A Multi-AI Agent System for Autonomous Optimization of Agentic AI Solutions via Iterative Refinement and LLM-Driven Feedback Loops

Kamer Ali Yuksel **Thiago Castro Ferreira** **Mohamed Al-Badrashiny** **Hassan Sawaf**
aiXplain Inc., San Jose, CA, USA
{kamer, thiago, mohamed, hassan}@aixplain.com

Abstract

Agentic AI systems use specialized agents to handle tasks within complex workflows, enabling automation and efficiency. However, optimizing these systems often requires labor-intensive, manual adjustments to refine roles, tasks, and interactions. This paper introduces a framework for autonomously optimizing Agentic AI solutions across industries, such as NLG-driven enterprise applications. The system employs agents for Refinement, Execution, Evaluation, Modification, and Documentation, leveraging iterative feedback loops powered by an LLM (*Llama 3.2-3B*). The framework achieves optimal performance without human input by autonomously generating and testing hypotheses to improve system configurations. This approach enhances scalability and adaptability, offering a robust solution for real-world applications in dynamic environments. Case studies across diverse domains illustrate the transformative impact of this framework, showcasing significant improvements in output quality, relevance, and actionability. All data for these case studies, including original and evolved agent codes, along with their outputs, are here: anonymous.4open.science/r/evolver-1D11/

1 Introduction

Agentic AI systems, composed of specialized agents working collaboratively to achieve complex objectives, have transformed industries such as market research, business process optimization, and product recommendation. These systems excel in automating decision-making and streamlining workflows. However, their optimization remains challenging due to the complexity of agent interactions and reliance on manual configurations.

Recent advancements in large language models (LLMs) provide a solution by enabling automated refinement of Agentic AI systems. LLMs can autonomously generate and evaluate complex hypotheses, facilitating iterative improvements in agent roles and workflows with minimal human oversight. Case studies conducted demonstrate how these advancements address domain-specific challenges. These examples highlight the framework's scalability and adaptability, making it particularly effective for dynamic, evolving environments.

This paper presents a framework for autonomous optimization of Agentic AI systems using LLM-powered feedback loops. The framework improves efficiency and scalability by refining agent configurations based on qualitative and quantitative metrics. Several case studies across various domains provide evidence of the framework's ability to overcome domain-specific challenges. Designed for deployment in enterprise systems, this framework addresses persistent challenges in optimizing complex workflows in real-world settings.

This work establishes a scalable, autonomous system for optimizing Agentic AI with broad applicability across industries. **Key contributions:**

- **Evolutionary Optimization:** Evolving agent configurations without manual intervention.
- **Autonomous Refinement:** Fully automated optimization through iterative feedback loops.
- **Validation Case Studies:** Empirical results from case studies in diverse domains, demonstrating significant performance gains.

2 Background

Agentic AI systems automate complex processes across industries, providing significant efficiency gains. However, their optimization requires addressing the intricacies of agent interactions, particularly in dynamic environments with evolving objectives. Recent advancements in LLMs offer transformative capabilities by enabling autonomous generation and evaluation of hypotheses to improve workflows. This framework distinguishes itself by enabling fully autonomous optimization of Agentic

Proceedings of the 1st Workshop for Research on Agent Language Models (REALM 2025), pages 52–62

AI systems. The system enhances scalability, adaptability, and domain independence through LLM-driven feedback loops, hypothesis generation, and iterative modifications, setting a new standard for optimizing complex AI workflows. Unlike previous approaches reliant on predefined tasks or manual intervention, this method offers a superior solution for real-world, dynamic environments.

Prior research has explored various aspects of agentic systems' optimization and LLM integration. For instance, Huang et al. (2024) introduced *MLAgentBench*, a benchmark for evaluating language agents across diverse tasks. While this framework provides valuable insights, it focuses primarily on performance evaluation rather than iterative workflow refinement, which is our study's focus. Similarly, Smith et al. (2023) explored the use of *Large Model Agents (LMAs)* to enhance cooperation between agents using LLMs, highlighting the potential of LLMs for iterative feedback loops among agents. This aligns closely with the current study's emphasis on autonomous refinement processes.

Johnson and Liu (2023) demonstrated how LLMs enable agents to autonomously refine their roles and workflows, underscoring the significance of optimizing agentic AI systems. Meanwhile, Pan and Zhang (2024) proposed using automated evaluators to refine agent performance in tasks like web navigation. While effective in that domain, it lacks the scalability and domain independence targeted by the current framework. Similarly, Wang et al. (2024) introduced an LLM-based skill discovery framework, resonating with the iterative task proposals discussed in this study. Furthermore, Hu et al. (2024) emphasized the importance of modular components and foundational models for planning and executing agentic systems, focusing on the design phase, whereas the current study emphasizes continuous refinement and workflow optimization.

Masterman et al. (2024) reviewed emerging AI agent architectures, focusing on modularity and scalability, which are central to this framework's refinement approach. Mitra et al. (2024) proposed a framework for generating synthetic data with agents, showcasing their potential to refine outputs via feedback loops. Yu et al. (2024) introduced the *Reflective Tree* for multistep decision-making, aligning with this study's iterative design. Pan et al. (2024) addressed feedback loop risks like reward hacking, which this framework mitigates through robust evaluation. Finally, Miller et al. (2024) emphasized agent benchmarks, reinforcing this framework's focus on qualitative metrics.

