POLITECNICO DI TORINO

MSc in Management Engineering (LM-31)


# Benchmarking Large Language Models for Decision-Making in Supply Chain

**Supervisors**

Giovanni Zenezini

Filippo Maria Ottaviani

**Candidate**

Alberto Bersano

October 2025

## Abstract

The growing diffusion of Large Language Models (LLMs) has stimulated an increasing interest in their application to supply chain management, a field where managerial decisions require precision, efficiency, and adaptability. Despite the widespread use of general-purpose benchmarks such as MMLU or HELM, the literature highlights the absence of systematic evaluation frameworks specifically designed for supply chain contexts. This thesis addresses this gap by developing a set of benchmarks to assess the reliability, efficiency, and managerial usefulness of LLMs. The research is guided by two central questions: (QR1) which combinations of datasets, evaluation metrics, and prompting strategies enable the construction of meaningful benchmarks for supply chain tasks; (QR2) which language model currently offers the best balance among accuracy, speed, and cost. The overall objective is to verify whether LLMs can serve as valid tools to support managerial decision-making. To answer these questions, a multi-layered methodology was designed around a "pyramid of difficulty" dataset, progressing from single-choice questions to numerical problems with exact answers, up to complex tasks requiring explicit reasoning. The benchmarks integrate different prompting strategies (Zero-Shot, Role Prompting, Chain-of-Thought) and evaluate multiple dimensions such as accuracy, cost, latency, token usage, and reasoning quality. The Analytic Hierarchy Process (AHP) was employed to synthesize these metrics into a single comparative index, while acknowledging the subjectivity of the survey-based weights. The experimental analysis of eight state-of-the-art models revealed systematic differences. GPT-5 achieved the highest and most stable accuracy but at significantly higher computational costs and latency. Gemini-2.5 Flash reached similar accuracy while proving more efficient, whereas GPT-5 mini offered a balanced trade-off. By contrast, DeepSeek-v3.1, the Claude series, and Gemini 2.5 Flash-Lite delivered less consistent outcomes, though competitive in speed and lower costs. A key insight concerns prompting. The implicit use of Chain-of-Thought, adding *"Let's think step by step"* without requiring explicit reasoning, did not improve accuracy and sometimes reduced it, especially in complex tasks. In contrast, explicit reasoning (Benchmark 5) produced clear improvements, confirming that transparency in reasoning improves reliability. The comparison of question formats further showed that LLMs perform better with single-choice tasks, where predefined options act as anchors, while struggling with numerical problems that require generating the correct

value independently. In general, the thesis demonstrates that LLMs can support managerial decision-making in supply chain contexts, provided that their adoption is guided by structured benchmarking capable of balancing accuracy with efficiency. The work contributes theoretically by proposing a replicable, domain-specific evaluation framework and by introducing a qualitative method for analyzing reasoning errors, distinguishing between interpretation and planning failures. On the practical side, it offers guidelines for formulating queries to optimize reasoning and reduce errors, as well as for selecting models by balancing accuracy, cost, and latency. Future research should extend the framework to other domains, type of questions, and interactive benchmarks, and assess robustness in dynamic and uncertain supply chain environments.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Problem Existence

Modern projects are becoming more complicated, necessitating the integration of multiple tools and methodologies by managers to successfully address difficulties. This increasing complexity, paired with stringent time and financial limits, necessitates the exploration of novel solutions that can aid decision-making and enhance efficiency. In light of this, Large Language Models (LLMs) have recently gained traction in both academic research and managerial practice. Companies and organizations are experimenting with their usage in data analysis, forecasting, knowledge management, and automated decision-making processes. LLMs are attractive in supply chain management, particularly due to their potential to improve efficiency, predictive accuracy, and adaptability to changing circumstances. However, their actual performance in managerial settings remains to be fully established. Although LLMs promise quick, scalable, and flexible support, questions remain regarding their accuracy, robustness, and reliability in intricate, real-world decision-making.

For managers, this raises a concrete dilemma: whether to implement these technologies without strong proof of where they deliver real added value, risking wasting both expense and time. For academics, the lack of defined and reproducible benchmarks adapted to supply chain contexts makes it impossible to objectively judge LLMs. These limitations underscore the need for research that can bridge empirical experimentation and methodological rigor.

## 1.2 Problem Importance

This topic has profound implications for both practitioners and researchers. For managers and companies, adopting Large Language Models without detailed evaluation may lead to wasteful investments, planning errors, cost overruns, and, ultimately, lost competitiveness. In dynamic and global supply chain scenarios, the ability to make accurate and fast decisions frequently determines success or failure. When an LLM makes errors in projections, simulations, or trade-off calculations, the economic and operational consequences can be severe.

For academics and researchers, developing methodologically robust benchmarks is critical to advancing comprehension of the real strengths and weaknesses of these models in applied settings. Such benchmarks allow to develop hypotheses about when generative technologies add value, establish which evaluation metrics are genuinely significant, and understand how prompting techniques affect performance.

In conclusion, solving this issue is essential to preventing expensive managerial errors and establishing a strong scientific basis for the ethical and successful application of large language models in Supply Chain Management.

## 1.3 Old and State-of-the-Art Literature Recap

From the early conception of Turing's test of machine intelligence (Russell & Norvig, 2016), artificial intelligence has developed into a vast field that currently encompasses generative AI, deep learning, and machine learning (Pahuja et al., 2025; Banh & Strobel, 2023).

Supervised, unsupervised, and reinforcement learning established the foundation for Neural Networks (Goodfellow, Bengio, et al., 2017), facilitating Deep Learning (Dol & Geetha, 2021) and the advent of Generative AI capable of generating realistic material (Banh & Strobel, 2023).

The Transformer architecture signified a pivotal transformation in natural language processing (Bengesi et al., 2023), resulting in the development of Large Language Models (LLMs) such as *BERT* and *GPT* (Haleem et al., 2022; Bengesi et al., 2023), subsequently refined through reinforcement learning from human feedback (RLHF) (Christiano et al., 2017), in-context learning (Brown et al., 2020), and prompt engineering (Clavié et al., 2023; White et al., 2023). Prompting techniques, including zero-shot (Wei et al., 2022), few-shot (Brown et al., 2020), chain-of-thought (Sivarajkumar et al., 2024), and ReAct (Yao, Zhao, et al., 2022), have enhanced reasoning powers; nonetheless, limitations such as bias (Ferrara, 2023; Schramowski

et al., 2022), hallucinations(Ji et al., 2023; Susarla et al., 2023), and lack of transparency (Janiesch et al., 2021; Meske et al., 2022) persist.

These advancements have expanded applications in IT (Kshetri et al., 2024), healthcare(S. Liu et al., 2023; Savage, 2023), marketing (Brand et al., 2023), and management sectors, including supply chain (Jackson et al., 2024) and project management (Prieto et al., 2023).

As Large Language Models' capabilities and usage rose, so did the need for systematic evaluation. Benchmarks give objective and repeatable assessments of performance, identifying strengths, weaknesses, and hazards across tasks and domains (Chang et al., 2024).

Benchmarks ranged from general-purpose datasets like *MMLU* (Hendrycks et al., 2021), *AGIEval* (Zhong et al., 2023), and *HELM* (Liang et al., 2023) to reasoning-focused tasks like *HotpotQA* (Yang et al., 2018), *2WikiMultiHopQA* (Ho et al., 2020), and *FanOutQA* (Zhu et al., 2024), as well as domain-focused frameworks like *EconLogicQA* (Quan & Z. Liu, 2024), *FinEval* (Guo et al., 2025).

Other contributions dealt with organizational contexts, with benchmarks for inventory management (Z. Li et al., 2024) and business process management (Busch & Leopold, 2024), and conversational quality, with *LLM-EVAL* (Lin & Y.-N. Chen, 2023) rating open-domain dialogues across several dimensions.

Despite these gains, challenges remain, ranging from prompt sensitivity (Ferrara, 2023) and benchmark gaming (Balloccu et al., 2024) to linguistic narrowness (Mushtaq et al., 2025) and a lack of standardized documentation (McIntosh et al., 2024).

Recent research reveals both significant gains and ongoing limitations in Large Language Models. Frontier models have outstanding capabilities, but they still struggle with complicated reasoning, domain transfer, and extended context management (Guo et al., 2025; Lunardi et al., 2025).

While advancements like Multi-Agent reasoning (P. Chen et al., 2024) and Chain-of-Thought prompting (Wei et al., 2022) help to minimize some of the shortcomings, they are still limited by scale and design.

These findings underscore the significance of transitioning to transparent, context-aware evaluation frameworks that are matched with real-world managerial targets, ensuring that improvements in LLM performance transfer into actual benefit in domains such as Project and Supply Chain Management.

## 1.4 Gap

Despite the rapid evolution of Generative Artificial Intelligence and the growing adoption of Large Language Models in organizational contexts such as Supply Chain Management and Project Management, the existing literature still lacks systematic benchmarks tailored to these domains.

This gap is an important obstacle because without domain-specific benchmarks, it is not possible to accurately determine whether Large Language Models can support decision-making processes, enhance forecasting reliability, or contribute to the reduction of project delays and cost overruns in supply chains.

Moreover, the absence of structured evaluation frameworks constrains both theoretical progress in learning how these models work in managerial contexts and practical recommendations for firms considering their applications.

Therefore, further investigation is needed to develop and apply dedicated benchmarks that capture the distinctive needs of Supply Chain and Project Management, allowing robust comparisons among models and supporting their effective and responsible incorporation into business operations and practices.

## 1.5 Objective(s)

This thesis aims to determine whether Large Language Models can be considered reliable tools for facilitating managerial decision-making in Supply Chain Management.

The objective is to analyze findings not solely based on technical performance, but also by assessing factors that represent the real operational needs of managers, to see whether these models can offer solid support in both strategic and routine decision-making.

To achieve this goal, the study develops and implements objective benchmarks for evaluating Large Language Models in this domain. Every design decision in the creation of the benchmarks is explicitly justified to ensure transparency and replicability, allowing the results to be independently reproduced. The investigation is guided by two distinct research questions:

- RQ1: Which combinations of datasets, evaluation metrics, and prompting techniques enable the development of valuable benchmarks to assess LLM performance in Supply Chain settings?

- RQ2: Which Large Language Model currently exhibits the best overall performance, offering a comparative framework to guide managerial choices?

Finally, the study closes methodological gaps and offers researchers and managers a trustworthy reference point for assessing the function of Large Language Models in the Supply Chain domain.

## 1.6 Structure

This thesis is organized as follows. Chapter 1 introduces the research problem, discussing its existence and importance, providing a brief recap of the previous state of the art, highlighting existing gaps, and presenting the objectives of the study. Chapter 2 presents a comprehensive literature review, covering the emergence of Generative AI and Large Language Models (LLMs), as well as the current approaches and challenges in benchmarking these models. Chapter 3 details the research methodology, including the formulation of research questions and the exploratory framework, the construction of domain-specific benchmarks, covering datasets, evaluation metrics, prompting techniques, and the creation of final benchmarks, and the implementation and testing of benchmarks, including the selection of LLMs and statistical significance testing. Chapter 4 reports the results, presenting both benchmark-level performance and cross-benchmark comparisons, and also includes the findings from a survey conducted. Chapter 5 discusses the results, highlighting their theoretical and practical implications/the contribution to the theory and practice. Lastly, Chapter 6 summarizes the study, outlines its (de)limitations, and suggests possible streams of future research.

# Chapter 2

# Literature Review

## 2.1 Introduction to Generative AI

This section outlines the evolution of Artificial Intelligence, tracing its development up to the emergence of contemporary Generative AI. The concept of Large Language Models will be introduced, with particular attention to prompting techniques and the current limitations. Finally, the section will conclude with an examination of Generative AI and Large Language Model applications in organizational contexts, with a specific focus on *Supply Chain* and *Project Management*.

### 2.1.1 From Artificial Intelligence to Generative AI

The term *Artificial Intelligence* (*AI*) refers to a machine's ability to perform tasks that would typically require human cognitive abilities (Gignac & Szodorai, 2024), including language comprehension, complex pattern recognition, experiential learning, and autonomous decision-making (Banh & Strobel, 2023; Winston, 1993). In 1950, Alan Turing introduced a test to determine whether a machine is capable of exhibiting intelligent behavior (Russell & Norvig, 2016). According to his operational criterion, a machine is deemed intelligent if, when interacting through written language, it is able to convince a human interlocutor that they are conversing with another human being (Jiang et al., 2022).

Over time, Artificial Intelligence has evolved and is now an umbrella term encompassing various subfields and methodologies (Pahuja et al., 2025; Banh & Strobel, 2023) (Figure 2.1).

*Machine Learning* (*ML*) is a part of this model, and it is now widely recognized as one of the foundational pillars of modern Artificial Intelligence (Lv, 2023). ML focuses on the

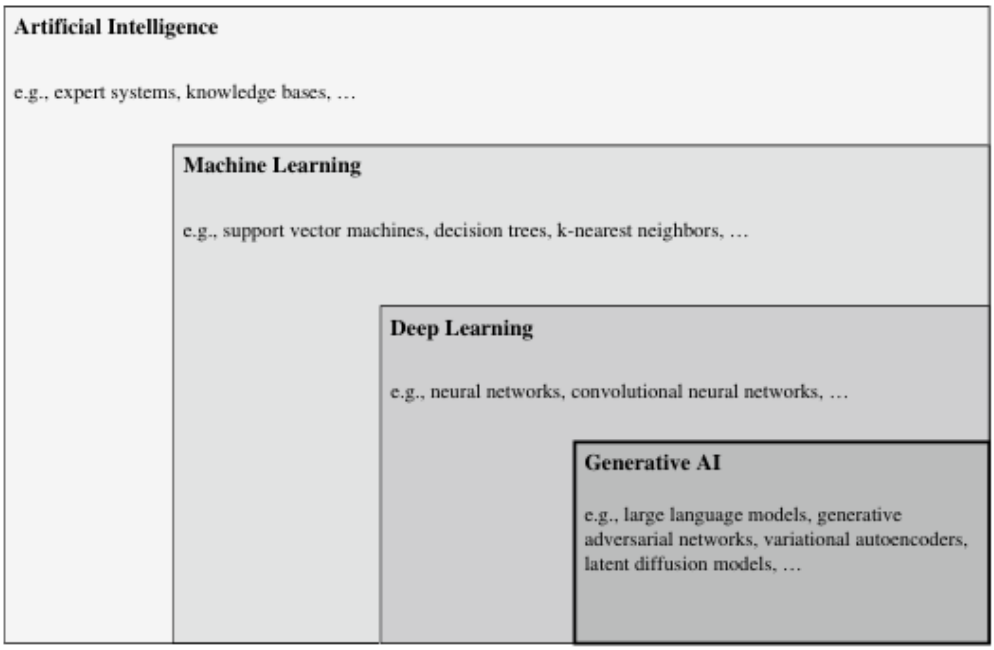



Figure 2.1: Generative AI and other AI concepts (Banh & Strobel, 2023).

development of algorithms that can identify patterns in data and improve their performance over time, without the need for explicit reprogramming for each new task (Brynjolfsson & Mitchell, 2017; Dol & Geetha, 2021). This ability to generalize from experience allows AI systems to adapt to dynamic environments and tackle complex, data-driven problems across various domains (Lv, 2023).

Machine Learning methods can be categorized into different types, depending on the nature of the training data and the specific objectives of the algorithm (Mohri et al., 2012). The most important method in ML is *Supervised Learning*. In this approach the algorithm is trained on a labeled dataset: each input instance is associated with a corresponding output label (Cunningham et al., 2008). The model learns the mapping between inputs and outputs and is then able to make predictions on new unseen test data. This is the most frequently applied approach in tasks such as classification, regression, and ranking (Mohri et al., 2012). Unfortunately, Supervised Learning by definition relies on a human supervisor to provide an output example for each input example. Due to this, many researchers have shifted their focus toward studying *Unsupervised Learning* (Goodfellow, Pouget-Abadie, et al., 2020). In this case the data provided to the model are unlabeled. The algorithm relies on its internal mechanisms to autonomously identify patterns or correlations within the data (Dol & Geetha, 2021). This type of learning is often used for tasks such as clustering (Tyagi et al., 2022) but it's also used in generative

modelling (Goodfellow, Pouget-Abadie, et al., 2020). Lastly, in *Reinforcement Learning*, the algorithm, referred to as an agent, interacts with an environment and learns through a system of rewards and penalties, following a trial-and-error process (Pahuja et al., 2025; Mohri et al., 2012).

These learning paradigms provide the foundation for the implementation of *Artificial Neural Networks*, computational models inspired by the structure of the human brain (Goodfellow, Bengio, et al., 2017). When these models consist of multiple hidden layers, the approach is referred to as *Deep Learning* (*DL*) (Figure 2.2). Deep Learning is a subfield of Artificial Intelligence that enables systems to learn and classify objects by interpreting data in a manner inspired by the human brain. It is particularly effective in making predictions and informed decisions based on current data (Dol & Geetha, 2021).

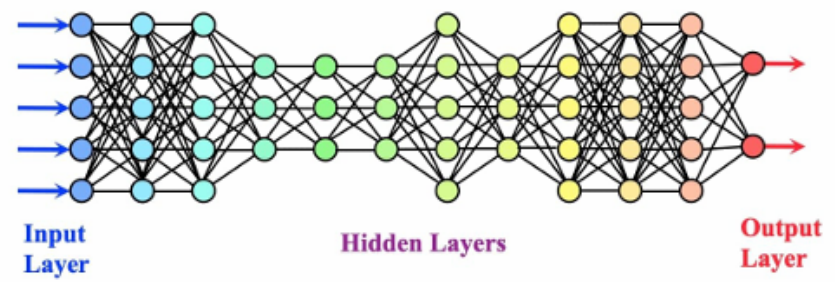



Figure 2.2: Example of Deep Neural Network (Horzyk et al., 2023).

The evolution of Deep Neural Networks, supported by advances in computational power and the availability of large-scale datasets, has enabled the development of increasingly sophisticated models (Lecun et al., 2015). In recent years, this progress has paved the way for the rise of *Generative Artificial Intelligence* (*GenAI*), which represents a fundamental shift from a purely predictive and discriminative paradigm toward a generative one (Banh & Strobel, 2023). In this new paradigm, the objective is not merely to analyze or classify data, but to autonomously and realistically generate novel content.

### 2.1.2 Generative AI and LLMs

Over the years, the shift of scientific interest from discriminative to generative models has fostered the development of numerous architectures that have transformed fields such as natural language processing and the generation of images and videos (e.g., *VAE, GAN, diffusion models* and *Transformer*) (Bengesi et al., 2023; Pahuja et al., 2025). The *Transformer* architecture, in

particular, signaled a significant change in the field of Generative Artificial Intelligence.

Transformers, which were first presented by a group of Google researchers under the direction of Vaswani in the 2017 paper *"Attention Is All You Need"* (Vaswani et al., 2017), have revolutionized the state-of-the-art in a variety of tasks, particularly in *Natural Language Processing (NLP)* (Bengesi et al., 2023). The innovation of the Transformer lies in its *attention* and *self-attention* mechanisms (Shen et al., 2023), which enable it to evaluate the importance of various input sequence elements, such as words in a sentence or pixels in an image, in a similar way to how people concentrate on particular words when attempting to comprehend a sentence (Bengesi et al., 2023; P. Chen et al., 2024).

The success of Transformers became evident with the introduction of models such as *BERT (Bidirectional Encoder Representations from Transformers)* (Devlin et al., 2019) developed by researchers at Google, and *GPT (Generative Pre-trained Transformer)* by OpenAI (Haleem et al., 2022; Bengesi et al., 2023). These models, more generally, belong to the family of *Large Language Models (LLMs)*, which refers to large pre-trained transformer models that are typically trained for prediction tasks, where the objective is to predict the next word given some textual input (Pahuja et al., 2025; Chang et al., 2024).

Beyond the self-attention mechanism, the progressive evolution of Large Language Models has been accompanied by the introduction of several key innovations that have significantly enhanced their capabilities. Among these, *Reinforcement Learning from Human Feedback (RLHF)* has played a particularly important role. By incorporating human judgments into the *fine-tuning* process, this approach allows guiding the model's behavior more precisely, helping to align its outputs with human preferences and expectations (Christiano et al., 2017).

Equally relevant is the development of *in-context learning*, a capability that allows LLMs to perform complex tasks without the need for additional training. Instead, the model learns to interpret and respond appropriately to the information provided within the prompt itself, demonstrating an impressive ability to generalize across tasks simply by leveraging contextual cues (Brown et al., 2020).

Finally, the emergence of *prompt engineering* has transformed the way users interact with these systems. Rather than writing code, users can now shape model behavior through carefully crafted natural language inputs. In this sense, prompt engineering represents a new kind of programming, one that is accessible and intuitive, yet capable of eliciting highly sophisticated outputs from the model (Clavié et al., 2023; White et al., 2023).

### 2.1.3 LLMs Prompt Techniques

In the following, some prompting techniques taken from the literature will be discussed.

#### Zero-Shot

The *Zero-Shot* prompt is the simplest type of prompt (Wei et al., 2022). It consists of providing the model with only a textual description of the task to be performed, without including explicit input-output examples (Sivarajkumar et al., 2024). In this approach, the LLM relies only on its pre-trained knowledge to interpret and complete the task (Reynolds & McDonell, 2021). Some researchers (Reynolds & McDonell, 2021) have shown that well-designed zero-shot prompts can achieve strong performance, sometimes even outperforming Few-Shot prompts (Sivarajkumar et al., 2024). However, in tasks such as comprehension of the language, answering questions, and inference of natural language, Few-Shot prompting generally leads to better performance (Wei et al., 2022).

#### One-Shot and Few-Shot

When designing prompts for LLM models, it can be advantageous to incorporate clear examples within the input provided (Reynolds & McDonell, 2021). In a *One-Shot* prompt, the model is given a single illustrative example of the task, followed by a new instance to solve. The *Few-Shot* prompt, by contrast, involves presenting the model with several examples, typically ranging from two to five or more, before the test prompt (Brown et al., 2020). These examples help to establish context and are particularly valuable in handling more complex tasks (Sivarajkumar et al., 2024). They are especially effective when aiming to guide the model toward a particular format or structure in its output. In fact, research shows that providing examples that closely align with the nature of the target task improves the performance of the model (Y. Li, 2023).

#### Role Prompting

This technique involves explicitly assigning a role to the model, instructing it to act as, for example, a professor, an expert, or a student (Kong et al., 2023). The role context helps the model adjust the tone, style, and level of expertise in its responses. Assigning a functional identity to the LLM is an effective way to guide the model's behavior toward answers that are

more relevant and consistent with the intended communicative goal (Zhao et al., 2025).

### Chain-of-Thought (CoT)

The *Chain-of-Thought (CoT)* technique is based on explicitly prompting the model to break down a problem into successive logical steps, thereby simulating a step-by-step reasoning process. It is particularly useful for tasks that require deduction, calculations, or multi-step problem solving (Sivarajkumar et al., 2024). Making the reasoning chain explicit not only enhances the transparency of the reasoning process but also allows for the diagnosis of potential intermediate errors. The CoT technique has often been associated with Few-Shot prompting (Wei et al., 2022) and, more recently, with Zero-Shot prompting (Kojima et al., 2022). In the *Zero-Shot CoT* the prompt is augmented with a simple instruction such as *"Let's think step by step"*, without providing specific examples. This minimal modification has proven surprisingly effective in improving model performance in the absence of additional data (Y. Li, 2023). (Figure 2.3)

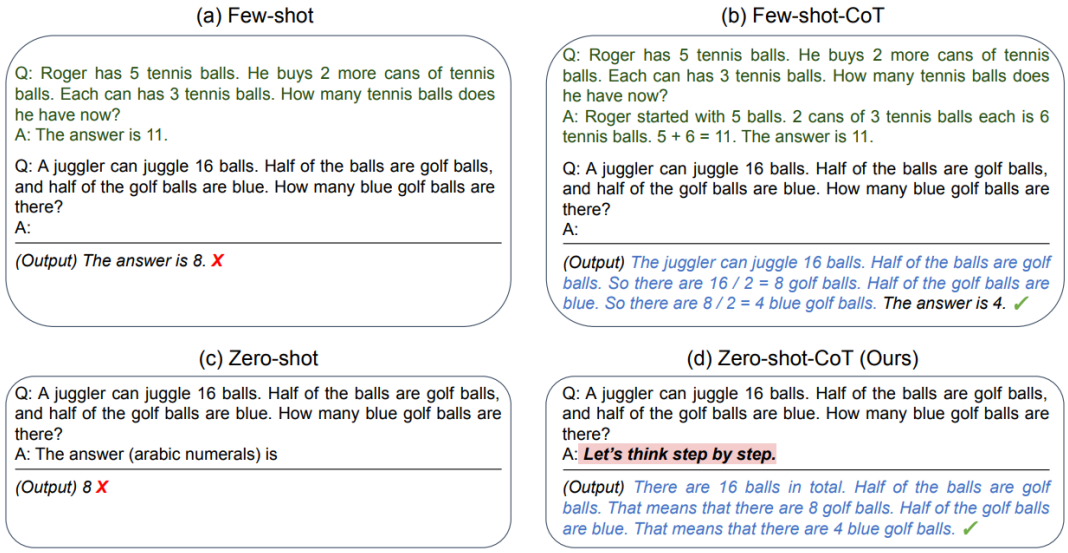



Figure 2.3: Positive impact of the CoT prompting technique in Zero-Shot and Few-Shot cases (Kojima et al., 2022).

### Self-Consistency

This strategy addresses the variability in the outputs generated by LLMs through a process of multiple sampling. The model is executed several times with the same prompt, producing different reasoning paths. Among the various responses obtained, the most frequent or most

consistent is selected. This mechanism leverages the principle that the correct reasoning paths tend to converge towards the same solution, whereas the incorrect ones produce more scattered outcomes (X. Wang et al., 2022). Based on the intuition that complex tasks can be solved through multiple reasoning pathways leading to a correct outcome (Stanovich & West, 2000), this technique is frequently combined with Chain-of-Thought prompting to address complex problems.

### Tree-of-Thoughts (ToT)

Going beyond the linearity of the Chain-of-Thought, the *Tree-of-Thoughts* technique enables the model to explore multiple reasoning branches simultaneously. Each *"thought"* is treated as a node within a logical tree, from which new trajectories may emerge. This approach is particularly well suited to solving complex, open-ended problems, where the deliberate exploration of alternatives enhances the quality of the final decision. (Yao, Yu, et al., 2023) (Figure 2.4)

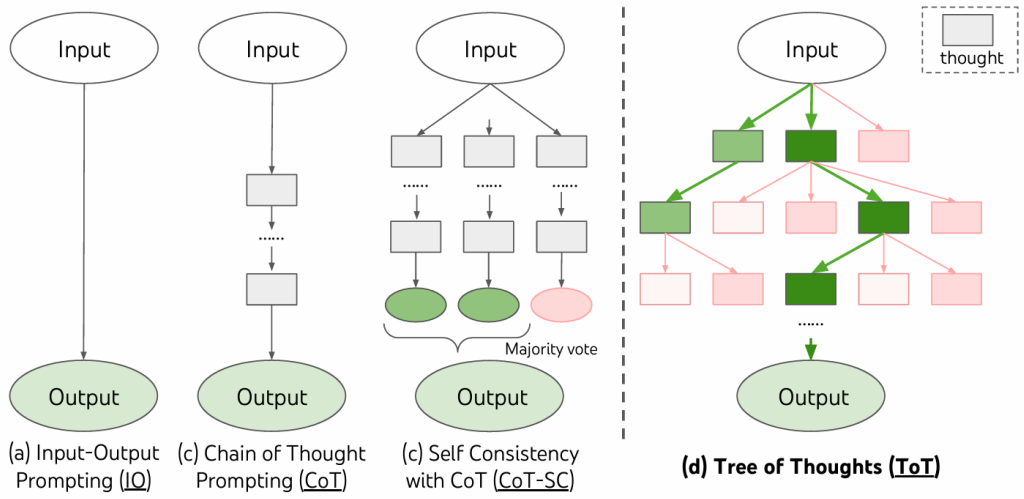



Figure 2.4: Comparison of various approaches to problem solving with LLMs (Yao, Yu, et al., 2023).

### Reason e Act (ReAct)

The *ReAct* technique combines linguistic reasoning with the execution of actions. In this framework, the model alternates between phases of reasoning and operational phases (acting), such as consulting external sources or interacting with digital tools. This paradigm, which mirrors human behavior in problem solving, is one of the foundational components of recent LLM-based

agents, enabling them to interact dynamically with their environment to complete complex tasks. (Yao, Zhao, et al., 2022)

### 2.1.4 Limitations

Although Large Language Models have reached significant milestones in recent years, they still present limitations that compromise the overall quality of their outputs. The most critical issues include *bias*, the risk of *hallucinations*, and the *lack of transparency and explainability* in their decision-making processes.

#### Bias

The performance of Generative AI systems is strongly influenced by the quality of the training data. As highlighted in the literature, GenAI models are prone to bias causing biased decisions, disadvantages, and discriminations (Ferrara, 2023; Schramowski et al., 2022). Such biases may emerge during the training phase, due to datasets that are non-representative, imbalanced, or incorrectly labeled, but can also appear during inference, when algorithmic choices such as overfitting introduce distortions not present in the original data. These dynamics make it challenging to ensure fairness and reliability in different applications (Banh & Strobel, 2023).

#### Hallucinations

A recurring limitation of LLMs is their tendency to produce hallucinations, namely outputs that are coherent and convincing but factually incorrect. *'Hallucinations [...] manifest themselves in confidently generated results that seem plausible but are unreasonable with respect to the source of information'* (Ji et al., 2023; Susarla et al., 2023). This phenomenon is mainly related to the probabilistic nature of generative models and to the use of training data containing contradictory or unreliable information (Dziri et al., 2022). The result is text that may deviate from reality, thus reducing user trust in the reliability of the system (Banh & Strobel, 2023; Pahuja et al., 2025).

#### Lack of Trasparency and Explainability

A further challenge is represented by the opacity of these systems. ML models function as black boxes (Janiesch et al., 2021; Meske et al., 2022), since it is rarely possible to trace how a given

output was produced. This absence of interpretability prevents users from fully validating or understanding model behavior, which is particularly critical in areas where accountability and decision traceability are required (Banh & Strobel, 2023).

### 2.1.5 Applications in Supply Chain and Project Management

Over the past few years, Generative Artificial Intelligence, and especially Large Language Models, has significantly reshaped the way organizations work. The ability of these technologies to combine analytical capabilities, predictive modeling, and creativity enables the automation of repetitive tasks, the improvement of output quality, and the reduction of execution times (Pahuja et al., 2025; Banh & Strobel, 2023).

From a business and industry perspective, applications cover a wide range of use cases. In the software and IT sector, tools such as *GitHub Copilot*, powered by OpenAI Codex, help developers write code, reducing completion times by up to 56% (Pahuja et al., 2025). In digital services, *Microsoft Bing* integrates ChatGPT to provide contextual responses in web searches, while in the marketing domain, GenAI is used for the generation of personalized content and offerings and the optimization of the sales lead generation process (Kshetri et al., 2024). In the financial sector, applications range from automated analysis of financial statements and transactions to the generation of forecasts for the stock and currency markets (George et al., 2023). LLM and GenAI also play an important role in the healthcare sector, supporting medical imaging diagnostics, the discovery of new drugs, and patient communication, thus contributing to the reduction of development times for therapies and clinical protocols (S. Liu et al., 2023; Savage, 2023).

In addition to these cross-sector applications, GenAI and LLMs are increasingly being applied in domains with high managerial complexity, such as Supply Chain Management and Project Management.

#### Supply Chain Management

In the field of Supply Chain and Operations Management (SCOM), Generative AI is demonstrating transformative potential in multiple decision-making areas. According to the framework proposed by Jackson et al. (2024), the capabilities of *learning*, *perception*, *prediction*, *interaction*, *adaptation*, *reasoning*, and *creativity* offered by GenAI can be applied in at least thirteen strategic domains, including *demand forecasting, inventory management, supply chain*

*design, production planning and control, quality management*, and *supply chain risk management*. (Figure 2.5)

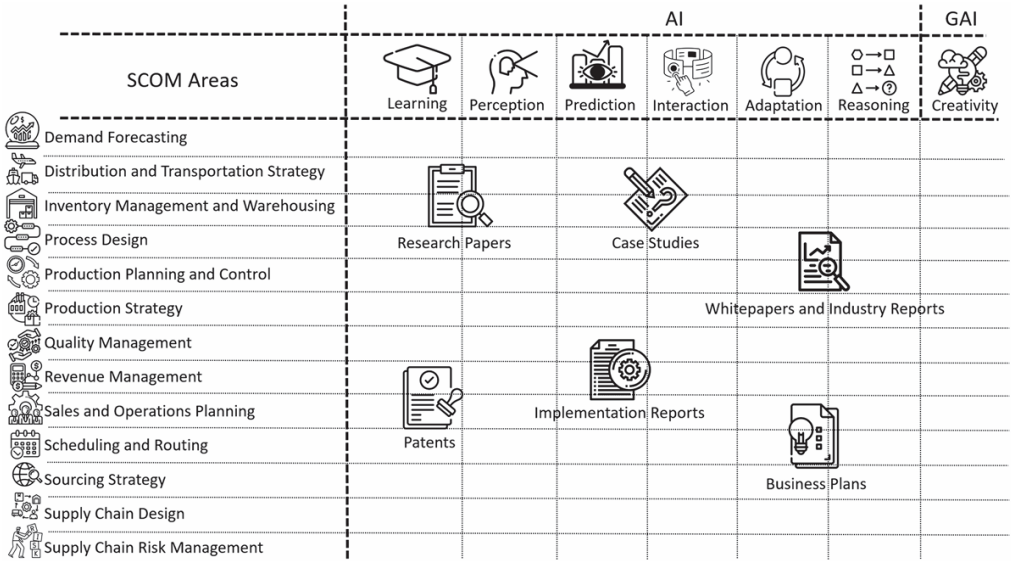



Figure 2.5: SCOM areas (Jackson et al., 2024).

In this regard, Skórnóg & Kmiecik (2023) demonstrate how ChatGPT can be employed for material forecasting in the manufacturing sector, in some cases achieving more accurate results than commonly used models for demand forecasting in business operations, such as *ARIMA*.

Other examples concern organizations that have integrated GenAI into their systems. *Walmart* has adopted Pactum AI, a generative chatbot-based system, to automate supplier negotiations (Hoek et al., 2022); *Maersk* has implemented GenAI to optimize logistics planning and improve resilience (Handley, 2023); while *DHL* is experimenting with ChatGPT to automate communications and warehouse operations (Moller, 2023). *Instacart* employs a conversational assistant powered by OpenAI to facilitate orders and personalize recommendations (Zhuang, 2023), and *Amazon Business* leverages AI models to analyze purchasing data and suggest more cost-effective alternatives.

Finally, another emerging development concerns the integration of LLM with optimization systems, as exemplified by *Microsoft's OptiGuide* framework (B. Li et al., 2023), which translates requests in natural language (e.g., *'What happens if I use supplier B instead of A?'*) into queries for mathematical solvers, returning intelligible results and intuitive visualizations. This approach facilitates communication between planners and complex systems, enhancing decision-making transparency and reducing response times (B. Li et al., 2023).

**Project Management**

In Project Management, although LLMs such as ChatGPT are not yet ready to replace professional software, studies such as that by Prieto et al. (2023) highlight their usefulness in generating coherent project plans and rapidly adjusting operational sequences in response to changing requirements. A notable example is the LLM-Project initiative (Zhen et al., 2024), in which LLMs, trained on Standard Operating Procedures (SOPs) and simulated data, were able to produce *Work Breakdown Structures (WBS)* complete with time coding (*Finish–Start*, *Start–Start* relationships) and resource allocation.

Further insights are provided by the study of Cinkusz et al. (2024), which introduces *CogniSim*, a framework that integrates *cognitive agents* powered by Large Language Models within the *Scaled Agile Framework (SAFe)* to strengthen software project management. Simulations revealed measurable improvements in various metrics, including task completion times, quality of deliverables, and communication coherence.

Looking ahead, the strategic adoption of GenAI and LLMs in Supply Chain and Project Management goes beyond improving operational efficiency: it paves the way for more resilient, transparent, and adaptive supply chains, where human–machine collaboration becomes a key driver of competitiveness.

Such widespread applicability, however, calls for a careful assessment of its ethical, security, and labor-related implications. While GenAI can boost productivity and create new professional roles (e.g., prompt engineers), it also introduces risks associated with data quality, the protection of sensitive information, and the potential replacement of low-skilled jobs (Einola & Khoreva, 2023).

## 2.2 Introduction to Benchmarking

This section covers the evolution of benchmarking from its origins to its widespread adoption across fields such as computing, finance, and management. It outlines the key lifecycle stages and discusses the fundamental principles that underpin the design of high-quality benchmarks.

### 2.2.1 Benchmark Definition

The term *benchmark* comes from measurement science, where it originally referred to a physical mark used as a reference point for leveling operations in geodesy (Zairi & Leonardo, 1996).

Over time, this concept evolved into a broader idea of a standardized reference for performance comparison, and has since been adopted across several disciplines, including computing, finance, and management (Zhan, 2022). In computer science, the first formal benchmarks were introduced in the early 1960s by the *Auerbach Corporation* to measure system speed through predefined routines. A primary limitation of these initial benchmarks was that their findings were not acquired through direct execution on the systems under examination, but instead derived from performance metrics published by vendors, so diminishing their impartiality and comparability (Lewis & Crews, 1985; Zhan, 2022).

The initial step in this endeavor was workload modeling, which entailed choosing a representative subset of programs from the diverse array of tasks commonly performed by users. The concept was that, by concentrating on a meticulously selected sample, one might emulate the overall behavior of real workloads while maintaining a manageable review process. Then, to compare performance on real-world jobs, researchers suggested application benchmarks, which are real programs running on different systems. Although they were more representative than abstract metrics, they were expensive and difficult to apply across diverse architectures (Lewis & Crews, 1985). The idea of synthetic benchmarks was developed in order to overcome these restrictions. Instead of running complete apps, synthetic benchmarks created smaller programs that mimicked the key functions of actual applications. These benchmarks allowed for more realistic and economical system comparisons by removing specifics while maintaining performance-critical features (Y. Liu, Khandagale, et al., 2021).

Together, these approaches established the foundation for performance evaluation in computing and continue to influence modern benchmark design. In parallel, the concept of benchmarking took root in the management sector. *Xerox Corporation* pioneered competitive benchmarking in the late 1970s, systematically analyzing the products, processes, and organizational practices of competitors to identify and adopt superior methods (Zairi & Leonardo, 1996). Over time, this evolved into a broader quality improvement strategy based on comparing internal operations with industry best practices. Across disciplines, the benchmark has emerged as an important scientific and engineering tool: it defines quantifiable objectives, establishes standard conditions, and allows for consistent performance comparison (Zairi & Leonardo, 1996).

### 2.2.2 Benchmark Design

According to the literature, a high-quality benchmark must go through four critical lifecycle stages: *design, implementation, documentation, and maintenance* (Figure 2.6).

Insights from domains such as transistor hardware, environmental science, and bioinformatics identify four fundamental characteristics of good benchmarks (Reuel et al., 2024).

- First, tasks should be planned for downstream *utility*, reflecting real-world conditions and use cases.
- Second, to ensure *validity*, benchmarks should use large test sets, avoid bias from gold standards, and be periodically updated to prevent overfitting (Y. L. Liu et al., 2024; Reuel et al., 2024).
- Third, *score interpretability* requires benchmarks to clearly define their purpose, scope, and procedures, avoiding misleading or absolute statements.
- Finally, *accessibility* promotes reproducibility through open data and code (Bartz-Beielstein et al., 2020; Reuel et al., 2024).

