Enhancing Reasoning to Adapt Large Language Models for Domain-Specific Applications

Bo Wen[1], **Xin Zhang**[1,2]
[1] IBM T. J. Watson Research Center, Yorktown Heights, NY, USA
[2] MIT-IBM Watson AI Lab, Cambridge, MA
Emails: bwen@us.ibm.com, xzhang@us.ibm.com

Abstract

This paper presents SOLOMON, a novel Neuro-inspired Large Language Model (LLM) Reasoning Network architecture that enhances the adaptability of foundation models for domain-specific applications. Through a case study in semiconductor layout design, we demonstrate how SOLOMON enables swift adaptation of general-purpose LLMs to specialized tasks by leveraging Prompt Engineering and In-Context Learning techniques. Our experiments reveal the challenges LLMs face in spatial reasoning and applying domain knowledge to practical problems. Results show that SOLOMON instances significantly outperform their baseline LLM counterparts and achieve performance comparable to state-of-the-art reasoning model, o1-preview. We discuss future research directions for developing more adaptive AI systems that can continually learn, adapt, and evolve in response to new information and changing requirements.

1 Introduction

The rapid advancements in large language models (LLMs) have revolutionized various aspects of artificial intelligence, enabling them to understand and generate human-like text with remarkable proficiency. However, adapting these general-purpose models to domain-specific tasks remains a significant challenge. In this paper, we introduce SOLOMON (System for Optimizing Language Outputs through Multi-agent Oversight Networks), a Neuro-inspired LLM Reasoning Network Architecture that leverages Prompt Engineering and In-Context Learning techniques, and demonstrate how SOLOMON can effectively adapt from its original design purpose in medical applications to a new domain: semiconductor layout design. Section 2 presents the SOLOMON architecture and highlights its design principles that contribute to enhanced adaptability.

To provide context for our experiment, we first examine how a designer might attempt to use ChatGPT (with GPT-4o mode) for a via connection design task in section 3. This exploration reveals a critical limitation: while LLMs can accurately recite textbook definitions of domain-specific concepts, they struggle to extract and apply expert knowledge to solve practical tasks. Human needs to translate high-level concepts into specific geometric requirements, which the LLM can then use to generate code for drawing shapes. This highlights the key challenge in adapting LLMs for domain-specific applications: their limited reasoning capabilities.

In section 4, we developed a set of 25 tasks ranging from basic geometric shapes to complex semiconductor structures, to evaluate our SOLOMON architecture against five different LLMs. These tasks assess spatial reasoning capabilities and adaptability across various complexity levels. Through these experiments, we demonstrate SOLOMON's superior performance compared to standalone LLMs, and reaching the level of state-of-the-art reasoning models like O1-preview.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

Our findings emphasize the crucial role of reasoning in enhancing LLMs' adaptability to diverse domain applications. This study contributes to ongoing research in adaptive foundation models, providing insights into how to improve reasoning capabilities with inspiration from neuroscience.

2 Neuro-inspired LLM Reasoning Network Architecture

SOLOMON's architecture (Fig. 1) is inspired by two neuro-inspired theories: Brain-like AGI Byrnes [2022] and the Free Energy Principle (FEP) Parr et al. [2022]. The former inspired us to use a pool of thoughts from multiple LLMs to discover the best reasoning plan. From the latter, we applied the FEP's main claim, *human attention focuses on minimizing the differences between goals and perceptions*, to select relevant information and avoid common pitfalls. The key components of SOLOMON are:

