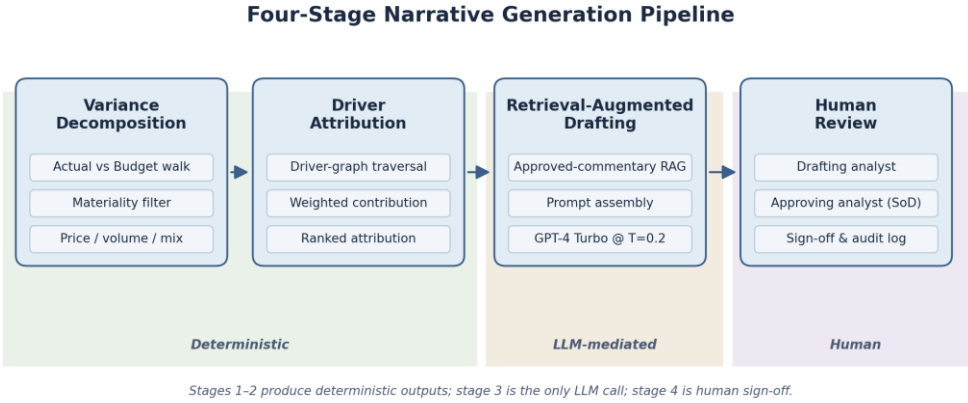

Fig. 3. Four-stage narrative generation pipeline. Stages 1–2 are deterministic; stage 3 is the only LLM call; stage 4 is human sign-off.

Variance decomposition runs on the staged warehouse, and the engine walks actuals against budget, prior year, and forecast using the materiality threshold controllership has set for the period, usually a 5% relative threshold, plus an absolute-dollar floor that scales with entity size. The output is the variance pack, which includes every material movement, broken out into price, volume, and mix wherever the underlying data supports that breakdown, with any unexplained remainder explicitly residual rather than quietly folded into another component. Driver attribution takes each material variance and finds the operational signals that the driver graph links to it, and in practice, several drivers usually contribute to a single movement. The graph carries a weight for each driver-account pair, which is used in the next step, and the output is a ranked attribution, which shows the top operational drivers and their estimated contribution for a given dollar movement in a given account. Retrieval-augmented drafting is where the prompt is assembled and passed to the LLM.

We are on GPT-4 Turbo on Azure OpenAI Service at a temperature of 0.2, which is low enough that wording does not vary between runs, but high enough that the text doesn't sound like a rigid template. The prompt template has five parts: a system message that sets the role, tone, and disclosure rules; the structured variance pack as a JSON-shaped payload; the top-k retrieved historical commentaries with period tags; the relevant slice of the driver graph with weights; and a governance block that states the materiality threshold that was actually used, and a list of topics that must not appear in this draft.

Human review is the last step, and it is non-negotiable, because the drafting analyst verifies every number against the deterministic pack, every attribution against the driver graph, and every paragraph for tone and compliance with disclosure rules. Then, an approving analyst, who is not the drafter, signs off before anything is published, and in a SOX environment, that segregation of duties is mandatory, not optional.

B. Variance Decomposition and Driver Attribution

The engine searches for movements that exceed the negotiated materiality threshold in revenue, expense, margin, and key operational lines, and calculates both absolute and percentage variance against every comparison base for each. The engine in effect exchanges a little completeness for a lot of noise reduction, which is exactly what reviewers and audit committees want to see in the commentary, because this approach helps to focus on the most significant changes.

Driver attribution then weights each contributing driver by its operational importance to the impacted account, not by its raw dollar impact, and this distinction is deliberate: a $2M wage-cost swing due to headcount is more decision-relevant than a $2M FX revaluation on the same line, because management can act on the former but not the latter. The attribution formula makes this explicit:

$$DriverContribution_i = (w_i \cdot Impact_i) / \sum_{j=1..n} (w_j \cdot Impact_j) \quad (2)$$

where w_i is the operational-importance weight the driver graph assigns to driver i for the affected account, and Impact_i is its dollar contribution. The weights are set by FP&A judgment and tuned against the historical commentary that reviewers have approved; they are not learned from the LLM. Attribution traverses the driver graph (Fig. 4) rather than guessing across all available signals.

