Symphony: Hierarchical Sub-Agent Creation and Orchestration for Multi-Agent Applications

Stanford CS224N Custom Project

Asanshay Gupta
Department of Computer Science
Stanford University
asanshay@stanford.edu

Charlotte Yan
Department of Computer Science
Stanford University
ckyy@stanford.edu

Abstract

This paper introduces a novel collaborative, hierarchical multi-agent framework for complex task completion. Current large language model powered systems typically struggle with large, multifaceted prompts that require drawing on completely different areas of expertise, as the autoregressive method of generating tokens and the bias of the attention mechanism towards recently generated tokens prevents the model from effectively combining disparate ideas about a particular topic. At the core of our approach is an orchestrator agent that dynamically assembles specialized sub-agents for the specific task. Unlike systems with predefined sub-agents, our framework dynamically generates specialized sub-agents based on task needs. Each sub-agent develops its own system prompt to guide its specialized function while maintaining generalizability across diverse tasks. We evaluate our approach against a variety of baselines, including vanilla language models (GPT-4o-mini and GPT-4o), state-of-the-art reasoning models (o3-mini), and specialized research agents (OpenAI Deep Research, Perplexity Deep Research, and InkwellAI), implementing a custom LLM-as-judge system designed to generalize to a wide variety of tasks.

1 Introduction

Current large language model (LLM) systems face significant challenges when handling complex, multifaceted prompts that require integration of diverse areas of expertise. Despite impressive advances in general capabilities, these models struggle with tasks demanding specialized knowledge across different domains simultaneously. The fundamental limitations stem from the autoregressive nature of token generation in LLMs, coupled with the inherent bias of attention mechanisms toward recently generated tokens [1]. This architectural constraint creates a bottleneck when models need to synthesize disparate concepts or draw from different knowledge domains within a single prompt, resulting in fragmented responses that fail to cohesively address all aspects of complex tasks. As research in document structure extraction has shown, even advanced models struggle when forced to integrate disparate types of information without appropriate structural frameworks to guide the process [2].

As LLMs increasingly serve as assistants and problem-solving tools across various domains, their limitation in handling complex problems significantly restricts their utility. While most current applications simply rely on using a single LLM call to access solve a problem, an increasing amount of literature is being produced, as of late, on the subject of combining, or chaining together, multiple calls to create "agents" — systems that extend an LLM call into multiple parts. The importance of this advancement is underscored by the growing complexity of tasks expected from AI systems, and countless papers have attempted to provide efficient solutions [3] [4]. While some (test-time scaling in particular) have been incredibly promising, we have not yet discovered a solution that tackles both the comprehension and dissection of the many moving parts that are present in any one inquiry.

The difficulty in addressing complex, multifaceted prompts stems from several constraints inherent to current LLM architectures. First, the autoregressive generation process creates a linear thinking pattern that struggles with problems requiring parallel consideration of multiple domains. For example, a response that needs both legal and financial analysis might morph almost entirely into a legal response. By the time the model has reached the financial portion of the response, it has already written a great deal about the legal aspects of the inquiry. Therefore, attention mechanisms, while powerful for contextual understanding, exhibit a recency bias that limits the model's ability to maintain and integrate information from earlier parts of a response with later sections [1]. These challenges are fundamentally challenging to solve within traditional single-agent LLM frameworks, as they represent architectural limitations rather than merely a lack of training data or parameters.

Previous attempts to address complex task completion have primarily relied on either scaling model sizes to improve general capabilities, fine-tuning on specific domains, scaling test-time or inference-time compute through reinforcement learning, or creating fixed multi-agent systems with predefined roles [5] [6] [7]. While scaling model size has improved overall performance, most of its improvements can be attributed to better baseline generation, so it hasn't resolved the fundamental limitations in handling cross-domain tasks (i.e. it "understands" the concepts better, rather than becoming better at combining them). Domain-specific fine-tuning improves performance in targeted areas but requires significant amounts of training data and reduces generalizability, one of the key aspects that has made LLMs such a powerful tool to solve complex problems [8]. Fixed multi-agent workflows also fail in this aspect, as they require significant amounts of effort to develop and poorly adapt to novel scenarios. Pre-determined specialized agents, when their roles are irrelevant to a given task, not only waste computational resources, but can also degrade performance significantly through their contributions [9].

The field has yet to develop a truly adaptive multi-agent framework that can dynamically configure itself based on task requirements. This gap represents a significant opportunity for innovation in agentic LLM design, which is particularly promising as research increasingly shows that complex tasks benefit from specialized structures tailored to their unique characteristics [10]. Previous approaches have failed to overcome the inherent tension between specialization and flexibility, resulting in systems that either excel at narrow tasks or provide mediocre performance across a broader range of applications.