Recently, Automated Design of Agentic Systems (ADAS) was introduced by Hu et al. (2024), focusing on creating new agentic designs through a meta-agent that programs novel agents by combining and refining building blocks. While ADAS is designed to invent new agents, the framework presented in this study is focused on optimizing existing agent systems. Through iterative LLM-driven feedback loops, agent roles, tasks, and workflows are refined to enhance adaptability and scalability in dynamic environments. Unlike ADAS, which prioritizes agent creation, this work focuses on continuously improving and optimizing established systems.

Algorithm 1 Agentic AI Refinement Process

Input:
    C_0: Initial code
    criteria: Qualitative evaluation criteria
    ϵ: Improvement threshold
    max_iterations: Maximum number of iterations
Output:
    C_{best}: Best-known code variant
    O_{best}: Best-known code variant output
    Mem: Stores the best performing variant
Initialization:
$C_{\text{best}} \leftarrow C_0$
$O_{\text{best}} \leftarrow \text{execute}(C_0)$
$S_{\text{best}} \leftarrow f(O_{\text{best}}, \text{criteria})$
iteration $\leftarrow 0$
while iteration $<$ max_iterations **do**
    iteration $\leftarrow$ iteration $+ 1$
    $E_{\text{best}} \leftarrow \text{evaluate}(O_{\text{best}}, \text{criteria})$
    $\mathcal{H}_i \leftarrow \text{generate\_hypotheses}(E_{\text{best}})$
    $M \leftarrow$ Modification agent
    $C_{i+1} \leftarrow M(\mathcal{H}_i, C_{\text{best}})$
    $O_{i+1} \leftarrow \text{execute}(C_{i+1})$
    $S_{i+1} \leftarrow f(O_{i+1}, \text{criteria})$
    if $S_{i+1} > S_{\text{best}}$ **then**
        $C_{\text{best}} \leftarrow C_{i+1}$
        $O_{\text{best}} \leftarrow O_{i+1}$
        $S_{\text{best}} \leftarrow S_{i+1}$
        $Mem \leftarrow (C_{\text{best}}, O_{\text{best}})$
    end if
    if $|S_{i+1} - S_{\text{best}}| < \epsilon$ **then**
        break ▷ Stop if improvement is below threshold
    end if
end while
Return $Mem.C_{\text{best}}, Mem.O_{\text{best}}$

3 Architecture

The proposed method for autonomous refinement and optimization of Agentic AI systems leverages several specialized agents, each responsible for a specific phase in the refinement process. This method operates in iterative cycles, continuously refining agent roles, goals, tasks, workflows, and dependencies based on qualitative and quantitative output evaluation. Moreover, the system is designed with scalability, ensuring its deployment across industries. The LLM-driven feedback loops offer a foundational infrastructure for adapting the system to various NLP applications, ensuring broad applicability across domains. The proposed method's optimization process is guided by two core frameworks: the Synthesis Framework and the Evaluation Framework. Synthesis Framework framework generates hypotheses based on the system's output. The Hypothesis Agent and Modification Agent collaborate to synthesize new configurations for the Agentic AI system, proposing modifications to agent roles, goals, and tasks; to be tested by the Evaluation Framework.

The refinement and optimization process is structured into these frameworks, contributing to the continuous improvement of the Agentic AI solution. The proposed method operates autonomously, iterating through cycles of hypothesis generation, execution, evaluation, and modification until optimal performance is achieved. A detailed report of a refinement iteration is provided in Appendix A. This method begins by deploying a baseline version of the Agentic AI system. Agents are assigned predefined roles, tasks, and workflows, and the system generates initial qualitative and quantitative criteria based on the system's objectives. An LLM is used to analyze the system's code and extract evaluation metrics, which serve as benchmarks for assessing future outputs. Human input can be introduced to revise or fine-tune the evaluation criteria to better align with project goals; which is optional, as the method is designed to operate autonomously.

The proposed method begins with a baseline version of the Agentic AI system, assigning initial agent roles, goals, and workflows. The first execution is run to generate the initial output and establish the baseline for comparison. After evaluating the initial output, the Hypothesis Agent generates hypotheses for modifying agent roles, tasks, or workflows based on the evaluation feedback. These hypotheses are then passed to the Modification Agent, which synthesizes changes to agent logic, interactions, or dependencies, producing new system variants. The Execution Agent executes the newly modified versions of the system, and performance metrics are gathered. The outputs generated are evaluated using qualitative and quantitative criteria (e.g., clarity, relevance, execution time). The Selection Agent compares the newly generated outputs against the best-known variant, ranks the variants, and determines whether the new output is superior. Memory Module stores the best-performing variants for future iterations. The cycle repeats as the proposed method continues refining the agentic workflows, improving overall performance until predefined or generated (and optionally revised) criteria are satisfied.

3.1 Synthesis Framework

The Refinement Agent manages the iterative optimization process by delegating tasks to other agents and synthesizing hypotheses for improving the system. It evaluates agent outputs against qualitative and quantitative criteria, identifying areas where agent roles, tasks, or workflows can be improved. The Refinement Agent leverages evaluation metrics such as clarity, relevance, depth of analysis, and actionability to propose modifications that enhance system output. The Hypothesis Generation Agent proposes specific changes to the agent system based on the output analysis. This module generates hypotheses for improving agent roles, tasks, and interactions based on evaluation feedback. For example, if agents are underperforming due to inefficiencies in their task delegation, the hypothesis module might suggest altering task hierarchies or reassigning specific roles.