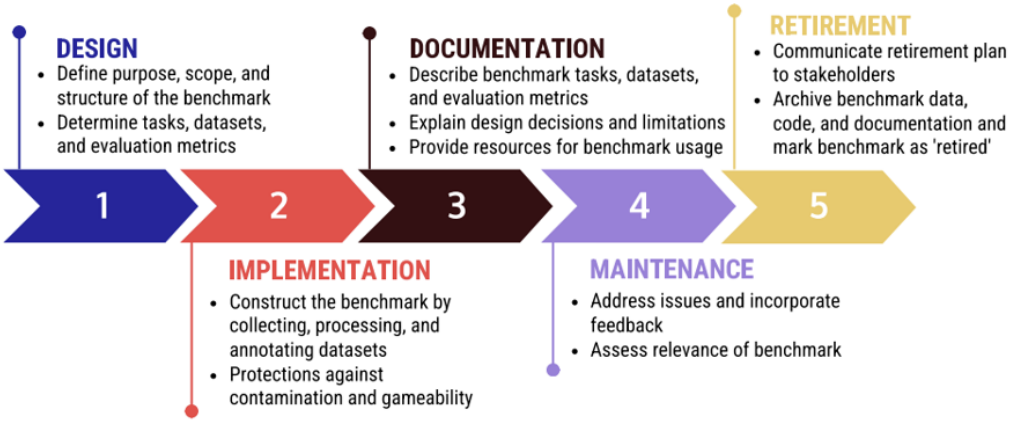



Figure 2.6: Five stages of the benchmark lifecycle (Reuel et al., 2024).

## 2.3 Benchmarking Large Language Models (LLMs)

As Large Language Models (LLMs) are increasingly utilized in fields such as education, healthcare, marketing, and finance, apprehensions about their reliability and influence have escalated reinforcing the need for rigorous and systematic evaluation recognized by both academic and

industrial stakeholders (Chang et al., 2024). Benchmarks fulfill this purpose by giving objective and repeatable performance measurements, allowing for the identification of strengths, shortcomings, and potential risks. Effective evaluation goes beyond simply assessing accuracy; it provides insights for optimizing human–AI interaction workflows, establishes safeguards for deployment in domains such as healthcare, and verifies system robustness in specialized tasks where errors may have high costs. For example, in market research, LLMs might complement traditionally high-priced methods such as conjoint studies, which assess how consumers value different product attributes through trade-off analysis, or focus groups, allowing for rapid, cost-effective, and iterative testing of marketing or pricing strategies prior to product launch (Brand et al., 2023). As models develop in size and ability, benchmarks must evolve to include not only task-specific skills, but also resilience, trustworthiness, and domain relevance (Busch & Leopold, 2024).

This chapter investigates how the evaluation of LLMs has been addressed in the literature. First, it covers the main types of tasks and datasets that are commonly used for benchmarking. Second, it examines the evaluation metrics employed to measure model performance. Third, it summarizes the findings from existing benchmark studies. Finally, it analyzes the limitations and open challenges discovered in several contributions.

### 2.3.1 Task Types and Datasets

#### Task Types

Benchmarks in Natural Language Processing have traditionally focused on generic and constrained tasks (Busch & Leopold, 2024). These include question answering, where models are asked to provide accurate answers based on a given passage or dataset; sentiment analysis, which assesses the ability to identify the emotional tone of a text, such as positive or negative reviews (Kumar et al., 2023); and natural language inference, which evaluates whether a model can determine whether one sentence logically follows from another (Miralles-González et al., 2025).

While these standardized exercises have helped to assess development, they do not fully capture the complexities of how LLMs are used in everyday or domain-specific contexts (Miller & Tang, 2025).This gap, evident when models score highly on benchmark datasets but underperform in real-world applications requiring contextual adaptation (Kiela et al., 2021), has

driven the creation of more sophisticated benchmarks designed to stress-test reasoning, interaction, and applied knowledge.

The most popular general-purpose benchmarks are *MMLU* (Hendrycks et al., 2021), which uses almost exclusively multiple-choice questions to assess knowledge in 57 academic and professional subjects; *AGIEval* (Zhong et al., 2023), which draws on standardized exams and employs multiple-choice and fill-in-the-blank formats; and *HELM* (Liang et al., 2023), which utilizes a combination of multiple-choice, short-answer, and free-text tasks to provide a more comprehensive assessment. These formats were specifically intended to minimize subjectivity and guarantee reproducible scoring, with multiple-choice and close questions providing unambiguous correctness standards, while free-text tasks add more open-ended evaluation to capture broader model abilities.

Other datasets explore more difficult goals beyond these all-purpose benchmarks, expanding on preexisting frameworks. *HotpotQA* (Yang et al., 2018) and *2WikiMultiHopQA* (Ho et al., 2020), for instance, examine whether models can respond to queries that call for integrating fragments of data from several sources rather than depending solely on a single finding. By constructing reasoning paths that are longer and less linear, *FanOutQA* (Zhu et al., 2024) makes this process even more difficult, requiring models to pass through an average of seven intermediate steps before arriving at the right answer.

Moreover, new benchmarks have been developed to assess performance in specific domains. Using multiple-choice questions to capture consistency in economic logic, *EconLogicQA* (Quan & Z. Liu, 2024) assesses sequential thinking in economics by asking models to predict and order interconnected economic events across numerous situations.The finance sector is the topic of FinEval (Guo et al., 2025), which assesses LLMs' proficiency in handling domain-specific knowledge and reasoning tasks using both multiple-choice and real-world case-based scenarios that mimic financial decision-making.

In addition to domain-specific reasoning, conversational quality has been an important area of evaluation. *LLM-EVAL* (Lin & Y.-N. Chen, 2023) offers a unified multi-dimensional framework for analyzing open-domain conversations and automatically assigns scores for appropriateness, grammar, relevance, and content quality.

Furthermore, organizational contexts have been covered in recent contributions. The multi-agent framework for inventory management by Z. Li et al. (2024) and the BPM benchmark by Busch & Leopold (2024) are two examples that extend evaluation toward fields directly re-

lated to Supply Chain and Project Management. However, systematic benchmarks specifically designed for Supply Chain and Project Management remain to be developed.

### Datasets

In the literature, benchmark datasets are built using various methodologies based on the skills to be evaluated.

Some benchmarks are based on real examinations, such as *AGIEval* (Zhong et al., 2023), which gathers items from standardized tests and employs only objective formats to guarantee trustworthy scoring. Another example is the warehousing study by (Franke et al., 2025), where undergraduate exams originally in German were translated into English to make them accessible to the international research community and then administered to LLMs for comparison.

As demonstrated in *FanOutQA*, where students created intricate "fan-out" questions that required information from multiple Wikipedia articles and were then broken down into smaller questions that could be answered from single sources, another method entails creating new datasets through manual annotation (Zhu et al., 2024).

Lastly, *ZhuJiu* (Zhang et al., 2023) combines both approaches: it incorporates publically accessible datasets and adds newly created datasets produced using a ChatGPT-based self-instruction pipeline, with manual seeding and evaluation to prevent leakage and guarantee fairness.

These examples demonstrate the various ways to dataset compilation, but benchmark resources are smaller and more static than the massive, heterogeneous corpora needed for LLM training. The design of training and evaluation datasets also plays a central role in shaping benchmark outcomes. LLM development is based on vast and diverse collections of knowledge data, including books, journals, and websites, as well as structured data and multimodal sources such as images, audio, and video. How well a model generalizes across many contexts depends on the quality and diversity of these datasets, but benchmark datasets frequently reduce this richness to small, static samples (Miao et al., 2024). Furthermore, to avoid redundancy, bias, or toxicity, data management is based on organized pre-processing pipelines that include collection, filtering, deduplication, standardization, and review. Every step affects the model's ultimate capability.

For example, the data collection step necessitates identifying task-specific needs, selecting credible sources, and assuring privacy and legal compliance. Filtering stages frequently use

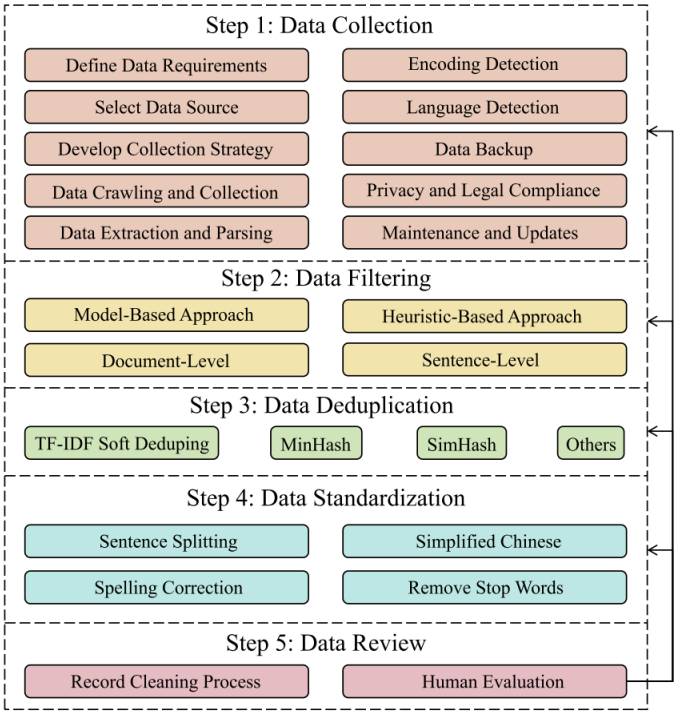



Figure 2.7: Preprocessing pipeline for pre-training corpora(Y. Liu, Cao, et al., 2024).

heuristic or model-based methods to filter low-quality, dangerous, or irrelevant content. Deduplication methods like *TF-IDF (Term Frequency-Inverse Document Frequenc) Soft Deduping* are used to remove redundant or too similar text segments, lowering noise in the corpus. Sentence segmentation, encoding correction, spelling normalization, and stop word removal are all part of the standardization process, which aims to provide cleaner and more consistent input. Finally, both automated and manual review systems ensure that errors or biases found earlier in the process are iteratively remedied. These procedures heavily influence model quality and fairness, but benchmarks seldom represent them, instead relying on static and simplified datasets that neglect the dynamic and curated character of genuine training corpora (Y. Liu, Cao, et al., 2024).

### 2.3.2 Evaluation Metrics

The literature shows that Large Language Model evaluation techniques can be broadly categorized into three different categories (Chang et al., 2024).

### 1) Metrics-Based Evaluation

The first approach for LLM assessment is *metrics-based evaluation*, which relies on predetermined quantitative criteria to measure model performance on existing datasets, providing objective and repeatable results. The most frequently used metrics in LLM benchmarks are *Accuracy* and the *F1-Score*.

#### *Accuracy*

Accuracy is defined as the proportion of the number of correct instances, both true positives and true negatives, out of to the total number of cases. It reflects the likelihood of randomly encountering a correctly classified occurrence, whether positive or negative. Equation 2.1

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \approx \frac{Number\ of\ correct\ instances}{Number\ of\ total\ instances} \tag{2.1}$$

#### *F1-Score*

F1-Score, also known as the *F-measure*, is the harmonic mean of *precision* and *recall*, giving equal weight to both. Precision and recall are defined as the probability of finding a truly relevant instance while predicting a positive, and the probability of finding the right instance when predicting correctly. Equation 2.2

$$F1{-}Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{2.2}$$

- Precision measures the proportion of correctly predicted positive instances out of all instances predicted as positive. In other words, it answers the question: *"When the model predicts positive, how often is it correct?"*. Equation 2.3

$$Precision = \frac{TP}{TP + FP} \tag{2.3}$$

- Recall measures the proportion of actual positive instances that were correctly identified by the model. It answers the question: *"Of all the real positives, how many did the model capture?"*. Equation 2.4

$$Recall = \frac{TP}{TP + FN} \tag{2.4}$$

While these metrics are valued for their simplicity and interpretability, they also have limitations. Accuracy can be misleading in highly imbalanced datasets, while the F1-score does not account for true negatives and may bias in favor of a majority class (Powers, 2015).

### 2) LLM-Based Evaluation

A second approach is *LLM-based evaluation*, also known as the *LLM-as-a-Judge* paradigm. In this setting, high-performing models are used to evaluate the outputs of other LLMs This method typically involves techniques such as prompt engineering, few-shot learning, and labeled responses, supported by repeated trials to enhance accuracy and stability (Gu et al., 2025). It can take three forms:

- *Pairwise comparison*, in which the judge chooses the better of two outputs or declares a tie;
- *Single answer grading*, in which the judge assigns a direct score to a single output;
- *Reference-guided grading*, in which the grading decision is based on a reference solution, which is especially useful in fields like mathematics.

Each method comes with trade-offs. Pairwise comparison provides robust relative judgments but scales poorly as the number of models grows. Single answer grading is more scalable but risks overlooking subtle quality differences. Reference-guided grading helps address domain-specific challenges but heavily depends on high-quality reference data.

Less reliance on human assessors, faster benchmarking cycles, and outputs that are interpretable and full of explanations are just a few benefits of the LLM-as-a-Judge. However, it is still susceptible to flaws such as verbosity bias (favoring solutions that are longer but equally correct), position bias (favoring responses in specific positions), and potential self-enhancement bias (favoring responses from the same LLM serving as judge) (Shi et al., 2025). The significance of continuous human monitoring in automated grading is further highlighted by the fact that LLM judges have the ability to improperly assess math or reasoning problems, even ones that they could solve correctly on their own (Zheng et al., 2023).

### 3) Human Evaluation

Lastly, Human evaluation is a fundamental aspect of LLM benchmarking, as it integrates subjective human judgment into the evaluation of model results. Many studies engage experts, stu-

dents, or professionals to evaluate model replies, which are often scored on a numerical scale (e.g., 1 to 5) to assess dimensions such as accuracy, completeness, or clarity. For instance, in a study conducted by (Mehri & Eskenazi, 2020), six researchers specialized in conversational AI rated system outputs across multiple qualitative aspects, such as understandability, naturalness, context maintenance, interestingness, and knowledge usage, before aggregating them into an overall quality score on a 1–5 scale.

Less frequently, human evaluation takes the form of academic grading, in which LLM responses get evaluated using the same criteria as university exams. A pertinent case is the study by (Franke et al., 2025), in which a faculty researcher scored ChatGPT's answers to three warehouse exams using the official sample solutions and the same grading system as students. By contrast, comparative evaluation takes a more natural approach, putting models in one-on-one arenas where human assessors directly compare their results, as popularized by Chatbot Arena (Zheng et al., 2023; Zhang et al., 2023). Chatbot Arena is a crowdsourced benchmarking tool that allows models to compete anonymously in head-to-head matches (Figure 2.8). Users engage with two unidentified models simultaneously, asking the same question and voting on their preferred response. Model identities are revealed only after voting, which mitigates evaluator bias. Unlike benchmarks with predefined prompts, Chatbot Arena allows users to ask unrestricted, spontaneously occurring inquiries, allowing for evaluation across a wide range of real-world use cases and interests (Zheng et al., 2023).

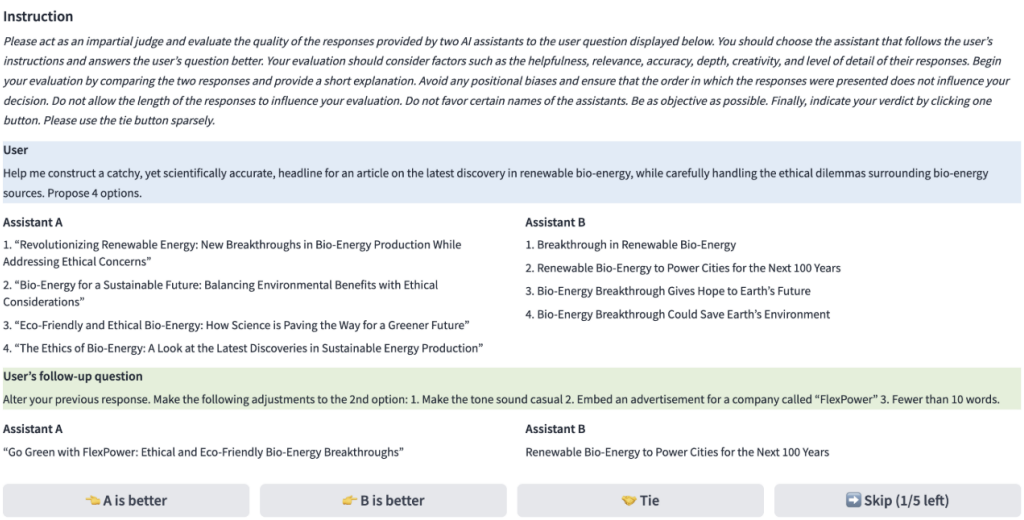



Figure 2.8: Chatbot Arena normal voting interface (Zheng et al., 2023).

The difference between structured evaluations (numerical scales or academic grading) and crowdsourced comparative ones emphasizes their complementary roles. While controlled set-

tings provide consistency and comparability, arena-based evaluation gives practical validity and alignment with real user expectations. However, both remain partial: structured evaluation is limited in variety and expensive to scale (Y. Wang et al., 2023), while crowdsourced votes could be noisy or biased (Zhang et al., 2023).

### 2.3.3 Challenges and Limitations

Evaluating Large Language Models remains challenging, as existing benchmarks often struggle to capture their true real-world performance and usefulness. Stability is a critical concern, since even minor changes to a prompt can lead to drastically different outcomes (Dam et al., 2024). Furthermore, designing fair assessments is complicated by ethical issues such as bias, privacy violations, and potential misuse, particularly in high-stakes industries where errors can have dire repercussions. A significant number of individuals utilizing AI lack technical expertise and engage with it across many text-based contexts (Figure 2.9).

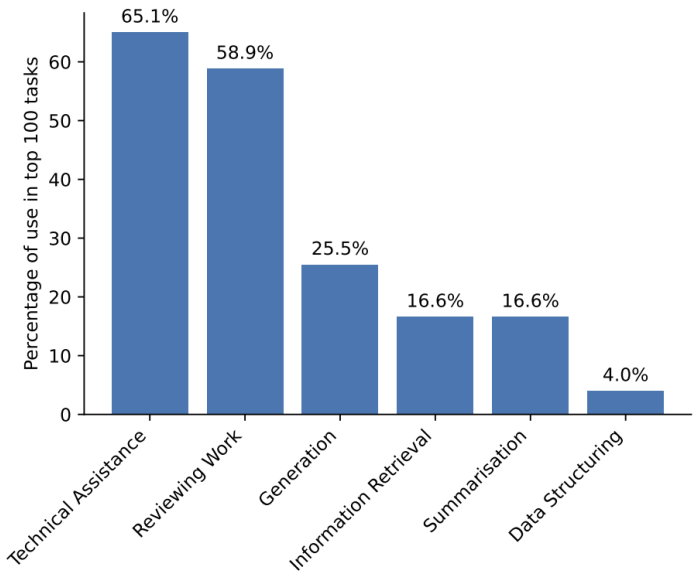



Figure 2.9: Prevalence of AI capabilities across the top 100 occupational tasks (Miller & Tang, 2025).

However, the majority of benchmarks assess limited tasks that are straightforward to evaluate, such as coding or recalling facts. Consequently, there is a gap between what benchmarks measure and how people actually use AI, since prevalent activities such as reviewing and refining written work are not included (Miller & Tang, 2025).

Moreover, most assessments neglect crucial aspects such as time savings, clarity, and sim-

plicity of integration into current workflows, prioritizing correctness over efficiency, interpretability, and contextual relevance (Eriksson et al., 2025). Beyond these limitations, existing benchmarks such as EconLogicQA (2024)(Quan & Z. Liu, 2024), PredictaBoard (2025) (Pacchiardi et al., 2025), FanOutQA (2024)(Zhu et al., 2024), frequently ignore linguistic and cultural diversity, preferring English over other languages such as Chinese.

Narrowness overlooks cultural nuances and alternative valid solutions shaped by different social, religious, or political contexts, thereby limiting inclusiveness and generalizability (Mushtaq et al., 2025). ZhuJiu was presented as the first comprehensive Chinese benchmark for LLMs in order to rectify this discrepancy. Although its uptake remains limited compared to English-centric frameworks, it provides both Chinese- and English-based evaluations and constitutes a step toward culturally grounded assessment (Zhang et al., 2023). These limitations are not only linguistic but also methodological.

Many evaluation methods rely on static forms, like multiple-choice questions or single-turn dialogue prompts, which fail to replicate the dynamic, multi-turn nature of real-world human-AI interactions, where consistency, coherence, and adaptability are essential (McIntosh et al., 2024). A related and ongoing issue is differentiating genuine reasoning from technical optimization, as models may learn to exploit benchmark-specific patterns or overfit to test structures rather than demonstrate real comprehension. This phenomenon, known as *benchmark gaming*, can artificially inflate outcomes and misrepresent a model's true capabilities, especially when evaluation datasets overlap with training data (Balloccu et al., 2024). Such concerns undermine the validity of benchmark results and may foster to overconfidence in deployment decisions.

The way benchmarks are used and interpreted is another limitation. Their proper application necessitates a thorough comprehension of methodological limitations and design choices. However, this knowledge is frequently underreported or ignored. This has resulted in cases where benchmarks such as *MMLU* (Hendrycks et al., 2021) or *BBQ* (Parrish et al., 2022) are applied inconsistently or their results are accepted at face value without taking into account the underlying assumptions.

These challenges are further compounded by the lack of standardized documentation for LLM benchmarks. At present, no specific frameworks exist to ensure consistent reporting of benchmark design, datasets, metrics, and evaluation assumptions, although some tools are available for characterizing AI datasets. The absence of such *benchmark metadata* makes it difficult for practitioners, regulators, and academics to evaluate benchmarks, choose appropri-

ate ones, and interpret results in light of real-world dangers (Reuel et al., 2024). The literature highlights several attempts to bridge this documentation gap. For instance, Sokol et al. (2025) presented *BenchmarkCards*, a structured framework designed to standardize the reporting of benchmark design, assumptions, metrics, and limitations, with the aim of improving transparency and alignment with intended use cases. Addressing these shortcomings requires the development of more comprehensive, transparent, and context-aware benchmarking methodologies that more accurately capture the diverse applications of LLMs in real-world contexts.

## 2.4 Benchmark Results Across LLMs

This section will present the results of benchmark analyses on several Large Language Models. It will show how models perform on a wide range of tasks and datasets, compare their strengths and weaknesses, and highlight developing patterns in capabilities and reliability.

### 2.4.1 Strengths and Limitations of LLM Performance

Recent benchmarks reveal heterogeneous outcomes that demonstrate both the benefits and limits of contemporary LLMs in various sectors.Across benchmarks, evidence shows that LLMs achieve strong results in structured and reference-based tasks but face difficulties with multi-step reasoning, domain-specific knowledge, and sophisticated real-world applications,as highlighted by recent benchmarks like *FanOutQA* (Zhu et al., 2024) and *EconLogicQA* (Quan & Z. Liu, 2024). In this latter benchmark, GPT-4-Turbo has the highest accuracy in both 1-shot and 5-shot settings, with GPT-4 following closely behind. This suggests that larger frontier models in sequential economic reasoning have a distinct benefit.

More broadly, LLM strengths emerge most clearly in standardized formats such as the natural language understanding tasks originally codified by *GLUE* (A. Wang et al., 2019), or multiple-choice question answering as exemplified by *MMLU* (Hendrycks et al., 2021), where task boundaries are explicit and scoring criteria are objective. However, performance drops considerably when tasks involve integrating information across multiple documents, sustaining logical coherence over long contexts, or applying technical expertise within specialized domains (Guo et al., 2025)); in the *FanOutQA* benchmark, GPT-4-Turbo and Claude 2.1 performed best overall, particularly in the evidence-provided setting, though major obstacles remain for smaller or less specialized models (Zhu et al., 2024).

These gaps imply that, while current LLMs excel at surface-level recognition and recall, they remain limited in deeper reasoning, contextual adaptation, and specialized competence. This discrepancy explains why models that rank highly on benchmark leaderboards may not necessarily prove reliable in professional or educational settings (Mishra & Arunkumar, 2021; Talby, 2025),as demonstrated by (Lunardi et al., 2025), who showed that linguistic variance in prompts can considerably affect accuracy even when leaderboard rankings stay unchanged.

Evidence from applied domains supports this view: although LLMs perform consistently well on routine knowledge tasks, they significantly underperform in quantitative reasoning and domain transfer. For example, studies of economic reasoning (Quan & Z. Liu, 2024) and warehousing applications (Franke et al., 2025) show that human participants often retain a comparative advantage.

At the same time, advances in prompting techniques like chain-of-thought or multi-agent prompting suggest that cognitive constraints are shaped not only by task complexity but also by the strategies used to structure reasoning.Nevertheless, smaller-scale models do not appear to benefit from these methods to the same extent (Wei et al., 2022).

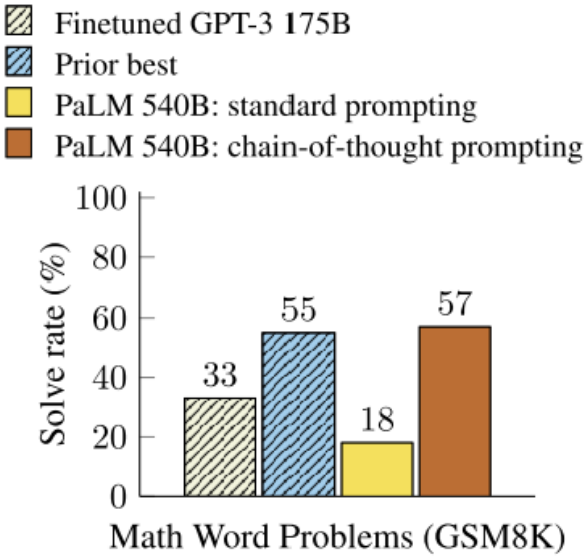



Figure 2.10: Impact of Chain-of-Thought prompting on mathematical problem-solving (Wei et al., 2022).

Building on this perspective, recent studies have introduced collaborative multi-agent and multi-path reasoning frameworks in which multiple independent instances of the same model act as agents, each tasked with a distinct reasoning role before exchanging their perspectives. By mimicking a collaborative approach to problem-solving, this technique allows models to take into account several points of view before reaching a decision (Z. Li et al., 2024). For

example, *Minstrel* (M. Wang et al., 2024) leverages structured prompt generation through agent collaboration to coordinate distinct reasoning paths, while *CoMM* (P. Chen et al., 2024) distributes complementary reasoning techniques across multiple agents and integrates their outputs to enhance robustness. Empirical evidence shows that such approaches improve performance in complex domains such as moral or ethical reasoning, where agent-to-agent dialogue helps balance conflicting opinions, even though highly technical disciplines like physics continue to reveal persistent problems (P. Chen et al., 2024).

In conclusion, these findings show that domain-specific benchmarks highlight the limitations of LLMs' applied competence, whereas advances in prompting,ranging from structured reasoning chains to collaborative multi-agent interaction, provide partial possibilities for closing these gaps. However, such methods remain constrained by scale and design, suggesting that strengthening reasoning capabilities requires not only improved data and evaluation practices but also new frameworks for orchestrating the cognitive processes of models.

# Chapter 3

# Research Methodology

This chapter illustrates the methodology adopted for the development of the research. After presenting the research questions and the exploratory framework that guided the work, the methodological choices related to the construction of the benchmarks will be described, along with the implementation and testing procedures that enabled the collection of results, which are discussed in the following chapter.

## 3.1 Research Questions and Exploratory Framework

In the previous chapter, the theoretical foundations of Generative AI, and more specifically Large Language Models, were analyzed in two distinct managerial domains: Project Management and Supply Chain Management. This analysis highlighted numerous contributions already available in the literature, bringing to light both the potential of LLMs in automating and supporting decision-making processes, as well as the methodological limitations that still remain. In particular, a clear gap emerged regarding the absence of systematic benchmarks that allow for comparable and replicable evaluation of LLM performance in real-world project and supply chain management contexts.

Building on this observation, the present research aims to contribute to filling this methodological gap. However, in order to ensure a more focused and coherent approach, from this chapter onward the analysis is narrowed exclusively to the supply chain domain, leaving the development of benchmarks for project management as a direction for future research.

The decision to focus on the supply chain made it possible to design specific benchmarks, built on datasets, evaluation metrics, and prompting strategies, capable of reflecting the actual

decision-making needs of companies in this sector.

In this perspective, the purpose of the present section is to introduce the research questions guiding the study and to define the exploratory framework that served as a methodological reference for benchmark design.

### 3.1.1 Research Questions

The research questions stem from two complementary needs. On the one hand, companies require reliable and accurate tools to support decision-making in sensitive areas such as Supply Chain Management. On the other hand, the literature has highlighted the absence of systematic frameworks for evaluating LLMs, which could represent a valuable resource for this domain.

Building on these premises, the research is guided by the following main objective: to determine whether Large Language Models can be considered reliable tools for supporting managerial decision making in supply chain management.

In line with this objective, two research questions have been formulated as follow:

- **RQ1**: Which combinations of datasets, evaluation metrics, and prompting techniques enable the construction of meaningful benchmarks for assessing LLM performance in supply chain contexts? $\rightarrow$ This question seeks to identify the most suitable methodological configurations to transform LLM experimentation into a systematic, replicable, and comparable process.

- **RQ2**: Which LLM currently demonstrates the best performance? $\rightarrow$ This question aims to determine, on the basis of the developed benchmarks, which model best integrates the main evaluation criteria, providing a comparative overview useful for guiding managerial selection.

These research questions are designed to ensure consistency between methodological rigor and practical relevance, linking the construction of benchmarks to their applicability in real managerial contexts.

### 3.1.2 Exploratory Framework

The exploratory framework is built around the concept of the benchmark as an instrument for integrated evaluation. It is structured into three main elements:

1. **Dataset**: consisting of a set of carefully selected questions that reflect typical supply chain issues in order to simulate decision-making scenarios.

2. **Evaluation metrics**: representing the criteria for measuring the performance of LLMs, including not only accuracy indicators but also other performance measures relevant to managerial use.

3. **Prompting techniques**: serving as the means through which the interaction with the models is shaped, guiding their behavior and optimizing their effectiveness in different scenarios.

The exploratory logic assumes that the interaction among these three elements generates different benchmark configurations. Each combination makes it possible to observe how LLMs respond to specific tasks. The practical implementation will then provide the results necessary to answer the third research question, namely how these benchmarks can effectively reflect the ability of LLMs to support managerial decision-making in complex supply chain contexts. The analysis of performance in real scenarios will allow not only for the comparison of different configurations but also for assessing their applicability in practice, with the aim of identifying the most effective strategies and improving the decision-making process. Furthermore, the systematic comparison of results will make it possible to address the second research question, which seeks to determine which LLM currently represents the most effective solution.

## 3.2 Benchmark Construction

Once the research objectives have been clearly defined, the next phase concerns the construction of the benchmarks. This section therefore presents the procedures adopted for dataset creation, starting from the analysis of question types as the foundation for building the datasets, and continuing with the criteria employed in the selection of evaluation metrics and the prompting techniques considered.

### 3.2.1 Question Type

In the design of a benchmark aimed at evaluating the performance of Large Language Models (LLMs), a crucial methodological aspect concerns the selection of the types of questions to be proposed. The structure of the questions, in fact, influences both the nature of the skills elicited,

such as calculation, reasoning, and planning,and the measurability of the results, affecting aspects such as objectivity of grading, reproducibility, and inter-rater reliability.

In the current literature on LLM benchmarks, there is a clear predominance of datasets based on multiple-choice questions. Tests such as MMLU (Massive Multitask Language Understanding) or similar tools are primarily built on multiple-choice tasks, where the model must identify the correct answer within a set of alternatives. This approach has evident advantages: It allows for standardized evaluation, reduces interpretative ambiguity, and makes results easily comparable across different models. However, due to their highly structured nature, such benchmarks tend to explore only a limited portion of model capabilities,particularly those related to pattern recognition or the retrieval of already encoded knowledge,while neglecting more complex aspects such as autonomous quantitative reasoning or the handling of articulated application scenarios. These skills, however, are fundamental in concrete supply chain applications. To overcome these limitations, the present research has chosen not to rely exclusively on the multiple-choice format, but to include heterogeneous types of questions, in order to construct a benchmark that is more comprehensive and representative of the real challenges an LLM may encounter in supply chain applications.

The main types of questions considered during the benchmark design phase are reported in Table 3.1.

| ID | Question Type | Description |
|---|---|---|
| Q1 | Close question – single-choice | A multiple-choice question with a finite set of options (typically 3–5), of which only one is correct. |
| Q2 | Close question – multiple-choice | A multiple-choice question in which two or more options may be correct. |
| Q3 | True/false | A closed-ended question presenting a statement to be answered by indicating whether it is true or false. |
| Q4 | Numerical answer | A question requiring an exact numerical response, usually derived from a calculation or quantitative data. |
| Q5 | Open question | A question requiring a discursive or argumentative response, without predefined options. |
| Q6 | Case study | A realistic and complex scenario requiring critical analysis and problem solving through a set of related questions. |

Table 3.1: Question types with their descriptions

***Q1: Close question - Single-choice***

Single-choice questions are among the most traditional formats used in evaluation. In this case, the model is given a finite set of options, usually three to five, with only one correct answer. This format offers objectivity in assessment, allows for automated grading, and minimizes ambiguity. Another strength of the format is flexibility: single-choice questions can be theoretical, aimed at testing definitions or conceptual knowledge, or numerical, where the model has to perform a calculation and select the correct answer from among the options. This two-sidedness makes them particularly well-suited to combining the evaluation of conceptual knowledge with basic quantitative skills. At the same time, there are some problems that remain, such as the possibility of guessing the correct answer, the over dependence on the quality of distractors, and the danger of cueing, when unconscious linguistic cues make the correct option more identifiable.

***Q2: Close question: Multiple-choice***

This type allows two or more options to be correct simultaneously, making it possible to assess more articulated knowledge compared to the single-choice format. Unlike single-choice, it is poorly suited to testing numerical skills, as its emphasis lies mainly on theoretical or conceptual knowledge. Moreover, some critical issues emerge: identifying the exact subset of correct answers may be ambiguous, the evaluation process is more complex, requiring decisions on whether to assign partial credit, apply an all-or-nothing approach, or use differentiated weighting, and the increased cognitive load does not always correspond to a real gain in informational value.

***Q3: True/false questions***

The true/false format represents the simplest modality. The model is presented with a statement and asked to determine its truthfulness. The construction and correction of such items are immediate and easily automatable, but the format has evident limitations. The most obvious is the high probability of a correct response by chance (50%), which drastically reduces the discriminatory power of the test. Furthermore, the presence of negations or ambiguous linguistic formulations can lead to misleading evaluations that do not accurately reflect the model's actual competence.

***Q4: Numerical answer questions***

Numerical answer questions require the model to produce a precise value derived from a calculation or a formula. They provide a high degree of objectivity, since the expected output is a unique number that can be directly compared with the correct solution. This type is particularly relevant in the field of supply chain management, where activities such as the calculation of the Economic Order Quantity (EOQ), the reorder point, infrastructure capacity, or service levels depend on numerical results. The main criticalities concern formatting issues (for example, the use of decimal separators or measurement units), rounding, and the need for clear and consistent normalization criteria for results.

### *Q5: Open-ended questions*

Open-ended questions are characterised by the absence of formal constraints: when presented with a theoretical prompt, the model is required to produce a discursive answer, support an argument, or provide an explanation. This format highlights argumentative ability, logical coherence, and the capacity to connect different concepts. However, the very lack of constraints also represents the main limitation. Evaluation inevitably becomes more subjective, reducing reproducibility of results; moreover, the analysis and correction of responses demand significant time and resources. Finally, the stylistic variability typical of different LLMs can further complicate comparison, as formally different answers may contain substantially similar content, or fluent texts may conceal conceptual errors.

### *Q6: Case studies*

Case studies represent the most complex type, and the one closest to real-world scenarios. In this format, the model is not required to identify a single answer, but rather to analyse an articulated problem, formulate hypotheses, and propose motivated solutions. This type enables the evaluation of advanced skills such as strategic reasoning, the ability to manage trade-offs, and decision-making consistency. At the same time, evaluation is complex and requires structured rubrics and the intervention of human assessors.

### From Bloom's taxonomy to the "difficulty pyramid"

A fundamental starting point for the construction of the benchmark was to identify a theoretical framework capable of guiding the definition of question complexity levels. In this regard, Bloom's taxonomy (Figure 3.1) represents a particularly useful tool. It describes cognitive processes as a hierarchy, ranging from the simple recall of information (Remember) to the production of new knowledge and solutions (Create). The intermediate levels (Understand, Apply, Analyze) are especially relevant in managerial and operational contexts, as they reflect the progression from recognizing basic concepts, to applying them in concrete situations, and finally analyzing them critically.

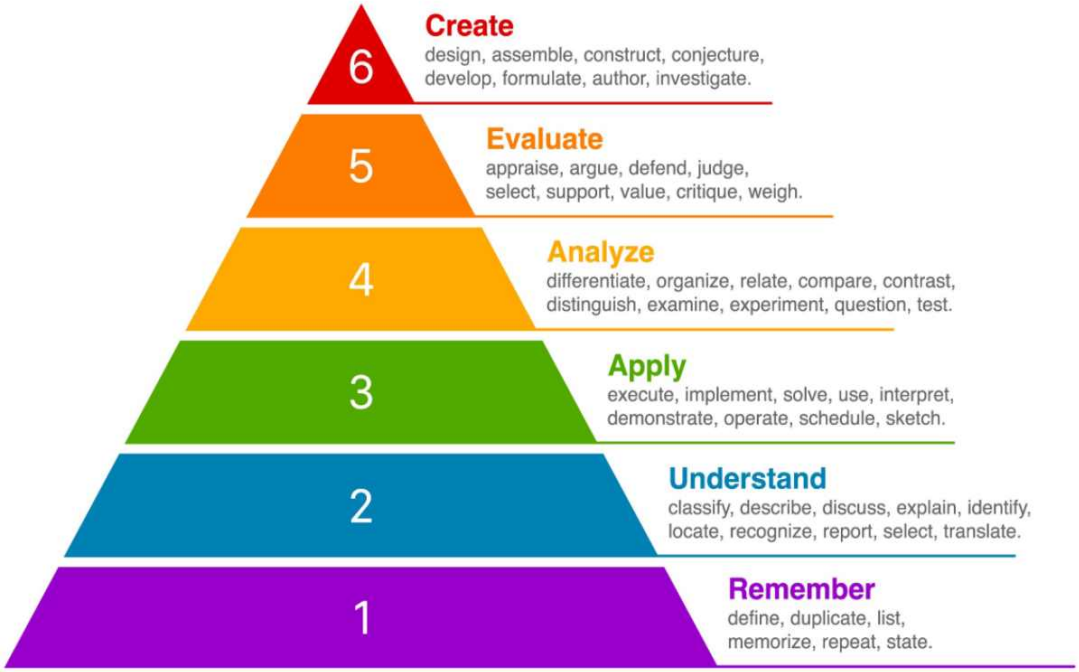



Figure 3.1: Bloom tassionomy

Based on this framework, it was necessary to select, among the different types of questions potentially suitable for a benchmark (as described in the previous section), a subset consistent with the supply chain context while remaining methodologically sound. The aim was to overcome the limitations of existing benchmarks, which rely almost exclusively on multiple-choice questions. In this perspective, three types of questions were selected to progressively reflect the different levels of Bloom's taxonomy:

- **Single-choice questions (Q1)**: testing basic knowledge and immediate recognition or comprehension skills, positioned at the lower levels of Bloom's hierarchy (Remember/Understand).
- **Numerical answer questions (Q4)**: requiring the application of formulas, manipulation of numerical data, and independent production of a result, corresponding to the Apply level.
- **Numerical answer questions with reasoning (Q4+)**: corresponding to the Analyze level, as they require not only the correct calculation but also the explicit explanation of the procedure, formulas used, and assumptions adopted.

To make this progression clearer and more operational, the logic of Bloom's taxonomy was

translated into a simplified representation adapted to the objectives of this work: the “difficulty pyramid” (Figure 3.2).

