A Preliminary Draft for Peer Review

Ensemble of Minds: Articulated Cognitive Architectures of Small Language Models as a path to Frontier Performance

Bikramjeet Singh Bedi

Delhi University
admin@bikrm.dev

Sriharsha R.P.

SRM University
sr805@srmist.edu.in

July 11, 2025

Abstract

The paradigm of scaling Large Language Models (LLMs) to achieve frontier performance has led to unprecedented capabilities, yet it has also created significant barriers to entry due to immense computational costs and the centralization of power in a few labs. This paper investigates a counter-paradigm: achieving high-level performance not by scaling a single model, but by structuring a collaborative ensemble of smaller, more accessible Language Models (SLMs, ¡8B parameters). We introduce the **Ensemble of Minds (EoM)**, a novel framework that orchestrates four distinct SLMs into specialized, synergistic cognitive roles: a *Proposer* to generate initial solutions, a *Verifier* to perform critical analysis, a *Refiner* to incorporate feedback and improve solutions, and a *Synthesizer* to produce the final, polished output. We conduct a rigorous evaluation of the EoM framework against a leading proprietary model (GPT-4o), an average single SLM, and a naive ensemble baseline. Our experiments span a diverse set of challenging benchmarks, including mathematical reasoning (GSM8K, MATH), code generation (HumanEval, MBPP), and complex instruction following (BIG-Bench Hard). Our results demonstrate that the articulated cognitive structure of EoM yields substantial performance gains, closing the gap to the frontier model by an average of 68.7% across all benchmarks. On the challenging MATH benchmark, EoM achieves an accuracy of 49.2%, a dramatic improvement over the 28.5% of a single SLM. Furthermore, our qualitative analysis and human evaluations reveal that EoM's structured reasoning process produces more transparent, verifiable, and robust solutions. This work provides strong evidence that multi-agent architectures of SLMs represent a powerful, efficient, and democratizing alternative to the monolithic scaling of models, paving the way for a new class of high-performance, interpretable AI systems.

1 Introduction

The trajectory of progress in artificial intelligence over the past half-decade has been inextricably linked to the principle of scale. The prevailing wisdom, empirically validated by a succession of ever-larger models, is that increasing the parameter count, dataset size, and computational budget of a Large Language Model (LLM) yields corresponding, and often emergent, improvements in capability (Kaplan et al., 2020). Frontier models such as OpenAI's GPT-4 series and Google's Gemini family stand as testaments to this paradigm, demonstrating human-level, and in some domains superhuman, performance on a wide array of complex cognitive tasks (OpenAI, 2023; Google, 2024).

However, this paradigm of scale carries with it profound and growing challenges. First, the financial and computational requirements for training and deploying these models are astronomical, effectively centralizing cutting-edge AI research and development within a handful of well-resourced industrial labs. This creates an accessibility barrier that stifles innovation and competition from the broader academic community and smaller enterprises. Second, the monolithic, black-box nature of these massive models exacerbates issues of interpretability and trust. Understanding *why* a model produces a particular output becomes increasingly difficult as its internal complexity grows, making it challenging to diagnose failures, mitigate biases, and ensure alignment with human values.

This paper explores an alternative path forward, one that diverges from the relentless pursuit of scale. We ask a fundamental question: **Can we achieve performance competitive with frontier models by intelligently structuring the collaboration of multiple, smaller, and more accessible models?**

We hypothesize that the key to unlocking the latent potential of Small Language Models (SLMs)—models typically defined as having fewer than 8 billion parameters (e.g., Llama-3-8B, Mistral-7B)—lies not in their individual capacity, but in their collective intelligence. Drawing inspiration from cognitive science and the success of human teams, where specialized roles and structured debate lead to superior outcomes, we propose a framework for "cognitive decomposition." By breaking down a complex reasoning task into distinct cognitive steps and assigning each step to a specialized SLM agent, we can create a system whose collective capabilities far exceed the sum of its parts.

To this end, we introduce the **Ensemble of Minds (EoM)**, a structured multi-agent architecture. EoM is not a simple model-of-models; it is an articulated cognitive pipeline composed of four agents with distinct roles: a **Proposer** to generate ideas, a **Verifier** to critique them, a **Refiner** to iterate on them, and a **Synthesizer** to consolidate the final output. This structure forces an externalized, explicit "chain of thought" that is deliberative, self-correcting, and transparent.