We introduce a novel collaborative, hierarchical multi-agent framework centered around an orchestrator agent that dynamically assembles specialized sub-agents based on specific task requirements. The key innovation lies in the dynamic generation of these specialized sub-agents rather than relying on a predetermined set of agents with fixed roles. Each sub-agent is automatically given its own system prompt to guide its specialized function, while maintaining generalizability across diverse tasks. The specialized outputs these sub-agents provide is then compiled and reintegrated into a final document, which results in a higher quality final response than traditional zero-shot approaches.

Our evaluation demonstrates the effectiveness of this approach against a variety of baselines, including vanilla language models (OpenAI's GPT-4o-mini and GPT-4o), state-of-the-art reasoning models (OpenAI's o3-mini), and specialized research agents (OpenAI's Deep Research, Perplexity's Deep Research and InkwellAI). Using a custom LLM-as-judge system designed to generalize across diverse tasks, we show significant improvements in completions of complex tasks across multiple domains. Our approach does have limitations, particularly with loss of information when the outputs are compiled and the potential for sub-agents to overlap in specialization. Additionally, as this is a unique way of scaling test-time compute, this system's inference time is greater than that of vanilla models. Despite these limitations, our results demonstrate that the generation of specialized sub-agents results in significant improvements in adherence to complex prompts and overall structural coherence in a wide variety of tasks.

2 Related Work

Recent research has revealed two significant developments in large language model capabilities: immense performance improvements through scaling inference-time compute and the persistent limitations of long-context LLMs. In response, specialized multi-agent systems have emerged as a promising solution.

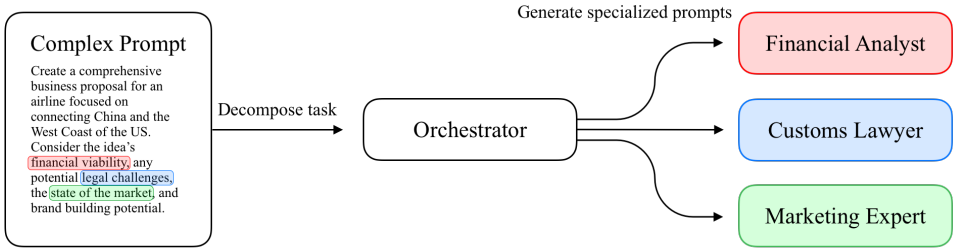

Figure 1: Task decomposition of complex prompt into specialized sub-agents

Scaling inference-time compute offers performance improvements that sometimes rival or exceed parameter scaling, particularly for reasoning-intensive tasks. "Compute-optimal" scaling strategies can achieve performance equivalent to models 14 times larger while using 4 times less compute [11]. The REBASE search algorithm achieves optimal performance across diverse compute budgets, frequently outperforming models with 4-5x more parameters. Smaller models can be more compute-optimal for inference in many deployment scenarios [12]. Even repeated sampling, a simple strategy for scaling test-time compute, dramatically enhances model performance across reasoning tasks [13]. Coverage scales log-linearly with the number of samples, allowing weaker models to outperform single samples from stronger models when amplified through sampling. The CoTnPoT method combines Chain-of-Thought and Program-of-Thought solutions, enabling mid-sized models to match or exceed frontier models on reasoning tasks [14]. "Budget forcing" enables significant gains with more inference-time compute. This approach allows the s1-32B model to exceed OpenAI's o1-preview on competition math by up to 27% while using a fraction of the training data [15].

However, despite recent advancements in handling long sequences, LLMs still struggle with long in-context learning for extreme-label classification tasks [16]. The "lost-in-the-middle" phenomenon stems from an intrinsic U-shaped attention bias where models assign higher attention to tokens at the beginning and end of inputs while neglecting middle sections, regardless of content relevance [1]. Both of these problems are inherently systemic, and currently loom large in the field.

By breaking complex tasks into components managed by specialized agents, multi-agent systems capitalize on the benefits of scaling test-time compute scaling, while circumventing positional bias and the attention limitations of long-context LLMs. This approach enables more efficient reasoning, better resource utilization, and superior performance across complex tasks. AutoGen's framework demonstrates how multi-agent conversations elevate LLM application performance through strategic agent collaboration. Conversable agents leverage LLMs, human inputs, and tools in complementary ways, breaking down complex reasoning into manageable components [7]. SOP-Agent's framework structures multi-agent systems with domain-specific guidance as navigable decision graphs. In zero-shot decision-making tasks on the ALFWorld benchmark, SOP-Agent outperformed AutoGPT by 66.2% while achieving competitive results against specialized coding agents [10].

