Multi-Agent Coordination Strategies vs Retrieval-Augmented Generation in LLMs: A Comparative Evaluation

Irina Radeva *, Ivan Popchev, Lyubka Doukovska, Miroslava Dimitrova

Posted Date: 18 November 2025

doi: 10.20944/preprints202511.1168.v1

Keywords: retrieval-augmented generation (RAG); multi-agent coordination strategies; large language models (LLMs); comparative evaluation; performance evaluation

 doi:10.20944/preprints202511.1168.v1

Article

Multi-Agent Coordination Strategies vs Retrieval-Augmented Generation in LLMs: A Comparative Evaluation

Irina Radeva [1,*], Ivan Popchev [2], Lyubka Doukovska [3] and Miroslava Dimitrova [4]

[1] Intelligent Systems Department, Institute of Information and Communication Technologies, Bulgarian Academy of Sciences, 1113 Sofia, Bulgaria
[2] Bulgarian Academy of Sciences, 1040 Sofia, Bulgaria
[3] Faculty of Informatics and Mathematics, Trakia University Stara Zagora, Bulgaria
[4] Intelligent Systems Department, Institute of Information and Communication Technologies, Bulgarian Academy of Sciences, 1113 Sofia, Bulgaria
* Correspondence: irina.radeva@iict.bas.bg

Abstract

This paper evaluates multi-agent coordination strategies against retrieval-augmented generation for 7-8B open-source models. Four coordination strategies (collaborative, sequential, competitive and hierarchical) were evaluated across three open-source models: Mistral 7B, Llama 3.1 8B and Granite 3.2 8B. The study determined whether multi-agent reasoning enhances retrieval-augmented generation performance. The evaluation employed 100 question-answer pairs. In total, 2,000 model-question evaluations were conducted. Performance was assessed using Composite Performance Score (CPS) and Threshold-aware Composite Performance Score (T-CPS), two metrics developed to aggregate nine dimensions spanning lexical overlap, semantic similarity, and linguistic quality. Results revealed that 87.5% of multi-agent configurations underperformed baseline systems, with coordination overhead identified as the primary limiting factor. Llama 3.1 8B tolerated Sequential and Hierarchical coordination with minimal degradation, while Granite 3.2 8B and Mistral 7B showed severe degradation across all strategies. Collaborative coordination failed universally despite highest output consistency. These findings suggest that single-agent baselines may be preferable for most deployment scenarios under similar conditions. Future research should explore following developments: evaluation of role-specific prompts, investigation of advanced consensus methods, exploration of adaptive systems for strategy selection, and joint tuning of retrieval thresholds and coordination strategies.

Keywords: retrieval-augmented generation (RAG); multi-agent coordination strategies; large language models (LLMs); comparative evaluation; performance evaluation

1. Introduction

Retrieval-Augmented Generation (RAG) enhances large language model capabilities by integrating external knowledge retrieval into the generation process [1]. RAG systems have been explored across various application contexts [2–5], with ongoing challenges in threshold configuration and architectural design [6].

Multi-agent coordination has been proposed as a mechanism to improve LLM reasoning quality through distributed processing and consensus mechanisms [7–9]. Multi-agent coordination strategies refer to the organisational frameworks and interaction protocols that govern how multiple agents work together, communicate and adjust their actions within these systems [10]. This study evaluates the four primary coordination architectures—collaborative (peer-to-peer deliberation), sequential

 doi:10.20944/preprints202511.1168.v1

(pipeline refinement), competitive (selection-based) and hierarchical (manager-worker)—which represent distinct approaches to distributed processing.

However, the effectiveness of multi-agent coordination strategies when applied to RAG systems remains largely unexamined. Existing multi-agent evaluations typically compare coordination approaches against simple single-agent prompting baselines rather than against RAG systems that already incorporate external knowledge retrieval. Whether coordination mechanisms provide sufficient performance improvements to justify their computational overhead when added to functioning RAG systems has not been systematically investigated.

1.2. Research Objectives

Multi-agent systems for large language models have generated considerable attention in recent research, with claims of improved reasoning through agent collaboration [7,8,11].

This study addresses whether adding multi-agent coordination to configured RAG systems enhances or degrades performance compared to single-agent baselines.

Four specific research objectives were formulated.

1. Comparative performance assessment was conducted. Performance of four coordination strategies (collaborative, sequential, competitive, hierarchical) was evaluated across three open-source models (Mistral 7B, Llama 3.1 8B, Granite 3.2 8B). Whether multi-agent configurations outperform, match, or underperform calibrated single-agent RAG baselines was determined.
2. Degradation source identification was performed. The relative contributions of coordination overhead versus retrieval fragmentation to performance changes were isolated. Independent retrieval and shared context retrieval configurations (Granite-SCR) were compared to decompose these effects quantitatively.
3. Model-strategy interaction analysis was undertaken. Whether coordination effectiveness depends on model architecture was investigated. Differential responses to identical coordination protocols across different model families were characterized.
4. Consistency-performance trade-offs were assessed. Whether multi-agent coordination affects output variability alongside mean performance was examined. The Threshold-aware Composite Performance Score (T-CPS) was employed to evaluate stability-performance trade-offs simultaneously.

Evidence-based deployment guidance is provided to practitioners based on these findings. Three questions are addressed. Conditions under which multi-agent coordination is justified were identified if such conditions exist. Conditions under which single-agent RAG baselines demonstrated superior performance were identified. Performance patterns were characterized across coordination strategies.

1.3. Approach

Four multi-agent coordination strategies (collaborative, sequential, competitive, hierarchical) were evaluated across three 7-8B open-source models (Mistral 7B [12], Llama 3.1 8B [13], Granite 3.2 8B [14,15]) using 100 domain-specific question-answer pairs from the Climate-Smart Agriculture Sourcebook [16]. Performance was assessed using Composite Performance Score (CPS) [6] and Threshold-aware CPS (T-CPS) within the PaSSER evaluation framework [17], comparing all multi-agent configurations against tuned single-agent RAG baselines. Experimental methodology and performance metrics are detailed in Section 3.

The remainder of this paper is structured as follows: Section 2 reviews related work in retrieval-augmented generation and multi-agent language model systems. Section 3 describes the experimental methodology, PaSSER framework extensions, coordination strategies, evaluation corpus and performance metrics. Section 4 presents results across performance, stability, and efficiency dimensions with statistical validation. Section 5 discusses coordination performance

 doi:10.20944/preprints202511.1168.v1

patterns, computational trade-offs and comparisons with prior literature. Section 6 concludes with key findings, deployment guidelines, limitations, and future research directions.

2. Related Work

Retrieval-Augmented Generation systems enhance language model capabilities by incorporating external knowledge retrieval into the generation process. The foundational RAG approach was introduced in 2020 [1], establishing a paradigm that has since been evaluated across diverse application contexts [3,5]. A comprehensive review traces the evolution of RAG systems from their roots in information retrieval to current modular architectures supporting dynamic reasoning and real-time knowledge integration [18]. Recent work has identified architectural challenges and failure modes in RAG deployments [2], leading to developments in adaptive retrieval control [4], complex reasoning support [19], knowledge graph integration [20], and domain-specific embedding configurations [21]. Recent work has explored adaptive retrieval strategies that dynamically adjust retrieval parameters based on query characteristics [22]. Such frameworks feature heterogeneous weighted graph indices and adaptive planning that selects appropriate retrieval strategies based on query features, demonstrating improved compatibility with both small and large language models. These advances demonstrate that RAG systems can achieve substantial performance through architectural refinement and parameter tuning. However, existing research focuses primarily on single-agent RAG configurations. Whether multi-agent coordination mechanisms provide additional benefits beyond well-configured single-agent RAG remains an open question.

Multi-agent systems for large language models employ distributed processing to address complex tasks through agent collaboration. Comprehensive surveys characterize coordination mechanisms, organizational structures, and application domains [23–25], identifying four primary coordination architectures. Collaborative (peer-to-peer) strategies enable agents to work cooperatively toward shared objectives with specialized roles, with code generation frameworks demonstrating improvements through explicit communication protocols [7,26]. Competitive (debate-based) approaches leverage adversarial interaction between agents to enhance factual accuracy through iterative refinement and disagreement-driven error correction [8]. Hierarchical (manager-worker) architectures delegate subtasks from manager agents to specialized workers, excelling in complex workflows requiring strategic planning and resource allocation [27]. Sequential (pipeline) strategies implement multi-stage refinement where agents progressively improve outputs through iterative processing, effective for tasks requiring diverse perspectives [9,28]. Recent applications demonstrate multi-agent systems' potential across diverse domains. Cognitive agents powered by LLMs have been integrated within the Scaled Agile Framework for software project management, demonstrating advanced task delegation and inter-agent communication capabilities [29]. Hierarchical multi-agent architectures for power grid anomaly detection have shown that lower-layer agents handling specialized monitoring tasks combined with upper-layer coordinators performing multimodal feature fusion and global decision-making can achieve high precision through distributed collaboration [30].

Recent studies highlight the importance of coordination structure selection based on task characteristics [23], with adaptive strategies outperforming fixed architectures in specific domains. However, most evaluations compare multi-agent approaches against simple single-agent prompting baselines rather than against retrieval-augmented generation systems that already incorporate external knowledge. This evaluation gap obscures whether coordination overhead is justified when added to functioning RAG systems.

The selection of embedding models significantly impacts RAG system performance. Systematic evaluation of embedding model similarity using Centered Kernel Alignment and retrieval result comparison across five BEIR datasets reveals distinct performance patterns [31]: (1) models from the same family exhibit high embedding similarity while cross-family similarity varies substantially, (2) optimal similarity thresholds differ significantly across model families, with some benefiting from high selectivity (0.90-0.95) while others perform better with moderate thresholds (0.55-0.70), and (3)

 doi:10.20944/preprints202511.1168.v1

top-k retrieval similarity shows high variance at low k values, stabilizing only for k ≥ 10. The Massive Text Embedding Benchmark (MTEB) provides standardized evaluation across 58 datasets spanning eight task categories [32]. However, MTEB primarily assesses single-query retrieval performance, not addressing multi-turn interactions or agent-based scenarios that emerge in multi-agent RAG deployments.

Evaluation methodologies for LLM performance continue to evolve beyond traditional benchmarks. Systematic performance evaluation through strategic decision-making tasks reveals significant variations across models under consistent evaluation protocols [33]. Customizable evaluation frameworks that accommodate domain-specific quality criteria address limitations of standardized benchmarks that fail to capture specialized application requirements [34].

Embedding models exhibit distinct characteristics that influence threshold behavior. Distance metric properties vary across embedding spaces [35], semantic granularity affects optimal threshold selection [31,32], and different pre-training objectives produce distinct embedding space geometries [36,37]. These model-specific characteristics inform baseline RAG configuration in the present study. The present study addresses the identified evaluation gap through systematic comparison of coordination strategies against optimized RAG baselines, determining when multi-agent coordination enhances or degrades system performance.

3. Methods

This section describes the experimental method, including the PaSSER framework extensions, multi-agent coordination strategies, evaluation corpus, and performance metrics.

3.1. Experimental Infrastructure

The PaSSER framework was originally designed for systematic evaluation of RAG threshold configurations [17]. The present study extends PaSSER to support multi-agent reasoning while maintaining backward compatibility with single-agent RAG evaluation. The extended PaSSER system comprises three primary components. A Python Flask API server manages LLM inference, multi-agent coordination protocols, and evaluation logging. A React frontend enables configuration, real-time monitoring, and visualization. A vector-indexed knowledge base uses sentence-transformers embeddings for efficient retrieval.

