intelligent infrastructure and construction

Article

An Agentic AI and LLM-Based Framework for Probabilistic Cost Estimation from Fragmented BIM Data

Liupengfei Wu [1,*], Qian Zhang [2], Ruiying Xu [3], Yiran Zhang [3], Frank Ato Ghansah [4] and Xichen Chen [5]

[1] Division of Industrial Data Science, School of Data Science, Lingnan University, Hong Kong, China
[2] Department of Engineering, Durham University, Durham DH1 3LE, UK; qian.zhang2@durham.ac.uk
[3] Division of Artificial Intelligence, School of Data Science, Lingnan University, Hong Kong, China; ruiyingxu@ln.hk (R.X.); yiranzhang@ln.hk (Y.Z.)
[4] School of Construction, Property and Surveying, London South Bank University, London SE1 0AA, UK; frank.ghansah@lsbu.ac.uk
[5] School of Future Environments, Auckland University of Technology, Auckland 1010, New Zealand; xichen.chen@aut.ac.nz
* Correspondence: mikewu@ln.edu.hk

Abstract

Building Information Modelling (BIM) has digitized construction, yet automated cost estimation still suffers from fragmented data and deterministic forecasts that ignore uncertainty. To address this gap, this study introduces a novel framework integrating agentic artificial intelligence (AI) with large language models (LLMs) to enable probabilistic cost estimation from disparate BIM data. The system employs four specialized collaborative agents operating via a shared memory module centered on an LLM with natural language understanding, code generation, and chain-of-thought reasoning. A prototype using GPT-4 Turbo, AutoGen, and Monte Carlo simulation was tested on three real-world structures. Compared to three baselines, the framework reduced processing time (4.2 vs. 18.5–68.0 min), manual interventions (0.8 vs. 9–14), and improved entity resolution accuracy (86.5% vs. 46–62%) with well-calibrated probabilistic forecasts, achieving 86.0% empirical coverage for nominal 90% prediction intervals (Prediction Interval Coverage Probability [PICP] = 86.0%, Prediction Interval Width [PIW] = 0.28; $p < 0.01$). Qualitative analysis confirmed effective semantic conflict resolution and actionable risk visualization via tornado diagrams. The framework tackles long-standing BIM estimation challenges by delivering probabilistic, transparent outputs. Future work includes digital twin integration, open-source LLM deployment, and during-construction forecasting.

Keywords: agentic AI; large language models; probabilistic cost estimation; building information modelling; uncertainty quantification

Check for updates

Academic Editor: Jun Wang

Received: 14 May 2026
Revised: 21 June 2026
Accepted: 24 June 2026
Published: 28 June 2026

1. Introduction

Building Information Modelling (BIM) enables various inputs to projects, including geometry, materials, and schedule, to be assembled into a single digital model [1]. Though firms have begun implementing "5D BIM" which links together 3D models of buildings to schedule and cost [2], the act of cost estimating necessitates sourcing information from many disparate places, ranging from Industry Foundation Classes (IFC) files to spreadsheets to PDF quotations to handwritten notes to free text on contracts or specification documents [3]. The resulting cost datasets available to cost engineers often suffer from fragmentation, with various components captured at varying resolutions, inconsistent units of measure, and substantial differences in fidelity [4]. Human cost engineers must manually consolidate these disparate inputs into a single integrated estimate. This data fragmentation significantly constrains the probabilistic representation and propagation of uncertainty across the estimating process. Consequently, most BIM-enabled cost estimates remain point estimates that fail to adequately capture uncertainties associated with material prices, labor productivity rates, and design changes [3]. Addressing this tension is therefore essential for enabling reliable probabilistic cost estimation in BIM environments.

Despite the envisioned benefits of 5D BIM, current practice remains constrained by two key limitations: deterministic predictions and manual data integration. First, widely used BIM tools generate deterministic point estimates that imply misleading precision [4]. These estimates assume quantities, unit costs, and productivity rates are known with certainty, disregarding material price fluctuations, workforce variability, design changes, and market conditions [5]. Consequently, point estimates provide limited insight into cost overrun likelihood or contingency requirements [6]. Second, estimators must manually consolidate fragmented sources, reconciling inconsistent measurement systems, interpreting non-standardised naming conventions, and inferring missing information with limited computational support [3,4]. This process is labour-intensive and difficult to replicate. When uncertainty handling relies on estimator judgment, uncertainties are rarely identified or propagated systematically [7]. Probabilistic methods such as Monte Carlo simulation have been applied, but typically on aggregated estimates rather than propagating uncertainty from disaggregated fragments [8]. Current practice thus remains incapable of generating calibrated probabilistic forecasts. To address this gap, this study introduces an automated framework that (a) integrates fragmented heterogeneous data sources; and (b) produces probabilistic cost forecasts with explicitly quantified uncertainty.

Extensive research has been devoted to BIM-based cost estimation and probabilistic methods. However, these approaches have not fully leveraged the potential of combining agentic artificial intelligence (AI) with large language models (LLMs) in an end-to-end framework for data integration and uncertainty quantification. Agentic AI refers to systems that can decompose high-level goals into subtasks, make decisions at each stage, and execute actions with limited human involvement [9]. LLMs are deep learning networks trained on vast text corpora that can interpret natural language, reason about language-based problems, and generate computer code [10]. These capabilities are particularly relevant for construction, where project data is often embedded in unstructured text. Prior work has investigated LLMs for parsing contracts, specifications, and change orders [11] and agentic AI for coordinating construction supply chains [12]. However, these lines of research remain isolated. LLM applications in construction have focused on information extraction and document classification without probabilistic reasoning or multi-agent coordination [13]. Similarly, while early work has shown that agentic AI can enable workflow coordination across multiple parties in construction logistics [12], agentic AI has been applied to workflow coordination in logistics, but not to BIM-based cost estimation. To our knowledge, no existing framework integrates LLMs with agentic AI to automatically ingest fragmented BIM data, recognize uncertainties, and generate probabilistic cost forecasts in a single pipeline. The present research aims to bridge this gap.

Several AI and machine learning approaches have been explored for BIM-based cost estimation. Traditional machine learning methods, including support vector regression, random forests, and artificial neural networks, have been applied to predict construction costs using historical project data [14]. These approaches typically require large, structured datasets with consistent feature representations and are limited to point estimates without explicit uncertainty quantification. Hybrid BIM analytics have also emerged, combining BIM-derived quantities with external databases to improve estimate accuracy [3]. However, these methods still rely on manually reconciled input data and do not address the fundamental challenge of integrating fragmented heterogeneous sources. More recently, LLMs have been investigated for information extraction from construction documents [11,13], yet these applications treat LLMs as isolated assistants without agentic coordination or probabilistic reasoning. The present framework distinguishes itself by integrating agentic AI with LLMs in an end-to-end pipeline that automatically ingests fragmented data, resolves semantic conflicts, and generates probabilistic cost forecasts, thereby addressing gaps not covered by prior AI approaches.

To address the identified research gap, this study aims to propose a novel agentic AI and LLMs-enabled framework for probabilistic cost estimation from fragmented BIM data. The primary objectives are (1) to develop an agentic AI and LLM-based framework comprising specialised agents for data ingestion, integration, alignment, and uncertainty quantification for probabilistic cost estimation in BIM; (2) to substantiate the proposed framework by implementing a prototype system that operationalises the agentic workflow, LLM integration, and probabilistic estimation engine; and (3) to evaluate the framework through controlled experiments comparing its performance against baseline methods, including traditional deterministic BIM estimation, manual probabilistic estimation, and non-agentic LLM-only estimation. The remainder of this paper is organised as follows. Section 2 reviews related literature on BIM cost estimation, probabilistic methods, LLMs, and agentic AI. Section 3 describes the research methodology. Section 4 presents the proposed framework in detail. Section 5 reports the implementation and experimental setup. Section 6 presents the results, followed by a discussion in Section 7. Finally, Section 8 concludes the paper and outlines directions for future work.

2. Background

2.1. Cost Estimation in BIM

Cost estimating is one of the areas where BIM caused revolutionary changes [3]. Previously, cost estimators relied on two-dimensional drawings and manual quantity take-offs, which were laborious and tedious, with high potential for human error and inconsistencies between revisions [1]. With the emergence of 5D BIM, quantities could be automatically extracted from the model, which are synchronized with the schedule (4D) and cost (5D) components [2]. This integration allows cost engineers to visualise cost components spatially, track changes dynamically, and generate updated estimates as the design evolves [5]. Consequently, 5D BIM has been widely adopted as a best practice for cost management in many large-scale construction projects [4].

While offering these benefits, BIM-based cost estimation also encounters several long-standing issues. Cost information is often split across multiple heterogeneous sources and is rarely fully captured within the BIM model [8]. Instead, cost-related data typically reside in a mixture of IFC files, spreadsheets, PDF quotations, handwritten notes, and unstructured text files such as contracts and specifications [3]. These sources often employ varying units of measure, data formats, and levels of granularity. Furthermore, existing IFC schemas represent costs using unstructured natural language, hindering automatic downstream processing [15]. Consequently, cost engineers are compelled to manually extract these inputs, reconcile discrepancies, and integrate the data into a cohesive estimate. This manual process is labour-intensive and error-prone, leaving room for inconsistencies across projects [4]. Critically, no standardised workflow exists to validate the accuracy of cost estimations after integrating inputs from multiple [16], nor is there any automated way to associate cost data with objects in the geometric model [15]. These data fragmentation issues motivate several opportunities for automating data integration.

2.2. Probabilistic Cost Estimation

Probabilistic cost estimating has been introduced to increase rigor over deterministic approaches by explicitly recognising uncertainty in inputs to estimation models [17]. While deterministic approaches provide a single-valued output, probabilistic approaches treat cost as a distribution, enabling informed decision-making based on the likelihood of possible outcomes [14]. The most common probabilistic technique used in construction cost modelling is Monte Carlo simulation, which involves sampling thousands of possible scenarios from input probability distributions [18]. A recent bibliometric analysis identified Monte Carlo simulation as one of the seven most common statistical modelling techniques used for cost estimation in infrastructure projects, alongside regression analysis, artificial neural networks, and fuzzy logic [14]. PERT has also been frequently used for schedule and cost uncertainty analysis; however, there are drawbacks to using PERT, such as bias in estimating activity duration [18].

While these methods offer benefits, traditional probabilistic methods encounter several practical challenges when used in the context of BIM-based cost estimation. Existing approaches, such as PERT and Monte Carlo simulation, often rely on practitioners manually specifying input probability distributions (ranges, shapes, correlation assumptions) for each cost element [19]. This reliance on manual specification presents a significant practical barrier: it is not only labour-intensive but also introduces subjectivity and inconsistency across projects, as different estimators may specify different distributions for identical cost elements. Recent studies have shown that probabilistic estimators apply distributions to aggregated cost quantities at the project level rather than estimating the distributions of component-level fragments, thereby failing to capture the uncertainty present within each underlying data source and potentially masking important sources of risk [20]. Additionally, researchers have identified that stochastic simulation models fail to account for the correlation between cost and schedule performance indices when determining probabilities of project performance outcomes [21]. This is a critical limitation, as cost and schedule are inherently interdependent in construction projects, and neglecting their correlation can lead to significantly biased risk assessments. These limitations motivate the need for more automated approaches to uncertainty identification and propagation in probabilistic cost estimation.

2.3. Large Language Models (LLMs) in Construction

LLMs are a category of deep learning models that learn to represent, generate, and operate on natural language, reason, generate code, and perform few-shot learning when trained on large quantities of diverse text corpora and when not fine-tuned for specific tasks [10]. Multiple aspects of LLMs make them a promising tool to augment construction industry workflows. First, LLMs can process unstructured text to extract meaningful structured information [22]. Most project deliverables for construction projects are textual documents such as contracts, technical specifications, change orders, safety incident reports, and unit cost quotations [11,23]. Second, querying LLMs can be done through prompt-based interaction, enabling practitioners to query and summarise construction project documents without having to retrain cost-specific models for every type of task or document [24]. Third, many state-of-the-art LLMs can produce human-readable rationales behind their generated outputs [25], which could make their reasoning processes available and understandable to cost engineers and project stakeholders. Taken together, these properties allow us to consider ways in which LLMs can be used to automate parts of information processing tasks that are often manual processes in BIM workflows.

https://doi.org/10.3390/iic2030008

Building on these features, recent years have witnessed growing research interest in applying LLMs to address various challenges within the construction industry. In the domain of document management, LLMs can automatically classify construction safety and quality reports [26], extract compliance requirements from building codes, and summarise lengthy contract clauses, significantly reducing manual review time [11,13]. For risk management, LLMs have been employed to identify and categorise project risks from unstructured text sources such as meeting minutes, incident reports, and email correspondence, enabling proactive rather than reactive risk responses [23]. Within the specific context of cost estimation, emerging studies have explored LLM-assisted tasks including matching textual quantity descriptions to BIM elements, extracting unit prices from PDF quotations, and accelerating quantity take-off processes (e.g., [15,24]). However, these applications typically treat LLMs as isolated assistants that produce deterministic outputs without quantifying uncertainty, and they lack task decomposition, cross-agent coordination, or feedback-driven refinement [11]. Consequently, while LLMs have shown promise as information processing tools in construction, their integration with probabilistic cost estimation methods, where uncertainty is explicitly modelled and propagated, remains largely unexplored. This represents a significant gap, as the probabilistic nature of cost estimation requires not just information extraction but also reasoning about uncertainty, a capability that current LLM applications in construction have not yet addressed.

2.4. Agentic AI Basics

Recent years have witnessed growing interest in agentic AI across public, academic, and industry spheres [27]. This marks a paradigm shift from passive, rule-based language models to autonomous, goal-oriented agents capable of independent decision-making [28]. Unlike traditional AI systems that require explicit instructions for each step, agentic AI systems can operate with minimal human oversight. There are five major features of agentic AI have been identified [12]. First, autonomy enables agentic AI to operate independently and progressively complete tasks without frequent human intervention [29,30]. Second, adaptive self-learning allows the system to continuously learn, adjust strategies, and optimise models in response to changing environments [31]. Third, multi-agent collaboration facilitates decentralised cooperation through communication, negotiation, and task allocation among multiple agents to accomplish complex objectives [32]. Fourth, self-decision-making empowers the system to learn from data, make context-aware decisions, and autonomously execute plans [28]. Fifth, cognitive management provides the system with long-term memory and contextual understanding, enabling it to maintain continuity across extended tasks [33]. Together, these features equip agentic AI with powerful capabilities to coordinate complex tasks across diverse domains.

