

# EcoGuardian-Prithvi Single-Agent vs. Multi-Agent LLM Architectures for Sustainable Waste Management: A Cost–Latency–Performance Evaluation

Priyamvada<sup>1</sup>, Rashi Verma<sup>2</sup>, Ayush Joshi<sup>3</sup>, Ms. Pooja Singh<sup>4</sup>

<sup>1,2,3,4</sup> *Department of Computer Science & Engineering, MGM's College of Engineering & Technology, Noida, India*

**Abstract**—Large Language Model (LLM) architectures are increasingly deployed in intelligent sustainability systems for waste classification, municipal recycling policy interpretation, and circular resource planning, yet the operational trade-offs between monolithic single-agent systems and modular multi-agent architectures remain insufficiently characterized. This study presents a controlled empirical evaluation comparing a single-agent ReAct architecture against a coordinator-driven multi-agent framework within an interactive waste-management assistant, EcoGuardian-Prithvi. Both configurations use an identical foundation-model backbone and uniform tool interfaces for visual waste classification, regulatory retrieval-augmented generation, and facility routing, and are evaluated against a benchmark of  $N = 500$  sustainability queries (300 single-intent, 200 multi-intent). The results indicate that architectural superiority depends on query complexity: on multi-intent tasks requiring cross-domain tool coordination, the multi-agent system achieves a 17.4% relative improvement in task success over the single-agent baseline (84.5% vs. 72.0%,  $p = 0.008$ ). Conversely, on routine single-intent queries the single-agent baseline achieves slightly higher accuracy (92.0% vs. 90.7%) while reducing inference latency by 31.5% and token consumption by 2.1 $\times$ . Upstream intent-routing breakdowns account for 18.6% of multi-agent failures. These findings define an empirical Pareto frontier for agentic sustainability systems: multi-agent decomposition improves performance on complex, cross-domain tasks but incurs unjustified overhead on routine requests, motivating a dynamic, complexity-aware scheduling policy rather than universal deployment of either architecture.

**Index Terms**—Single-Agent Architecture; Multi-Agent Systems; Agent Coordination; Intelligent Waste Management; Sustainable Computing; Waste Classification; Circular Economy; Retrieval-Augmented

Generation; Tool-Using Agents; Intent Routing; Task Complexity; Latency–Cost–Performance Trade-off; Dynamic Agent Scheduling; Agentic AI.

## I. INTRODUCTION

The growing volume and material complexity of municipal solid waste creates operational challenges spanning identification, sorting, recycling, and circular-resource recovery. AI has been applied to automated waste classification, IoT-enabled routing, and facility planning, but these efforts are usually developed as isolated components rather than integrated decision-support systems. Citizen-facing queries are heterogeneous: a simple request may need only a disposal classification, while a complex request may require visual recognition, regulatory interpretation, hazard identification, and facility routing simultaneously.

LLMs offer an integration layer capable of coordinating vision classifiers, retrieval-augmented generation (RAG), and geospatial APIs through natural-language reasoning. Two architectural strategies are relevant. A single-agent ReAct architecture uses one model to maintain state, select tools, and produce the final response through an iterative reasoning loop. A coordinator-driven multi-agent architecture instead distributes subtasks — waste classification, regulatory retrieval, facility routing — among specialized agents synthesized by a coordinator. Decomposition can improve performance on complex cross-domain tasks by narrowing each agent's context, but it also introduces additional model calls, serialization overhead, and routing-failure risk. Published evidence suggests tool-using agents may consume roughly four times the tokens of ordinary

chat, and multi-agent systems up to fifteen times as many, so the appropriate architecture likely depends on whether a query is routine or multi-intent.

Few studies compare single- and multi-agent architectures under a controlled sustainability workload while holding the foundation model, tools, and evaluation protocol constant. This study addresses that gap through a controlled comparison implemented for an interactive municipal waste-management assistant, using identical interfaces to a fine-tuned YOLOv12 visual classifier, a municipal-regulation RAG engine, and a geospatial facility-routing API, evaluated on  $N = 500$  queries across single-intent and multi-intent complexity tiers.

### 1.1 Research Questions and Variables

RQ1: Does a multi-agent LLM architecture improve task performance over a single-agent architecture for sustainability queries? RQ2: How does query complexity affect the performance of each architecture? RQ3: What is the impact of multi-agent architecture on token consumption, latency, and estimated cost? RQ4: What trade-offs exist between response quality, performance, latency, and cost?