Three open-source models with comparable parameter counts were selected for evaluation. Mistral 7B (version 7b-instruct-v0.3) is a 7.3 billion parameter instruction-tuned model released under Apache 2.0 license. Llama 3.1 8B (version 8b-instruct) is Meta's 8 billion parameter instruction-tuned model with extended context support. Granite 3.2 8B (version 8b-instruct) is IBM's 8 billion parameter enterprise-focused instruction-tuned model. Model selection prioritized comparable parameter counts (7-8B) to isolate architectural and training methodology effects, open-source availability under permissive licenses, active maintenance and documented performance characteristics, and instruction-tuning suitable for question-answering tasks. All models were obtained from the Hugging Face model repository.

Experiments were conducted on two hardware configurations to assess model performance across different computational environments. Configuration A (Llama 3.1 8B experiments) used iMac 24-inch with Apple M1 chip featuring 8-core CPU and 10-core GPU, 16 GB unified memory, 512 GB SSD storage, running macOS 14. The M1's unified memory architecture enables efficient model-memory data transfer through shared memory space between CPU and GPU cores. Configuration B (Mistral 7B and Granite 3.2 8B experiments) used dual Intel Xeon processor server with 128 GB DDR4 RAM, 20 TB HDD storage, running Ubuntu 24.04 LTS. No GPU acceleration was employed for these experiments. All models were deployed locally. Local deployment eliminated network latency variability and external service dependencies, ensuring consistent experimental conditions. These hardware configurations represent modest computational resources. This demonstrates that the evaluation methodology does not require specialized infrastructure. However, this setup reflects available on-premise resources and does not represent typical GPU-accelerated production

 doi:10.20944/preprints202511.1168.v1

environments. The heterogeneous hardware configuration across models precludes direct comparison of absolute performance metrics such as inference latency.

Figure 1 presents the complete PaSSER framework architecture. The three-layer design enables systematic evaluation of multi-agent coordination strategies through modular components for model deployment, retrieval-augmented generation, performance assessment, and blockchain-based result verification.

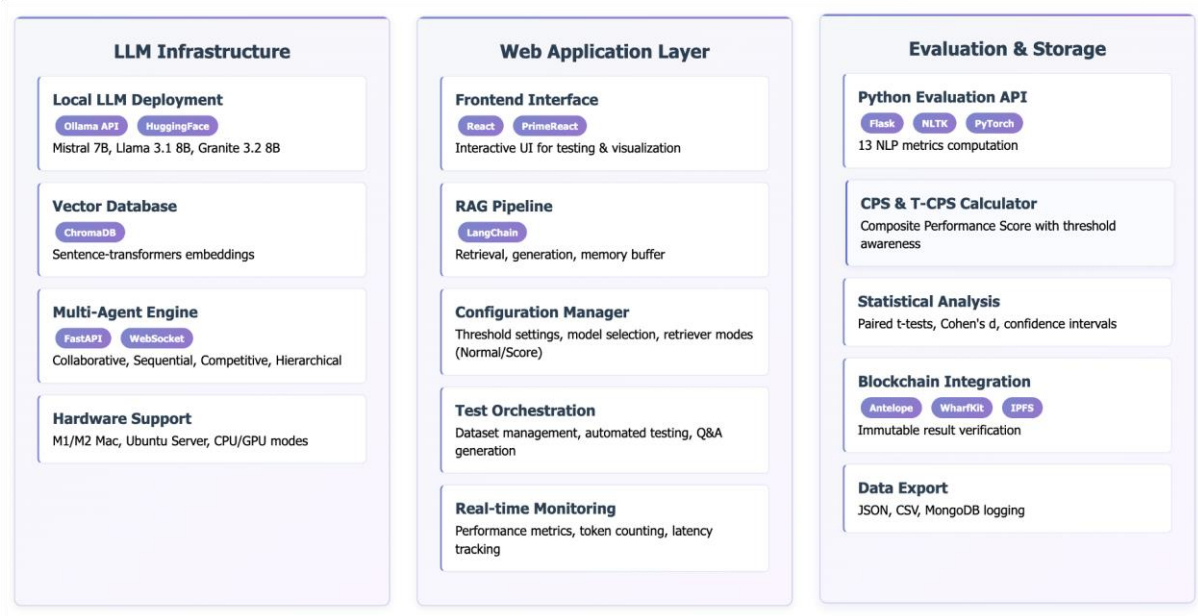

Figure 1. PaSSER Framework Architecture. The framework comprises three integrated layers: (1) LLM Infrastructure layer supporting local deployment of Mistral 7B, Llama 3.1 8B, and Granite 3.2 8B models with ChromaDB vector database and multi-agent coordination engine implementing four strategies (Collaborative, Sequential, Competitive, Hierarchical); (2) Web Application layer providing React/PrimeReact frontend with LangChain RAG pipeline, configuration management, and real-time monitoring; (3) Evaluation & Storage layer featuring Python-based metric computation (Flask API with NLTK, PyTorch), CPS/T-CPS calculation, statistical analysis, and blockchain integration via Antelope for immutable result verification.

3.2. Multi-Agent Coordination Strategies and Experimental Design

The evaluation dataset consisted of 100 question-answer pairs derived from the "Climate Smart Agriculture Source Book" [16]. Each model-strategy combination was evaluated on the same question set.

Figure 2 illustrates the seven-step experimental workflow, from initial vector store creation to final result verification. This workflow was applied consistently to all model-strategy combinations.

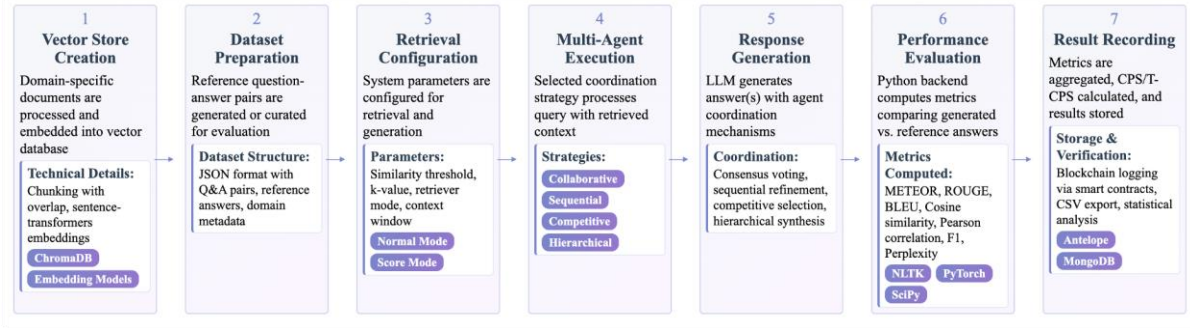

Figure 2. Testing Workflow in the PaSSER Framework. The seven-step experimental protocol encompassing: (1) vector store creation with domain-specific document embeddings via ChromaDB, (2) dataset preparation with

 doi:10.20944/preprints202511.1168.v1

reference question-answer pairs, (3) retrieval configuration establishing similarity thresholds and context parameters, (4) multi-agent execution implementing coordination strategies (Collaborative, Sequential, Competitive, Hierarchical), (5) response generation with agent coordination mechanisms (as Python API), (6) performance evaluation computing metrics, and (7) result recording.

Four coordination strategies were implemented with the following operational protocols:

3.2.1. Collaborative Strategy

Implementation: All three agents processed each query independently in parallel using identical retrieved context and system prompts. Each agent generated a complete response with confidence estimation based on internal coherence metrics. The final output aggregated all agent responses into a comprehensive summary that preserved individual perspectives while highlighting areas of consensus and divergence.

Consensus Mechanism: No explicit selection or filtering occurred; all agent outputs contributed equally to the final response. Confidence scores were averaged to produce an aggregate consensus score. This approach assumes that multiple independent perspectives enhance coverage and reduce individual agent blind spots.

3.2.2. Competitive Strategy

Implementation: Three agents processed each query independently in parallel using identical retrieved context. Each agent assigned a confidence score (0-100 scale) to its generated response based on semantic coherence, factual consistency with context, and answer completeness.

Selection Mechanism: The agent with the highest confidence score was designated as the winner. Only this agent's response was returned as the final answer, with the winner's confidence serving as the consensus score. Disagreement was quantified by calculating variance in confidence scores across agents (normalized to 0-1 range).

3.2.3. Hierarchical Strategy

Implementation: Two specialist agents processed the query first, generating independent preliminary analyses. A third agent acting as manager received truncated summaries (first 400 characters) of specialist outputs along with the original query and retrieved context. The manager agent synthesized specialist insights into a unified final response with expanded context window (1.5× standard context size) to accommodate specialist inputs.

Coordination Protocol: Sequential two-phase processing: (1) parallel specialist analysis (2 agents), followed by (2) manager synthesis (1 agent). The manager could selectively integrate, reconcile, or prioritize specialist perspectives based on relevance and quality assessment.

3.2.4. Sequential Strategy

Implementation: Agents processed the query in defined sequential order (Agent 1 → Agent 2 → Agent 3). Each agent received: (1) the original query, (2) retrieved context passages, and (3) the previous agent's complete response (for Agent 2 and Agent 3). Context accumulation enabled progressive refinement, with each agent building upon predecessors' work.

Refinement Protocol: Agent 1 generated an initial response from retrieved context. Agent 2 reviewed Agent 1's response and refined or extended it based on identified gaps, errors, or opportunities for elaboration. Agent 3 performed final review and refinement, producing the definitive system response. This iterative approach aimed to progressively improve response quality through multiple processing passes.

3.2.5. Retrieval Context Configuration

 doi:10.20944/preprints202511.1168.v1

Two retrieval context delivery modes were evaluated to isolate coordination effects from retrieval fragmentation:

- Independent Retrieval (Granite 3.2 8B, Mistral 7B, Llama 3.1 8B): Each agent independently queries the vector database and retrieves its own document subset based on similarity ranking. This configuration allows agents to access potentially different retrieved passages, introducing retrieval diversity but also potential inconsistency in available context.
- Shared Context Retrieval (Granite-SCR): All agents receive identical retrieved document sets extracted through a single vector database query. This configuration ensures uniform input context across all agents, eliminating retrieval fragmentation as a confounding variable and isolating pure coordination effects.

Both configurations maintain identical similarity thresholds (0.95), chunk sizes, embedding models (sentence-transformers with Mistral), and k-value settings (k=5). The comparison enables assessment of whether multi-agent performance degradation stems from coordination overhead or from inconsistent retrieval across agents.

Each model-strategy combination underwent evaluation on all 100 test instances, yielding 1,500 total evaluations. Three models, four multi-agent strategies, and 100 tests generated 1,200 multi-agent evaluations. Three models, one RAG baseline, and 100 tests generated 300 baseline evaluations. Test execution followed randomized instance ordering within each configuration to mitigate temporal drift effects. For each test, the system recorded all nine component metrics for CPS calculation, response generation time (milliseconds), token counts (prompt tokens, completion tokens, total tokens), and coordination metrics for multi-agent configurations only (agent participation count, consensus rounds, disagreement rate, consensus score).

3.3. Performance Evaluation Framework

A comprehensive 9-metric evaluation suite was employed, capturing complementary quality dimensions: lexical fidelity (BLEU, ROUGE-1, ROUGE-L), semantic alignment (METEOR, cosine similarity, Pearson correlation), answer quality (F1 score), and fluency (Laplace and Lidstone perplexity).