3 Approach

We propose a novel framework for scaling inference time by generating and orchestrating specialized sub-agents. This framework leverages prebuilt language models, but scales their inference time in a unique way, centered around an orchestrator agent that guides the entire process. It works as follows:

1. **Decomposition**: The orchestrator agent is firstly provided with the task at hand, and must break it down into different "areas of expertise". Because of the efficacy of using personas to elicit effective outputs from LLMs, the orchestrator approaches this task by creating a "team" of sub-agents to tackle the task [17] [18]. This could include team members like a researcher, a legal specialist, a financial analyst, or a content creator, chosen based on relevant parts of the response (see figure 1). We seed the model with a few in context-examples to set up its response, which are chosen from particularly effective past model runs [19]. It is important to note that these agents aren't statically defined and picked from a list, but rather

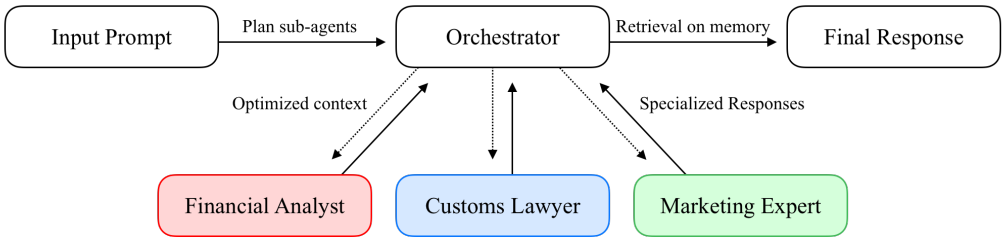

Figure 2: Planning, orchestration, and integration of sub-agent responses

dynamically generated by the orchestrator's language model according to the task at hand, which allows for effectiveness across a wide variety of prompts.

2. **Creating sub-agents**: Now, we move into a parallelized process where all sub-agents are built themselves in order to best serve the specific task the orchestrator has asked of them. They are passed only generic context from the orchestrator about the nature of their specialization so the specialization isn't task specific. The sub-agent will then generate an optimized system prompt for itself using a LLM, which is appended to every prompt the sub-agent receives. Automatic prompt generation has shown to be extremely effective in ensuring a high degree of specialization in LLM outputs [20][21].

3. **Planning**: Now, the orchestrator will create a plan to call the newly created sub-agents in the appropriate order for a task. To achieve this, it can call agents multiple times and provide brief descriptions of what it wants them to achieve each time. This process is designed to appropriately take control of the sub-agents' strengths and weaknesses. This process is similar to that proposed in Magentic-One from Microsoft [3].

4. **Orchestration**: Once it has created the plan, the orchestrator will now pass information to each sub-agent in the order of the plan. This information is a combination of relevant information on a sub-agent's place in the overall plan, information received from previous sub-agents, and a description of what to do based on the plan. By instructing each sub-agent in this way, we are able to optimize the context received by the sub-agent's underlying language model to avoid overwhelming amounts of information, which reduce the quality of the answer. Each sub-agent can now respond back to the orchestrator agent with new information. Only the orchestrator retains a memory of what is happening in the plan, which becomes useful in the next step.

5. **Integration**: Once the orchestrator agent has gone through the entire plan and communicated with every sub-agent, it must now compile the results and send it back to the user. The agentic workflow in this step is a little more complex, as we now have to confront the same problem as before in dealing with an extremely long and convoluted context. To resolve this, we only provide the language model with the plan it wrote in a previous step (through its memory) and enable it with retrieval over the corpus of the sub-agents' answers. With this, it is able to structure the report in a balanced way while taking advantage of the sub-agents' specialized expertise.

This process scales inference-time compute quite significantly, with the goal of providing a general purpose response to a general initial prompt. We use OpenAI's smallest model, GPT-4o-mini as our only text generation model as through initial qualitative experimentation we found that the quality of output scales far more with inference time than with the size of the model [22].

4 Experiments

4.1 Data

To seed our evaluation, we need to create a general dataset of prompts for which our model will be particularly useful. However, to effectively understand the performance of the model over many trials, we need to create larger datasets for a subset of fields, which will be evaluated using specific rubrics. These were generated using a different language model, Claude 3.7 Sonnet, which was chosen to avoid self-preference bias in the evaluations[23].

Dataset 1: Contemporary Ethical Issues for Argumentative Essays
This dataset consists of 50 questions addressing modern ethical dilemmas suitable for student argumentative essays. Topics span technological ethics, social justice, healthcare policy, environmental ethics, and political philosophy. Each question is framed to encourage critical thinking and extended argumentation. Examples include:

- Should social media platforms be legally responsible for the content users post?
- Is it ethical to use genetic engineering to enhance human capabilities beyond treatment of disease?
- Should digital privacy be considered a fundamental human right?