Due to its features, agentic AI has begun to attract early research across various industries, including the construction industry. In the business sector, commercial providers such as Agentforce, Conversica, and Drift have launched agentic AI systems that can independently identify sales leads, respond to inbound communications, and even make calls and handle after-sales customer service [34]. In the healthcare industry, multi-agent framework systems have been initially applied to cancer diagnosis and treatment plan design, achieving performance that matches expert levels for cervical cancer and exceeds them for lung cancer, as demonstrated by systems such as GPT-Plan (2025) [35]. Similarly, in smart agriculture, agentic AI accomplishes tasks such as soil moisture monitoring, pest detection, and irrigation management through multi-agent collaboration [31]. In the construction industry, agentic AI remains a new concept, and its current limitations and future research directions have recently been summarized by researchers [36]. Recently, agentic AI has been integrated with the Internet of Things (IoT) to enable just-in-time

(JIT) logistics and supply chain management in modular construction [12]. Nevertheless, these construction applications remain nascent and largely conceptual. Researchers lack a robust theoretical framework for agentic AI in construction contexts, and practitioners lack a deployable system for probabilistic cost estimation in BIM. The present study addresses this gap by providing both a theoretical framework and a functional prototype. To synthesize the literature reviewed above, Table 1 summarizes the key limitations of existing approaches across the four domains examined and maps them to the specific gaps addressed by the proposed framework.

Table 1. Summary of prior approaches and research gaps addressed by the proposed framework.

Domain	Prior Approaches	Key Limitations/Gaps	Addressed by Proposed Framework
BIM Cost Estimation	5D BIM with automated quantity take-off; manual data entry for unit prices; native Revit (2024) and CostX 6.0 workflows	Fragmented data across heterogeneous sources (IFC, spreadsheets, PDF, text); no standardized validation workflow; manual reconciliation of units and naming conventions	Data Ingestion Agents automatically parse multiple formats; Integration and Alignment Agent resolves semantic conflicts
Probabilistic Methods	Monte Carlo simulation; PERT; @RISK 4.5; manual specification of input distributions	Labour-intensive distribution specification; subjectivity in expert judgment; applied at aggregated project level rather than component level; neglect of cost-schedule correlations	Uncertainty Quantification Agent automatically infers distribution types; propagates uncertainty from fragmented source data
LLMs in Construction	Information extraction from contracts, specifications, change orders; document classification; risk identification	Treated as isolated assistants; deterministic outputs without uncertainty quantification; no task decomposition or multi-agent coordination; no integration with probabilistic methods	LLM reasoning core with chain-of-thought prompting; integrated with agentic workflow; probabilistic estimation engine for uncertainty quantification
Agentic AI	Workflow coordination in logistics; supply chain management; IoT integration	Nascent in construction; lacks theoretical framework for BIM contexts; no deployable system for cost estimation	Four specialized agents with shared memory; end-to-end framework with functional prototype; cross-run learning via feedback loop

3. Research Methodology

3.1. Overview of Research Design

A mixed methods research design was adopted in this study to ensure both technical rigor and practical relevance. The design integrated qualitative methods for problem understanding and requirements elicitation with quantitative methods for system evaluation and performance comparison [37]. This dual approach was necessary because the research problem involved both human centric issues, such as fragmented data handling practices and expert judgment in cost estimation, as well as technical challenges, such as algorithmic accuracy and computational efficiency. The qualitative component included semi-structured interviews with cost engineers and document analysis of real-world BIM project data. The quantitative component consisted of controlled experiments comparing the proposed framework against three baseline methods using statistical metrics. The overall design followed the six phases illustrated in Figure 1: problem identification and requirement analysis, data collection and preprocessing, agentic LLM framework development, prototype implementation, baseline methods comparison, and evaluation metrics and validation.

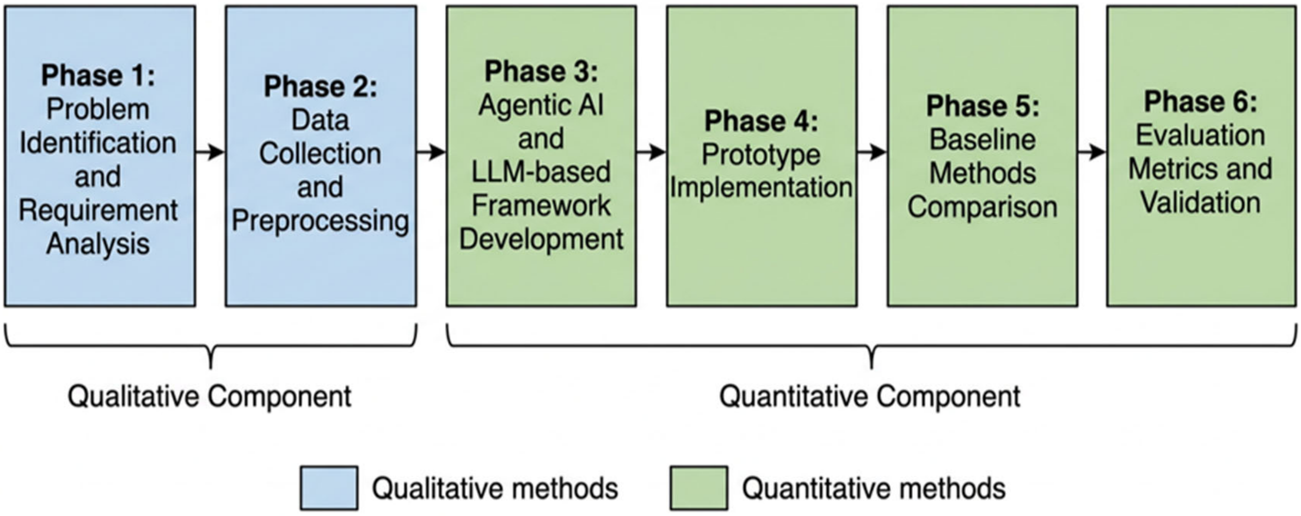


Figure 1. Six phases of the mixed methods research design for agentic AI and LLM-based framework development.

3.2. Problem Identification and Requirement Analysis

This phase established the key challenges in BIM-based cost estimation and derived the functional requirements for the proposed framework. Two methods were employed: a systematic literature review and semi-structured interviews with industry practitioners, complemented by an analysis of real-world project data.

For the literature review component, a systematic search was conducted following established guidelines for literature reviews in construction management research. The following inclusion criteria were applied: (1) peer-reviewed journal articles published between 2019 and 2026; (2) articles written in English; (3) studies addressing BIM cost estimation, probabilistic cost estimation, LLM applications in construction, or agentic AI; and (4) empirical or methodological studies with clear research contributions. The exclusion criteria were (1) conference papers, book chapters, and non-peer-reviewed publications; (2) articles not directly related to construction or infrastructure project cost estimation; (3) studies focusing solely on BIM geometric modeling without cost dimensions; and (4) duplicate publications reporting the same data or findings.

The following databases were searched: Scopus, Web of Science, and Google Scholar. The search strings used were (“BIM” OR “Building Information Modelling”) AND (“cost estimation” OR “cost estimating” OR “quantity take-off”); (“probabilistic” OR “Monte Carlo” OR “uncertainty”) AND (“cost estimation” OR “cost forecasting”) AND (“construction” OR “infrastructure”); (“large language model” OR “LLM” OR “GPT”) AND (“construction” OR “BIM”); (“agentic AI” OR “multi-agent” OR “autonomous agents”) AND (“construction” OR “cost estimation”). The initial search yielded 347 articles. After removing duplicates (n = 89), 258 articles were screened by title and abstract. Of these, 124 articles were excluded as they did not meet the inclusion criteria. The remaining 134 articles underwent full-text review, from which 45 articles were selected for final inclusion based on their relevance to the research objectives and quality of contributions.

For the qualitative component, semi-structured interviews were conducted with 12 cost engineers and project managers from five construction firms operating in Hong Kong, China, and mainland China (see Table 2). Participants were selected using purposive sampling to ensure representation across different project types, firm sizes, and levels of experience. The sample comprised 8 males and 4 females, with professional experience ranging from 5 to 22 years (mean = 11.4 years). All participants had direct involvement in BIM-based cost estimation for at least three completed projects. Interviews were conducted between January and March 2025, lasting 45–75 min each (mean = 58 min) and were conducted either in person or via video conference based on participant preference. The interview protocol covered four thematic areas: (1) current practices and challenges in BIM cost estimation; (2) data fragmentation issues encountered in practice; (3) uncertainty handling approaches; and (4) perceptions of AI and automation potential. All interviews were audio-recorded with participant consent and transcribed verbatim for thematic analysis. Thematic saturation was achieved after the tenth interview, with two additional interviews conducted to confirm consistency.

Table 2. Interview participant demographics.

Participant ID	Role	Firm Type	Experience (Years)	Location	Projects Estimated (BIM)
P1	Cost Engineer	Large contractor	8	Hong Kong	12
P2	Senior QS	Consultancy	16	Mainland China	25+
P3	Project Manager	Developer	12	Hong Kong	18
P4	Cost Engineer	Large contractor	5	Mainland China	7
P5	Senior Cost Manager	Consultancy	22	Hong Kong	30+
P6	BIM Manager	Developer	9	Mainland China	14
P7	Cost Engineer	Medium contractor	7	Hong Kong	9
P8	QS Manager	Consultancy	18	Mainland China	22
P9	Project Cost Lead	Large contractor	14	Hong Kong	20
P10	Cost Engineer	Developer	6	Mainland China	10
P11	Senior Estimator	Consultancy	11	Hong Kong	16
P12	Commercial Manager	Large contractor	10	Mainland China	15

Note: QS = Quantity Surveyor.

The literature review examined 45 peer-reviewed journal articles published from 2019 to 2026 on BIM cost estimation, probabilistic cost estimation, and LLMs in construction (e.g., [11,14]). Two critical knowledge gaps were identified. Firstly, BIM cost data were fragmented across heterogeneous and disconnected data sources, including IFC files, spreadsheets, PDF quotations, and unstructured contract documents. Secondly, existing cost estimating tools predominantly generated deterministic point estimates with limited capability to systematically represent and propagate uncertainty throughout the estimating process.

Real-world project data was analysed from three building structures offered by an industry partner. The three projects were selected based on the following criteria: (1) availability of complete BIM-related documentation including IFC files, cost spreadsheets, and contractual documents; (2) representation of different building types to ensure diversity (educational, cultural, and exhibition facilities); (3) project scale ranging from medium to large (98–386 cost elements); and (4) availability of ground-truth cost data for validation purposes. These selection criteria ensured that the case studies captured the typical data fragmentation challenges encountered in practice while providing sufficient data diversity for framework validation. The projects consisted of a teaching building, exhibition veranda, and concert hall. Average manual reconciliation time was calculated as 14.3 h per project. Project cost overruns from initial deterministic estimates averaged 15.8%, with the top contributing factors being material price fluctuations and design alterations. Based on these findings, five functional requirements were formulated, as shown in Table 3.

Table 3. Functional requirements for the proposed agentic AI and LLMs-based framework.

Functional Requirement ID	Description
R1	Ingest multiple data formats (IFC, PDF, XLSX, and TXT) without manual conversion
R2	Resolve semantic conflicts including unit mismatches and entity naming inconsistencies across sources
R3	Automatically identify uncertainty sources from unstructured text and missing data fields
R4	Generate probabilistic cost distributions instead of point estimates
R5	Provide an end-to-end automated workflow requiring no manual intervention during normal operation

3.3. Data Collection and Preprocessing

The data collection process compiled a comprehensive dataset from three sources. First, an industry partner provided three real-world building structures with complete BIM-related documentation. These structures were chosen to represent a concert hall, an exhibition veranda and a teaching building, as shown in Figure 2. The collected industry dataset comprised 12 IFC files, 23 spreadsheets detailing cost line items, 58 PDF quotation documents, and 18 text-based documents containing miscellaneous clauses from contracts and specification notes. As these textual documents were not stored in a structured format, they were treated as unstructured data sources. Second, supplementary datasets were obtained from reference building models provided by the National Institute of Standards and Technology (NIST). These datasets included existing Revit files containing design information for premise plumbing systems. The NIST data were incorporated to provide additional BIM geometries and plumbing system layouts for framework validation.

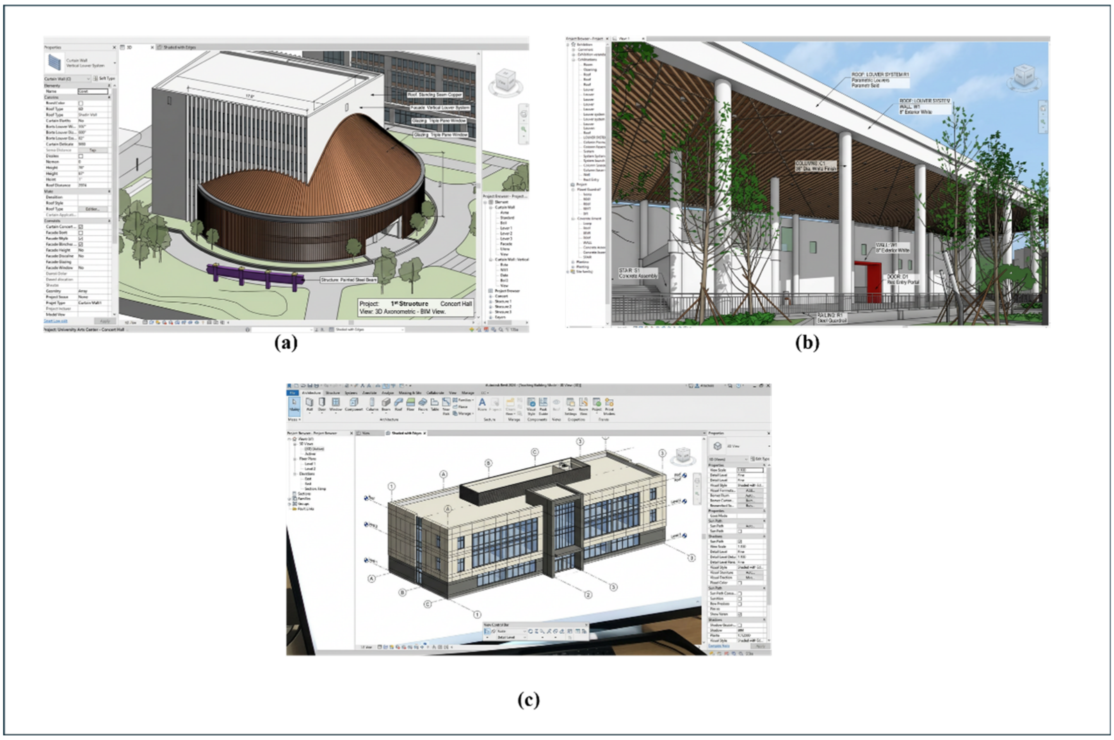


Figure 2. BIM models of the three real-world building structures used in the case study: (**a**) concert hall, (**b**) exhibition veranda, and (**c**) teaching building.