METEOR measures text quality through precision-recall alignment accounting for exact matches, stems, synonyms, and word order. Implementation used nltk.translate.meteor_score.

ROUGE measures n-gram overlap between generated and reference texts. Two variants were used: ROUGE-2 (bigram overlap) and ROUGE-L (longest common subsequence), both computing F1-scores combining precision and recall. Implementation used the rouge library's get_scores function.

BERTScore evaluates contextual alignment using transformer-based embeddings, computing token-level cosine similarities aggregated into F1 scores. Implementation used the bert-score library

B-RT (BERT-based Reference-free Text Evaluation) provides reference-free assessment across coherence, consistency, fluency, and relevance using BERT representations. Two variants were used: B-RT.average (overall quality) and B-RT.fluency (grammatical naturalness).

F1 Score combines precision and recall as their harmonic mean: $F1 = 2 \times (Precision \times Recall)/(Precision + Recall)$. Implementation used scikit-learn for token-level agreement measurement.

Perplexity quantifies model uncertainty in predicting text sequences, with lower values indicating better predictive performance. Two smoothing variants were used: Laplace (add-one, α = 1.0) and Lidstone (parameterized, α = 0.5) to handle zero-probability n-grams. Implementation used the nltk.lm module.

All metrics were computed using the PaSSER framework and aggregated across 100 test instances per configuration to provide mean and standard deviation values for statistical analysis. Detailed implementation is available in GitHub repository [38] (backEnd.py and maBackEnd.py scripts).

 doi:10.20944/preprints202511.1168.v1

Multiple assessment dimensions were aggregated into a unified metric enabling systematic comparison across configurations. The Composite Performance Score (CPS) provides this unified framework by integrating nine evaluation metrics spanning lexical overlap, semantic similarity, predictive accuracy, and linguistic quality into a single weighted score [6].

The CPS formulation addresses three fundamental challenges in multi-metric evaluation: (1) heterogeneous metric scales requiring normalization, (2) opposing directionality where some metrics improve with higher values while others improve with lower values, and (3) differential importance of evaluation dimensions necessitating weighted aggregation.

For each query q evaluated under model m with coordination strategy s, the CPS is computed as:

$$CPS_q^{(m,s)} = \sum_{i=1}^{n} w_i \left[\frac{d_i\left(m_{i,q} - min_i\right)}{max_i - min_i} + \frac{1 - d_i}{2} \right] \quad (1)$$

where:

- $d_i \in -1, +1$ indicates the polarity of metric i: $d_i = +1$ if higher values indicate better performance, $d_i = -1$ if lower values indicate better performance
- w_i is the assigned weight for metric i, with $\sum_{i=1}^{n} w_i = 1$

The normalization procedure ensures all metrics are scaled to the [0,1] range while preserving their performance directionality. For metrics where higher values indicate better performance ($d_i = +1$), the standard min-max normalization is applied:

$$\text{Normalized value} = \frac{m_i - min_i}{max_i - min_i}$$

For metrics where lower values indicate better performance ($d_i = -1$), specifically the perplexity metrics in this study, the normalization is inverted to ensure that superior performance (lower perplexity) maps to higher normalized scores:

$$\text{Normalized value} = \frac{max_i - m_i}{max_i - min_i}$$

The aggregated CPS for model m with strategy s across Q queries is:

$$\mu_{m,s} = \frac{1}{Q} \sum_{q=1}^{Q} CPS_q^{(m,s)} \quad (2)$$

This aggregated score provides the basis for comparing mean performance across different coordination strategies and model architectures. Metric weights in the CPS calculation (Equation 1) prioritize content accuracy and completeness: F1 score (20%), METEOR (15%), and BLEU (15%) for information retrieval quality. Cosine similarity and Pearson correlation (10% each) assess semantic relevance. ROUGE-1, ROUGE-L, Laplace perplexity, and Lidstone perplexity (7.5% each) balance lexical overlap and language modeling quality. These weights reflect established priorities in RAG evaluation where content accuracy and semantic fidelity are primary concerns [6]. The selection of evaluation metrics for LLM systems requires careful consideration of task-specific requirements and model capabilities. Recent evaluations demonstrate that even state-of-the-art LLMs achieve only 60% accuracy on complex reasoning tasks, emphasizing the importance of comprehensive evaluation frameworks that capture both performance and reasoning reliability [39].

While mean CPS provides comprehensive performance assessment across evaluation instances, output variability was also assessed. This consideration is particularly salient for multi-agent systems, which introduce multiple sources of output variability beyond those present in single-agent configurations. Multi-agent coordination mechanisms can amplify output inconsistency through several pathways: (1) voting ambiguities in collaborative strategies where small changes in individual agent outputs can flip majority decisions, producing substantially different final responses; (2) sequential error propagation where early-stage agent mistakes compound through refinement chains; (3) evaluator selection variability in competitive strategies where the meta-agent's quality assessment may be inconsistent across similar response sets; and (4) coordinator synthesis

 doi:10.20944/preprints202511.1168.v1

inconsistency in hierarchical approaches where the integration of multiple perspectives introduces additional decision points with associated variability.

T-CPS integrates mean performance with consistency through a reward-penalty structure, penalizing high variability configurations.

The T-CPS for model m with strategy s is computed as:

$$T - CPS_{m,s} = \mu_{m,s} \cdot \left(1 + \alpha \cdot \left(1 - CV_{m,s}\right)\right) - \beta \cdot CV_{m,s}^2 \quad (3)$$

where:

- $\mu_{m,s}$ is the mean CPS for model m with strategy s
- $CV_{m,s} = \frac{\sigma_{m,s}}{\mu_{m,s}}$ is the coefficient of variation, with $\sigma_{m,s}$ denoting the standard deviation of CPS across evaluation instances
- α defines the reward coefficient for stable configurations
- β defines the penalty coefficient for high variability
- The coefficient of variation normalizes variability assessment by expressing standard deviation as a proportion of the mean, enabling fair comparison across configurations with different baseline performance levels. Lower CV values indicate more consistent behavior across queries and evaluation runs.

The reward term $\mu_{m,s} \cdot \left(1 + \alpha \cdot \left(1 - CV_{m,s}\right)\right)$ increases scores for stable configurations. When $CV_{m,s} = 0$ (perfect consistency), the configuration receives the maximum reward of $\mu_{m,s} \cdot (1 + \alpha)$. As variability increases, the reward diminishes, reaching $\mu_{m,s}$ when $CV_{m,s} = 1$ (standard deviation equals mean).

The penalty term $\beta \cdot CV_{m,s}^2$ applies a quadratic penalty for high variability, ensuring that unstable configurations are appropriately downweighted. The quadratic form provides progressive penalization: configurations with moderate variability receive modest penalties, while highly unstable configurations (large CV) are substantially penalized.

Parameter values $\alpha = 0.1$ and $\beta = 0.05$ follow standard practices in machine learning [40,41]. These values ensure that both average quality and stability matter in the evaluation. They do not over-penalize normal variability in language model outputs [42].

When $\alpha = 0.1$, a perfectly consistent system ($CV = 0$) gets a 10% reward. Systems with some variability get smaller rewards. When $\beta = 0.05$, systems with low variability get small penalties. Systems with high variability get larger penalties because the penalty uses CV^2. These parameters favor systems with good average performance and reasonable consistency. This matches what is needed for deployed language models [43].

This formulation is adapted from our previous work [6] on threshold selection in RAG systems. In the present multi-agent context, T-CPS serves to identify coordination strategies that achieve strong mean performance while maintaining acceptable output stability—a dual objective critical for practical system deployment.

The T-CPS framework is particularly valuable in multi-agent evaluation because coordination mechanisms inherently introduce additional variability sources beyond single-agent sampling randomness. By simultaneously evaluating performance and stability, T-CPS identifies configurations that not only achieve high average quality but also deliver predictable, reliable outputs—characteristics essential for user-facing applications where inconsistent behavior undermines trust and usability.

3.4. Baseline Configuration and Statistical Analysis

Single-agent RAG baseline configurations were established through systematic threshold evaluation for each model. The selection procedure employed the same evaluation framework—PaSSER with a 369-question Climate-Smart Agriculture dataset and comprehensive metric assessment.

 doi:10.20944/preprints202511.1168.v1

Each model was evaluated across similarity thresholds ranging from 0.50 to 0.95 in increments of 0.05. For each threshold configuration, both CPS and T-CPS were computed across all 369 questions. Statistical significance was assessed through paired t-tests comparing each threshold configuration against an untuned retrieval baseline without explicit threshold filtering (α = 0.05). Performance improvements and consistency measures (coefficient of variation) were evaluated to identify threshold configurations that balance both mean performance gains and output stability.

Baseline thresholds were selected by identifying configurations that maximized T-CPS while maintaining statistically significant performance improvements. This approach ensures baselines represent not only strong mean performance but also reliable, consistent outputs across diverse queries—a critical requirement for practical RAG system deployment.

The evaluation identified threshold 0.95 for Mistral 7B (T-CPS: 0.5911, +5.37% improvement) and Granite 3.2 8B (T-CPS: 0.5622, +1.26% improvement), and threshold 0.90 for Llama 3.1 8B (T-CPS: 0.5495, +1.87% improvement). These threshold-calibrated configurations establish the single-agent RAG baselines against which multi-agent coordination strategies are compared.

Table 1 presents the baseline selection analysis, showing the performance characteristics of optimal thresholds alongside the baseline (no threshold) configuration and alternative competitive thresholds for comparison.

Table 1. Baseline Configuration Selection for Multi-Agent Comparison.

Model	Configuration	Threshold	CPS	T-CPS	CV	CPS Δ%	T-CPS Δ%	Balance Score	Selection
Mistral 7B	Baseline	—	0.5181	0.5610	0.1501	—	—	—	
Mistral 7B	SELECTED	0.95	0.5448	0.5911	0.1339	+5.16%	+5.37%	40.11	Max T-CPS
Mistral 7B	Alternative 1	0.90	0.5332	0.5787	0.1312	+2.93%	+3.16%	24.07	
Mistral 7B	Alternative 2	0.70	0.5321	0.5776	0.1283	+2.70%	+2.97%	23.13	
Granite 3.2 8B	Baseline	—	0.5112	0.5552	0.1243	—	—	—	
Granite 3.2 8B	SELECTED	0.95	0.5176	0.5622	0.1240	+1.25%	+1.26%	10.12	Max T-CPS
Granite 3.2 8B	Alternative 1	0.80	0.5172	0.5619	0.1220	+1.18%	+1.20%	9.85	
Granite 3.2 8B	Alternative 2	0.75	0.5168	0.5613	0.1239	+1.10%	+1.10%	8.91	
Llama 3.1 8B	Baseline	—	0.4982	0.5394	0.1497	—	—	—	
Llama 3.1 8B	SELECTED	0.90	0.5074	0.5495	0.1479	+1.85%	+1.87%	12.65	Max T-CPS
Llama 3.1 8B	Alternative 1	0.55	0.5046	0.5478	0.1286	+1.30%	+1.55%	12.05	
Llama 3.1 8B	Alternative 2	0.80	0.5039	0.5450	0.1589	+1.15%	+1.04%	6.57	

Notes: All configurations evaluated on 369 question-answer pairs from Climate-Smart Agriculture dataset; CPS: Composite Performance Score (see Section 3.3 for metric details); T-CPS: Threshold-aware CPS with stability

 doi:10.20944/preprints202511.1168.v1

reward ($\alpha = 0.1$) and variability penalty ($\beta = 0.05$); CV: Coefficient of Variation (σ/μ; lower values indicate more consistent performance); Balance Score: $(T - CPS\ \Delta\%)\ /\ CV$ (measures stability-performance trade-off); Selection criterion: Maximum T-CPS among configurations with positive improvements and acceptable stability.