The Modification Agent implements changes based on the hypotheses generated by the Refinement Agent. These changes may involve adjusting agent logic, modifying workflows, or altering agent dependencies. By synthesizing these changes, our method creates multiple variants of the Agentic AI solution. Each variant is stored and documented, with details regarding the expected improvements. The Execution Agent runs modified versions of the system, executing the newly generated variants and collecting performance data for subsequent evaluation. It ensures that agents perform their tasks as specified in the new configuration and debug issues as they arise. The Execution Agent tracks qualitative and quantitative outputs, feeding this information into the evaluation process.

3.2 Evaluation Framework

The Evaluation Framework is responsible for assessing the outputs of each system variant. The Evaluation Agent employs Llama 3.2-3B to evaluate both qualitative and quantitative aspects of the system's performance. The Evaluation Framework ensures that each iteration aligns with the system's overarching objectives, focusing on continuous improvement. The Evaluation Agent assesses the outputs of each system variant using a LLM. The LLM evaluates outputs based on predefined or generated qualitative criteria, including clarity, relevance to the task, depth of analysis, actionability, and quantitative metrics such as execution time and success rate. The Evaluation Agent provides a comprehensive system performance analysis, identifying areas for further improvement. After each iteration, the Selection Agent compares the outputs of the modified system against the best-known configuration. It ranks the new variants based on the evaluation scores provided by the Evaluation Agent, determining which configuration yields the highest performance. The top-ranked variant is stored for future iterations, ensuring continuous improvement.

3.3 Refinement Process

Agentic AI refinement process begins with the initialization of the best-known code variant, denoted as C_0, and the generation of its corresponding output, O_{C_0}. The performance of the output is evaluated using a set of qualitative criteria (e.g., clarity, relevance, depth of analysis), where the evaluation function $f(O_C, \text{criteria})$ produces a score $S(C_0) = f(O_{C_0}, \text{criteria})$ based on these criteria. This initial score, $S(C_0)$, is the baseline for comparison in subsequent iterations. At each iteration i, the current best-known output, O_{C_i}, is evaluated, and a set of hypotheses, $\mathcal{H}_i = \text{generate\_hypotheses}(E_{C_i})$, is generated from the qualitative evaluation E_{C_i} to suggest improvements. The hypotheses $\mathcal{H}_i$ are then applied to the code C_i, resulting in a new variant $C_{i+1} = M(\mathcal{H}_i, C_i)$. The new code variant C_{i+1} is executed, producing a new output $O_{C_{i+1}}$. The new output is evaluated using the same evaluation function $f(O_C, \text{criteria})$, yielding a new score $S_{i+1} = f(O_{C_{i+1}}, \text{criteria})$. If the new score S_{i+1} is greater than the best-known score $S_{\text{best}} = \max(S_{i+1}, S_{\text{best}})$, the new variant is considered superior, and the best-known variant is updated as follows. The process continues iteratively until a stopping condition is met, either when the improvement between iterations becomes smaller than a predefined threshold $|S_{i+1} - S_{\text{best}}| < \epsilon$, or when a maximum number of iterations is reached. Upon termination, the proposed method returns the best-known variant C_{best} and its output O_{best}.

Once initialized, the proposed method enters the execution phase, where agents perform their assigned tasks according to the baseline configuration. The Execution Agent runs the system, producing initial outputs that serve as a baseline for comparison in subsequent iterations. The results of this execution phase are stored for future analysis and comparison. The Evaluation Agent evaluates the outputs produced in the execution phase. The proposed method employs qualitative and quantitative criteria to assess the quality of the outputs. Qualitative metrics include relevance, clarity, depth of analysis, and actionability, while quantitative metrics include execution time, task completion rate, and overall system efficiency. The Evaluation Agent uses an LLM to generate detailed feedback, identifying areas where the system can be improved. The Hypothesis Generation Agent analyzes the evaluation data, generating hypotheses for improving agent roles, tasks, and workflows. These hypotheses may include changes such as altering task delegation, modifying agent goals, or restructuring the interdependencies between agents. Once the hypotheses are generated, the Modification Agent implements the proposed changes, creating new system variants based on these modifications. The modified versions of the system are re-executed by the Execution Agent, and the Evaluation Agent again evaluates their outputs. This iterative process continues, with each new variant compared against the previous best-known configuration. The Selection Agent ranks the system variants based on performance, ensuring that the Memory Module only stores the top-performing versions.

4 Case Studies

The evolution of agent systems in various domains highlights the need for continuous refinement to meet the dynamic demands of industry standards and user expectations. This section presents an overview of several case studies that illustrate the transformative process of refining agent systems across diverse applications, including market research, AI architecting, career transitions, outreach strategies, LinkedIn posts, meeting facilitation, lead generation, content creation, and presentation

development. Each case study showcases the challenges faced by the original systems, the strategic modifications implemented, and the resultant improvements in output quality. The findings underscore the significance of specialization and data-driven decision-making in enhancing agent system performance. All data for these case studies, including original and evolved agent codes, along with their outputs and evaluation reports, are here: anonymous.4open.science/r/evolver-1D11/

4.1 Market Research Agent

The original market research agent system was developed to provide strategic insights. However, it encountered several challenges, including inadequate market research depth, subpar strategy development, and limited output quality. These deficiencies hindered the system's ability to effectively align with user needs, resulting in low scores across evaluation criteria. The evolved agent system introduced specialized roles such as Market Research Analyst, Data Analyst, and User Experience Specialist to address these issues. These changes aimed to enhance the depth of market analysis, create a data-driven decision-making framework and prioritize user-centered design principles. The system was better equipped to understand emerging trends and deliver actionable insights by incorporating specialized agents. The refined agent system achieved remarkable improvements in output quality, scoring 0.9 in alignment and relevance, accuracy and completeness, and clarity and actionability. The evolved outputs provided a coherent strategy framework, significantly enhancing the overall effectiveness of the market research agent.