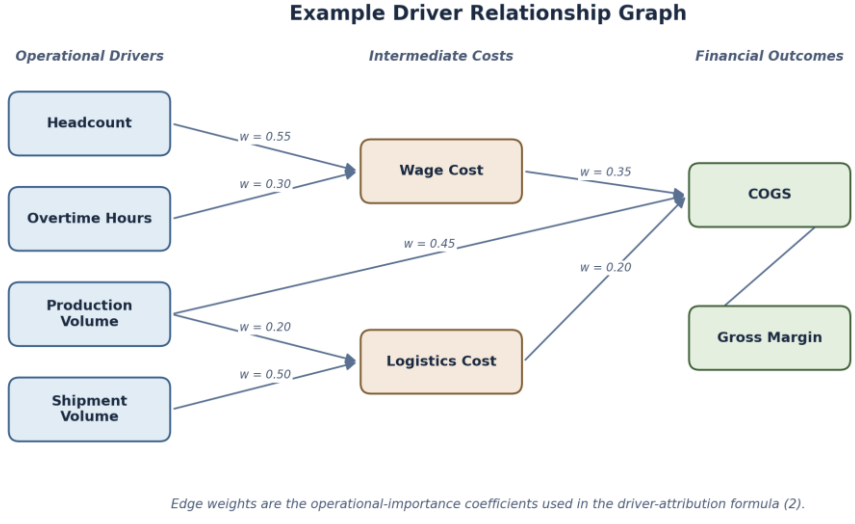

Fig. 4. Example driver-relationship graph. Nodes are operational and financial measures; edges encode the attribution mappings used in stage 2 of the pipeline.

C. Retrieval-Augmented Drafting

The drafting step retrieves before it generates. Approved historical commentary — eight quarters' worth, filtered to comparable entity and period — is embedded with text-embedding-ada-002 and indexed in Azure AI Search. At retrieval time the variance-pack summary is embedded with the same model, hybrid retrieval is run (BM25 + dense, fused at the score level), and the top eight chunks are returned with their period tags intact. Similarity is computed with the standard cosine measure:

$$Similarity(A, B) = (A \cdot B) / (||A|| \cdot ||B||) \quad (3)$$

The fetched chunks bear most of the burden of anchoring the language because the drafts are built around prior, approved commentary, they will inherently inherit the organisation's own disclosure vocabulary (seasonality, pricing actions, restructuring, FX, etc), without having to explicitly code those into the prompt. The model's job is then simply to drop in the period-specific facts from the variance pack and the relevant segment of the driver graph.

The prompt template as described in IV.A is version controlled, and each draft stores the version of the prompt template it was generated with, the retrieval cohort it drew from, and the model and temperature settings, which allows any reviewers or auditors to reproduce any output that is later disputed, with the exact same inputs and settings.

D. Evaluation Metrics

There are two sections of the test, and one section is automated. The other section is manual, and no section of the test is weighted more heavily than another. The automated section verifies the values, and it examines each number within the passage. It compares them against the known change list, and if any discrepancy exists, then the result is a failure.

The human then judges whether the text selected appropriate reasons, and they assign a score between 1 and 4. The human also assesses whether the style is appropriate, because this involves comparing the text to permitted texts. The final measure is the number of texts accepted without modification, thus providing a clear indication of the test's outcome. The full evaluation framework is summarised in Table II.

TABLE II: EVALUATION FRAMEWORK

Metric	Definition	Measurement Method	Target Threshold
Factual Accuracy	Numerical correctness against variance pack	Automated parse-and-match	> 98%
Attribution Validity	Correct driver identified for each material variance	Reviewer rubric (4-point)	> 90%
Stylistic Fit	Alignment with approved prior commentary	Embedding cosine similarity	> 85%