Figure 3 compares threshold configurations, confirming that selected baselines optimize both mean performance and consistency. Selected baselines (gold-highlighted) consistently achieve superior T-CPS performance: Mistral 7B demonstrates the largest improvement potential (+5.37% at threshold 0.95), while Granite 3.2 8B (+1.26% at threshold 0.95) and Llama 3.1 8B (+1.87% at threshold 0.90) show more modest but reliable gains. The close alignment between CPS and T-CPS bars confirms that selected thresholds optimize mean performance without compromising output consistency, establishing rigorous reference points for multi-agent comparison.

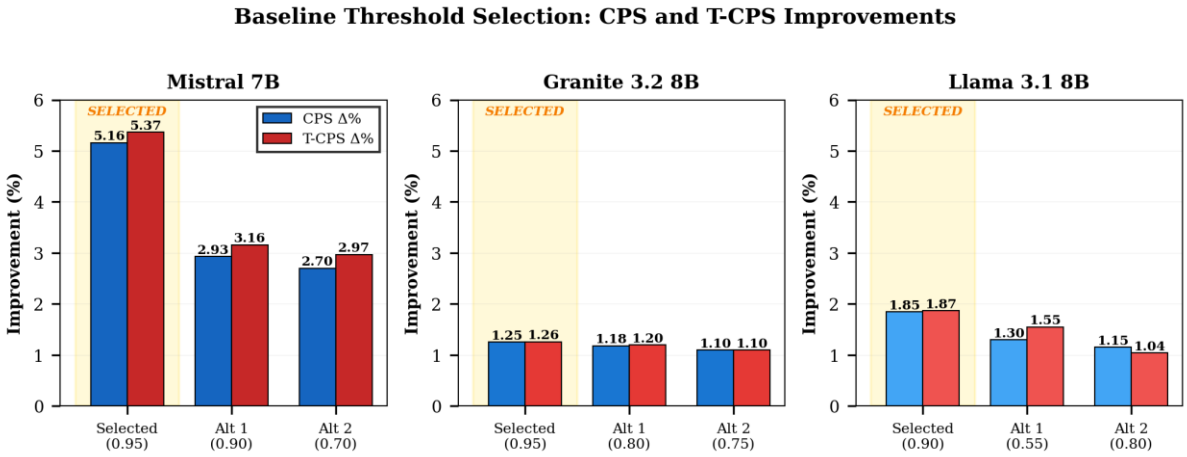

Figure 3. Baseline Threshold Selection: Performance Comparison Across Threshold Configurations. Grouped bar charts comparing CPS improvement (blue bars) and T-CPS improvement (red bars) for three threshold configurations per model. Selected baselines (highlighted in gold): Mistral 7B at threshold 0.95 (CPS: +5.16%, T-CPS: +5.37%), Granite 3.2 8B at threshold 0.95 (CPS: +1.25%, T-CPS: +1.26%), Llama 3.1 8B at threshold 0.90 (CPS: +1.85%, T-CPS: +1.87%). Alternative configurations (Alt 1, Alt 2) show competing threshold choices with lower T-CPS improvements. Values displayed on bars indicate percentage improvement relative to untuned baseline. Selected thresholds maximize T-CPS while maintaining statistically significant CPS improvements, balancing mean performance with output consistency.

The selected baseline configurations demonstrate statistically significant improvements over untuned retrieval while maintaining stable, consistent output quality. Mistral 7B achieves the largest improvement magnitude (+5.37% T-CPS) with strong balance between performance gains and stability (Balance Score: 40.11). Granite 3.2 8B exhibits the most stable performance profile (CV: 0.1240) with modest but reliable improvements (+1.26% T-CPS). Llama 3.1 8B demonstrates intermediate characteristics, balancing moderate improvement (+1.87% T-CPS) with acceptable consistency (CV: 0.1479).

These threshold-calibrated baselines establish rigorous comparison conditions for evaluating multi-agent coordination effectiveness, ensuring that any observed performance differences reflect coordination mechanisms rather than suboptimal baseline configurations.

Performance comparisons employed percentage change metrics relative to RAG baselines:

$$CPS_{chage\ (\%)} = \left(\frac{(CPS_{MA}CP\ -\ CPS_{RAG})}{CPS_{RAG}}\right) \times\ 100 \tag{4}$$

Positive values indicate multi-agent improvements. Negative values indicate degradation. Similar calculations applied to T-CPS. Statistical significance testing used paired t-tests, Cohen's d effect sizes, and 95% confidence intervals. Configuration-level aggregation computed mean CPS, standard deviation, and coefficient of variation across all 100 test instances for each model-strategy

 doi:10.20944/preprints202511.1168.v1

combination. Strategy-level analysis averaged results across the three models to identify general coordination patterns independent of specific model characteristics.

4. Results

4.1. Experimental Overview

The multi-agent evaluation used a 100-question subset drawn from the 369-question Climate-Smart Agriculture Sourcebook dataset employed for baseline threshold determination (Section 3.4). This subset enabled intensive multi-agent testing while maintaining experimental tractability. The baseline configurations used similarity thresholds determined through the full 369-question evaluation: a threshold of 0.95 for Mistral 7B and Granite 3.2 8B, and a threshold of 0.90 for Llama 3.1 8B.

The experimental design encompassed 2,000 test instances across 20 configurations evaluated on 100 questions each. Configurations comprised 4 single-agent RAG baselines (Mistral 7B, Llama 3.1 8B, Granite 3.2 8B with independent retrieval, and Granite-SCR with shared-context retrieval), 12 standard multi-agent configurations (3 models × 4 coordination strategies with independent retrieval), and 4 Granite-SCR multi-agent configurations (4 coordination strategies with shared-context retrieval). The four coordination strategies—Collaborative, Sequential, Competitive, and Hierarchical—are detailed in Section 3.2. The Granite-SCR configurations provided all agents with identical retrieved context to isolate coordination overhead effects from retrieval fragmentation.

Performance was assessed using the CPS and T-CPS metrics (see Section 3.3). Statistical validation employed paired t-tests to compare each multi-agent configuration with its corresponding baseline ($\alpha = 0.05$), with effect sizes quantified using Cohen's d.

4.2. Overall Performance Comparison

Table 2 presents comprehensive performance results ranked by degradation magnitude across all model-strategy combinations. The unified presentation includes both CPS and T-CPS scores, percentage changes from baseline, statistical significance assessments, and effect size measurements. Configurations were grouped by degradation severity to facilitate interpretation of boundary conditions for multi-agent coordination effectiveness.

Table 2. Multi-Agent Coordination Performance Ranked by Degradation Magnitude.

Rank	Model-Strategy	CPS	T-CPS	Δ CPS (%)	Δ T-CPS (%)	t-stat	p-value	Cohen's d	|d|	Effect Size	Sig
BASELINES											
—	Mistral 7B	0.5796	0.6267	—	—	—	—	—	—	—	—
—	Granite 3.2 8B / SCR	0.5658	0.6125	—	—	—	—	—	—	—	—
—	Llama 3.1 8B	0.5397	0.5829	—	—	—	—	—	—	—	—
MINIMAL DEGRADATION											
1	Llama 3.1 8B Sequential	0.5139	0.5550	-4.78	-4.79	-1.915	0.058	-0.19	0.19	Negl.	ns
2	Llama 3.1 8B Hierarchical	0.5113	0.5499	-5.26	-5.66	-1.975	0.051	-0.20	0.20	Negl.	ns
MODERATE DEGRADATION											

 doi:10.20944/preprints202511.1168.v1

3	Llama 3.1 8B Competitive	0.5047	0.5450	-6.48	-6.50	-2.511	0.014	-0.25	0.25	Small	*
4	Granite-SCR Competitive	0.4854	0.5244	-14.21	-14.38	-6.632	<0.001	-0.66	0.66	Mdium	***
5	Granite-SCR Hierarchical	0.4746	0.5134	-16.13	-16.18	-7.966	<0.001	-0.80	0.80	Large	***
	SEVERE DEGRADATION										
6	Granite-SCR Sequential	0.4361	0.4685	-22.92	-23.51	-10.628	<0.001	-1.06	1.06	Large	***
7	Granite 3.2 8B Competitive	0.4224	0.4591	-25.35	-25.05	-14.393	<0.001	-1.44	1.44	Large	***
8	Mistral 7B Competitive	0.4344	0.4719	-25.05	-24.70	-12.724	<0.001	-1.27	1.27	Large	***
9	Mistral 7B Hierarchical	0.4211	0.4558	-27.34	-27.26	-13.624	<0.001	-1.36	1.36	Large	***
10	Granite 3.2 8B Hierarchical	0.4038	0.4389	-28.63	-28.34	-16.326	<0.001	-1.63	1.63	Large	***
11	Granite 3.2 8B Sequential	0.3999	0.4346	-29.32	-29.04	-17.033	<0.001	-1.70	1.70	Large	***
12	Mistral 7B Sequential	0.4121	0.4450	-28.90	-28.99	-13.632	<0.001	-1.36	1.36	Large	***
	UNIVERSAL COLLABORATIVE DEGRADATION										
13	Llama 3.1 8B Collaborative	0.3873	0.4215	-28.24	-27.69	-14.733	<0.001	-1.47	1.47	Large	***
14	Granite 3.2 8B Collaborative	0.3655	0.3984	-35.40	-34.96	-20.935	<0.001	-2.09	2.09	Very Large	***
15	Granite-SCR Collaborative	0.3871	0.4215	-31.58	-31.19	-19.485	<0.001	-1.95	1.95	Large	***
16	Mistral 7B Collaborative	0.3755	0.4084	-35.22	-34.83	-19.470	<0.001	-1.95	1.95	Large	***

Notes: Statistical significance: *** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$, ns = not significant ($p \geq 0.05$); Δ values: Percentage difference from respective baseline (negative indicates degradation); Cohen's d: Standardized effect size computed as $d = t/\sqrt{n}$ where $n = 100$; |d| (Absolute effect size): Magnitude interpretation using Cohen's

 doi:10.20944/preprints202511.1168.v1

conventions: Negligible: |d| < 0.2 (minimal practical significance); Small: 0.2 ≤ |d| < 0.5 (noticeable effect); Medium: 0.5 ≤ |d| < 0.8 (clear practical significance); Large: 0.8 ≤ |d| < 1.3 (substantial practical significance); Very Large: |d| ≥ 1.3 (profound practical significance); Grouping: Configurations grouped by degradation severity and statistical/practical significance; SCR: Shared Context Retrieval (eliminates retrieval fragmentation for Granite 3.2 8B).

The performance evaluation provides empirical evidence regarding multi-agent coordination effectiveness under the evaluated conditions. Only 2 of 16 multi-agent configurations (12.5%) demonstrated non-significant degradation relative to baseline RAG systems, with both configurations employing Llama 3.1 8B architecture (Sequential and Hierarchical strategies, ranks 1-2, |d| = 0.19-0.20). The remaining 87.5% showed statistically significant degradation ($p < 0.05$), with 81.25% reaching highly significant levels ($p < 0.001$). The complete performance hierarchy is visualized in Figure 4, which groups configurations by degradation severity from minimal (green) through moderate (orange) and severe (red) to universal collaborative failure (purple).