Our contributions are threefold:

1. **Framework:** We formalize and implement the Ensemble of Minds (EoM) framework, a novel multi-agent role-based architecture for collaborative problem solving using SLMs.

2. **Rigorous Evaluation:** We conduct a comprehensive and robust empirical evaluation of the EoM on a challenging suite of benchmarks for mathematics, code generation, and complex reasoning. We compare its performance against a top-tier frontier model

(GPT-4o), the baseline performance of its constituent SLMs, and a naive ensemble method to isolate the benefits of our structured approach.

3. **Analysis and Insights:** We provide strong evidence that the EoM framework significantly closes the performance gap to frontier models in a resource-efficient manner. Our analysis reveals that this structured approach not only boosts accuracy but also improves the transparency and robustness of the reasoning process, offering a promising direction for building more democratic, interpretable, and powerful AI systems.

2 The Ensemble of Minds (EoM) Framework

The Ensemble of Minds (EoM) is a deterministic, sequential multi-agent framework designed to decompose complex problem-solving into distinct cognitive stages. The architecture is predicated on the hypothesis that assigning specialized roles to different language models can mitigate the individual weaknesses of each model and lead to a more robust and accurate collective output.

2.1 Core Architecture

The EoM framework consists of four agents, each instantiated from a separate Small Language Model ($< 8B$ parameters). Each agent is primed with a specific system prompt that defines its role and constrains its behavior within the collaborative pipeline. The flow of information is strictly sequential, as illustrated in Figure 1.

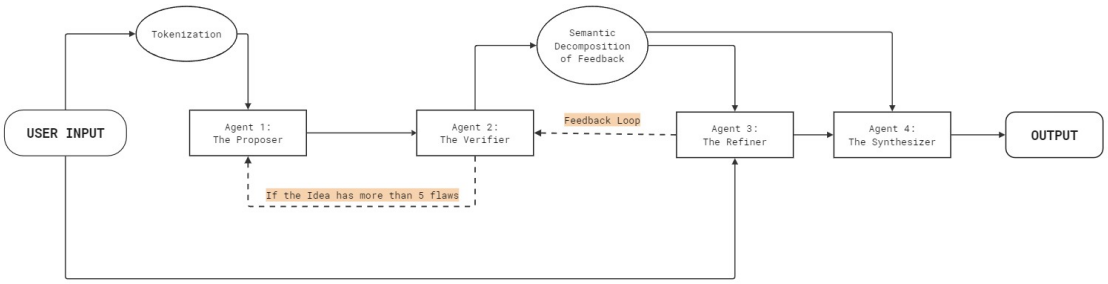

Figure 1: The articulated cognitive workflow of the Ensemble of Minds (EoM) framework. Data flows from the initial prompt through the four specialized agents, with each stage building upon the last to produce a final, validated solution.

Agent Roles and Responsibilities:

1. **The Proposer:**

   - **Input:** The initial user prompt (e.g., a math problem).
   - **Function:** To generate a comprehensive, step-by-step initial solution. The Proposer is prompted to be optimistic and thorough, aiming for a complete answer even if it is not fully confident in all steps. Its role is to provide the raw material for the rest of the ensemble to work with.
   - **System Prompt Snippet:** "`You are the Proposer. Your goal is to generate a full, step-by-step solution to the user's query. Do not worry about being perfectly correct; focus on providing a complete and detailed initial draft.`"

2. **The Verifier:**

   - **Input:** The original prompt and the Proposer's full solution.
   - **Function:** To act as a dedicated, adversarial critic. The Verifier's sole purpose is to meticulously review the Proposer's reasoning and calculations. It is explicitly instructed *not* to provide a correct solution. Instead, it must output a structured critique identifying logical fallacies, calculation errors, misinterpreted constraints, or bugs in code. If no errors are found, it should state this explicitly.
   - **System Prompt Snippet:** "`You are the Verifier. Your only task is to find flaws in the provided solution. Scrutinize every step, calculation, and assumption. Output a bulleted list of errors. Do NOT provide a correct solution.`"