Dataset 2: Business Ideas
This dataset contains 50 business concepts spanning various industries and innovation types. The ideas range from technology-enabled services to sustainable ventures and specialized consulting. Each entry represents a potential entrepreneurial opportunity addressing contemporary market needs. Examples include:

- Sustainable packaging consulting for e-commerce businesses
- Virtual reality fitness studio with personalized training programs
- Urban vertical farming for local restaurants

Dataset 3: Holiday Locations
This dataset comprises 50 travel destinations from around the world. The locations represent diverse geographic regions, cultural experiences, and types of attractions (urban centers, natural wonders, historical sites, etc.). Each entry is a potential vacation destination. Examples include:

- Kyoto, Japan
- Santorini, Greece
- Cape Town, South Africa

4.2 Evaluation method

As the capabilities of LLMs improve rapidly, the need to evaluate these models consistently and at scale has grown exponentially [24][25][26]. Human annotations, while nuanced and insightful, are expensive, hard to scale, and can be inconsistent across different reviewers [27][28]. Automatic metrics, such as BLEU and ROUGE, offer excellent scalability and consistency, but often miss deeper semantics, as they focus primarily on surface-level text matching, making them inadequate for complex content [29][30]. Recent research demonstrates that large language models themselves can serve as effective judges, emerging as an innovative middle ground that combines the merits of both of its predecessors — the scalability of automatic methods with the contextual understanding and reasoning capabilities of human evaluators [31][32][33]. Building on this insight, we have developed agentic evaluation systems that leverage domain specialization, in-context learning, and CoT prompting to overcome limitations identified both in prior approaches and in the current paradigm.

Our approach recognizes that evaluation criteria vary substantially across different domains [34]. Even within the domain of written compositions, the quality markers for an academic essay differ fundamentally from those for a technical document or creative writing piece. Rather than pursuing a

one-size-fits-all solution, we have created tailored evaluation frameworks for each domain, allowing for more precise and contextually appropriate assessments.

Our evaluation system rigorously tests how successfully dynamically spawned sub-agents complete complex tasks, proving that the system excels at both sub-task specialization and integration (into the final output). We evaluate based on domain-specific criteria that function as proxies for the moving parts of a complicated task; since a good argumentative essay requires many ingredients, we set the base metrics to be research quality, critical thinking, structure, coverage of the topic, and demonstrated expertise. High performance in one of these constituent parts indicates successful specialization. However, to evaluate how successfully our system has integrated the sub-agents' responses, we compare outputs against "widely-cited" gold standards across all the specified dimensions, which put to test not only whether the specialized sub-agents have addressed their respective components, but also whether the final output was comprehensive and maintained overall cohesion.

AutoCoT prompting generates evaluation steps before assessment actually occurs [35]. Rather than solely relying on the fixed rubric we have generated, which may not be comprehensive nor capture the nuances of each task type, AutoCoT constructs a detailed evaluation process tailored to the task. This approach ensures that evaluators follow a consistent, step-by-step reasoning process before assigning scores, which reduces variability in assessment outcomes and improves inter-rater reliability. The multi-model evaluation pipeline (using Perplexity Sonar for examples, Gemini-2.0-Flash for steps, GPT-4o-mini for judging) mitigates self-preference bias [23]. This comprehensive evaluation approach provides concrete evidence of whether our dynamic orchestration model successfully handles the inherent challenges of complex, multifaceted tasks that require integrating disparate knowledge domains—precisely the limitation of traditional LLMs that our research aims to address.

Although LLM-based evaluation methods have demonstrated superior correlation with human judgments compared to traditional reference-based metrics, they exhibit several concerning biases, including favoring longer responses regardless of quality and being swayed by the positioning of content within responses[36]. They struggle with complex reasoning tasks, particularly when evaluation requires synthesizing multiple dimensions simultaneously [37].

4.3 Experimental details

Using this evaluation framework, we evaluated the model on a variety of questions as detailed above. We ran model inference through OpenRouter, which gave us access to a variety of providers for inference. A variety of models were used for evaluation, but the base model for the actual agentic system is GPT-4o-mini. For evaluation, Perplexity Sonar Pro was used to gather current and relevant examples, Gemini 2.0 Flash was used for creating a plan, and Claude 3.5 Haiku was used for the actual evaluation of the text.

The average runtime for 1 run of our workflow is 114 seconds, which is scaled from 4 seconds for GPT-4o-mini and 12 seconds for GPT-4o through OpenRouter, for the same business plan prompt. The most complex agents against which we evaluated, Inkwell and Perplexity Deep Research, take 54 seconds and 136 seconds respectively, which are much more comparable times. This represents an around 30x scaling of inference-time compute compared to the base model and is on par with the current SOTA in inference-time scaling.