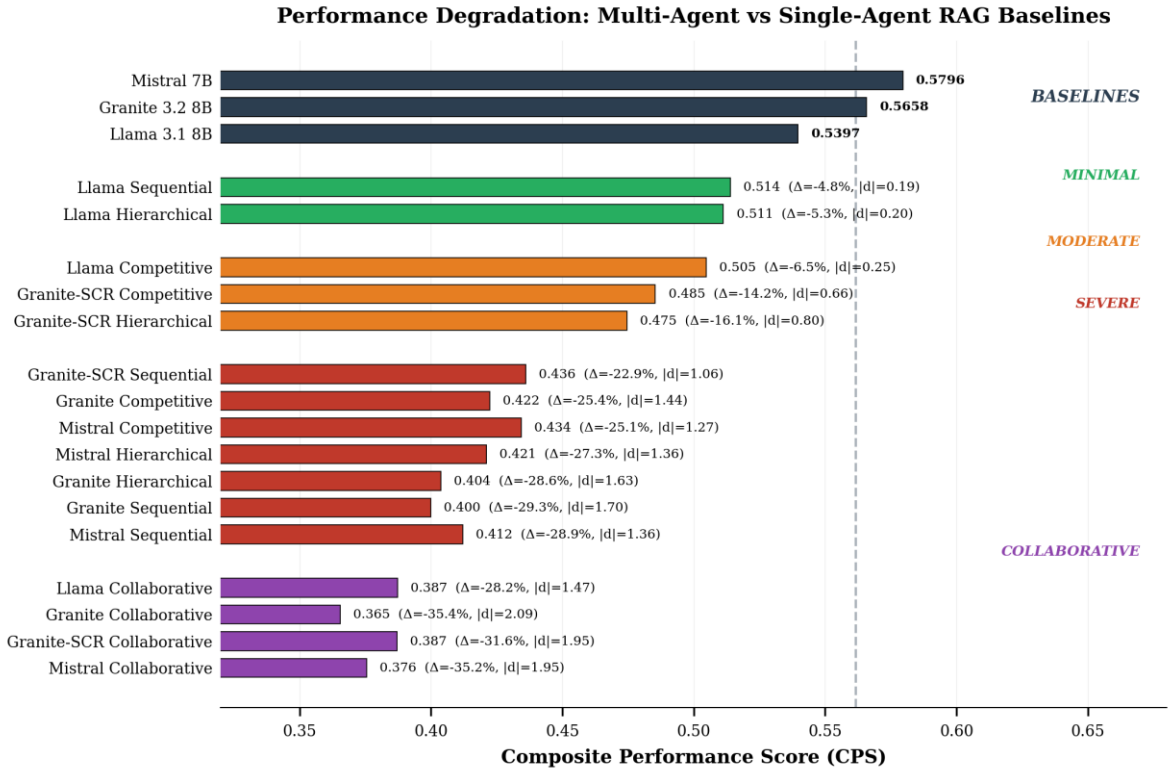

Figure 4. Multi-Agent Coordination Performance Degradation Relative to Single-Agent RAG Baselines. Horizontal bar chart displaying CPS scores organized by performance from highest (top) to lowest (bottom). Baselines (dark bars, top): Mistral 7B (0.5796), Granite 3.2 8B (0.5658), Llama 3.1 8B (0.5397). Multi-agent configurations grouped by degradation severity: Minimal (green, Δ = -4.78% to -5.26%, |d| < 0.20, ns), Moderate (orange, Δ = -6.48% to -16.13%, |d| = 0.25-0.80), Severe (red, Δ = -22.92% to -29.32%, |d| = 1.06-1.70, $p < 0.001$), Collaborative (purple, Δ = -28.24% to -35.40%, |d| = 1.47-2.09, $p < 0.001$). Bar annotations show CPS values, percentage degradation (Δ), and effect sizes (|d|). Dashed line indicates baseline average (0.5617). All 16 multi-agent configurations underperform baselines; 87.5% show statistically significant degradation ($\alpha = 0.05$).

Multi-agent coordination strategies consistently underperformed baselines, with CPS scores spanning 0.3655 to 0.5139. Performance degradation relative to baseline ranged from 4.78% (Llama 3.1 8B Sequential, rank 1) to 35.40% (Granite 3.2 8B Collaborative, rank 14). T-CPS degradation followed similar patterns, ranging from 4.79% to 34.96%, confirming that multi-agent approaches degraded both mean performance and consistency.

 doi:10.20944/preprints202511.1168.v1

Systematic patterns emerged across degradation severity groupings. The Minimal Degradation group (ranks 1-2) contained only Llama 3.1 8B configurations with negligible effect sizes ($|d| < 0.2$) and non-significant degradation. These represented the only configurations approaching baseline performance levels. The Moderate Degradation group (ranks 3-5) showed small to large effect sizes ($|d| = 0.25$-0.80) with mixed statistical significance, demonstrating measurable but varied coordination overhead. The The Severe Degradation group (ranks 6–12) exhibited large effect sizes ($|d| = 1.06$–1.70) with consistent statistical significance ($p < 0.001$), indicating substantial coordination costs for the Mistral 7B and Granite 3.2 8B models in the evaluated implementation. The Universal Collaborative Degradation group (ranks 13–16) demonstrated the most profound degradation, with large-to-very-large effect sizes ($d = 1.47$–2.09) across all models. This indicates that, in its implemented form, collaborative coordination produced consistently poor results. This failure pattern is clearly visible in Figure 4 as the purple bars at the bottom of the chart, representing the worst-performing configurations.

Effect size analysis revealed the magnitude of practical significance beyond statistical thresholds. Granite 3.2 8B demonstrated the strongest sensitivity to coordination overhead, with effect sizes ranging from $|d| = 1.44$ (Competitive, rank 7) to $|d| = 2.09$ (Collaborative, rank 14). These values substantially exceeded conventional thresholds for "large" effects ($|d| > 0.8$), indicating profound practical significance. Mistral 7B exhibited a similar pattern, with effect sizes from $|d| = 1.27$ (Competitive, rank 8) to $|d| = 1.95$ (Collaborative, rank 16).

Llama 3.1 8B presented a markedly different profile. While Collaborative strategy showed severe degradation ($|d| = 1.47$, rank 13, $p < 0.001$), the Sequential ($|d| = 0.19$, rank 1) and Hierarchical ($|d| = 0.20$, rank 2) strategies exhibited only negligible, non-significant effects. Competitive strategy showed small but statistically significant degradation ($|d| = 0.25$, rank 3, $p = 0.014$). These differential responses suggested that architectural characteristics influenced susceptibility to coordination overhead.

4.3. Statistical Significance and Effect Size Analysis

Statistical analysis confirmed that observed performance degradation in multi-agent strategies was not attributable to random variation. Of the sixteen multi-agent configurations evaluated, fourteen (87.5%) showed statistically significant degradation at $p < 0.05$, with thirteen (81.25%) reaching highly significant levels ($p < 0.001$). Only two configurations—Llama 3.1 8B Sequential (rank 1) and Hierarchical (rank 2)—failed to reach statistical significance, both with p-values marginally above the $\alpha = 0.05$ threshold ($p = 0.058$ and $p = 0.051$, respectively).

Effect sizes, quantified through Cohen's d and reported as $|d|$ in Table 2 and annotated in Figure 4, revealed the magnitude of practical significance beyond statistical thresholds. The effect size groupings demonstrated clear patterns. Negligible effects ($|d| < 0.2$) appeared only in ranks 1-2, representing the two non-significant Llama 3.1 8B configurations. Small effects ($0.2 \leq |d| < 0.5$) appeared only at rank 3 (Llama Competitive, $|d| = 0.25$). Medium effects ($0.5 \leq |d| < 0.8$) appeared only at rank 4 (Granite-SCR Competitive, $|d| = 0.66$). Large effects ($0.8 \leq |d| < 1.3$) spanned ranks 5-13, encompassing 9 of 16 configurations (56.25%). Very large effects ($|d| \geq 1.3$) appeared at ranks 7-8, 10-12, and 14, representing the most severe degradation cases.

Granite 3.2 8B demonstrated the strongest sensitivity to coordination overhead, with effect sizes ranging from $|d| = 1.44$ (Competitive, rank 7) to $|d| = 2.09$ (Collaborative, rank 14). These values substantially exceeded conventional thresholds for "large" effects ($|d| > 0.8$), indicating profound practical significance. Mistral 7B exhibited a similar pattern, with effect sizes from $|d| = 1.27$ (Competitive, rank 8) to $|d| = 1.95$ (Collaborative, rank 16).

Llama 3.1 8B presented a markedly different profile. While Collaborative strategy showed severe degradation ($|d| = 1.47$, rank 13, $p < 0.001$), the Sequential ($|d| = 0.19$, rank 1) and Hierarchical ($|d| = 0.20$, rank 2) strategies exhibited only small, non-significant effects. Competitive strategy showed small but statistically significant degradation ($|d| = 0.25$, rank 3, $p = 0.014$). These differential

 doi:10.20944/preprints202511.1168.v1

responses suggested that architectural characteristics influenced susceptibility to coordination overhead.

The Granite-SCR configuration demonstrated statistically significant improvement over standard Granite 3.2 8B across all strategies, with all comparisons reaching $p < 0.001$. Effect sizes for SCR improvement ranged from $|d| = 0.66$ (Competitive, rank 4) to $|d| = 1.95$ (Collaborative, rank 15), confirming that shared context delivery provided meaningful benefit. However, the effect sizes relative to baseline confirmed that even with shared context, coordination overhead remained prohibitive, with all SCR configurations ranking between 4th and 15th place among the 16 multi-agent approaches.

4.4. CPS and T-CPS Relationship Analysis

The relationship between CPS and T-CPS revealed critical insights about the interplay between mean performance and output consistency. For baseline configurations, T-CPS scores consistently exceeded CPS scores by 8.0% to 8.2% (Mistral: +8.1%, Llama: +8.0%, Granite: +8.2%), reflecting the reward term in the T-CPS formulation for stable outputs. This consistent elevation indicated that baseline RAG achieved both high performance and good consistency.

Multi-agent configurations exhibited more complex CPS-T-CPS relationships. Collaborative strategies (ranks 13-16) demonstrated a paradoxical pattern: despite having the lowest CPS scores across all configurations (0.3655-0.3873), they achieved relatively high T-CPS scores compared to their CPS values. Llama 3.1 8B Collaborative (rank 13) showed T-CPS elevation of 8.9% over CPS, while Mistral 7B Collaborative (rank 16) achieved 8.8% elevation, and Granite 3.2 8B Collaborative (rank 14) reached 9.0% elevation. This "stable mediocrity" pattern indicated that consensus mechanisms reduced variability by converging toward safe but suboptimal solutions rather than achieving consistently high quality.

In contrast, Competitive strategies (ranks 3, 4, 7-8) showed modest T-CPS elevation (6-9%), indicating higher variability in outputs. Hierarchical strategies (ranks 2, 5, 9-10) exhibited similar patterns with T-CPS elevation ranging from 7.5% to 8.2%. Sequential strategies (ranks 1, 6, 11-12) demonstrated intermediate behavior with T-CPS elevation around 8.0-8.2%. These patterns suggested that different coordination mechanisms produced distinct consistency profiles independent of mean performance levels.