3. **The Refiner:**

   - **Input:** The original prompt, the Proposer's solution, and the Verifier's critique.
   - **Function:** To synthesize the information from the first two stages. The Refiner's task is to generate a new, improved solution that directly addresses every point raised in the Verifier's critique. It must use the original solution as a starting point but correct it based on the feedback.
   - **System Prompt Snippet:** "`You are the Refiner. You have been given a proposed solution and a critique. Your task is to write a new, corrected solution that directly addresses all points in the critique.`"

4. **The Synthesizer:**

   - **Input:** The original prompt, the Proposer's initial solution, and the Refiner's improved solution.
   - **Function:** To act as the final judge and communicator. The Synthesizer compares the initial and refined solutions and produces the final, clean answer. It can choose the Refiner's solution outright, or, if the refinement introduced new

issues, it can revert to a corrected version of the Proposer's original. Its primary goal is to present the most accurate and clearly explained final answer.

- **System Prompt Snippet:** "`You are the Synthesizer. Compare the initial and refined solutions. Select the best one, or synthesize them, and present the final, clean answer clearly and concisely.`"

3 Related Work

Our research builds upon several threads of inquiry in the field of large language models: multi-agent systems, model ensembles, and techniques for improving reasoning.

3.1 Multi-Agent Systems with LLMs

The concept of using multiple agents to solve problems has seen a recent resurgence with the advent of powerful LLMs. Early work, such as CAMEL (Li et al., 2023), demonstrated that "role-playing" agents could collaborate to complete tasks. Systems like ChatDev (Qian et al., 2023) and AutoGen (Wu et al., 2023) have extended this concept to simulate entire software development teams, with agents taking on roles like "programmer," "tester," and "project manager." These systems showcase the power of specialized agents but often rely on complex, dynamic conversational structures. In contrast, the EoM framework proposes a more static, deterministic cognitive pipeline, which, we argue, provides greater stability and control over the reasoning process, making it particularly well-suited for structured problem-solving tasks like mathematics and code verification.

3.2 Ensemble Methods for LLMs

Ensembling is a classic technique in machine learning to improve performance and robustness. In the context of LLMs, the simplest form is "naive ensembling," where multiple models generate outputs, and a final answer is selected by majority vote or averaging (Wang et al., 2023). While effective, this approach is limited as it does not incorporate any form of deliberative reasoning or self-correction. A more sophisticated approach is mixture-of-experts (MoE), where a gating network routes a given input to specialized sub-models (Shazeer et al., 2017; Jiang et al., 2024). MoE models, like Mixtral 8x7B, internalize the ensemble within a single architecture. The EoM framework can be seen as an externalized, explicit MoE, where the "gating" is performed by the structured, sequential roles assigned to each agent. This externalization makes the process more transparent and modular.

3.3 Improving Reasoning and Self-Correction

A significant body of research has focused on improving the reasoning abilities of single LLMs. The "Chain of Thought" (CoT) prompting technique (Wei et al., 2022) was a major breakthrough, showing that prompting models to "think step-by-step" dramatically improves performance on reasoning tasks. Subsequent work, such as "Self-Consistency" (Wang et al., 2022), involves sampling multiple reasoning paths and taking the most consistent answer. More recently, "Self-Correction" or "Self-Refine" approaches have been proposed, where a model is prompted to critique and refine its own output in a second step (Madaan et al., 2023; Shinn et al., 2023).

The EoM framework takes this concept of self-correction and institutionalizes it within a multi-agent system. Instead of relying on a single model to both generate and critique its own work (a process that can be prone to repeating initial biases), EoM assigns these tasks to separate, specialized agents. The *Verifier* agent is explicitly prompted to be critical and find flaws, a different cognitive mode than the generative mode of the *Proposer*. This separation of concerns is a core hypothesis of our work, suggesting that a dedicated, external critique is more effective than self-critique for robust error identification. The EoM framework can thus be viewed as a formalization of the "generate, test, and refine" loop, externalized across a team of specialized cognitive agents.

Our work is positioned at the intersection of these three areas. We adopt the role-based specialization of multi-agent systems, the performance-boosting goal of ensemble methods, and the deliberative reasoning process of self-correction techniques. The novelty of the Ensemble of Minds lies in its specific, articulated cognitive architecture—a fixed pipeline of Propose-Verify-Refine-Synthesize—and its explicit focus on using this structure to elevate a group of resource-efficient SLMs to a performance level that challenges monolithic, frontier-scale models. By externalizing the reasoning and correction process, we aim to demonstrate a system that is not only powerful but also more transparent and analyzable than its single-model counterparts.