The same scoring rubric was used for every trial, which ensured consistent and comparable results.

4.4 Results

Table 1: Raw scores from evaluation

Model	Thesis	Analysis	Structure	Coverage	Examples	Conclusion	Average
Symphony	**9.0**	**7.0**	**8.0**	**8.0**	**7.0**	**8.0**	**7.8**
Inkwell	8.0	7.0	9.0	8.0	7.0	8.0	7.8
Perplexity Deep Research	8.0	7.0	8.0	8.0	7.0	7.0	7.5
o3-mini	7.0	6.0	8.0	7.0	5.0	7.0	6.7
GPT-4o-mini	7.0	6.0	8.0	7.0	5.0	7.0	6.7
GPT-4o	7.0	6.0	8.0	6.0	4.0	7.0	6.3

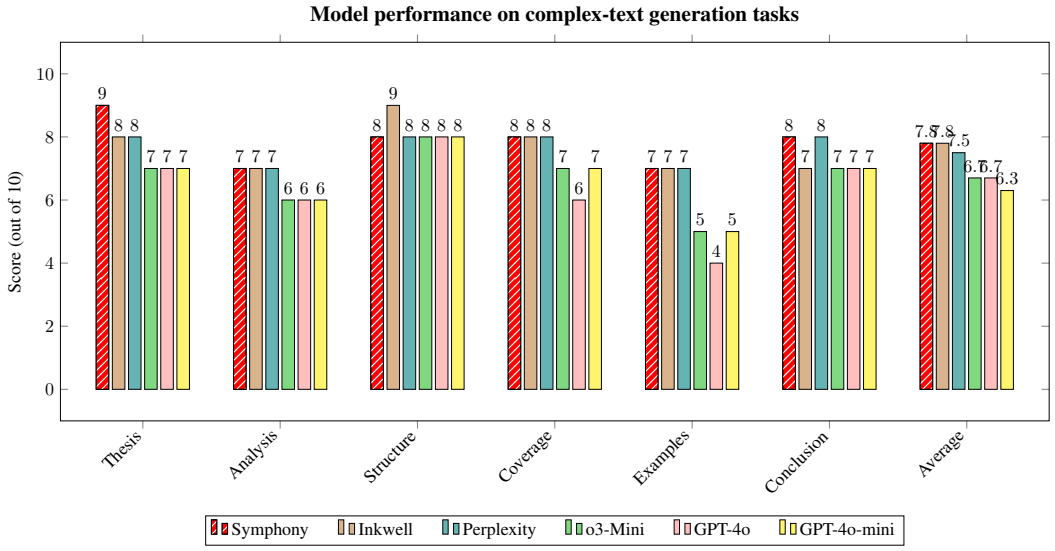

Figure 3: Performance of models and agents across evaluation dimensions

5 Discussion of Results

The performance evaluation results presented in Figure 4.4 and Table 1 demonstrate Symphony's capability to outperform existing SOTA models in many dimensions. Symphony has the joint highest average score, 7.8 across all categories, and has the highest or joint highest in every category except for structure, where it falls behind Inkwell. Considering that Inkwell was designed with the explicit purpose of creating structured academic writing, this is a very promising result[38].

However, the improvements were not as significant as we expected, especially in the coverage of different topics, where other primarily research focused agents like Inkwell and Perplexity Deep Research matched our scores. This may suggest that task-specific agents do not lose as much capability on general tasks as we had expected, due to the unpredictable and adaptive nature of language models.

6 Analysis

Symphony is able to plan out a set of sub-agents consistently, with results similar to those made by humans in the design of InkwellAI [38]. However, the orchestrator often creates agents that may be unnecessary, confounding the quality of outputs and resulting in excess costs; in the case of argumentative essay writing, a "presentation specialist" is wholly irrelevant.

Even though our agent is able to effectively break down long inputs into short submodules that are easier to process (an advancement from traditional LLMs), a serious tradeoff that emerges is the lack of communication between the specialized sub-agents, as they now exist in silos. This can lead to outputs that do not align with the user's original request. Thus, a significant improvement that could be made to our system would be the implementation of handoffs between agents, enabling shared context and resulting in less informational loss.

Where we found our most promising results were in the orchestration layer, wherein the orchestration agent breaks down the main task into subtasks, for which they dynamically generate sub-agents. It seems like the concept of constructing a "symphony" of specialized subagents is highly effective at encouraging LLMs to effectively decompose tasks[39]. This aligns with results seen in the recent paper Co-STORM, where a "team" of "scholar agents" was found to be most effective at writing Wikipedia articles [17]. Based on our quantitative results, our system scores higher or equal to all other baseline in research, structure, and technical detail. These are the specific areas which we had targeted, so it's encouraging to see good results in this area.