Review Acceptance	Drafts approved without substantive edit	Workflow log	> 90%

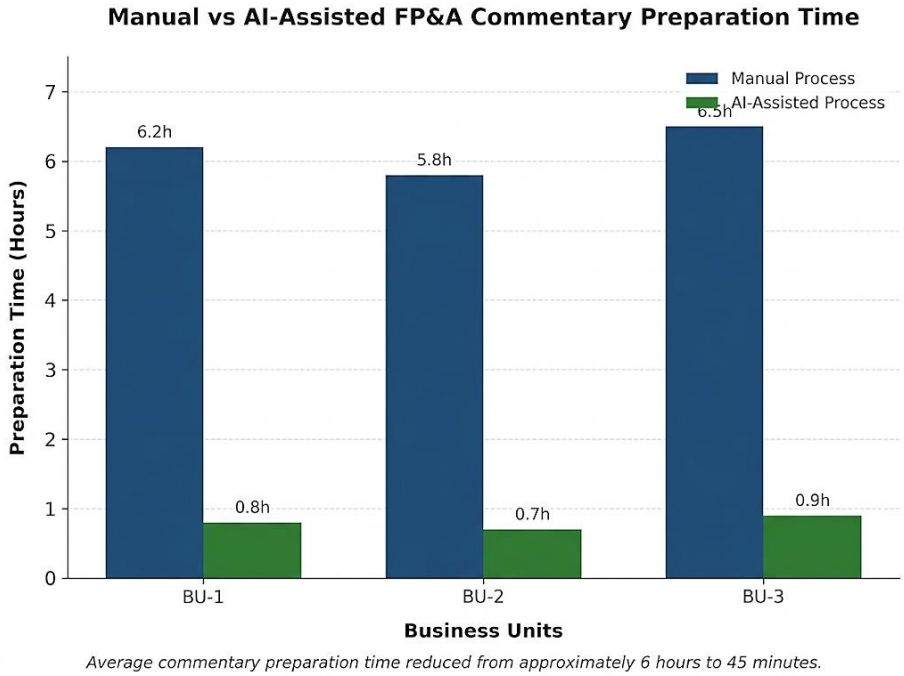

Fig. 2. Manual vs AI-assisted commentary preparation time across the three pilot business units.

FP&A Operating Model Transformation

A. Pilot Implementation

We prototyped the architecture in 3 business units selected to span a broad cross-section of operations (a manufacturing unit on SAP S/4HANA with Adaptive Planning, a services unit on Oracle Fusion with Hyperion, and a regional sales unit on NetSuite with Anaplan), and the prototype ran for two full quarterly close cycles plus a mid-quarter flash review.

We rolled out in three phases: the first built the historical commentary corpus (8 quarters of approved variance narrative across the 3 business units, divided into ~1,200 paragraph-level chunks, cleaned and indexed in Azure AI Search), the second wired up the variance-pack pipeline (nightly extract from the relevant GL into Snowflake, joined to plan and forecast cubes pulled from Adaptive, Hyperion and Anaplan, and operational drivers pulled from the HRIS, MES and order-management system), and the third implemented the human-review workflow in the existing close tooling (so generated drafts and approvals lived alongside the rest of the reporting pack, rather than in a separate system).

Once the integrations stabilised, the cycle ran end-to-end without analyst intervention up to the review step. The deterministic engine created the variance pack within minutes of the period-

end accruals being posted, and the LLM then created a draft for each business unit in less than a minute, after which reviewers could pick up and start editing those drafts later that same day.

B. Quantitative Results

The headline result was cycle time, and the same analysts, who had spent an average of 6.2 hours per business unit per cycle writing commentary manually before this architecture, were now producing reviewed and approved commentary in under an hour. We measure efficiency as:

Efficiency Gain = (Manual Time − Automated Time) / Manual Time × 100

And this efficiency gain was over 85% across the three pilot units. The biggest gains were in the most repetitive parts of the pack — opex flash and recurring, driver-led explanations — where retrieval brought up near-identical prior-period examples that the analyst could check and adapt in minutes, therefore quality did not suffer. Reviewers evaluated the AI-assisted drafts against four dimensions — factual accuracy, attribution validity, clarity, and adherence to disclosure style — and found the AI-assisted commentary to be as good as or better than the fully manual drafts on every measure, because they were able to assess numerical errors, which were effectively zero, which is exactly what the deterministic numerical layer is meant to guarantee: the LLM never sees raw figures it might miscopy, only the structured variance pack. The pilot also revealed a second-order benefit we had not intended to capture, however, cross-business unit consistency improved dramatically, because three business units that had hitherto described seasonality, inflation and operational delays in three different ways began to produce commentary that looked as if it came from the same firm, since they were all drawing from the same retrieval corpus.Table III summarises the pilot results.

TABLE III: PILOT RESULTS

Business Unit	Manual Preparation Time	AI-Assisted Time	Accuracy Score	Reviewer Rating
BU-1	6.2 hrs	0.8 hrs	98.4%	4.6 / 5
BU-2	5.8 hrs	0.7 hrs	97.9%	4.5 / 5
BU-3	6.5 hrs	0.9 hrs	98.7%	4.7 / 5

C. Governance, Compliance, and Risk Management

The architecture is designed to be deployable in a SOX-controlled environment, so governance was a first-class design constraint rather than a post-hoc consideration. Three risks dominated the threat model.

Prompt injection is the first. Approved historical commentary is trusted input; everything else is not. The retrieval layer is restricted to the approved corpus, with read-only access enforced at the storage tier. The prompt template is versioned and code-reviewed. The LLM has no tool calls and no network access beyond its grounding payload, so an injected instruction has nowhere to exfiltrate to and nothing to trigger.