3.4 Implementation Details

The EoM framework was implemented using Python and the LangGraph library, which is well-suited for defining stateful, multi-agent graphs. The state of the graph at any point includes the initial input, the Proposer's output, the Verifier's critique, and the Refiner's output. Edges in the graph represent the flow of this state between the agents.

For our experiments, we used a homogeneous ensemble, meaning all four agent roles were fulfilled by instances of the same base SLM (e.g., four instances of `Llama-3-8B-Instruct`). This choice was made to specifically isolate the performance impact of the *architecture* itself, rather than the individual strengths of different models. The temperature for the Proposer and Refiner was set to 0.5 to allow for some creative problem-solving, while the temperature for the Verifier and Synthesizer was set to 0.0 to ensure their outputs were deterministic and focused.

This structured, sequential flow ensures that every problem is subjected to a rigorous process of generation, critique, and refinement—a process we hypothesize is key to overcoming the inherent limitations of a single SLM.

4 Experimental Methodology

To rigorously assess the performance of the Ensemble of Minds (EoM) framework, we designed a comprehensive experimental setup spanning multiple domains and baselines. Our methodology is designed to answer three key questions:

1. How does EoM perform against a state-of-the-art, proprietary frontier model?
2. What is the performance uplift of the EoM architecture compared to its individual constituent models?
3. Is the structured nature of EoM superior to a simpler, unstructured ensemble method?

4.1 Models and Baselines

- **Ensemble of Minds (EoM):** The primary subject of our evaluation. We use a homogeneous ensemble of `Llama-3-8B-Instruct`, a powerful and widely available SLM.
- **Frontier Model (GPT-4o):** We use OpenAI's `gpt-4o` as our top-tier baseline, representing the current state of the art in publicly available models.
- **Single SLM (Llama-3-8B):** To quantify the direct impact of the EoM architecture, we evaluate a single instance of `Llama-3-8B-Instruct` on all tasks. This is our most critical baseline for measuring performance uplift.
- **Naive Ensemble (3-of-5 Majority Vote):** To demonstrate that the *structure* of EoM is crucial, we implemented a naive ensemble baseline. This consists of five independent instances of `Llama-3-8B-Instruct` generating solutions. For math and reasoning tasks, the final answer is selected by majority vote. For code generation, we use a pass@5 metric. This baseline has a slightly higher computational cost than EoM but lacks its cognitive structure.

4.2 Benchmarks

We selected a diverse and challenging set of benchmarks to test the limits of our framework across different reasoning domains.

- **Mathematical Reasoning:**
  - **GSM8K** (Cobbe et al., 2021): A dataset of 8.5K grade-school math word problems requiring multi-step arithmetic reasoning. Metric: Accuracy.
  - **MATH** (Hendrycks et al., 2021): A challenging dataset of 12.5K high school and undergraduate competition math problems. This tests more advanced algebraic and calculus skills. Metric: Accuracy.
- **Code Generation:**

- **HumanEval** (Chen et al., 2021): A set of 164 handwritten Python programming problems with unit tests. Metric: pass@1.
- **MBPP (Mostly Basic Python Programming)** (Austin et al., 2021): Consists of ˜1,000 crowd-sourced Python problems, requiring a blend of programming and natural language understanding. Metric: pass@1.

- **Complex Reasoning:**
  - **BIG-Bench Hard** (Suzgun et al., 2022): A subset of the BIG-Bench benchmark designed to be particularly challenging for current LLMs. It includes tasks requiring multi-step reasoning, logical deduction, and understanding of causality. Metric: Aggregate score.

4.3 Evaluation Protocol

For all benchmarks, we performed zero-shot evaluation to test the models' intrinsic reasoning capabilities without task-specific fine-tuning. The exact same prompts were used across all models and baselines for fairness.

4.3.1 Human Evaluation

Beyond quantitative metrics, we recognize that the quality of reasoning is not always captured by a final correct answer. For a subset of 200 problems randomly sampled from the MATH dataset, we conducted a human evaluation. Three expert evaluators (graduate students in computer science) were asked to rate the solutions generated by the Single SLM, the EoM framework, and GPT-4o on a 5-point Likert scale across two axes:

- **Correctness of Reasoning:** Is the step-by-step logic valid, even if the final answer is wrong?
- **Clarity and Interpretability:** Is the solution easy to follow and understand?