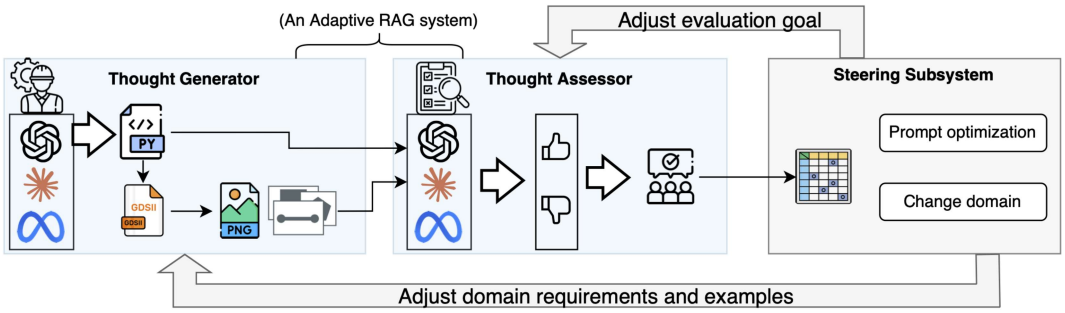

Figure 1: SOLOMON Architecture Diagram

Thought Generators: A diverse pool of LLMs generating thoughts for the target task. This component forms an efficient parallel search engine through the Tree-of-Thoughts Yao et al. [2023], Zhang et al. [2024], Besta et al. [2024a,b] and functions as an adaptive RAG system for the Thought Assessor. By pooling thoughts from multiple LLMs with distinct knowledge bases and reasoning abilities, it provides a more flexible and effective mechanism for sampling diverse ideas compared to common embedding-based RAG. This approach also mitigates biases inherent in single LLM knowledge bases. Noted that the individual LLMs in the Thought Generators can be further enhanced with proprietary knowledge through classic RAG techniques.

Thought Assessor: An LLM-based system that analyzes the proposed "Thoughts" to generate a refined output. It conducts in-context learning on the Thought Generators' output and follows the Free Energy Principle for goal-oriented assessments on consensus and differences. This approach enhances the LLM-as-a-Judge method Zheng et al. [2023], Lin and Chen [2023], enabling self-reflection Ji et al. [2023] and guarding against hallucinations Guerreiro et al. [2023], thus improving AI safety and reliability.

Steering Subsystem: A human-operated component that controls the attention of the Thought Generators and Thought Assessor. It uses Prompt Engineering to modify the goals of other components, enabling swift adaptation to different domain requirements through goal-directed exploration of the search space. This enhances the system's versatility across various applications by simply adjusting the attention focus.

This architecture offers significant advantages over traditional fine-tuning approaches, eliminating the need for recurrent fine-tuning associated with upgrading underlying LLMs or updating domain-specific knowledge. Basing on Prompt Engineering techniques, SOLOMON enables building more flexible AI systems capable of addressing diverse specialized contexts.

3 Problem: Spatial Reasoning and Domain Knowledge Application

Layout design in semiconductor processes requires not only generating correct basic geometric shapes but also spatial reasoning to create proper "layouts" that meet specific requirements. Via connections,

which create 3D electrical pathways between different chip layers, exemplify this challenge. While seemingly simple, typically consisting of circular vias and rectangular metal connections, they demand precise positioning and sizing to ensure no short or open circuits when building the 2D layout into a 3D structure.

We conducted a series of tests by providing a sketch(image) together with different text prompts to ChatGPT (GPT-4o). The sketch are color-coded to represent different layers (e.g., yellow for via, blue for metal layer, red for contact pad) to help ChatGPT understand the spatial relationships. Figure 2 illustrates the sketch inputs and corresponding ChatGPT-generated outputs for each test case. In

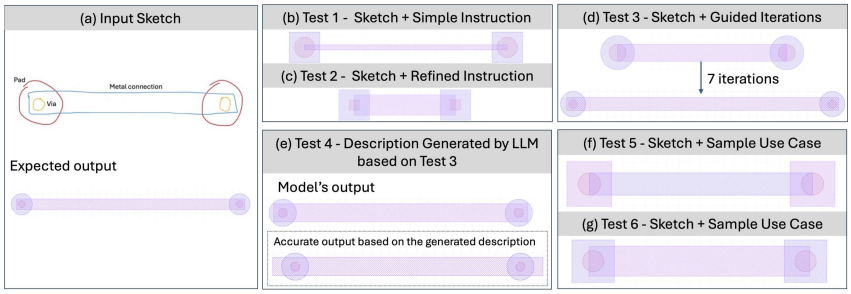

Figure 2: Sketch input and ChatGPT-generated outputs for the via connection experiment. The sketch depicts a desired layout with two vias connected by a metal layer and circular contact pads on top. The outputs show the progression of ChatGPT's understanding and refinement of the layout based on iterative feedback and context provided by the user.

Test 1, we provided the sketch (Figure 2(a)) with simple instructions. ChatGPT generated a runable code, but the metal layer width is narrower than the via. **Test 2** refined the instructions, but the output incorrectly used square vias [1], and failed to properly cover the vias with metal. In **Test 3**, we provided a more detailed description. After seven iterations of feedback, the LLM finally produced the correct layout (Figure 2(d)). See Appendix A.4 for detailed prompts and the iterative process.