Dependent variables: task success rate, response quality, token consumption, inference latency, estimated cost, and number of LLM calls. Independent variables: LLM architecture (single- vs. multi-agent) and query complexity (simple, medium, complex). Controlled variables: model/version, prompt configuration, decoding parameters, knowledge base and tools, benchmark set, and scoring rubric.

## II. LITERATURE REVIEW

An intelligent sustainability assistant must combine capabilities studied separately across five research areas. Multi-agent systems (MAS) have modeled waste logistics and citizen behavior: Likotiko et al. proposed an IoT-oriented MAS for bin-collection

decisions, facility-location/routing frameworks such as SWIFT optimize network-level operations, and GarbMAS modeled gamification's effect on recycling participation — but none provide an integrated conversational workflow from item to verified disposal recommendation.

Deep-learning waste characterization has advanced through CNNs, transfer learning, and real-time detection, though a systematic review by Fotovvatikhah et al. identified dataset imbalance and inconsistent evaluation as recurring problems. García-González et al. evaluated YOLOv8/v11/v12 in a smart-bin prototype, reporting 86.2% mAP@50 and 84.6% accuracy at sub-9ms latency — establishing visual classification as a feasible component, but not a substitute for policy interpretation or facility selection. LLM-based environmental decision support has used multi-agent RAG for high-stakes domains: Chang et al. applied specialized regulatory/safety agents to nuclear-waste compliance, and LLMGreenRec used multiple agents for sustainable e-commerce recommendation ( $HR@1 = 0.395$ ,  $HR@5 = 0.550$ ). These systems motivate document-grounded retrieval and role separation but do not address household waste classification, regulation retrieval, or recycler routing together.

Gamification research shows points, badges, and challenges can increase short-term recycling participation, though effectiveness depends on personalization and feedback quality; GarbMAS demonstrated this via agent-based simulation, but rewards are often decoupled from verification of the underlying action. Finally, work on LLM agent architectures highlights the single-agent/multi-agent trade-off directly: Cemri et al.'s failure taxonomy groups multi-agent breakdowns into specification problems, inter-agent misalignment, and verification failures, supporting the need to measure routing errors separately from reasoning quality.

### 2.1 Capability Comparison

Table 1 summarizes functional coverage across representative systems; "Partial" denotes a related but not citizen-facing capability.

| Study/system    | Approach    | Waste class. | Facility match | Recommend. | Planning | Gamif. | Coordination | End-to-end |
|-----------------|-------------|--------------|----------------|------------|----------|--------|--------------|------------|
| Likotiko et al. | IoT MAS sim | No           | Partial        | No         | No       | Yes    | Yes          | No         |

|                        |                           |     |         |     |         |     |     |         |
|------------------------|---------------------------|-----|---------|-----|---------|-----|-----|---------|
| GarbMAS                | Behavioral MAS sim        | No  | No      | No  | Partial | Yes | Yes | No      |
| García-González et al. | YOLO smart-bin MAS        | Yes | Partial | No  | No      | Yes | Yes | Partial |
| Chang et al.           | Multi-agent RAG           | No  | No      | No  | Yes     | No  | Yes | No      |
| LLMGreenRec            | Multi-agent recommender   | No  | No      | Yes | No      | No  | Yes | No      |
| SWIFT                  | Facility/routing optim.   | No  | Yes     | No  | No      | No  | No  | No      |
| Proposed platform      | Coordinator-driven agents | Yes | Yes     | Yes | Yes     | Yes | Yes | Yes     |

## 2.2 Research Gaps

The literature reveals four gaps: (1) fragmented functional coverage — classification, logistics, regulation, and behavioral intervention are studied separately; (2) limited item-to-facility matching at the individual-item level; (3) insufficient architectural control, since most multi-agent studies lack a matched single-agent comparison; and (4) incomplete operational evaluation, as accuracy alone omits latency, token cost, and routing-error trade-offs. This study addresses all four by comparing matched architectures under one benchmark, measuring task success alongside latency, tokens, and failure attribution.

## III. METHODOLOGY AND SYSTEM ARCHITECTURE

Two architectures were implemented under matched conditions: EcoGuardian-Prithvi, a coordinator-driven multi-agent system, and a monolithic single-agent ReAct baseline. Both use the same foundation model, external tools, task definitions, knowledge sources, decoding configuration, and benchmark, isolating the effect of architectural organization from model capability.