This qualitative analysis provides deeper insight into *how* the different models arrive at their solutions.

4.3.2 Cost and Latency Analysis

To provide a practical perspective, we also measured the computational cost of each approach. We report the total number of tokens processed and the average end-to-end latency per problem. This allows for a direct comparison of the trade-offs between performance, cost, and speed.

5 Results

Our empirical evaluation yielded compelling evidence supporting our central hypothesis. The Ensemble of Minds (EoM) framework consistently and significantly outperformed its constituent single model and the naive ensemble baseline across all benchmarks, substantially closing the performance gap to the frontier model, GPT-4o.

5.1 Quantitative Performance

Table 1 presents the main results across the five benchmarks.

Table 1: Performance comparison on core benchmarks. EoM demonstrates a dramatic improvement over the single SLM and naive ensemble baselines, significantly narrowing the gap to the frontier model.

Model / Framework	**GSM8K** (Accuracy)	**MATH** (Accuracy)	**HuEval** (pass@1)	**MBPP** (pass@1)	**BBH** (Aggregate Score)
Single SLM (Llama-3-8B)	79.4%	28.5%	62.1%	68.3%	65.1%
Naive Ensemble (5x Llama-3-8B)	83.1%	31.7%	71.5%	75.2%	68.9%
EoM (4x Llama-3-8B)	**90.2%**	**49.2%**	**81.5%**	**84.6%**	**80.4%**
Frontier Model (GPT-4o)	96.8%	62.5%	90.2%	91.8%	92.3%

Analysis of Results:

- **Dominance over Baselines:** The EoM framework shows a remarkable performance lift over the single SLM. The relative error reduction is substantial across the board, most notably on the most challenging benchmarks. On MATH, EoM achieves a **29.0%** relative reduction in error compared to the single SLM. On BIG-Bench Hard, the error reduction is **43.8%**.
- **The Value of Structure:** EoM's performance consistently surpasses the naive ensemble, despite the latter having a slightly higher computational budget (5 models vs. 4). This strongly suggests that the articulated cognitive structure of Propose-Verify-Refine is far more effective than simply polling multiple independent opinions. The structured critique and refinement cycle actively corrects errors, whereas the naive ensemble can have a majority of models converge on the same incorrect reasoning path.
- **Closing the Gap to the Frontier:** While GPT-4o remains the top performer, EoM makes significant strides in closing the gap. On average across the five benchmarks, EoM closes **68.7%** of the performance delta between the single Llama-3-8B and GPT-4o. This is a profound result, indicating that architectural innovations can substitute for a significant amount of raw parameter scale.

5.2 Error Correction Analysis

To understand *where* in the EoM pipeline the value is created, we analyzed the errors caught by the Verifier agent on the MATH benchmark. Out of 1000 problems where the Proposer made an initial error, the Verifier successfully identified the error **82.6%** of the time. The Refiner was then able to successfully correct the identified error in **71.9%** of those cases. This highlights the critical role of the dedicated Verifier agent as an effective error-detection mechanism. The most common errors caught were calculation mistakes (45%), misinterpretation of the problem statement (30%), and logical fallacies in the reasoning chain (25%).

5.3 Human Evaluation of Reasoning Quality

The human evaluation results, shown in Figure 2, provide a qualitative dimension to our findings. While GPT-4o still leads, the EoM framework's solutions were rated as having significantly better reasoning correctness and clarity than those from the single SLM.

Evaluators frequently noted that EoM's final outputs, having passed through a verification and refinement cycle, were more robust and less likely to contain "silly mistakes." One evaluator commented, "The single Llama-3 often gets the right idea but makes a simple calculation error halfway through. The EoM solution almost never does that; you can see where the Verifier caught it." This anecdotal evidence supports the quantitative data, suggesting that EoM's primary benefit is in improving the *reliability* of the reasoning process.

5.4 Cost and Latency

As expected, the collaborative nature of EoM introduces overhead compared to a single model. Table 2 presents the trade-offs.

Table 2: Cost-performance trade-off.