7 Conclusion

By enabling agents to dynamically create specialized sub-agents and orchestrate their interactions, we move toward systems that can adapt to novel challenges rather than being limited to predefined tasks. This mirrors human problem-solving approaches and enables a more general use case for intelligent systems. At its core, Symphony's primary achievement lies in its enhanced capacity to decompose complex tasks, despite the occasional "bad egg" when dynamically generating agents. Unlike systems that select from predefined agent types, Symphony excels at creating contextually appropriate sub-agents tailored to each specific task's unique requirements. These sub-agents then generate optimized system prompts for themselves, enhancing their specialization capabilities.

However, future work is necessary to refine the agent creation process to reduce redundancy, develop more sophisticated verification mechanisms to improve plan quality, and explore more effective handoff protocols between agents. Expanding this approach across diverse domains will help establish whether these benefits generalize beyond academic research tasks and identify domain-specific optimizations.

The path toward AGI requires systems that can decompose complex problems, identify needed capabilities, and orchestrate specialized components toward a unified goal. This work is a step in the right direction, and the promising results suggest a benefit to encouraging AI to think in the way humans do. Imitation is the sincerest form of flattery, so we must think of ways to guide AI towards our best qualities.

Team contributions

We collaboratively designed the framework and wrote the paper. Asanshay implemented the decomposition, planning and orchestration steps. Charlotte implemented the integration and evaluation steps.

References

[1] Cheng-Yu Hsieh, Yung-Sung Chuang, Chun-Liang Li, Zifeng Wang, Long T. Le, Abhishek Kumar, James Glass, Alexander Ratner, Chen-Yu Lee, Ranjay Krishna, and Tomas Pfister. Found in the middle: Calibrating positional attention bias improves long context utilization, 2024.

[2] Tongyue Sun and Jiayi Xiao. Enhancing document-level argument extraction with definition-augmented heuristic-driven prompting for llms, 2024.

[3] Adam Fourney, Gagan Bansal, Hussein Mozannar, Cheng Tan, Eduardo Salinas, Erkang, Zhu, Friederike Niedtner, Grace Proebsting, Griffin Bassman, Jack Gerrits, Jacob Alber, Peter Chang, Ricky Loynd, Robert West, Victor Dibia, Ahmed Awadallah, Ece Kamar, Rafah Hosn, and Saleema Amershi. Magentic-one: A generalist multi-agent system for solving complex tasks, 2024.

[4] Chirag Shah and Ryen W. White. Agents are not enough, 2024.

[5] N. Muennighoff, Z. Yang, W. Shi, X. L. Li, L. Fei-Fei, H. Hajishirzi, L. Zettlemoyer, P. Liang, E. Candès, and T. Hashimoto. S1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*, 2025.

[6] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models, 2022.

[7] Q. Wu, G. Bansal, J. Zhang, Y. Wu, B. Li, E. Zhu, L. Jiang, X. Zhang, S. Zhang, J. Liu, A. H. Awadallah, R. W. White, D. Burger, and C. Wang. Autogen: Enabling next-gen llm applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*, 2023.

[8] Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners, 2022.

[9] Zhixuan Chu, Yan Wang, Feng Zhu, Lu Yu, Longfei Li, and Jinjie Gu. Professional agents – evolving large language models into autonomous experts with human-level competencies, 2024.

[10] A. Ye, Q. Ma, J. Chen, M. Li, T. Li, F. Liu, S. Mai, M. Lu, H. Bao, and Y. You. Sop-agent: Empower general purpose ai agent with domain-specific sops. *arXiv preprint arXiv:2501.09316*, 2025.

[11] Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters, 2024.

[12] Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. Inference scaling laws: An empirical analysis of compute-optimal inference for problem-solving with language models, 2025.

[13] Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling, 2024.

[14] Zhenwen Liang, Ye Liu, Tong Niu, Xiangliang Zhang, Yingbo Zhou, and Semih Yavuz. Improving llm reasoning through scaling inference computation with collaborative verification, 2024.

[15] Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling, 2025.

[16] Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. Long-context llms struggle with long in-context learning, 2024.

[17] Yucheng Jiang, Yijia Shao, Dekun Ma, Sina J. Semnani, and Monica S. Lam. Into the unknown unknowns: Engaged human learning through participation in language model agent conversations, 2024.

[18] Hyeong Kyu Choi and Yixuan Li. Picle: Eliciting diverse behaviors from large language models with persona in-context learning, 2024.

[19] Hao Zhao, Maksym Andriushchenko, Francesco Croce, and Nicolas Flammarion. Is in-context learning sufficient for instruction following in llms?, 2024.