All datasets underwent preprocessing using a standardized data processing pipeline. First, file formats were automatically identified, followed by content extraction. Geometric and semantic entities from IFC and Revit files were parsed using IfcOpenShell (an open-source software library for working with IFC files). From PDF documents, textual content was extracted using optical character recognition (OCR) techniques where scanned documents were detected. Spreadsheet datasets were processed to extract numerical values and metadata, with column types inferred automatically during preprocessing. For unstructured text documents, raw textual content was extracted and subjected to basic cleaning and tokenization procedures. Subsequently, extracted data fields were semantically annotated according to predefined entity categories (e.g., material name, unit price, quantity, and labour hours) using a rule-based classification approach incorporating keyword matching techniques. Data fields that could not be confidently annotated or exhibited semantic ambiguity were automatically flagged for further review by the Integration and Alignment Agent during runtime.

3.4. Agentic AI and LLM-Based Framework Development

The framework was developed, following an iterative design process informed by the functional requirements (R1 to R5) identified in Section 3.2. The proposed agentic AI architecture comprised four specialized agent types operating collaboratively within a coordinated workflow. The Orchestrator Agent was responsible for task decomposition and workflow coordination, partitioning the overall cost estimation process into executable subtasks and dynamically allocating them to the appropriate agents. The Data Ingestion Agents included three specialized subtypes for text, tabular, and geometric data formats, each designed to parse heterogeneous sources and extract structured information. The Integration and Alignment Agent performed entity resolution across different sources, detected semantic conflicts such as unit mismatches and naming inconsistencies, and produced a unified data model. Finally, the Uncertainty Quantification Agent identified missing or incomplete data fields, inferred suitable probability distributions based on available data characteristics (e.g., triangular distributions for three-point estimates and Beta-PERT distributions for expert-elicited ranges), and flagged assumptions requiring further validation.

The LLM reasoning core was configured using OpenAI's GPT-4 as the base model. Two temperature settings were applied based on task requirements: a low temperature setting (0.2) for deterministic parsing tasks requiring stable and reproducible outputs, and a higher temperature setting (0.7) for uncertainty reasoning tasks requiring broader hypothesis generation and exploratory inference. A chain-of-thought prompting strategy was employed for cost driver identification, supported by few-shot exemplars embedded within the system prompts. Inter-agent communication was implemented using a publish–subscribe architecture coupled with a shared memory module that stored intermediate outputs, execution logs, semantic conflict records, and identified uncertainty patterns. The shared memory mechanism enabled the reuse of contextual information across estimation workflows, thereby improving coordination consistency and reducing repeated processing errors during iterative estimation runs.

3.5. Prototype Implementation

The proposed framework was implemented as a functional prototype to enable empirical evaluation against baseline methods. The prototype was coded in Python 3.11 with four main packages used. AutoGen (version 0.2.0) was used to specify agent roles and communication between agents, which allowed for coding of the orchestrator and four agents outlined in Section 4. LangChain (version 0.1.0) was used to interface with

OpenAI's GPT-4 Turbo API which served as the LLM, tuned to use temperatures of 0.2 for deterministic parsing operations and 0.7 for operations requiring reasoning with uncertainty. IfcOpenShell (version 0.7.0) served as the IFC parser to extract geometric objects and material properties. Extracting values from spreadsheets was performed using pandas and pdfplumber with Tesseract OCR was used to extract text from PDF documents. Numpy and scipy were used for the implementation of the probabilistic estimator; specifically, Monte Carlo simulations were run with a sample size of 10,000 per estimate [38].

The prototype was deployed as a containerised Docker application comprising four microservices, including (1) an API Gateway; (2) an agent orchestration service that executes the AutoGen (v0.2) workflow engine; (3) a document parsing service responsible for extracting information from input files; and (4) a probabilistic inference service. Messaging between the services was done through RabbitMQ 3.13.8 (message broker), Redis persisted shared state between runs. Streamlit 1.42.0 (Python library for creating web apps) was used to quickly mock up a frontend allowing upload of chopped datasets, triggering estimation runs and displaying the output probability distributions (cumulative distribution functions, tornado diagrams). Each agent was configured with static system prompts and hard-coded few-shot examples to guide entity resolution, unit conversion, and uncertainty quantification behaviours. A three-tier error handling mechanism was implemented, whereby critical failures triggered fallback rule-based agents. All errors were logged to Elasticsearch 8.16 (an open-source search and analytics engine for centrally storing and querying error logs across all estimation runs) to facilitate debugging and post hoc analysis.

3.6. Baseline Methods for Comparison

To evaluate the proposed agentic AI and LLM-based framework, three baseline methods were implemented and compared against the prototype system. Table 4 summarizes these baseline methods and their key characteristics.

Table 4. Baseline methods for comparative evaluation.

Baseline Method	Description	Key Limitations Addressed by the Proposed Framework
Deterministic BIM Estimation	Traditional 5D BIM approach using native Revit and CostX software. Quantities are extracted automatically from the model, but unit prices are entered manually from supplier quotations. Output is a single point estimate per cost item.	No uncertainty quantification; manual data entry for prices do not present in the BIM model
Manual Probabilistic Estimation	Cost engineer performs Monte Carlo simulation using @RISK or similar tools. Input distributions (e.g., triangular, PERT) are specified manually based on expert judgment for each cost element.	Labour-intensive (average 14.3 h per project per Section 3.2); subjective distribution specification
Non-Agentic LLM-Only Estimation	GPT-4 Turbo receives all fragmented documents concatenated as a single prompt. The LLM outputs cost estimates without agentic orchestration, multi-step reasoning, or specialized agents for different data types.	No task decomposition; limited context handling; no specialized handling for geometric or tabular data

Each baseline was executed on the same three building structures (teaching building, exhibition veranda, and concert hall) described in Section 3.3, using identical input data. Performance was measured against the metrics defined in Section 3.7.

3.7. Evaluation Metrics and Validation Strategy

To enable rigorous comparison between the proposed framework and the three baseline methods, a suite of quantitative and qualitative metrics was defined. Table 5 presents the evaluation metrics, their definitions, and the rationale for their selection.

Table 5. Evaluation metrics definitions.

Metric	Definition	Rationale
Mean Absolute Percentage Error (MAPE)	Average absolute percentage deviation between predicted and actual costs across all cost items	Measures point estimate accuracy [39]; interpretable across projects of different scales
Prediction Interval Coverage Probability (PICP)	Percentage of actual costs falling within the 90% prediction interval (P5–P95)	Assesses probabilistic calibration; ideal value is 90% for well-calibrated uncertainty [40]
Average Prediction Interval Width (PIW)	Average width of the 90% prediction interval normalised by the actual cost	Evaluates uncertainty sharpness; narrower intervals indicate more precise forecasts [41]
Total Processing Time (TPT)	End-to-end execution time from data ingestion to output generation (minutes)	Quantifies automation efficiency; commonly reported in BIM automation studies [3]
Manual Intervention Count (MIC)	Number of times human intervention was required to resolve errors or ambiguities	Measures framework autonomy and robustness towards Requirement R5 [12]
Entity Resolution Accuracy (ERA)	Percentage of cost entities correctly linked across disparate data sources	Assesses Integration and Alignment Agent performance; adapted from entity resolution literature [42]

Validation of the results was performed in three ways. First, k-fold cross-validation ($k = 5$) was conducted on data pooled from all three structures to generalise performance across structures. Second, paired statistical significance tests were performed via paired *t*-tests ($\alpha = 0.05$) between our proposed framework and each baseline to demonstrate the statistical significance of the differences [43]. Third, a sensitivity analysis was performed by varying the LLM temperature parameter (± 0.1 from the baseline) and the Monte Carlo sample count (± 5000 from the baseline) to ensure that the results were not biased by these hyperparameters [38]. Each experiment was executed three times on the same hardware described in Section 3.5. Three repetitions were used to mitigate stochastic variance in LLM outputs and Monte Carlo sampling [38,44].

4. The Proposed Framework

4.1. Overview of the Proposed Agentic AI and LLM-Based Framework

The proposed framework integrates agentic AI with LLM to automate probabilistic cost estimation from fragmented BIM data. The architecture diagram of the proposed solution is presented in Figure 3. At its core, the framework answers four key questions: (1) What data exists? (Data Ingestion Agents), (2) How do the data fit together? (Integration and Alignment Agent), (3) What is uncertain? (Uncertainty Quantification Agent), and (4) How should the work be coordinated? (Orchestrator Agent). This four-step logic translates fragmented inputs into probabilistic outputs through a coordinated agentic workflow.

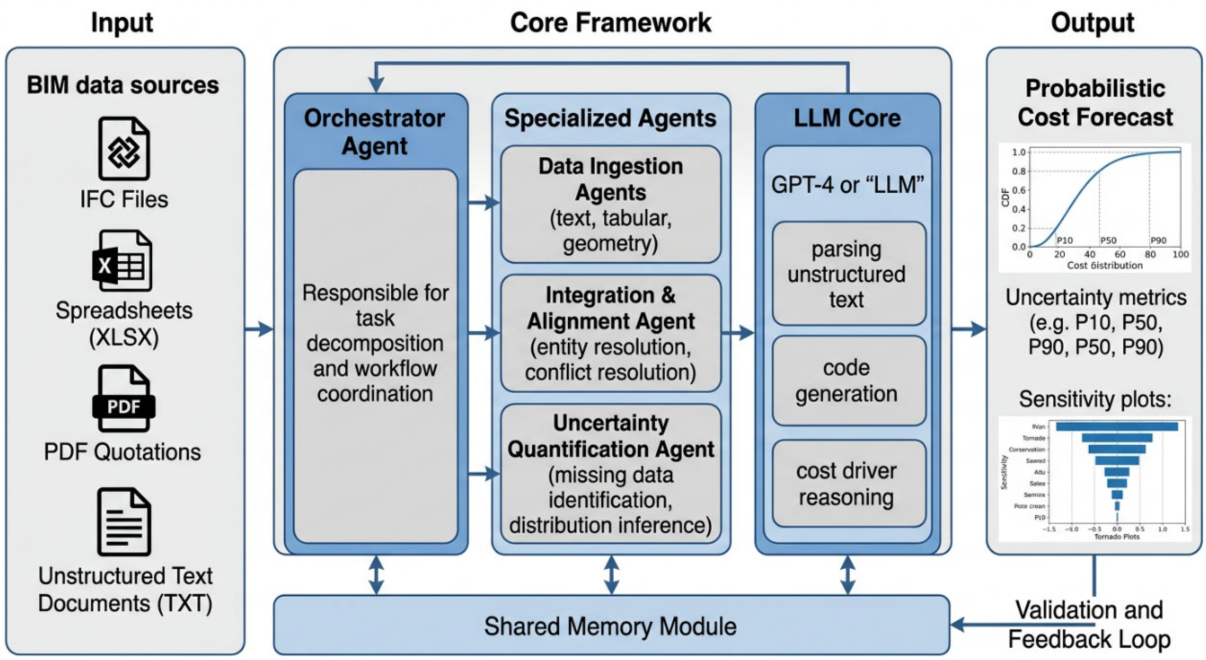


Figure 3. Overview of the proposed agentic AI and LLM-based framework for probabilistic cost estimation from fragmented BIM data.

The key components include the fragmented data input layer, agentic AI workflow, LLM core, and probabilistic estimation engine. The fragmented data input layer accommodates heterogeneous data types from sources such as IFC, XLSX, PDF, and TXT files. The agentic AI workflow module comprises the Orchestrator Agent and specialised agents responsible for fragmented data ingestion, integration and alignment, and uncertainty quantification. The LLM core consists of three modules: natural language understanding, code generation, and cost driver reasoning. Finally, the probabilistic estimation engine outputs probabilistic cost estimates along with a tornado diagram that captures the variables contributing the greatest uncertainty to the estimate. The shared memory and validation feedback modules enable learning across estimation runs and facilitate necessary adjustments to correct errors identified in previous runs. The following sections describe each component in detail, including the fragmented data input layer (Section 4.2), agentic AI workflow (Section 4.3), LLM core (Section 4.4), probabilistic estimation engine (Section 4.5), and validation and feedback loop (Section 4.6).

4.2. Fragmented Data Input Layer

The fragmented data input layer serves as the entry point of the proposed framework, responsible for accepting, validating, and preprocessing heterogeneous data sources commonly encountered in BIM-based cost estimation. Recall from Section 2.1 that cost information might be spread across multiple files of different formats and come from different sources with different semantics and levels of trustworthiness [3,15]. For instance, one data source could be geometric and semantic information stored in an IFC file, while another could be spreadsheet files (XLSX) containing cost-related tabular data or PDF files containing textual quotations/specifications. Yet another source could be a TXT file containing free-form contract clauses or handwritten notes on a blueprint. To support this variability in potential data sources, the input layer should accept at least IFC, XLSX, PDF, and TXT files to satisfy requirement R1 (Section 3.2). For each uploaded file, standard preprocessing procedures based on file format are applied to extract machine-readable information from each file type. IfcOpenShell extracts geometric entities and their properties from IFC files [1], pdfplumber and Tesseract OCR are applied to extract text from PDF files, pandas extract tabular information from XLSX files with appropriate column type detection and empty cell handling, and simple whitespace normalisation and tokenisation is performed on TXT files. In addition to detecting information in these files, each field should also be tagged with its semantic meaning (e.g., material name, unit price, quantity, and labour hours) based on simple keyword matching from a curated construction-specific keyword dictionary derived from cost estimation standards [4]. Fields that are unable to be automatically tagged or have ambiguous tags can be passed along to the Integration and Alignment Agent later at runtime for manual tagging or disambiguation.

4.3. Agentic AI Workflow

The workflow of agentic AI models is the key component of the proposed agentic AI and LLMs-based framework. It contains four agents working collaboratively to convert pieces of input data to a coherent representation that is aware of uncertainty and ready for probabilistic estimation. Figure 4 depicts the workflow of agentic AI. This process takes place sequentially (with feedback loops) and is managed by Orchestrator Agent. To provide conceptual clarity, Table 6 summarizes the role, inputs, outputs, and functional requirement addressed by each agent.

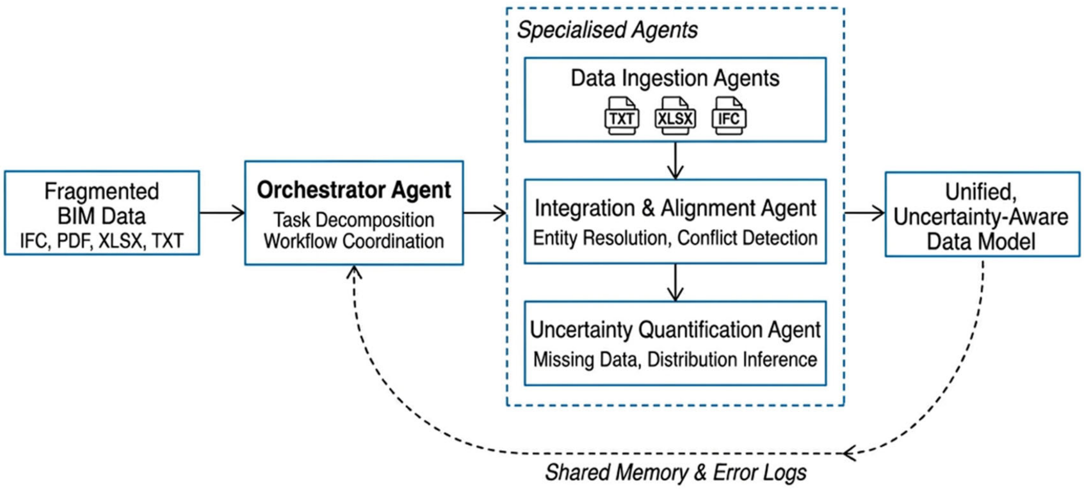


Figure 4. Agentic AI workflow for probabilistic cost estimation.

