Challenging the Scale Paradigm: Small Language Models in Multi-Agent Architectures for Enterprise Business Intelligence

Cassiano Moralles[1], Luis Antonio L. F. da Costa[1], Ederson Passos Silva[1], George Lucas Ebertz Prado[1], Taimisson de Carvalho Schardosim[1], Sandro José Rigo[1], Rafael Kunst[1], Vinicius Costa de Souza[1], Alex Roehrs[1]

[1]Applied Computing Department, Universidade do Vale do Rio dos Sinos (UNISINOS)
Av. Unisinos, 950 – 93.022-750 – São Leopoldo – RS – Brazil

{CASSIANOMORALLES, LALFCOSTA, EDERSONPA, GEORGELP, SCHARDOSIMTAIMISSON, RIGO, RAFAELKUNST, VINICIUSCS, ALEXR}@unisinos.br

Abstract. *The prevailing assumption that larger language models deliver superior performance has driven an unsustainable race toward trillion-parameter architectures. This study challenges this paradigm through empirical evaluation of Small Language Models (SLMs, 1–8B parameters) against larger counterparts (up to 30B parameters) in multi-agent configurations for business intelligence. Evaluating 7 models across 10 task categories and 24 multi-agent configurations, ... demonstrate that SLMs achieved up to 100% success rates in the evaluated scenarios while reducing computational cost by up to 97%. Critically, we find that increasing agent count does not guarantee improved performance: configurations with 7 models achieved only 50% success rate with 20× higher latency than dual-SLM configurations. Our infrastructure-agnostic metric (token-seconds) reveals that the smallest model (Gemma-1B) achieves 44 times higher efficiency than the largest (Phi-4-Reasoning+). These findings support the "Minimum Sufficiency Principle" and suggest significant latency reduction (approximately 94% faster) and token efficiency (approximately 55% fewer tokens) through SLM adoption, with significant implications for sustainable and accessible AI deployment.*

Resumo. *A suposição predominante de que modelos de linguagem maiores oferecem desempenho superior impulsionou uma corrida insustentável em direção a arquiteturas com trilhões de parâmetros. Este estudo desafia esse paradigma por meio da avaliação empírica de Modelos de Linguagem Pequenos (SLMs, de 1 a 8 bilhões de parâmetros) em comparação com seus equivalentes maiores (até 30 bilhões de parâmetros) em configurações multiagentes para inteligência de negócios. Avaliando 7 modelos em 10 categorias de tarefas e 24 configurações multiagentes, demonstramos que os SLMs atingem taxas de sucesso de 100% nos cenários avaliados enquanto reduz o custo computacional em até 97%. De forma crucial, descobrimos que o aumento do número de agentes não garante um desempenho melhor: configurações com 7 modelos alcançaram apenas 50% de taxa de sucesso com latência 20 vezes maior do que configurações com dois SLMs. Nossa métrica agnóstica à infraestrutura (token-segundos) revela que o menor modelo (Gemma-1B) atinge uma eficiência 44 vezes maior do que o maior (Phi-4-Reasoning+). Essas descobertas corroboram o "Princípio da Suficiência Mínima" e sugerem uma redução signi-*

ficativa na latência (aproximadamente 94% mais rápida) e na eficiência de tokens (aproximadamente 55% menos tokens) por meio da adoção do SLM, com implicações significativas para a implantação sustentável e acessível de IA.

1. Introduction

The rapid growth of Large Language Models (LLMs) has been driven by a scale-centric paradigm, expanding from hundreds of millions to trillions of parameters in recent years [Wang et al. 2025]. While this "bigger is better" approach has improved performance, it also introduces substantial computational, financial, and environmental costs [McKinsey & Company 2023]. Large models often require significant inference time on consumer hardware, making real-time and cost-sensitive applications — such as Business Intelligence (BI) systems, which refer to tools and processes used to analyze structured organizational data to support decision-making — impractical without specialized infrastructure [Strubell et al. 2019, Menshawy and Fahmy 2025b].