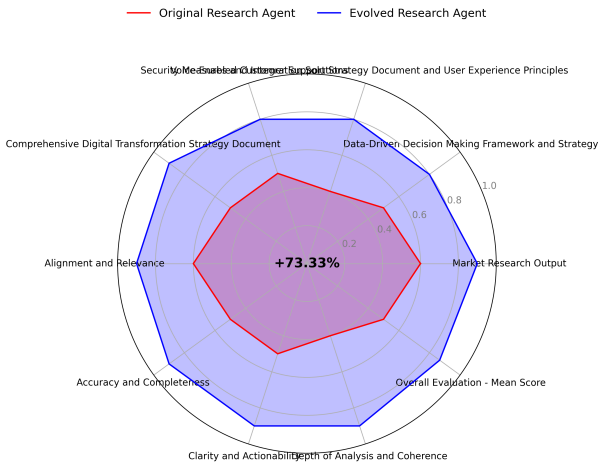

Figure 1: Market Research Agent Refinement

4.2 Medical AI Architect Agent

The architect agent system for medical imaging faced challenges related to regulatory compliance, patient engagement, and explainability of AI-driven decision-making processes. These limitations resulted in moderate evaluation scores, undermining the system's effectiveness in addressing critical healthcare needs. In response, the evolved system incorporated specialized agents, including a Regulatory Compliance Specialist and a Patient Advocate, to ensure adherence to standards and prioritize patient needs. Developing transparency frameworks strengthened the focus on explainability, while continuous monitoring mechanisms were established for ongoing performance assessment. The evolved system's outputs demonstrated significant improvements across multiple evaluation criteria, including regulatory compliance (0.9), patient-centered design (0.8), and explainability (0.8). These enhancements underscore the importance of specialization in developing systems that meet complex healthcare demands, ultimately leading to improved patient care and outcomes.

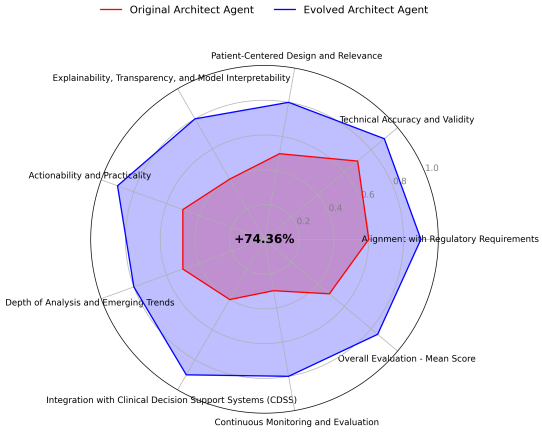

Figure 2: AI Architect Agent Refinement

4.3 Career Transition Agent

The original AI transition agent system was intended to assist software engineers in transitioning to AI specialist roles. However, it struggled with alignment to industry expertise and clarity in career growth goals. This disconnect resulted in ineffective action plans and poor communication clarity. The evolved system adopted a multifaceted approach, introducing new agents such as Domain Specialist and Skill Developer. The tasks were refined to ensure specificity and clarity, enhancing communication through detailed timelines and

structured outputs. The modifications led to substantial improvements in evaluation scores, with notable advancements in alignment with AI domain expertise (91%) and communication clarity (90%). The enhanced system provides clear, actionable goals, facilitating a more effective transition for software engineers into AI roles and highlighting the importance of agent system refinement.

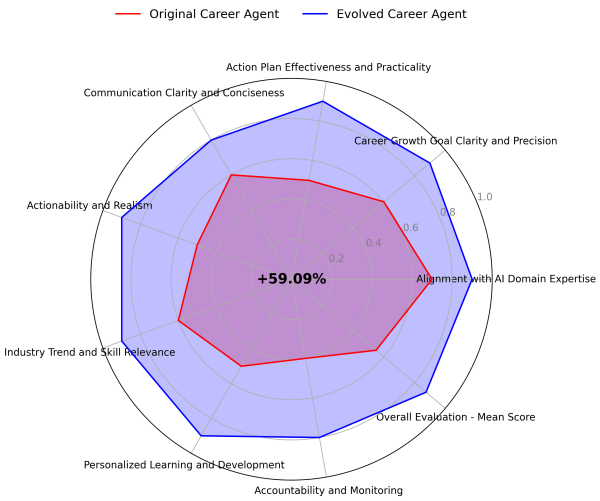

Figure 3: Career Transition Agent Refinement

4.4 Outreach Agent

Initially, the outreach agent system designed for the supply chain faced limitations due to its narrow focus and poor output quality. The original system was characterized by basic roles, such as Email Drafter, which failed to address the complexities of supply chain management. Five specialized roles were introduced to enhance the system, focusing on supply chain analysis, optimization, and sustainability. This comprehensive approach allowed for a deeper analysis of supply chain challenges and operational inefficiencies. The evolved agent system demonstrated significant improvements, with enhanced clarity, accuracy, and actionability in outputs. The modifications led to outputs that exceeded the refined evaluation criteria, establishing the system as a valuable tool for e-commerce companies seeking effective supply chain solutions.