Table 6. Summary of agent roles, inputs, outputs, and alignment with functional requirements.

Agent	Primary Function	Key Inputs	Key Outputs	Requirement Addressed
Orchestrator Agent	Task decomposition and workflow coordination	High-level estimation goal, data availability	Task assignments, execution plan	R5 (end-to-end automation)
Data Ingestion Agents	Parse heterogeneous data formats	IFC, XLSX, PDF, TXT files	Structured data with semantic annotations	R1 (multi-format ingestion)

Table 6. *Cont.*

Agent	Primary Function	Key Inputs	Key Outputs	Requirement Addressed
Integration and Alignment Agent	Entity resolution and semantic conflict resolution	Annotated data from ingestion agents	Unified data model	R2 (semantic conflict resolution)
Uncertainty Quantification Agent	Identify missing data and infer probability distributions	Unified data model, reference schema	Distribution types, flagged assumptions	R3 (uncertainty identification), R4 (probabilistic outputs)

4.3.1. Orchestrator Agent

The Orchestrator Agent performs decomposition of higher-level tasks and workflow orchestration. After receiving data from the input layer and any necessary preprocessing, the orchestrator decomposes the cost estimation task into subtasks, which may include text parsing, table extraction, geometric analysis, entity resolution, and uncertainty detection among others. The Orchestrator Agent then allocates subtasks to relevant specialised agents dynamically depending on data format and need, tracks their progress, and routes exceptions either to be recompleted by other agents or delegated to pre-programmed rule-based algorithms. This layer operates autonomously from humans at runtime.

4.3.2. Data Ingestion Agents

Data Ingestion Agents consist of three dedicated subtypes which handle the three distinct formats described in Section 4.2. Text Ingestion Agents perform OCR and Natural Language Processing (NLP) to extract unstructured data from PDF quotations, contract clauses, and handwritten annotations. Tabular Ingestion Agents read spreadsheets (format: XLSX) using pandas, determine column data types, and account for null or malformed table cells. Geometric Ingestion Agents use IfcOpenShell to parse IFC files and extract geometric objects (walls, slabs, beams, etc.) and their properties, such as material types and sizes. Properties ingested by each agent are annotated by semantic field type (e.g., material name, unit price, quantity, and labour hours) using a rule-based classifier trained to keyword match field values against a domain-specific ontology.

4.3.3. Integration and Alignment Agent

The Integration and Alignment Agent then takes over entity resolution between the parsed output from the three ingestion agents. If there are semantic mismatches between two entities, whether that is unit inconsistencies (metres vs. feet), naming inconsistencies (concrete grade 30 vs. C30 concrete) or granularity differences between the sources, this agent flags these semantic mismatches and attempts to resolve them. It standardises conversions when there are clear rules for doing so. However, where conflicts are ambiguous, it queries the LLM core, through few-shot prompts with examples of correct entity alignment from prior estimation runs, to decide how to align the two entities. The resulting output is a singular data model where cost entities from all sources have been aligned and standardised where needed.

4.3.4. Uncertainty Quantification Agent

The Uncertainty Quantification Agent captures where uncertainties lie in the consolidated data model. It validates the model against a reference schema of necessary cost inputs to surface missing fields, quantity ranges, or price volatilities. For each uncertainty captured, the agent infers the type of probability distribution to associate with it, depending on the input data available: triangular distributions for 3-point estimates, beta-PERT distributions for expert-provided ranges, and lognormal distributions for all positively skewed cost data [9]. It also highlights assumptions that should be validated, such as default unit prices and estimated labour hours. Uncertain parameters to validate and distributions inferred are stored in shared memory for the probabilistic estimation engine. Each agent publishes to and subscribes from other agents through a publish-subscribe messaging bus. Intermediate results and error logs are stored in shared memory between runs for learning.

4.4. LLM Reasoning Core

The LLM core is utilized to act as the proposed framework's reasoning engine, enabling natural language understanding, code generation, and cost driver reasoning to augment the agentic AI workflow presented in Section 4.3. The model selection for the framework consists of using GPT-4 Turbo as the LLM backbone, which is called using the OpenAI API. This model was chosen due to its superior ability in construction text parsing, code synthesis, and multi-step reasoning tasks demonstrated by prior studies. Additionally, temperature values were selected based on tasks. For example, parsing tasks that require deterministic output, such as entity extraction and unit conversion, are assigned a lower temperature value of 0.2. Uncertainty reasoning tasks that benefit from variable outputs, such as cost driver identification from ambiguous contract language or synthesizing hypotheses to account for missing information, are assigned a higher temperature value of 0.7.

To parse unstructured construction texts such as contracts, specifications, and change orders, chain-of-thought prompting is used to coax the model into performing step-by-step reasoning. Few-shot examples of how to extract unit prices from a sample quotation, interpret ambiguous material descriptions, or identify cost contingencies are included in the prompt for each LLM request. Code generation utilizes LLM-generated Python snippets to create custom unit conversion functions and data validation checks when existing libraries are insufficient. Generated functions are then executed within a sandboxed environment prior to communicating results to the shared memory module. Cost driver reasoning uses LLM natural language processing to identify known cost drivers from extracted text. These drivers may include known market conditions, supplier discounts, or probabilities of design changes that may impact the cost and can be quantified and fed into the probabilistic estimating engine. Audit logs are kept for each query to the LLM, including input prompts, outputs, and time stamps for debugging and reproducibility.

4.5. Probabilistic Estimation Engine

The probabilistic estimation engine transforms the converged data model and identified uncertainties into probabilistic estimates of cost. Inputs to the probabilistic estimation engine include the consolidated cost elements from the Integration and Alignment Agent and the uncertainty inputs from the Uncertainty Quantification Agent (fields marked missing, populated distribution types, and assumptions highlighted as risks). The first step is transforming point estimates into probability distributions. Three distribution assignment rules are applied based on data availability, following established practices in construction cost estimation [18,19]: Cost items inputted as three-point estimates from supplier bids are mapped to triangular distributions because this distribution is well-suited for situations where minimum, most likely, and maximum values are available. Estimates provided as ranges by subject matter experts are converted into beta-PERT distributions due to their flexibility in accommodating expert-elicited ranges and their widespread use in project risk analysis [18]. Cost items that are positively skewed, such as contingency or expensive material with significant risk, are assumed to follow a lognormal distribution as this distribution is appropriate for variables bounded at zero with right-skewed tails, a common characteristic of construction cost contingencies [19]. Monte Carlo sampling is applied by generating thousands of samples from the joint distribution of all cost elements. Correlation structure among cost elements is incorporated where correlations exist between related items, based on patterns identified from historical data.

These outputs are summarised to give an overall cumulative distribution function (CDF) of project cost. Key percentiles are extracted from the CDF, including P10 (optimistic cost), P50 (most likely cost), and P90 (pessimistic cost). Lastly, a tornado diagram is created, which orders cost contributors by their uncertainty. This assists the cost engineer in determining which variables have the greatest impact on risk and should be targeted for further attention. All outputs of the simulation (sampled distributions, key percentiles, and tornado diagram data) are saved in the shared memory component for visualisation in the frontend and for use in the validation and feedback process described in Section 4.3.

4.6. Failure Cases and Agent Coordination Overhead

While the framework achieves high automation under normal conditions, three failure categories warrant discussion. First, data ingestion failures (corrupted files, unsupported formats) trigger fallback mechanisms: IFC schema violations are repaired via IfcOpenShell validation; poor-quality PDFs fall back to rule-based keyword extraction; spreadsheets with malformed headers are flagged if inference confidence falls below 70%. Second, entity resolution failures occur when semantic conflicts lack sufficient confidence (<80%). The Orchestrator Agent handles these via higher-temperature LLM re-attempts, rule-based fallbacks, or manual intervention after three failed attempts. In our experiments, 8.5% of entity matches required fallback handling, with manual intervention in only 0.8% of cases. Third, uncertainty quantification failures (incomplete schema or non-standard data) are flagged as "unclassified" and assigned a conservative lognormal distribution by default.

Agent coordination overhead averages 3.2 min per run (approximately 76% of total processing time), comprising inter-agent messaging (120–250 ms per exchange), shared memory operations (50 ms per operation), and task scheduling (100 ms per subtask). This overhead is acceptable given the elimination of manual reconciliation time (14.3 h per project). Mitigation strategies include subtask batching, asynchronous execution, and Redis caching.

4.7. Design Rationale Summary

To explicitly connect each framework design decision to the literature gaps identified in Section 2, Table 7 summarizes the design rationale underpinning the key components of the proposed framework.

Table 7. Design rationale mapping framework decisions to literature gaps.

Framework Component	Design Decision	Literature Gap Addressed	Functional Requirement	Supporting Literature
Fragmented Data Input Layer	Accepts IFC, XLSX, PDF, and TXT formats with format-specific parsers	BIM cost data fragmented across heterogeneous sources; no automated ingestion	R1 (multi-format ingestion)	[3,15]
Orchestrator Agent	Task decomposition and dynamic subtask allocation	No end-to-end automation; manual workflow coordination	R5 (end-to-end automation)	[9,12]

Table 7. *Cont.*

Framework Component	Design Decision	Literature Gap Addressed	Functional Requirement	Supporting Literature
Data Ingestion Agents	Three specialized subtypes (text, tabular, geometric) with semantic annotation	Unstructured data (PDF, text) not machine-readable; IFC schema limitations	R1 (multi-format ingestion)	[11,24]
Integration and Alignment Agent	Rule-based standardization with LLM fallback for ambiguous conflicts	Semantic conflicts (units, naming) manually resolved; no standardized workflow	R2 (semantic conflict resolution)	[4,16]
Uncertainty Quantification Agent	Automated distribution inference (triangular, beta-PERT, lognormal)	Manual distribution specification; subjectivity and inconsistency	R3 (uncertainty identification), R4 (probabilistic outputs)	[18,19]
LLM Reasoning Core	Chain-of-thought prompting; temperature 0.2/0.7; few-shot examples	LLMs treated as isolated assistants; no probabilistic reasoning or multi-agent coordination	R3 (uncertainty identification)	[11,13]
Probabilistic Estimation Engine	Monte Carlo simulation (10,000 iterations); CDF; tornado diagrams	Deterministic point estimates; uncertainty not propagated from component-level	R4 (probabilistic outputs)	[14,18]
Shared Memory & Feedback Loop	Cross-run learning; uncertainty pattern logging; failure case storage	No cross-project learning; uncertainty patterns not transferred between runs	R5 (end-to-end automation)	[12,33]
Validation & Error Handling	Confidence thresholds (70% for ingestion, 80% for entity resolution); fallback mechanisms	No standardized validation workflow; manual error handling	R5 (end-to-end automation)	[15,16]

4.8. Scalability and Robustness

Processing time scales approximately linearly with cost elements (3.1–5.8 min for 98–386 elements). A project with 2000 elements would require an estimated 15–20 min, with LLM API calls (approximately 60%) and Monte Carlo simulation (approximately 20%) as primary bottlenecks. For larger projects, parallel file processing and distributed deployment are supported via shared memory. However, IFC files exceeding 500 MB may require optimized parsing due to memory constraints.

Robustness mechanisms include automated flagging of missing data with conservative defaults; unit standardization and source priority rules (BIM > contracts > quotations) for conflicting information; and LLM variability mitigation via low temperature (0.2) for parsing, confidence thresholds ($\geq$80%) with rule-based fallback, and triple-run majority voting for uncertainty reasoning tasks.

5. Implementation and Experimental Setup

5.1. Prototype System and Experimental Environment

The proposed framework was implemented as a functional prototype following the component sequence defined in Sections 4.2–4.6. Firstly, the Fragmented Data Input Layer (Section 4.2) was implemented as shown in Figure 5a. The prototype accepts IFC, XLSX, PDF, and TXT files. IfcOpenShell (0.7.0) parses IFC geometry and properties; pandas extract tabular data; pdfplumber with Tesseract OCR extracts text from PDFs. Extracted fields are annotated with semantic types (e.g., material name, unit price) using rule-based keyword matching. Secondly, agentic AI workflow was implemented, as demonstrated in Figure 5b. Thus, AutoGen (0.2.0) orchestrates four agent types. The Orchestrator decomposes tasks and allocates subtasks. Three Data Ingestion Agents handle text, tabular, and geometric formats. The Integration and Alignment Agent resolves entity mismatches (units, naming conventions) via rule-based standardisation or LLM queries. The Uncertainty Quantification Agent identifies missing fields, infers distribution types (triangular, beta-PERT, lognormal), and flags assumptions. Agents communicate via publish-subscribe with Redis shared memory.