Test 4 reversed the process by asking the LLM to create a detailed prompt based on the correct layout from **Test 3** (Appendix A.4). The LLM described each component's size and location in detail but hallucinated an additional requirement: *a 50-unit space between the vias and the edges of the metal connection*. This would result in the layout shown in the dotted rectangle in Figure 2(e), where the metal extends beyond the contact pad. Interestingly, when given this "wrong" prompt, GPT-4o ignored the added specification and produced a layout matching the original design, with the metal not extending beyond the pad. Code inspection revealed that the LLM used another requirement, *Leave a margin of 10 units between the edge of the metal and the pads*, to calculate the metal edge position in both x and y directions, although this statement was intended only for the y-direction margin. Using this version of prompt in the baseline evaluations (Section 4), o1-preview and Llama-3.1-405B each produced the "non-extending" version in one out of 5 runs, indicating some ambiguity in the specification.

To further test our hypothesis, we conducted **Tests 5** and **6**, removing numerical values from the prompt and incorporating domain-specific context (e.g., 3D packaging and Through-Silicon Vias). This approach, however, degraded LLM performance, revealing a critical limitation: while LLMs possess textbook knowledge of semiconductor concepts, they struggle to translate this into practical design requirements. For instance, LLMs failed to apply common engineering knowledge, such as using wider metal layers to connect vias or leaving margin space between components in different layers to account for layer misalignment.

This finding highlights a critical insight: to enhance the adaptability of LLM-based AI systems, simply increasing the model size to memorize more information is insufficient; instead, we should prioritize developing LLMs' reasoning capacity to effectively utilize their knowledge in practical problem-solving scenarios.

[1] square vias are difficult to fabricate in semiconductor etching process, so they are not used in practice

4 SOLOMON Performance and Comparison

To evaluate SOLOMON's effectiveness in enhancing spatial reasoning for semiconductor layout design, we created a dataset of 25 layout design tasks. These tasks were categorized into four groups: Basic Shapes 1 and 2 included simple geometric shapes such as circles, polygons and text, which serve as the building blocks for more complex layouts. The Advanced Shapes category involved more intricate designs, such as serpentine and spirals, to test the models' ability to handle complex geometries. Finally, the Complex Structures category included tasks that required the composition of multiple shapes to form functional layouts, such as a Dense Layer Diode (DLD) chip, MicrofluidicChip, and the ViaConnection test case. These tasks were designed to benchmark the AI systems' capability in generating layouts that are representative of real-world semiconductor design needs.

We provided task requirements with a system prompt asking the LLMs to use Chain-of-Thought to analyze the task and write Python code to create a GDSII output. The evaluation process involved running the generated code to produce GDSII files, which were then converted to PNG images. Human evaluators categorized the output into five categories: correct, scaling error, partially correct, shape error, and runtime error. Five LLMs (GPT-4o, Claude-3.5-Sonnet, Llama-3.1-70B, Llama-3.1-405B, and o1-preview) were used for the baseline experiment, with each task run 5 times per model. (See Appendix A.2 for details of prompts and example outputs.)

To evaluate SOLOMON, we utilized 20 thoughts generated by GPT-4o, Claude, and two Llama-3.1 models from the baseline experiment. We created four SOLOMON instances, each using one of these LLMs as a Thought Assessor, excluding o1-preview which served as our benchmark for state-of-the-art reasoning performance.

Figure 3 presents a performance comparison between the SOLOMON instances, their baseline counterparts, and the o1-preview model.

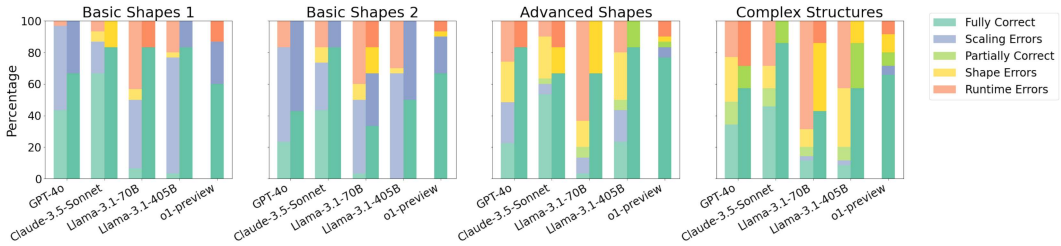

Figure 3: Performance comparison between SOLOMON instances, their baseline counterpart single LLMs, and o1-preview across different layout design task categories. Lighter colored bars on left represent baseline performance of individual LLMs, while darker bars on right show the performance of corresponding SOLOMON instances. O1-preview results serve as a benchmark for state-of-the-art reasoning performance.