4.5 LinkedIn Agent

The original generative AI agent system struggled with limitations in-depth, audience engagement, and source credibility when creating LinkedIn posts on generative AI trends. These challenges affected the system's ability to generate insightful

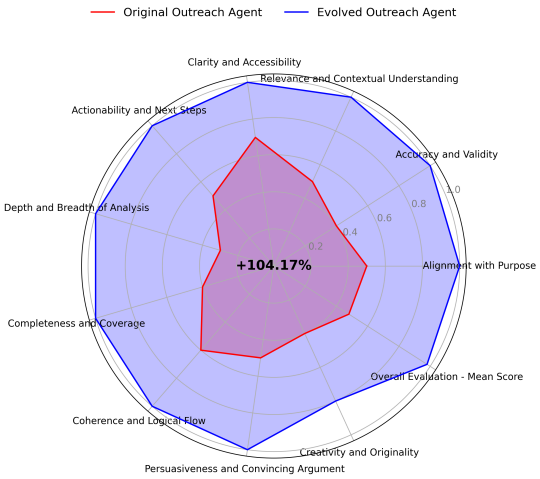

Figure 4: Outreach Agent Refinement

and engaging content. The evolved system incorporated four specialized roles, including an Audience Engagement Specialist, to enhance content development and audience interaction. To ensure relevancy, a dynamic content strategy emphasizing audience metrics and adaptability was implemented. The refined outputs significantly improved contextual relevance, accuracy, audience engagement potential, and clarity. The enhanced system positioned itself as a valuable resource for stakeholders interested in generative AI trends, highlighting the importance of specialized roles in content creation.

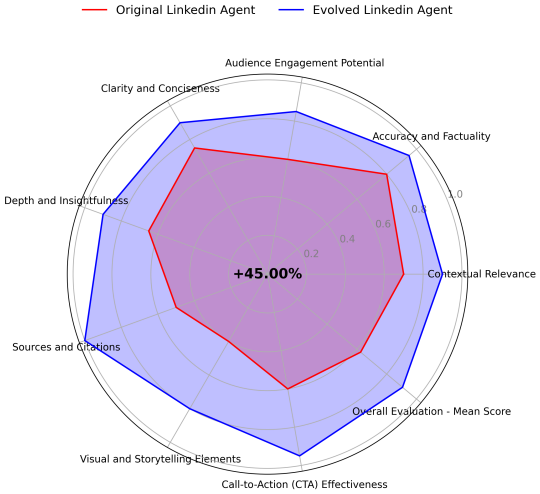

Figure 5: LinkedIn Agent Refinement

4.6 Meeting Agent

The meeting agent system designed for AI-powered drug discovery did not meet qualitative evaluation criteria due to poor alignment with industry trends and insufficient analytical depth. These shortcom-

ings limited its effectiveness in supporting pharmaceutical stakeholders. The evolved system introduced specialized roles, including AI industry experts and regulatory compliance leads, to provide comprehensive insights and ensure outputs were aligned with stakeholder needs. This overhaul aimed to enhance the system's relevance and actionability. The comparison of outputs revealed substantial improvements, with the evolved system achieving scores of 0.9 or higher across all evaluation categories. The refined system effectively addressed the needs of the pharmaceutical industry, demonstrating the impact of targeted modifications.

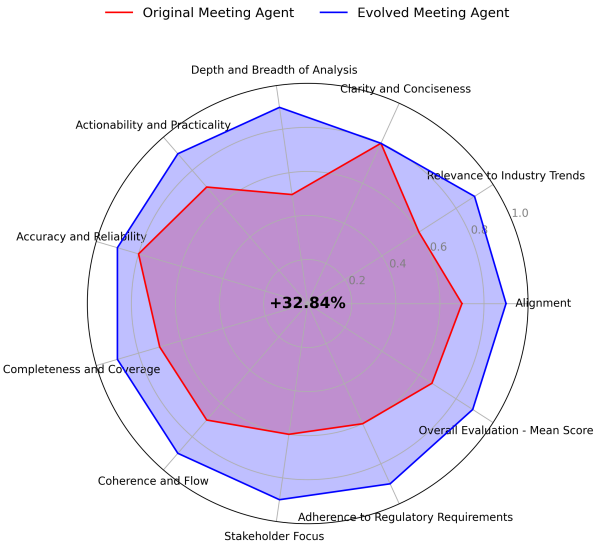

Figure 6: Meeting Agent Refinement

4.7 Lead Generation Agent

The lead generation agent for the "AI for Personalized Learning" platform faced challenges regarding alignment with business objectives and data accuracy. These limitations hindered the system's ability to generate valuable leads for the EdTech industry. New specialized roles were created to enhance the system, including Market Analyst and Business Development Specialist, to improve lead qualification processes and data integrity. The task structure was broadened to incorporate detailed analyses and actionable recommendations. The evolved agent system significantly improved evaluation criteria, including alignment with business objectives (91%) and data accuracy (90%). The enhancements underscore the importance of specialized roles and a structured approach in lead identification and qualification processes.