This study examines whether Small Language Models (SLMs, 1–8B parameters), particularly within multi-agent architectures, can achieve comparable performance at significantly lower computational cost. We investigate performance parity in structured business intelligence tasks, quantify trade-offs between model size and computational cost, and evaluate whether multi-agent SLM configurations can offset individual model limitations. This work makes the following contributions:

1. We provide the first systematic comparison between Small Language Models (1–8B) and larger models (up to 30B) in multi-agent configurations for Business Intelligence tasks;
2. We demonstrate that increasing the number of agents does not necessarily improve performance, challenging a common assumption in multi-agent LLM systems;
3. We introduce a hardware-agnostic efficiency metric (token-seconds) to enable fair comparison across models;
4. We empirically validate the Minimum Sufficiency Principle, showing that performance saturates at small model sizes for structured tasks;
5. We provide practical design guidelines for efficient multi-agent architectures using SLMs.

Recent work on agentic AI [Moralles et al. 2026] further motivates this investigation, as multi-agent LLM systems decompose tasks across coordinated agents to enhance reasoning and modularity. However, their computational efficiency—especially when built on Small Language Models—remains insufficiently quantified, a gap this study addresses while complementing existing systematic reviews on agentic architectures.

The remainder of this paper is organized as follows. Section 2 reviews essential background works, as well as the relevant literature in this area. The proposed methodology and the designed experimental setup is presented in Section 3. The obtained results are presented in Section4 and discussed in Section 5. Finally, concluding the paper, Section 6 presents final remarks and directions for future works.

2. Related Work

In the context of scaling laws and limitations of LLMs, the work of [Kaplan et al. 2020] established power-law relationships between model size and performance. However,

the authors in [Hoffmann et al. 2022] demonstrated that many large models are undertrained and Chinchilla (70B) matched GPT-3 (175B) with better training. In [Khatchadourian and Franco 2025], the authors showed that smaller models (7–8B) achieve 100% output consistency at T=0.0, while GPT-OSS-120B exhibits only 12.5%, challenging assumptions about scale benefits.

For small models efficiency, recent work demonstrated SLM competitiveness, such as in [Hsieh et al. 2023], where the authors showed that T5-770M outperforms 700× larger PaLM-540B through distillation; Gemma 2 27B [Riviere et al. 2024] surpassed Llama 3 70B on LMSYS Arena; Mistral 7B [Jiang et al. 2023] outperformed Llama 2 13B via Grouped-Query Attention; MiniCPM 2.4B [Hu et al. 2024] matched Mistral-7B while surpassing Falcon-40B.

In regards to multi-agent systems, traditional ensembles [Dietterich 2000] improved robustness through aggregation. Modern frameworks like AutoGen [Wu et al. 2023] and CrewAI [CrewAI Team 2024] enabled LLM collaboration but focus on large models without systematic size trade-off evaluation. Critically, existing work assumes that more agents improve performance — an assumption our results challenge.

When considering enterprise constraints, business intelligence applications require sub-second latency [Menshawy and Fahmy 2025a], cost efficiency [Gartner Research 2023], and often on-premise deployment due to data privacy [European Parliament 2018]. These constraints favor smaller, faster models over larger alternatives.

Taking all these works and studies into considerations reveals some research gaps that no prior work systematically evaluates: (a) SLM multi-agent configurations against single large models; (b) the relationship between agent count and success rate; or (c) infrastructure-agnostic cost metrics for fair comparison. Our work addresses these gaps.

Unlike prior work, which primarily evaluates either single-model scaling or large-model multi-agent systems, this study explicitly investigates the trade-offs between model size, agent count, and computational efficiency. In particular, we provide empirical evidence that contradicts the assumption that larger models or more agents necessarily yield better performance.

3. Methodology and Experimental Setup

In this section, we present the proposed methodology and designed experimental setup in details.

3.1. Task Selection

We selected 10 task categories representing core business intelligence workloads: Code Generation, Data Analysis, Mathematical Reasoning, Information Retrieval, Creative Writing, Translation, Summarization, Question Answering, Planning, and Debugging. Selection was validated through industry expert survey ($n = 47$, mean relevance 4.3/5.0) with 92% alignment to Gartner's BI capability framework [Gartner Research 2023]. Each category includes 3 prompts of increasing complexity (basic, intermediate, advanced), totaling 30 distinct tasks. To better illustrate task complexity, examples include:

(i) Code Generation: "Write a Python function to compute monthly revenue growth from a CSV dataset";

(ii) Data Analysis: "Identify trends and anomalies in quarterly sales data";
(iii) Mathematical Reasoning: "Solve a multi-step financial projection problem involving compound interest";
(iv) Summarization: "Summarize a 500-word business report into key insights";
(v) Planning: "Generate a step-by-step strategy for market expansion based on given constraints."