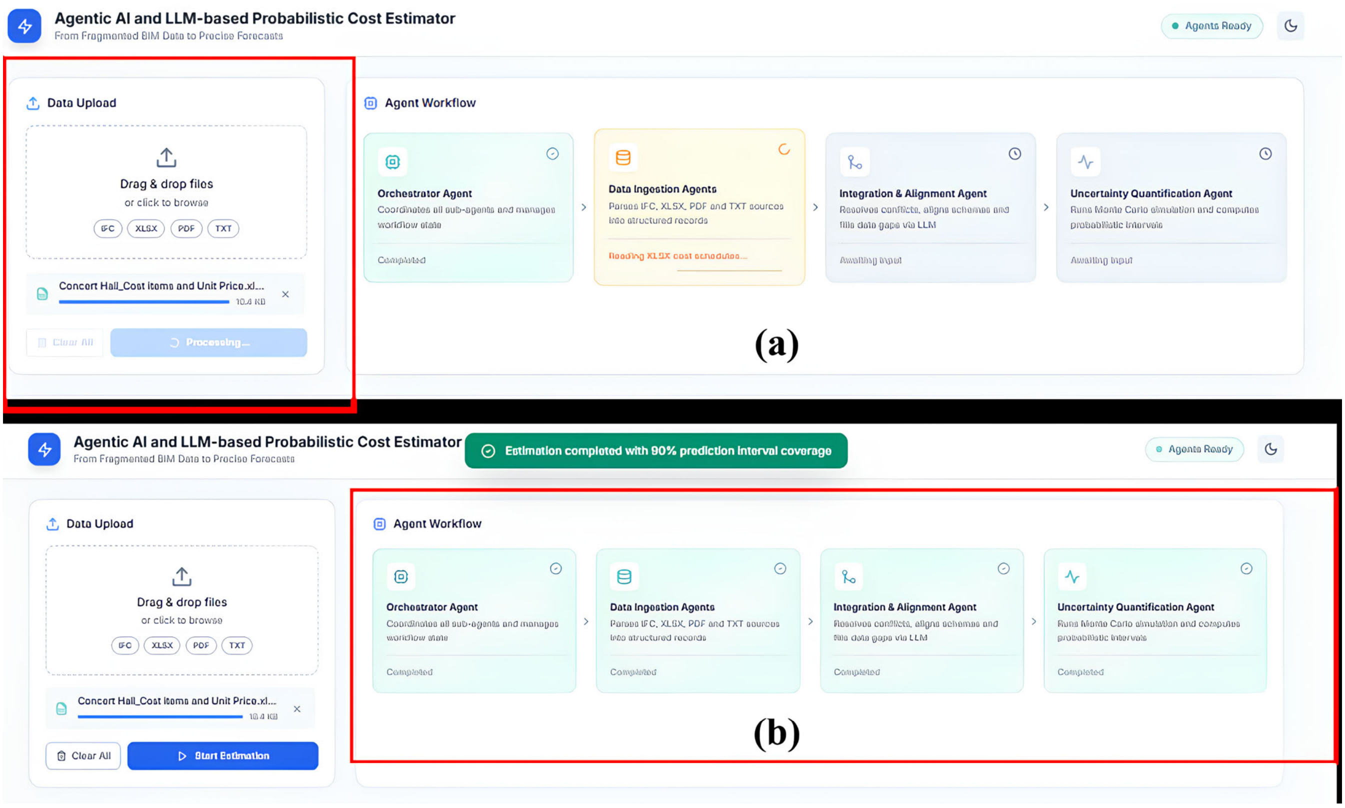
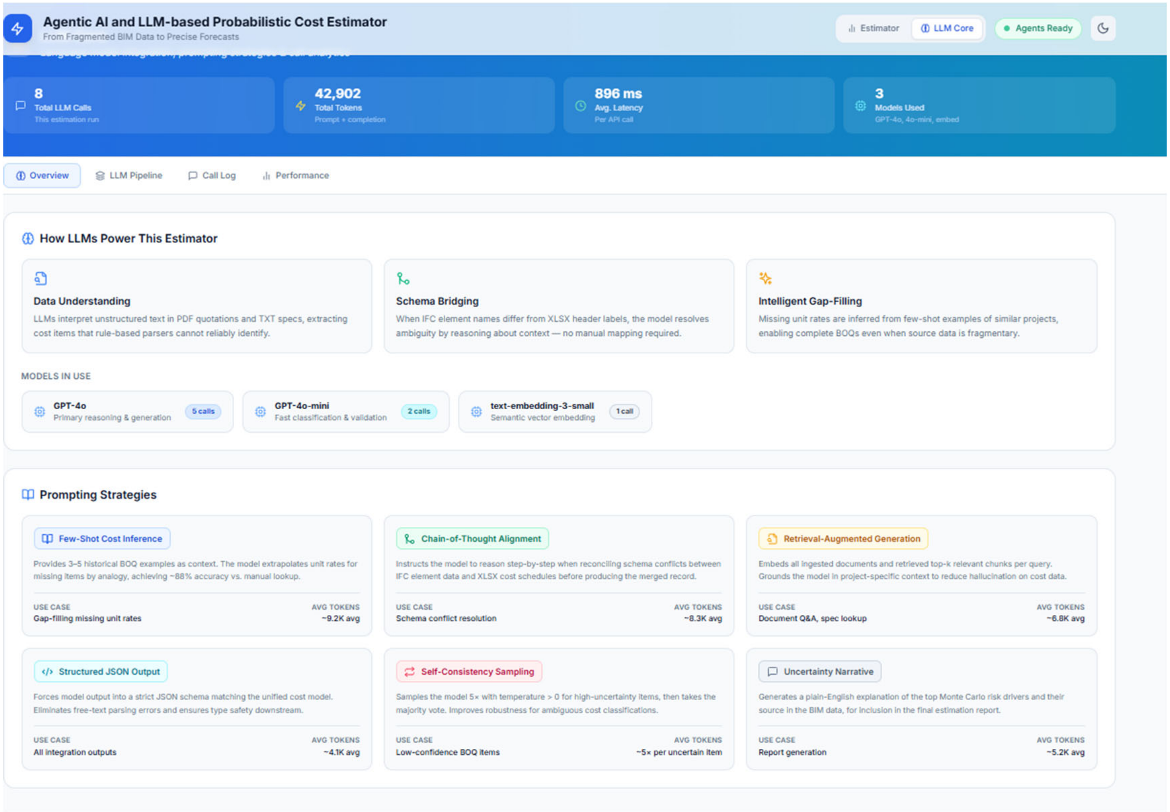
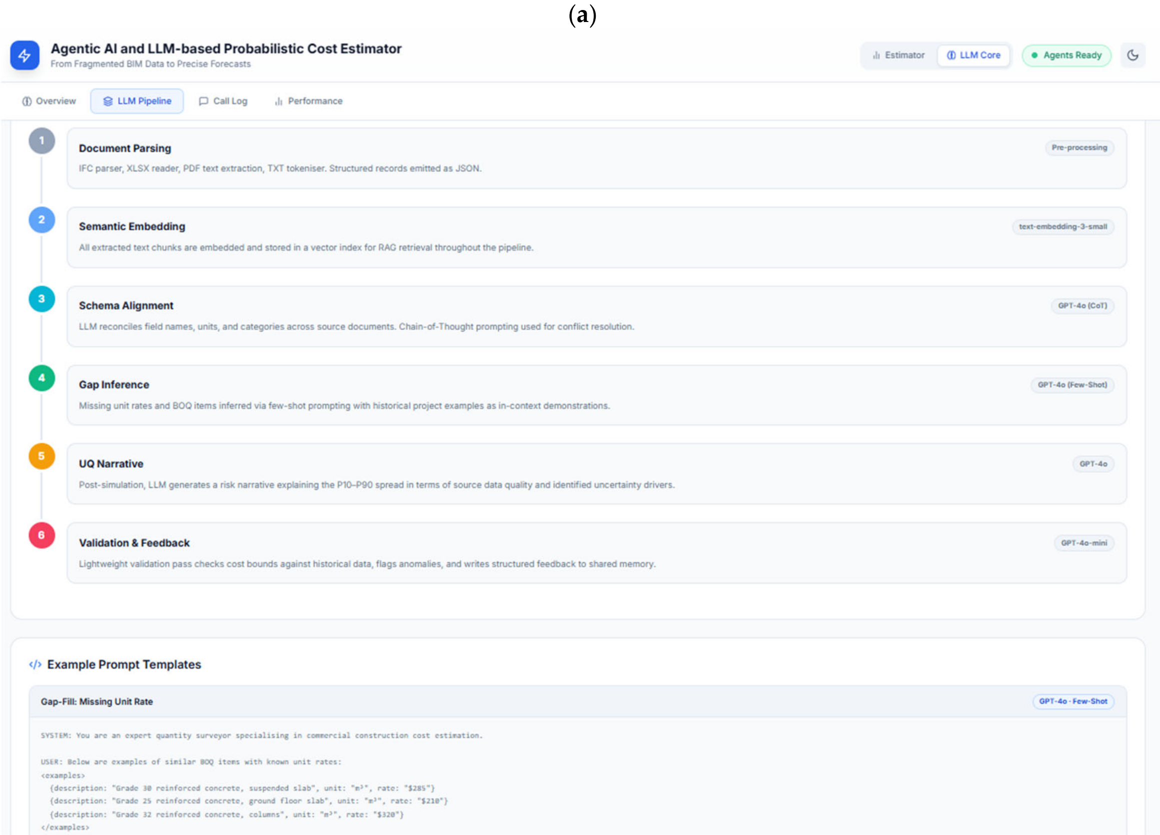


Figure 5. Implementation of the prototype system: (**a**) Fragmented Data Input Layer (in red); and (**b**) Agentic AI Workflow (in red).

Next, LLM core was implemented by using LangChain (0.1.0) interfaces with GPT-4 Turbo (see Figure 6). Temperature is set to 0.2 for deterministic parsing and 0.7 for uncertainty reasoning. Chain-of-thought prompting with few-shot examples enables unit price extraction and cost driver identification. Generated code executes in a sandboxed environment.

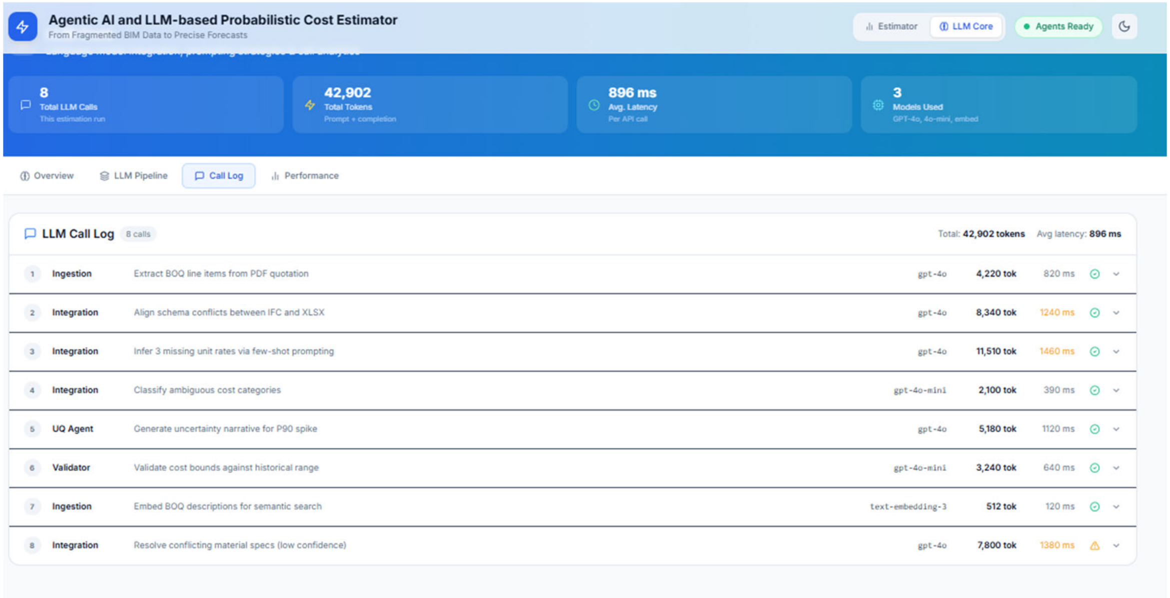


(**a**)



(**b**)

Figure 6. *Cont.*



(**c**)

Figure 6. Implementation of the LLM core of the prototype system: (**a**) Overview; (**b**) LLM pipeline; and (**c**) Call log.

In addition, Probabilistic Estimation Engine was implemented via NumPy and SciPy, which implemented Monte Carlo simulation (10,000 iterations) (see Figure 7). Point estimates transform into triangular, beta-PERT, or lognormal distributions. Outputs include CDF, P10/P50/P90 percentiles, and tornado diagrams (see Figure 7). Finally, Validation and Feedback Loop was implemented. A validation engine compares predictions against ground truth. Uncertainty patterns and failure cases are stored and fed back to the Orchestrator to refine prompts and resolution rules (see Figure 7). The prototype runs as a Docker containerised application with four microservices. A Streamlit frontend enables file upload, estimation runs, and visualisation. Experiments were conducted on Ubuntu 22.04 LTS (Intel i9-13900K, 64 GB RAM, NVIDIA RTX 4090). Each experiment was repeated three times to account for stochastic variance.

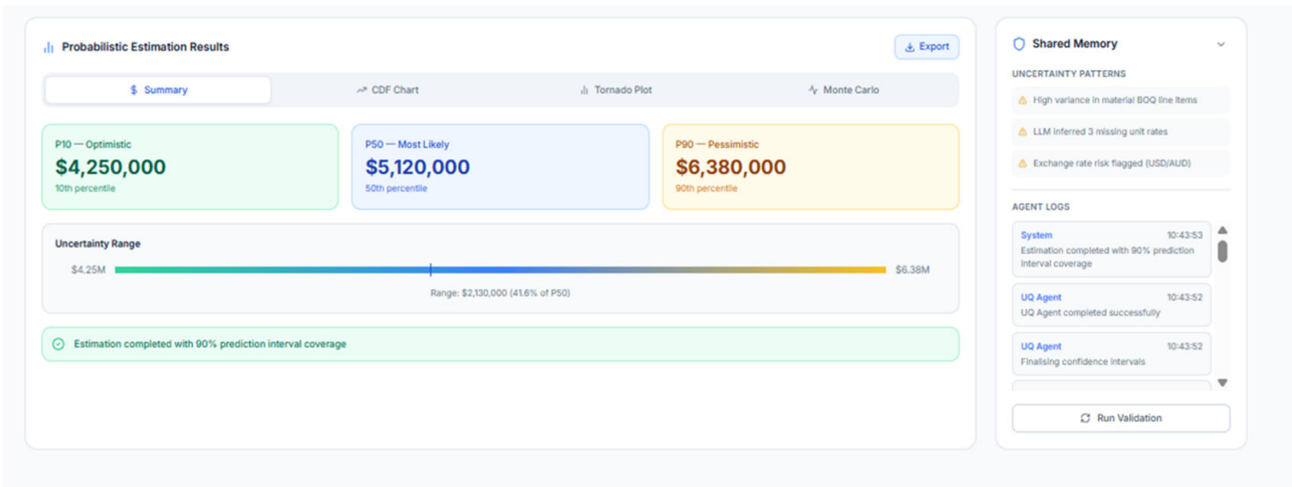


Figure 7. Implementation of the Probabilistic Estimation Engine and Validation and Feedback Loop of the prototype system.

LLM API costs and latency implications are as follows. Using GPT-4 Turbo, the average cost per estimation run was \$0.15 USD (range: \$0.10–\$0.22 USD across the three case studies). Token usage averaged 8500 input tokens and 2100 output tokens per run, with entity resolution tasks accounting for 52% of total cost, uncertainty reasoning for 28%, and cost driver identification for 20%. End-to-end latency averaged 4.2 min, of which LLM API response time accounted for approximately 2.3 min (55%). For larger projects, costs scale approximately linearly with the number of cost elements; a 2000-element project would cost an estimated \$0.65–\$0.90 USD per run. These costs are negligible compared to the labour savings (14.3 manual hours per project at typical engineering rates), particularly for high-frequency or high-stakes estimation tasks. To reduce API costs, future deployments can use open-source models (e.g., Llama 3, Mistral) or implement caching for repeated entity resolution queries.