The Granite-SCR configurations (ranks 4-6, 15) maintained T-CPS elevation patterns similar to standard Granite configurations (ranks 7, 10-11, 14), with both groups showing 7-9% elevation. This indicated that shared context delivery affected mean performance without substantially altering output consistency characteristics. This finding confirmed that retrieval fragmentation primarily impacted mean quality rather than variability.

Across all configurations, T-CPS remained substantially below baseline values, confirming that multi-agent coordination degraded both mean performance and consistency-aware metrics. The dual degradation pattern established that coordination overhead affected not only average quality but also output reliability—a critical consideration for production deployment where predictable behavior is essential.

4.5. Model-Specific Coordination Response Patterns

The three evaluated models demonstrate distinct response patterns to multi-agent coordination, revealing architectural dependencies in coordination effectiveness.

Llama 3.1 8B exhibits selective tolerance to coordination overhead. Sequential (rank 1) and Hierarchical (rank 2) strategies produced minimal, non-significant degradation (Sequential: -4.78%, $p = 0.058$, $|d| = 0.19$; Hierarchical: -5.26%, $p = 0.051$, $|d| = 0.20$), positioning these configurations as marginally viable alternatives when baseline RAG cannot be deployed. These configurations appear as green bars immediately below the baselines in Figure 4, visually distinct from all other multi-agent approaches. while Collaborative (rank 13) failed catastrophically (-28.24%, $p < 0.001$, $|d| = 1.47$), while Collaborative fails catastrophically (-28.2%, $p < 0.001$). This selective pattern suggests that Llama's

 doi:10.20944/preprints202511.1168.v1

architecture accommodates certain coordination patterns—specifically sequential processing and hierarchical management—while remaining vulnerable to consensus-based mechanisms.

Granite 3.2 8B demonstrated universal severe degradation across all coordination strategies (ranks 7, 10, 11, 14), with all configurations showing highly significant performance losses ($p < 0.001$) and large to very large effect sizes ($|d| = 1.44$ to 2.09). Figure 4 illustrates this model's high sensitivity to coordination overhead through the concentration of red and purple Granite bars in the middle and lower sections of the chart. The magnitude of degradation ranged from 25.35% (Competitive, rank 7) to 35.40% (Collaborative, rank 14), establishing Granite as highly sensitive to coordination overhead. The Granite-SCR analysis (Section 4.6) demonstrated that while shared context reduced retrieval fragmentation effects by 5.9-17.5%, coordination overhead remained dominant, accounting for 14.2-31.6% of total degradation.

Mistral 7B exhibited uniform severe degradation similar to Granite (ranks 8, 9, 12, 16), with all strategies producing highly significant losses ($p < 0.001$) ranging from 25.05% (Competitive, rank 8) to 35.22% (Collaborative, rank 16). Effect sizes ($|d| = 1.27$ to 1.95) indicated profound practical significance across all coordination approaches. Unlike Llama, Mistral shows no favorable coordination configurations, suggesting architectural characteristics that fundamentally conflict with distributed processing mechanisms.

4.6. Context-Sharing Impact on Granite 3.2 8B

The comparison between Granite 3.2 8B with independent retrieval and Granite-SCR with shared context delivery isolates the relative contributions of coordination overhead and retrieval fragmentation to multi-agent performance degradation. Shared context retrieval improved CPS by 5.9-17.5% and T-CPS by 5.6-17.0% relative to independent retrieval, with strategy-specific sensitivity: Hierarchical achieved the largest gains (rank 5 vs rank 10: CPS +17.5%, T-CPS +17.0%), followed by Competitive (rank 4 vs rank 7: CPS +14.9%, T-CPS +14.2%), Sequential (rank 6 vs rank 11: CPS +9.1%, T-CPS +7.8%), and Collaborative (rank 15 vs rank 14: CPS +5.9%, T-CPS +5.8%).

This differential response reveals architectural dependencies. Manager-worker (Hierarchical) and selection-based (Competitive) strategies rely more heavily on input consistency, as the coordinator or selection mechanism must process agent outputs that reference common context. Sequential and Collaborative mechanisms may introduce additional challenges—sequential error propagation and consensus dynamics—that shared context alone cannot address.

Despite these improvements, all Granite-SCR configurations (ranks 4-6, 15) performed below baseline RAG (CPS: 0.5658, T-CPS: 0.6125). Statistical analysis showed significant degradation for all SCR configurations (all $p < 0.001$), with CPS degradation ranging from 14.21% (Competitive, rank 4) to 31.58% (Collaborative, rank 15) and T-CPS degradation from 14.38% to 31.19%. Effect sizes ($d = -0.663$ to -1.948) indicated substantial practical significance.

This pattern suggests that while shared context reduces retrieval fragmentation effects, coordination overhead appeared to be the dominant factor affecting multi-agent performance in the evaluated implementation. The quantitative decomposition—with 5.9-17.5% performance loss attributable to retrieval fragmentation and 14.2-31.6% to coordination overhead—provides clear evidence that coordination mechanisms, not information inconsistency, primarily determine multi-agent viability.

4.7. Summary of Results

The comprehensive evaluation across 1,900 test instances establishes clear empirical boundaries for multi-agent coordination effectiveness:

1. Baseline RAG configurations consistently outperformed multi-agent strategies, with 87.5% showing statistically significant degradation ($p < 0.05$).
2. Performance degradation ranged from -4.78% (Llama Sequential) to -35.40% (Granite Collaborative), with effect sizes $|d| = 0.19$ to 2.09.

 doi:10.20944/preprints202511.1168.v1

3. Shared context retrieval (Granite-SCR) improved performance by 5.9-17.5% but remained 14.2-31.6% below baseline, confirming coordination overhead as the dominant limiting factor.
4. Llama 3.1 8B demonstrated selective tolerance (Sequential and Hierarchical strategies showed minimal degradation), while Granite 3.2 8B and Mistral 7B exhibited universal severe degradation.
5. Collaborative coordination failed universally (all $p < 0.001$, $|d| > 1.47$) despite highest output consistency.
6. T-CPS analysis confirmed that coordination degrades both mean performance and consistency simultaneously.

Figure 4 summarises the comprehensive evaluation results across all 1,900 test instances, providing a visual representation of the performance hierarchy established through systematic comparison. This visualisation highlights the key empirical findings presented in Section 4.

These findings suggest that single-agent RAG baselines demonstrated superior performance in most evaluated scenarios, with multi-agent coordination showing minimal degradation only in specific cases where model architecture (Llama 3.1 8B) and coordination strategy (Sequential or Hierarchical) aligned favorably.

5. Discussion

5.1. Multi-Agent Coordination Degrades

Nineteen experimental conditions were evaluated: three model baselines (Llama 3.1 8B, Granite 3.2 8B/Granite-SCR, Mistral 7B) and sixteen multi-agent configurations across four coordination strategies. As visualized in Figure 4, fourteen of sixteen multi-agent configurations (87.5%) showed statistically significant performance degradation relative to their respective single-agent RAG baselines ($p < 0.05$) under the evaluated implementation conditions. As detailed in Section 4.3, statistical analysis confirmed multi-agent degradation. The two non-significant configurations (Llama Sequential and Hierarchical, ranks 1-2 in Table 2) still showed negative performance trends. The total underperformance rate reaches 100% when directionality rather than statistical significance is considered.

These findings suggest caution regarding assumptions that multi-agent systems automatically improve LLM reasoning in all contexts. The results align with recent observations about multi-agent architecture limitations [44–47].

Two potential mechanisms may contribute to the observed degradation. First, coordination overhead—voting, synthesis, sequential processing—may consume resources without proportional quality improvements in certain implementations. Second, consensus mechanisms in their evaluated form appeared to converge toward compromise solutions rather than amplifying the strongest individual responses. Collaborative coordination showed the worst performance across all models (ranks 13-16), with degradation ranging from -28.24% to -35.40% and universally large effect sizes ($|d| = 1.47$-2.09, all $p < 0.001$). The purple bars at the bottom of Figure 4 illustrate this universal failure pattern, with all collaborative configurations occupying the lowest performance tier regardless of model architecture. This suggests that averaging homogeneous agent outputs reduces precision.

T-CPS analysis (Section 4.4) revealed a "stable average" pattern: several configurations achieved low output variability while producing consistently poor-quality results. This demonstrates that reliability and quality are distinct properties.

This demonstrates that specific model-strategy combinations can minimize both performance loss and variability, though such configurations remain exceptional rather than typical. This demonstrates that properly matched model-strategy combinations can improve both quality and consistency, though such configurations remain rare.

5.2. Performance Patterns Across Models and Strategies

 doi:10.20944/preprints202511.1168.v1

The three evaluated models demonstrated distinct response patterns to multi-agent coordination. Architectural dependencies in coordination effectiveness were revealed by these patterns.

Llama 3.1 8B exhibited greater tolerance to coordination overhead in the evaluated implementation. Sequential (rank 1) and Hierarchical (rank 2) strategies produced minimal, non-significant degradation (-4.78%, $p = 0.058$ and -5.26%, $p = 0.051$ respectively). These configurations may represent viable alternatives in scenarios where baseline RAG presents implementation challenges. Competitive strategy (rank 3) showed small but significant degradation (-6.48%, $p = 0.014$). Collaborative strategy exhibited substantial degradation (rank 13: -28.24%, $p < 0.001$). This selective pattern suggests that Llama's architecture accommodates sequential processing and hierarchical management. Vulnerability to consensus-based mechanisms persists.

Mistral 7B exhibited uniform, severe degradation across all strategies. Highly significant losses ($p < 0.001$) ranged from 25.05% (Competitive, rank 8) to 35.22% (Collaborative, rank 16). Effect sizes ($|d| = 1.27$ to 1.95) indicate profound practical significance across all coordination approaches. Unlike Llama, Mistral showed no favorable coordination configurations. Architectural characteristics that conflict fundamentally with distributed processing mechanisms are suggested.

Granite 3.2 8B demonstrated severe, universal degradation across all coordination strategies (ranks 7, 10, 11, 14), with all configurations showing significant performance losses ($p < 0.001$) and large to very large effect sizes ($|d| = 1.44$ to 2.09). Degradation magnitude ranged from 25.35% (Competitive, rank 7) to 35.40% (Collaborative, rank 14). Granite was established as highly sensitive to coordination overhead.

Strategy-specific patterns emerged across models. Sequential strategies showed the least degradation among multi-agent approaches for Mistral 7B (rank 12: -28.90%) and Granite 3.2 8B (rank 11: -29.32%), though both remained severely degraded. This relative performance advantage within the severe degradation category is apparent in Figure 4, where sequential configurations appear in the upper portion of the red (severe) section. Llama 3.1 8B Sequential (rank 1) demonstrated the minimal degradation pattern (-4.78%, ns). Competitive strategy succeeded only for Llama 3.1 8B (rank 3: -6.48%, small effect), while showing severe degradation for Mistral 7B (rank 8: -25.05%) and Granite 3.2 8B (rank 7: -25.35%). Hierarchical strategies uniformly underperformed baselines across all models (ranks 2, 5, 9, 10), though Llama 3.1 8B Hierarchical (rank 2: -5.26%, ns) showed minimal degradation. Mistral and Granite showed degradation with Competitive strategy. Hierarchical showed small improvements over other multi-agent strategies but uniformly underperformed baselines. Collaborative coordination failed universally (ranks 13-16: -28.24% to -35.40%, all $p < 0.001$, $|d| = 1.47$-2.09). High consensus scores (near 100%) indicate agents produced homogeneous outputs. Diversity benefits were eliminated while requiring multiple coordinated inference calls.