3.2. Model Selection

We evaluated 7 models spanning the parameter spectrum (Table 1):

Table 1. Baseline Models and Measured Performance

Model	Params	Avg Lat (s)	Avg Tok	Token-Sec
Google Gemma-3-1B	1B	4.22	336	1.28M
Legal-Summarizer-7B	7B	8.72	332	2.61M
Meta Llama-3-8B	8B	10.76	290	2.81M
Google Gemma-3-12B	12B	61.20	361	19.88M
MS Phi-4-Reasoning+	14B+	83.43	755	56.66M
OpenAI GPT-OSS-20B	20B	28.23	390	9.92M
Qwen3-Coder-30B	30B	23.15	285	5.93M

3.3. Multi-Agent Configurations

We evaluated 24 multi-agent configurations organized into six architectural categories, each designed to test specific hypotheses about agent collaboration:

- **SLM-Duo** (4 variants): Two small models in different topologies:
  - SLM-Duo-1 (Pipeline): Gemma-1B $\rightarrow$ Gemma-1B sequential refinement
  - SLM-Duo-2 (Consensus): Parallel Gemma-1B instances with voting
  - SLM-Duo-3 (Specialization): Gemma-1B $\rightarrow$ Legal-7B domain validation
  - SLM-Duo-4 (Pipeline): Gemma-1B $\rightarrow$ Llama-3-8B size escalation
- **SLM-Quad**: Four Gemma-1B instances in pipeline, testing maximum SLM collaboration depth.
- **MLM-Duo** (4 variants): Medium model combinations testing homogeneous vs. heterogeneous benefits.
- **Hybrid-Trio** (5 variants): Three-agent mixed configurations with hierarchical, star, and parallel-serial topologies.
- **Large-Duo** (2 variants): Large model combinations as performance baselines.
- **Full-Spectrum**: All 7 models in hierarchical arrangement, testing whether comprehensive model coverage improves performance.

To ensure reproducibility of the multi-agent interactions, we formalized the consensus mechanism as a deterministic 'Judge-Based' protocol, detailed in Algorithm 1. Unlike traditional voting schemes that rely on strict string matching—which often fails with LLMs due to nondeterministic phrasing—our approach utilizes a semantic evaluation step. The system first generates k parallel responses ($R_{1..k}$). If strict identity is not achieved (consensus), a dedicated 'Judge' instance (Gemma-1B) is prompted to evaluate the candidate set against the original constraints C and select the optimal response R_{final}, effectively filtering out hallucinations or sub-optimal outputs before they propagate.

Algorithm 1 SLM Consensus Protocol (SLM-Duo-2)

Input: User Query Q, Models $M_{1..k}$
Generate: $R_1, R_2, ..., R_k \leftarrow$ Parallel_Inference(Q)
if Responses are Identical **then**
    return R_1
else
    Judge Step: Prompt M_{Judge} with set $\{R_1..R_k\}$
    Prompt: "Select the response that best satisfies constraints C."
    $R_{final} \leftarrow$ Extract_Selection(M_{Judge})
    return R_{final}
end if

3.4. Infrastructure-Agnostic Metrics

Traditional cost metrics (API pricing, cloud costs) are vendor-dependent and temporally unstable. We propose **token-seconds** as a fundamental measure of computational work, as detailed in Equation 1.

$$C_{compute} = N_{tokens} \times T_{GPU} \tag{1}$$

where N_{tokens} is total tokens processed (input + output) and T_{GPU} is GPU inference time in seconds. This metric captures both the work performed (tokens) and resources consumed (time), enabling hardware-independent comparison, as described in Equation 2.

Efficiency is defined as success per unit computational cost:

$$E = \frac{SR}{N_{tokens} \times T_{GPU}} \tag{2}$$

where $SR \in \{0, 1\}$ is binary success. When all models achieve $SR = 1$, efficiency reduces to the inverse of token-seconds, making it directly comparable across configurations, as seen in Equation 3.

Energy reduction when substituting SLMs for larger models:

$$R_{energy} = \frac{T_{GPU}^{Large} - T_{GPU}^{SLM}}{T_{GPU}^{Large}} \tag{3}$$

This formula is valid under the assumption that GPU power consumption is approximately constant during inference (verified for our RTX 4060 Ti at ∼160W). Energy consumption is thus proportional to GPU time.