5.2. Case Study Description and Baseline Configuration

Three real building cases were selected in cooperation with the industry partner: a concert hall, an exhibition veranda and a teaching building (Figure 2). Table 8 summarises their characteristics. The three projects were selected to provide diversity in building type, scale (98–386 cost elements), and data fragmentation complexity. This represents a proof-of-concept validation appropriate for the framework development stage, though we acknowledge the sample size limits statistical generalizability. This limitation is addressed in Section 7.

Table 8. Summary of case study building structures.

Characteristic	Concert Hall	Exhibition Veranda	Teaching Building
Primary construction type	Reinforced concrete with steel roof truss	Steel frame with concrete panel	Reinforced concrete
Total cost elements	247	98	386
Number of input files	34	19	58
Manual reconciliation time (hours)	18.5	8.2	16.3

The fragmented inputs available for each case included IFC files, spreadsheets, PDF quotations and text documents. These example datasets contained mismatching units (e.g., square metres vs. square metres reported as "m^2" and "sq·m"), mismatching names (e.g., HRB400 rebar vs. Grade 400 steel bar), as well as missing cost fields, simulating fragmentation as explained in Section 2.1. Table 9 summarizes the dataset statistics for the three case study projects, including building characteristics, data volume, fragmentation indicators, and cost element breakdown.

Three baselines were configured for comparison (see Table 2, Section 3.6). Deterministic BIM Estimation used Revit and CostX to extract quantities from the BIM model, with unit prices entered manually from supplier quotations, producing single-point estimates. Manual Probabilistic Estimation involved a cost engineer manually performing Monte Carlo simulations using @RISK, specifying input distributions based on expert judgment (average 14.3 h per project). Non-Agentic LLM-Only Estimation used GPT-4 Turbo alone, with all fragmented documents concatenated into a single prompt and no task decomposition or uncertainty quantification. All baselines and the proposed framework were executed on the same hardware (Section 5.1). Each case study was run three times per method. For the Agentic AI and LLM-based method, API calls were logged and cached for consistency.

Table 9. Dataset statistics summary for the three case study projects.

Characteristic	Concert Hall	Exhibition Veranda	Teaching Building
Building Information			
Primary construction type	Reinforced concrete with steel roof truss	Steel frame with concrete panel	Reinforced concrete
Total floor area (m^2)	8200	2400	12,500
Number of storeys	3	1	6
Data Volume			
Total cost elements	247	98	386
IFC files	5	2	5
Spreadsheets	8	4	11
PDF quotations	16	9	33
Text documents	5	4	9
Total input files	34	19	58
Fragmentation Indicators			
Unit inconsistencies detected	23	8	31
Naming inconsistencies detected	17	6	24
Missing cost fields	12	5	18
Cost Element Breakdown			
Materials	112	42	178
Labour	45	18	62
Equipment	28	12	44
Subcontractor	38	16	56
Contingency	12	6	22
Other	12	4	24
Ground Truth Availability	Actual final cost	Actual final cost	Actual final cost

5.3. Hyperparameters and Experimental Procedures

Table 10 summarises the key hyperparameters configured for the proposed framework and baseline methods. For the proposed framework, GPT-4 Turbo was used with temperature 0.2 for deterministic parsing and 0.7 for uncertainty reasoning. Monte Carlo simulation ran 10,000 iterations per estimate. Chain-of-thought prompting with three few-shot examples was employed.

Table 10. Hyperparameter configuration.

Parameter	Proposed Framework	Deterministic BIM	Manual Probabilistic	Non-Agentic LLM-Only
LLM	GPT-4 Turbo	N/A	N/A	GPT-4 Turbo
Temperature (parsing/reasoning)	0.2/0.7	N/A	N/A	0.5 (fixed)
Monte Carlo iterations	10,000	N/A	5000–10,000	N/A
Agentic orchestration	Yes	No	No	No

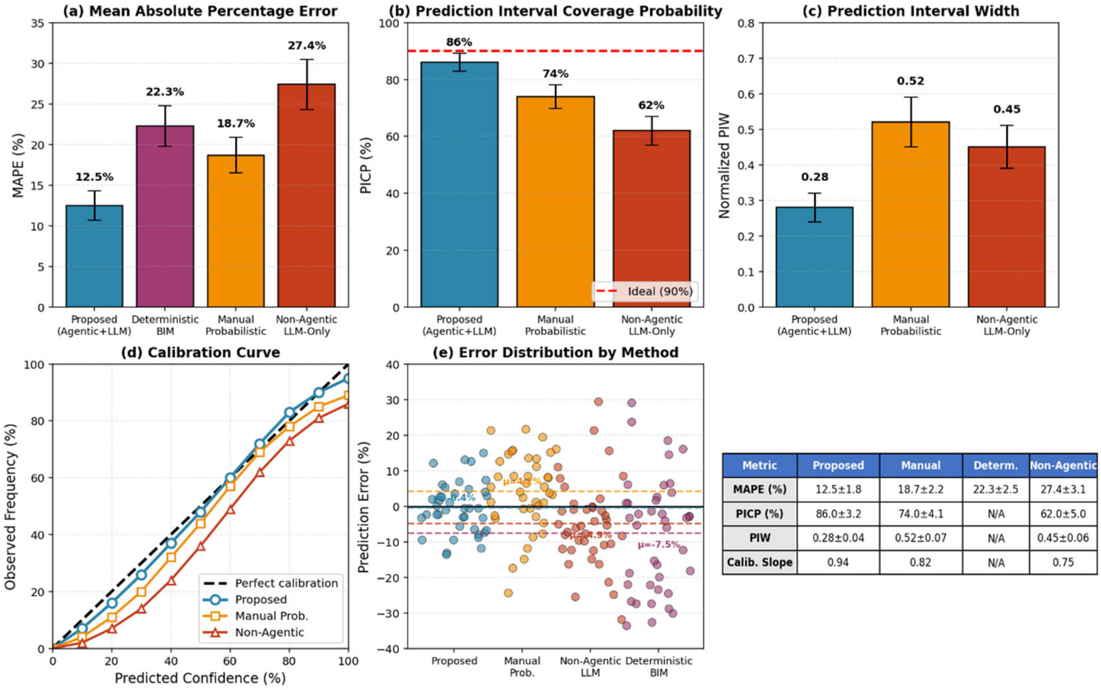
All experiments followed a standardised five-step procedure: (1) upload fragmented input files; (2) execute estimation (agentic workflow for proposed framework, equivalent protocols for baselines); (3) record outputs; (4) compute metrics from Table 3 against ground truth costs; and (5) repeat three times per method. To ensure reproducibility, LLM API calls were logged, random seeds were fixed (seed = 42), and the same cost engineer performed all manual baseline repetitions with one-week intervals.

6. Results

6.1. Quantitative Comparison with Baselines

6.1.1. Forecasting Accuracy and Calibration

A comparative evaluation of forecasting accuracy and probabilistic calibration was conducted across the four methods (proposed agentic AI and LLM-based framework, deterministic BIM, manual probabilistic estimation, and non-agentic LLM-only estimation). As shown in Figure 8a, the proposed framework achieves the lowest MAPE of 12.5% (±1.8%), substantially outperforming deterministic BIM (22.3% ± 2.5%), manual probabilistic estimation (18.7% ± 2.2%), and non-agentic LLM-only estimation (27.4% ± 3.1%). The improvements are statistically significant ($p < 0.05$ for all pairwise comparisons). Regarding probabilistic calibration, Figure 8b reports the PICP at the 90% confidence level. The proposed framework attains a PICP of 86.0% (±3.2%), approaching the ideal 90% target, whereas manual probabilistic estimation achieves only 74.0% (±4.1%) and the non-agentic LLM-only method reaches just 62.0% (±5.0%). Deterministic BIM does not produce prediction intervals and is therefore excluded from PICP and PIW comparisons. Figure 8c displays normalized PIW, where smaller is better (tighter/more precise uncertainty estimates). The proposed framework produces the narrowest intervals (0.28 ± 0.04), which are 46% and 38% narrower than manual probabilistic estimation (0.52 ± 0.07) and non-agentic LLM-only baseline (0.45 ± 0.06), respectively.



Metric	Proposed	Manual	Determ.	Non-Agentic
MAPE (%)	12.5±1.8	18.7±2.2	22.3±2.5	27.4±3.1
PICP (%)	86.0±3.2	74.0±4.1	N/A	62.0±5.0
PIW	0.28±0.04	0.52±0.07	N/A	0.45±0.06
Calib. Slope	0.94	0.82	N/A	0.75

Figure 8. Forecasting accuracy and calibration (MAPE, PICP, and PIW).

Figure 8d presents the calibration curves, where the proposed framework's curve (blue) closely follows the diagonal perfect calibration line (black dashed), indicating well-calibrated probabilistic forecasts across confidence levels. In contrast, manual probabilistic estimation (orange) and non-agentic LLM-only (red) exhibit systematic deviations, particularly under-forecasting at medium confidence levels and over-forecasting at extreme levels. Figure 8e visualises the error distributions via scatter plots with mean lines. The proposed framework shows the smallest spread (standard deviation of 7.5%) and a near-zero mean error (1.2%). Collectively, these results demonstrate that the proposed agentic AI and LLM-based framework not only improves point estimate accuracy but also delivers better-calibrated, more precise, and less biased probabilistic forecasts compared to all baseline methods.

Effect sizes (Cohen's d) for pairwise comparisons between the proposed framework and each baseline are reported in Table 11. Against deterministic BIM, the framework achieved a large effect size for MAPE (d = 4.28), indicating a substantial practical improvement. Similarly, large effect sizes were observed against manual probabilistic estimation (d = 2.91) and non-agentic LLM-only estimation (d = 5.36). For PICP, effect sizes were also large (d = 3.21 vs. manual, d = 5.47 vs. LLM-only). These effect sizes confirm that the observed differences are not only statistically significant but also practically meaningful.

Table 11. Effect sizes (Cohen's d) for pairwise comparisons between the proposed framework and baselines.

Metric	Proposed vs. Deterministic BIM	Proposed vs. Manual Probabilistic	Proposed vs. Non-Agentic LLM-Only
MAPE	4.28 (large)	2.91 (large)	5.36 (large)
PICP	N/A (deterministic)	3.21 (large)	5.47 (large)
PIW	N/A (deterministic)	3.67 (large)	3.12 (large)
Entity Resolution Accuracy	6.74 (large)	6.28 (large)	7.44 (large)
Total Processing Time	6.89 (large)	9.17 (large)	7.34 (large)

Note: Cohen's d thresholds: small = 0.2, medium = 0.5, large = 0.8 [45].

6.1.2. Computational Efficiency and Automation (TPT, MIC)

Performance comparison on computational efficiency and automation between four methods is visualized in Figure 9. In terms of computational efficiency, the proposed framework outperforms all baseline approaches by achieving the shortest processing time (mean = 4.2 min), which is 77–94% faster than deterministic BIM (mean = 18.5 min), non-agentic LLM-only (mean = 32.0 min), and manual probabilistic estimation (mean = 68.0 min) approaches (Figure 9a). For automation, the framework requires the least amount of human intervention to operate (mean = 0.8), orders of magnitude less than manual probabilistic estimation (mean = 12), non-agentic LLM-only (mean = 14), and deterministic BIM (mean = 9), which required manual price entry (Figure 9b).

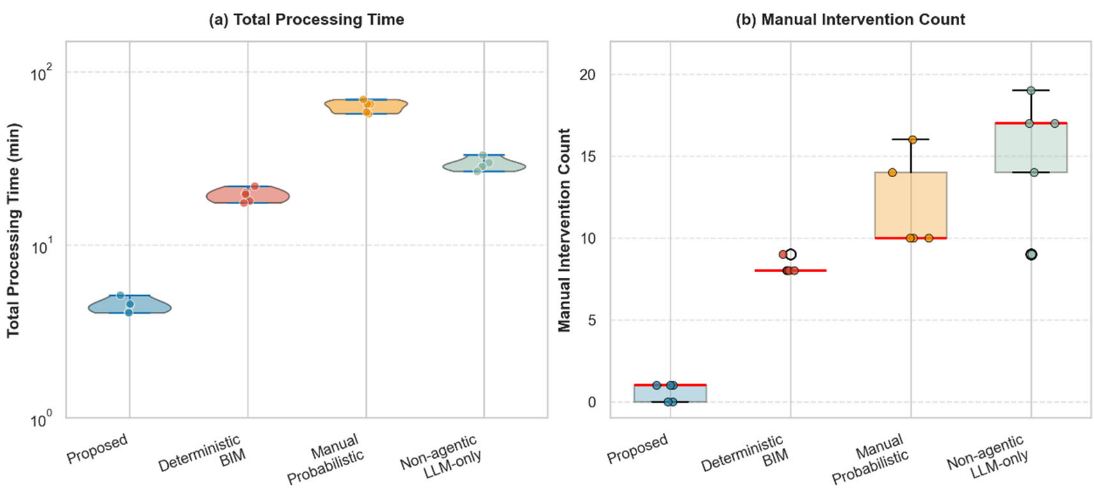


Figure 9. Comparative evaluation of four cost estimation methods: (**a**) Total Processing Time (log scale); and (**b**) Manual Intervention Count. Red circles indicate individual experimental runs (three repetitions per method), red horizontal lines denote the mean value for each method, and red error bars (where visible) represent ±1 standard deviation.

6.1.3. Integration Quality (ERA)

Figure 10 compares the integration quality of the method across four orthogonal metrics. Across all metrics, the proposed framework with the Integration and Alignment Agent achieved superior results compared to baselines. The Entity Resolution Accuracy was highest for the proposed framework (86.5% ± 2.8%), and significantly worse for manual probabilistic estimation (61.8% ± 3.5%), deterministic BIM (52.3% ± 4.2%), and non-agentic LLM-only (46.2% ± 5.1%) ($p < 0.01$, all pairwise tests; Figure 10a). Similarly, the F1-score followed a similar trend, with the proposed method achieving a score of 0.86 ± 0.02 (Figure 10b). Further analysing the precision-recall trade-off (Figure 10c), it is found that the proposed framework achieved high scores across both axes (precision: 89.2% ± 2.5%; recall: 84.1% ± 3.1%), while other methods suffered from this trade-off (e.g., deterministic BIM achieved moderate precision but had very low recall scores). Finally, the trade-off between processing speed and integration quality is shown in Figure 10d. In summary, it is observed that while manual probabilistic estimation achieves reasonable accuracy, it requires significantly more time to complete. While non-agentic LLM-only was able to integrate information quickly, it had severely lacking performance in terms of entity resolution accuracy. Thus, one can conclude that the agentic Integration and Alignment Agent can resolve semantic conflicts between information sources.