The Granite 3.2 8B versus Granite-SCR comparison isolated the relative contributions of coordination overhead and retrieval fragmentation. Shared context retrieval improved CPS by 5.9–17.5% and T-CPS by 5.6–17.0% relative to independent retrieval. Strategy-specific sensitivity was demonstrated. The largest gains were achieved by Hierarchical (CPS: +17.5%, T-CPS: +17.0%), followed by Competitive (CPS: +14.9%, T-CPS: +14.2%), Sequential (CPS: +9.1%, T-CPS: +7.8%), and Collaborative (CPS: +5.9%, T-CPS: +5.8%).

Architectural dependencies are revealed by this differential response. Hierarchical and Competitive strategies rely more heavily on input consistency. The coordinator or selection mechanism must process agent outputs referencing common context. Sequential and Collaborative mechanisms introduce additional failure modes. Sequential error propagation and consensus deadlock cannot be mitigated by shared context

Despite these improvements, all Granite-SCR configurations were significantly inferior to baseline RAG performance (CPS: 0.5658; T-CPS: 0.6125). Degradation ranged from 14.2% (Competitive) to 31.6% (Collaborative). All p values were less than 0.001. Consistency metrics remained nearly identical across retrieval configurations (Granite-SCR CV: 0.0380–0.0907 vs Granite 3.2 8B CV: 0.0327–0.0482). Coordination mechanisms, rather than retrieval fragmentation, primarily

 doi:10.20944/preprints202511.1168.v1

determine output stability. This confirms that coordination overhead remains the dominant factor limiting multi-agent effectiveness. Shared context reduces retrieval fragmentation effects but cannot overcome fundamental coordination costs.

5.3. Implementation Considerations and Generalizability

Several implementation factors affect generalizability.

Agent Homogeneity: All agents used identical base models with different system prompts. Heterogeneous agents (different architectures, specialized fine-tuning) might demonstrate improved coordination effectiveness not observed here.

Prompt Engineering: Role prompts were generated using AI assistance (GitHub Copilot with Claude Sonnet 4.5). Realistic development practices were reflected. Hand-crafted, domain-specific prompts with careful role differentiation might yield different outcomes at substantial development cost.

Consensus Mechanisms: Collaborative strategy implemented straightforward aggregation without sophisticated synthesis or conflict resolution. Advanced protocols (weighted voting, structured argumentation, learned aggregation) might preserve individual strengths while mitigating weaknesses. Complexity and overhead would be added.

Domain Specificity: Evaluation used single-domain question-answering (smart agriculture). Performance patterns may differ for complex reasoning tasks requiring multi-step decomposition, heterogeneous agent teams with complementary specializations, or multi-domain generation tasks.

This study does not include computational efficiency analysis. Multi-agent configurations inherently require multiple inference calls, but rigorous efficiency assessment requires hardware-consistent conditions not available in the present work.

5.4. Hardware and Deployment Limitations

A key limitation of this study is the use of CPU-only inference for Mistral 7B and Granite 3.2 8B, combined with heterogeneous hardware across models (Apple M1 unified memory architecture for Llama 3.1 8B versus Intel Xeon CPU-only execution for other models). These choices were dictated by available infrastructure rather than deployment best practices. Consequently, the present analysis focuses on quality and stability metrics (CPS and T-CPS) and on relative performance patterns within each model family, rather than on absolute timing or efficiency comparisons.

5.5. Comparison with Prior Literature

These findings extend prior work by comparing multi-agent strategies against RAG baselines rather than simple prompting baselines. This distinction matters. Most studies evaluate multi-agent systems against basic single-agent prompts. Coordination can appear more effective than it is through this comparison. RAG systems already incorporate external knowledge retrieval. A higher performance bar is set. Multi-agent coordination added on top of RAG is harder to justify.

The baseline RAG outperformed 87.5% of multi-agent configurations. This confirms the concerns raised in recent literature that the benefits do not outweigh the coordination overhead.

An unexpected pattern was also observed. Sequential coordination performed best among multi-agent approaches. Debate-based and hierarchical approaches are emphasized in most papers papers [7,8,11]. No universal “best” strategy exists. This is suggested by the pattern. Effectiveness depends on specific task characteristics and domain requirements.

6. Conclusions

This study evaluated four multi-agent coordination strategies across three open-source language models against single-agent RAG baselines. Of the sixteen multi-agent configurations evaluated under the specific implementation conditions of this work, fourteen (87.5%) showed statistically significant performance degradation compared to baseline systems ($p < 0.05$), with degradation

 doi:10.20944/preprints202511.1168.v1

ranging from 6.48% to 35.40%. This finding suggests that multi-agent coordination effectiveness may be more limited than commonly assumed in certain retrieval-augmented generation contexts.

Granite 3.2 8B with independent retrieval and Granite-SCR with shared context delivery were compared. Shared context delivery improved performance by 5.9-17.5% relative to independent retrieval, demonstrating that retrieval fragmentation contributed measurably to performance degradation. However, all configurations remained 14.2-31.6% below baseline. It is suggested that coordination overhead rather than retrieval inconsistency is the main limitation of multi-agent approaches.

Model-specific response patterns were also observed. Llama 3.1 8B demonstrated greater tolerance of coordination overhead, particularly with Sequential (4.78% degradation, ns) and Hierarchical (5.26% degradation, ns) strategies. Sequential coordination produced the least degradation for Llama 3.1 8B, though results were model-dependent, with severe degradation observed for Mistral 7B and Granite 3.2 8B. Collaborative coordination produced universal severe degradation across all models (28.24-35.40%, all $p < 0.001$). The effectiveness of coordination depends on model architecture rather than representing a universal property of coordination strategies. These findings establish this.

Several factors should be considered when interpreting these findings. The evaluation employed 7–8 billion parameter models with homogeneous agent architectures on single-domain question-answering tasks. Different patterns may emerge with larger models, heterogeneous agent teams, or tasks requiring complex, multi-step reasoning. Agent role prompts were generated using AI assistance (GitHub Copilot with Claude Sonnet 4.5), reflecting standard development practices. Alternative approaches employing hand-crafted, domain-specific prompts with careful role specialization might yield different coordination effectiveness patterns.

Future research should investigate role-specific prompting, advanced consensus mechanisms, adaptive strategy selection, and joint threshold-coordination tuning.

Author Contributions: Conceptualization, I.R. and I.P.; methodology, I.R.; software, I.R. and M.D.; validation, I.P., I.R. and L.D.; formal analysis, I.R. and I.P.; investigation, I.R. and L.D.; resources, M.D.; data curation, I.R; writing—original draft preparation, I.R.; writing—review and editing, I.P. and M.D.; visualization, I.R.; supervision, I.P.; project administration, L.D.; funding acquisition, Y.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Operational Programme "Innovation and Competitiveness" 2014-2020, co-financed by the European Union through the European Regional Development Fund, under grant number BG16RFPR002-1.014-0013-C01 "Digitization of the Economy in a Big Data Environment (DIGD)".

Data Availability Statement: The complete datasets generated and analyzed during this study are publicly available in the project's GitHub repository at https://github.com/scpdxtest/maPaSSER (accessed on 13 November 2025). The repository contains: (1) raw experimental results in CSV format (Mistral 7B, Llama 3.1 8B, Granite 3.2 8B) and five configurations (four multi-agent coordination strategies plus RAG baseline), including individual metric scores (METEOR, ROUGE-1, ROUGE-L, BLEU, Laplace perplexity, Lidstone perplexity, cosine similarity, Pearson correlation, F1 score) and composite performance measures (CPS, T-CPS); (2) the enhanced PaSSER framework implementation with Python Flask API server (multiagent_server_api_2.py) supporting multi-agent coordination protocols, React-based frontend for configuration and real-time monitoring, and MongoDB integration for result persistence; (3) documented implementations of four coordination strategies (collaborative, sequential, competitive, hierarchical) with explicit voting, aggregation, and selection logic; (4) the 100-item question-answer evaluation corpus derived from the Climate-smart Agriculture Sourcebook; (5) Python analysis scripts for statistical validation (tcps_script_2.py, tcps_analysis_2.txt) including CPS/T-CPS calculations and performance comparisons; (6) operational efficiency metrics including inference latency, token consumption, and coordination overhead automatically logged by the system; and (7) comprehensive installation instructions and reproducibility documentation in README.md. The original PaSSER framework for RAG evaluation is available at https://github.com/scpdxtest/PaSSER (accessed on 1 April 2024).

 doi:10.20944/preprints202511.1168.v1

Acknowledgments: In this section, you can acknowledge any support given which is not covered by the author contribution or funding sections. This may include administrative and technical support, or donations in kind (e.g., materials used for experiments). Where GenAI has been used for purposes such as generating text, data, or graphics, or for study design, data collection, analysis, or interpretation of data, please add "During the preparation of this manuscript/study, the author(s) used [tool name, version information] for the purposes of [description of use]. The authors have reviewed and edited the output and take full responsibility for the content of this publication."

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

LLM	Large Language Model
CPS	Composite Performance Score
T-CPS	Threshold-aware Composite Performance Score
API	Application Programming Interface
JSON	JavaScript Object Notation
CV	Coefficient of Variation

References

1. Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.; Rocktäschel, T.; et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In Proceedings of the Proceedings of the 34th International Conference on Neural Information Processing Systems; Curran Associates Inc.: Red Hook, NY, USA, 2020.
2. Barnett, S.; Kurniawan, S.; Thudumu, S.; Brannelly, Z.; Abdelrazek, M. Seven Failure Points When Engineering a Retrieval Augmented Generation System. In Proceedings of the Proceedings of the IEEE/ACM 3rd International Conference on AI Engineering - Software Engineering for AI; Association for Computing Machinery: New York, NY, USA, 2024; pp. 194–199.
3. Chen, J.; Lin, H.; Han, X.; Sun, L. Benchmarking Large Language Models in Retrieval-Augmented Generation. *Proceedings of the AAAI Conference on Artificial Intelligence* **2024**, *38*, 17754–17762, doi:10.1609/aaai.v38i16.29728.
4. Yu, W.; Zhang, H.; Pan, X.; Cao, P.; Ma, K.; Li, J.; Wang, H.; Yu, D. Chain-of-Note: Enhancing Robustness in Retrieval-Augmented Language Models. In Proceedings of the Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing; Al-Onaizan, Y., Bansal, M., Chen, Y.-N., Eds.; Association for Computational Linguistics: Miami, Florida, USA, November 2024; pp. 14672–14685.
5. Salemi, A.; Zamani, H. Evaluating Retrieval Quality in Retrieval-Augmented Generation. In Proceedings of the Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval; Association for Computing Machinery: New York, NY, USA, 2024; pp. 2395–2400.
6. Radeva, I.; Popchev, I.; Dimitrova, M. Similarity Thresholds in Retrieval-Augmented Generation. In Proceedings of the 2024 IEEE 12th International Conference on Intelligent Systems (IS); 2024; pp. 1–7.
7. Hong, S.; Zhuge, M.; Chen, J.; Zheng, X.; Cheng, Y.; Zhang, C.; Wang, J.; Wang, Z.; Yau, S.K.S.; Lin, Z.H.; et al. MetaGPT: Meta Programming for A Multi-Agent Collaborative Framework. In Proceedings of the International Conference on Learning Representations; 2023.
8. Liang, T.; He, Z.; Jiao, W.; Wang, X.; Wang, Y.; Wang, R.; Yang, Y.; Shi, S.; Tu, Z. Encouraging Divergent Thinking in Large Language Models through Multi-Agent Debate. In Proceedings of the Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing; Al-Onaizan, Y., Bansal, M., Chen, Y.-N., Eds.; Association for Computational Linguistics: Miami, Florida, USA, November 2024; pp. 17889–17904.