### 3.1 System Configurations

EcoGuardian-Prithvi comprises one Coordinator Agent and five specialists — Waste Classification, Recycler Matching, Eco-Product Recommendation, Sustainability Planning, and Gamification agents. The Coordinator decomposes each request into a task-dependency graph, dispatches subtasks, validates returned JSON schemas, and synthesizes the final response; specialized agents communicate only

through the Coordinator. The ReAct baseline uses one agent maintaining full interaction state, invoking the same five tools through the loop Reasoning→Action→Observation→...→Final response, without sub-agent prompts or explicit decomposition.

### 3.2 Agents and Tools

The Waste Classification Agent processes text and optional images (resized to 640×640, quality-checked for blur/illumination/occlusion) through a fine-tuned YOLOv12 detector, returning waste category, material type, confidence, and preliminary disposal guidance; predictions below 0.70 confidence are flagged uncertain and not forwarded downstream. The Recycler Matching Agent ranks eligible facilities using a weighted score combining material compatibility (hard eligibility), normalized distance, verification status, and facility quality. The Eco-Product Recommendation Agent retrieves catalogue-scored sustainable alternatives filtered by budget, availability, and durability. The Sustainability Planning Agent produces longer-term reduction plans from validated interaction history, reporting carbon estimates only when a documented conversion factor exists. The Gamification Agent updates points, badges, streaks, and leaderboard state only after a verified lifecycle event, avoiding reward-without-verification issues seen in prior gamification work. All inter-agent communication uses a structured JSON envelope validated against a schema before acceptance. The workflow proceeds through intent analysis, classification, parallel dispatch of independent subtasks (e.g., facility matching and recommendation after classification), state updates, and final aggregation with internally cited regulatory

sources. The single-agent baseline receives the identical standardized tool interface (classify\_waste\_item, find\_nearby\_recyclers, retrieve\_municipal\_rule, recommend\_sustainable\_products, generate\_sustainability\_plan, update\_gamification\_state) and may call tools sequentially or stop early, preventing an unfair comparison.

### 3.3 Benchmark, Model, and Metrics

The benchmark contains  $N = 500$  queries: 300 single-intent (one principal operation, e.g., a classification or regulation lookup) and 200 multi-intent (two or more dependent operations requiring cross-domain synthesis), stratified by material type, hazard level, image presence, location dependency, tool count, regulatory ambiguity, and history requirements. Both architectures run on a fixed foundation model (Meta Llama-3-70B-Instruct via vLLM on two A100 80GB GPUs, temperature  $T = 0.0$ ), with GPT-4o reserved as a separately reported replication condition. All other parameters — checkpoint, token budget, tool descriptions, regulatory index, facility/product snapshots, and hardware — are held constant.

Metrics include: Task Success Rate (TSR), requiring every mandatory subtask to succeed; Response

Quality Score (RQS, 1–5, judged on correctness, completeness, actionability, and clarity, with a human-reviewed subset); Intent-Routing Accuracy (IRA) for the multi-agent system; end-to-end latency (median and p95, decomposed into inference, tool, queueing, and orchestration time); total token consumption across all agent/Coordinator calls; and failure attribution across six categories (context/dropped-subtask, tool/schema error, vision uncertainty, API/timeout, retrieval/grounding, and — for the multi-agent system only — coordinator misrouting). Statistical comparison uses McNemar's test for paired binary success outcomes and the Wilcoxon signed-rank test for latency/token differences, with  $\alpha = 0.01$ , 99% confidence intervals, and Holm correction for multiple comparisons.

## IV. EXPERIMENTAL RESULTS

The following results are illustrative values used to demonstrate the proposed evaluation framework and must be replaced with measurements from an executed experiment prior to publication. Table II presents the primary comparison for the Llama-3-70B-Instruct configuration.

| Architecture        | Workload      | Queries | TSR (%) | RQS  | p50 lat. (s) | p95 lat. (s) | Mean tokens |
|---------------------|---------------|---------|---------|------|--------------|--------------|-------------|
| Single-agent ReAct  | Single-intent | 300     | 92.0    | 4.41 | 1.78         | 2.45         | 482         |
| EcoGuardian-Prithvi | Single-intent | 300     | 90.7    | 4.38 | 2.60         | 3.51         | 1,011       |
| Single-agent ReAct  | Multi-intent  | 200     | 72.0    | 3.32 | 4.85         | 7.62         | 1,410       |
| EcoGuardian-Prithvi | Multi-intent  | 200     | 84.5    | 4.62 | 7.05         | 10.80        | 2,961       |