MNPI handling is the second. FP&A operates on material non-public information by definition. The deployment runs on Azure OpenAI Service with the no-training and no-retention contractual posture in place, in a tenancy that inherits the same access controls as the underlying ERP and EPM systems. Items flagged by counsel — litigation, in-process M&A, pre-announcement results — are excluded from the retrieval corpus and listed in the governance-constraints block of the prompt, so the model has neither the grounding nor the instruction to discuss them.

The audit-trail requirement is the third. Every published narrative is traceable to four artefacts: (i) the deterministic variance pack it was generated from, with row-level pointers back to the GL extract; (ii) the retrieved exemplars and their period tags; (iii) the exact prompt version, including temperature setting and model identifier; and (iv) the analyst who reviewed and the analyst who approved, with timestamps. Segregation of duties is enforced — drafting reviewer and approving reviewer are different individuals — and the audit trail is exported into the close-process evidence repository alongside the other SOX artefacts.

Explainability follows from the architecture rather than being added on the side. Every numerical claim points to its row in the variance pack; every driver attribution points to the edge in the driver graph that produced it; every stylistic choice points to the retrieved exemplars that grounded it. The chain from output to evidence is short and inspectable.

D. Near-Term Roadmap

There are three follow-on workstreams in flight today.

First, integration with EPM forecasting closes the loop between explanation and projection, and the same driver graph that explains this quarter's variance can also carry run-rate adjustments into the rolling forecast, with the LLM drafting forecast commentary against the updated cube. The technical change is small; however, the bigger shift is operating-model: FP&A's narrative and forecasting work moves onto a shared substrate.

Second, we are applying the approach to multilingual disclosure for multi-jurisdictional reporting, because all approved commentary in the pilot corpus is in English, but subsidiaries that need to report locally in French, German, Spanish, or Japanese need the same grounding. The retrieval-and-draft pattern generalises naturally by giving each entity its own language-specific corpus, although simple translation of English commentary is not sufficient, because local disclosure conventions and statutory phrasing differ enough that an audit-grade narrative cannot be produced by a translation pass over an English original.

Third, we are beefing up the driver graphs for the opex categories, and the pilot graph is richest around revenue, COGS, and the operating-leverage roll into EBITDA, while SG&A, R&D capitalization, and indirect-cost allocation are the weakest categories, where attribution is limited by less explicit driver mappings. Building these mappings is a manual exercise, but it is the work that matters most for the next phase of the pilot, consequently, FP&A is working with cost-center owners to walk through the operational signals that actually move each account.

Conclusion

Pilot shows you can have reliable variance commentary in a SOX-controlled environment, provided you keep the arithmetic deterministic, anchor the language to approved prior commentary, pipe attribution through an explicit driver graph and require human sign-off before publication. Cycle time goes down by about an order of magnitude, numerical accuracy is virtually 100% (the LLM never generates raw numbers) and reviewer ratings remain high (drafts sound like the organization's own voice) because all of this can be built on technology stacks that most enterprises already own.

What remains is three clear lines of work: tighter integration with EPM forecasting, multilingual support for multi-jurisdictional reporting and richer driver graphs for opex and indirects, and none of this requires new model capabilities, but rather the same careful construction of operational mappings that FP&A has always had to do, therefore the model really is the easy part.

References

. Yi, X. Cao, Z. Chen, and S. Li, *"Artificial intelligence in Accounting and Finance: Challenges and opportunities," IEEE Access*, vol. 11, pp. 129100–129123, Jan. 2023, doi: 10.1109/access.2023.3333389.

N. K. Verma, *"Accounting Analytics in the Era of Open AI Transforming Financial Analysis through Machine Learning Models,"* Aug. 16, 2023. [Online]. Available: https://www.ijisae.org/index.php/IJISAE/article/view/7237

M. Kureljusic and E. Karger, *"Forecasting in financial accounting with artificial intelligence – A systematic literature review and future research agenda," Journal of Applied Accounting Research*, vol. 25, no. 1, pp. 81–104, May 2023, doi: 10.1108/jaar-06-2022-0146.

D. Zhu, T. Lappas, and T. Rachidi, *"Commentary generation for financial markets," Expert Systems with Applications*, vol. 211, p. 118364, Aug. 2022, doi: 10.1016/j.eswa.2022.118364.