 doi:10.20944/preprints202511.1168.v1

9. Park, J.S.; O'Brien, J.; Cai, C.J.; Morris, M.R.; Liang, P.; Bernstein, M.S. Generative Agents: Interactive Simulacra of Human Behavior. In Proceedings of the Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology; Association for Computing Machinery: New York, NY, USA, 2023.
10. Bond, A.H.; Gasser, L. *Readings in Distributed Artificial Intelligence*; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1988; ISBN 978-0-934613-63-7.
11. Kim, Y.H.; Park, C.; Jeong, H.; Chan, Y.S.; Xu, X.; McDuff, D.; Breazeal, C.; Park, H.W. Adaptive Collaboration Strategy for LLMs in Medical Decision Making. *ArXiv* **2024**, *abs/2404.15155*.
12. Jiang, A.Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D.S.; Casas, D. de las; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L.; et al. Mistral 7B 2023.
13. Grattafiori, A.; Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Vaughan, A.; et al. The Llama 3 Herd of Models 2024.
14. Mishra, M.; Stallone, M.; Zhang, G.; Shen, Y.; Prasad, A.; Soria, A.M.; Merler, M.; Selvam, P.; Surendran, S.; Singh, S.; et al. Granite Code Models: A Family of Open Foundation Models for Code Intelligence 2024.
15. Ibm-Granite/Granite-3.2-8b-Instruct · Hugging Face Available online: https://huggingface.co/ibm-granite/granite-3.2-8b-instruct (accessed on 12 November 2025).
16. Climate Smart Agriculture Sourcebook | Food and Agriculture Organization of the United Nations Available online: https://www.fao.org/climate-smart-agriculture-sourcebook/en/ (accessed on 12 November 2025).
17. Radeva, I.; Popchev, I.; Doukovska, L.; Dimitrova, M. Web Application for Retrieval-Augmented Generation: Implementation and Testing. *Electronics* **2024**, *13*, doi:10.3390/electronics13071361.
18. Dimitrova, M. Retrieval-Augmented Generation (RAG): Advances and Challenges. *PROBLEMS OF ENGINEERING CYBERNETICS AND ROBOTICS* **2025**, *83*, 32–57, doi:https://doi.org/10.7546/PECR.83.25.03.
19. Xu, K.; Zhang, K.; Li, J.; Huang, W.; Wang, Y. CRP-RAG: A Retrieval-Augmented Generation Framework for Supporting Complex Logical Reasoning and Knowledge Planning. *Electronics* **2025**, *14*, doi:10.3390/electronics14010047.
20. Knollmeyer, S.; Caymazer, O.; Grossmann, D. Document GraphRAG: Knowledge Graph Enhanced Retrieval Augmented Generation for Document Question Answering Within the Manufacturing Domain. *Electronics* **2025**, *14*, doi:10.3390/electronics14112102.
21. Choi, Y.; Kim, S.; Bassole, Y.C.F.; Sung, Y. Enhanced Retrieval-Augmented Generation Using Low-Rank Adaptation. *Applied Sciences* **2025**, *15*, doi:10.3390/app15084425.
22. Ji, X.; Xu, L.; Gu, L.; Ma, J.; Zhang, Z.; Jiang, W. RAP-RAG: A Retrieval-Augmented Generation Framework with Adaptive Retrieval Task Planning. *Electronics* **2025**, *14*, doi:10.3390/electronics14214269.
23. Guo, T.; Chen, X.; Wang, Y.; Chang, R.; Pei, S.; Chawla, N.V.; Wiest, O.; Zhang, X. Large Language Model Based Multi-Agents: A Survey of Progress and Challenges. In Proceedings of the Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24; Larson, K., Ed.; International Joint Conferences on Artificial Intelligence Organization, August 2024; pp. 8048–8057.
24. Zhang, X.; Dong, X.; Wang, Y.; Zhang, D.; Cao, F. A Survey of Multi-AI Agent Collaboration: Theories, Technologies and Applications. In Proceedings of the Proceedings of the 2nd Guangdong-Hong Kong-Macao Greater Bay Area International Conference on Digital Economy and Artificial Intelligence; Association for Computing Machinery: New York, NY, USA, 2025; pp. 1875–1881.
25. Jimenez-Romero, C.; Yegenoglu, A.; Blum, C. Multi-Agent Systems Powered by Large Language Models: Applications in Swarm Intelligence. *Front. Artif. Intell.* **2025**, *8*, doi:10.3389/frai.2025.1593017.
26. Qian, C.; Liu, W.; Liu, H.; Chen, N.; Dang, Y.; Li, J.; Yang, C.; Chen, W.; Su, Y.; Cong, X.; et al. ChatDev: Communicative Agents for Software Development. In Proceedings of the Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers); Ku, L.-W., Martins, A., Srikumar, V., Eds.; Association for Computational Linguistics: Bangkok, Thailand, August 2024; pp. 15174–15186.
27. Wu, Q.; Bansal, G.; Zhang, J.; Wu, Y.; Li, B.; Zhu, E.; Jiang, L.; Zhang, X.; Zhang, S.; Liu, J.; et al. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation 2023.

 doi:10.20944/preprints202511.1168.v1

28. Bo, X.; Zhang, Z.; Dai, Q.; Feng, X.; Wang, L.; Li, R.; Chen, X.; Wen, J.-R. Reflective Multi-Agent Collaboration Based on Large Language Models. In Proceedings of the Advances in Neural Information Processing Systems; Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., Zhang, C., Eds.; Curran Associates, Inc., 2024; Vol. 37, pp. 138595–138631.
29. Cinkusz, K.; Chudziak, J.A.; Niewiadomska-Szynkiewicz, E. Cognitive Agents Powered by Large Language Models for Agile Software Project Management. *Electronics* **2025**, *14*, doi:10.3390/electronics14010087.
30. Ji, X.; Zhang, L.; Zhang, W.; Peng, F.; Mao, Y.; Liao, X.; Zhang, K. LEMAD: LLM-Empowered Multi-Agent System for Anomaly Detection in Power Grid Services. *Electronics* **2025**, *14*, doi:10.3390/electronics14153008.
31. Caspari, L.; Dastidar, K.G.; Zerhoudi, S.; Mitrovic, J.; Granitzer, M. Beyond Benchmarks: Evaluating Embedding Model Similarity for Retrieval Augmented Generation Systems 2024.
32. Muennighoff, N.; Tazi, N.; Magne, L.; Reimers, N. MTEB: Massive Text Embedding Benchmark. In Proceedings of the Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics; Vlachos, A., Augenstein, I., Eds.; Association for Computational Linguistics: Dubrovnik, Croatia, May 2023; pp. 2014–2037.
33. Topsakal, O.; Harper, J.B. Benchmarking Large Language Model (LLM) Performance for Game Playing via Tic-Tac-Toe. *Electronics* **2024**, *13*, doi:10.3390/electronics13081532.
34. Zografos, G.; Moussiades, L. Beyond the Benchmark: A Customizable Platform for Real-Time, Preference-Driven LLM Evaluation. *Electronics* **2025**, *14*, doi:10.3390/electronics14132577.
35. Li, B.; Han, L. Distance Weighted Cosine Similarity Measure for Text Classification. In Proceedings of the Intelligent Data Engineering and Automated Learning – IDEAL 2013; Yin, H., Tang, K., Gao, Y., Klawonn, F., Lee, M., Weise, T., Li, B., Yao, X., Eds.; Springer: Berlin, Heidelberg, 2013; pp. 611–618.
36. Wang, L.; Yang, N.; Huang, X.; Jiao, B.; Yang, L.; Jiang, D.; Majumder, R.; Wei, F. Text Embeddings by Weakly-Supervised Contrastive Pre-Training 2024.
37. Ni, J.; Qu, C.; Lu, J.; Dai, Z.; Hernandez Abrego, G.; Ma, J.; Zhao, V.; Luan, Y.; Hall, K.; Chang, M.-W.; et al. Large Dual Encoders Are Generalizable Retrievers. In Proceedings of the Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing; Goldberg, Y., Kozareva, Z., Zhang, Y., Eds.; Association for Computational Linguistics: Abu Dhabi, United Arab Emirates, December 2022; pp. 9844–9855.
38. PaSSER: Platform for Smart Testing and Evaluation of RAG Systems. *Journal Name* 2025, *X*, XX.
39. Batsakis, S.; Tachmazidis, I.; Mantle, M.; Papadakis, N.; Antoniou, G. Model Checking Using Large Language Models—Evaluation and Future Directions. *Electronics* **2025**, *14*, doi:10.3390/electronics14020401.
40. Kuncheva, L.I.; Whitaker, C.J. Measures of Diversity in Classifier Ensembles and Their Relationship with the Ensemble Accuracy. *Machine Learning* **2003**, *51*, 181–207, doi:10.1023/A:1022859003006.
41. Breiman, L. Random Forests. *Machine Learning* **2001**, *45*, 5–32, doi:10.1023/A:1010933404324.
42. Holtzman, Ari; Buys, Jan; Du, Li The Curious Case of Neural Text Degeneration. In Proceedings of the Proceedings of International Conference on Learning Representations; Online, May 6 2020.
43. Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.L.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. Training Language Models to Follow Instructions with Human Feedback. In Proceedings of the Proceedings of the 36th International Conference on Neural Information Processing Systems; Curran Associates Inc.: Red Hook, NY, USA, 2022.
44. Chen, W.; Su, Y.; Zuo, J.; Yang, C.; Yuan, C.; Chan, C.-M.; Yu, H.; Lu, Y.; Hung, Y.-H.; Qian, C.; et al. AgentVerse: Facilitating Multi-Agent Collaboration and Exploring Emergent Behaviors. In Proceedings of the International Conference on Representation Learning; Kim, B., Yue, Y., Chaudhuri, S., Fragkiadaki, K., Khan, M., Sun, Y., Eds.; 2024; Vol. 2024, pp. 20094–20136.
45. Chan, {Chi Min}; Chen, W.; Su, Y.; Yu, J.; Xue, W.; Zhang, S.; Fu, J.; Liu, Z. CHATEVAL: TOWARDS BETTER LLM-BASED EVALUATORS THROUGH MULTI-AGENT DEBATE.; Vienna, Austria, 2024.
46. Li, G.; Hammoud, H.A.A.K.; Itani, H.; Khizbullin, D.; Ghanem, B. CAMEL: Communicative Agents for "Mind" Exploration of Large Language Model Society 2023.
47. Wang, Z.; Mao, S.; Wu, W.; Ge, T.; Wei, F.; Ji, H. Unleashing the Emergent Cognitive Synergy in Large Language Models: A Task-Solving Agent through Multi-Persona Self-Collaboration. In Proceedings of the

 doi:10.20944/preprints202511.1168.v1

Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers); Duh, K., Gomez, H., Bethard, S., Eds.; Association for Computational Linguistics: Mexico City, Mexico, June 2024; pp. 257–279.