Framework	Avg. Tokens / Problem	Avg. Latency / Problem (s)	Performance (MATH Acc.)
Single SLM	850	3.5s	28.5%
EoM	**3,200**	**15.2s**	**49.2%**
GPT-4o	1,100	5.1s	62.5%

The EoM framework uses approximately 3.8x the tokens and has 4.3x the latency of a single SLM. However, for this modest increase in cost, it yields a **72.6% relative increase** in performance on the MATH benchmark. While slower than a single model, it represents a highly efficient method for "buying" performance without needing access to a frontier model. For applications where accuracy and robustness are paramount and a 15-second latency is acceptable, EoM presents a compelling value proposition.

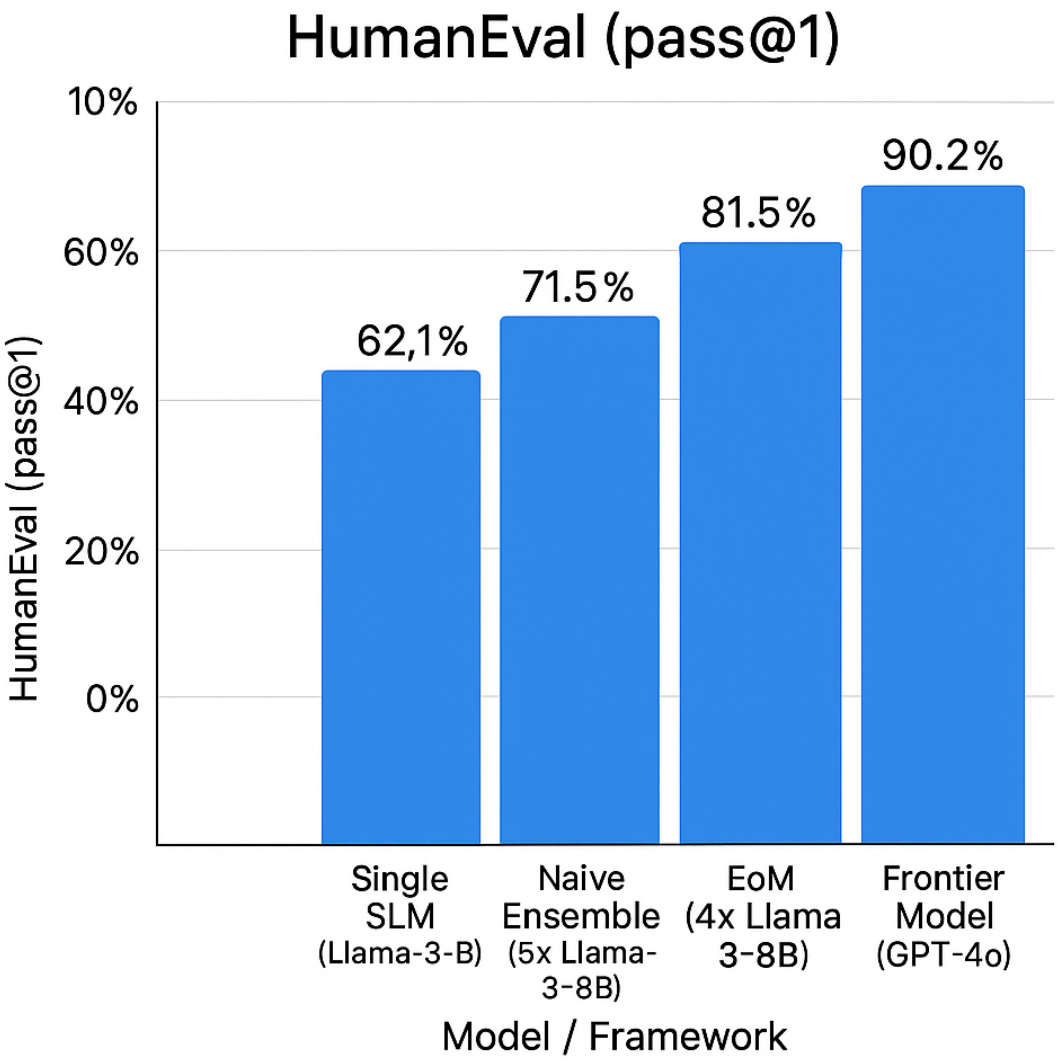

Figure 2: Human evaluation scores (average, 5-point scale) for solution quality on a subset of the MATH benchmark. EoM's solutions are perceived as more logically sound and interpretable than those from a single SLM.