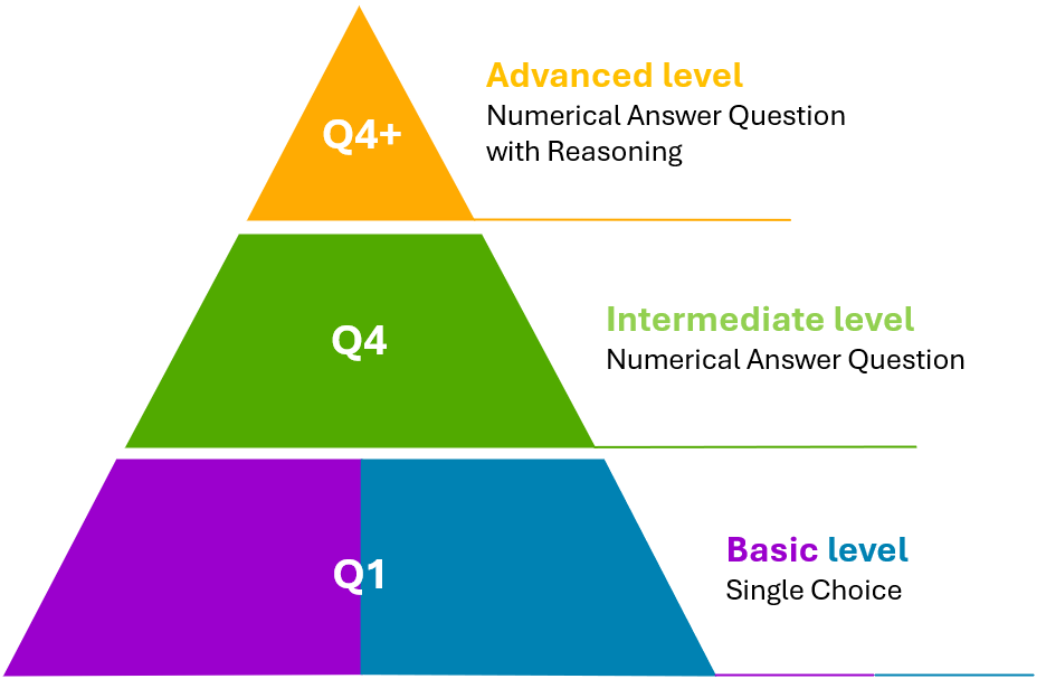



Figure 3.2: Difficulty pyramid

This pyramid, inspired by Bloom but tailored to the needs of the benchmark, organizes the question types into three progressive levels of difficulty:

- **Basic Level — Q1: Single Choice** Placed at the base of the pyramid, this represents the starting point of the evaluation. Single-choice questions provide an optimal compromise between ease of administration and objectivity of assessment. The presence of predefined options reduces ambiguity and makes it possible to test both theoretical knowledge and numerical skills, thus establishing a solid baseline reference.

- **Intermediate Level — Q4: Numerical Answer Question** At this level, the model is no longer guided by predefined options but must independently produce a numerical output. This introduces a higher degree of complexity and makes it possible to assess active skills such as logical-mathematical rigor and accurate calculation ability, both central to managerial and operational applications.

- **Advanced Level — Q4+: Numerical Answer Question with Reasoning** The final level combines the requirement of producing a numerical result with the obligation to make

the reasoning process explicit: applied formulas, logical steps, assumptions, and measurement units. This distinction makes it possible to separate calculation errors from conceptual gaps and reflects more closely the needs of supply chain contexts, where traceability and justification of the calculation process are as essential as the final result.

The transition from Bloom's taxonomy to the difficulty pyramid thus enables the integration of a general theoretical framework with a targeted application model. The outcome is a benchmark capable of going beyond the limits of traditional tests, providing a more realistic, structured, and context-relevant evaluation of supply chain challenges.

**Exclusion of other question types**

The definition of the Q1–Q4–Q6 triad simultaneously implied the exclusion of the remaining identified types. This decision did not stem from an underestimation of their potential, but rather from the need to guarantee methodological coherence, robustness of evaluation, and comparability of results.

***Q2 – Multiple-choice questions with more than one correct answer***

The multiple-answer format was excluded primarily because it introduces an excessive cognitive load, disproportionate to the actual informational value gained, particularly in a domain such as supply chain management, where clarity and verifiability of responses are essential. In addition, the higher complexity and arbitrariness involved in defining evaluation criteria risk undermining the methodological soundness of the benchmark.

***Q3 – True/false questions***

Binary statements were excluded as they are overly simplistic and weakly discriminative. The 50% chance of a correct response drastically reduces the statistical robustness of the test, while sensitivity to negations or linguistic nuances can produce misleading results, not always related to the model's actual level of knowledge or reasoning. In this format, the noise introduced tends to outweigh the useful information.

***Q5 – Open-ended questions***

Open-ended questions have the advantage of highlighting the discursive and argumentative abilities of the model but are of limited usefulness in the supply chain context, where the true challenge lies in assessing logical reasoning and problem-solving rather than the mere exposition of theoretical knowledge. Moreover, their evaluation inevitably requires human intervention, reducing reproducibility and comparability of results. For these reasons, this type was excluded in favour of more controllable and objective formats.

***Q6 – Case study***

Case studies were excluded because, although they represent a form of assessment closely aligned with real-world supply chain scenarios, they introduce a level of methodological complexity that is difficult to reconcile with the construction of a systematic and replicable benchmark. Their heterogeneity makes it challenging to define standardized evaluation criteria, increasing the risk of results that are not easily comparable across models. Moreover, analyzing a case study almost always requires a subjective interpretative process, involving human judgment in scoring, which reduces both replicability and automation in the evaluation framework. For these reasons, this question type was set aside in favor of more controllable and objective formats, while still acknowledging its relevance for future experimental or applied investigations.

### 3.2.2 Dataset Construction

Once the types of questions to be included in the pyramid of difficulty had been defined, the next step was the construction of the dataset, designed to coherently reflect the three selected categories: single-choice (Q1), numerical answer (Q4), and numerical answer with reasoning (Q4+). The dataset development phase is central, as the coherence of the questions, their level of difficulty, and their adherence to the application domain largely determine the reliability of the benchmark and, consequently, the robustness of the conclusions drawn. In defining the questions, a heterogeneous approach was adopted, integrating both academic and professional sources:

- Teaching materials from lectures and exercises at Politecnico di Torino (Italy) and RWTH Aachen University (Germany), two leading academic institutions in engineering educa-

tion and supply chain management research;

- Specialist seminars held at Politecnico di Torino, which provided insights into current issues and concrete cases in supply chain management;

- Examination materials from the ESCP Business School, a European institution of reference for managerial education. ESCP (École Supérieure de Commerce de Paris) is one of the world's oldest and most prestigious business schools, renowned for its international orientation and multidisciplinary programmes connecting theory, practice, and intercultural contexts.

The diversification of sources made it possible to construct a coherent and well-balanced dataset, capable of integrating different dimensions, conceptual knowledge, quantitative skills, and complex reasoning ability, and of organically reflecting the multi-level structure defined in the pyramid of difficulty.

***Database of single-choice questions (Q1)***

For the first type, single-choice questions, a database of 300 items was created, divided into:

- 200 theoretical questions, aimed at verifying the knowledge of concepts, definitions, and standard rules of supply chain management, without requiring calculations;

- 100 numerical questions, which instead involve calculations or applications of formulas related to supply chain management, in order to also assess logical and quantitative reasoning skills.

The questions were drawn from various sources: 100 from Politecnico di Torino, 100 from RWTH Aachen University, 80 from ESCP, and 20 from specialist seminars. This distribution ensures not only cultural and academic variety but also robustness, as the questions reflect different didactic and methodological perspectives. Examples of included questions:

- **Theoretical question (Q1):** *The three fundamental flows in any supply chain follow a typical order. What is the correct chronological sequence in traditional B2B trade?*

  a) *Physical* $\rightarrow$ *Information* $\rightarrow$ *Financial*

  b) *Financial* $\rightarrow$ *Information* $\rightarrow$ *Physical*

c) *Information → Physical → Financial*

d) *Information → Financial → Physical*

- **Numerical question (Q1):** *A manufacturer has a production cost of €50 per unit and sells to the retailer at a wholesale price of €75 per unit. The retailer, in turn, sells to customers at €100 per unit. A total of 15,000 units were produced, but only 10,000 units were sold. Under a traditional wholesale contract, what is the manufacturer's profit?*

  a) *400,000 €*

  b) *125,000 €*

  c) *375,000 €*

  d) *400,000 €*

***Database of numerical answer questions (Q4)***

For the second type, numerical answer questions, a database of 100 items was created, drawn from different sources: 50 from Politecnico di Torino, 40 from RWTH Aachen University, and 10 from specialist seminars. A methodologically relevant aspect is that these 100 questions coincide exactly with the numerical questions already used in the single-choice database (Q1). This choice was made to enable a direct comparison between two different administration modes: in the case of Q1, the model has four numerical options to choose from, thus being guided towards the solution; in the case of Q4, instead, the options disappear, and the model is required to calculate the correct result autonomously, without any external constraints or hints. This makes it possible to evaluate whether LLM performance depends on the ability to recognise the correct value among proposed alternatives, or on the actual ability to compute it independently. The questions were structured across three levels of difficulty:

- **Simple level.** Questions based on elementary formulas, with essential data and no superfluous information.

  – *Example: A Kanban card links two adjacent workstations whose combined demand rate is 600 units/day and container size 50 units. With a safety factor of 0.2 and lead time 1 day, how many Kanbans are required? (round down to the nearest whole number).*

- **Medium level.** Questions with more articulated data requiring intermediate logical steps.
  - *Example: A supermarket is supplied with agnolotti by an industrial pasta maker. Given the following data: the weekly demand (kg) is: 10 12 11 14 12.5 11 10 9 14 13.5 9.5 12 10 13 14.5 9 9.5 12 13 11.5 9.5 14 13 12 11. The Order lead time (days) is: 4 4 5 3 4.5 5 5 3 4 3.5 5 5.5 3 4 4 5 4.5 5 6 6 4 4 5 3 3. The target service level is 98%. What is the order quantity? (consider 7 working days per week).*
- **Difficult level.** Questions simulating complex business scenarios, with texts rich in information, not all relevant. The model must select the pertinent data and carry out chained reasoning.
  - *Example: "Giga is a company operating in the retail sector and in particular sells textile products. In the past, the company experienced very strong growth linked both to the increase in the number of stores and to the increase in sales per unit of surface area. In the last 4 years, the company sold 7,000, 8,000, 9,000, and 10,000 items. The company has grown in the last 4 years from 70 to 75, 80, and 88 stores. This growth has made the company very financially solid and, therefore, the cost of capital is 6% per year. The company has historically been based on the Zara model: short life-cycle products at very low prices, especially for young people. However, as its loyal customer base ages, the company has decided to launch a new product line called Basic, which today accounts for 40% of revenues. These products are characterized by higher-quality raw materials and more classic designs. This has reduced the level of demand uncertainty (before the beginning of the sales season, from the traditional 60% for fashion products to 30% for the Basic line products). For these reasons, the company has decided to change its commercial strategies for the new Basic product line. The articles traditionally marketed by Giga are produced by local suppliers with a raw material cost equal to 40% of the final price, to which processing costs equal to 25% of the final price are added. The retail price of an average item (of this type) is €80. The supply chain is not flexible enough to adapt during the season to sales: orders are set once and for all before the beginning of the season, even if several delivery dates are scheduled (from August to November) to the central warehouse, from which individual stores are then supplied. Logistics costs amount to about €4 per item.*

*The average retail price of a typical item is €80. In a typical store, 537 different articles are stocked, for a total area of 200 square meters. The annual rent cost for a typical store is €14,000/year. In a typical store, 10 people work, 4 of whom are part-time (for a total of 8 Full Time Equivalent). The staff cost for an average store is €240k/year, although larger stores can reach a cost of €500k/year. At the end of the sales season, unsold products are repackaged and shipped to the company's outlets where they are typically all sold during the two months following the end of the season. The handling and transport cost to the outlets is €5/item. In the outlets, the products are sold off on average at a price equal to 60% of the initial full price. The outlets are very popular and, therefore, normally all unsold goods during the regular season are completely sold off during the following season (in the first two months). Consider a product for which you expect to sell 250 units over the entire year. How many units do you decide to purchase?"*

The three-tier difficulty structure allows for the analysis not only of the accuracy of the calculation, but also of the ability of LLMs to handle complex texts and isolate relevant information from redundant information, an essential skill in decision-making contexts within the supply chain.

***Database of numerical answer questions with reasoning (Q4+)***

For the Q4+ level, conceived as an extension of basic numerical questions, a specific database was created based on the material already included in Q4. Specifically, only medium- and high-difficulty questions were selected, as they were deemed more suitable for eliciting structured reasoning in addition to the calculation of results. The overall database comprises 50 items,30 of medium difficulty and 20 of high difficulty, while maintaining the same distribution of sources already used for the other datasets: Politecnico di Torino, RWTH Aachen University, specialist seminars, and ESCP. The defining element of this level does not concern the content of the questions themselves but the mode of response expected: the model is no longer required merely to provide the correct numerical value but must also accompany it with a structured explanation of the logical procedure followed.

### 3.2.3 Evaluation Techniques

After defining the types of question on which to test the language models and constructing the corresponding datasets, it was necessary to identify the evaluation techniques to be employed in analyzing the generated responses. The definition of metrics represents a central element in the design of the benchmark, as the reliability of the results and the possibility of conducting meaningful comparisons across different models depend directly on their robustness. The literature has identified several evaluation approaches, as already shown in the previous chapter, which can be grouped into three main categories: metric-based evaluation, human-based evaluation, and LLM-based evaluation, as reported in the Table 3.2.

The first category, metric-based evaluation, relies on automatically computable quantitative indicators such as accuracy, F1-score, response time (latency), number of tokens used, and computational cost. These metrics have the advantage of ensuring objectivity, reproducibility, and ease of comparison, thus enabling a standardized assessment of model performance. However, they mainly capture the surface aspects of responses (formal correctness, computational efficiency) without fully representing the quality of reasoning or the depth of content.

The second category, human-based evaluation, involves the direct intervention of human annotators, who assign scores to each response according to predefined criteria (human grade) or compare two outputs in pairs, selecting the one deemed superior (human comparative judgment). This approach makes it possible to capture qualitative dimensions that are difficult to measure through automatic metrics, such as argumentative coherence, clarity of exposition, or contextual relevance. On the other hand, human evaluation entails higher costs in terms of time and resources, while also introducing elements of subjectivity and reducing the reproducibility of results.

Finally, the third category, LLM-based evaluation, designates a language model itself as the evaluator, judging responses generated by other models (or by itself) according to criteria specified in the prompt. This technique combines execution speed with the ability to capture more nuanced qualitative aspects. Nevertheless, it raises critical concerns regarding reliability, potential bias, and dependence on the formulation of the evaluation prompt.

| | | Evaluation Techniques | Definition |
|---|---|---|---|
| **Metrics-based Evaluation** | **E1** | Accuracy | Percentage of correct answers over the total number of questions. |
| | **E2** | F1 Score | Harmonic mean of Precision and Recall. Evaluate the quality of a classifier when it is important to consider both type I errors (false positives) and type II errors (false negatives).<br>Precision: proportion of correctly predicted positive cases over all predicted positives.<br>Recall: proportion of correctly predicted positive cases over all true positives in the dataset. |
| | **E3** | Latency | Response time of a model from the reception of an input to the generation of the complete output. |
| | **E4** | Token used | Total number of tokens processed by a model in an interaction, including both the input (prompt) and the output (generated response). |
| | **E5** | Cost | Economic expenditure required for the execution of the model, calculated as a function of the total tokens used (input + output) according to the provider's pricing. |
| **Human-based Evaluation** | **E6** | Human grade | Annotators assign a score from 1 to 5 based on predefined evaluation criteria. |
| | **E7** | Human comparative Judgment | Evaluation methodology based on pairwise comparison: human judges compare two responses generated by different models and select the better one. The process follows a tournament-style format, allowing rankings among models to be derived. |
| **LLM-based Evaluation** | **E8** | LLM as a judge | An LLM is employed as an evaluator to assess responses generated by other models (or by itself), based on prompts that define the evaluation criteria. |

Table 3.2: Evaluation techniques and their definitions

### Application matrix of evaluation techniques

After identifying the main evaluation techniques, it was necessary to define systematically their application across the different types of questions included in the difficulty pyramid. To this end, a correspondence matrix between evaluation techniques and question types was constructed, as reported in the Table 3.3, representing a fundamental methodological step. This approach makes it possible to restrict the analysis to combinations that are genuinely meaningful, thereby avoiding, on the one hand, redundant or uninformative applications, and on the other, the use of metrics inconsistent with the nature of the question. For completeness, the matrix also includes question types not selected in the difficulty pyramid, together with the related methodological considerations. In this way, the table does not merely present the combinations adopted in the present research, but instead provides a broader and comparative view of the possible alternatives.

| QUESTION TYPE | E1 | E2 | E3 | E4 | E5 | E6 | E7 | E8 |
|---|---|---|---|---|---|---|---|---|
| Q1 | X | | X | X | X | | | |
| Q2 | | X | X | X | X | | | |
| Q3 | X | | X | X | X | | | |
| Q4 | X | | X | X | X | X | | |
| Q5 | | | X | X | X | X | X | X |
| Q6 | | | X | X | X | X | | X |

Table 3.3: Evaluation techniques applied to different question types

| **Legend** | |
|---|---|
| Accuracy | E1 |
| F1 Score | E2 |
| Latency | E3 |
| Token used | E4 |
| Cost | E5 |
| Human-grade | E6 |
| Human comparative Judgment | E7 |
| LLM as a judge | E8 |
| Close question - single-choice | Q1 |
| Close question - multiple-choice | Q2 |
| True/false | Q3 |
| Numerical answer | Q4 |
| Open question | Q5 |
| Case study | Q6 |

Table 3.4: Legend of evaluation techniques (E) and question types (Q)

In the matrix, the "X" marks indicate the combinations considered methodologically appropriate. Each choice was guided by a careful reflection on the relationship between the characteristics of the question and the ability of the metric to provide useful information.

- **Accuracy (E1)**: this metric was associated with question types characterized by objective and unambiguous answers (Q1, Q3, Q4). In these cases, correctness can be verified without margins of ambiguity, making accuracy a simple yet reliable measure. For multiple-answer questions (Q2), however, accuracy proves less representative, as it does not distinguish between completely wrong responses and partially correct ones.

- **F1-score (E2)**: this metric was applied exclusively to Q2, where multiple answers can simultaneously be correct. Unlike accuracy, which evaluates responses in a binary way (all correct or all wrong), F1-score is able to recognize partially correct answers. In practice, this metric assigns an intermediate score when the model identifies only part of the correct options or includes both correct and incorrect ones. In this way, F1 provides a more nuanced and faithful measure of the overall quality of the response compared to accuracy alone, which in such cases would simply return a value of zero.

- **Latency, Token used, and Cost (E3–E4–E5)**: these metrics were considered transversal, as they measure aspects of computational efficiency and economic sustainability regardless of the question's content. For this reason, they were applied to all question types (from Q1 to Q6). Their inclusion was deemed essential, since a model capable of providing correct answers but with excessive execution times or disproportionate costs would be unsuitable for concrete use in supply chain processes.

- **Human grade (E6)**: the use of human evaluators was limited to contexts where subjective judgment adds real value. This applies to numerical questions (Q4) of medium-to-high difficulty, where answers may show slight deviations yet remain methodologically valid, and especially to case studies (Q6), which require qualitative evaluations of aspects such as reasoning consistency, plausibility of assumptions, or clarity of exposition. For closed and objective questions, on the other hand, human intervention would have been redundant and difficult to justify.

- **Human comparative judgment and LLM as a judge (E7–E8)**: these techniques were reserved for open questions and complex scenarios (Q5 and Q6), where no univocal solution exists. Comparing multiple outputs or employing an LLM as evaluator allows for capturing qualitative nuances and stylistic differences that cannot be measured with standard metrics. For closed-ended questions, their use would instead have been excessively resource-intensive and of limited added value.

Once the general mapping was completed, the selection of the combinations effectively adopted in the present research focused exclusively on the question types identified in the difficulty pyramid (Q1, Q4, Q4+). For these categories, the metrics considered most appropriate were highlighted in green.

For single-choice questions (Q1), the selected metrics were E1 (accuracy), E3 (latency) and E5 (cost). Accuracy represents the most immediate and objective measure of correctness, while the other metrics were chosen to monitor operational dimensions such as execution time, token consumption, and associated costs, essential elements for assessing efficiency.

For numerical questions (Q4), the metrics adopted were E1 (accuracy), E3 (latency) and E5 (cost). Accuracy ensures an objective measure of the correctness of the returned value, while the other three metrics allow monitoring of operational aspects related to execution times, resource consumption, and economic sustainability. Together, these dimensions provide a com-

prehensive evaluation of both the model's effectiveness and its computational efficiency.

For numerical questions with reasoning (Q4+), it was deemed necessary to complement the metrics already used for Q4 (E1, E3, E5) with human evaluation (E6 – human grade). In this case, it is not sufficient to verify the numerical correctness of the output: it becomes crucial to assess the quality of the reasoning provided, the coherence of logical steps, the relevance of assumptions, and the correctness of the formulas employed. These aspects, which cannot be quantified through automatic measures, require human intervention to ensure a complete and reliable evaluation.

The E4 metric (token cost) was used to calculate the costs, but it was not considered in this research as a proper evaluation metric.

**Analytic Hierarchy Process (AHP)**

In addition to the techniques described above, the Analytic Hierarchy Process (AHP) was adopted with the aim of synthesizing model performance into a single comparative measure, integrating heterogeneous dimensions such as accuracy, latency, and cost. For each benchmark, starting from the results of these three metrics, AHP was applied to derive a final ranking that reflects the overall set of criteria in a balanced manner. The application of the method involved two main steps:

- ***Assignment of preferences***

  To compare models, a preference scale was defined from 1 to 10, where a model was considered preferable to another depending on the metric under consideration (for example, a lower cost was considered preferable to a higher one, while higher accuracy was preferred to lower accuracy). At first, the possibility of assigning scores based solely on each model's rank was considered. However, this approach proved inaccurate as it did not account for the actual distance between values: models that were very close would have been penalized in the same way as models that were far apart, with the risk of underestimating or overestimating real differences.

  To overcome this limitation, the range between the maximum and minimum values of each metric was divided into ten equal-width classes (quantiles). Each class was associated with a score from 1 to 10, proportional to the observed gap. Formally, letting $\Delta$ be the absolute difference between two models for a given metric, and $M_{\text{max}}$ and $M_{\text{min}}$ the maximum and minimum values observed, the preference classes were defined as follows:

$$\Delta \in (0, 0.1\,(M_{\text{max}} - M_{\text{min}})) \Rightarrow 1$$

$$\Delta \in [0.1\,(M_{\text{max}} - M_{\text{min}}), 0.2\,(M_{\text{max}} - M_{\text{min}})) \Rightarrow 2$$

$$\dots$$

$$\Delta \in [0.9\,(M_{\text{max}} - M_{\text{min}}), +\infty) \Rightarrow 10$$

In this way, the scale not only reflected the relative ranking but also incorporated the magnitude of the actual difference. For example, two models with response times of 10.5 and 11 seconds received a very low preference score (class 1), while two models with latencies of 10.5 and 55 seconds fell into a high class, more realistically highlighting the superiority of the faster model.

- ***Determination of metric weights***

  The second step concerned the assignment of relative importance to accuracy, latency, and cost. To this end, a survey was conducted among a group of evaluators, invited to place themselves in the role of supply chain managers and to assign each criterion an importance score between 1 and 7, according to the traditional scale used in AHP. The evaluators were Master's students at Politecnico di Torino, specialized in Management Engineering for the Supply Chain, selected as a representative profile of future decision-makers in business contexts. The questionnaire included three main questions:

  – *Accuracy* – How important do you think it is that the answer provided by the LLM is correct?

  – *Cost* – How important do you think the cost of generating the answer is? (Considering that an answer to a complex question can vary from $0.01 to $0.10)

  – *Latency* – How important do you think execution time is to generating the answer? (Considering that an answer to a complex question can vary from a few seconds to 6 minutes)

  The aggregation of the results made it possible to derive the final weights to be applied in the multicriteria synthesis process, which were then used to build the AHP rankings of the different models.

The choice of AHP was motivated by the need for a tool capable of integrating objective data and managerial preferences within a coherent and transparent methodological framework. Compared to other multi-criteria methods, it allows for balancing trade-offs among different criteria, actively involving decision makers in the definition of priorities, and providing a final result in the form of a ranking of language models. Such a ranking serves as a practical reference for identifying the model most suitable for real operational scenarios, as it balances answer accuracy, execution speed, and economic sustainability.

### 3.2.4 Prompt Techniques

Once the dataset structure and the evaluation metrics have been defined, the next step is to understand how LLMs can interact with them.

The literature highlights that one of the key features of LLMs is their ability to interpret prompts expressed in natural language and adapt their responses according to the specific request. Consequently, in order to provide a tool capable of maximizing LLM performance, it is essential to assess different prompting techniques. The table below reports the prompting techniques identified in the literature, together with a brief description to facilitate the reading of this chapter. Table 3.5

| **Prompt Techniques** | **Description** |
|---|---|
| Zero-shot | The model is provided with only a textual description of the task to be performed, without including any explicit input-output examples. |
| One-shot / Few-shot | The model is provided with one or few illustrative examples of the task, followed by a new instance to solve. |
| Role prompting | A functional identity is assigned to the model (professor, expert, etc.) to adjust the tone, style, and level of expertise in its responses. |
| Chain-of-Thought (CoT) | The model is exhorted to solve the problem step-by-step, explaining the logical steps. |
| Self-consistency | The model is executed several times on the same prompt. The most frequent or most consistent response is selected. |
| Tree of Thoughts (ToT) | The model explores multiple reasoning branches simultaneously. |
| ReAct | The model alternates between phases of reasoning and operational phases (acting), such as consulting external sources or interacting with digital tools. |

Table 3.5: Prompt techniques with their descriptions

Building on this comprehensive overview of prompting techniques, it was necessary to evaluate which approaches were most suitable for meeting the specific objectives of this research. In particular, returning to the primary aim of the study, the focus was placed on identifying prompting strategies that could best support management in real operational contexts.

From this perspective, the *Zero-Shot* approach proved to be more appropriate than the *One-Shot* and *Few-Shot* alternatives. A manager typically expects the model to provide a solution to a problem without relying on predefined examples, either due to limited domain-specific knowledge or constraints of time and resources. Including examples within the prompt does not reflect this scenario, whereas *Zero-Shot* prompting represents a more realistic condition. Moreover, from a computational standpoint, *Zero-Shot* enables the rapid processing of large volumes of data and questions, while respecting the time and resource limitations of this research.

Another technique well suited to the analyzed scenario is *Role Prompting*. As highlighted in the literature, this approach not only allows for the definition of a specific conversational tone but also leads to improved performance. In this study, the instruction *"You are a Supply Chain*

*Manager"* was added to the prompt, so that responses would reflect a technical style aligned with a managerial perspective. Since this is a stylistic choice supported by well-established findings in the literature, role prompting was applied consistently across all analyzed scenarios.

In order to investigate possible improvements in response performance, and following the direction suggested by several academic contributions, an additional benchmark was designed for the single-choice (Q1) and numerical-answer (Q4) scenarios by introducing the *Chain of Thought (CoT)* technique. The prompt was enriched with the instruction *"Let's think step by step"*, intended to stimulate a gradual reasoning process before reaching the final solution. In these cases (Q1 and Q4), the CoT remains *implicit*: the reasoning unfolds internally, but the intermediate steps are not displayed in the answer. This configuration was chosen to examine how performance changes under such conditions and, more specifically, to test a setting aligned with situations in which managers primarily need a concise result without additional explanatory material.

In contrast, the numerical-answer scenario with reasoning (Q4+), *CoT* is required *explicitly*: the response must include not only the final value but also the logical progression leading to it. This option reflects an essential requirement in the field of Supply Chain Management, where the quality of a decision is assessed not only on the outcome but also on the reasoning that supports it, allowing potential weaknesses in the decision-making process to be identified.

*Self-Consistency* is often used in combination with *Chain of Thought*. In more complex tasks that require advanced reasoning, multiple logical pathways may emerge, and this technique allows the consideration of several responses generated from different reasoning chains. *Self-consistency* helps validate the robustness of the answers by comparing the various solutions produced by the model and selecting those that are the most frequent or consistent.

However, in the present study, this technique was not adopted as it would have resulted in a significant increase in computational and processing costs, without aligning with predefined analytical objectives. Therefore, it is considered an avenue for future research, where it can be explored to assess potential benefits in terms of accuracy and reliability of the responses.

Unlike *Self-Consistency*, *Tree of Thoughts (ToT)* develops along multiple reasoning paths and autonomously selects the branch leading to the final result. However, this technique demands considerable computational resources, both in terms of processing power and memory, to handle multiple decision pathways, backtracking activities, and alternative explorations. Such requirements reduce its scalability and limit its applicability in contexts characterized by

resource constraints or the need for rapid responses. For these reasons, this study chose not to adopt the technique, leaving its potential use to future research developments.

Similar reasoning applies to the *ReAct* technique, which alternates between the reasoning and acting phases, but also requires substantial computational effort. Its complexity limited its application at this stage of the research, though it may be considered in later phases where technological resources and contextual conditions are more favorable.

### 3.2.5 Final Benchmarks

The table 3.6 summarizes the details of each benchmark, including the datasets employed, the evaluation techniques applied, and the prompting strategies adopted. This overview provides a clear representation of the methodological choices made in each test scenario.

| **Benchmark** | **Question type** | **Evaluation** | **Prompting** |
|---|---|---|---|
| Benchmark 1 | Single choice | Accuracy, Latency, Cost | Zero-shot, Role prompting |
| Benchmark 2 | Single choice | Accuracy, Latency, Cost | Zero-shot, Role prompting, Implicit CoT |
| Benchmark 3 | Numerical answer | Accuracy, Latency, Cost | Zero-shot, Role prompting |
| Benchmark 4 | Numerical answer | Accuracy, Latency, Cost | Zero-shot, Role prompting, Implicit CoT |
| Benchmark 5 | Numerical answer with reasoning | Human grade, Latency, Cost | Zero-shot, Role prompting, Explicit CoT |

Table 3.6: Benchmarks with question type, evaluation criteria, and prompting techniques

#### General structure of the benchmarks

All benchmarks shared a common methodological framework, based on three main metrics: *Accuracy, Latency, Cost* (calculated as a function of tokens used). Among these, *Accuracy*

represented the central indicator and was analyzed at multiple levels:

- *Overall Accuracy*, measuring the global correctness of each model's answers;
- Accuracy by question type (*theoretical* and *numerical*), in order to highlight potential differences in behavior depending on content nature;
- Accuracy by difficulty level (*Easy, Medium, Hard*) for numerical questions only, with the aim of observing how model performance varied as task complexity increased.

These analyses made it possible to distinguish not only overall performance but also the models' sensitivity to question type and difficulty level. Finally, to integrate the set of evaluation criteria, the *Analytic Hierarchy Process (AHP)* was applied, enabling the synthesis of results into a comparative ranking of models that balances accuracy, time efficiency, and computational cost.

**Benchmark 1**

The first benchmark considered *Single-Choice* questions formulated in a *Zero-shot* setting with the addition of *Role Prompting* to steer the model toward behavior consistent with the decision-making context. The main objective was to establish a performance baseline in theoretical and numerical classification scenarios.

**Benchmark 2**

The second benchmark retained the same question type as B1 but added the *Chain-of-Thought* instruction. In this case, reasoning remained implicit (not shown in the final answer), allowing assessment of whether prompting step-by-step reasoning affected the correctness of choices.

**Benchmark 3**

The third benchmark focused on *Numerical* questions and required models to return only the value of the answer, with no explanation. This format enabled testing of "pure" calculation accuracy without explicit reasoning support.

**Benchmark 4**

In continuity with B3, the fourth benchmark added implicit *CoT* via the instruction *"Let's think step by step"*. The aim was to observe whether encouraging progressive reasoning could yield benefits, especially for more complex numerical items.

**Benchmark 5**

The fifth benchmark differed from the others as it evaluated not only the correctness of the final numerical result but also the quality of the reasoning made explicit by the model. To this end, a dedicated scoring system was introduced, assigning each LLM a maximum score of 1, distributed across three dimensions:

- *Calculation* (0–0.2): ability to correctly perform calculation steps;
- *Reasoning* (0–0.4): coherence and completeness of the reasoning provided;
- *Correctness* (0 or 0.4): accuracy of the final answer.

In addition to quantitative measurement, a qualitative analysis of reasoning errors was carried out, classified into two non-mutually exclusive categories:

- *Interpretation error*: related to misunderstandings of the problem statement or the incorrect use of available data;
- *Planning error*: stemming from flawed logical sequences, improper formula application, or disorganized solution steps.

From these evaluations, so-called category accuracies were derived, calculated as the ratio between the actual scores obtained and the theoretical maximum scores for each dimension. The final metrics therefore considered both traditional aspects (calculation, correctness) and qualitative aspects (reasoning, error type), providing a more granular representation of model performance.

## 3.3 Benchmark Implementation & Testing

After defining the benchmarks, the next phase concerned the experimentation on LLMs, with the objective of systematically analyzing their performance. The following section presents the

models selected for testing and describes the practical implementation, with reference to the software and tools employed.

### 3.3.1 LLMs selection

The first step was to select the LLM models to be evaluated among the numerous solutions currently available. The rapid diffusion of these models in recent years has fostered the entry of many companies into the sector, giving rise to a broad and continuously evolving market. As highlighted in the literature and confirmed by an exploratory analysis conducted online, the current offering includes multiple models, each designed to meet specific usage needs.

These solutions differ primarily with respect to three efficiency parameters: performance, latency, and cost. Each developer proposes variants of their model in an attempt to optimize the combination of these factors. However, achieving efficiency across all three dimensions simultaneously is not feasible: no model can deliver high performance with low latency while also maintaining low costs. As a result, different versions are developed that prioritize one characteristic at the expense of the others.

An example of this approach is represented by several next-generation language models, which offer different variants to balance performance, latency, and cost. A flagship version may be designed to guarantee high performance but with higher costs and greater latency due to the complexity of the computations involved. Conversely, a version optimized for speed and affordability may sacrifice performance on complex tasks. Finally, an intermediate variant may represent a compromise among these factors, offering a balanced solution for scenarios with variable requirements.

In practice, following an in-depth review, and after excluding certain models for geographical reasons (e.g., *Grok* by *xAI*, not yet available in the UK or EU) or due to issues with API acquisition and/or malfunction (such as *LLaMA* by *Meta* and *Qwen* by *Alibaba*), the following providers were selected:

#### Opena AI

*OpenAI*, a U.S.-based company founded in 2015, is one of the key players in the field of generative artificial intelligence and in the development of state-of-the-art language models. With the release of the *GPT-5* series, the company introduced three model variants, each designed to address different requirements in terms of performance, cost, and latency. *GPT-5* represents

the company's flagship model, characterized by strong reasoning capabilities and suitable for complex applications, though with longer response times and significantly higher costs. At the opposite end is *GPT-5 nano*, conceived to maximize speed and cost efficiency while sacrificing the ability to handle complex, cognitively demanding tasks. Positioned in between is *GPT-5 mini*, which offers a compromise among accuracy, speed, and economic sustainability.

The availability of these three differentiated variants motivated their inclusion in the comparative analysis, in order to evaluate how different trade-offs among performance, latency, and cost may affect practical managerial applications. Previous models (the *GPT-4* series and earlier versions) were not considered, as they are deemed obsolete and have been officially replaced by the *GPT-5* generation.

### Anthropic

*Anthropic* is a U.S.-based company founded in 2021 by former *OpenAI* members, with a strong focus on the safety and reliability of artificial intelligence systems. The *Claude* family of models stands out for its emphasis on reasoning capabilities and suitability for supporting complex scenarios. Within this family, *Claude Opus 4.1* represents the most advanced model, capable of delivering high-level performance but characterized by significantly higher costs and greater latency. *Claude-Sonnet 4*, by contrast, offers an intermediate solution, balancing accuracy and speed at a more sustainable cost level. Finally, *Claude-Haiku 3.5* prioritizes response speed and efficiency, while partially sacrificing the ability to manage particularly complex tasks.

Given the economic constraints of the present research, the comparative analysis included *Claude-Sonnet 4* and *Claude-Haiku 3.5*, while excluding *Claude Opus 4.1*, which was considered excessively costly ($15.00 / 1M input tokens and $75.00 / 1M output tokens) relative to the study's objectives.

### Google

*Google* is one of the leading global players in the field of artificial intelligence, supported by *DeepMind*'s contributions to the development of deep learning. In 2023, it launched the *Gemini* family of models, the successor to the *PaLM 2* line, designed to provide multimodal capabilities and native integration with the Google Cloud ecosystem. The most recent *Gemini-2.5* generation is characterized by a particularly large context length (up to 1M tokens), enabling a wide range of use cases. *Gemini-2.5 Pro* is the most powerful and versatile model but also

the most costly, with expenses varying depending on prompt length. *Gemini-2.5 Flash* is designed to optimize the price-performance ratio, offering solid performance at lower cost, while *Gemini-2.5 Flash-Lite* represents the most lightweight version, suitable for scenarios requiring high speed and low cost.

The *Gemini-2.5 Pro* version was not included in the analysis because, during preliminary testing, access through the model's beta API produced errors attributable to Google server malfunctions, which prevented correct code execution and, consequently, reliable evaluation.

**DeepSeek**

*DeepSeek* is a more recent Chinese provider that has distinguished itself in the LLM market through a highly competitive approach in terms of cost and accessibility, while still maintaining satisfactory baseline performance. The latest version (*V3.1*) features a maximum context length of 128k tokens and prices significantly lower than those applied by major international competitors. These characteristics make the model particularly attractive in scenarios where budget constraints play a decisive role.
In conclusion, the Table 3.7 reports the selected versions.

| **Owner** | **Version** | **Context Length** | **Input price ($/Mtok)** | **Output price ($/Mtok)** |
|---|---|---|---|---|
| OpenAI | GPT-5 | 400k | 1.25 | 10.00 |
| | GPT-5 mini | 400k | 0.25 | 2.00 |
| | GPT-5 nano | 400k | 0.05 | 0.40 |
| Anthropic | Claude-Sonnet 4 | 200k | 3.00 | 15.00 |
| | Claude-Haiku 3.5 | 200k | 0.80 | 4.00 |
| Google | Gemini-2.5 Flash | 1000k | 0.30 | 2.50 |
| | Gemini-2.5 Flash-Lite | 1000k | 0.10 | 0.40 |
| DeepSeek | DeepSeek-v3.1 | 128k | 0.56 | 1.68 |

Table 3.7: Comparison of LLM providers, version, context length, and pricing

The table also reports additional characteristics:

- *Version*: identifies the specific variant of the model released by the provider;

- *Context length*: indicates the maximum number of tokens a model can process in a single interaction, including both input (prompt, instructions, documents) and output;
- *Input price*: represents the cost, expressed in U.S. dollars, for processing 1 million input tokens (i.e., the text provided to the model as a prompt). (The cache miss price was considered);
- *Output price*: represents the cost, expressed in U.S. dollars, for processing 1 million output tokens.

### 3.3.2 Implementation

After defining the benchmarks and selecting the models, the practical implementation phase was initiated. The experiments were conducted in *Google Colab*, an environment that enabled straightforward management of both dataset loading and interaction with the APIs of the different LLMs. Each notebook followed the same logical sequence: importing libraries, loading the questions from file, defining execution parameters, setting the system prompt, calling the model, and finally recording the responses together with the corresponding token consumption and estimated costs.