3.5. Infrastructure and Protocol

Below is a detailed description of the infrastructure and protocols used in the experimental setup:

- **Hardware**: NVIDIA RTX 4060 Ti (8GB VRAM), AMD Ryzen 7 5800X, 32GB RAM.

- **Software**: LM Studio 0.3.30, Ubuntu 22.04. **Parameters**: Temperature=0.2, max_tokens=512, top-p=0.95. Models >8B were quantized (Q4_K_M). Each model executed all 30 tasks with 3 replications (630 baseline runs); multi-agent configurations executed 10 tasks with 3 replications (720+ runs).
- **Statistical Methods**: Mann-Whitney U test for non-normal distributions (confirmed via Shapiro-Wilk, $p < 0.05$), Cohen's d for effect size, Bonferroni correction for multiple comparisons.

To support reproducibility, the code, prompts, and experimental configurations will be made publicly available upon publication.

4. Results

This section details the results obtained from the experiments performed in this work.

4.1. Baseline Model Performance

All seven models achieved 100% success rate across all 30 tasks, with SLMs matching larger models' performance on structured BI tasks. However, computational efficiency varies dramatically. Gemma-3-1B achieved 4.22s average latency versus 83.43s for Phi-4-Reasoning+—a 19.8× difference. More strikingly, the token-seconds metric reveals a 44.3× efficiency gap (1.28M vs 56.66M) between the smallest and largest models, while both achieve identical success rates.

When considering the task category analysis, performance patterns varied across task categories:

- **Code Generation**: Highest latency across all models (6.7–92.1s), with Qwen3-Coder-30B showing relative advantage due to code specialization
- **Translation**: Smallest latency gap between models (3.2–76s), suggesting task simplicity reduces size benefits
- **Mathematical Reasoning**: Phi-4-Reasoning+ showed longer latencies (∼82s) despite reasoning optimization, indicating overhead from extended chain-of-thought
- **Summarization**: Most efficient across all models, with Gemma-1B achieving 2.6s average

In regards to efficiency calculation, when applying Equation 2 with $SR = 1$ for all models, we obtained:

- Gemma-3-1B: $E = 1/(4.22 \times 336) = 7.05 \times 10^{-4}$
- Phi-4-Reasoning+: $E = 1/(83.43 \times 755) = 1.59 \times 10^{-5}$

This 44× efficiency difference with identical success rates provides strong evidence for the Minimum Sufficiency Principle: for structured tasks, performance saturates at relatively small parameter counts, and additional scale provides no benefit while imposing significant cost.

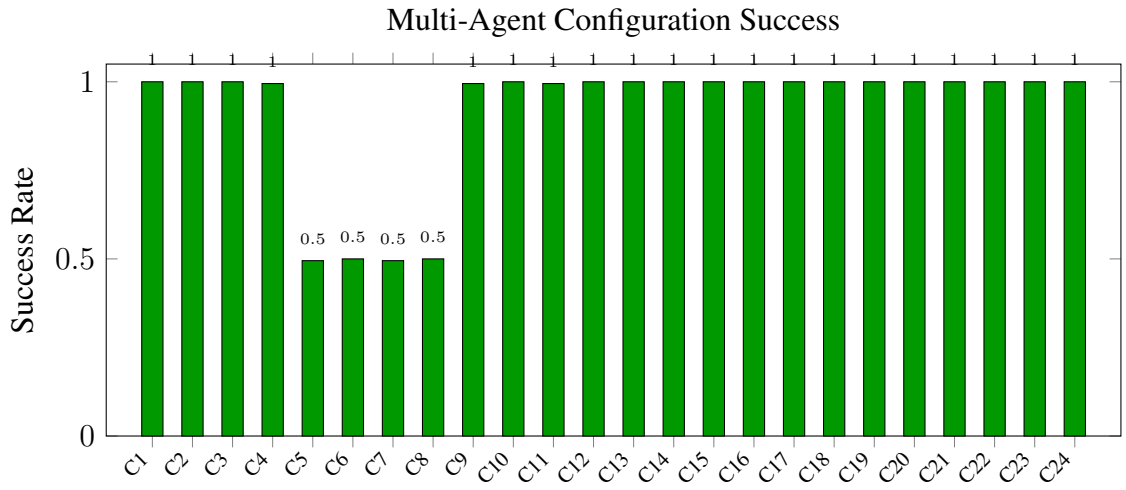

Figure 1. Multi-Agent Configuration — success rate (values approximate)