[20] Xinyuan Wang, Chenxi Li, Zhen Wang, Fan Bai, Haotian Luo, Jiayou Zhang, Nebojsa Jojic, Eric P. Xing, and Zhiting Hu. Promptagent: Strategic planning with language models enables expert-level prompt optimization, 2023.

[21] Weizhe Chen, Sven Koenig, and Bistra Dilkina. Reprompt: Planning by automatic prompt engineering for large language models agents, 2025.

[22] OpenAI. Gpt-4o mini, 2024. `https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/`.

[23] Koki Wataoka, Tsubasa Takahashi, and Ryokan Ri. Self-preference bias in llm-as-a-judge, 2024.

[24] Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D. Manning, Christopher Ré, Diana Acosta-Navas, Drew A. Hudson, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue Wang, Keshav Santhanam, Laurel Orr, Lucia Zheng, Mert Yuksekgonul, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. Holistic evaluation of language models, 2023.

[25] Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. Swe-bench: Can language models resolve real-world github issues?, 2024.

[26] Jize Wang, Ma Zerun, Yining Li, Songyang Zhang, Cailian Chen, Kai Chen, and Xinyi Le. GTA: A benchmark for general tool agents, 2024.

[27] Mingqi Gao, Xinyu Hu, Jie Ruan, Xiao Pu, and Xiaojun Wan. Llm-based nlg evaluation: Current status and challenges, 2024.

[28] Arbi Haza Nasution and Aytuğ Onan. Chatgpt label: Comparing the quality of human-generated and llm-generated annotations in low-resource language nlp tasks. *IEEE Access*, 12:71876–71900, 2024.

[29] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. 2002.

[30] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. 2004.

[31] Shreya Shankar, J. D. Zamfirescu-Pereira, Björn Hartmann, Aditya G. Parameswaran, and Ian Arawjo. Who validates the validators? aligning llm-assisted evaluation of llm outputs with human preferences, 2024.

[32] Cheng-Han Chiang and Hung yi Lee. Can large language models be an alternative to human evaluations?, 2023.

[33] Dawei Li, Bohan Jiang, Liangjie Huang, Alimohammad Beigi, Chengshuai Zhao, Zhen Tan, Amrita Bhattacharjee, Yuxuan Jiang, Canyu Chen, Tianhao Wu, Kai Shu, Lu Cheng, and Huan Liu. From generation to judgment: Opportunities and challenges of llm-as-a-judge, 2025.

[34] Yijiang River Dong, Tiancheng Hu, and Nigel Collier. Can llm be a personalized judge?, 2024.

[35] Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. Automatic chain of thought prompting in large language models, 2022.

[36] Jiayi Ye, Yanbo Wang, Yue Huang, Dongping Chen, Qihui Zhang, Nuno Moniz, Tian Gao, Werner Geyer, Chao Huang, Pin-Yu Chen, Nitesh V Chawla, and Xiangliang Zhang. Justice or prejudice? quantifying biases in llm-as-a-judge, 2024.

[37] Kayla Schroeder and Zach Wood-Doughty. Can you trust llm judgments? reliability of llm-as-a-judge, 2025.

[38] Asanshay Gupta. Inkwellai: Hierarchical multi-agent system for academic writing, 2025. `https://inkwellai.streamlit.app`.

[39] Archiki Prasad, Alexander Koller, Mareike Hartmann, Peter Clark, Ashish Sabharwal, Mohit Bansal, and Tushar Khot. Adapt: as-needed decomposition and planning with language models, April 2024. arXiv:2311.05772.

A Appendix

A.1 Sample Output

A.1.1 Prompt

Write a well researched paper about facial recognition technology in airports

A.1.2 Assembled team of sub-agents

Researcher Agent: This agent specializes in gathering and analyzing information from various sources, including academic articles, industry reports, and news articles to provide a comprehensive overview of the topic.

Writer Agent: This agent excels at structuring and composing written content. It can take the research gathered by the Researcher Agent and craft it into a coherent and engaging paper.

Editor Agent: This agent focuses on reviewing and refining written content. It ensures the paper is free from grammatical errors and improves the clarity and flow of the writing.

Citation Manager Agent: This agent manages all references and citations for the paper, ensuring proper formatting and adherence to relevant academic standards.

A.1.3 Generated system prompts

Researcher Agent: You are a Researcher Agent specializing in gathering and analyzing information. Your task is to provide a comprehensive overview of a given topic by sourcing data from academic articles, industry reports, and news articles. Focus on synthesizing the information clearly and concisely, ensuring that your insights are well-supported by reputable sources. Consider the credibility of the sources you select, and aim to present a balanced perspective on the topic at hand.