#### *Library import*

The first step common to all benchmarks was the import of the libraries required for running the scripts. Some core libraries were included in every script to provide essential functions:

- *Time* was used to measure execution duration;
- *Pandas*: enabled the reading and management of datasets in Excel format;
- *Google.colab.files* allowed datasets to be uploaded directly into the Colab environment, simplifying the handling of input data.

```
Import time
From google import google.colab.files
Import panda as pd
```

In addition, each provider requires its own dedicated package, which makes it necessary to use different libraries for interacting with the models. Table 3.8

| Provider | Library |
|---|---|
| Anthropic | `!pip install -q anthropic`<br>`from anthropic import Anthropic` |
| OpenAI | `!pip install -q openai`<br>`from openai import OpenAI` |
| Google | `!pip install -q google-generativeai`<br>`import google.generativeai as genai` |
| DeepSeek (CoT) | `!pip install -q deepseek`<br>`from deepseek import DeepSeek` |

Table 3.8: Library installation and import examples by provider

Finally, the auxiliary *re* library was added in the numerical-answer benchmarks, as it was necessary to correctly extract the numerical values produced by the model.

### *Loading questions from file*

After importing the libraries, the next step in each benchmark was loading the dataset containing the questions and their corresponding correct answers. This phase was essential both to ensure consistency across tests and to maintain flexibility with respect to different task types.

The file upload was performed using the *files.upload()* function, which allows an Excel file to be uploaded directly from the local computer. Subsequently, with *pandas.read_excel()*, the data were read into a DataFrame, and the columns were renamed consistently as *"question"* and *"correct_answer"*.

```
from google.colab import files
uploaded = files.upload()
file_name = next(iter(uploaded))
df = pd.read_excel(file_name, header=None)
df.columns = ["question", "correct_answer"]
```

This procedure, identical across all benchmarks and models (*Claude, ChatGPT, Gemini, DeepSeek*), ensured that the pipeline consistently received a standardized data format as input.

***Definition of execution parameters***

Each benchmark was configured by setting the key parameters governing the interaction with the models: maximum number of tokens, and input/output costs.

- *Maximum number of tokens*: In all benchmarks, this parameter was set to the maximum value allowed by the provider for the specific model (for example, 8,192 for *Claude-Haiku 3.5*). This approach avoided the risk of truncation in open-ended tasks, while acknowledging that in closed tasks the actual consumption remained much lower;
- *Input and output costs*: Cost calculations were based on the official pricing declared by the providers, distinguishing between input tokens (prompts) and output tokens (generated responses). For example, for *Claude-Haiku 3.5* the cost is $0.80 per million input tokens and $4.00 per million output tokens.;
- *Input price*: represents the cost, expressed in U.S. dollars, for processing 1 million input tokens (i.e., the text provided to the model as a prompt). (The cache miss price was considered).

```
MAX_TOKENS  = 8192
in_cost   = 0.80 / 1_000_000
out_cost = 4.00 / 1_000_000
```

***Definition of the system prompt***

The definition of the system prompt represented a central step in the implementation phase, as it made it possible to put into practice the prompting techniques previously discussed. While the execution parameters (temperature, maximum number of tokens, costs) remained relatively standardized, prompt design required significant adjustments depending on the benchmark type and the chosen strategy.

In all benchmarks, role prompting was applied through the constant instruction “You are a Supply Chain Manager”. This choice aimed to give the responses a managerial and technical character, consistent with the perspective guiding the research scenario.

Furthermore, zero-shot prompting was adopted in all cases, without including explicit input-output examples, in order to reproduce conditions closer to real-world scenarios: a manager is expected to receive answers to new questions without the need for predefined examples.

- *Benchmark 1*: Only role prompting was applied, constraining the model to return a single letter between A and D. No additional reasoning cues were used

```
system_prompt = (
    "You are a supply chain Manager."
    "Answer the question you are asked with the letter from
    A to D."
    "Do not say anything else except the letter.")
```

- *Benchmark 2*: In addition to role prompting, Chain of Thought (CoT) was introduced with *"Let's think step by step"*. The goal was to test whether encouraging internal reasoning improved correctness, while keeping the final output to a single letter

```
system_prompt = (
    "You are a supply chain Manager. "
    "Let's think step by step. "
    "Answer the question you are asked with the letter from A to
    D. "
    "Do not say anything else except the letter.")
```

- *Benchmark 3*: The objective was to obtain a purely numerical output; role prompting was used

```
system_prompt = (
    "You are a supply chain Manager. "
    "Given a numerical problem, return ONLY the final number in
    digits, "
    "without text, symbols, or units. "
    "Use the dot as the decimal separator (e.g., 1234.56).")
```

- *Benchmark 4*: As in Benchmark 3, but augmented with implicit CoT to encourage step-by-step internal reasoning before producing the final number

```
system_prompt = (
    "You are a supply chain Manager. "
    "Let's think step by step. "
    "Given a numerical problem, return ONLY the final number in
```

```
    digits, "
    "without text, symbols, or units. "
    "Use the dot as the decimal separator (e.g., 1234.56).")
```

- *Benchmark 5*: A more structured prompt combined role prompting and explicit CoT. To facilitate parsing and evaluation, a strict output format was imposed.

```
system_prompt = (
    "You are a supply chain manager. "
    "Think step by step. Solve numerical problems showing clear,
    numbered steps "
    "with formulas and substitutions. "
    "After the reasoning, print the final answer on a new last
    line EXACTLY as: "
    "ANSWER=<number>")
```

***Model call***

Once the parameters and the system prompt were defined, the next step was the actual interaction with the model through the respective APIs. This phase was common to all benchmarks: for each question in the dataset, a request was generated to the selected model, passing as arguments the system prompt, the question, and the configuration parameters (temperature, max_tokens).

The structure of the call was almost identical for all providers (*Anthropic, OpenAI, Google, DeepSeek*):

```
response = client.messages.create(
    model=model,
    system=system_prompt,
    max_tokens=MAX_TOKENS,
    messages=[{"role": "user", "content": str(question)}])
```

The *response* object contained both the text generated by the model and the metadata related to token consumption and execution time.

The differences concerned the type of output expected and, consequently, the logic used to process the model's response:

- *Benchmark 1 and 2*: The model's output was reduced to a single letter (A–D). The script extracted the first valid occurrence contained in the response.

```
risposta = response.content[0].text.strip().lower()
prima_lettera = next((c for c in risposta if c in 'abcd'), '')
print(prima_lettera)
```

- *Benchmark 3 and 4*: A function with regular expressions was used to isolate the final number from the generated text, discarding any unwanted characters.

```
import re
num_pat = re.compile(r'[-+]?\d+(?:[\.,]\d+)?(?:[eE][-+]?\d+)?')

def only_number(s: str) -> str:
    m = num_pat.search(str(s))
    return m.group(0).replace(",", ".") if m else ""

raw = response.content[0].text.strip()
num = only_number(raw)
print(num)
```

- *Benchmark 5*: In this case, the model produced a detailed reasoning followed by a final line in the format ANSWER=<number>.

```
answer_pat = re.compile(r'ANSWER\s*=\s*([-+]?\d+(?:[
.,]\d+)?(?:[eE][-+]?\d+)?)')
num_pat_all = re.compile(r'[-+]?\d+(?:[.,]\d+)?(?:[eE][-+]?\d+)?
')

def extract_final_number(text: str) -> str:
    m = answer_pat.search(text)
    if m:
        return m.group(1).replace(",", ".")
    nums = num_pat_all.findall(text)
    return nums[-1].replace(",", ".") if nums else ""

raw = response.content[0].text.strip()
```

```
final_number = extract_final_number(raw)
print(final_number)
```

***Recording responses, usage, and estimated costs***

The final phase of each benchmark concerned the recording of the responses generated by the models, together with usage data (tokens, costs, and execution time). This step enabled the transformation of the model's output into a structured dataset, useful both for evaluating accuracy and for analyzing economic and computational efficiency.

In all scripts, the following values were computed:

- input tokens (prompt provided to the model);
- output tokens (generated response);
- estimated cost (calculated by multiplying tokens by the official rates);
- total execution time.

```
usage = response.usage
prompt_tokens = usage.input_tokens
completion_tokens = usage.output_tokens

total_prompt_tokens     += prompt_tokens
total_completion_tokens += completion_tokens
total_cost += prompt_tokens * in_cost + completion_tokens * out_cost

end_time = time.time()
elapsed_time = end_time - start_time

print(f"Estimated total cost: {total_cost:.4f} $")
print(f"TOTAL EXECUTION TIME: {elapsed_time:.2f} seconds")
print(f"TOTAL INPUT TOKENS: {total_prompt_tokens}")
print(f"TOTAL OUTPUT TOKENS: {total_completion_tokens}")
```

For the specific benchmarks:

- *Benchmark 1 and 2*: Only the letter corresponding to the answer was recorded. Token and cost information was printed at the end of execution, without additional intermediate details.

```
print(first_letter)
```

- *Benchmark 3 and 4*: The extracted numerical value was saved, ignoring any accessory characters. Token usage and costs were also recorded, with only the final number printed.

```
print(num)
```

- *Benchmark 5*: In addition to the final number, the complete explanation generated by the model was stored, allowing for a qualitative analysis of the reasoning process. In this benchmark, the aim was not only to verify the correctness of the result but also to assess the quality of the reasoning.

```
print("QUESTION:", question)
print("REASONING:", raw)
print("FINAL RESULT:", final_number)
```

## 3.4 Statistical Significance Testing

In addition to the evaluation metrics already discussed, it was necessary to verify the statistical significance of the differences observed between the benchmarks. Simple variations in accuracy do not guarantee that such differences are due to the introduction of a prompting technique or to the task itself, rather than to random fluctuations.

For this purpose, the *McNemar* test was employed, a widely used non-parametric method for comparing the performance of two classifiers on the same data. The test does not focus on overall accuracy but rather on the discordant cases: that is, the instances where one model provides the correct answer while the other fails, and vice versa. The idea is to assess whether these discrepancies are evenly distributed or whether one situation clearly prevails. In the first case, no significant differences between the models can be detected, whereas in the second it is possible to conclude that one of the classifiers exhibits a real advantage.

In this research, the *McNemar* test was used mainly to compare benchmarks based on the same type of questions but differing in the use of Chain-of-Thought. It was also applied to sce-

narios with *implicit* and *explicit* CoT. Analyses were conducted both on the overall samples of 300 questions per benchmark and on sub-samples by question type (theoretical and numerical) and difficulty level (easy, medium, hard).

The tests were run using the software *Stata*, which provides the $\chi^2$ statistic and the related p-values. Two approaches were considered: the asymptotic p-value based on the $\chi^2$ approximation; the exact p-value, based on combinatorial calculations, more reliable with small samples or few discordances.

Since in this study the maximum sample size was 300 questions and the number of discordances was often limited, the exact p-value was used as the main reference. Results were interpreted according to the conventional threshold of 95% significance ($\alpha = 0.05$): a difference was considered significant only when the p-value was below this level.

The use of the McNemar test strengthens the methodological validity of the analysis. It reduces the risk of over-interpreting marginal differences and provides a more reliable picture of LLM performance. The outcomes of the tests are presented in the next chapter, alongside the descriptive metrics, to give a complete evaluation in both descriptive and inferential terms.

# Chapter 4

# Results

## 4.1 Introduction

This chapter presents the results of the experimental activity, organized on several levels of analysis. It opens with the outcomes of the survey, which collected evaluators preferences and defined the weights used in the Analytic Hierarchy Process (AHP).

The section **Benchmark-level Results** reports the findings of the individual benchmarks, describing model performance in terms of overall accuracy, question type and difficulty, as well as operational parameters such as cost and latency. Each block of results is then summarized through the AHP, which makes it possible to combine different dimensions into a single comparative index.

The section **Cross-benchmark Comparison** adopts a transversal perspective, comparing the results obtained across the various benchmarks. In this context, statistical significance tests (McNemar) are also considered, in order to verify whether the observed differences should be interpreted as real effects or as random fluctuations.

The structure of the chapter thus makes it possible to move from the detailed analysis of individual benchmarks to a comparative and statistically validated reading, laying the groundwork for the critical discussion developed in the following chapter.

## 4.2 Survey

The survey is used to determine the relative importance of the evaluation criteria to be integrated into the Analytic Hierarchy Process (AHP). Participants, invited to put themselves in the role

of supply chain managers, were asked to assign each criterion (accuracy, cost, and latency) an importance score on a scale from 1 to 7, following the standard AHP approach.

A total of 30 evaluators were interviewed, and the reported mean values therefore reflect the average of the preferences expressed by this sample. The aggregation of the responses made it possible to derive the mean values, which were subsequently normalized and used as weights within the multi-criteria process. The results highlight a clear priority assigned to accuracy (mean 6.76), followed by latency (2.64) and, to a lesser extent, cost (2.3).

The results of the survey are reported below, respectively for the criteria of *Accuracy* (Figure 4.1), *Cost* (Figure 4.2), and *Latency* (Figure 4.3).

Table 4.1 reports the final normalized weights:

| **Criterion** | **Mean Value** | **Normalized weight** |
|---|---|---|
| Accuracy | 6.76 | 0.577 |
| Cost | 2.3 | 0.197 |
| Latency | 2.64 | 0.225 |

Table 4.1: Survey results

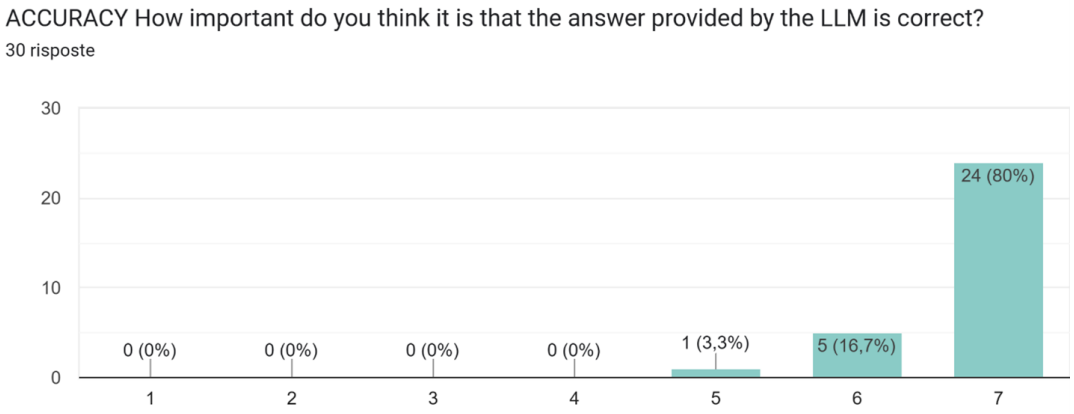



Figure 4.1: Survey results for Accuracy

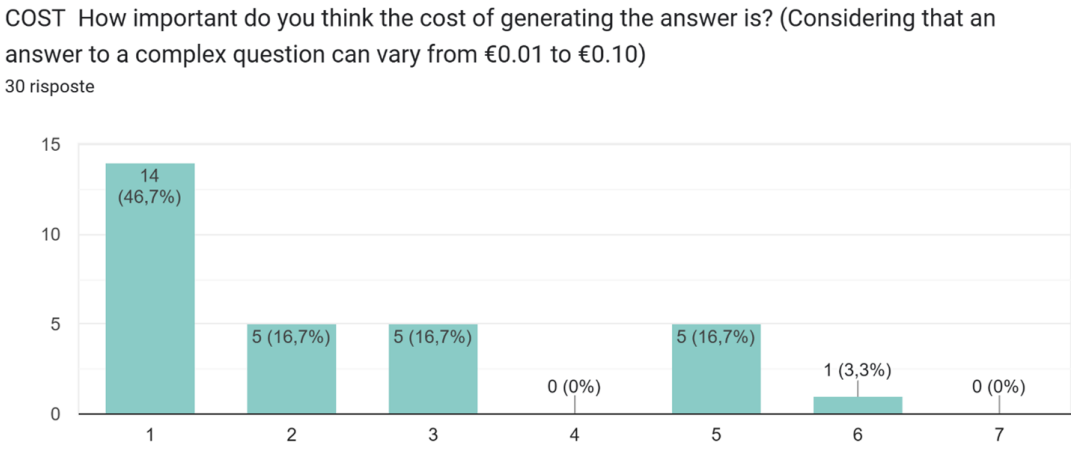



Figure 4.2: Survey results for Cost

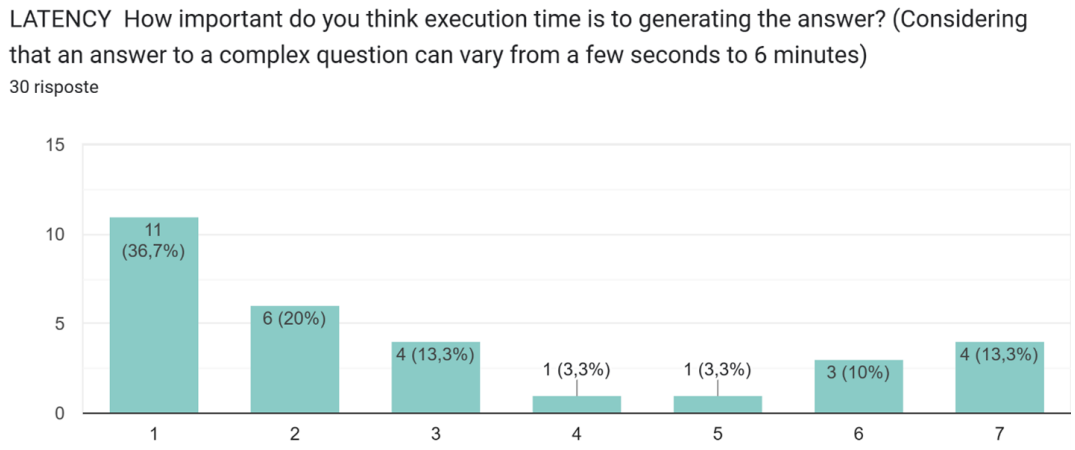



Figure 4.3: Survey results for Latency

## 4.3 Benchmark-level Results

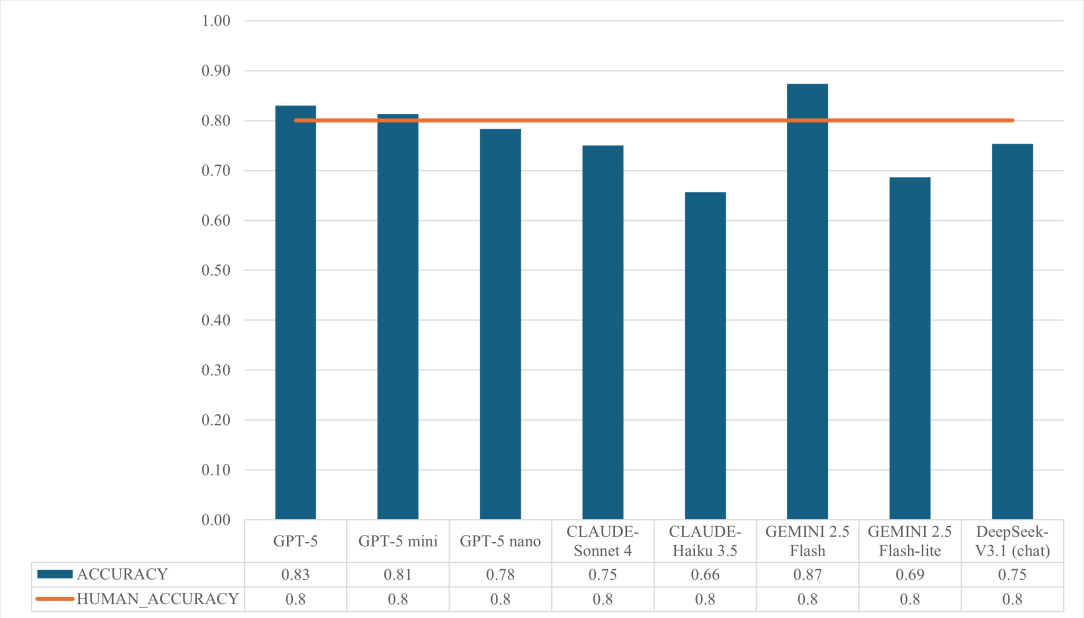
This section reports the results obtained from the different benchmarks designed. The analysis follows a progressive structure: for each benchmark, the overall and disaggregated accuracy values are presented (distinguishing between theoretical and numerical questions, as well as by difficulty level), followed by the comparison with the human evaluation threshold and the description of the operational parameters (cost and latency). Finally, the results are synthesized through the application of the Analytic Hierarchy Process (AHP), which makes it possible to integrate the different criteria into a single composite index and to produce a final ranking of the models. The weights used in the AHP were derived from the survey.

The following paragraphs present in detail the performance of the models across the five benchmarks.

### 4.3.1 Benchmark 1 Results

The first benchmark, based on single-choice questions, provided a baseline for evaluating model performance. Overall accuracy ranges from a minimum of 0.66 (Claude-Haiku 3.5) to a maximum of 0.87 (Gemini-2.5 Flash), with intermediate values for GPT-5 (0.83), GPT-5 mini (0.81), GPT-5 nano (0.78), Claude-Sonnet 4 (0.75), DeepSeek-v3.1 (0.77), and Gemini-2.5 Flash-Lite (0.69).

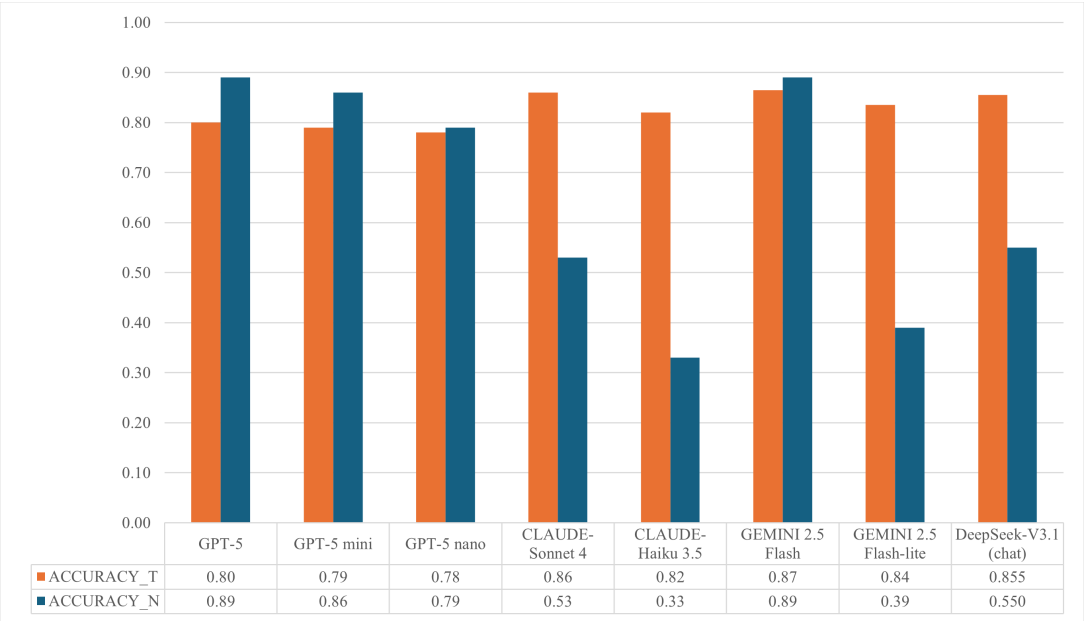
The comparison with human accuracy (set at 0.8) positions model results against a human baseline, defined as the average performance expected from a fifth-year management engineering student. On this basis, GPT-5 (0.83), GPT-5 mini (0.81), and Gemini-2.5 Flash (0.87) exceed the human average, while all other models fall below it. (Figure 4.4)




| | GPT-5 | GPT-5 mini | GPT-5 nano | CLAUDE-Sonnet 4 | CLAUDE-Haiku 3.5 | GEMINI 2.5 Flash | GEMINI 2.5 Flash-lite | DeepSeek-V3.1 (chat) |
|---|---|---|---|---|---|---|---|---|
| ACCURACY | 0.83 | 0.81 | 0.78 | 0.75 | 0.66 | 0.87 | 0.69 | 0.75 |
| HUMAN_ACCURACY | 0.8 | 0.8 | 0.8 | 0.8 | 0.8 | 0.8 | 0.8 | 0.8 |

Figure 4.4: Accuracy of LLMs Compared to Human Baseline (Benchmark 1)

A more detailed analysis, distinguishing between theoretical questions (Accuracy_T) and numerical questions (Accuracy_N), highlights several relevant discrepancies. Some models present balanced values (GPT-5, GPT-5 mini, GPT-5 nano, Gemini-2.5 Flash), while others show greater heterogeneity, such as Claude-Sonnet 4 (0.86 T vs. 0.53 N), DeepSeek-v3.1 (0.85 T vs. 0.55 N), and Gemini Flash-Lite (0.84 T vs. 0.39 N). (Figure 4.5)




| | GPT-5 | GPT-5 mini | GPT-5 nano | CLAUDE-Sonnet 4 | CLAUDE-Haiku 3.5 | GEMINI 2.5 Flash | GEMINI 2.5 Flash-lite | DeepSeek-V3.1 (chat) |
|---|---|---|---|---|---|---|---|---|
| ACCURACY_T | 0.80 | 0.79 | 0.78 | 0.86 | 0.82 | 0.87 | 0.84 | 0.855 |
| ACCURACY_N | 0.89 | 0.86 | 0.79 | 0.53 | 0.33 | 0.89 | 0.39 | 0.550 |

Figure 4.5: Comparison of Theoretical vs. Numerical Accuracy (Benchmark 1)

Within the numerical subset, an additional analysis was performed by difficulty level (easy, medium, hard) (Figure 4.6). In the *easy* questions, GPT-5 (0.90), GPT-5 mini (0.86), and Gemini-2.5 Flash (0.94) achieved high values, while significantly lower performances were observed for Claude-Haiku 3.5 (0.36) and Gemini Flash-Lite (0.42). On *medium* questions, GPT-5 and Gemini Flash maintained high accuracy (0.90), with GPT-5 mini at 0.87. In the same category, several models showed significant difficulties, including Claude-Sonnet 4 (0.37), Claude-Haiku 3.5 (0.37), Gemini Flash-Lite (0.33), and DeepSeek-v3.1 (0.37). Finally, on *hard* questions, the highest accuracies were again achieved by GPT-5 (0.85) and GPT-5 mini (0.85), followed by Gemini-2.5 Flash (0.75). The lowest values were reported by Claude-Haiku 3.5 (0.20), DeepSeek-v3.1 (0.30), and Claude-Sonnet 4 (0.40).

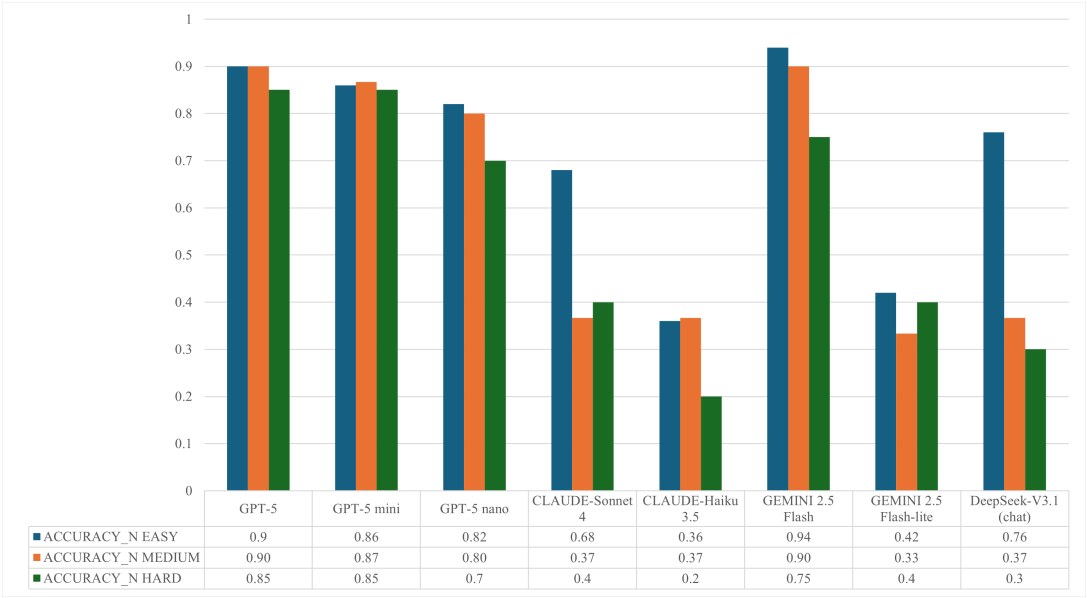



Figure 4.6: Comparison of Numerical Accuracy by Difficulty Level (Benchmark 1)

From the perspective of operational parameters, the results highlight notable differences. GPT-5 recorded the highest cost ($2.81) and latency (3981 s). GPT-5 mini and GPT-5 nano reported lower costs ($0.41 and $0.085) with similar latencies (3212 s and 3252 s). Gemini-2.5 Flash presented a cost of $0.035 and a latency of 2796 s, while Gemini-2.5 Flash-Lite stood out for its minimum cost ($0.004) and latency of 221 s. Claude-Sonnet 4 and Claude-Haiku 3.5 recorded $0.257 and $0.0458, with latencies of 811 s and 321 s, respectively. Finally, DeepSeek-v3.1 reported a cost of $0.022 and a latency of 1003 s. Table 4.2

| **LLM** | **Accuracy** | **Cost ($)** | **Latency (s)** |
|---|---|---|---|
| GPT-5 | 0.83 | 2.8103 | 3981.26 |
| GPT-5 mini | 0.81 | 0.4079 | 3212.85 |
| GPT-5 nano | 0.78 | 0.0852 | 3252.23 |
| Claude-Sonnet 4 | 0.75 | 0.2567 | 810.78 |
| Claude-Haiku 3.5 | 0.66 | 0.0458 | 320.64 |
| Gemini-2.5 Flash | 0.87 | 0.0351 | 2796.14 |
| Gemini-2.5 Flash-lite | 0.69 | 0.0042 | 221.07 |
| DeepSeek-v3.1 | 0.77 | 0.0220 | 1002.94 |

Table 4.2: Performance comparison of LLMs in terms of accuracy, cost, and latency

To integrate the different evaluation criteria, the Analytic Hierarchy Process (AHP) was applied, combining accuracy, cost, and latency into a single synthetic index. The final ranking

resulting from the AHP is reported in Table 4.3. The results place Gemini-2.5 Flash in first position, followed by DeepSeek-v3.1 and GPT-5. In the subsequent positions are GPT-5 mini, Gemini-2.5 Flash-Lite, Claude-Haiku 3.5, Claude-Sonnet 4, and finally GPT-5 nano.

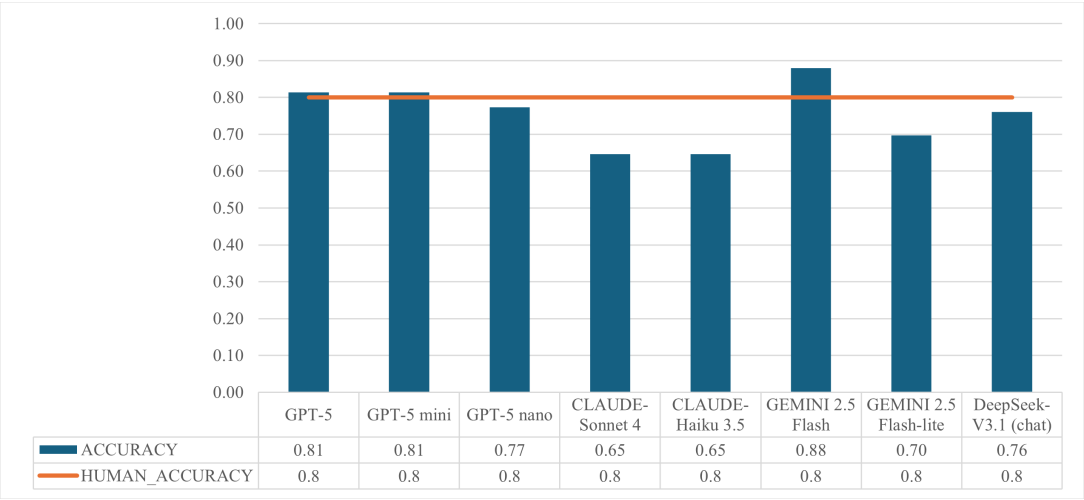
| Rank | Model |
|---|---|
| 1 | Gemini-2.5 Flash |
| 2 | DeepSeek-v3.1 |
| 3 | GPT-5 |
| 4 | GPT-5 mini |
| 5 | Gemini-2.5 Flash-Lite |
| 6 | Claude-Haiku 3.5 |
| 7 | Claude-Sonnet 4 |
| 8 | GPT-5 nano |

Table 4.3: Final ranking of models according to the AHP index (Benchmark 1).

### 4.3.2 Benchmark 2 Results

The second benchmark assessed model performance on single-choice questions with the addition of the Chain-of-Thought (CoT) technique, in order to examine the effect of reasoning on answer quality.

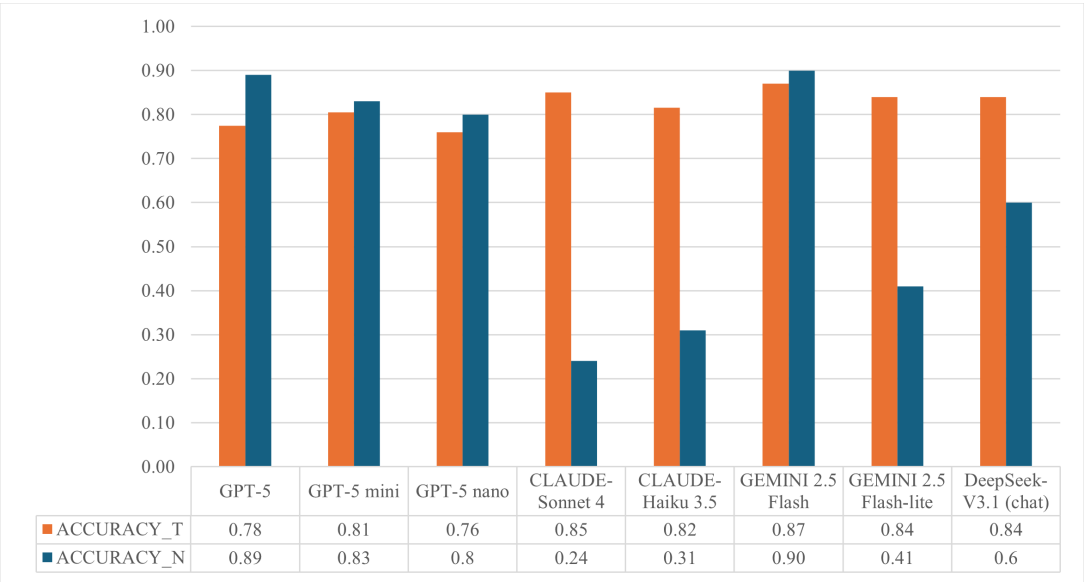
The comparison between model accuracy and human accuracy (set at 0.8, taken as the reference corresponding to the average performance of a fifth-year management engineering student) highlights notable differences (Figure X). GPT-5 (0.83), GPT-5 mini (0.81), and Gemini-2.5 Flash (0.87) exceed the human baseline. GPT-5 nano (0.78) and DeepSeek-v3.1 (0.77) are positioned very close to the human level, while Claude-Sonnet 4 (0.75), Claude-Haiku 3.5 (0.66), and Gemini-2.5 Flash-Lite (0.69) remain clearly below it. (Figure 4.7)




| | GPT-5 | GPT-5 mini | GPT-5 nano | CLAUDE-Sonnet 4 | CLAUDE-Haiku 3.5 | GEMINI 2.5 Flash | GEMINI 2.5 Flash-lite | DeepSeek-V3.1 (chat) |
|---|---|---|---|---|---|---|---|---|
| ACCURACY | 0.81 | 0.81 | 0.77 | 0.65 | 0.65 | 0.88 | 0.70 | 0.76 |
| HUMAN_ACCURACY | 0.8 | 0.8 | 0.8 | 0.8 | 0.8 | 0.8 | 0.8 | 0.8 |

Figure 4.7: Accuracy of LLMs Compared to Human Baseline (Benchmark 2)

The distinction between theoretical questions (Accuracy_T) and numerical questions (Accuracy_N) (Figure 4.6) highlights a certain variability. GPT-5 and Gemini-2.5 Flash show balanced performance (0.78 T – 0.89 N and 0.87 T – 0.90 N, respectively), while other models display marked discrepancies: Claude-Sonnet 4 records a theoretical value of 0.85 but a numerical value of 0.24, Claude-Haiku 3.5 scores 0.82 T and 0.31 N, Gemini Flash-Lite 0.84 T and 0.41 N, and DeepSeek-v3.1 0.84 T and 0.60 N.

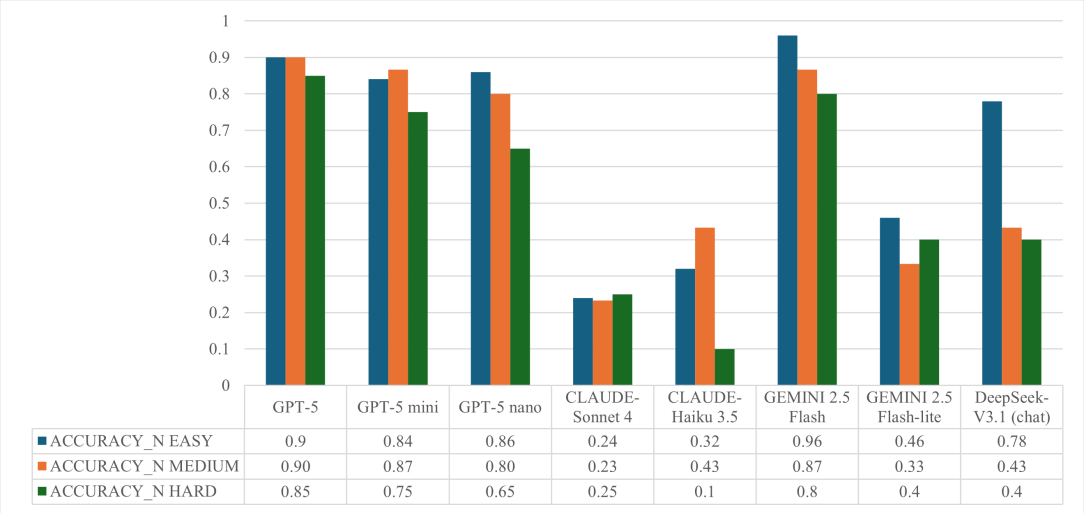



| | GPT-5 | GPT-5 mini | GPT-5 nano | CLAUDE-Sonnet 4 | CLAUDE-Haiku 3.5 | GEMINI 2.5 Flash | GEMINI 2.5 Flash-lite | DeepSeek-V3.1 (chat) |
|---|---|---|---|---|---|---|---|---|
| ACCURACY_T | 0.78 | 0.81 | 0.76 | 0.85 | 0.82 | 0.87 | 0.84 | 0.84 |
| ACCURACY_N | 0.89 | 0.83 | 0.8 | 0.24 | 0.31 | 0.90 | 0.41 | 0.6 |

Figure 4.8: Comparison of Theoretical vs. Numerical Accuracy (Benchmark 2)

For numerical questions divided by difficulty level (easy, medium, hard), the data reported in Figure 4.9 show clear differences. In easy questions, the highest values are achieved by Gemini-2.5 Flash (0.96), GPT-5 (0.90), and GPT-5 nano (0.86). On medium questions, scores

remain high for GPT-5 (0.90), GPT-5 mini (0.87), and Gemini-2.5 Flash (0.87), while models such as Claude-Sonnet 4 (0.23) and Gemini Flash-Lite (0.33) show notable difficulties. On hard questions, GPT-5 and GPT-5 mini (both 0.85) and Gemini-2.5 Flash (0.80) again perform best, while the lowest values are recorded for Claude-Haiku 3.5 (0.10) and Claude-Sonnet 4 (0.25).




| | GPT-5 | GPT-5 mini | GPT-5 nano | CLAUDE-Sonnet 4 | CLAUDE-Haiku 3.5 | GEMINI 2.5 Flash | GEMINI 2.5 Flash-lite | DeepSeek-V3.1 (chat) |
|---|---|---|---|---|---|---|---|---|
| ACCURACY_N EASY | 0.9 | 0.84 | 0.86 | 0.24 | 0.32 | 0.96 | 0.46 | 0.78 |
| ACCURACY_N MEDIUM | 0.90 | 0.87 | 0.80 | 0.23 | 0.43 | 0.87 | 0.33 | 0.43 |
| ACCURACY_N HARD | 0.85 | 0.75 | 0.65 | 0.25 | 0.1 | 0.8 | 0.4 | 0.4 |

Figure 4.9: Comparison of Numerical Accuracy by Difficulty Level (Benchmark 2)

From the perspective of operational parameters (Table 4.4), GPT-5 registered the highest cost ($2.92) and the longest latency (4066 s). GPT-5 mini and GPT-5 nano reported lower costs ($0.45 and $0.19) and latencies of 3497 s and 3010 s, respectively. Gemini-2.5 Flash reached $0.026 and 2598 s, while Gemini Flash-Lite showed the lowest values ($0.0047 and 179 s). Claude-Sonnet 4 and Claude-Haiku 3.5 recorded costs of $0.53 and $0.052, with latencies of 829 s and 314 s. DeepSeek-v3.1 reported $0.023 and 943 s.

| LLM | Accuracy | Cost ($) | Latency (s) |
|---|---|---|---|
| GPT-5 | 0.81 | 2.9218 | 4066.17 |
| GPT-5 mini | 0.81 | 0.4541 | 3496.77 |
| GPT-5 nano | 0.77 | 0.1872 | 3009.69 |
| Claude-Sonnet 4 | 0.65 | 0.5337 | 828.97 |
| Claude-Haiku 3.5 | 0.65 | 0.0517 | 313.55 |
| Gemini-2.5 Flash | 0.88 | 0.0264 | 2598.25 |
| Gemini-2.5 Flash-lite | 0.70 | 0.0047 | 178.63 |
| DeepSeek-v3.1 | 0.76 | 0.0230 | 942.63 |

Table 4.4: Performance comparison of LLMs in terms of accuracy, cost, and latency

Finally, the application of the Analytic Hierarchy Process (AHP) made it possible to synthesize accuracy, cost, and latency into an overall ranking (Table 4.5). The results place Gemini-2.5 Flash in first position, followed by Gemini-2.5 Flash-Lite and DeepSeek-v3.1. Subsequent positions are occupied by GPT-5 mini, GPT-5, Claude-Haiku 3.5, GPT-5 nano, and Claude-Sonnet 4.

| Rank | Model |
|---|---|
| 1 | Gemini-2.5 Flash |
| 2 | Gemini-2.5 Flash-Lite |
| 3 | DeepSeek-v3.1 |
| 4 | GPT-5 mini |
| 5 | GPT-5 |
| 6 | Claude-Haiku 3.5 |
| 7 | GPT-5 nano |
| 8 | Claude-Sonnet 4 |

Table 4.5: Final ranking of models according to the AHP index (Benchmark 2).