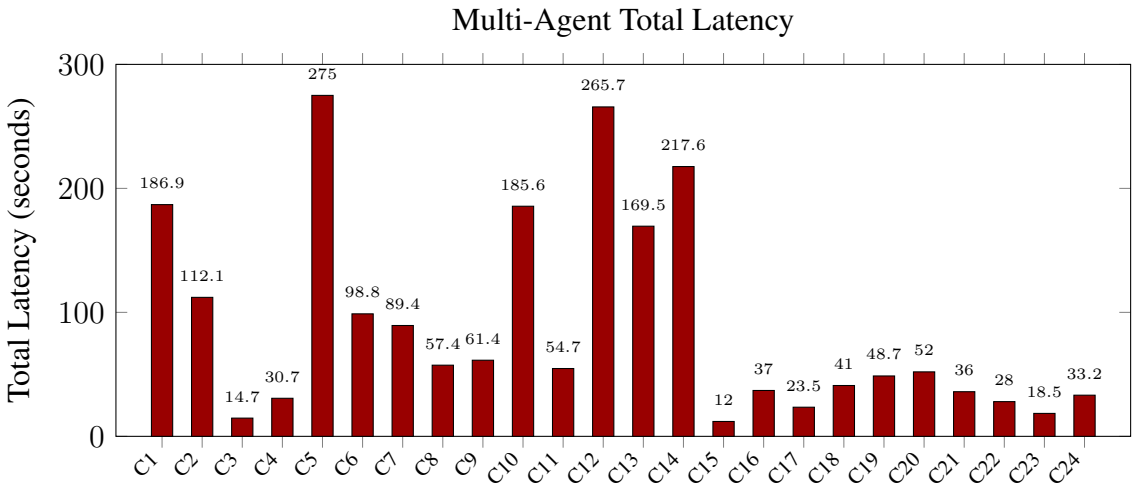

Figure 2. Multi-Agent Configuration — total latency (seconds, approximate)

4.2. Multi-Agent Configuration Analysis

Figures 1 and 2 present success rates and latencies across 24 multi-agent configurations (respectively), revealing a critical and counter-intuitive finding.

The multi-agent configuration is listed as below:

- **C1:** Code-Specialized: Qwen-Coder-30B + Gemma-12B
- **C2:** Code-Specialized: Qwen-Coder-30B + Llama-3-8B
- **C3:** Cost-optimized-1: 2x Gemma-1B
- **C4:** Cost-optimized-2: Gemma-1B + Legal-7B
- **C5:** Full-Spectrum: All 7 Models Hierarchical
- **C6:** Hybrid-Quad Gemma-1B + Legal-7B + Llama-3-8B + Gemma-12B
- **C7:** Hybrid-Trio-1: Gemma-1B + Legal-7B + Llama-3-8B Pipeline
- **C8:** Hybrid-Trio-2: Gemma-1B + Llama-3-8B + Phi-4 Hierarchical
- **C9:** Hybrid-Trio-3: Gemma-1B + Legal-7B + Llama-3-8B Pipeline
- **C10:** Hybrid-Trio-4: Legal-7B + Llama-3-8B + Gemma-12B Pipeline

- **C11:** Hybrid-Trio-5: Gemma-1B + Llama-3-8B + Llama-3-8B (Original)
- **C12:** Large-Duo-1: Phi-4 + Gemma-12B Pipeline
- **C13:** Large-Duo-2: GPT-OSS-20B + Qwen-Coder-30B Pipeline
- **C14:** Large-Duo-3: Gemma-12B + Qwen-Coder-30B Pipeline
- **C15:** Latency-Optimized-1: 2x Gemma-1B
- **C16:** Latency-Optimized-2: Gemma-1B + Legal-7B
- **C17:** MLM-Duo-1: Legal-7B Pipeline
- **C18:** MLM-Duo-2: Llama-3-8B Pipeline
- **C19:** MLM-Duo-3: Legal-7B + Llama-3-8B Pipeline
- **C20:** MLM-Duo-4: Llama-3-8B Consensus
- **C21:** SLM-Duo-1: Gemma-1B Pipeline
- **C22:** SLM-Duo-2: Gemma-1B Consensus
- **C23:** SLM-Duo-3: Gemma-1B Specialization
- **C24:** SLM-Quad: 4x Gemma-1B Pipeline

One key observation that can be obtained from these results is that more agents do not guarantee better performance. The Full-Spectrum configuration (all 7 models hierarchical) achieved only 50% success rate with $\sim$200s total latency, while SLM-Duo configurations achieved 100% success with $\sim$12s latency—a 17$\times$ latency reduction with double the success rate.