A. H. Huang, H. Wang, and Y. Yang, *"FinBERT: A Large Language Model for Extracting Information from Financial Text," Contemporary Accounting Research*, vol. 40, no. 2, pp. 806–841, Sep. 2022, doi: 10.1111/1911-3846.12832.

B. Chen, Z. Wu, and R. Zhao, *"From fiction to fact: the growing role of generative AI in business and finance," Journal of Chinese Economic and Business Studies*, vol. 21, no. 4, pp. 471–496, Aug. 2023, doi: 10.1080/14765284.2023.2245279.

Y. Cao and J. Zhai, *"Bridging the gap – the impact of ChatGPT on financial research," Journal of Chinese Economic and Business Studies*, vol. 21, no. 2, pp. 177–191, Apr. 2023, doi: 10.1080/14765284.2023.2212434.

H. Li, H. Gao, C. Wu, and M. A. Vasarhelyi, *"Extracting financial data from unstructured sources: leveraging large language models," SSRN Electronic Journal*, Jan. 2023, doi: 10.2139/ssrn.4567607.

Y. Gao et al., *"Retrieval-Augmented Generation for Large Language Models: A survey," arXiv preprint arXiv:2312.10997*, Dec. 2023, doi: 10.48550/arxiv.2312.10997.

P. Ranade and A. Joshi, *"FABULA: Intelligence Report Generation using Retrieval-Augmented Narrative Construction,"* UMBC Ebiquity Research Lab, Nov. 06, 2023. [Online]. Available: https://ebiquity.umbc.edu/paper/html/id/1177/

Z. Wen et al., *"GROVE: A Retrieval-augmented Complex Story Generation Framework with A Forest of Evidence," in Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 3980–3998, 2023, doi: 10.18653/v1/2023.findings-emnlp.262.

Y. Choi, C. Na, H. Kim, and J.-H. Lee, *"READSUM: Retrieval-Augmented Adaptive Transformer for Source Code Summarization," IEEE Access*, vol. 11, pp. 51155–51165, Jan. 2023, doi: 10.1109/access.2023.3271992.

J. Han and C. Wang, *"COORDNET: Data generation and Visualization Generation for Time-Varying Volumes via a Coordinate-Based Neural Network," IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 12, pp. 4951–4963, Aug. 2022, doi: 10.1109/tvcg.2022.3197203.

P. Weber, K. V. Carl, and O. Hinz, *"Applications of Explainable Artificial Intelligence in Finance — a systematic review of Finance, Information Systems, and Computer Science literature," Management Review Quarterly*, vol. 74, no. 2, pp. 867–907, Feb. 2023, doi: 10.1007/s11301-023-00320-0.

X.-Q. Chen, C.-Q. Ma, Y.-S. Ren, Y.-T. Lei, N. Q. A. Huynh, and S. Narayan, *"Explainable artificial intelligence in finance: A bibliometric review," Finance Research Letters*, vol. 56, p. 104145, Jun. 2023, doi: 10.1016/j.frl.2023.104145.

T. Martins, A. M. De Almeida, E. Cardoso, and L. Nunes, *"Explainable Artificial Intelligence (XAI): A systematic literature review on taxonomies and applications in finance," IEEE Access*, vol. 12, pp. 618–629, Dec. 2023, doi: 10.1109/access.2023.3347028.

D. Valensky and M. Mohaghegh, *"Evaluating Transparency: A Cross-Model Exploration of Explainable AI in Financial Forecasting," in Proc. 2023 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)*, pp. 1–6, Dec. 2023, doi: 10.1109/csde59766.2023.10487732.

L. Zhang, A. Fitzgibbon, J. Garofolo, A. Kota, E. Papenhausen, and K. Mueller, *"An Explainable AI Approach to Large Language Model Assisted Causal Model Auditing and Development," in Proc. IEEE VIS 2023*, Oct. 2023. [Online]. Available: https://virtual.ieeevis.org/year/2023/paper_w-nlviz-1018.html

S. García-Méndez, F. De Arriba-Pérez, A. Barros-Vila, F. J. González-Castaño, and E. Costa-Montenegro, *"Automatic detection of relevant information, predictions and forecasts in financial news through topic modelling with Latent Dirichlet Allocation," Applied Intelligence*, vol. 53, no. 16, pp. 19610–19628, Mar. 2023, doi: 10.1007/s10489-023-04452-4.

C. Tantithamthavorn, J. Cito, H. Hemmati, and S. Chandra, *"Explainable AI for SE: challenges and future directions," IEEE Software*, vol. 40, no. 3, pp. 29–33, Apr. 2023, doi: 10.1109/ms.2023.3246686.