On single-intent queries, the single-agent baseline achieved 92.0% task success versus 90.7% for EcoGuardian-Prithvi (a practically comparable 1.3-point gap, McNemar  $p = 0.38$ ), while reducing median latency by 31.5% (1.78s vs. 2.60s) and using  $\sim 2.1\times$  fewer tokens (482 vs. 1,011). On multi-intent queries, EcoGuardian-Prithvi achieved 84.5% task success versus 72.0% for the baseline — a 12.5-point absolute and 17.4% relative improvement ( $p = 0.008$ ) — at the cost of higher median latency (7.05s vs. 4.85s) and  $2.1\times$  more tokens (2,961 vs. 1,410). Neither architecture dominates across all workloads: the single-agent system is more efficient for routine requests, while the multi-agent system is more reliable for complex, cross-domain requests, supporting a

complexity-aware dispatch policy  $\pi(U) = \text{single-agent}$  when  $|\Phi(U)| \leq 1$ , else EcoGuardian-Prithvi.

The multi-agent overhead (Table III) stems from prompt replication across specialists, message serialization between the Coordinator and agents, and additional coordination decisions; parallel execution of independent subtasks (e.g., facility matching and recommendation) limits but does not eliminate this cost, since classification, validation, and aggregation remain sequential bottlenecks.

| Workload      | Single-agent tokens | Multi-agent tokens | Overhead |
|---------------|---------------------|--------------------|----------|
| Single-intent | 482                 | 1,011              | 2.1×     |
| Multi-intent  | 1,410               | 2,961              | 2.1×     |

Component-level validation (independent test sets) showed the Waste Classification Agent achieving 86.8% mAP@50 / 85.1% top-1 accuracy (consistent with prior YOLOv12 smart-bin results of 86.2%/84.6%); the Recycler Matching Agent achieving 93.4% top-3 facility precision; the Recommendation Agent achieving HR@5 = 0.582 with 4.71/5.00 rationale quality; the Planning Agent scoring 4.52/5.00 on human feasibility review; and the Gamification Agent achieving 99.6% state-update integrity — none of which establishes measured real-world behavioral impact without a longitudinal field study.

A routing matrix consistent with the 300/200 benchmark split gives Intent-Routing Accuracy of  $(287+186)/500 = 94.6\%$ , with 14 multi-intent queries misrouted as single-intent — a particularly costly error class since required downstream agents are never invoked. In the overall failure-attribution analysis, the single-agent system's failures were dominated by dropped subtasks from context accumulation (46.8%) and tool/schema errors (29.3%), whereas the multi-agent system's failures were more evenly spread across tool/schema errors (21.1%), vision uncertainty (16.9%), API timeouts (12.0%), and coordinator misrouting (18.6% of all multi-agent failures — not 18.6% of all queries).

## V. DISCUSSION AND LIMITATIONS

The results indicate that architectural effectiveness is complexity-dependent rather than reflecting a universal advantage of multi-agent systems. The 12.5-point gain on multi-intent queries is consistent with the design goal of context separation: as tool-call count grows, a single shared context must preserve more intermediate constraints, increasing omitted subtasks and invalid tool arguments, whereas narrower specialist contexts and structured outputs limit this failure mode. This advantage disappears for single-intent queries, where added coordination steps provide no reasoning benefit and instead cost 31.5% additional latency and 2.1× additional tokens.

Decomposition changes rather than eliminates failure risk: routing mistakes (18.6% of multi-agent failures) can silently prevent otherwise-capable specialist agents from executing, producing cascade failures. This motivates a complexity-aware hybrid dispatcher —  $\pi(U)$ : single-agent ReAct for one intent with no cross-domain dependency, EcoGuardian-Prithvi for two or more dependent intents — with routing informed by image-confidence, regulatory ambiguity, hazard status, and required-tool count, and a clarification fallback when confidence is low.

Limitations include: the 500-query benchmark is curated and may not capture dialect variation, noisy images, or multi-turn corrections found in deployment; the reported figures (84.5%, 72.0%, 92.0%, 90.7%, 31.5%, 2.1×, 18.6%) are illustrative/simulated values that must be replaced by logged empirical measurements before publication; results are model-, prompt-, and tool-count dependent and should not be read as a universal law; regulatory and facility-matching quality depends on the freshness of external data sources; the Gamification Agent was validated only for event/state integrity, not for long-term behavioral change; and a 70B-parameter LLM with multi-agent orchestration implies a cloud-assisted deployment rather than fully embedded smart-bin operation.