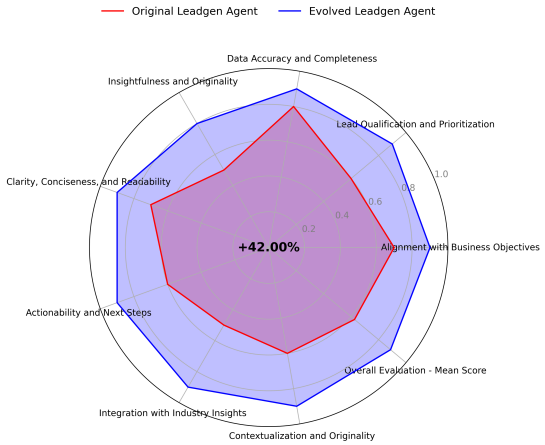

Figure 7: Lead Generation Agent Refinement

4.8 Evaluation Results

Figure 8 illustrates the comparative evaluation of original and evolved systems across various agentic applications, including customer support, medical imaging, supply chain management, and more. The box plots represent the distribution of evaluation scores on key criteria such as alignment, clarity, relevance, and actionability; with notable insights:

- **Consistent Improvements:** The evolved systems achieve markedly higher scores across all case studies, with median values near or exceeding 0.9, demonstrating the benefits.
- **Variability Reduction:** The reduced spread in scores for evolved systems reflects more consistent and reliable outputs, attributable to the specialized agent roles and tasks.
- **Targeted Enhancements:** Systems such as the Outreach Agent, Market Research Agent, and Medical AI Architect showcase substantial improvements, highlighting the value of user-centric and data-driven approaches.

These findings underscore the transformative impact of continuous refinement in agent systems, emphasizing the importance of domain-specific roles, strategic modifications, and adaptability to meet the dynamic needs of industries and users.

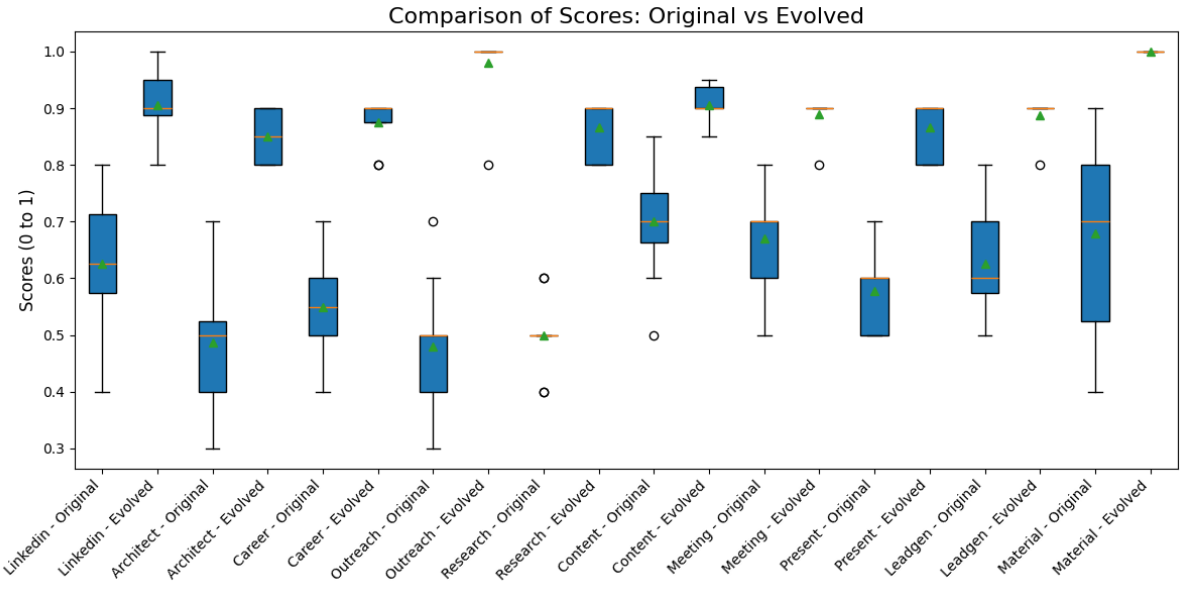

Figure 8: Original vs Evolved System Comparisons across Multiple Case Studies: Each pair of bars represents the evaluation scores for original and evolved systems, highlighting significant alignment, clarity, relevance, and actionability improvements achieved by refining agents, tasks, and workflows. The evolved systems consistently demonstrate higher scores, indicating the effectiveness of introducing specialized roles and targeted modifications.

5 Discussion

The collective findings from case studies illustrate the transformative impact of targeted modifications and the introduction of specialized roles within agent systems. Each system's evolution resulted in substantial improvements across various evaluation criteria, including alignment, accuracy, relevance, clarity, and actionability. These enhancements not only addressed the initial challenges faced by the original systems but also positioned the evolved agent systems as valuable tools for their respective domains. The experiments conducted across these diverse case studies underscore the necessity for continuous refinement in agent systems to meet the evolving needs of industries and users. Introducing specialized roles and a user-centric design focus have proven instrumental in enhancing output quality and effectiveness. The insights gained from case studies will serve as a basis for future agent systems, emphasizing the importance of specialization and adaptability in achieving optimal results. A critical insight from Sulc et al. (2024) is the importance of self-improving agents that adjust roles and interactions autonomously via feedback loops. Experiment results demonstrate the potential for dynamic adaptation and continuous enhancement, making it well-suited for environments with evolving objectives and conditions.