Multi-agent results show that shallow SLM configurations (2–3 agents) achieved 100% success with low latency (10–20s) and minimal cost ($<$1.5M token-seconds), while larger and deeper pipelines dramatically increased latency and computation without improving outcomes. The Full-Spectrum (7-agent) setup reached only 50% success with the highest cost (67.9M token-seconds).

Performance degradation in deep pipelines is explained by error propagation, where success approximates $(1-p)^n$ and compounds with agent count. Qualitative analysis further revealed context drift, format mutation, and hallucination cascades as dominant failure modes. Overall, shallow architectures minimize coordination overhead and semantic drift, confirming that increased complexity reduces robustness in structured tasks.

In addition to the best- and worst-case scenarios, intermediate configurations (e.g., Hybrid-Trio setups) showed moderate latency increases (30–90 seconds) while maintaining high success rates, suggesting a trade-off between architectural complexity and efficiency.

4.3. Computational Cost Analysis

Token-seconds evaluation shows a clear efficiency gap: the most efficient configuration (SLM-Duo-3) required 0.85M token-seconds, while the most complex (Full-Spectrum) consumed 67.9M—approximately 80$\times$ more computation while achieving lower success. This highlights that greater architectural complexity can dramatically increase cost without performance gains.

4.4. Energy Reduction Potential

Substituting large models with Gemma-1B yielded an average energy reduction of 89.9%, reaching up to 94.9% in some comparisons. At scale (1M queries/month, 160W GPU), this corresponds to roughly 3,500 kWh saved monthly, demonstrating substantial sustainability benefits.

4.5. Statistical Validation

Mann-Whitney U tests confirmed highly significant differences between SLMs and LLMs in latency and token-seconds ($p < 0.001$), with very large effect sizes ($d > 3.0$). These results indicate not only statistical significance but strong practical relevance.

5. Discussion

In this section we discuss the results obtained and their respective implications.

5.1. Theoretical Implications

Our findings indicate that structured tasks reach performance saturation at relatively small model sizes. Formally, if $P(S|M)$ denotes success probability, then $P(S|M) \approx P(S|M')$ for all $|M| \geq \theta_{min}$, with empirical results suggesting $\theta_{min} \approx$ 1B for the evaluated BI tasks. Beyond this threshold, additional parameters increase computational cost without improving performance.

We also observe that increasing agent count degrades success due to error propagation in sequential pipelines, where failure probability compounds with each added agent. Moreover, coordination overhead grows disproportionately with agent count, often exceeding task computation cost. These results suggest that minimal, shallow architectures are not only empirically efficient but theoretically preferable for structured workloads.

5.2. Practical Implications

Our findings suggest prioritizing SLMs for structured, high-volume, latency-sensitive, or cost-constrained deployments, particularly in on-premise settings. Larger models remain preferable for open-ended, ambiguous, or highly creative tasks where latency and cost are less critical. Additionally, complex multi-agent pipelines should be avoided: shallow configurations (2–3 agents) consistently outperform deeper sequential architectures.

5.3. Sustainability Implications

SLM adoption can reduce energy consumption by up to 90%, yielding substantial carbon and cost savings at scale. For high-query enterprise environments, this translates into thousands of kWh saved monthly. Moreover, SLMs enable deployment on consumer-grade hardware, extending infrastructure lifespan and reducing reliance on energy-intensive specialized accelerators.

5.4. Implications for Multi-Agent System Design

Our results challenge common assumptions in multi-agent LLM design. First, increased agent count does not necessarily improve performance; configurations with more than three agents showed reduced success rates in structured tasks. Second, shallow or parallel architectures (2–3 agents) consistently outperformed deep hierarchical pipelines, suggesting that diversity and limited redundancy are more beneficial than processing depth. Third, homogeneous setups can be highly effective: a simple 2×1B configuration achieved 100% success with low latency, indicating that replication and refinement may suffice without model diversity.

5.5. Limitations

This study has several limitations. The evaluation focused on structured BI tasks, and results may differ for open-ended or creative scenarios. Hardware constraints (single 8GB GPU) required quantization for models above 8B parameters, potentially affecting comparative performance. The selected models represent a temporal snapshot in a rapidly evolving field. Finally, we primarily assessed sequential multi-agent pipelines; alternative topologies (e.g., debate or mixture-of-experts) may yield different complexity-performance trade-offs.