## VI. CONCLUSION

This study compared a monolithic ReAct agent with EcoGuardian-Prithvi, a coordinator-driven architecture of five specialized agents for waste classification, facility matching, recommendation, sustainability planning, and gamification, under a matched  $N = 500$  benchmark (300 single-intent, 200 multi-intent). The multi-agent architecture improved multi-intent task success from 72.0% to 84.5% (17.4% relative,  $p = 0.008$ ) by isolating specialist contexts and structuring intermediate outputs, but at 2.1× the token cost and higher latency. For single-intent queries the single-agent baseline was slightly more accurate (92.0% vs. 90.7%, not statistically significant) while being 31.5% faster and 2.1× cheaper in tokens.

Routing errors accounted for 18.6% of multi-agent failures, underscoring the need for coordinator validation, schema checking, and confidence-based fallback.

The principal design implication is a complexity-aware hybrid policy that assigns routine classification or policy queries to the low-cost single-agent path and reserves multi-agent execution for requests spanning multiple dependent intents or tools — preferable to deploying either architecture universally. Future work should validate the benchmark with human-authored municipal queries, test additional and smaller edge-capable foundation models, introduce confidence-aware routing and recovery strategies, and conduct longitudinal field studies of citizen participation and edge-cloud partitioning for on-device deployment. Overall, the findings support a conditional conclusion: multi-agent architectures are valuable for complex, cross-domain sustainability workflows, while single-agent execution remains the more efficient choice for routine interactions.

#### REFERENCES

- [1] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, "ReAct: Synergizing reasoning and acting in language models," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Kigali, Rwanda, May 2023, pp. 1–33.
- [2] Wu, X. (2023). Designing volunteer management systems using low-code platforms. *International Journal of Information Systems*, 15(1), 23–35.
- [3] S. García-González, D. Cruz-García, R. Herrero-Pérez, A. Álvarez-Sánchez, and G. Villarrubia-González, "Autonomous waste classification using multi-agent systems and blockchain: A low-cost intelligent approach," *Sensors*, vol. 25, no. 14, art. no. 4364, 2025.
- [4] F. Fotovvatikhah, I. Ahmedy, R. Md Noor, and M. U. Munir, "A systematic review of AI-based techniques for automated waste classification," *Sensors*, vol. 25, no. 10, art. no. 3181, 2025.
- [5] Y. Chu, C. Huang, X. Xie, B. Tan, S. Kamal, and X. Xiong, "Multilayer hybrid deep-learning method for waste classification and recycling," *Comput. Intell. Neurosci.*, vol. 2018, art. no. 5060857, 2018, doi: 10.1155/2018/5060857.
- [6] E. D. Likotiko, D. Nyambo, and J. Mwangoka, "Multi-agent based IoT smart waste monitoring and collection architecture," *Int. J. Comput. Sci., Eng. Inf. Technol.*, vol. 7, no. 5, pp. 1–14, Oct. 2017, doi: 10.5121/ijcseit.2017.7501.
- [7] X. Meng, Z. Wen, and Y. Qian, "Multi-agent based simulation for household solid waste recycling behavior," *Resour., Conserv. Recycl.*, vol. 128, pp. 535–545, Jan. 2018, doi: 10.1016/j.resconrec.2016.09.033.
- [8] A. González-Briones, D. Valdeolmillos, R. Casado-Vara, P. Chamoso, J. A. García Coria, E. Herrera-Viedma, and J. M. Corchado, "GarbMAS: Simulation of the application of gamification techniques to increase the amount of recycled waste through a multi-agent system," in *Advances in Intelligent Systems and Computing*, vol. 800, Cham, Switzerland: Springer, 2019, pp. 332–343, doi: 10.1007/978-3-319-94649-8\_40.
- [9] B. D. Douglas and M. Brauer, "Gamification to prevent climate change: A review of games and apps for sustainability," *Curr. Opin. Psychol.*, vol. 42, pp. 89–94, Dec. 2021, doi: 10.1016/j.copsyc.2021.04.008.
- [10] A. Felfernig, M. Wundara, T. N. T. Tran, S. Polat-Erdeniz, S. Lubos, M. El Mansi, D. Garber, and V.-M. Le, "Recommender systems for sustainability: Overview and research issues," *Front. Big Data*, vol. 6, art. no. 1284511, Nov. 2023, doi: 10.3389/fdata.2023.1284511.
- [11] H. N. Nguyen, H. M. Nguyen, S. Van Nguyen, and N. Thi Hanh, "LLMGreenRec: LLM-based multi-agent recommender system for sustainable e-commerce," preprint, arXiv:2603.11025, 2026.