The results demonstrate that the SOLOMON architecture significantly improves the performance of all four LLMs compared to their baseline. The most notable improvements are observed in the reduction of runtime errors, which can be attributed to the Thought Assessor seeing the error log of previous generated code and knowing what to avoid. This aligns with the design principle of the hierarchical, self-reflection mechanism, which aims to mitigate individual LLM's hallucination and blind-spot.

The second most notable problem in the baseline is scaling errors. We intentionally requested basic shapes to be drawn in millimeters to challenge the LLMs: they need to recall that the default unit in the gdspy library is micrometers. Sometimes LLMs simply failed to notice this and produced incorrect results. Additionally, each LLM seems to have bias when they hallucinate the default unit: Llama models perfers millimeters, Claude models sometimes recalls nanometers, and GPT-4o occasionally used meters. This issue was particularly problematic for Llama-3 models, when sometimes it correctly recalled the micrometer default but would insist that the user was wrong to request millimeters and proceed to draw without scaling, justifying it with comments like "not mm, as the GDSII format is in micrometers." Such "arrogant" behavior and misalignment with simple

instructions could be harmful for deploying LLMs as fully autonomous AI agents. A recent Nature paper Zhou et al. [2024] has also discussed similar observations.

The SOLOMON architecture improves performance across all models, including Llama-3. By incorporating diverse perspectives, it reduces stubbornness and increases accuracy. SOLOMON instances also show enhanced ability to handle shape errors and partial correctness issues, as the Thought Assessor can identify and correct errors related to arithmetic miscalculations or incorrect relative positioning of shapes. For more details, see Appendix A.3.

Comparing SOLOMON instances with o1-preview, we find that SOLOMON achieves comparable or superior results. All SOLOMON instances outperformed o1-preview in Basic Shape 1 categories, with the Claude-based SOLOMON surpassing o1-preview in 3 categories overall.

Interestingly, Llama-3 based SOLOMON instances also received significant performance boost, eventhough they don't receive the image inputs, suggesting that the thought assessment mechanism indeed works for more than just image understanding. Additionally, insufficient information linking images to corresponding code and error logs sometimes resulted in misinterpretation for GPT-4o and Claude.

Analysis of SOLOMON errors reveals that performance depends heavily on the quality and consensus of initial thoughts. Tasks with ambiguous requirements often leads to significant disagreement among initial thoughts, leading to confusion of Thought Assessor and degraded performance, see Appendix Table 9. These areas present opportunities for future improvements to the SOLOMON architecture.

5 Conclusion and Future Work

The introduction of the SOLOMON architecture significantly improved performance in semiconductor layout design tasks, particularly in reducing runtime errors and enhancing spatial reasoning capabilities. Our experiments demonstrated that SOLOMON instances outperformed their baseline LLM counterparts across various task categories, with some instances even surpassing the state-of-the-art o1-preview model in certain areas. This improvement validates the effectiveness of our neuro-inspired approach in enhancing LLMs' adaptability to domain-specific applications.

However, challenges remain in translating domain knowledge into practical design requirements. Our via connection experiment revealed that while LLMs can accurately recite textbook definitions of domain-specific concepts, they struggle to extract and apply expert knowledge to solve practical tasks. Investigating the potential of stacking multiple SOLOMON layers to form a hierarchical reasoning model capable of recalling and reasoning with domain knowledge for task-solving is one of our major future focus.

Other future research directions include: (1) Developing more comprehensive benchmark datasets for evaluating AI systems in layout design tasks. (2) Improving the linking between multimodal inputs (images and corresponding code+error) in the thoughts to enhance the Thought Assessor's interpretation abilities. (3) Exploring SOLOMON's performance when initial thoughts are of lower quality, and developing goal-oriented iterative learning mechanisms to improve thought quality through feedback loops. (4) Applying the SOLOMON architecture to a broader range of domain-specific tasks, such as power grid design and financial modeling.

In conclusion, while our results demonstrate the promise of LLMs as layout design copilots, further advancements in reasoning capabilities and domain knowledge application are necessary for their effective integration into semiconductor design processes and other specialized domains. The SOLOMON architecture represents a significant step towards creating more adaptive and capable AI systems for complex, domain-specific applications.

Open Source Code and Dataset: Visit our GitHub repository for the complete benchmark dataset of 25 tasks and LLM-calling code under the Apache 2.0 license at `https://github.com/wenboown/generative-ai-for-semiconductor-physical-design`. See Appendix A.1 for more details on Experiments Compute Resources requirements.

Acknowledgment: We thank Kuan Yu Hsieh for her valuable contribution in creating the dataset of 25 tasks with ground truth and her exploration work on the via connection test cases.