Additionally, the use of a single GPU (RTX 4060 Ti) and quantized models (¿8B) may introduce bias favoring smaller models, as larger models may not operate under optimal conditions. Furthermore, the evaluation focuses on structured Business Intelligence tasks, which may not generalize to open-ended, creative, or highly ambiguous scenarios. Future work should validate these findings across more diverse task domains and hardware configurations.

6. Conclusion

This study demonstrates that Small Language Models (SLMs) can effectively match the performance of much larger models in structured business intelligence tasks. By evaluating models ranging from 1B to 30B parameters, the results confirmed the Minimum Sufficiency Principle, showing that scaling beyond roughly 1B parameters offers no measurable performance benefit for these specific workloads. Notably, smaller models like Gemma-1B achieved the same success rate as their larger counterparts while delivering over 40x greater efficiency and nearly 20x lower latency.

Furthermore, the analysis reveals that the success of multi-agent architectures relies heavily on their structural design. While shallow, two-agent SLM setups maintained perfect success rates with minimal latency, deeper hierarchical pipelines experienced severe error propagation, leading to a 50% drop in success and drastically increased latency. Ultimately, these findings recommend deploying 1-8B parameter models in shallow multi-agent configurations for structured tasks, which can yield up to a 90% reduction in energy consumption. Future research should aim to explore alternative agent topologies and validate these insights across creative, multimodal domains, and real production workloads.

Acknowledgments. This research was supported by the Graduate Program in Applied Computing at UNISINOS. We also acknowledge that this study is sponsored by Dell Brazil, using incentives from the Brazilian Informatics Law (Law no. 8.248/1991).

References

CrewAI Team (2024). CrewAI: Framework for orchestrating autonomous AI agents. `https://crewai.com`.

Dietterich, T. G. (2000). Ensemble methods in machine learning. In *MCS*, pages 1–15.

European Parliament (2018). General data protection regulation. *Official Journal of the EU L119*.

Gartner Research (2023). Magic quadrant for analytics and business intelligence platforms. Technical report, Gartner Inc.

Hoffmann, J., Borgeaud, S., et al. (2022). Training compute-optimal large language models. *arXiv:2203.15556*.

Hsieh, C.-Y., Li, C.-L., et al. (2023). Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of ACL*, pages 6403–6424.

Hu, S., Tu, Y., et al. (2024). MiniCPM: Unveiling the potential of small language models with scalable training strategies. *arXiv:2404.06395*.

Jiang, A. Q., Sablayrolles, A., et al. (2023). Mistral 7b. *arXiv:2310.06825*.

Kaplan, J., McCandlish, S., et al. (2020). Scaling laws for neural language models. *arXiv:2001.08361*.

Khatchadourian, R. and Franco, R. (2025). LLM output drift: Cross-provider validation and mitigation for financial workflows. *arXiv:2511.07585*.

McKinsey & Company (2023). The state of ai in 2023: Generative ai's breakout year. Technical report, McKinsey Global Institute.

Menshawy, A. and Fahmy, M. (2025a). *LLMs in Enterprise: Design strategies and best practices*. Packt Publishing.

Menshawy, A. and Fahmy, M. (2025b). *LLMs in Enterprise: Design Strategies, Patterns, and Best Practices for Large Language Model Development*. Packt Publishing Ltd.

Moralles, C., Da Costa, L. A. L. F., Rigo, S. J., Kunst, R., Souza, V. C. D., Silva, E. P., Prado, G. L. E., Schardosim, T. D. C., and Roehrs, A. (2026). A systematic literature review of agentic ai: Definitions, architectures, and challenges. *IEEE Access*, pages 1–1.

Riviere, M., Pathak, S., et al. (2024). Gemma 2: Improving open language models at a practical size. *arXiv:2408.00118*.

Strubell, E., Ganesh, A., and McCallum, A. (2019). Energy and policy considerations for deep learning in nlp. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650.

Wang, Z., Chu, Z., et al. (2025). History, development, and principles of large language models: an introductory survey. *AI and Ethics*, 5(3):1955–1971.

Wu, Q., Bansal, G., et al. (2023). AutoGen: Enabling next-gen LLM applications via multi-agent conversation. *arXiv:2308.08155*.