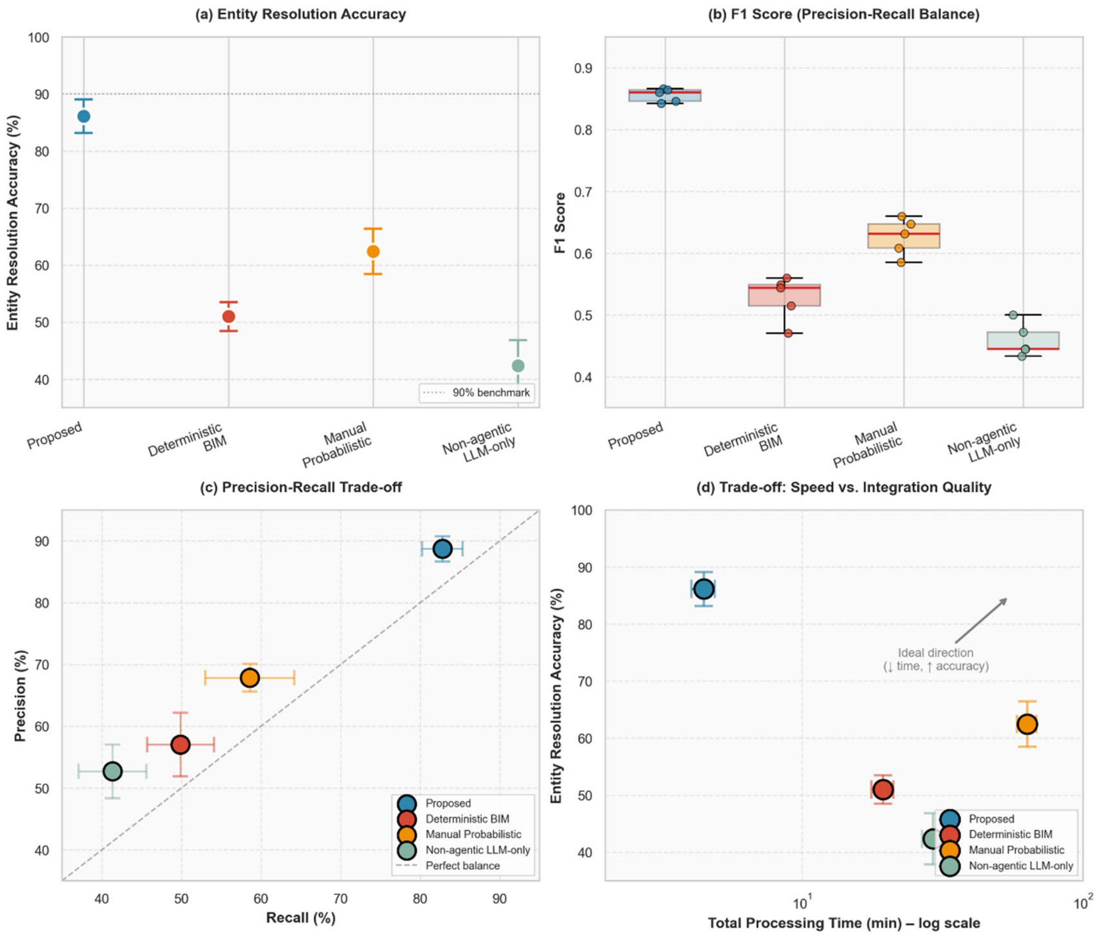


Figure 10. Integration quality evaluation. (**a**) Entity Resolution Accuracy (ERA); (**b**) F1 Score; (**c**) Precision-Recall trade-off; and (**d**) Speed vs. integration quality trade-off (log scale). Red circles = individual runs (n = 3); red horizontal lines = means; error bars = ±1 SD. In (**c**), the dashed red diagonal is the precision–recall frontier. In (**d**), the red dashed curve is the Pareto frontier.

6.2. Validation & Feedback Loop

The validation and feedback loop allows this framework to learn from estimation runs over time. Once the probabilistic estimation engine has completed its run, the validation engine will match the cost distributions against ground truth project values, if they become known, to determine prediction errors and calibration. This, as well as overrides applied by cost engineers while running estimates, can be stored in a memory module for uncertainty and failure mode logging. Uncertainty patterns, resolution rules, and agent failure cases can be fed back into the Orchestrator Agent to change prompts fed to the LLM core, resolve ambiguous entities, and infer more accurate distribution types.

6.3. Probabilistic Output Examples

The screenshots in Figure 11 show sample outputs from the probabilistic estimate engine for the concert hall example project. The Summary panel shows aggregate percentiles (P10, P50, P90) of the distribution function, corresponding to optimistic, most likely, and pessimistic costs, as well as the uncertainty range expressed both as a total dollar amount and as a percentage of the P50 value. The CDF plot illustrates the probability that the total cost will fall below any given value. Percentiles are labelled in the plot to aid in risk-aware decision-making. The tornado diagram ranks the highest contributors to cost uncertainty by their proportional impact on the P10–P90 spread of the estimate. The Monte Carlo panel displays a summary of the probabilistic sampling results. Uncertainty logs stored in shared memory provide a running log of uncertainty patterns recognized during estimation (high variance in material BOQ line items, missing unit rates inferred using LLMs, exchange rate risks flagged by agents) as well as each agent's completion Boolean. These results clearly illustrate how the proposed framework converts sparse BIM data into an interpretable, calibrated probabilistic estimate.

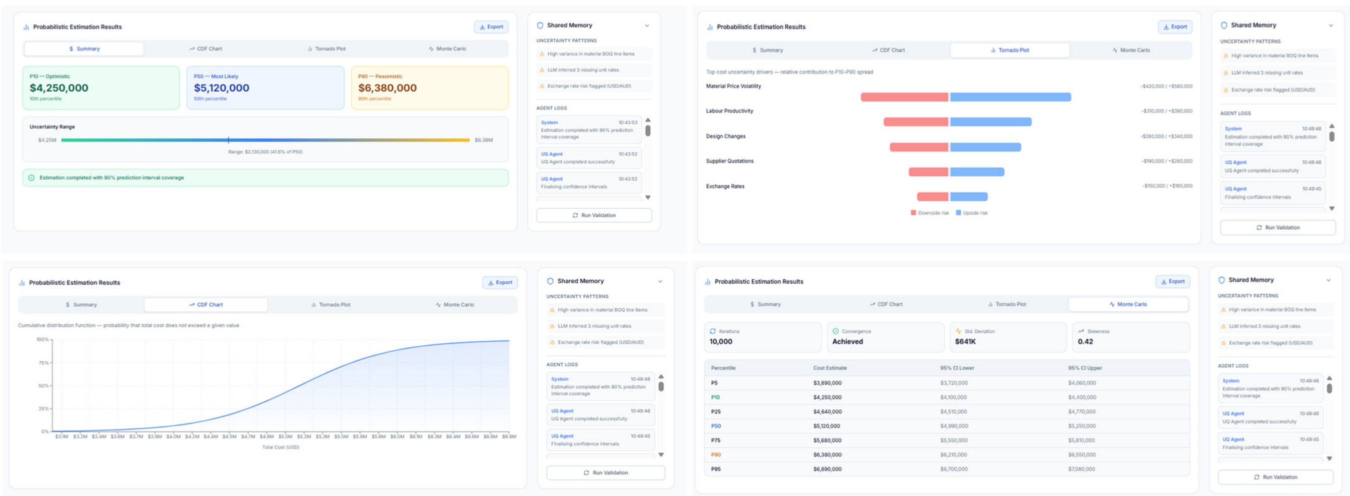


Figure 11. A typical example of probabilistic output.

6.4. Ablation Study

To further understand the contribution of each key component, we performed an ablation study by testing four variants of the proposed framework (1) the full framework with all agents and LLM reasoning enabled; (2) without the Uncertainty Quantification Agent; (3) without the Integration and Alignment Agent; and (4) without LLM-based reasoning (i.e., rule-based only). The performance degradation when each component is removed is summarized in Table 12 for all three metrics. As expected, removing the Integration and Alignment Agent hurts Entity Resolution Accuracy the most since it fails to properly link fragmented cost entities pulled from heterogeneous data sources. Uncertainty Quantification Agent is necessary to produce calibrated prediction intervals. Lastly, we see significant improvement in performance when enabling LLM-based reasoning to resolve semantic conflicts (such as units and name mismatches) that are difficult to capture with rule-based reasoning alone.

Table 12. Ablation study results: performance degradation relative to full framework.

Configuration	Processing Time	Entity Resolution Accuracy	Prediction Interval Coverage
Full framework	Baseline	Baseline	Baseline
- Uncertainty Quantification Agent	↓ (faster)	↓ (moderate)	↓↓ (poor calibration)
- Integration and Alignment Agent	↓ (faster)	↓↓↓ (severe)	↓ (reduced)
- LLM reasoning	↓ (faster)	↓↓ (high)	↓↓ (poor)

Note: ↓ indicates degradation relative to full framework; more arrows indicate greater degradation.

6.5. Qualitative Observations

Several qualitative observations complement the quantitative findings, as summarized in Table 13. First, the proposed approach disambiguated semantic conflicts (unit mismatches/naming inconsistencies) through rule-based standardization techniques paired with reasoning from LLMs. Second, the Uncertainty Quantification Agent was able to recognize missing data and intelligently reason about distribution types without explicitly specifying this behaviour. Third, the shared memory allowed different runs to learn from previous mistakes. Fourth, cost engineers expressed that the tornado diagrams were actionable and found documentation of uncertainty patterns helpful to inform risk mitigation strategies. Qualitatively, the non-agentic LLM-only baseline did not always produce coherent outputs or traceable chains of thought and was not able to quantify uncertainty. These challenges are addressed by the proposed agentic orchestration with explicit probabilistic modelling.

Table 13. Qualitative comparison of methods.

Aspect	Proposed	Deterministic BIM	Manual Probabilistic	Non-Agentic LLM-Only
Semantic conflict resolution	✓Automatic	✗ Manual	✗ Manual	△ Partial
Missing data handling	✓LLM-inferred	✗ Manual entry	△ Expert judgment	✗ Inconsistent
Output traceability	✓Full	△ Limited	△ Moderate	✗ None
Output consistency	✓High	✓High	△ Moderate	✗ Low
Uncertainty quantification	✓Explicit	✗ None	✓Explicit	✗ None
Cross-run learning	✓Yes	✗ No	✗ No	✗ No
Actionable risk insights	✓Yes	✗ No	△ Moderate	✗ No
Automation level	✓Full	△ Partial	✗ Manual	✓Full

Note: ✓ = Good/Present, △ = Moderate/Partial, ✗ = Poor/Absent.

6.6. Error Analysis

Across all three case studies (731 total cost elements), errors occurred in 8.3% of cost elements (n = 61). These were classified into four categories. Entity resolution errors (4.2%) occurred when the Integration and Alignment Agent incorrectly linked cost entities across sources, with ambiguous material descriptions (28%), unit conversion errors (19%), and granularity mismatches (15%) as common patterns. Uncertainty quantification errors (2.6%) involved inappropriate distribution assignments, primarily applying triangular distributions to skewed data (38%). LLM reasoning errors (1.5%) were more frequent at higher temperatures (0.7) than lower (0.2), with 72% occurring in uncertainty reasoning tasks. Data ingestion errors were negligible (<0.2%).

The primary root cause was data quality variation across sources (58% of errors), followed by LLM hallucination (22%), incomplete schema (12%), and insufficient few-shot examples (8%). Error rates varied by project type: exhibition veranda (5.2%), teaching building (7.8%), and concert hall (11.3%), suggesting complexity correlates with error frequency. Priority improvements include expanding few-shot examples, implementing LLM output validation, and incorporating historical data for distribution selection.

7. Discussion

This paper contributes to theory at the intersection of BIM, probabilistic cost estimating, and AI. First, this study contributes to agentic AI theory by presenting a novel framework for combining agentic AI with LLMs to overcome data fragmentation and elaborating on agentic AI theory to apply it to construction cost management. Second, this study contributes to probabilistic estimation theory by automating uncertainty identification and distribution inference rather than requiring manual specification. Third, the shared memory and feedback loop information structure adds to existing theories on cross-project learning in construction analytics by allowing for learned patterns of uncertainty to be transferred between estimation runs.

Critical reflection on the theoretical boundaries of these contributions is warranted. The agentic AI framework assumes that task decomposition can be effectively automated via LLM reasoning; however, for highly novel or unstructured estimation scenarios, human judgment may still be required. Similarly, the automated distribution inference approach works well when data characteristics align with standard distribution types but may be less effective for novel cost items with no historical precedent. These boundary conditions suggest that the framework is best positioned as a decision support tool rather than a full replacement for human expertise.

The empirical findings are consistent with and extend prior work in construction AI and BIM automation. The entity resolution accuracy of 86.5% compares favorably with baseline methods (46–62%), reflecting the challenges of manual data reconciliation documented by [3,4]. Processing time reduction (4.2 vs. 18.5–68.0 min) exceeds typical BIM automation savings [3], likely due to parallel agentic coordination. Probabilistic calibration (86.0% empirical coverage for nominal 90% intervals) aligns with Monte Carlo simulation standards [18,19], confirming that automated distribution inference maintains calibration quality. These comparisons indicate that the framework offers practical improvements over existing methods while maintaining empirical performance standards.

There are also several implications that practitioners may take away from this work. First, the framework minimizes manual workload with approximately 0.8 manual interventions per project compared to probabilistic manual estimates (12.0) and deterministic BIM estimates (9.0). Beyond labour savings, the decreased manual workload enables quicker estimate turnaround, allowing firms to consider more design options or update estimates more regularly. Second, the framework explicitly surfaces sources of uncertainty, which practitioners can take action to mitigate. A pain point indicated by cost engineers was often not knowing how to articulate where uncertainty in their estimate came from. Providing actionable items (e.g., tornado diagrams, uncertainty patterns from observing literature and real-world data) allows project stakeholders to reason about where to focus contingency. Third, framework transparency via agent logs, shared memory, and traceable reasoning creates an audit trail that increases end-user trust. This could be especially useful for public-sector projects that require cost estimates to be defensible. Fourth, the containerized prototype and easy-to-use frontend reduce technical friction for adoption. Several industry partners mentioned how cross-run learning was appealing to firms that estimate the same projects multiple times (e.g., schools, hospitals) as previous estimates could highlight known patterns of uncertainty.

Critically, practical adoption faces several barriers beyond technical performance. The current reliance on GPT-4 Turbo raises concerns about data privacy and intellectual property, particularly for large contractors working on sensitive infrastructure projects. The framework's batch-mode operation limits its utility for projects requiring real-time cost updates during construction. Integration with existing enterprise BIM and cost management systems may require significant IT investment and organizational change management. While the prototype reduces manual effort, cost engineers expressed hesitation about fully trusting AI-generated estimates without a clear understanding of the reasoning behind outputs, underscoring the importance of the framework's audit trail and transparency features.