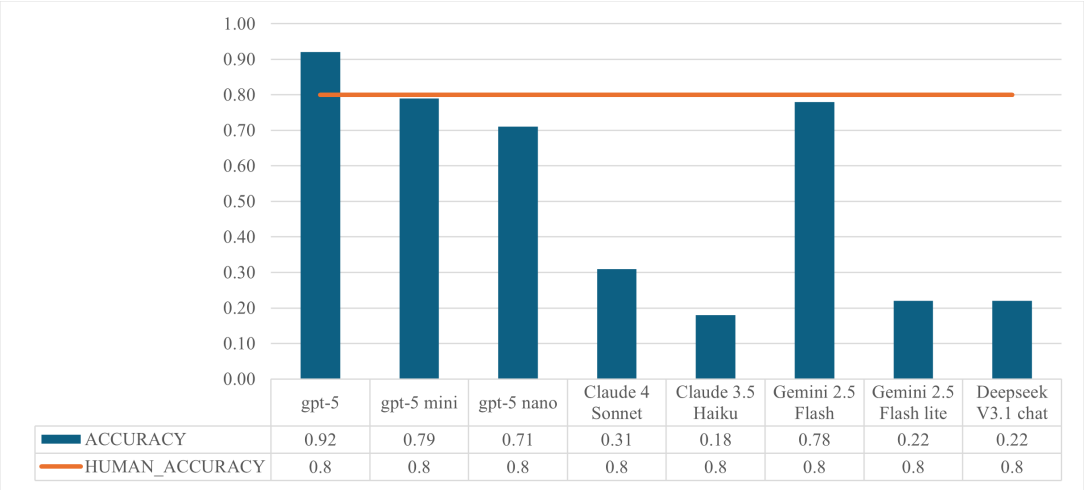
### 4.3.3 Benchmark 3 Results

The third benchmark evaluated model performance on numerical response questions, where no textual explanation was required and only the provision of a value was expected. This format allows for a direct assessment of the ability to perform calculations and return the correct

numerical result.

Overall accuracy (Figure 4.10) ranges from 0.92 (GPT-5) to 0.18 (Claude-Haiku 3.5). High results were achieved by GPT-5 mini (0.79), GPT-5 nano (0.71), and Gemini-2.5 Flash (0.78). Considerably lower values were observed for Claude-Sonnet 4 (0.31), Gemini Flash-Lite (0.22), and DeepSeek-v3.1 (0.22).

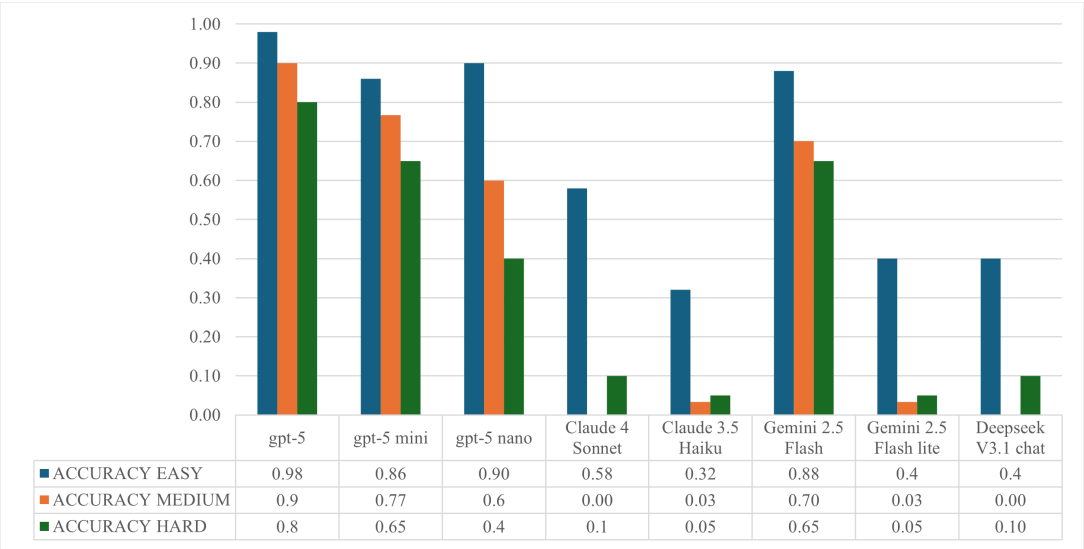
The comparison with human evaluation (threshold 0.8) shows that only GPT-5 (0.92) exceeds the reference level. All other models fall below the threshold: GPT-5 mini (0.79), GPT-5 nano (0.71), Gemini-2.5 Flash (0.78), Claude-Sonnet 4 (0.31), Claude-Haiku 3.5 (0.18), Gemini Flash-Lite (0.22), and DeepSeek-v3.1 (0.22).




| | gpt-5 | gpt-5 mini | gpt-5 nano | Claude 4 Sonnet | Claude 3.5 Haiku | Gemini 2.5 Flash | Gemini 2.5 Flash lite | Deepseek V3.1 chat |
|---|---|---|---|---|---|---|---|---|
| ACCURACY | 0.92 | 0.79 | 0.71 | 0.31 | 0.18 | 0.78 | 0.22 | 0.22 |
| HUMAN_ACCURACY | 0.8 | 0.8 | 0.8 | 0.8 | 0.8 | 0.8 | 0.8 | 0.8 |

Figure 4.10: Accuracy of LLMs Compared to Human Baseline (Benchmark 3)

The breakdown by difficulty level (easy, medium, hard) highlights substantial differences (Figure 4.11). On easy questions, very high accuracies are reported for GPT-5 (0.98), GPT-5 mini (0.86), GPT-5 nano (0.90), and Gemini-2.5 Flash (0.88). On medium questions, values remain strong for GPT-5 (0.90), GPT-5 mini (0.77), and Gemini-2.5 Flash (0.70), while models such as Claude-Sonnet 4 (0.00), Claude-Haiku 3.5 (0.03), Gemini Flash-Lite (0.03), and DeepSeek-v3.1 (0.00) show significant difficulties. On hard questions, GPT-5 (0.80), GPT-5 mini (0.65), and Gemini-2.5 Flash (0.65) again confirm superior performance compared to the other models, which remain at very low values.




| | gpt-5 | gpt-5 mini | gpt-5 nano | Claude 4 Sonnet | Claude 3.5 Haiku | Gemini 2.5 Flash | Gemini 2.5 Flash lite | Deepseek V3.1 chat |
|---|---|---|---|---|---|---|---|---|
| ACCURACY EASY | 0.98 | 0.86 | 0.90 | 0.58 | 0.32 | 0.88 | 0.4 | 0.4 |
| ACCURACY MEDIUM | 0.9 | 0.77 | 0.6 | 0.00 | 0.03 | 0.70 | 0.03 | 0.00 |
| ACCURACY HARD | 0.8 | 0.65 | 0.4 | 0.1 | 0.05 | 0.65 | 0.05 | 0.10 |

Figure 4.11: Comparison of Numerical Accuracy by Difficulty Level (Benchmark 3)

The operational parameters (Table 4.6) highlight substantial differences. GPT-5 shows the highest cost ($2.17) and the longest latency (2769 s). The GPT-5 mini and nano versions report lower costs ($0.24 and $0.11) with latencies of 1810 s and 1427 s. Gemini-2.5 Flash records a cost of $0.0077 and a latency of 960 s, while Gemini Flash-Lite reports the lowest cost ($0.0026) and a latency of 67 s. The Claude models present intermediate values: Sonnet 4 ($0.16 and 257 s) and Haiku 3.5 ($0.021 and 96 s). Finally, DeepSeek-v3.1 registers $0.012 and 345 s.

| **LLM** | **Accuracy** | **Cost ($)** | **Latency (s)** |
|---|---|---|---|
| GPT-5 | 0.92 | 2.1683 | 2768.99 |
| GPT-5 mini | 0.79 | 0.2432 | 1809.67 |
| GPT-5 nano | 0.71 | 0.1090 | 1427.19 |
| Claude-Sonnet 4 | 0.31 | 0.1608 | 257.05 |
| Claude-Haiku 3.5 | 0.18 | 0.0210 | 96.13 |
| Gemini-2.5 Flash | 0.78 | 0.007741 | 960.01 |
| Gemini-2.5 Flash-lite | 0.22 | 0.002561 | 67.39 |
| DeepSeek-v3.1 | 0.22 | 0.012414 | 344.89 |

Table 4.6: Performance comparison of LLMs in terms of accuracy, cost, and latency

The application of the Analytic Hierarchy Process (AHP) made it possible to synthesize the three evaluation criteria (accuracy, cost, and latency) into an overall ranking (Table 4.7). The

results place GPT-5 in first position, followed by Gemini-2.5 Flash and GPT-5 mini. The subsequent positions are occupied by GPT-5 nano, Gemini-2.5 Flash-Lite, Claude-Haiku 3.5, Claude-Sonnet 4, and DeepSeek-v3.1.

| Rank | Model |
|---|---|
| 1 | GPT-5 |
| 2 | Gemini-2.5 Flash |
| 3 | GPT-5 mini |
| 4 | GPT-5 nano |
| 5 | Gemini-2.5 Flash-Lite |
| 6 | Claude-Haiku 3.5 |
| 7 | Claude-Sonnet 4 |
| 8 | DeepSeek-v3.1 |

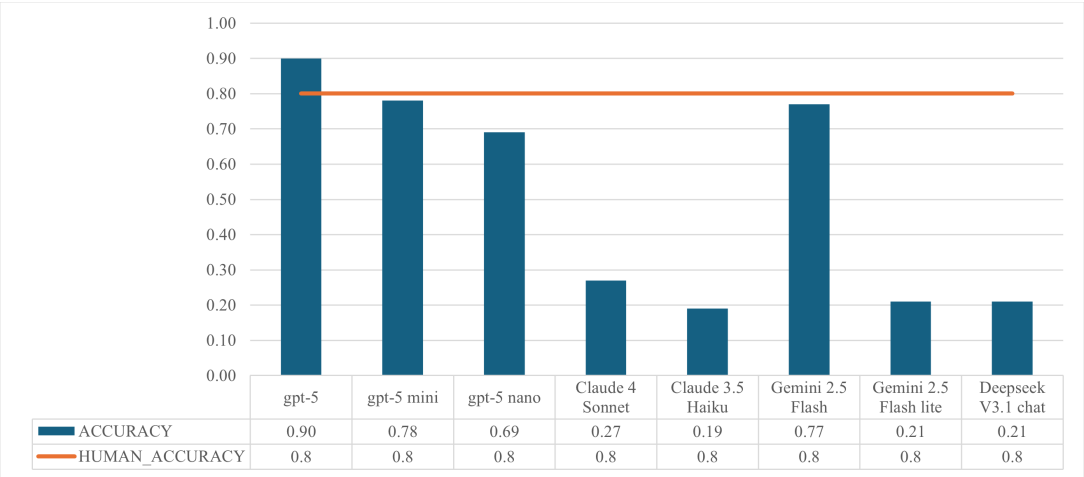
Table 4.7: Final ranking of models according to the AHP index (Benchmark 3).

### 4.3.4 Benchmark 4 Results

The fourth benchmark evaluated the performance of LLMs on numerical response questions using the chain-of-thought (CoT) technique. The goal was to assess whether the addition of implicit reasoning could improve accuracy.

Overall accuracy (Figure 4.12) ranges from a maximum of 0.90 (GPT-5) to a minimum of 0.19 (Claude-Haiku 3.5). GPT-5 mini reaches 0.78, while GPT-5 nano achieves 0.69. Gemini-2.5 Flash records 0.77, while Gemini Flash-Lite and DeepSeek-v3.1 remain at 0.21. Claude-Sonnet 4 shows an intermediate value of 0.27.

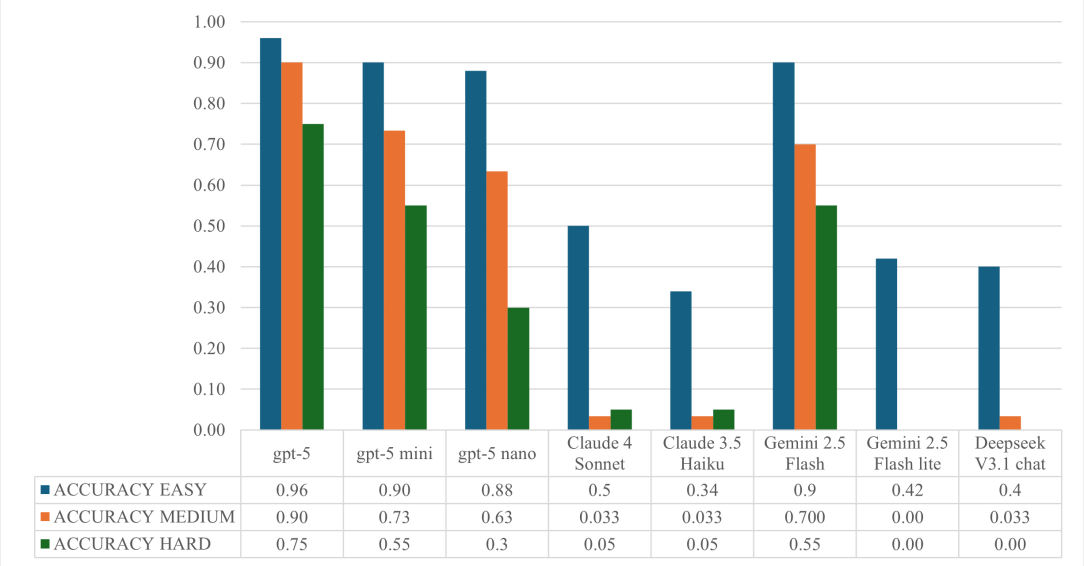
Comparison with human evaluation (threshold 0.8) highlights that only GPT-5 (0.90) exceeds the reference level. All other models remain below: GPT-5 mini (0.78), GPT-5 nano (0.69), Gemini-2.5 Flash (0.77), Claude-Sonnet 4 (0.27), Claude-Haiku 3.5 (0.19), Gemini Flash-Lite (0.21), and DeepSeek-v3.1 (0.21).




| | gpt-5 | gpt-5 mini | gpt-5 nano | Claude 4 Sonnet | Claude 3.5 Haiku | Gemini 2.5 Flash | Gemini 2.5 Flash lite | Deepseek V3.1 chat |
|---|---|---|---|---|---|---|---|---|
| ACCURACY | 0.90 | 0.78 | 0.69 | 0.27 | 0.19 | 0.77 | 0.21 | 0.21 |
| HUMAN_ACCURACY | 0.8 | 0.8 | 0.8 | 0.8 | 0.8 | 0.8 | 0.8 | 0.8 |

Figure 4.12: Accuracy of LLMs Compared to Human Baseline (Benchmark 4)

The breakdown by difficulty level (easy, medium, hard) shows differentiated patterns (Figure 4.13). In easy questions, high values are observed for GPT-5 (0.96), GPT-5 mini (0.90), GPT-5 nano (0.88) and Gemini-2.5 Flash (0.90). In medium questions, the accuracy remains high for GPT-5 (0.90) and GPT-5 mini (0.73), but is lower for GPT-5 nano (0.63) and Gemini Flash (0.70). In hard questions, GPT-5 maintains relatively high values (0.75), followed by GPT-5 mini (0.55) and Gemini Flash (0.55), while the other models show very low performance, in some cases close to zero (Gemini Flash-Lite and DeepSeek).




| | gpt-5 | gpt-5 mini | gpt-5 nano | Claude 4 Sonnet | Claude 3.5 Haiku | Gemini 2.5 Flash | Gemini 2.5 Flash lite | Deepseek V3.1 chat |
|---|---|---|---|---|---|---|---|---|
| ACCURACY EASY | 0.96 | 0.90 | 0.88 | 0.5 | 0.34 | 0.9 | 0.42 | 0.4 |
| ACCURACY MEDIUM | 0.90 | 0.73 | 0.63 | 0.033 | 0.033 | 0.700 | 0.00 | 0.033 |
| ACCURACY HARD | 0.75 | 0.55 | 0.3 | 0.05 | 0.05 | 0.55 | 0.00 | 0.00 |

Figure 4.13: Comparison of Numerical Accuracy by Difficulty Level (Benchmark 4)

The operational parameters (Table 4.8) show significant differences. GPT-5 recorded the highest cost ($2.16) and latency (2714 s). GPT-5 mini and GPT-5 nano reported lower costs ($0.25 and $0.12) with latencies of 1868 s and 1805 s. Gemini-2.5 Flash showed a low cost ($0.0079)

and latency of 1151 s, while Gemini Flash-Lite registered the lowest values ($0.0026 and 78 s). Claude-Sonnet 4 and Claude-Haiku 3.5 reported costs of $0.29 and $0.0226, with latencies of 443 s and 97 s, respectively. DeepSeek-v3.1 was placed at $0.013 and 348 s.

| **LLM** | **Accuracy** | **Cost ($)** | **Latency (s)** |
|---|---|---|---|
| GPT-5 | 0.90 | 2.1589 | 2713.86 |
| GPT-5 mini | 0.78 | 0.2496 | 1868.40 |
| GPT-5 nano | 0.69 | 0.1159 | 1804.93 |
| Claude-Sonnet 4 | 0.27 | 0.2943 | 442.89 |
| Claude-Haiku 3.5 | 0.19 | 0.0226 | 97.36 |
| Gemini-2.5 Flash | 0.77 | 0.007929 | 1150.67 |
| Gemini-2.5 Flash-lite | 0.21 | 0.002633 | 77.79 |
| DeepSeek-v3.1 | 0.21 | 0.013357 | 347.72 |

Table 4.8: Performance comparison of LLMs in terms of accuracy, cost, and latency

The application of the Analytic Hierarchy Process (AHP) made it possible to integrate accuracy, cost, and latency into a synthetic ranking (Table **??**). The results place GPT-5 in the first position, followed by Gemini-2.5 Flash and GPT-5 mini. Subsequent positions are occupied by GPT-5 nano, Gemini-2.5 Flash-Lite, Claude-Haiku 3.5, DeepSeek-v3.1, and Claude-Sonnet 4.

| **Rank** | **Model** |
|---|---|
| 1 | GPT-5 |
| 2 | Gemini-2.5 Flash |
| 3 | GPT-5 mini |
| 4 | GPT-5 nano |
| 5 | Gemini-2.5 Flash-Lite |
| 6 | Claude-Haiku 3.5 |
| 7 | DeepSeek-v3.1 |
| 8 | Claude-Sonnet 4 |

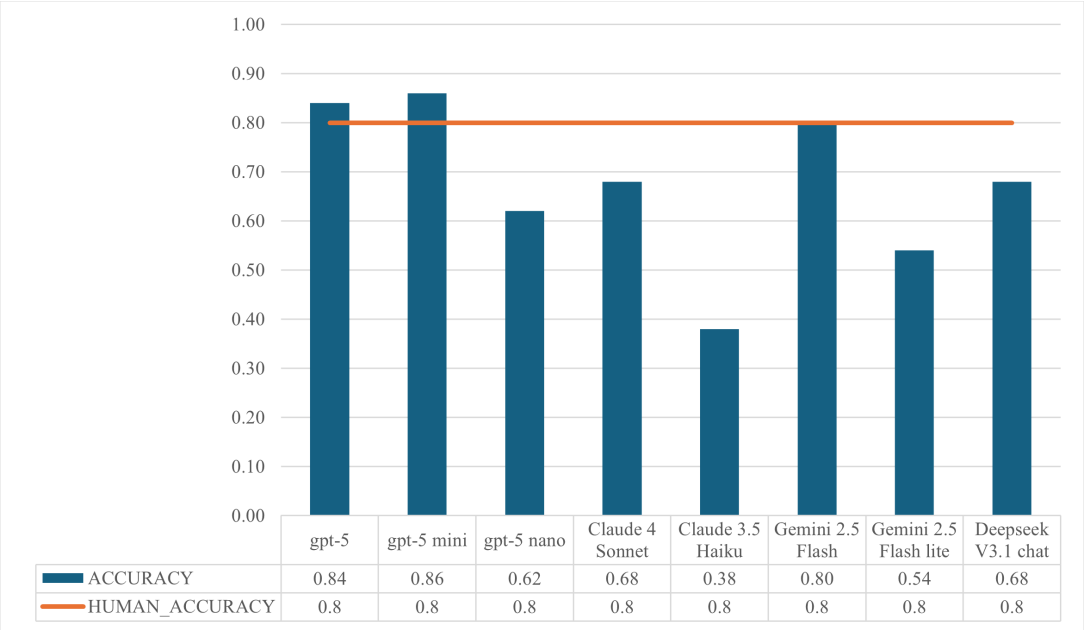
Table 4.9: Final ranking of models according to the AHP index (Benchmark 4).

### 4.3.5 Benchmark 5 Results

The fifth benchmark analyzed the performance of LLMs on numerical response questions requiring explicit reasoning. In addition to providing the numerical result, the models also produced detailed explanations of the process, which were evaluated in terms of accuracy (considering correctness only), calculation ability, reasoning coherence, and type of reasoning error.

Taking into account only the correctness of the numerical value, the results shown in Figure 4.14 range from a maximum of 0.86 (GPT-5 mini) to a minimum of 0.38 (Claude-Haiku 3.5). GPT-5 reaches 0.84, Gemini-2.5 Flash 0.80. Claude-Sonnet 4 and DeepSeek-v3.1 report 0.68, while Gemini Flash-Lite achieves 0.54 and GPT-5 nano 0.62.

Comparison with human evaluation (threshold 0.8) indicates that GPT-5 (0.84), GPT-5 mini (0.86) and Gemini-2.5 Flash (0.80) exceed this level. All other models fall below the reference threshold.




| | gpt-5 | gpt-5 mini | gpt-5 nano | Claude 4 Sonnet | Claude 3.5 Haiku | Gemini 2.5 Flash | Gemini 2.5 Flash lite | Deepseek V3.1 chat |
|---|---|---|---|---|---|---|---|---|
| ACCURACY | 0.84 | 0.86 | 0.62 | 0.68 | 0.38 | 0.80 | 0.54 | 0.68 |
| HUMAN_ACCURACY | 0.8 | 0.8 | 0.8 | 0.8 | 0.8 | 0.8 | 0.8 | 0.8 |

Figure 4.14: Accuracy of LLMs Compared to Human Baseline (Benchmark 5)

The breakdown by difficulty level (medium and hard) shows different patterns (Figure 4.15). In medium questions, GPT-5 mini achieves the highest value (0.93), followed by GPT-5 (0.90) and Gemini-2.5 Flash (0.83). In hard questions, the highest values are observed for GPT-5, GPT-5 mini, and Gemini-2.5 Flash (0.75 each), while the lowest scores are recorded for Claude-Haiku 3.5 (0.25) and GPT-5 nano (0.45).

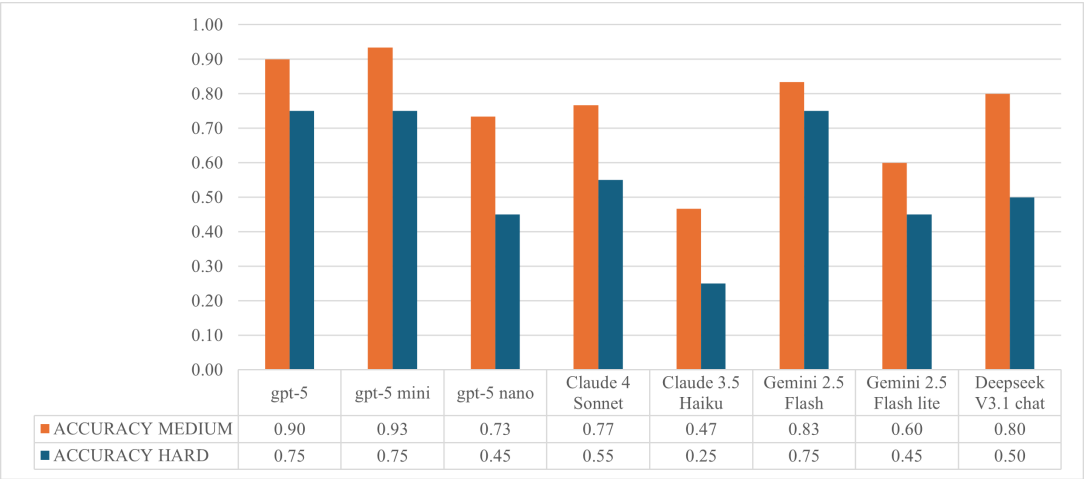



Figure 4.15: Comparison of Numerical Accuracy by Difficulty Level (Benchmark 5)

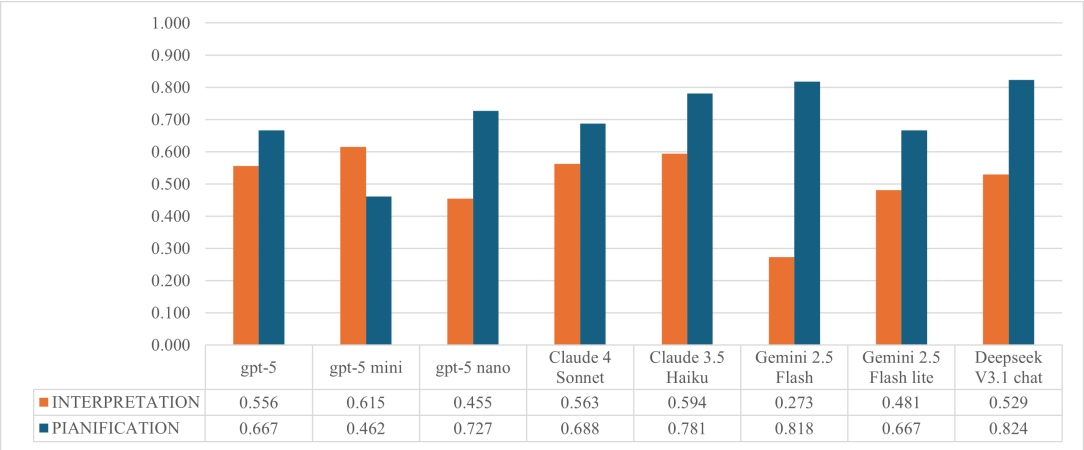
The analysis of the calculation and reasoning components (Table 4.10) shows that the calculation ability is generally high across all models, with values ranging from 0.81 (Claude-Haiku 3.5) to 1.00 (GPT-5 mini and DeepSeek-v3.1). Greater variability is observed in reasoning scores, which range from 0.896 (Gemini-2.5 Flash) to 0.515 (Claude-Haiku 3.5). GPT-5 and GPT-5 mini report values of 0.867 and 0.875, respectively, while Claude-Sonnet 4 reaches 0.805.

| **LLM** | **Calculation** | **Reasoning** |
|---|---|---|
| GPT-5 | 0.99 | 0.867 |
| GPT-5 mini | 1.00 | 0.875 |
| GPT-5 nano | 0.94 | 0.744 |
| Claude-Sonnet 4 | 0.94 | 0.805 |
| Claude-Haiku 3.5 | 0.81 | 0.515 |
| Gemini-2.5 Flash | 0.93 | 0.896 |
| Gemini-2.5 Flash-Lite | 0.87 | 0.628 |
| DeepSeek-v3.1 | 1.00 | 0.770 |

Table 4.10: Calculation and reasoning scores for LLMs (Benchmark 5).

For models with reasoning scores below 0.4, a qualitative analysis of error types was conducted, distinguishing between interpretation and pianification errors (Figure 4.16). The values show that planning errors tend to be more frequent than interpretation errors. For example, DeepSeek-v3.1 records 0.824 for planning and 0.529 for interpretation, while Gemini-2.5 Flash

reaches 0.818 and 0.273, respectively. GPT-5 mini scores 0.615 for interpretation and 0.462 for planning, while Claude-Haiku 3.5 reports 0.594 and 0.781.




| | gpt-5 | gpt-5 mini | gpt-5 nano | Claude 4 Sonnet | Claude 3.5 Haiku | Gemini 2.5 Flash | Gemini 2.5 Flash lite | Deepseek V3.1 chat |
|---|---|---|---|---|---|---|---|---|
| INTERPRETATION | 0.556 | 0.615 | 0.455 | 0.563 | 0.594 | 0.273 | 0.481 | 0.529 |
| PIANIFICATION | 0.667 | 0.462 | 0.727 | 0.688 | 0.781 | 0.818 | 0.667 | 0.824 |

Figure 4.16: Comparison of Reasoning Errors: Interpretation vs. Pianification (Benchmark 5)

Finally, operational parameters (Table 4.11) reveal relevant differences among models. GPT-5 presents the highest cost ($1.76) and latency (2531 s). GPT-5 mini records $0.26 and 1787 s, while GPT-5 nano reports $0.10 and 1448 s. Gemini-2.5 Flash records $0.10 and 947 s, and Gemini Flash-Lite the lowest values ($0.039 and 324 s). Claude-Sonnet 4 and Claude-Haiku 3.5 report $0.47 and $0.075 with latencies of 510 s and 273 s, respectively. DeepSeek-v3.1 records $0.093 and 2662 s.

| **LLM** | **Accuracy** | **Cost ($)** | **Latency (s)** |
|---|---|---|---|
| GPT-5 | 0.84 | 1.7635 | 2530.80 |
| GPT-5 mini | 0.86 | 0.2628 | 1786.95 |
| GPT-5 nano | 0.62 | 0.1043 | 1448.18 |
| Claude-Sonnet 4 | 0.68 | 0.4745 | 509.87 |
| Claude-Haiku 3.5 | 0.38 | 0.0755 | 272.56 |
| Gemini-2.5 Flash | 0.80 | 0.102135 | 947.36 |
| Gemini-2.5 Flash-lite | 0.54 | 0.038566 | 324.11 |
| DeepSeek-v3.1 | 0.68 | 0.092539 | 2662.32 |

Table 4.11: Performance comparison of LLMs in terms of accuracy, cost, and latency

The application of the Analytic Hierarchy Process (AHP) made it possible to synthesize model results into a single ranking by integrating accuracy, cost, and latency (Table 4.12). The final ranking places GPT-5 mini in first position, followed by Gemini-2.5 Flash and GPT-5.

The intermediate positions are occupied by Claude-Sonnet 4 and Gemini-2.5 Flash-Lite, while Claude-Haiku 3.5 ranks sixth. GPT-5 nano and DeepSeek-v3.1 close the ranking.

| Rank | Model |
|---|---|
| 1 | GPT-5 mini |
| 2 | Gemini-2.5 Flash |
| 3 | GPT-5 |
| 4 | Claude-Sonnet 4 |
| 5 | Gemini-2.5 Flash-Lite |
| 6 | Claude-Haiku 3.5 |
| 7 | GPT-5 nano |
| 8 | DeepSeek-v3.1 |

Table 4.12: Final ranking of models according to the AHP index (Benchmark 5).

## 4.4 Cross-benchmark Comparison

After the analysis of the individual benchmarks, this section adopts a comparative perspective, relating the results obtained in the different experimental scenarios. The comparisons make it possible to assess to what extent changes in question format or the introduction of prompting techniques have influenced LLM performance.

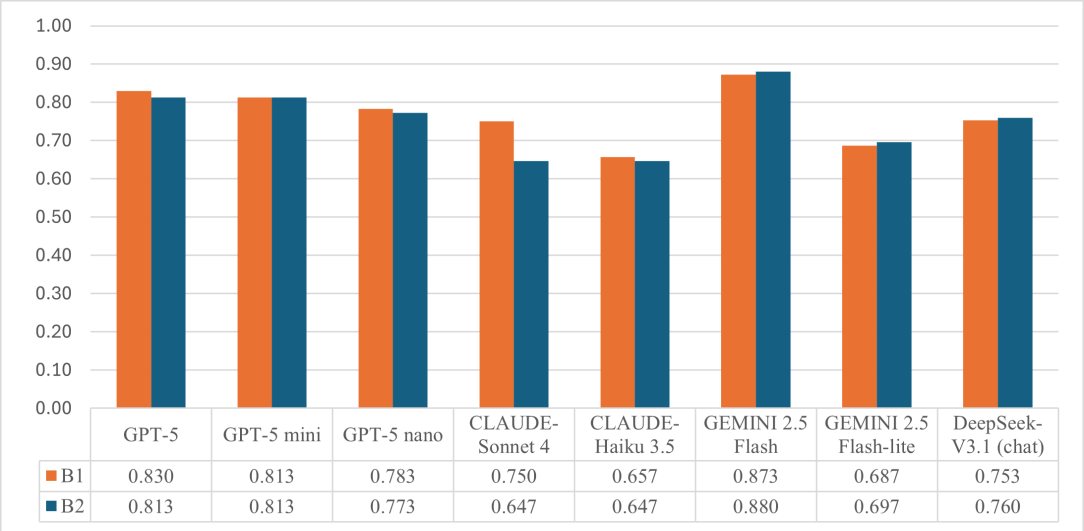
The analysis proceeds through direct comparisons between pairs of benchmarks, in order to isolate the effect of specific experimental variables, such as the use of Chain-of-Thought or the shift from single-choice to numerical response questions. For each comparison, the observed variations in accuracy, cost, and latency are reported, with the aim of highlighting recurring patterns or systematic differences between configurations.

Beyond the descriptive level, the comparisons were subjected to a statistical significance analysis using the McNemar test, applied to verify whether the differences between two benchmarks can be considered statistically relevant at the 95% confidence level. In this way, the conclusions are based not only on numerical variations but also on their inferential robustness, reducing the risk of overinterpreting marginal differences.

### 4.4.1 Benchmark 1 vs Benchmark 2

The comparison between Benchmark 1 (without Chain-of-Thought) and Benchmark 2 (with Chain-of-Thought) makes it possible to assess the impact of introducing implicit reasoning on single-choice questions.