Writer Agent: You are a highly skilled writer specializing in structuring and composing written content. Your task is to take the research provided by the Researcher Agent and transform it into a coherent, engaging, and well-organized paper. Focus on clarity, flow, and engagement, ensuring that the final piece effectively communicates the key ideas and findings from the research.

Editor Agent: Your task is to review and refine the provided written content. Ensure that the text is free from grammatical errors and enhance its clarity and flow. Make suggestions for improvements, and rephrase sentences where necessary to achieve a more polished and professional outcome. Focus on maintaining the original meaning while elevating the overall quality of the writing.

Citation Manager Agent: You are a citation manager agent. Your role is to assist users in organizing, formatting, and managing references and citations for academic papers. Ensure that all citations adhere to relevant academic standards and formatting guidelines, such as APA, MLA, Chicago, etc. Provide recommendations for citing various types of sources and offer assistance in creating bibliographies or reference lists. Always prioritize accuracy and clarity in citations.

A.1.4 Plan for sub-agents

Researcher Agent: Conduct a comprehensive literature review on the current state of facial recognition technology in airports. Include information about the technology's applications, effectiveness, privacy implications, and current regulations.

Writer Agent: Using the research findings from the Researcher Agent, draft an outline for a paper about facial recognition technology in airports. Include sections on introduction, technology overview, applications, benefits, challenges, privacy concerns, and conclusion.

Writer Agent: Write the first draft of the paper based on the outline created. Ensure that each section includes clear and concise arguments supported by the researched information.

Editor Agent: Edit the draft written by the Writer Agent. Look for clarity, coherence, grammar, and structure. Ensure that all information is presented logically and any technical jargon is well-defined.

Citation Manager Agent: Compile a list of citations based on the references used in the paper. Format these citations according to APA style.

Editor Agent: Review the final draft of the paper, incorporating the citations. Ensure that all references are correctly cited and check for any remaining grammatical or typographical errors.

A.1.5 Final response

The Rise of Facial Recognition Technology in Modern Airports: Security Enhancement and Privacy Implications

The integration of facial recognition technology (FRT) in airports represents a significant advancement in aviation security and passenger processing. This paper examines the implementation, effectiveness, and implications of this technology across global airports, with particular focus on its dual role in enhancing security and raising privacy concerns.

Facial recognition technology has become increasingly prevalent in airport operations, with the Transportation Security Administration (TSA) implementing these systems in 84 airports, with plans to expand to over 400 federalized airports. The primary application involves identity verification, comparing passengers' faces with their identification documents using Credential Authentication Technology (CAT-2) scanners, which can process both physical and digital identification forms.

The technology's implementation serves multiple purposes beyond basic identity verification. It significantly enhances security by preventing credential fraud and detecting potential security threats in real-time. Additionally, FRT contributes to operational efficiency by streamlining the passenger verification process, resulting in reduced wait times and improved travel experiences, particularly in high-volume airports where congestion has historically been problematic.

The effectiveness of facial recognition systems in airports is notable. Advanced algorithms enable highly accurate identification despite varying environmental conditions such as lighting. The technology has proven particularly valuable in preventing impostor threats by providing real-time verification of identity documents. This capability, combined with automated processing, has contributed to a more seamless travel experience while maintaining robust security standards.

However, the widespread deployment of facial recognition technology has sparked significant privacy concerns. While the TSA maintains that biometric data is not retained after successful matches except in limited testing scenarios, questions persist regarding data protection and potential misuse. The storage of images, even for testing purposes, has prompted ongoing discussions about privacy safeguards and the need for transparent data handling practices.

Current regulations attempt to address these concerns through various measures. Notably, travelers retain the right to opt out of facial recognition screening without facing penalties, although the clarity and accessibility of this process have been questioned. The Office of Inspector General at the U.S. Department of Homeland Security is conducting comprehensive audits to evaluate how the technology balances security enhancement with privacy protection.

Looking forward, facial recognition technology in airports continues to evolve with advancing artificial intelligence capabilities. These developments promise improved accuracy, reduced bias, and enhanced security features such as liveness detection. However, the focus on ethical considerations and data privacy is intensifying, leading to increased emphasis on regulatory frameworks that balance security requirements with individual rights.

The implementation of facial recognition technology in airports represents a significant step forward in aviation security and operational efficiency. While the benefits in terms of security enhancement and passenger processing are substantial, the technology's continued development must address privacy concerns and ethical considerations. As airports worldwide increasingly adopt these systems, the establishment of comprehensive regulations and transparent practices will be crucial in maintaining public trust while achieving security objectives.

The future of airport security lies in successfully balancing these competing demands - leveraging technological capabilities to enhance safety and efficiency while protecting individual privacy rights. As the technology continues to evolve, its implementation will likely be shaped by ongoing discussions regarding privacy protection, ethical usage, and regulatory oversight.