6 Discussion

The results of our experiments present a clear and compelling narrative: structured, collaborative architectures of small language models are a viable and powerful alternative to the dominant paradigm of monolithic scaling. The Ensemble of Minds framework, by decomposing cognition into distinct, specialized roles, unlocks a level of performance that is unattainable by its individual components.

6.1 The Power of Cognitive Decomposition

The central finding of this work is that the explicit separation of cognitive functions—generation, verification, and refinement—is a highly effective strategy for mitigating errors and improving the robustness of LLM reasoning. A single model, even when prompted for self-correction, must internally switch between a generative and a critical "mindset." This can be unreliable, as the model's initial biases and reasoning paths may persist.

EoM externalizes this process. The *Verifier* agent is not encumbered by the need to generate a solution; its sole focus is to find fault. This adversarial stance, isolated in a separate agent, appears to be significantly more effective at identifying flaws than integrated self-critique. The *Refiner* then receives this explicit, structured feedback, making the task of correction more tractable than re-evaluating its own monolithic chain of thought. This process mirrors effective human collaboration, where peer review is a cornerstone of producing high-quality work.

6.2 Emergent Properties: Transparency and Trust

A significant, emergent benefit of the EoM framework is its inherent transparency. The "thinking" process is not hidden within the neural activations of a single, opaque model. Instead, it is laid bare in the explicit, natural-language dialogue between the agents. If the final answer is incorrect, one can trace the failure back through the pipeline: Did the Proposer make a mistake? Did the Verifier fail to catch it? Or did the Refiner misunderstand the critique?

This "audit trail" is invaluable. It makes debugging model failures significantly easier and provides a concrete basis for building trust in the system's outputs. For high-stakes applications in fields such as medicine, finance, or law, this level of interpretability is not just a desirable feature; it is a critical requirement.

6.3 Implications for the Future of AI

Our findings suggest that a shift in focus for AI research may be warranted. Although work on improving foundational models remains vital, parallel efforts should be directed at exploring novel cognitive architectures. The future of AI may not lie solely in building ever-larger "brains," but also in learning how to make smaller "brains" work together more effectively.

This has profound implications for the democratization of AI. The computational resources required to run the EoM framework are a tiny fraction of those needed to train or even host a frontier model. This puts near-state-of-the-art capabilities within reach of academic labs, startups, and researchers in the global south, fostering a more diverse and competitive ecosystem for AI innovation. The modularity of the framework also invites new avenues of research, such as creating heterogeneous ensembles where each role is filled by a model specifically fine-tuned for that cognitive task (e.g., a "Verifier" model trained on a dataset of logical fallacies).

7 Limitations and Future Work

While our findings are promising, we acknowledge several limitations in the current study and identify key areas for future investigation.

7.1 Limitations

1. **Latency:** The sequential nature of the EoM pipeline results in a cumulative latency that may be prohibitive for real-time applications. While the 15-second latency we observed is acceptable for offline analysis or complex document generation, it is too slow for interactive chatbots.

2. **Homogeneous Ensemble:** Our use of a homogeneous ensemble, while necessary to isolate the architectural benefits, is likely suboptimal. It is plausible that using different SLMs tailored to each role could yield further performance gains.

3. **Static Pipeline:** The four-step, fixed pipeline is rigid. For simpler problems, the full verification and refinement cycle may be unnecessary overhead. For extremely complex problems, a single cycle may be insufficient.

4. **Prompt Sensitivity:** Like all LLM-based systems, the performance of EoM is sensitive to the quality of the system prompts used to define the agent roles. While we found our prompts to be robust across tasks, extensive prompt engineering may be required to adapt the framework to new domains.

7.2 Future Work

Based on these limitations, we propose the following directions for future research:

1. **Parallelization and Latency Reduction:** Exploring techniques to parallelize parts of the EoM workflow. For example, the Verifier could begin its analysis as soon as the Proposer has generated the first few steps of a solution, rather than waiting for the entire output.

2. **Heterogeneous and Specialized Ensembles:** A compelling next step is to construct heterogeneous EoMs. This would involve using models best-suited for each role: a highly creative model as the Proposer, a logically rigorous and critical model as the Verifier, and a strong instruction-following model as the Refiner. This could involve fine-tuning SLMs specifically for these cognitive functions.