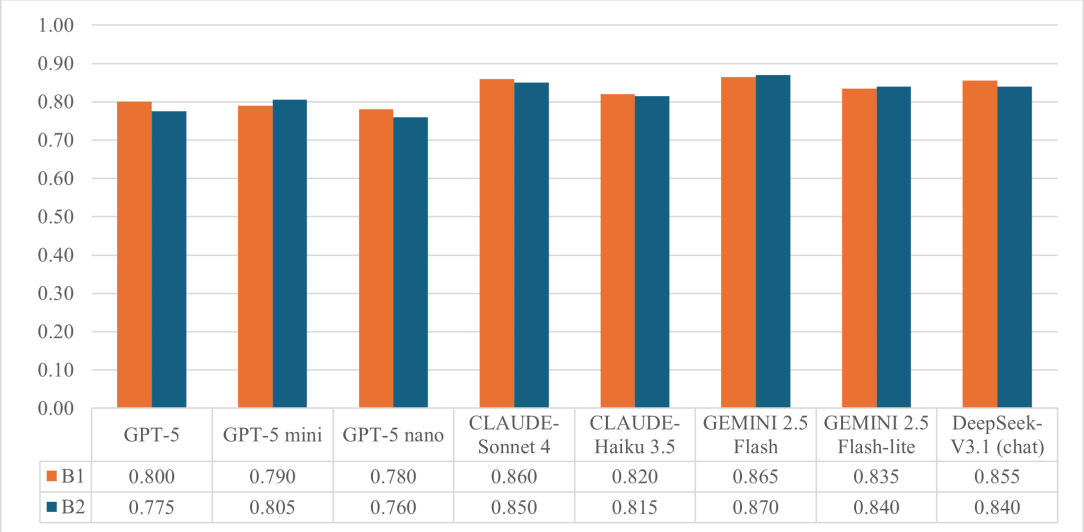
In terms of overall accuracy, the addition of implicit CoT in Benchmark 2 caused a decrease in performance for four out of eight models (Figure 4.17). The most notable reductions were observed for Claude-Sonnet 4 (–10 pp), followed by GPT-5 (–2 pp), GPT-5 nano (–1 pp), and Claude-Haiku 3.5 (–1 pp). Conversely, the Gemini models and DeepSeek-v3.1 recorded a slight improvement of +1 pp, while GPT-5 mini remained unchanged (0 pp).




| | GPT-5 | GPT-5 mini | GPT-5 nano | CLAUDE-Sonnet 4 | CLAUDE-Haiku 3.5 | GEMINI 2.5 Flash | GEMINI 2.5 Flash-lite | DeepSeek-V3.1 (chat) |
|---|---|---|---|---|---|---|---|---|
| B1 | 0.830 | 0.813 | 0.783 | 0.750 | 0.657 | 0.873 | 0.687 | 0.753 |
| B2 | 0.813 | 0.813 | 0.773 | 0.647 | 0.647 | 0.880 | 0.697 | 0.760 |

Figure 4.17: Overall Accuracy Comparison between Benchmark 1 and Benchmark 2

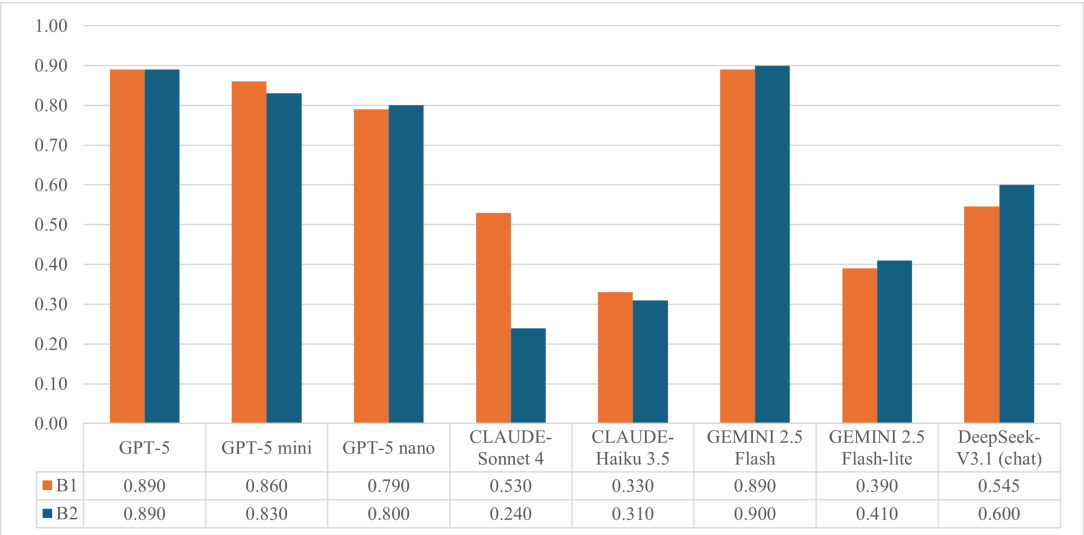
When single-choice questions are split into theoretical and numerical (Figure 4.18 ), heterogeneous trends emerge. For theoretical questions, several models show decreases: GPT-5 (–3 pp), GPT-5 nano (- 2 pp), DeepSeek-v3.1 (- 2 pp) and both Claude models (–1 pp). In contrast, GPT-5 mini improves by +2 pp and the Gemini models by +1 pp. The impact of implicit CoT on this subset therefore appears marginal, with changes contained within ±3 pp.




| | GPT-5 | GPT-5 mini | GPT-5 nano | CLAUDE-Sonnet 4 | CLAUDE-Haiku 3.5 | GEMINI 2.5 Flash | GEMINI 2.5 Flash-lite | DeepSeek-V3.1 (chat) |
|---|---|---|---|---|---|---|---|---|
| B1 | 0.800 | 0.790 | 0.780 | 0.860 | 0.820 | 0.865 | 0.835 | 0.855 |
| B2 | 0.775 | 0.805 | 0.760 | 0.850 | 0.815 | 0.870 | 0.840 | 0.840 |

Figure 4.18: Theoretical Accuracy Comparison between Benchmark 1 and Benchmark 2

A more varied picture emerges for numerical questions (Figure 4.19 ). Here, GPT-5 remains stable (0 pp), GPT-5 nano increases by +1 pp, and DeepSeek-v3.1 shows a more substantial improvement (+5 pp). The Gemini models also gain slightly (+1 pp and +2 pp). By contrast, the Anthropic models show clear difficulties: Claude-Sonnet 4 experiences a sharp drop (–29 pp), Claude-Haiku 3.5 declines by –2 pp, and GPT-5 mini loses –3 pp.




| | GPT-5 | GPT-5 mini | GPT-5 nano | CLAUDE-Sonnet 4 | CLAUDE-Haiku 3.5 | GEMINI 2.5 Flash | GEMINI 2.5 Flash-lite | DeepSeek-V3.1 (chat) |
|---|---|---|---|---|---|---|---|---|
| B1 | 0.890 | 0.860 | 0.790 | 0.530 | 0.330 | 0.890 | 0.390 | 0.545 |
| B2 | 0.890 | 0.830 | 0.800 | 0.240 | 0.310 | 0.900 | 0.410 | 0.600 |

Figure 4.19: Numerical Accuracy Comparison between Benchmark 1 and Benchmark 2

From the perspective of statistical significance, the comparison between Benchmark 1 and Benchmark 2 was verified using the ***McNemar*** test, applied both to the overall sample and to the subgroups by type of question (theoretical and numerical). As shown in Table 4.24 (Overall Accuracy), for nearly all models the observed differences do not reach the conventional 95%

confidence level ($p > 0.05$). The only exception is Claude-Sonnet 4, which displays a statistically significant reduction in accuracy ($p = 0.0002$). A similar result emerges in the numerical subset (Table 4.15), where Claude-Sonnet 4 again shows a significant decline ($p = 0.0001$). For theoretical questions (Table 4.14), no significant differences are found between the two benchmarks.

| **LLM** | **B1** | **B2** | **Sample** | **Chi-square** | **p-value** | **Exact p-value** | **Significance (95%)** |
|---|---|---|---|---|---|---|---|
| GPT-5 | 0.830 | 0.813 | 300 | 1.19 | 0.2752 | 0.3833 | NO |
| GPT-5 mini | 0.813 | 0.813 | 300 | 0.00 | 1.0000 | 1.0000 | NO |
| GPT-5 nano | 0.783 | 0.773 | 300 | 0.21 | 0.6473 | 0.7608 | NO |
| Claude-Sonnet 4 | 0.750 | 0.647 | 300 | 14.34 | 0.0002 | 0.0002 | YES |
| Claude-Haiku 3.5 | 0.657 | 0.647 | 300 | 0.33 | 0.5637 | 0.7011 | NO |
| Gemini-2.5 Flash | 0.873 | 0.880 | 300 | 0.25 | 0.6171 | 0.8036 | NO |
| Gemini-2.5 Flash-lite | 0.687 | 0.697 | 300 | 0.31 | 0.5775 | 0.7111 | NO |
| DeepSeek-v3.1 | 0.753 | 0.760 | 300 | 0.29 | 0.5930 | 0.7905 | NO |

Table 4.13: Overall Accuracy: comparison between Benchmark 1 and Benchmark 3 with statistical significance test results

| **LLM** | **B1** | **B2** | **Sample** | **Chi-square** | **p-value** | **Exact p-value** | **Significance (95%)** |
|---|---|---|---|---|---|---|---|
| GPT-5 | 0.800 | 0.775 | 200 | 1.92 | 0.1655 | 0.2668 | NO |
| GPT-5 mini | 0.790 | 0.805 | 200 | 0.69 | 0.4054 | 0.5811 | NO |
| GPT-5 nano | 0.780 | 0.760 | 200 | 0.62 | 0.4328 | 0.5572 | NO |
| Claude-Sonnet 4 | 0.860 | 0.850 | 200 | 0.33 | 0.5637 | 0.7744 | NO |
| Claude-Haiku 3.5 | 0.820 | 0.815 | 200 | 0.11 | 0.7389 | 1.0000 | NO |
| Gemini-2.5 Flash | 0.865 | 0.870 | 200 | 0.11 | 0.7389 | 1.0000 | NO |
| Gemini-2.5 Flash-lite | 0.835 | 0.840 | 200 | 0.11 | 0.7389 | 1.0000 | NO |
| DeepSeek-v3.1 | 0.855 | 0.840 | 200 | 1.29 | 0.2568 | 0.4531 | NO |

Table 4.14: Theoretical Accuracy: comparison between Benchmark 1 and Benchmark 3 with statistical significance test results

| LLM | B1 | B2 | Sample | Chi-square | p-value | Exact p-value | Significance (95%) |
|---|---|---|---|---|---|---|---|
| GPT-5 | 0.890 | 0.890 | 100 | 0.00 | 1.0000 | 1.0000 | NO |
| GPT-5 mini | 0.860 | 0.830 | 100 | 1.00 | 0.3173 | 0.5078 | NO |
| GPT-5 nano | 0.790 | 0.800 | 100 | 0.06 | 0.8084 | 1.0000 | NO |
| Claude-Sonnet 4 | 0.530 | 0.240 | 100 | 15.29 | 0.0001 | 0.0001 | YES |
| Claude-Haiku 3.5 | 0.330 | 0.310 | 100 | 0.22 | 0.6374 | 0.8145 | NO |
| Gemini-2.5 Flash | 0.890 | 0.900 | 100 | 0.14 | 0.7055 | 1.0000 | NO |
| Gemini-2.5 Flash-lite | 0.390 | 0.410 | 100 | 0.20 | 0.6547 | 0.8238 | NO |
| DeepSeek-v3.1 | 0.545 | 0.600 | 100 | 3.57 | 0.0588 | 0.1250 | NO |

Table 4.15: Accuracy Numerical: comparison between Benchmark 1 and Benchmark 3 with statistical significance test results

With respect to costs (Table 4.16), the introduction of CoT leads to increases for most models. GPT-5 increases by 3. 9%, GPT-5 mini by 11.3%, and GPT-5 nano more than doubles (+119%). The Anthropic models also show marked growth: Claude-Sonnet 4 +108% and Claude-Haiku 3.5 +12.9%. By contrast, Gemini-2.5 Flash reduces its costs by around 25%, while Gemini Flash-Lite (+11.9%) and DeepSeek-v3.1 (+4.6%) record smaller increases.

| LLM | B1 ($) | B2 ($) | Variation (%) |
|---|---|---|---|
| GPT-5 | 2.810 | 2.922 | 3.97% |
| GPT-5 mini | 0.408 | 0.454 | 11.33% |
| GPT-5 nano | 0.085 | 0.187 | 119.72% |
| Claude-Sonnet 4 | 0.257 | 0.534 | 107.91% |
| Claude-Haiku 3.5 | 0.046 | 0.052 | 12.88% |
| Gemini-2.5 Flash | 0.035 | 0.026 | -24.79% |
| Gemini-2.5 Flash-lite | 0.004 | 0.005 | 11.90% |
| DeepSeek-v3.1 | 0.022 | 0.023 | 4.55% |

Table 4.16: Cost comparison between Benchmark 1 and Benchmark 2 with percentage variation

Latency (Table 4.17) shows more contained variations compared to costs. GPT-5 increases slightly (+2%), as do GPT-5 mini (+8.8%) and Claude-Sonnet 4 (+2.2%). In contrast, GPT-5 nano (-7.5%) and Claude-Haiku 3.5 (–2.2%) achieve small reductions. Gemini-2.5 Flash (–7%) and Gemini Flash-Lite (–19%) show more notable improvements, while DeepSeek-v3.1 reduces latency by about 6%.

| LLM | B1 (s) | B2 (s) | Variation (%) |
|---|---|---|---|
| GPT-5 | 3981.26 | 4066.17 | 2.13% |
| GPT-5 mini | 3212.85 | 3496.77 | 8.84% |
| GPT-5 nano | 3252.23 | 3009.69 | -7.46% |
| Claude-Sonnet 4 | 810.78 | 828.97 | 2.24% |
| Claude-Haiku 3.5 | 320.64 | 313.55 | -2.21% |
| Gemini-2.5 Flash | 2796.14 | 2598.25 | -7.08% |
| Gemini-2.5 Flash-lite | 221.07 | 178.63 | -19.20% |
| DeepSeek-v3.1 | 1002.94 | 942.63 | -6.01% |

Table 4.17: Latency comparison between Benchmark 1 and Benchmark 2 with percentage variation

### 4.4.2 Benchmark 3 vs Benchmark 4

The comparison between Benchmark 3 (numerical answer) and Benchmark 4 (numerical answer with implicit Chain-of-Thought) makes it possible to assess the impact of introducing implicit reasoning in numerical response tasks.

In terms of overall accuracy, the results show modest variations between –4 and +1 pp. GPT-5 and GPT-5 nano both decrease by –2 pp, GPT-5 mini by –1 pp, while Claude-Sonnet 4 drops by –4 pp. Claude-Haiku 3.5 records a slight improvement (+1 pp), whereas the Gemini models and DeepSeek-v3.1 each decline by one percentage point. (Figure 4.20)

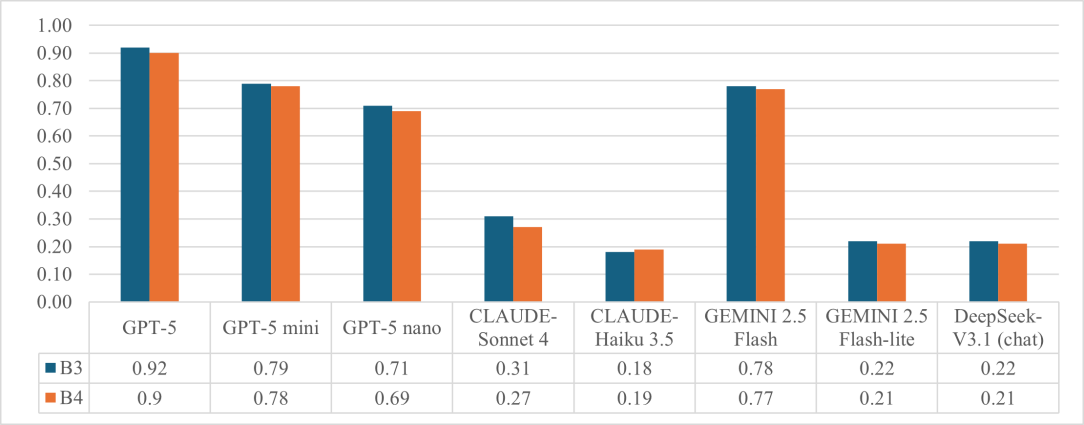
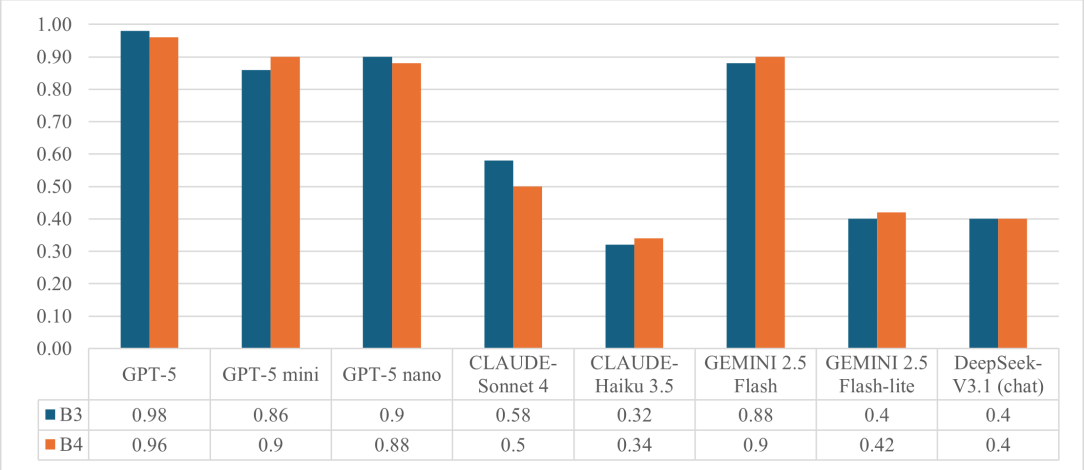



Figure 4.20: Overall Accuracy Comparison between Benchmark 3 and Benchmark 4

For easy questions (Figure 4.21), four models improve their performance: GPT-5 mini (+4 pp), Claude-Haiku 3.5 (+2 pp), and the two Gemini models (+2 pp). GPT-5 and GPT-5 nano both

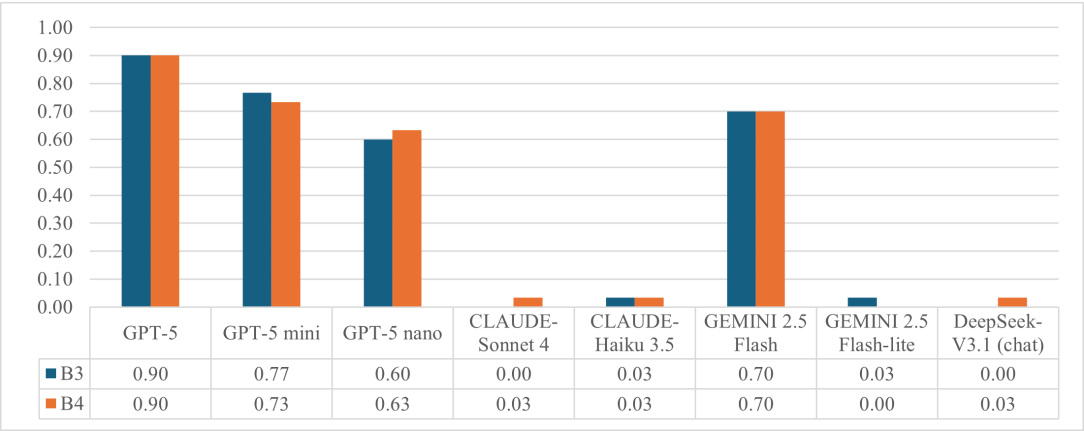
lose 2 pp, while Claude-Sonnet 4 shows a marked decline of –8 pp. DeepSeek-v3.1 remains unchanged. Overall, the effect of implicit CoT on easy questions appears limited and not systematic.




| | GPT-5 | GPT-5 mini | GPT-5 nano | CLAUDE-Sonnet 4 | CLAUDE-Haiku 3.5 | GEMINI 2.5 Flash | GEMINI 2.5 Flash-lite | DeepSeek-V3.1 (chat) |
|---|---|---|---|---|---|---|---|---|
| B3 | 0.98 | 0.86 | 0.9 | 0.58 | 0.32 | 0.88 | 0.4 | 0.4 |
| B4 | 0.96 | 0.9 | 0.88 | 0.5 | 0.34 | 0.9 | 0.42 | 0.4 |

Figure 4.21: Easy-Level Accuracy Comparison between Benchmark 3 and Benchmark 4

For medium questions (Figure 4.22), the picture is more mixed. GPT-5, Claude-Haiku 3.5, and Gemini-2.5 Flash remain unchanged, while GPT-5 nano, Claude-Sonnet 4, and DeepSeek-v3.1 improve by +3 pp each. Conversely, GPT-5 mini and Gemini-2.5 Flash-Lite both decrease by –3 pp.




| | GPT-5 | GPT-5 mini | GPT-5 nano | CLAUDE-Sonnet 4 | CLAUDE-Haiku 3.5 | GEMINI 2.5 Flash | GEMINI 2.5 Flash-lite | DeepSeek-V3.1 (chat) |
|---|---|---|---|---|---|---|---|---|
| B3 | 0.90 | 0.77 | 0.60 | 0.00 | 0.03 | 0.70 | 0.03 | 0.00 |
| B4 | 0.90 | 0.73 | 0.63 | 0.03 | 0.03 | 0.70 | 0.00 | 0.03 |

Figure 4.22: Medium-Level Accuracy Comparison between Benchmark 3 and Benchmark 4

In the most difficult questions (Figure 4.23), implicit CoT tends to reduce performance in almost all models. GPT-5, Claude-Sonnet 4, and Gemini-2.5 Flash-Lite each decline by –5 pp, while GPT-5 mini, GPT-5 nano, Gemini-2.5 Flash, and DeepSeek-v3.1 show larger decreases of –10 pp. Only Claude-Haiku 3.5 remains stable. This suggests that, in harder tasks, the additional instruction does not support the models but instead contributes to reduced accuracy.

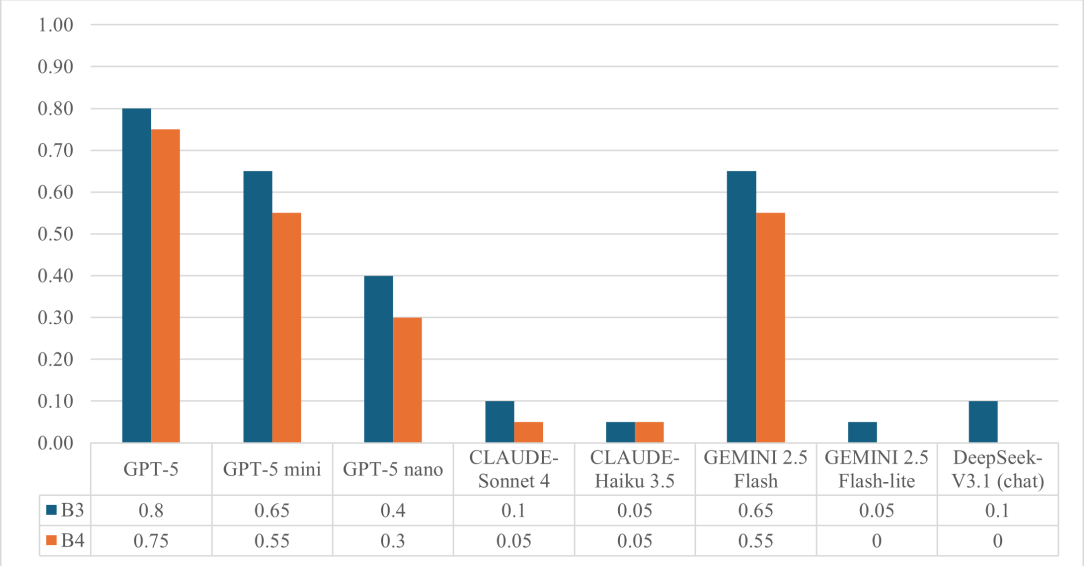



Figure 4.23: Hard-Level Accuracy Comparison between Benchmark 3 and Benchmark 4

The statistical significance analysis, conducted both in the general sample and in the subgroups by difficulty level, is reported in Tables Y.1 to Y.4. As shown in Table 4.18 (Overall Accuracy), none of the models reach the conventional 95% threshold ($p > 0.05$). The same applies to the medium and hard subsets (Tables 4.20 and Table 4.21), where no statistically significant differences emerge between Benchmark 3 and Benchmark 4. The only case of interest is Claude-Sonnet 4 in the easy subset (Table 4.19), where the asymptotic test suggests a value close to the significance threshold ($p = 0.045$). However, the exact p-value does not confirm this result ($p = 0.13$), indicating that the effect is not statistically robust.

| **LLM** | **B3** | **B4** | **Sample** | **Chi-square** | **p-value** | **Exact p-value** | **Significance (95%)** |
|---|---|---|---|---|---|---|---|
| GPT-5 | 0.92 | 0.90 | 100 | 0.50 | 0.48 | 0.73 | NO |
| GPT-5 mini | 0.79 | 0.78 | 100 | 0.14 | 0.71 | 1.00 | NO |
| GPT-5 nano | 0.71 | 0.69 | 100 | 0.50 | 0.48 | 0.73 | NO |
| Claude-Sonnet 4 | 0.31 | 0.27 | 100 | 2.67 | 0.10 | 0.22 | NO |
| Claude-Haiku 3.5 | 0.18 | 0.19 | 100 | 0.14 | 0.71 | 1.00 | NO |
| Gemini-2.5 Flash | 0.78 | 0.77 | 100 | 0.07 | 0.80 | 1.00 | NO |
| Gemini-2.5 Flash-lite | 0.22 | 0.21 | 100 | 0.14 | 0.71 | 1.00 | NO |
| DeepSeek-v3.1 | 0.22 | 0.21 | 100 | 0.20 | 0.65 | 1.00 | NO |

Table 4.18: Overall Accuracy: comparison between Benchmark 3 and Benchmark 4 with statistical significance test results

| LLM | B3 | B4 | Sample | Chi-square | p-value | Exact p-value | Significance (95%) |
|---|---|---|---|---|---|---|---|
| GPT-5 | 0.98 | 0.96 | 50 | 0.33 | 0.56 | 1.00 | NO |
| GPT-5 mini | 0.86 | 0.90 | 50 | 2.00 | 0.16 | 0.50 | NO |
| GPT-5 nano | 0.90 | 0.88 | 50 | 1.00 | 0.32 | 1.00 | NO |
| Claude-Sonnet 4 | 0.58 | 0.50 | 50 | 4.00 | 0.045 | 0.13 | NO* |
| Claude-Haiku 3.5 | 0.32 | 0.34 | 50 | 0.14 | 0.71 | 1.00 | NO |
| Gemini-2.5 Flash | 0.88 | 0.90 | 50 | 0.20 | 0.65 | 1.00 | NO |
| Gemini-2.5 Flash-lite | 0.40 | 0.42 | 50 | 0.20 | 0.65 | 1.00 | NO |
| DeepSeek-v3.1 | 0.40 | 0.40 | 50 | 0.00 | 1.00 | 1.00 | NO |

Table 4.19: Accuracy Easy: comparison between Benchmark 3 and Benchmark 4 with statistical significance test results

| LLM | B3 | B4 | Sample | Chi-square | p-value | Exact p-value | Significance (95%) |
|---|---|---|---|---|---|---|---|
| GPT-5 | 0.90 | 0.90 | 30 | 0.00 | 1.00 | 1.00 | NO |
| GPT-5 mini | 0.7667 | 0.7333 | 30 | 0.33 | 0.56 | 1.00 | NO |
| GPT-5 nano | 0.60 | 0.6333 | 30 | 1.00 | 0.32 | 1.00 | NO |
| Claude-Sonnet 4 | 0.00 | 0.0333 | 30 | 1.00 | 0.32 | 1.00 | NO |
| Claude-Haiku 3.5 | 0.0333 | 0.0333 | 30 | 1.00 | 0.32 | 1.00 | NO |
| Gemini-2.5 Flash | 0.70 | 0.70 | 30 | 0.00 | 1.00 | 1.00 | NO |
| Gemini-2.5 Flash-lite | 0.0333 | 0.00 | 30 | 1.00 | 0.32 | 1.00 | NO |
| DeepSeek-v3.1 | 0.00 | 0.0333 | 30 | 1.00 | 0.32 | 1.00 | NO |

Table 4.20: Accuracy Medium: ccomparison between Benchmark 3 and Benchmark 4 with statistical significance test results

| LLM | B3 | B4 | Sample | Chi-square | p-value | Exact p-value | Significance (95%) |
|---|---|---|---|---|---|---|---|
| GPT-5 | 0.80 | 0.75 | 20 | 0.33 | 0.56 | 1.00 | NO |
| GPT-5 mini | 0.65 | 0.55 | 20 | 2.00 | 0.16 | 0.50 | NO |
| GPT-5 nano | 0.40 | 0.30 | 20 | 0.67 | 0.41 | 0.69 | NO |
| Claude-Sonnet 4 | 0.10 | 0.05 | 20 | 1.00 | 0.32 | 1.00 | NO |
| Claude-Haiku 3.5 | 0.05 | 0.05 | 20 | 1.00 | 0.32 | 1.00 | NO |
| Gemini-2.5 Flash | 0.65 | 0.55 | 20 | 1.00 | 0.32 | 0.63 | NO |
| Gemini-2.5 Flash-lite | 0.05 | 0.00 | 20 | 1.00 | 0.32 | 1.00 | NO |
| DeepSeek-v3.1 | 0.10 | 0.00 | 20 | 2.00 | 0.16 | 0.50 | NO |

Table 4.21: Accuracy Hard: ccomparison between Benchmark 3 and Benchmark 4 with statistical significance test results

With respect to costs (Table 4.22), the comparison between B3 and B4 shows that the introduction of implicit CoT in numerical answers does not entail major changes for most models. GPT-5 slightly reduces its cost (–0.4%), while GPT-5 mini (+2.6%), GPT-5 nano (+6.3%),

Gemini-2.5 Flash (+2.4%), Gemini Flash-Lite (+2.8%), and DeepSeek-v3.1 (+7.6%) show modest increases. The main exception is Claude-Sonnet 4, which records a sharp increase (+83%), while Claude-Haiku 3.5 grows by +7.6%. Overall, implicit CoT generates moderate cost increases in most models, with particularly strong effects only in specific cases.

| **LLM** | **B3 ($)** | **B4 ($)** | **Variation (%)** |
|---|---|---|---|
| GPT-5 | 2.1683 | 2.1589 | -0.43% |
| GPT-5 mini | 0.2432 | 0.2496 | 2.63% |
| GPT-5 nano | 0.1090 | 0.1159 | 6.33% |
| Claude-Sonnet 4 | 0.1608 | 0.2943 | 83.02% |
| Claude-Haiku 3.5 | 0.0210 | 0.0226 | 7.62% |
| Gemini-2.5 Flash | 0.007741 | 0.007929 | 2.43% |
| Gemini-2.5 Flash-lite | 0.002561 | 0.002633 | 2.81% |
| DeepSeek-v3.1 | 0.012414 | 0.013357 | 7.60% |

Table 4.22: Cost comparison between Benchmark 3 and Benchmark 4 with percentage variation

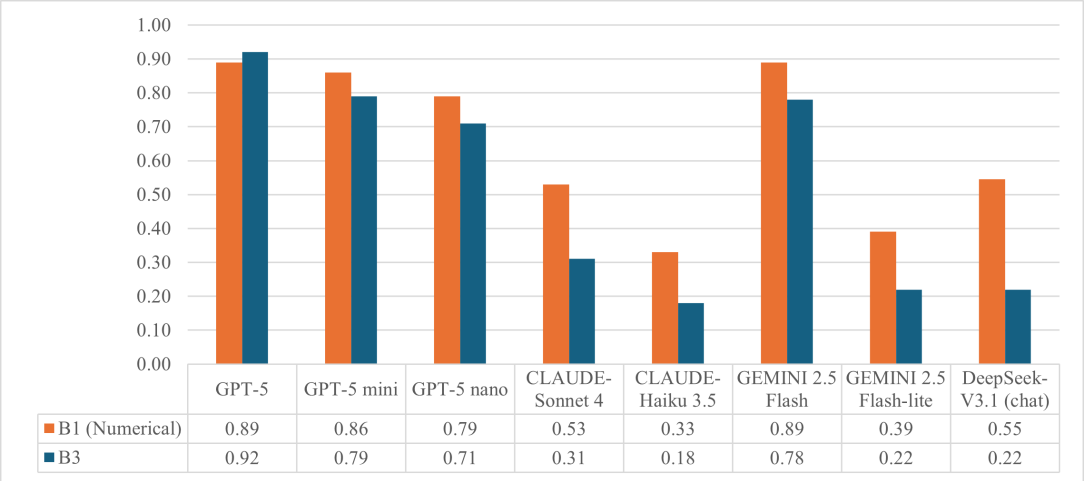
As for latency (Table 4.23), the introduction of implicit CoT tends to increase response times in several models. GPT-5 improves slightly (–2%), while GPT-5 mini rises by +3.3%. GPT-5 nano shows a significant increase (+26.5%), as do Claude-Sonnet 4 (+72%) and the two Gemini models (Flash +19.9%, Flash-Lite +15.4%). By contrast, Claude-Haiku 3.5 (+1.3%) and DeepSeek-v3.1 (+0.8%) show only minimal increases.

| **LLM** | **B3 (s)** | **B4 (s)** | **Variation (%)** |
|---|---|---|---|
| GPT-5 | 2768.99 | 2713.86 | -1.99% |
| GPT-5 mini | 1809.67 | 1868.40 | 3.25% |
| GPT-5 nano | 1427.19 | 1804.93 | 26.47% |
| Claude-Sonnet 4 | 257.05 | 442.89 | 72.30% |
| Claude-Haiku 3.5 | 96.13 | 97.36 | 1.28% |
| Gemini-2.5 Flash | 960.01 | 1150.67 | 19.86% |
| Gemini-2.5 Flash-lite | 67.39 | 77.79 | 15.43% |
| DeepSeek-v3.1 | 344.89 | 347.72 | 0.82% |

Table 4.23: Latency comparison between Benchmark 3 and Benchmark 4 with percentage variation

### 4.4.3 Benchmark 1 vs Benchmark 3

The comparison between Benchmark 1 (numerical single choice) and Benchmark 3 (numerical answer) highlights a loss of overall accuracy for most models when moving from single-choice to open numerical responses (Figure 4.24). GPT-5 is the only exception, with an improvement of +3 pp, while GPT-5 mini (–7 pp) and GPT-5 nano (–8 pp) show significant decreases. The Anthropic models are particularly affected, with Claude-Sonnet 4 (–22 pp) and Claude-Haiku 3.5 (–15 pp). Gemini models also decline (Flash –11 pp, Flash-Lite –17 pp), as does DeepSeek-v3.1 (–33 pp). Overall, the numerical answer format proves more challenging, with substantial negative shifts for nearly all models.




| | GPT-5 | GPT-5 mini | GPT-5 nano | CLAUDE-Sonnet 4 | CLAUDE-Haiku 3.5 | GEMINI 2.5 Flash | GEMINI 2.5 Flash-lite | DeepSeek-V3.1 (chat) |
|---|---|---|---|---|---|---|---|---|
| B1 (Numerical) | 0.89 | 0.86 | 0.79 | 0.53 | 0.33 | 0.89 | 0.39 | 0.55 |
| B3 | 0.92 | 0.79 | 0.71 | 0.31 | 0.18 | 0.78 | 0.22 | 0.22 |

Figure 4.24: Overall Accuracy Comparison between Benchmark 1 and Benchmark 3

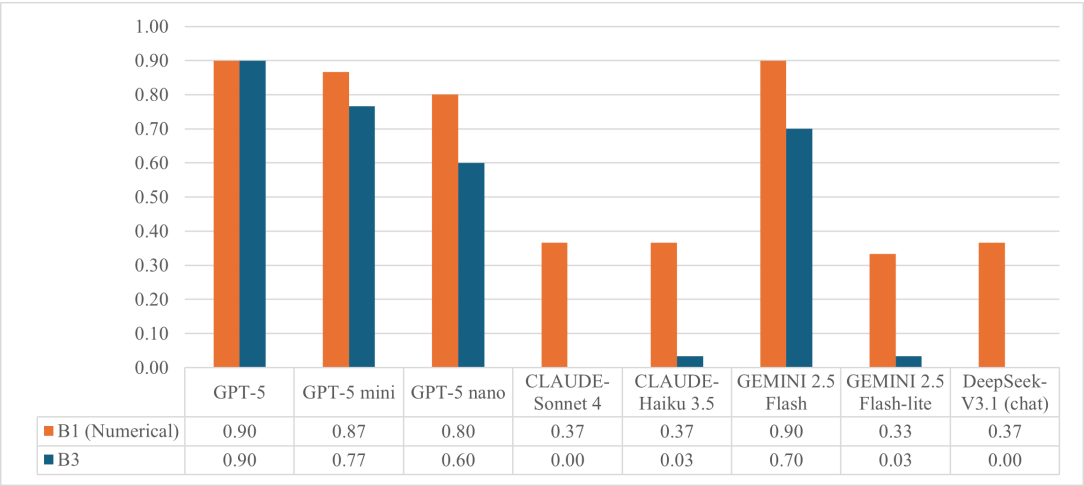
Breaking down overall accuracy by difficulty levels makes it possible to better understand how the change from numerical single choice (B1) to numerical answer (B3) affects performance, revealing dynamics not visible in the overall mean.

On easy questions (Figure 4.25), the impact is more limited. GPT-5 (+8 pp) and GPT-5 nano (+8 pp) improve, while GPT-5 mini remains stable. In contrast, the Anthropic models decline (Sonnet –10 pp, Haiku –4 pp), as do Gemini-2.5 Flash (–6 pp), 2.5 Flash-Lite (–2 pp), and DeepSeek-v3.1 (–36 pp).


| | GPT-5 | GPT-5 mini | GPT-5 nano | CLAUDE-Sonnet 4 | CLAUDE-Haiku 3.5 | GEMINI 2.5 Flash | GEMINI 2.5 Flash-lite | DeepSeek-V3.1 (chat) |
|---|---|---|---|---|---|---|---|---|
| B1 (Numerical) | 0.90 | 0.86 | 0.82 | 0.68 | 0.36 | 0.94 | 0.42 | 0.76 |
| B3 | 0.98 | 0.86 | 0.9 | 0.58 | 0.32 | 0.88 | 0.4 | 0.4 |

Figure 4.25: Easy-Level Accuracy Comparison between Benchmark 1 and Benchmark 3

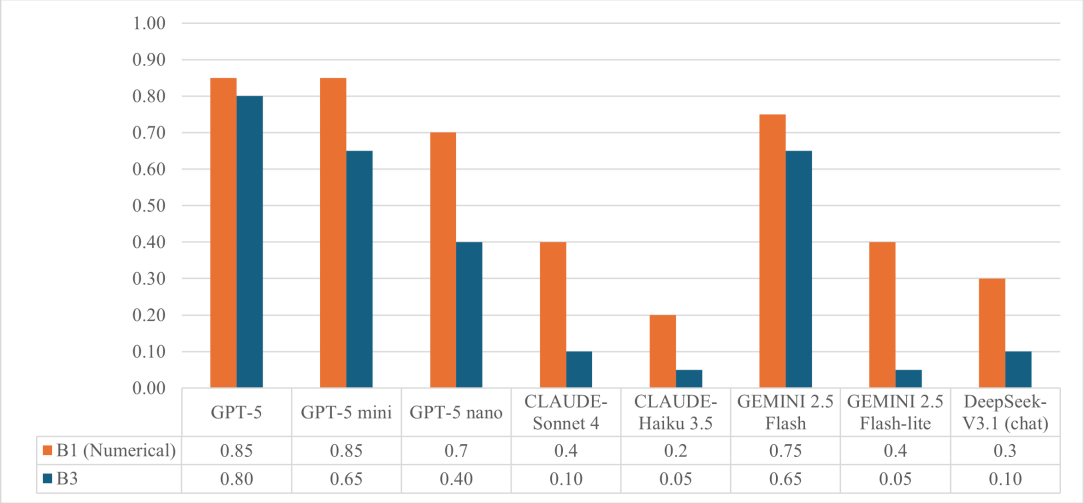
In medium questions (Figure 4.26), the transition to B3 leads to widespread decreases: GPT-5 remains unchanged, but GPT-5 mini (–10 pp) and GPT-5 nano (–20 pp) suffer significant drops. Both Claude models and DeepSeek worsen considerably (–30 to –37 pp), while Gemini Flash (–20 pp) and Flash-Lite (–30 pp) also record large losses.




| | GPT-5 | GPT-5 mini | GPT-5 nano | CLAUDE-Sonnet 4 | CLAUDE-Haiku 3.5 | GEMINI 2.5 Flash | GEMINI 2.5 Flash-lite | DeepSeek-V3.1 (chat) |
|---|---|---|---|---|---|---|---|---|
| B1 (Numerical) | 0.90 | 0.87 | 0.80 | 0.37 | 0.37 | 0.90 | 0.33 | 0.37 |
| B3 | 0.90 | 0.77 | 0.60 | 0.00 | 0.03 | 0.70 | 0.03 | 0.00 |

Figure 4.26: Medium-Level Accuracy Comparison between Benchmark 1 and Benchmark 3

In hard questions, the decline is even more pronounced (Figure 4.27): GPT-5 drops –5 pp, GPT-5 mini –20 pp, and GPT-5 nano –30 pp. The Claude models and DeepSeek record severe reductions (–15 to –30 pp), while the Gemini models lose between –10 and –35 pp. In this category, no model shows improvements: the transition to the numerical answer format systematically penalizes performance.




| | GPT-5 | GPT-5 mini | GPT-5 nano | CLAUDE-Sonnet 4 | CLAUDE-Haiku 3.5 | GEMINI 2.5 Flash | GEMINI 2.5 Flash-lite | DeepSeek-V3.1 (chat) |
|---|---|---|---|---|---|---|---|---|
| B1 (Numerical) | 0.85 | 0.85 | 0.7 | 0.4 | 0.2 | 0.75 | 0.4 | 0.3 |
| B3 | 0.80 | 0.65 | 0.40 | 0.10 | 0.05 | 0.65 | 0.05 | 0.10 |

Figure 4.27: Hard-Level Accuracy Comparison between Benchmark 1 and Benchmark 3

From the perspective of statistical significance, the comparison between Benchmark 1 and Benchmark 3 reveals numerous relevant differences. On the overall sample of 100 questions (Table 4.24), differences are statistically significant at the 95% level for Claude-Sonnet 4 (exact $p = 0.0001$), Gemini-2.5 Flash (exact $p = 0.0309$), Gemini-2.5 Flash-Lite (exact $p = 0.0243$), and DeepSeek-v3.1 (exact $p < 0.001$).