6 Conclusion

This paper presents a robust method for the autonomous refinement and optimization of Agentic AI solutions. The presented method continuously improves agent-based workflows by leveraging iterative feedback loops, hypothesis generation, and automated modifications, enhancing efficiency and effectiveness. The proposed method's autonomous nature minimizes human intervention, making it ideal for large-scale applications that require ongoing refinement. The method's scalability, flexibility, and ability to adapt to evolving objectives make it a powerful tool for optimizing complex AI agents.

While this method demonstrates promising advancements in Agentic AI, several avenues for future exploration could further enhance its capabilities. Investigating the role of human-in-the-loop strategies can bridge fully autonomous operations and scenarios where nuanced human judgment may be beneficial, especially during the initial deployment or in environments with high uncertainty. This could lead to hybrid systems where human expertise augments autonomous agent decision-making, ensuring safety and reliability without compromising autonomy. Collaborations with industry partners will also help tailor the method to real-world needs, ensuring adaptability and impact.

7 Limitations

The proposed framework for the autonomous refinement of Agentic AI systems has certain limitations that warrant consideration. Using LLMs for feedback, hypothesis formation, and evaluation may lead to inaccuracies, lack of explainability, and biases stemming from their training data. The framework's effectiveness relies on well-defined evaluation criteria. Poor or biased criteria can result in suboptimal refinements, as agents cannot independently identify missing dimensions. Minimal human involvement can be problematic in high-stakes or ambiguous tasks, where nuanced judgment and ethical considerations, such as privacy or unintended consequences, are crucial. Iterative processes like hypothesis generation and evaluation are computationally intensive, potentially limiting adoption in resource-constrained settings.

References

Shengran Hu, Cong Lu, and Jeff Clune. 2024. Automated design of agentic systems. *arXiv preprint arXiv:2408.08435*.

Qian Huang, Jian Vora, Percy Liang, and Jure Leskovec. 2024. Mlagentbench: Evaluating language agents on machine learning experimentation. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pages 20271–20309. PMLR.

Sarah Johnson and Ming Liu. 2023. Professional agents: Evolving large language models. In *Proceedings of the 36th Conference on Neural Information Processing Systems (NeurIPS)*.

Tula Masterman, Sandi Besen, Mason Sawtell, and Alex Chao. 2024. The landscape of emerging ai agent architectures for reasoning, planning, and tool calling: A survey. *arXiv preprint arXiv:2404.11584*.

Jason Miller, Kate O'Neill, and Deepak Ranjan. 2024. Ai agents that matter: Performance, scalability, and adaptation in agentic systems. In *Proceedings of the 40th International Conference on Autonomous Systems*. Springer.

Arindam Mitra, Luciano Del Corro, Guoqing Zheng, Shweti Mahajan, Dany Rouhana, Andres Codas, Yadong Lu, Wei ge Chen, Olga Vrousgos, Corby Rosset, Fillipe Silva, Hamed Khanpour, Yash Lara, and Ahmed Awadallah. 2024. Agentinstruct: Toward generative teaching with agentic flows. *Preprint*, arXiv:2407.03502.

Alexander Pan, Erik Jones, Meena Jagadeesan, and Jacob Steinhardt. 2024. Feedback loops with language models drive in-context reward hacking. *Preprint*, arXiv:2402.06627.

Wei Pan and Lei Zhang. 2024. Autonomous evaluation and refinement of digital agents. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence*.

Jordan Smith, Liam O'Connor, and Divya Patel. 2023. Large model agents: State-of-the-art cooperation. In *Proceedings of the 31st International Conference on Learning Representations (ICLR)*.

Antonin Sulc, Thorsten Hellert, Raimund Kammering, Hayden Houscher, and Jason St John. 2024. Towards agentic ai on particle accelerators. *arXiv preprint arXiv:2409.06336*.

Tao Wang, Jing Li, and Rui Huang. 2024. Agentic skill discovery with large language models. *Journal of Artificial Intelligence Research*, 72:145–178.

Xiao Yu, Baolin Peng, Vineeth Vajipey, Hao Cheng, Michel Galley, Jianfeng Gao, and Zhou Yu. 2024. Improving autonomous ai agents with reflective tree search and self-learning. *Preprint*, arXiv:2410.02052.

A Report for a Refinement Iteration

A.1 Initial Hypotheses and Justifications

A.1.1 Introducing Specialized Agents

Hypothesis: Creating specialized agents for distinct tasks will enhance depth and specialization, resulting in more thorough and expert analyses.

- **Market Identification Specialist Agent**
  - **Role:** Market Identification Specialist
  - **Goal:** Identify a wide range of potential markets using advanced search tools.
  - **Tools:** SerperDevTool, WebsiteSearchTool
- **Consumer Needs Analyst Agent**
  - **Role:** Consumer Needs Analyst
  - **Goal:** Analyze consumer needs using web scraping tools.
  - **Tools:** ScrapeWebsiteTool

Rationale: Specialized agents focusing exclusively on specific tasks will bring deeper knowledge and more targeted approaches, improving the precision and quality of market research.

A.1.2 Tool Integration

Hypothesis: Better utilization of available tools (SerperDevTool, WebsiteSearchTool, ScrapeWebsiteTool) by specialized agents will result in more comprehensive and data-driven analyses.

Rationale: Leveraging tools designed for specific purposes (search and scraping) will provide richer datasets and insights, producing a more robust market analysis.

A.1.3 Redefining Existing Tasks

Hypothesis: Redefining tasks to align with specialized roles will increase efficiency and clarity in the workflow, leading to better outcomes.