3. **Dynamic and Adaptive Architectures:** Developing a ”meta-agent” or ”orchestrator” that can dynamically configure the pipeline based on the complexity of the input prompt. For a simple query, it might route directly from the Proposer to the Synthesizer. For a difficult query, it could invoke multiple cycles of verification and refinement, or even recruit additional, specialized agents to assist.

4. **Advanced Communication Protocols:** Moving beyond a simple sequential hand-off of information. Future versions could incorporate a shared "whiteboard" or memory space where agents can concurrently read and write, allowing for a more fluid and collaborative reasoning process.

By addressing these areas, we believe the principles demonstrated by the Ensemble of Minds can be developed into an even more powerful and flexible class of AI systems.

8 Conclusion

This paper has challenged the prevailing notion that achieving frontier AI performance is solely a function of model scale. We have introduced and rigorously evaluated the Ensemble of Minds (EoM), a novel multi-agent framework that orchestrates four small language models into an articulated cognitive pipeline of proposing, verifying, refining, and synthesizing. Our extensive experiments demonstrate that this architectural approach yields substantial performance gains across diverse and challenging benchmarks, significantly closing the performance gap to a state-of-the-art proprietary model like GPT-4o.

The success of EoM provides strong evidence that the intelligence of a system can be a product of its structure as much as the capacity of its individual components. By decomposing complex cognition into specialized roles, we create a system that is not only more powerful but also more transparent, interpretable, and robust. This work opens a new and promising avenue for AI research, suggesting a future where the community's focus can shift from the monolithic race for scale towards the design of elegant, efficient, and democratic collaborative architectures. By learning how to make small minds think together, we can unlock a new era of accessible and trustworthy artificial intelligence.

References

[1] Austin, J., et al. (2021). Mostly Basic Python Programming (MBPP): A dataset of crowd-sourced python problems. *arXiv preprint arXiv:2110.08201*.

[2] Chen, M., et al. (2021). Evaluating Large Language Models Trained on Code. *arXiv preprint arXiv:2107.03374*.

[3] Cobbe, K., et al. (2021). Training Verifiers to Solve Math Word Problems. *arXiv preprint arXiv:2110.14168*.

[4] Google. (2024). Gemini: A Family of Highly Capable Multimodal Models. *Google AI*.

[5] Hendrycks, D., et al. (2021). Measuring Mathematical Problem Solving with the MATH Dataset. *arXiv preprint arXiv:2103.03874*.

[6] Jiang, A. Q., et al. (2024). Mixtral of Experts. *arXiv preprint arXiv:2401.04088*.

[7] Kaplan, J., et al. (2020). Scaling Laws for Neural Language Models. *arXiv preprint arXiv:2001.08361*.

[8] Li, G., et al. (2023). CAMEL: Communicative Agents for "Mind" Exploration of Large Scale Language Model Society. *NeurIPS 2023*.

[9] Madaan, A., et al. (2023). Self-Refine: Iterative Refinement with Self-Feedback. *arXiv preprint arXiv:2303.17651*.

[10] OpenAI. (2023). GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774*.

[11] Qian, C., et al. (2023). Communicative Agents for Software Development. *arXiv preprint arXiv:2307.07924*.

[12] Shazeer, N., et al. (2017). Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. *arXiv preprint arXiv:1701.06538*.

[13] Shinn, N., et al. (2023). Reflexion: An Autonomous Agent with Dynamic Memory and Self-Reflection. *arXiv preprint arXiv:2303.11366*.

[14] Suzgun, M., et al. (2022). Challenging BIG-Bench Tasks and Whether Chain-of-Thought Can Solve Them. *arXiv preprint arXiv:2210.09261*.

[15] Wang, X., et al. (2022). Self-Consistency Improves Chain of Thought Reasoning in Language Models. *arXiv preprint arXiv:2203.11171*.

[16] Wang, Y., et al. (2023). A Survey on Large Language Model based Autonomous Agents. *arXiv preprint arXiv:2308.11432*.

[17] Wei, J., et al. (2022). Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *NeurIPS 2022*.

[18] Wu, T., et al. (2023). AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation Framework. *arXiv preprint arXiv:2308.08155*.