Looking at the difficulty subgroups, in the Easy subset (Table 4.25) significance is observed only for DeepSeek-v3.1 (exact $p = 0.0002$), while GPT-5 shows an asymptotic signal ($p = 0.045$) that is not confirmed by the exact test (exact $p = 0.125$).

In the Medium subset (Table 4.26), differences are significant for Claude-Sonnet 4 (exact $p = 0.001$), Claude-Haiku 3.5 (exact $p = 0.0117$), Gemini-2.5 Flash (exact $p = 0.0313$), Gemini-2.5 Flash-Lite (exact $p = 0.0215$), and DeepSeek-v3.1 (exact $p = 0.001$).

In the Hard subset (Table 4.27), no statistically significant differences emerge, except for Gemini-2.5 Flash-Lite (exact $p = 0.0391$). For Claude-Sonnet 4, the asymptotic value ($p = 0.039$) suggests a potential effect, but this result is not confirmed by the exact test (exact $p = 0.073$), and therefore cannot be considered robust.

| LLM | B1 | B3 | Sample | Chi-square | p-value | Exact p-value | Significance (95%) |
|---|---|---|---|---|---|---|---|
| GPT-5 | 0.890 | 0.920 | 100 | 0.60 | 0.4386 | 0.6072 | NO |
| GPT-5 mini | 0.860 | 0.790 | 100 | 3.52 | 0.0606 | 0.0931 | NO |
| GPT-5 nano | 0.790 | 0.710 | 100 | 2.94 | 0.0863 | 0.1214 | NO |
| Claude-Sonnet 4 | 0.530 | 0.310 | 100 | 15.11 | 0.0001 | 0.0001 | YES |
| Claude-Haiku 3.5 | 0.330 | 0.180 | 100 | 3.67 | 0.0555 | 0.0801 | NO |
| Gemini-2.5 Flash | 0.890 | 0.780 | 100 | 5.56 | 0.0184 | 0.0309 | YES |
| Gemini-2.5 Flash-lite | 0.390 | 0.220 | 100 | 5.76 | 0.0164 | 0.0243 | YES |
| DeepSeek-v3.1 | 0.545 | 0.220 | 100 | 26.95 | 0.0000 | 0.0000 | YES |

Table 4.24: Overall Accuracy: comparison between Benchmark 1 and Benchmark 3 with statistical significance test results

| LLM | B1 | B3 | Sample | Chi-square | p-value | Exact p-value | Significance (95%) |
|---|---|---|---|---|---|---|---|
| GPT-5 | 0.900 | 0.980 | 50 | 4.00 | 0.0455 | 0.1250 | NO* |
| GPT-5 mini | 0.860 | 0.860 | 50 | 0.14 | 0.7055 | 1.0000 | NO |
| GPT-5 nano | 0.820 | 0.900 | 50 | 1.29 | 0.2568 | 0.4531 | NO |
| Claude-Sonnet 4 | 0.680 | 0.580 | 50 | 2.25 | 0.1336 | 0.2101 | NO |
| Claude-Haiku 3.5 | 0.360 | 0.320 | 50 | 0.05 | 0.8185 | 1.0000 | NO |
| Gemini-2.5 Flash | 0.940 | 0.880 | 50 | 0.67 | 0.4142 | 0.6875 | NO |
| Gemini-2.5 Flash-lite | 0.420 | 0.400 | 50 | 0.07 | 0.7963 | 1.0000 | NO |
| DeepSeek-v3.1 | 0.760 | 0.400 | 50 | 13.76 | 0.0002 | 0.0002 | YES |

Table 4.25: Accuracy Easy: comparison between Benchmark 1 and Benchmark 3 with statistical significance test results

| LLM | B1 | B3 | Sample | Chi-square | p-value | Exact p-value | Significance (95%) |
|---|---|---|---|---|---|---|---|
| GPT-5 | 0.900 | 0.900 | 30 | 0.00 | 1.0000 | 1.0000 | NO |
| GPT-5 mini | 0.867 | 0.767 | 30 | 2.00 | 0.1573 | 0.2891 | NO |
| GPT-5 nano | 0.800 | 0.600 | 30 | 3.27 | 0.0707 | 0.1185 | NO |
| Claude-Sonnet 4 | 0.367 | 0.000 | 30 | 11.00 | 0.0009 | 0.0010 | YES |
| Claude-Haiku 3.5 | 0.367 | 0.033 | 30 | 7.36 | 0.0067 | 0.0117 | YES |
| Gemini-2.5 Flash | 0.900 | 0.700 | 30 | 6.00 | 0.0143 | 0.0313 | YES |
| Gemini-2.5 Flash-lite | 0.333 | 0.033 | 30 | 6.40 | 0.0114 | 0.0215 | YES |
| DeepSeek-v3.1 | 0.367 | 0.000 | 30 | 11.00 | 0.0009 | 0.0010 | YES |

Table 4.26: Accuracy Medium: comparison between Benchmark 1 and Benchmark 3 with statistical significance test results

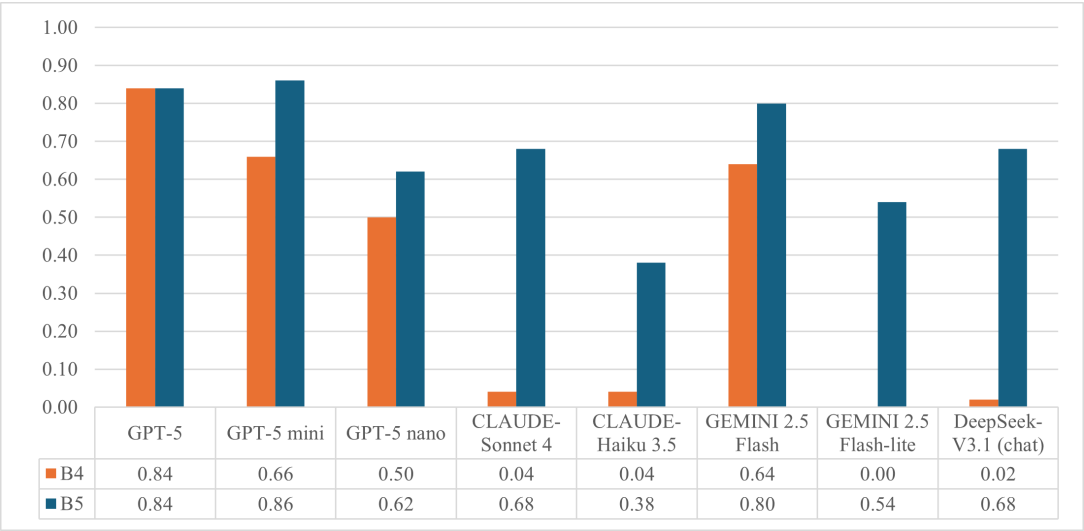
| LLM | B1 | B3 | Sample | Chi-square | p-value | Exact p-value | Significance (95%) |
|---|---|---|---|---|---|---|---|
| GPT-5 | 0.850 | 0.800 | 20 | 0.14 | 0.7055 | 1.0000 | NO |
| GPT-5 mini | 0.850 | 0.650 | 20 | 2.00 | 0.1573 | 0.2891 | NO |
| GPT-5 nano | 0.700 | 0.400 | 20 | 3.00 | 0.0833 | 0.1460 | NO |
| Claude-Sonnet 4 | 0.400 | 0.100 | 20 | 4.50 | 0.0339 | 0.0703 | NO* |
| Claude-Haiku 3.5 | 0.200 | 0.050 | 20 | 3.00 | 0.0833 | 0.2500 | NO |
| Gemini-2.5 Flash | 0.750 | 0.650 | 20 | 0.67 | 0.4142 | 0.6875 | NO |
| Gemini-2.5 Flash-lite | 0.400 | 0.050 | 20 | 5.44 | 0.0196 | 0.0391 | YES |
| DeepSeek-v3.1 | 0.300 | 0.100 | 20 | 2.67 | 0.1025 | 0.2188 | NO |

Table 4.27: Accuracy Hard: comparison between Benchmark 1 and Benchmark 3 with statistical significance test results

### 4.4.4 Benchmark 4 vs Benchmark 5

The comparison between Benchmark 4 (numerical answer with implicit Chain-of-Thought, limited to medium and hard questions) and Benchmark 5 (numerical answer with explicit Chain-of-Thought, requiring written reasoning) represents a crucial step in understanding whether implicit prompting to reason is sufficient, or whether the explicit articulation of logical steps is a necessary condition to improve performance.

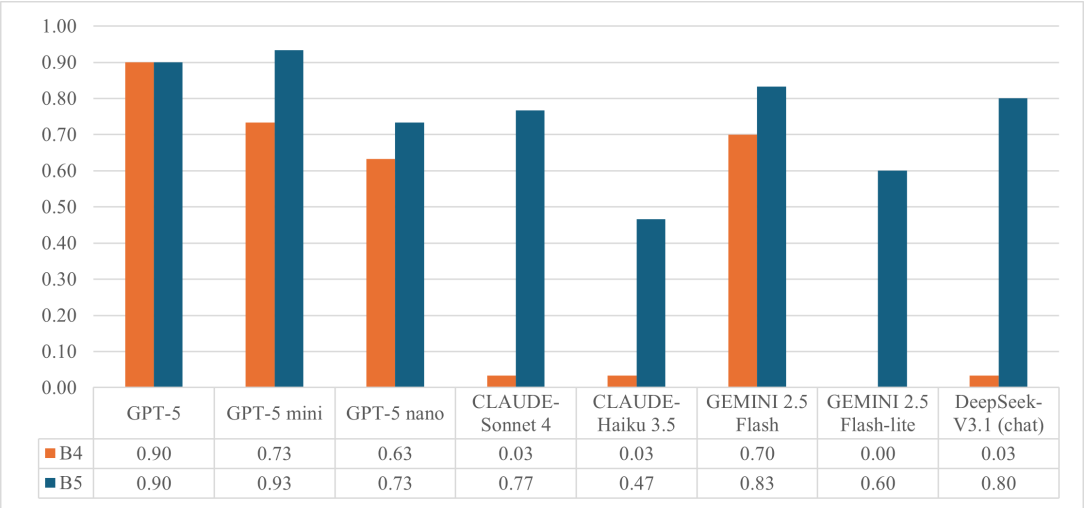
In terms of overall accuracy (Figure 4.28), the transition from implicit CoT (B4) to explicit CoT (B5) produces a substantial improvement for nearly all models. GPT-5 remains stable (0 pp), while GPT-5 mini (+20 pp) and GPT-5 nano (+12 pp) gain accuracy. The largest improvements are observed in the Anthropic models: Claude-Sonnet 4 (+64 pp) and Claude-Haiku 3.5 (+34 pp). The Gemini models also show marked gains (Flash +16 pp, Flash-Lite +54 pp), as does DeepSeek-v3.1 (+66 pp).




| | GPT-5 | GPT-5 mini | GPT-5 nano | CLAUDE-Sonnet 4 | CLAUDE-Haiku 3.5 | GEMINI 2.5 Flash | GEMINI 2.5 Flash-lite | DeepSeek-V3.1 (chat) |
|---|---|---|---|---|---|---|---|---|
| B4 | 0.84 | 0.66 | 0.50 | 0.04 | 0.04 | 0.64 | 0.00 | 0.02 |
| B5 | 0.84 | 0.86 | 0.62 | 0.68 | 0.38 | 0.80 | 0.54 | 0.68 |

Figure 4.28: Overall Accuracy Comparison between Benchmark 4 and Benchmark 5

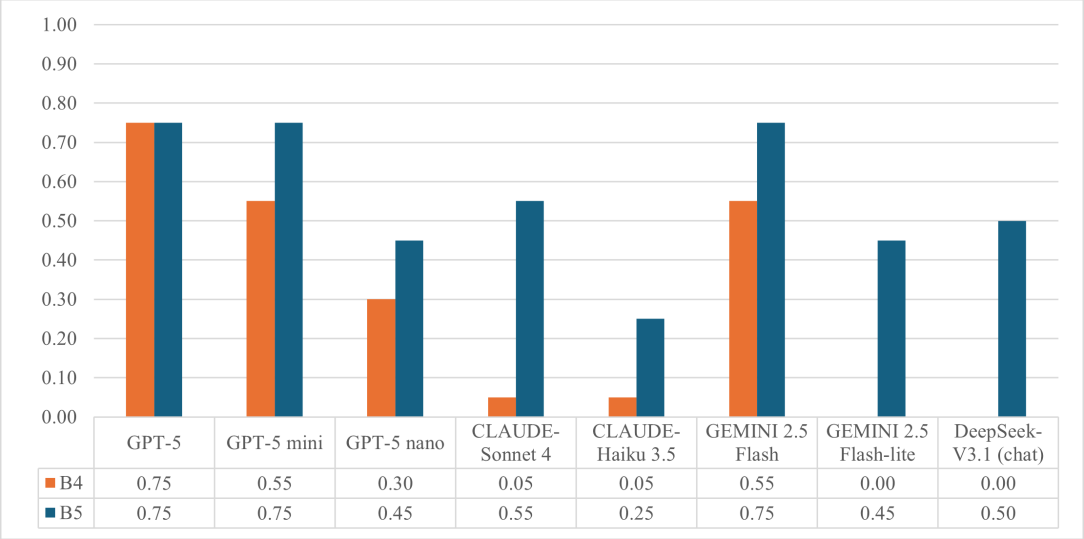
Considering only medium-difficulty questions (Figure 4.29), improvements are particularly evident. Claude-Sonnet 4 (+73 pp), Gemini Flash-Lite (+60 pp), and DeepSeek-v3.1 (+77 pp) achieve very large accuracy gains. Claude-Haiku 3.5 (+43 pp), GPT-5 mini (+20 pp), and GPT-5 nano (+10 pp) also benefit from explicit CoT, while GPT-5 remains unchanged.




| | GPT-5 | GPT-5 mini | GPT-5 nano | CLAUDE-Sonnet 4 | CLAUDE-Haiku 3.5 | GEMINI 2.5 Flash | GEMINI 2.5 Flash-lite | DeepSeek-V3.1 (chat) |
|---|---|---|---|---|---|---|---|---|
| B4 | 0.90 | 0.73 | 0.63 | 0.03 | 0.03 | 0.70 | 0.00 | 0.03 |
| B5 | 0.90 | 0.93 | 0.73 | 0.77 | 0.47 | 0.83 | 0.60 | 0.80 |

Figure 4.29: Medium-Level Accuracy Comparison between Benchmark 4 and Benchmark 5

For hard questions, explicit CoT again brings widespread benefits (Figure 4.30). GPT-5 mini (+20 pp), GPT-5 nano (+15 pp), Claude-Sonnet 4 (+50 pp), Claude-Haiku 3.5 (+20 pp), Gemini Flash (+20 pp), Gemini Flash-Lite (+45 pp), and DeepSeek-v3.1 (+50 pp) all register significant improvements. Once again, GPT-5 remains stable, with no variation.




| | GPT-5 | GPT-5 mini | GPT-5 nano | CLAUDE-Sonnet 4 | CLAUDE-Haiku 3.5 | GEMINI 2.5 Flash | GEMINI 2.5 Flash-lite | DeepSeek-V3.1 (chat) |
|---|---|---|---|---|---|---|---|---|
| B4 | 0.75 | 0.55 | 0.30 | 0.05 | 0.05 | 0.55 | 0.00 | 0.00 |
| B5 | 0.75 | 0.75 | 0.45 | 0.55 | 0.25 | 0.75 | 0.45 | 0.50 |

Figure 4.30: Hard-Level Accuracy Comparison between Benchmark 4 and Benchmark 5

From the perspective of statistical significance, the comparison between Benchmark 4 and Benchmark 5 highlights a substantial number of relevant differences. On the overall sample of 50 questions (Table 4.28), variations are significant for GPT-5 mini (exact $p = 0.0063$), Claude-Sonnet 4 (exact $p < 0.001$), Claude-Haiku 3.5 (exact $p < 0.001$), Gemini-2.5 Flash-Lite (exact $p < 0.001$), and DeepSeek-v3.1 (exact $p < 0.001$). For Gemini-2.5 Flash, the asymptotic test suggested a possible effect ($p = 0.0325$), but this was not confirmed by the exact test (exact $p = 0.0574$).

Looking at the subgroups, in the Medium set (Table 4.29) significant differences emerge for Claude-Sonnet 4 (exact $p < 0.001$), Claude-Haiku 3.5 (exact $p = 0.0002$), Gemini-2.5 Flash-Lite (exact $p < 0.001$), and DeepSeek-v3.1 (exact $p < 0.001$). GPT-5 mini shows an asymptotic value close to the threshold ($p = 0.0339$), but this is not confirmed by the exact test (exact $p = 0.0703$).

In the Hard set (Table 4.30), significance is confirmed for Claude-Sonnet 4 (exact $p = 0.002$), Gemini-2.5 Flash-Lite (exact $p = 0.0039$), and DeepSeek-v3.1 (exact $p = 0.002$). In this case, GPT-5 mini and Claude-Haiku 3.5 also show asymptotic signals ($p = 0.0455$), but the exact values ($p = 0.125$) do not confirm significance.

| LLM | B4 | B5 | Sample | Chi-square | p-value | Exact p-value | Significance (95%) |
|---|---|---|---|---|---|---|---|
| GPT-5 | 0.84 | 0.84 | 50 | 0.00 | 1.0000 | 1.0000 | NO |
| GPT-5 mini | 0.66 | 0.86 | 50 | 8.33 | 0.0039 | 0.0063 | YES |
| GPT-5 nano | 0.50 | 0.62 | 50 | 3.60 | 0.0578 | 0.1094 | NO |
| Claude-Sonnet 4 | 0.04 | 0.68 | 50 | 32.00 | 0.0000 | 0.0000 | YES |
| Claude-Haiku 3.5 | 0.04 | 0.38 | 50 | 17.00 | 0.0000 | 0.0000 | YES |
| Gemini-2.5 Flash | 0.64 | 0.80 | 50 | 4.57 | 0.0325 | 0.0574 | NO* |
| Gemini-2.5 Flash-lite | 0.00 | 0.54 | 50 | 27.00 | 0.0000 | 0.0000 | YES |
| DeepSeek-v3.1 | 0.02 | 0.68 | 50 | 31.11 | 0.0000 | 0.0000 | YES |

Table 4.28: OverallAccuracy: comparison between Benchmark 4 and Benchmark 5 with statistical significance test results

| LLM | B4 | B5 | Sample | Chi-square | p-value | Exact p-value | Significance (95%) |
|---|---|---|---|---|---|---|---|
| GPT-5 | 0.90 | 0.90 | 30 | 0.00 | 1.0000 | 1.0000 | NO |
| GPT-5 mini | 0.73 | 0.93 | 30 | 4.50 | 0.0339 | 0.0703 | NO* |
| GPT-5 nano | 0.63 | 0.73 | 30 | 1.80 | 0.1797 | 0.3750 | NO |
| Claude-Sonnet 4 | 0.03 | 0.77 | 30 | 22.00 | 0.0000 | 0.0000 | YES |
| Claude-Haiku 3.5 | 0.03 | 0.47 | 30 | 13.00 | 0.0003 | 0.0002 | YES |
| Gemini-2.5 Flash | 0.70 | 0.83 | 30 | 2.00 | 0.1573 | 0.2891 | NO |
| Gemini-2.5 Flash-lite | 0.00 | 0.60 | 30 | 18.00 | 0.0000 | 0.0000 | YES |
| DeepSeek-v3.1 | 0.03 | 0.80 | 30 | 21.16 | 0.0000 | 0.0000 | YES |

Table 4.29: Accuracy Medium: comparison between Benchmark 4 and Benchmark 5 with statistical significance test results

| LLM | B4 | B5 | Sample | Chi-square | p-value | Exact p-value | Significance (95%) |
|---|---|---|---|---|---|---|---|
| GPT-5 | 0.75 | 0.75 | 20 | 0.00 | 1.0000 | 1.0000 | NO |
| GPT-5 mini | 0.55 | 0.75 | 20 | 4.00 | 0.0455 | 0.1250 | NO* |
| GPT-5 nano | 0.30 | 0.45 | 20 | 1.80 | 0.1797 | 0.3750 | NO |
| Claude-Sonnet 4 | 0.05 | 0.55 | 20 | 10.00 | 0.0016 | 0.0020 | YES |
| Claude-Haiku 3.5 | 0.05 | 0.25 | 20 | 4.00 | 0.0455 | 0.1250 | NO* |
| Gemini-2.5 Flash | 0.55 | 0.75 | 20 | 2.67 | 0.1025 | 0.2188 | NO |
| Gemini-2.5 Flash-lite | 0.00 | 0.45 | 20 | 9.00 | 0.0027 | 0.0039 | YES |
| DeepSeek-v3.1 | 0.00 | 0.50 | 20 | 10.00 | 0.0016 | 0.0020 | YES |

Table 4.30: Accuracy Hard: comparison between Benchmark 4 and Benchmark 5 with statistical significance test results

## 4.5 Summary Results

The preliminary survey defined the weights to be applied in the Analytic Hierarchy Process (AHP), giving priority to *Accuracy* (normalized weight 0.56), followed by *Latency* (0.26) and *Cost* (0.18).

At the benchmark level, results show different patterns. In Benchmark 1 (Single Choice), accuracy ranged from 0.66 to 0.87, with three models above the human reference threshold (0.80). The AHP ranking placed Gemini-2.5 Flash first, followed by DeepSeek-v3.1 and GPT-5. With the introduction of implicit Chain-of-Thought in Benchmark 2, accuracies ranged from 0.65 to 0.88, again with three models above the baseline; AHP ranked Gemini-2.5 Flash first, followed by Gemini Flash-Lite and DeepSeek-v3.1.

Benchmark 3 (Numerical Answer) showed more polarized results, between 0.18 and 0.92, with only GPT-5 exceeding the human threshold. AHP placed GPT-5 first, followed by Gemini-2.5 Flash and GPT-5 mini. In Benchmark 4 (Numerical Answer with implicit CoT), accuracies ranged from 0.19 to 0.90, with only GPT-5 above the threshold; AHP again ranked GPT-5 first, followed by Gemini-2.5 Flash and GPT-5 mini. Finally, Benchmark 5 (Numerical Answer with explicit reasoning) reported accuracies from 0.38 to 0.86, with three models above the threshold. AHP ranked GPT-5 mini first, followed by Gemini-2.5 Flash and GPT-5.

Cross-benchmark comparisons helped isolate the effect of question format and prompting strategies. From Benchmark 1 to Benchmark 2, the addition of implicit CoT did not lead to systematic improvements: performance remained stable or slightly declined for most models, with limited gains for Gemini and DeepSeek. The only statistically significant difference concerned Claude-Sonnet 4, which recorded a marked drop ($p < 0.001$).

The comparison between Benchmark 1 and Benchmark 3, moving from single-choice numerical questions to open numerical answers, showed a decline in accuracy for almost all models, except GPT-5, which maintained strong performance. McNemar's tests confirmed significant differences for Claude-Sonnet 4, Claude-Haiku 3.5, Gemini Flash, Gemini Flash-Lite, and DeepSeek-v3.1, especially in medium-difficulty questions.

The comparison between Benchmark 3 and Benchmark 4, testing implicit CoT in numerical answers, did not show robust changes: accuracies were stable or slightly lower, with improvements only in easier questions. No significant differences were found, confirming that the variations were descriptive rather than systematic.

A clearer effect appeared in the comparison between Benchmark 4 and Benchmark 5, where

explicit CoT improved the performance of most models, particularly in medium and hard questions. Statistical tests confirmed significant differences for Claude-Sonnet 4, Claude-Haiku 3.5, Gemini Flash-Lite, and DeepSeek-v3.1, showing that explicit reasoning had robust effects.

Overall, the results provide a structured and replicable picture of LLM performance, highlight the trade-offs between accuracy, cost, and latency, and document in a systematic way the role of question format and prompting strategies. These findings form the basis for the critical discussion developed in the next chapter.

# Chapter 5

# Discussion

## 5.1 Main Results

The section dedicated to the main results aims to critically interpret the empirical evidence emerging from the benchmarks, relating it to the research objectives of the thesis. As recalled in the Introduction, the study is structured around two main research questions: (RQ1) which combinations of datasets, evaluation metrics, and prompting techniques enable the construction of meaningful benchmarks for the supply chain; (RQ2) which language model currently achieves the best overall performance, providing a comparative framework useful for supporting managerial choices.

To systematically address these questions, the results were discussed through four operational questions derived from the experimental benchmarks. These concern: the effectiveness of Chain of Thought, analyzed in both its implicit and explicit variants; the impact of question format on model performance; the insights provided by the rankings obtained through the Analytic Hierarchy Process (AHP); and finally, the extent to which LLMs are able to perform better than human evaluators in supply chain decision-making tasks.

This structure makes it possible to translate the two research questions into a more detailed analysis, capable of highlighting not only performance trends but also the theoretical and practical implications that follow.

The four questions discussed here therefore represent an operational declination of the research questions, aimed at highlighting strengths, weaknesses, and differences among models in scenarios of varying complexity. In this way, the results are interpreted not only in technical terms but also in light of the real needs of supply chain management, in line with the overall

objective of the thesis.

### 5.1.1 Question 1 – How does Chain of Thought (CoT), in its implicit and explicit forms, affect the performance of LLMs?

Recent literature attributes a central role to Chain of Thought in enhancing the performance of Large Language Models, suggesting that step-by-step reasoning can lead to higher accuracy, especially in complex tasks. However, most empirical evidence has focused on explicit CoT, in which the model is required to produce a sequence of logical steps before providing the final answer. In this research, in addition to that configuration, an implicit CoT was also introduced and tested, in which the instruction *"Let's think step by step"* was added to the prompt without requiring the reasoning to be made explicit: the expected output was only the final answer. This methodological choice allowed for a broader evaluation of CoT's impact, distinguishing between the two approaches and comparing their effects.

***Implicit CoT: B1 vs B2 and B3 vs B4***

The first level of analysis concerned the single-choice questions (B1 vs B2). Descriptive data showed a heterogeneous picture: some models, such as Gemini Flash, Flash-lite, and DeepSeek-v3.1, benefited from slight improvements, particularly in numerical questions; others, such as GPT-5 and GPT-5 nano, recorded declines in theoretical items and stability or small gains in numerical ones; while the Claude models, and especially Sonnet 4, exhibited a sharp deterioration. At first glance, these results could suggest that implicit CoT provides a marginal advantage in certain numerical contexts but penalizes other models, especially in theoretical questions. However, statistical significance testing tempers these observations: for almost all models, the differences between B1 and B2 are not significant, implying that implicit CoT does not have a systematic effect on single-choice accuracy. The only exception is Claude-Sonnet 4, for which the accuracy reduction is both evident and statistically significant, confirming the model's vulnerability to implicit reasoning instructions.

A similar picture emerges from the comparison B3 vs B4, focused on numerical answers, a format that requires the model to autonomously generate the correct response. Here, one might have expected a clearer benefit from implicit CoT, given the procedural nature of the task. Instead, results show that adding *"Let's think step by step"* does not lead to generalized

improvements: in easy questions there are occasional small gains, in medium ones the effect is ambiguous and inconsistent, while in hard ones implicit CoT tends to reduce accuracy, likely due to overthinking that introduces planning and calculation errors. Again, significance tests confirm the lack of robust effects: neither in the overall sample nor in the medium and hard subgroups do significant differences emerge. The only borderline signal concerns Claude-Sonnet 4 in the easy group, but the exact test does not confirm its robustness, suggesting that it is a contingent phenomenon.

Overall, the two comparisons show that implicit CoT is not a reliable strategy: the observed effects are weak, non-significant in most cases, and sometimes even negative. While in single-choice tasks the fluctuations tend to be negligible, in numerical answers implicit CoT becomes counterproductive as task difficulty increases.

***Explicit CoT: B4 vs B5***

A markedly different picture emerges from the comparison B4 vs B5, dedicated to explicit CoT. In this case, models were required not only to *"think step by step"* but also to make the reasoning sequence explicit. The results highlight a clear and consistent improvement: almost all models significantly increased their accuracy in medium and hard numerical answers. The benefits are particularly substantial for models that had shown the weakest performance with implicit CoT: Claude-Sonnet 4, Claude-Haiku 3.5, and DeepSeek-v3.1 report improvements ranging from +34 to +77 percentage points, recovering much of their performance gap. Gemini Flash and Flash-lite also show relevant improvements, while GPT-5 mini and GPT-5 nano record appreciable gains that strengthen their positioning, though without reaching the absolute levels of GPT-5. The latter, already highly performant, remains essentially stable. Unlike implicit CoT, these improvements are corroborated by significance tests, which confirm the robustness of the effects for most models.

From an interpretative perspective, these results demonstrate that the key difference lies not in the mere prompting of reasoning but in its explicit articulation. The requirement to display logical steps forces the model to structure its solution path, reducing the risk of disordered reasoning and uncontrolled overthinking typical of implicit CoT. In this sense, explicit CoT functions as an internal regularization mechanism, improving the coherence of responses and, especially in complex tasks, enhancing their reliability.

***General Synthesis***

Combining the three comparisons (B1 vs B2, B3 vs B4, and B4 vs B5), a clear conclusion emerges:

- Implicit CoT does not systematically improve LLM performance. The observed variations are weak, non-significant in most cases, and in complex scenarios tend even to reduce accuracy.
- Explicit CoT, by contrast, produces consistent and statistically robust improvements.

From a theoretical perspective, these results temper the notion, sometimes generalized in the literature, that CoT is inherently beneficial. Only the explicit articulation of reasoning steps leads to tangible improvements. From a practical perspective, this implies that implicit CoT should not be adopted as a default in managerial applications of supply chain management, while explicit CoT can represent an effective strategy for tackling complex tasks, provided that the higher costs and latency associated with generating extended reasoning are taken into account.

### 5.1.2 Question 2 – How does the performance of an LLM vary when addressing numerical questions in single-choice format compared to numerical answer format?

To address this question, a comparison was carried out between Benchmark 1, consisting exclusively of numerical questions in single-choice format, and Benchmark 3, composed of numerical answer questions. The aim was to verify whether the presence of predefined options implicitly guides the models toward the correct answer, as opposed to a format that requires autonomous generation of the numerical value.

The results show that the question format has a substantial impact on the performance of LLMs. In Benchmark 1, multiple-choice options act as an anchor, reducing the effort of autonomous generation and providing the model with useful references to narrow the space of possible answers. This mechanism translates into generally higher levels of accuracy. In Benchmark 3, by contrast, LLMs must produce the numerical value without any external support, highlighting increasing difficulties in maintaining precision, particularly in medium- and high-difficulty questions. The few improvements observed in models such as GPT-5 and GPT-5

nano in easy questions remain limited and do not alter the overall trend of declining performance.

The integration with statistical significance tests reinforces these conclusions: the differences observed between the two formats are not only visible at a descriptive level but, in many cases, also statistically robust, particularly for models such as Claude-Sonnet 4, Gemini-2.5 Flash, Gemini Flash-Lite, and DeepSeek-v3.1. This confirms that the advantage of single-choice questions over numerical answers is not a random phenomenon or limited to contingent variations, but rather a systematic effect linked to the structure of the question. Moreover, the fact that significance emerges especially in intermediate-difficulty questions suggests that the benefit of multiple-choice options is most evident when the task is sufficiently complex to challenge the models, yet not so difficult as to cause a generalized collapse in performance.

In summary, the comparison demonstrates that multiple-choice options provide a "positive bias" that guides LLMs toward the correct answer and supports their accuracy, whereas the absence of alternatives in the numerical answer format exposes their vulnerability in autonomous calculations. Statistical evidence consolidates this interpretation, showing that question format constitutes a decisive, rather than marginal, factor in model performance.

### 5.1.3 Question 3 – What insights emerge from the AHP rankings? Which LLM performs best in each benchmark, and why?

To integrate the different evaluation criteria emerging from the benchmarks, an Analytic Hierarchy Process (AHP) model was applied. This allowed the synthesis of three heterogeneous dimensions (accuracy, cost, and latency) into a single comparative index. The weights were derived from a survey conducted among Master's students in Management Engineering with a specialization in Supply Chain Management at Politecnico di Torino, with the aim of reflecting the priorities perceived by potential managerial users. It is important to emphasize, however, that these parameters have an inherently subjective character and that variations in the assigned weights could lead to changes in the final rankings.

The resulting rankings nevertheless allow for several relevant considerations. In the single-choice benchmarks (B1 and B2), the leading models are Gemini-2.5 Flash and, to varying degrees, DeepSeek-v3.1 and GPT-5. In particular, Gemini-2.5 Flash consistently secures the top position in both scenarios, thanks to its combination of solid accuracy and relatively low costs. DeepSeek-v3.1 rises to second place in B1 but falls to third in B2 due to its sensitivity

to implicit CoT. GPT-5, while achieving the highest absolute accuracy, is penalized by its high cost and latency, which push it down in the rankings.

In the numerical benchmarks without CoT (B3) and with implicit CoT (B4), GPT-5 clearly dominates, ranking first in both cases. Its superiority is driven by significantly higher accuracy compared to other models, sufficient to offset its higher computational costs and latency. In these settings, Gemini-2.5 Flash consistently holds the second position, offering a favorable balance between accuracy and cost, while GPT-5 mini remains stable in an intermediate position. The Claude models, on the other hand, consistently occupy the lowest ranks, penalized by their limited accuracy.

In the explicit CoT benchmark (B5), an interesting shift occurs: GPT-5 mini rises to the top, surpassing both GPT-5 and Gemini-2.5 Flash. This outcome highlights that, when explicit reasoning is required, lighter models can achieve a better balance between performance and resource consumption. GPT-5 nonetheless remains among the top performers, while Claude-Sonnet 4 shows a substantial improvement compared to previous benchmarks, indicating that explicit reasoning contributes significantly to reducing its errors.

In summary, the AHP analysis reveals that there is no single "absolute winner" across all benchmarks. Rather, each model excels in specific configurations depending on the trade-off between accuracy, cost, and latency. Several cross-cutting insights are worth highlighting:

- Gemini-2.5 Flash proves to be the strongest model in single-choice tasks and consistently secures second place in all other benchmarks, confirming its solidity and versatility.
- GPT-5 dominates numerical-answer tasks, both without CoT and with implicit CoT, owing to its superior accuracy, which compensates for its higher cost and latency.
- GPT-5 mini demonstrates a particularly noteworthy trajectory: as task complexity increases, it climbs steadily in the rankings until reaching the top in B5. Moving from fifth place in B1 to first in B5, it reveals strong adaptability to tasks requiring explicit reasoning.
- Overall, the models most frequently on the podium, apart from single-choice tasks, are Gemini-2.5 Flash, GPT-5, and GPT-5 mini, which together emerge as the most reliable across benchmarks.

Finally, it is essential to underscore the importance of the trade-off between accuracy, cost, and latency. No model excels simultaneously across all three dimensions. Some, like GPT-5,

deliver very high accuracy at the expense of resource efficiency, while others, such as Gemini Flash or GPT-5 mini, offer more balanced solutions. This represents a crucial point both theoretically, as it challenges the notion of a universally "best" model, and practically, as it guides managerial decision-making according to operational priorities and available resources.

### 5.1.4 Question 4 – Do LLMs perform better than humans in supply chain tasks?

To address this question, model performance was systematically compared against a human baseline, defined as the average accuracy achieved by fifth-year Management Engineering students. This threshold provides a realistic reference point for managerial decision-making capabilities, against which the outcomes of the experimental analysis can be critically interpreted.

The evidence reveals a heterogeneous picture. On the one hand, high-end models such as GPT-5, GPT-5 mini, and Gemini-2.5 Flash frequently exceed the human baseline, demonstrating the ability to provide answers that are not only comparable but, in several cases, superior to those of a human decision-maker. This is particularly evident in structured tasks and in numerical questions requiring explicit reasoning, where these models consistently reach or surpass the reference level. On the other hand, a substantial share of the models examined remain below the human threshold, showing marked difficulties especially in numerical questions of medium and high complexity.

From an interpretative perspective, this finding highlights the selective nature of performance: not all LLMs can be regarded as reliable substitutes or complements to human reasoning, but the most advanced models clearly demonstrate the capacity to compete with and in some cases outperform human evaluators. It follows that the validity of LLMs for decision making in the supply chain should not be assessed in absolute terms, but rather in relation to the specific model selected and the decision context in which it is applied.

In summary, comparison with the human baseline confirms that the technology has reached a sufficient degree of maturity to represent a concrete support for managerial decision making, provided that the variability across models is carefully considered. The ability of the best-performing LLMs to surpass human accuracy underscores their potential, while the weaker performance of other models calls for a cautious and selective adoption.

## 5.2 Secondary Results

The section dedicated to the secondary results complements the analysis of the main benchmarks by examining aspects that, while not directly linked to the core research questions, contribute to a more comprehensive understanding of LLM performance in supply chain decision making.

Three areas are particularly relevant: the trade-offs between accuracy, cost, and latency, which reflect the different design strategies adopted by providers; the perceptions collected through the survey, which shed light on the priorities of potential users; and finally, the analysis of error patterns in explicit reasoning tasks, which distinguishes between calculation, interpretation, and planning limitations.

These results do not alter the overall conclusions of the thesis but add depth and detail to the interpretation of the benchmarks, offering a more nuanced perspective that is closely aligned with the practical needs of supply chain management.

### 5.2.1 Performance Trade-offs in LLMs

A noteworthy secondary finding emerging from the benchmarking exercise concerns the heterogeneity of performance across models and providers in the supply chain domain. The observed differences are not incidental; rather, they reflect the underlying design choices and market positioning strategies explicitly outlined by the respective developers. Therefore, the analysis carried out in this study confirms that the general features ascribed to various LLMs, particularly in terms of accuracy, cost, and latency, also hold when these models are applied to specific operational contexts such as supply chain management.

For OpenAI, the GPT-5 family illustrates most clearly a strategy oriented towards maximizing accuracy. The flagship version delivered consistently high performance even in demanding numerical tasks, frequently standing out as the only model capable of exceeding the human-level accuracy threshold. However, this robustness comes at the expense of significantly higher costs and slower response times compared to other providers. The lighter variants, GPT-5 mini and nano, behaved as expected: they ensured faster outputs and lower costs, but with a marked decline in accuracy as task complexity increased.

A different pattern emerged for Anthropic. The Claude models (Sonnet 4 and Haiku 3.5) performed well in multiple-choice theoretical questions, where textual coherence plays a cen-

tral role, yet struggled with numerical answer tasks. This polarization reflects a design philosophy prioritizing safety and discursive consistency over autonomous calculation capabilities. It is therefore unsurprising that, particularly under implicit chain-of-thought conditions, some of these models exhibited sharp drops in accuracy.

The Gemini-2.5 family by Google demonstrated a more balanced approach. Flash proved to be one of the strongest compromises, approaching GPT-5's performance levels while maintaining considerably lower costs and latency. In contrast, Flash-Lite pushed efficiency to its limit: extremely fast and inexpensive, but with unstable accuracy and sensitivity to task complexity. This internal differentiation confirms Google's strategic positioning, which emphasizes flexible deployment options tailored to varying operational requirements.

Finally, DeepSeek-v3.1 reaffirmed its orientation toward cost-effectiveness. Extremely low costs and reduced latency make the model attractive in scenarios where efficiency outweighs accuracy. Nonetheless, performance proved inconsistent, with sharp declines when moving from single-choice to numerical-answer tasks, underscoring a pronounced sensitivity to task format.

Taken together, these secondary results show that the characteristics described by providers in their official documentation are consistently borne out within the supply chain domain. OpenAI stands out for accuracy, Anthropic for textual coherence, Google for balancing efficiency with performance, and DeepSeek for economic accessibility. In this respect, architectural and strategic choices do not remain abstract claims; they translate into concrete outcomes when LLMs are applied to complex and realistic settings such as supply chain management.

### 5.2.2 Impact of Implicit CoT on Costs and Latency

An interesting secondary result from the comparisons between B1 vs B2 and B3 vs B4 concerns the impact of introducing implicit Chain-of-Thought (CoT) on costs and latency. While implicit CoT did not consistently improve performance, its effect on response times and computational resources deserves further consideration.