- **Market Identification Task** (Assigned to Market Identification Specialist Agent)
  - **Description:** Identify potential markets for the new product using search tools.
  - **Expected Output:** A list of thoroughly researched potential markets.
- **Consumer Needs Analysis Task** (Assigned to Consumer Needs Analyst Agent)
  - **Description:** Analyze consumer needs in the identified markets using web scraping.
  - **Expected Output:** A detailed report on consumer needs supported by data from web scraping.

Rationale: Clear definition and reassignment of tasks will ensure each specialized agent can focus on their core activities, enhancing productivity and effectiveness.

A.1.4 Creating a New Task for Comprehensive Validation

Hypothesis: Adding a validation task to compile and confirm findings from specialized agents will ensure the final output is accurate and cohesive.

- **Market Confirmation Task** (Dependent on Market Identification and Consumer Needs Analysis)
  - **Description:** Validate and compile the final list of potential markets and their needs.
  - **Expected Output:** A comprehensive and validated report on potential markets and consumer needs.

Rationale: A final validation and compilation step will integrate insights from both specialized agents, ensuring the report is consistent and logically structured, thereby enhancing overall output quality.

A.2 Revised Workflow

1. **Market Identification Specialist Agent** executes the Market Identification Task.
2. **Consumer Needs Analyst Agent** performs the Consumer Needs Analysis Task.
3. **Market Research Agent** consolidates the findings through the Market Confirmation Task and produces the final report.

A.3 Detailed Report Outlining the Rationale

A.3.1 Introducing Specialized Agents

- **Market Identification Specialist Agent**
  - **Role:** Market Identification Specialist
  - **Goal:** Identify a wide range of potential markets using advanced search tools.
  - **Tools:** SerperDevTool, WebsiteSearchTool
  - **Rationale:** This agent's specialization in identifying markets using advanced search tools will enhance the depth and precision of market identification, providing a stronger foundation for subsequent analysis.
- **Consumer Needs Analyst Agent**
  - **Role:** Consumer Needs Analyst
  - **Goal:** Analyze consumer needs using web scraping tools.
  - **Tools:** ScrapeWebsiteTool
  - **Rationale:** By focusing exclusively on analyzing consumer needs using web scraping tools, this agent can generate more detailed and data-driven insights into consumer behavior and preferences.

A.3.2 Redefining Existing Tasks

- **Market Identification Task**
  - **Description:** Identify potential markets for the new product using search tools.
  - **Expected Output:** A list of thoroughly researched potential markets.
  - **Agent:** Market Identification Specialist Agent
  - **Tools:** SerperDevTool, WebsiteSearchTool
  - **Rationale:** Assigning this task to the specialized agent ensures focused and comprehensive market identification using appropriate tools.
- **Consumer Needs Analysis Task**

- **Description:** Analyze consumer needs in the identified markets using web scraping.
- **Expected Output:** A detailed report on consumer needs supported by data from web scraping.
- **Agent:** Consumer Needs Analyst Agent
- **Dependencies:** Market Identification Task
- **Tools:** ScrapeWebsiteTool
- **Rationale:** This specialized task leverages web scraping to provide deep consumer insights, ensuring that data accurately identifies and supports consumer needs.

A.3.3 Creating a New Task for Comprehensive Validation

- **Market Confirmation Task**
  - **Description:** Validate and compile the final list of potential markets and their needs.
  - **Expected Output:** A comprehensive and validated report on potential markets and consumer needs.
  - **Agent:** Market Identification Specialist Agent
  - **Dependencies:** Market Identification Task, Consumer Needs Analysis Task
  - **Tools:** None
  - **Rationale:** This final validation task ensures consistency and logical structuring of the integrated insights from both specialized agents, resulting in a more reliable and cohesive report.

A.4 Comprehensive Comparison Report

A.4.1 Evaluation of New Output vs. Best-Known Output

- **Potential Markets Identification:**
  - **Best-Known Output:** Identified two markets (India B2C E-Commerce, Sustainable Steel).
  - **New Output:** Identified seven markets (Health and Fitness, Sustainable Products, Smart Home, Elderly Care, Pet Care, Remote Work, Educational Tech.).
  - **Evaluation:** The new output is more comprehensive, covering a broader range of markets.
- **Consumer Needs Analysis:**
  - **Best-Known Output:** Detailed for two markets.
  - **New Output:** Detailed for seven markets, including market needs, growth drivers, and potential products.
  - **Evaluation:** The new output provides a more comprehensive and structured analysis.
- **Actionability:**
  - **Best-Known Output:** Provides actionable insights for two markets.
  - **New Output:** Provides actionable insights for seven markets.
  - **Evaluation:** The new output offers more actionable insights due to its broader scope.
- **Product Development Recommendations:**
  - **Best-Known Output:** Clear recommendations for two markets.
  - **New Output:** Clear recommendations for seven markets.
  - **Evaluation:** The new output provides more comprehensive recommendations.
- **Completeness and Coherence:**
  - **Best-Known Output:** Completes essential steps for two markets.
  - **New Output:** Completes essential steps for seven markets.
  - **Evaluation:** The new output is more complete.

Conclusion: The new output is superior to the best-known output as it provides:

1. A broader and clearer identification of potential markets.
2. A more comprehensive and structured consumer needs analysis.
3. More actionable insights and recommendations for product development.
4. Greater completeness and coherence in the market research process.

Thus, the new variant (its code and output) has been saved as the best-known variant.