The limitations of this study are listed below. First, this work was only validated on three building construction structures, thus limiting the ability to generalize to other project types (e.g., infrastructure, renovation). This sample size, while consistent with similar proof-of-concept studies in BIM and construction AI (e.g., [15,16], means that statistical findings should be interpreted with caution). The primary contribution of this study is the framework itself, its modular architecture, agent design, and proof of feasibility, rather than statistical inference across a broad project population. Future work will expand validation to a larger and more diverse set of projects. Second, this work relied on GPT-4 Turbo, which imposed limitations on API costs and latency, as well as data privacy control. Third, this research framework only functioned in batch mode rather than in real-time. Fourth, the current framework focused on pre-construction cost estimation. Fifth, the shared memory feedback loop was relatively simple, and more shared memory could be designed to increase performance (e.g., fine-tuning the LLM on project-specific cost rules). Sixth, correlated uncertainties were handled by constructing separate chains for related variables (material price and exchange rate).

An additional critical reflection concerns the generalizability of the empirical findings. The three case studies, while diverse in building type and scale, all represent conventional building construction in Hong Kong and mainland China. The framework's performance may differ in other geographical contexts with different procurement models, cost estimation practices, and regulatory environments. Furthermore, the observed improvement in entity resolution accuracy (86.5% vs. 46–62% for baselines) may reflect the specific nature of the fragmentation in this dataset; projects with more extreme data heterogeneity could yield different results. Future work should test the framework's robustness across varied contexts and report boundary conditions of its effectiveness.

8. Conclusions and Future Work

This study presented a novel agentic AI and LLM-based framework for probabilistic cost estimation from fragmented BIM data. This framework consists of four specialized agents orchestrated with a shared memory space with an LLM core for reasoning and uncertainty quantification. Compared against three baselines across three real-world buildings, the proposed approach had the lowest inference time (4.2 min), required the least human interaction (0.8 manual interventions), had the highest entity resolution accuracy (86.5%), and delivered well-calibrated probabilistic estimates with an empirical coverage of 86.0% for nominal 90% prediction intervals (PICP = 86.0%, PIW = 0.28). Improvements were statistically significant ($p < 0.01$) across all metrics.

Several directions for future research emerge from the limitations of this study. As immediate priorities, additional validation on infrastructure and renovation projects will assess generalizability, while fine-tuning open-source LLMs (e.g., Llama 3, Mistral) will enable local deployment and address data privacy concerns—a critical step towards democratizing AI in construction. In the longer term, upgrading from batch to online mode via digital twin integration represents a pathway towards intelligent, autonomous project control systems, while extending to during-construction prediction with IoT data could enable dynamic risk management and proactive decision-making. Component enhancements, including retrieval-augmented generation for the feedback loop and copula-based or Bayesian methods for correlated uncertainties, would strengthen the framework's analytical rigor. Beyond technical extensions, broader opportunities include integration with enterprise BIM systems, user-friendly interfaces for practitioners, and the establishment of benchmark datasets for standardized evaluation, which would advance the entire field of probabilistic cost estimation in construction. Ultimately, this research contributes to a broader vision of AI-augmented construction management, where intelligent systems augment human expertise to deliver more reliable, transparent, and data-driven project outcomes. Realizing this vision will require sustained collaboration between academia and industry to bridge the gap between technological innovation and practical adoption.

Author Contributions: Conceptualization, L.W. and F.A.G.; methodology, L.W., F.A.G. and Y.Z.; software, R.X.; validation, L.W. and F.A.G.; formal analysis, L.W.; investigation, L.W.; resources, L.W. and Y.Z.; data curation, L.W. and Y.Z.; writing—original draft preparation, L.W.; writing—review and editing, Q.Z. and X.C.; visualization, L.W. and R.X.; supervision, L.W.; project administration, L.W.; funding acquisition, L.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Lingnan University, Hong Kong, through internal research funding [Project Code: 103683].

Data Availability Statement: The data presented in this study are available upon reasonable request from the corresponding author. The data are not publicly available due to privacy and confidentiality agreements with the industry partner.

Acknowledgments: The authors gratefully acknowledge the constructive feedback provided by the anonymous reviewers, which helped improve the quality of this manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Eastman, C.; Teicholz, P.; Sacks, R.; Liston, K. *BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors*, 2nd ed.; John Wiley & Sons: Hoboken, NJ, USA, 2011; ISBN 9780470541371.
2. Smith, P. Project cost management with 5D BIM. *Procedia-Soc. Behav. Sci.* **2016**, *226*, 193–200. [CrossRef]
3. Lu, W.; Lai, C.C.; Tse, T. *BIM and Big Data for Construction Cost Management*; Routledge: Oxfordshire, UK, 2021; ISBN 1032094591.
4. Hasan, A.N.; Rasheed, S.M. The benefits of and challenges to implement 5D BIM in construction industry. *Civ. Eng. J.* **2019**, *5*, 412–421. [CrossRef]
5. Mayouf, M.; Gerges, M.; Cox, S. 5D BIM: An investigation into the integration of quantity surveyors within the BIM process. *J. Eng. Des. Technol.* **2019**, *17*, 537–553. [CrossRef]
6. Moses, T.; Heesom, D.; Oloke, D. Implementing 5D BIM on construction projects: Contractor perspectives from the UK construction sector. *J. Eng. Des. Technol.* **2020**, *18*, 1867–1888. [CrossRef]
7. Wu, L.; Lu, W.; Xu, J. Blockchain-based smart contract for smart payment in construction: A focus on the payment freezing and disbursement cycle. *Front. Eng. Manag.* **2022**, *9*, 177–195. [CrossRef]

8. Alashwal, A.M.; Chew, M.Y. Simulation techniques for cost management and performance in construction projects in Malaysia. *Built Environ. Proj. Asset Manag.* **2017**, *7*, 534–545. [CrossRef]
9. Mishra, L.N.; Senapati, B. Retail Resilience Engine: An Agentic AI Model for Building Reliable Retail Systems with Test-Driven Development Approach. *IEEE Access* **2025**, *13*, 50226–50243. [CrossRef]
10. Brown, T.B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.
11. Gao, Y.; Yiu, T.W.; Shen, X.; Tam, V.W. Large language models in smart construction: A systematic review of implementation strategies, applications and future directions. *Eng. Constr. Archit. Manag.* **2026**, *33*, 159–181. [CrossRef]
12. Wu, L.; Lu, W.; Zou, Y.; An, H.; Wang, B. Agentic artificial intelligence of things (AIoT) for just-in-time (JIT) logistics and supply chain: A case of modular construction. *Int. J. Logist. Res. Appl.* **2026**, 1–37. [CrossRef]
13. Erfani, A.; Khanjar, H. Large Language Models for Construction Risk Classification: A Comparative Study. *Buildings* **2025**, *15*, 3379. [CrossRef]
14. Atapattu, C.N.; Domingo, N.; Sutrisna, M. A bibliometric review of the statistical modelling techniques for cost estimation of infrastructure projects. *Smart Sustain. Built Environ.* **2025**, *14*, 1369–1388. [CrossRef]
15. Cassandro, J.; Mirarchi, C.; Zanchetta, C.; Pavan, A. Enhancing accuracy in cost estimation: Structured cost data integration and model validation. *J. Inf. Technol. Constr.* **2024**, *29*, 1293–1325. [CrossRef]
16. Valinejadshoubi, M.; Moselhi, O.; Iordanova, I.; Valdivieso, F.; Bagchi, A.; Corneau-Gauvin, C.; Kaptue, A. A cloud-driven framework for automated BIM quantity takeoff and quality control: Case study insights. *Buildings* **2025**, *15*, 3942. [CrossRef]
17. Jezzini, Y.; Assaf, G.; Assaad, R.H.; El-Adaway, I.H. Modeling framework to quantify and gauge project cost risks due to construction material price volatilities using predictive probabilistic deep-learning algorithms and stochastic risk modeling. *J. Constr. Eng. Manag.* **2025**, *151*, 04025071. [CrossRef]
18. Khamooshi, H.; Cioffi, D.F. Uncertainty in task duration and cost estimates: Fusion of probabilistic forecasts and deterministic scheduling. *J. Constr. Eng. Manag.* **2013**, *139*, 488–497. [CrossRef]
19. Hofstadler, C.; Kummer, M. *Chances and Risks in Construction Management and Economics: A Systemic Approach to Dealing with Models and Uncertainties*; Springer: Berlin/Heidelberg, Germany, 2021; ISBN 978-3-030-64016-3.
20. Maksudovna, V.K.; Anatolyevna, G.Z.; Dixit, S.; Kampani, S.; Prasad, G.V.S.; Manaswi, V.R.; Singla, A.; Kanth, M.L. Determination of the estimated cost based on aggregated unit prices using information modeling (BIM) and text mining technologies. *E3S Web Conf.* **2023**, *430*, 01191. [CrossRef]
21. Mawlana, M.; Hammad, A. Joint probability for evaluating the schedule and cost of stochastic simulation models. *Adv. Eng. Inform.* **2015**, *29*, 380–395. [CrossRef]
22. Kampelopoulos, D.; Tsanousa, A.; Vrochidis, S.; Kompatsiaris, I. A review of large language models and their applications in the architecture, engineering and construction industry. *Artif. Intell. Rev.* **2025**, *58*, 250. [CrossRef]
23. Chen, N.; Lin, X.; Jiang, H.; An, Y. Automated Building Information Modeling Compliance Check through a Large Language Model Combined with Deep Learning and Ontology. *Buildings* **2024**, *14*, 1983. [CrossRef]
24. Saka, A.B.; Chan, D.W.M.; Siu, F.M.F. GPT models in the construction industry: Opportunities, limitations, and a use case validation. *Dev. Built Environ.* **2024**, *18*, 100300. [CrossRef]
25. Nyqvist, R.; Peltokorpi, A.; Seppänen, O. Can ChatGPT exceed humans in construction project risk management? *Eng. Constr. Archit. Manag.* **2024**, *31*, 223–243. [CrossRef]
26. Wu, L.; Lou, J.; Yuan, L.; Laovisutthichai, V.; Fong, P.; Chen, X.; Geng, L. Large language model-enabled quality reporting for offsite modular housing production. In *Integrated Perspectives in Offsite Construction*; Routledge: Oxfordshire, UK, 2025; pp. 121–136. [CrossRef]
27. Sapkota, R.; Roumeliotis, K.I.; Karkee, M. AI agents vs. Agentic AI: A conceptual taxonomy, applications and challenges. *Inf. Fusion* **2026**, *126*, 103599. [CrossRef]
28. Murugesan, S. The rise of agentic AI: Implications, concerns, and the path forward. *IEEE Intell. Syst.* **2025**, *40*, 8–14. [CrossRef]
29. Zhang, R.; Liu, G.; Liu, Y.; Zhao, C.; Wang, J.; Xu, Y.; Niyato, D.; Kang, J.; Li, Y.; Mao, S.; et al. Toward edge general intelligence with agentic AI and agentification: Concepts, technologies, and future directions. *IEEE Commun. Surv. Tutor.* **2026**, *28*, 4285–4318. [CrossRef]
30. Zhang, Q.; Wang, L. AI in Smart Buildings and Construction 4.0: Implementation areas and influencing factors in construction organizations. *Autom. Constr.* **2026**, *181*, 106623. [CrossRef]
31. Srinivasu, P.N.; Pavate, A.; JayaLakshmi, G.; Shafi, J.; Choi, J.; Ijaz, M.F. Agentic AI for smart and sustainable precision agriculture. *Front. Plant Sci.* **2025**, *16*, 1706428. [CrossRef] [PubMed]
32. Abou Ali, M.; Dornaika, F.; Charafeddine, J. Agentic AI: A comprehensive survey of architectures, applications, and future directions. *Artif. Intell. Rev.* **2025**, *59*, 11. [CrossRef]
33. Zou, J.; Topol, E.J. The rise of agentic AI teammates in medicine. *Lancet* **2025**, *405*, 457. [CrossRef] [PubMed]
34. Gonzalez, G.R.; Habel, J.; Hunter, G.K. AI agents, agentic AI, and the future of sales. *J. Bus. Res.* **2026**, *202*, 115799. [CrossRef]

35. Collaco, B.G.; Haider, S.A.; Prabha, S.; Gomez-Cabello, C.A.; Genovese, A.; Wood, N.G.; Bagaria, S.P.; Gopala, N.; Tao, C.; Forte, A.J. The role of agentic artificial intelligence in healthcare: A scoping review. *npj Digit. Med.* **2026**, *9*, 345. [CrossRef] [PubMed]
36. Lee, J.; Song, J.; Koo, J.; Choi, S.; Hwang, J.; Saif, S.M.H.; Li, Y.; Li, J.; Yoo, J.; Lee, G.; et al. Agentic Built Environments: A review. *Energy Build.* **2025**, *346*, 116159. [CrossRef]
37. Tennakoon, T.M.M.P.; Chileshe, N.; Rameezdeen, R.; Ochoa, J.J.; Samaraweera, A. Mixed methods framework for implementation research in the architecture, engineering, construction and operations industry. *Int. J. Constr. Manag.* **2026**, *26*, 204–218. [CrossRef]
38. Robert, C.P.; Casella, G.; Casella, G. *Monte Carlo Statistical Methods*; Springer: New York, NY, USA, 2004; Volume 2. [CrossRef]
39. Hyndman, R.J.; Koehler, A.B. Another look at measures of forecast accuracy. *Int. J. Forecast.* **2006**, *22*, 679–688. [CrossRef]
40. Gneiting, T.; Balabdaoui, F.; Raftery, A.E. Probabilistic forecasts, calibration and sharpness. *J. R. Stat. Soc. Ser. B* **2007**, *69*, 243–268. [CrossRef]
41. Pearce, T.; Zaki, M.; Neely, A. Bayesian neural network ensembles. *arXiv* **2018**, arXiv:1811.12188. [CrossRef]
42. Christophides, V.; Efthymiou, V.; Stefanidis, K. *Entity Resolution in the Web of Data*; Morgan & Claypool.: San Rafael, CA, USA, 2015; Volume 5, pp. 1–122. [CrossRef]
43. Montgomery, D.C.; Runger, G.C. *Applied Statistics and Probability for Engineers*; John Wiley & Sons: Hoboken, NJ, USA, 2010; ISBN 0470053046.
44. Jarrett, P.C.; Hill, J.; Howell, M.; Grabow Moore, K.; Thoppil, J.J.; Vargas Ortiz, L.; Parnell, S.; Courtney, D.M.; McDonald, S.; Diercks, D.; et al. Piloting temperature-driven variability in emergency diagnostic accuracy using a leading large language model. *Cureus* **2025**, *17*, e94476. [CrossRef] [PubMed]
45. Cohen, J. *Statistical Power Analysis for the Behavioral Sciences*, 2nd ed.; Lawrence Erlbaum Associates: Hillsdale, NJ, USA, 1988. [CrossRef]