#### Costs

The introduction of implicit CoT resulted in a significant increase in operational costs for several models. For example, models like GPT-5 nano and Claude-Sonnet 4 showed a notable rise in costs, suggesting that adding CoT introduces computational complexity that does not always

lead to better performance. This is especially relevant in business settings where economic efficiency is crucial. In these cases, the extra computational effort required for CoT could outweigh its cognitive benefits.

#### Latency

The results show that while some models, such as Gemini-2.5 Flash-Lite, saw a reduction in latency (-19.20%) with the introduction of implicit CoT, more complex models like GPT-5 nano experienced an increase in latency (+7.46%). This suggests that the effect on latency depends on the model's complexity and the resources needed to handle the additional reasoning introduced by CoT. Increased latency may limit the use of implicit CoT in business applications that require high computational performance, where response times are critical.

These findings emphasize the importance of considering both costs and latency when evaluating the potential of CoT in real-world applications. While implicit CoT may offer some benefits in certain scenarios, its impact on operational efficiency could restrict its use in environments where speed and cost-effectiveness are essential.

### 5.2.3 Survey Results and Evaluators' Perceptions

An additional secondary result derives from the survey conducted to capture stakeholder preferences regarding evaluation criteria. Although initially conceived as a methodological tool to support the AHP process, the survey provided valuable insights into how LLMs are perceived in the context of supply chain management.

The responses revealed a very clear hierarchy of priorities: accuracy was regarded as the dominant criterion, while latency and cost were considered considerably less important. This orientation reflects the central role of correctness in supply chain decision-making, where errors in evaluation can lead to significant operational and economic consequences.

The lower ranking of cost and latency suggests that respondents are willing to accept longer response times or higher expenditures as long as output quality is ensured. This result partially diverges from the benchmarks, where models revealed the need to manage systematic trade-offs between performance, efficiency, and economic sustainability. An interesting contrast therefore emerges: while the experimental data highlight the inevitability of trade-offs, the perceptions collected through the survey tend to minimize them, placing accuracy as the decisive factor.

It is important to note, however, that these results are subjective, as they stem from the evaluations of a limited sample of 30 master's students from Politecnico di Torino, who were asked to assume the role of supply chain managers. Consequently, they cannot be considered representative of the entire community of potential users but rather provide a preliminary indication of the sensitivities and priorities perceived by a circumscribed group.

Overall, the survey highlights an aspect that complements the benchmarks: while the latter measure the actual performance of the models, the survey offers a snapshot of the subjective perceptions of those who might adopt them in operational settings. The integration of these two perspectives allows for a broader understanding, in which accuracy emerges as a non-negotiable criterion, whereas cost and latency play a secondary and instrumental role.

### 5.2.4 Performance Across Theoretical vs. Numerical Questions

The results presented in Figure 4.5 reveal clear differences in the ability of the models to answer theoretical versus numerical questions. Some models, such as GPT-5 and Gemini-2.5 Flash, demonstrate relatively balanced performance across both types of question, whereas others, including Claude-Sonnet 4 and DeepSeek-v3.1, perform considerably better on theoretical questions than on numerical ones. In contrast, lightweight variants, such as Gemini Flash-Lite, exhibit significant limitations in numerical calculations while performing adequately on conceptual tasks.

These discrepancies suggest that models are not universally suitable for all types of tasks: some excel when linguistic understanding and abstract reasoning are required, whereas others are better suited for numerical computation.

From an applied perspective, this underscores the importance of context-specific model selection in supply chain management: depending on the task requirements, whether conceptual reasoning or numerical accuracy, the choice of the most appropriate model may vary.

### 5.2.5 Understanding Error Patterns in Explicit Reasoning

A further result of the analysis relates to the decomposition of performance in numerical questions with explicit reasoning. Errors were classified into calculation and reasoning errors, with the latter divided into interpretation and pianification errors. The data show that the ability to perform calculations is consistently strong across all models, with scores ranging from 0.81 for

Claude-Haiku 3.5 up to 1.00 for GPT-5 mini and DeepSeek-v3.1. Greater variability emerges in the reasoning dimension, where results range from 0.896 for Gemini-2.5 Flash down to 0.515 for Claude-Haiku 3.5.

Looking more closely at the reasoning errors, planning mistakes appear more common than those linked to interpretation. In practical terms, this means that models are usually able to understand the request, but they often struggle to structure the reasoning steps correctly or to apply formulas in a coherent way. This trend is especially evident in DeepSeek-v3.1 and Gemini-2.5 Flash, which, despite achieving good overall performance, show a relatively high incidence of planning errors.

These findings suggest that numerical computation is a relatively stable capability among LLMs, while reasoning remains a more fragile area, particularly when tasks require building and executing a structured logical process. Introducing this distinction allows for a more detailed interpretation of model performance and brings the evaluation closer to cognitive perspectives, making it possible to identify not only how frequently errors occur but also their specific nature.

## 5.3 Theoretical Implications

The analysis conducted in this study goes beyond the presentation of empirical findings, offering theoretical reflections that enrich the broader debate on LLMs. Previous literature has largely assessed models using generic benchmarks, often focused on linguistic or abstract reasoning tasks. However, the results of this research highlight the need to revise and extend such approaches, incorporating perspectives more closely aligned with the challenges of professional domains such as supply chain management.

This section discusses the main theoretical implications that emerged, organized according to the methodological and analytical dimensions that guided the study.

### 5.3.1 Domain-specific benchmarks

A first theoretical contribution lies in the development of a methodology for constructing benchmarks tailored to professional domains, such as the supply chain. The adoption of targeted datasets, combined with dedicated prompting techniques and domain-calibrated evaluation tools, demonstrates the limitations of generic benchmarks which, although widely used in the

literature, fail to capture the complexity of real-world LLM applications. This points to the need for domain-specific benchmarking frameworks that are able to provide more meaningful measures of performance and of actual utility of the models in concrete scenarios.

### 5.3.2 Task design and the difficulty pyramid

The research also highlights the crucial role of task design in evaluating model performance. The use of different formats (single choice, numerical and numerical with reasoning) showed that, even with identical content, the results of the model can vary significantly. This indicates that performance depends not only on the intrinsic capabilities of LLMs, but also on the way in which tasks are structured.

To systematize this complexity, the study drew on Bloom's taxonomy, which classifies cognitive activities along a progression from basic to advanced levels. Building on this framework, a difficulty pyramid was developed to organize the questions gradually, from simple tasks related to recognition or recall to more complex ones that require articulated reasoning and the explicit presentation of logical steps. This progression is not only of methodological value but also reflects managerial practice: while it is unlikely that decision-makers deal with basic multiple-choice questions, they frequently face open-ended numerical problems requiring structured reasoning. The theoretical implication is that task design, when organized through a Bloom-inspired hierarchy of difficulty, becomes a critical variable for benchmarking LLMs. Only through this approach can one meaningfully assess their ability to support complex decision-making processes and address real managerial needs.

### 5.3.3 Error taxonomy

Another theoretical contribution stems from the introduction of a new taxonomy of errors. In numerical questions with explicit reasoning, evaluation went beyond the binary distinction between correct and incorrect answers, encompassing different error types: calculation errors (numerical mistakes) and reasoning errors. The latter were further divided into interpretation errors (misunderstanding of the task or prompt) and planning errors (mistakes in structuring the reasoning or applying formulas). This framework enriches the theoretical literature by aligning the evaluation of LLMs more closely with cognitive models, as it enables assessment not only of how often models fail, but also of how they fail.

### 5.3.4 The role of CoT

A key point concerns the theoretical reflection on Chain of Thought (CoT). The literature has predominantly focused on explicit CoT, where models are required to make their reasoning transparent, and it is often portrayed as universally beneficial. This study examined both explicit CoT and implicit CoT, the latter involving only the final answer without requiring logical steps to be displayed.

The results show that implicit CoT, in most cases, does not have a statistically significant impact on performance and, in some situations, may even reduce accuracy. Explicit CoT, by contrast, improved output quality for the majority of models tested by encouraging a more structured reasoning process. However, this effect was not universal: models such as GPT-5 and GPT-5 nano maintained stable performance regardless of whether explicit reasoning was used. The theoretical implication is that the benefits of CoT cannot be assumed to apply universally, but must be interpreted in relation to both the model and the application context. This calls for a more critical and contextualized perspective, challenging theories that frame CoT as an inherently advantageous strategy.

### 5.3.5 Survey and AHP

The research also integrated a survey with the Analytic Hierarchy Process (AHP), introducing a socio-technical dimension into the evaluation framework. This approach made it possible to consider three criteria simultaneously, accuracy, cost, and latency, and derive relative weights based on the preferences of the participants. The theoretical implication is twofold. First, it broadens the perspective of benchmarking models, shifting from a purely technical analysis to a multi-criteria evaluation. Second, it recognizes that stakeholder perceptions play a crucial role in adoption processes. The convergence between empirical results and subjective perceptions reinforces the robustness of the framework, while divergences highlight potential areas of tension between what models deliver and what users expect. In this way, the evaluation of LLMs is reframed not only as a technical exercise, but also as a social and contextual one.

### 5.3.6 Statistical validation of results

Finally, the use of statistical tools such as the McNemar test allowed verification of whether the observed differences between models and conditions were statistically significant. This

methodological step, still relatively uncommon in the LLM benchmarking literature, enabled a distinction between robust effects and random fluctuations. The theoretical implication is that evaluation models should rest on a solid inferential basis, moving beyond simple percentage comparisons and promoting a more rigorous and reliable approach to studying performance.

## 5.4 Practical Implications

Beyond their theoretical significance, the findings of this study also provide a set of practical insights that can assist organizations and managers in the adoption of LLMs within supply chain operations. Whereas earlier research has often described the potential of such models in broad or generic terms, the results presented here underline the need to turn empirical evidence into concrete guidance that can inform managerial decisions and support the selection of models suited to real operational settings.

This section outlines the principal practical implications emerging from the analysis and considers how they may influence both the strategic choices of firms and the development directions pursued by technology providers.

### 5.4.1 Defining priorities and making informed model choices

The results of this study indicate that there is no single "best" model in absolute terms. Each provider follows a distinct strategy and offers specific trade-offs between accuracy, cost, and latency. For companies, this means that selection cannot rely on generic rankings but must instead be guided by internal priorities and the operational context. Within this perspective, tools such as surveys combined with the Analytic Hierarchy Process (AHP) take on strategic value. When applied inside an organization, they make it possible to capture stakeholder preferences and translate them into concrete evaluation criteria, producing tailored rankings that reflect the firm's actual needs. For instance, a company that places the highest emphasis on accuracy is likely to opt for models such as GPT-5, while those facing tighter constraints on cost or latency may find Gemini or DeepSeek more suitable.

Surveys are not only useful for firms but also for providers. They offer a means of better understanding market needs and of steering the development of solutions that align with stakeholder priorities. The practical implication is that the selection and evolution of LLMs should follow a fit for purpose logic, built on a clear identification of priorities emerging both from the

demand side (companies and users) and the supply side (providers).

### 5.4.2 Formulating queries for LLMs

A second practical implication concerns the way managers interact with the models. The study demonstrates that the phrasing of queries has a decisive impact on the quality of responses. If a manager prefers to receive only the final answer without an explanation of the reasoning, at present GPT-5 is the only model that maintains a high level of accuracy even without an explicit Chain of Thought. For the other models, however, the results of Benchmark 5 suggest that explicit CoT should be used, as it makes the reasoning process transparent and reduces the risk of errors. This means that firms should not limit themselves to selecting a model but also need to develop skills and internal guidelines for prompting, so as to identify the most effective interaction style for their operational and decision-making needs.

### 5.4.3 Summary

In conclusion, the practical implications of this research revolve around two key aspects. First, the definition of priorities, supported by tools such as surveys and AHP, which enables companies to select models that truly match their requirements while also providing providers with valuable indications for future development. Second, the formulation of queries, which acts as a concrete lever for improving the reliability and usefulness of outputs, thereby ensuring that LLMs can be employed more effectively in supporting decision-making processes within the supply chain.

# Chapter 6

# Conclusions

This study started from a clear observation: as projects and supply chains become increasingly complex, managers need effective tools to support decision-making. Large Language Models (LLMs) have emerged rapidly and show significant potential. However, two central questions remained: can these models be trusted in real managerial contexts, and how can they be evaluated systematically and rigorously?

Two main challenges characterize the current debate. First, managers often adopt new technologies without sufficient evidence of their effectiveness, exposing organizations to costly or suboptimal decisions. Second, academic research has not yet developed benchmarks specific to professional domains, which are necessary for systematic and comparable evaluations, particularly in supply chain management.

This research addressed two primary questions: (RQ1) which combinations of datasets, evaluation metrics, and prompting techniques are best suited for constructing meaningful and replicable benchmarks; and (RQ2) which models provide the most reliable performance for practical managerial use.

To address RQ1, the study proposed a methodological framework that combines supply chain–specific datasets, calibrated prompting strategies, and a wide set of evaluation metrics. It integrates multiple question formats, a hierarchy of difficulty inspired by Bloom's taxonomy, a detailed error taxonomy, and statistical validation. The resulting benchmarks are systematic, replicable, and closely aligned with real-world decision-making requirements, providing both a foundation for future research and practical guidance for organizations.

The role of Chain of Thought (CoT) was carefully examined. Implicit CoT, where reasoning occurs internally without being displayed, does not consistently improve accuracy and, in

some cases, reduces it. Moreover, implicit CoT increases operational costs and latency, affecting efficiency and limiting its practical use in time- and resource-sensitive environments. By contrast, explicit CoT, which requires models to articulate their reasoning, generally improves the quality of responses, although the effect varies across tasks and models.

Regarding RQ2, no single model outperformed all others across all benchmarks. GPT-5 achieved the highest accuracy but incurred higher costs and slower response times. Gemini Flash offered a balanced trade-off between performance, cost, and latency. GPT-5 mini demonstrated strong adaptability, especially for reasoning-intensive questions. Claude performed well in open-ended textual tasks but struggled with numerical problems, while DeepSeek prioritized cost-efficiency at the expense of performance on complex tasks. Task design, including question format and difficulty, emerged as a critical factor, and error analysis revealed that many mistakes were due to weaknesses in reasoning rather than misinterpretation of the questions.

The survey integrated with the Analytic Hierarchy Process (AHP) added an additional socio-technical perspective. While the benchmarks showed trade-offs among accuracy, cost, and latency, respondents consistently emphasized accuracy as the top priority, confirming its central importance in managerial decision-making.

In conclusion, this study contributes both theoretically and practically. Theoretically, it provides replicable, domain-specific, and multidimensional benchmarks that account for cognitive processes and operational contexts. Practically, it offers managers and providers tools to make informed model selections, optimize prompting strategies, and align model performance with organizational priorities.

Ultimately, the research answers the key question: can LLMs be trusted to support decision-making in supply chain management? The answer is cautiously positive. Leading models, such as GPT-5, Gemini Flash, and in specific cases GPT-5 mini, not only match but sometimes surpass human performance, showing potential as decision-support partners. Nevertheless, variability across models and persistent challenges in reasoning indicate that adoption should remain selective, context-aware, and guided by robust benchmarks, with a clear understanding of the associated trade-offs.

## 6.1 Delimitations

The study was intentionally framed within specific boundaries to maintain alignment with its objectives. First, the focus was placed on supply chain management, a domain where systematic benchmarks for evaluating LLMs are still lacking, despite the fact that managerial decisions in this area directly affect efficiency and competitiveness.

The benchmark itself was based on pre-defined question types (single choice, numerical, and numerical with reasoning), organized in a hierarchy of difficulty inspired by Bloom's taxonomy. Broader assessment formats, such as extended case studies, were deliberately excluded. While such formats might mirror real-world practices more closely, they would have introduced methodological complexity inconsistent with the need for replicability and systematic comparison.

Regarding model selection, the analysis was restricted to a set of current commercial LLMs, excluding open-source solutions and earlier versions. This decision allowed the study to concentrate on technologies most relevant to present-day business applications.

Finally, prompting techniques such as Zero-Shot e Role prompting were chosen to reflect practical usage scenarios. More advanced approaches, such as Tree-of-Thought or ReAct, were not considered, as they require significantly greater computational resources and were deemed inconsistent with the pragmatic orientation of the study.

## 6.2 Limitations

Alongside the deliberate choices made in this study, a few limitations need to be acknowledged, as they affect how the results can be interpreted and applied.

First, the survey for the AHP analysis was conducted with 30 master's students from Politecnico di Torino, rather than industry professionals or managers. While their input provides useful insights, it may not fully capture the priorities or practical concerns of decision-makers in real-world supply chains.

Second, practical and financial constraints limited the number of times each model could be tested. Running additional trials would have made it possible to check the consistency of the results and reduce variability. The study also did not include high-cost models like Claude Opus or open-source solutions such as Llama, which naturally narrowed the comparison.

Third, LLMs themselves can be unpredictable. Factors such as temperature settings or

updates from the provider can affect their responses in ways that are difficult to control, adding a degree of uncertainty to the results and limiting how broadly they can be applied.

The benchmark, although carefully designed, was built using a relatively small set of questions and scenarios from three academic institutions (Politecnico di Torino, RWTH Aachen University, and ESCP). These sources are high-quality, but they cannot fully represent the wide variety of decision-making situations that occur across different supply chains.

In addition, the study did not directly address some intrinsic limitations of LLMs, including the risk of biased outputs, the possibility of generating hallucinations, and the lack of transparency and explainability that often characterizes these systems. These issues have important implications for fairness, reliability, and accountability.

Finally, the statistical analysis relied on McNemar's test, which is appropriate for comparing classifiers on the same set of cases. However, its reliability depends on the number of discordant results, which was small in this study. This means that the high p-values do not show that the models with and without Chain-of-Thought are equivalent, they simply indicate that, with the available data, no significant differences could be detected. Stronger conclusions would require a larger sample or more varied cases.

## 6.3 Future Research Streams

Several avenues for further research emerge from this study.

A first line of research concerns cross-domain applications. Extending the methodology to contexts beyond supply chain management, such as project management or finance, would make it possible to verify the replicability of the benchmark and to assess the adaptability of the results to heterogeneous professional domains. At the same time, within supply chain management itself, the use of domain-specific datasets in key areas such as demand forecasting, inventory management, supply chain design, production planning and control, quality management, and supply chain risk management would allow for more targeted testing of model performance, highlighting strengths and weaknesses across different operational contexts.

A second area of exploration concerns the integration of additional question types and evaluation metrics, as outlined in Table 3.3, to broaden the scope of the benchmark.

A third stream combines statistical robustness with the study of model variability. Increasing the total number of benchmark questions would strengthen the reliability of significance

tests, while targeted analyses of generation parameters (such as temperature) and advanced prompting strategies (e.g., self-consistency) would provide deeper insight into the stability of LLM outputs. Together, these steps would help distinguish random fluctuations from structural variability in model behavior.

A fourth development lies in the design of dynamic benchmarks, where interaction between user and model plays a central role. Incorporating *Multi-turns* could bring evaluations closer to real-world usage scenarios, where iterative and adaptive exchanges are common.

Finally, expanding the stakeholder sample for the AHP survey remains essential. Including managers and practitioners would allow perceptions to be compared with those of graduate students, revealing whether the identified priorities align with operational needs in industry.

# References


Balloccu, S. et al. (Feb. 2024). *Leak, Cheat, Repeat: Data Contamination and Evaluation Malpractices in Closed-Source LLMs*. Tech. rep.

Banh, L. & Strobel, G. (2023). "Generative artificial intelligence". *Electronic Markets*, 33, 63. 10.1007/s12525-023-00680-1.

Bartz-Beielstein, T. et al. (Dec. 2020). *Benchmarking in Optimization: Best Practice and Open Issues*. Tech. rep.

Bengesi, S. et al. (Nov. 2023). "Advancements in Generative AI: A Comprehensive Review of GANs, GPT, Autoencoders, Diffusion Model, and Transformers". *IEEE Access*, 12, 69812–69837. ISSN: 21693536. 10.1109/ACCESS.2024.3397775.

Brand, J. et al. (2023). *Using LLMs for Market Research*. Tech. rep.

Brown, T. B. et al. (May 2020). "Language Models are Few-Shot Learners". *Advances in Neural Information Processing Systems*, 2020-December. ISSN: 10495258.

Brynjolfsson, E. & Mitchell, T. (Dec. 2017). "What can machine learning do? Workforce implications: Profound change is coming, but roles for humans remain". *Science*, 358, 1530–1534. ISSN: 10959203. 10.1126/SCIENCE.AAP8062/SUPPL_FILE/AAP8062-BRYNJOLFSSON-SM.PDF.

Busch, K. & Leopold, H. (Oct. 2024). *Towards a Benchmark for Large Language Models for Business Process Management Tasks*. Tech. rep.

Chang, Y. et al. (Mar. 2024). *A Survey on Evaluation of Large Language Models*. Tech. rep. 10.1145/3641289.

Chen, P. et al. (2024). *CoMM: Collaborative Multi-Agent, Multi-Reasoning-Path Prompting for Complex Problem Solving*. Tech. rep., 1720–1738.

Christiano, P. F. et al. (June 2017). "Deep reinforcement learning from human preferences". *Advances in Neural Information Processing Systems*, 2017-December, 4300–4308. ISSN: 10495258.

Cinkusz, K. et al. (Jan. 2024). "Cognitive Agents Powered by Large Language Models for Agile Software Project Management". *Electronics (Switzerland)*, 14. ISSN: 20799292. 10.3390/ELECTRONICS14010087.

Clavié, B. et al. (2023). "Large Language Models in the Workplace: A Case Study on Prompt Engineering for Job Type Classification". *Lecture Notes in Computer Science (LNCS)*. Vol. 13913. Springer, 3–17. 10.1007/978-3-031-35320-8_1.

Cunningham, P. et al. (2008). "Supervised Learning". *Cognitive Technologies*, 21–49. ISSN: 16112482. 10.1007/978-3-540-75171-7_2.

Dam, S. K. et al. (Nov. 2024). *A Complete Survey on LLM-based AI Chatbots*. Tech. rep.

Devlin, J. et al. (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". *arXiv preprint arXiv:1810.04805*.

Dol, M. & Geetha, A. (Aug. 2021). "A Learning Transition from Machine Learning to Deep Learning: A Survey". *Proceedings of the 2021 International Conference on Emerging Techniques in Computational Intelligence, ICETCI 2021*, 89–94. 10.1109/ICETCI51973.2021.9574066.

Dziri, N. et al. (2022). "On the Origin of Hallucinations in Conversational Models: Is it the Datasets or the Models?" *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2022)*. Association for Computational Linguistics, 5271–5285. 10.18653/v1/2022.naacl-main.387.

Einola, K. & Khoreva, V. (Jan. 2023). "Best friend or broken tool? Exploring the co-existence of humans and artificial intelligence in the workplace ecosystem". *Human Resource Management*, 62, 117–135. ISSN: 1099050X. 10.1002/HRM.22147;CSUBTYPE:STRING:SPECIAL;PAGE:STRING:ARTICLE/CHAPTER.

Eriksson, M. et al. (May 2025). *Can We Trust AI Benchmarks? An Interdisciplinary Review of Current Issues in AI Evaluation*. Tech. rep.

Ferrara, E. (Nov. 2023). "Should ChatGPT be Biased? Challenges and Risks of Bias in Large Language Models". *First Monday*, 28. 10.5210/fm.v28i11.13346.

Franke, S. et al. (2025). "Can ChatGPT Solve Undergraduate Exams from Warehousing Studies? An Investigation". *Computers*, 14:2, 52. ISSN: 2073-431X. 10.3390/computers14020052.

George, S. et al. (Feb. 2023). “A Review of ChatGPT AI’s Impact on Several Business Sectors”. *Partners Universal International Innovation Journal*, 1, 9–23. ISSN: 2583-9675. 10.5281/ZENODO.7644359.

Gignac, G. E. & Szodorai, E. T. (May 2024). “Defining intelligence: Bridging the gap between human and artificial perspectives”. *Intelligence*, 104, 101832. ISSN: 0160-2896. 10.1016/J.INTELL.2024.101832.

Goodfellow, I., Bengio, Y., et al. (Oct. 2017). “Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning”. *Genetic Programming and Evolvable Machines 2017 19:1*, 19, 305–307. ISSN: 1573-7632. 10.1007/S10710-017-9314-Z.

Goodfellow, I., Pouget-Abadie, J., et al. (Oct. 2020). “Generative adversarial networks”. *Communications of the ACM*, 63, 139–144. ISSN: 15577317. 10.1145/3422622.

Gu, J. et al. (2025). “A Survey on LLM-as-a-Judge”. *arXiv preprint arXiv:2411.15594*.

Guo, X. et al. (2025). “FinEval: A Chinese Financial Domain Knowledge Evaluation Benchmark for Large Language Models”. *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies*. Vol. 1. Long Papers, 6258–6292.

Haleem, A. et al. (Oct. 2022). “An era of ChatGPT as a significant futuristic support tool: A study on features, abilities, and challenges”. *BenchCouncil Transactions on Benchmarks, Standards and Evaluations*, 2, 100089. ISSN: 2772-4859. 10.1016/J.TBENCH.2023.100089.

Handley, L. (June 2023). *Supply chains: How AI could ‘remove all human touchpoints’*.

Hendrycks, D. et al. (2021). “Measuring Massive Multitask Language Understanding”. *arXiv preprint arXiv:2009.03300*.

Ho, X. et al. (2020). “Constructing A Multi-hop QA Dataset for Comprehensive Evaluation of Reasoning Steps”. *arXiv preprint arXiv:2011.01060*.

Hoek, R. V. et al. (Nov. 2022). *How Walmart Automated Supplier Negotiations*.

Horzyk, A. et al. (2023). “Construction and Training of Multi-Associative Graph Networks”. *Lecture Notes in Computer Science*. Vol. 14171 LNAI. Springer, Cham, 277–292. ISBN: 978-3-031-43418-1. 10.1007/978-3-031-43418-1_17.

Jackson, I. et al. (2024). “Generative artificial intelligence in supply chain and operations management: a capability-based framework for analysis and implementation”. *International*

*Journal of Production Research*, 62, 6120–6145. ISSN: 1366588X. 10.1080/00207543.2024.2309309.

Janiesch, C. et al. (Sept. 2021). "Machine learning and deep learning". *Electronic Markets*, 31, 685–695. ISSN: 14228890. 10.1007/S12525-021-00475-2/TABLES/2.

Ji, Z. et al. (Dec. 2023). "Survey of Hallucination in Natural Language Generation". *ACM Computing Surveys*, 55. ISSN: 15577341. 10.1145/3571730/ASSET/CC5D3792-8BC0-4675-8584-B507476E20EC/ASSETS/IMAGES/LARGE/CSUR-2022-0173-F01.JPG.

Jiang, Y. et al. (Mar. 2022). "Quo vadis artificial intelligence?" *Discover Artificial Intelligence 2022 2:1*, 2, 1–19. ISSN: 2731-0809. 10.1007/S44163-022-00022-8.

Kiela, D. et al. (Apr. 2021). *Dynabench: Rethinking Benchmarking in NLP*. Tech. rep.

Kojima, T. et al. (2022). "Large Language Models are Zero-Shot Reasoners". *Proceedings of the 36th International Conference on Neural Information Processing Systems (NeurIPS)*.

Kong, A. et al. (2023). "Better Zero-Shot Reasoning with Role-Play Prompting". *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2024)*. Vol. 1. Association for Computational Linguistics, 4099–4113. 10.18653/v1/2024.naacl-long.228.

Kshetri, N. et al. (Apr. 2024). "Generative artificial intelligence in marketing: Applications, opportunities, challenges, and research agenda". *International Journal of Information Management*, 75, 102716. ISSN: 0268-4012. 10.1016/J.IJINFOMGT.2023.102716.

Kumar, S. et al. (2023). "A Comprehensive Review on Sentiment Analysis: Tasks, Approaches and Applications". *arXiv preprint arXiv:2311.11250*.

Lecun, Y. et al. (May 2015). "Deep learning". *Nature*, 521, 436–444. ISSN: 14764687. 10.1038/NATURE14539;SUBJMETA=117,639,705;KWRD=COMPUTER+SCIENCE,MATHEMATICS+AND+COMPUTING.

Lewis & Crews (1985). "The evolution of benchmarking as a computer performance evaluation technique. MIS Quarterly , 7–16." *MIS Quarterly , 7–16.*

Li, B. et al. (July 2023). *Large Language Models for Supply Chain Optimization*. Tech. rep.

Li, Y. (Sept. 2023). "A Practical Survey on Zero-shot Prompt Design for In-context Learning". *International Conference Recent Advances in Natural Language Processing, RANLP*. Incoma Ltd, 641–647. 10.26615/978-954-452-092-2_069.

Li, Z. et al. (2024). “Optimizing Inventory Management using a Multi-Agent LLM System”. *Proceedings of The International Conference on Electronic Business*, 12–13.

Liang, P. et al. (2023). “Holistic Evaluation of Language Models”. *arXiv preprint arXiv:2211.09110*.

Lin, Y.-T. & Chen, Y.-N. (2023). “LLM-EVAL: Unified Multi-Dimensional Automatic Evaluation for Open-Domain Conversations with Large Language Models”. *Proceedings of the 5th Workshop on NLP for Conversational AI (NLP4ConvAI 2023)*, 47–58.

Liu, S. et al. (July 2023). “Using AI-generated suggestions from ChatGPT to optimize clinical decision support”. *Journal of the American Medical Informatics Association*, 30, 1237–1245. ISSN: 1527974X. 10.1093/JAMIA/OCAD072,.

Liu, Y., Cao, J., et al. (Feb. 2024). *Datasets for Large Language Models: A Comprehensive Survey*. Tech. rep.

Liu, Y., Khandagale, S., et al. (Nov. 2021). *Synthetic Benchmarks for Scientific Research in Explainable Machine Learning*. Tech. rep.

Liu, Y. L. et al. (June 2024). *ECBD: Evidence-Centered Benchmark Design for NLP*. Tech. rep.

Lunardi, R. et al. (2025). “On Robustness and Reliability of Benchmark-Based Evaluation of LLMs”. *arXiv preprint arXiv:2509.04013*.

Lv, Z. (Jan. 2023). “Generative artificial intelligence in the metaverse era”. *Cognitive Robotics*, 3, 208–217. ISSN: 2667-2413. 10.1016/J.COGR.2023.06.001.

McIntosh, T. R. et al. (Oct. 2024). *Inadequacies of Large Language Model Benchmarks in the Era of Generative Artificial Intelligence*. Tech. rep. 10.1109/TAI.2025.3569516.

Mehri, S. & Eskenazi, M. (2020). “USR: An Unsupervised and Reference Free Evaluation Metric for Dialog Generation”. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*. Association for Computational Linguistics, 681–707.

Meske, C. et al. (Dec. 2022). “Explainable and responsible artificial intelligence”. *Electronic Markets*, 32, 2103–2106. ISSN: 14228890. 10.1007/S12525-022-00607-2/METRICS.

Miao, X. et al. (June 2024). “Demystifying Data Management for Large Language Models”. *Proceedings of the ACM SIGMOD International Conference on Management of Data*. Association for Computing Machinery, 547–555. ISBN: 9798400704222. 10.1145/3626246.3654683.

Miller, J. K. & Tang, W. (May 2025). *Evaluating LLM Metrics Through Real-World Capabilities*. Tech. rep.

Miralles-González, P. et al. (2025). "Pushing the Boundary on Natural Language Inference". *arXiv preprint arXiv:2504.18376*.

Mishra, S. & Arunkumar, A. (2021). "How Robust are Model Rankings: A Leaderboard Customization Approach for Equitable Evaluation". *arXiv preprint arXiv:2106.05532*.

Mohri, M. et al. (2012). *Foundations of machine learning*. MIT Press, 414. ISBN: 9780262018258.

Moller, P. (Mar. 2023). *ChatGPT and the Like: AI in Logistics | DHL Freight*.

Mushtaq, A. et al. (2025). "WorldView-Bench: A Benchmark for Evaluating Global Cultural Perspectives in Large Language Models". *arXiv preprint arXiv:2505.09595*.

Pacchiardi, L. et al. (June 2025). "PredictaBoard: Benchmarking LLM Score Predictability". *Findings of the Association for Computational Linguistics: ACL 2025*. Association for Computational Linguistics, 15245–15266.

Pahuja, S. et al. (Jan. 2025). "Comprehensive Review of Generative artificial Intelligence: Mechanisms, Models, and Applications". *Procedia Computer Science*, 258, 3731–3740. ISSN: 1877-0509. 10.1016/J.PROCS.2025.04.628.

Parrish, A. et al. (2022). "BBQ: A Hand-Built Bias Benchmark for Question Answering". *arXiv preprint arXiv:2110.08193*.

Powers, D. M. W. (2015). *What the F-measure doesn't measure... Features, Flaws, Fallacies and Fixes*. Tech. rep.

Prieto, S. A. et al. (Jan. 2023). "Investigating the use of ChatGPT for the scheduling of construction projects". *Buildings*, 13. 10.3390/buildings13040857.

Quan, Y. & Liu, Z. (May 2024). *EconLogicQA: A Question-Answering Benchmark for Evaluating Large Language Models in Economic Sequential Reasoning*. Tech. rep. 10.48550/arXiv.2405.07938.

Reuel, A. et al. (2024). "BetterBench: Assessing AI Benchmarks, Uncovering Issues, and Establishing Best Practices". *Proceedings of the 38th International Conference on Neural Information Processing Systems*.

Reynolds, L. & McDonell, K. (Feb. 2021). "Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm". *Conference on Human Factors in Computing Systems*. Association for Computing Machinery. ISBN: 9781450380959. 10.1145/3411763.3451760.

Russell, S. J. & Norvig, P. (2016). *Artificial intelligence : a modern approach*. Pearson, 1132. ISBN: 9781292153964.

Savage, N. (May 2023). "Drug discovery companies are customizing ChatGPT: here's how". *Nature biotechnology*, 41, 585–586. ISSN: 15461696. 10.1038/S41587-023-01788-7;KWRD=LIFE+SCIENCES.

Schramowski, P. et al. (Mar. 2022). "Large pre-trained language models contain human-like biases of what is right and wrong to do". *Nature Machine Intelligence*, 4, 258–268. ISSN: 25225839. 10.1038/S42256-022-00458-8;SUBJMETA=117,4007,4009,639,705;KWRD=COMPUTER+SCIENCE,LANGUAGE+AND+LINGUISTICS.

Shen, Y. et al. (Apr. 2023). "ChatGPT and Other Large Language Models Are Double-edged Swords". *Radiology*, 307, 2023. ISSN: 15271315. 10.1148/RADIOL.230163/ASSET/IMAGES/LARGE/RADIOL.230163.FIG1.JPEG.

Shi, J. et al. (2025). "Optimization-based Prompt Injection Attack to LLM-as-a-Judge". *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security (CCS)*. Association for Computing Machinery, 660–674. 10.1145/3658644.3690291.

Sivarajkumar, S. et al. (2024). "An Empirical Evaluation of Prompting Strategies for Large Language Models in Zero-Shot Clinical Natural Language Processing: Algorithm Development and Validation Study". *JMIR Medical Informatics*, 12. ISSN: 22919694. 10.2196/55318,.

Skórnóg, D. & Kmiecik, M. (Oct. 2023). "Supporting the inventory management in the manufacturing company by ChatGPT". *LogForum*, Vol. 19, 535–554. ISSN: 1734-459X. 10.17270/J.LOG.2023.917.

Sokol, A. et al. (June 2025). *BenchmarkCards: Standardized Documentation for Large Language Model Benchmarks*. Tech. rep.

Stanovich, K. E. & West, R. F. (2000). "Individual differences in reasoning: Implications for the rationality debate?" *Behavioral and Brain Sciences*, 23, 645–726. ISSN: 0140525X. 10.1017/S0140525X00003435,.

Susarla, A. et al. (June 2023). "The Janus Effect of Generative AI: Charting the Path for Responsible Conduct of Scholarly Activities in Information Systems". *Information Systems Research*, 34, 399–408. ISSN: 15265536. 10.1287/ISRE.2023.ED.V34.N2;CSUBTYPE:STRING:PERIODICAL;TAXONOMY:TAXONOMY:ACM-PUBTYPE;PAGEGROUP:STRING:PUBLICATION.

Talby, D. (June 2025). *Why Leaderboards Fall Short in Measuring AI Model Value*. Online article.

Tyagi, K. et al. (Jan. 2022). "Unsupervised learning". *Artificial Intelligence and Machine Learning for EDGE Computing*, 33–52. 10.1016/B978-0-12-824054-0.00012-5 .

Vaswani, A. et al. (2017). "Attention Is All You Need". *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 10.5555/3295222.3295349.

Wang, A. et al. (2019). "GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding". *arXiv preprint arXiv:1804.07461*.

Wang, M. et al. (Sept. 2024). *Minstrel: Structural Prompt Generation with Multi-Agents Coordination for Non-AI Experts*. Tech. rep.

Wang, X. et al. (2022). "Self-Consistency Improves Chain of Thought Reasoning in Language Models". *Proceedings of the 11th International Conference on Learning Representations (ICLR 2023)*. International Conference on Learning Representations.

Wang, Y. et al. (2023). "Self-Instruct: Aligning Language Models with Self-Generated Instructions". *arXiv preprint arXiv:2212.10560*.

Wei, J. et al. (2022). "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models Chain-of-Thought Prompting". *Proceedings of the 36th International Conference on Neural Information Processing Systems*.

White, J. et al. (2023). "A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT". *Proceedings of the 30th Conference on Pattern Languages of Programs*. 10.5555/3721041.3721046.

Winston, P. H. (1993). *Artificial intelligence*. Addison-Wesley Pub. Co., 737. ISBN: 0201533774.

Yang, Z. et al. (2018). "HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering". *arXiv preprint arXiv:1809.09600*.

Yao, S., Yu, D., et al. (2023). "Tree of Thoughts: Deliberate Problem Solving with Large Language Models". *Proceedings of the 37th International Conference on Neural Information Processing Systems (NeurIPS)*. Vol. 36. Neural Information Processing Systems Foundation.

Yao, S., Zhao, J., et al. (2022). "ReAct: Synergizing Reasoning and Acting in Language Models". *Proceedings of the 11th International Conference on Learning Representations (ICLR 2023)*.

Zairi, M. & Leonardo, P. (1996). *Practical Benchmarking: The Complete Guide. Springer, pp. 22–27.*

Zhan, J. (Apr. 2022). *Call for establishing benchmark science and engineering*. Tech. rep.

Zhang, B. et al. (2023). *ZhuJiu: A Multi-dimensional, Multi-faceted Chinese Benchmark for Large Language Models*. Tech. rep., 479–494. 10.18653/V1/2023.EMNLP-DEMO.44 .

Zhao, J. et al. (2025). *Role-Play Paradox in Large Language Models: Reasoning Performance Gains and Ethical Dilemmas*. Tech. rep. -.

Zhen, Y. et al. (2024). "LLM-Project: Automated Engineering Task Planning via Generative AI and WBS Integration". *Proceeding of the 14th IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems, CYBER 2024*. Institute of Electrical and Electronics Engineers Inc., 605–610. ISBN: 9798331506056. 10.1109/CYBER63482.2024.10749328.

Zheng, L. et al. (2023). "Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena". *Proceedings of the 37th International Conference on Neural Information Processing Systems (NeurIPS)*. Vol. 36. Neural Information Processing Systems Foundation.

Zhong, W. et al. (2023). "AGIEval: A Human-Centric Benchmark for Evaluating Foundation Models". *arXiv preprint arXiv:2304.06364*.

Zhu, A. et al. (2024). "FanOutQA: A Multi-Hop, Multi-Document Question Answering Benchmark for Large Language Models". *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*. Vol. 2, 18–37.

Zhuang, J. (Apr. 2023). *Introducing the Instacart Plugin for ChatGPT*.