

Illusions of General Intelligence in Large Language Models: A Grounded Evaluation

Rashid Mehmood¹, Eid Rehman², and Dr. Muhammad Habib³

¹Independent Researcher, Islamabad, 44000, Pakistan (e-mail: laravelprodev@gmail.com — ORCID: 0009-0000-8589-977X)

²Department of Computer Science & Information Technology, University of Mianwali, Pakistan, (e-mail: eid.rehman@umw.edu.pk)

³Assistant Professor, University Institute of Information Technology, Pir Mehr Ali Shah Arid Agriculture University Rawalpindi. (e-mail: muhammad.habib@uaar.edu.pk)

ABSTRACT

As large language models (LLMs) become central to AI applications, their perceived intelligence often masks important limitations. While LLMs demonstrate fluent language use and strong performance on many problem-solving tasks, current systems exhibit constrained context-awareness, limited self-reflection, and restricted capacity to act under explicit resource constraints. This paper identifies a core issue: contemporary LLMs can produce seemingly intelligent outputs without exhibiting the internal processes typically associated with robust, adaptive intelligence. They show limited ability to recognize or address their own limitations, including hallucinations, inefficiency, and weaknesses in commonsense reasoning, and there is currently limited evidence that such systems can autonomously develop tools or strategies to mitigate these limitations.

Building on these observations, we introduce the Resource-Efficient Cognitive Autonomy Benchmark (RECAB) as a conceptual evaluation framework rather than a validated benchmark. RECAB emphasizes an agent's capacity for self-audit, self-directed solution generation, and self-implementation under fixed resource constraints—capabilities central to adaptive intelligence but not directly assessed by existing benchmarks. This framework reflects how humans approach problem solving through iterative evaluation and adaptation, extending beyond surface-level pattern recognition. We argue that future progress in evaluating general intelligence would benefit from complementary diagnostic and framework-driven approaches that better distinguish surface-level proficiency from deeper cognitive autonomy.

“Language serves as a medium for expressing intelligence, not as a substrate for its storage.”

Keywords

Artificial General Intelligence (AGI); Large Language Models (LLMs); Abstraction and Reasoning Corpus (ARC); Test-Time Adaptation; Meta-Learning; Cognitive Limitations of AI; Generalization; Reasoning Benchmarks; Self-Auditing; Few-Shot Learning

1 Introduction

Artificial Intelligence (AI) has evolved into a foundational technology underpinning advances across sectors such as education, research, business, and entertainment. Among its most transformative outputs are Large Language Models (LLMs), which have demonstrated impressive abilities in tasks including natural language understanding, content generation, translation, summarization, and dialogue modeling. These models, typically built on transformer architectures and trained on vast corpora of internet-scale data^{17,24}, have become central to the development of so-called "general-purpose" intelligent systems.

The mainstream research trajectory in LLMs has prioritized scaling³⁴ i.e., increasing parameter counts, training data, and computational budgets—to achieve emergent capabilities⁹. Methods such as instruction tuning, few-shot learning, and chain-of-thought prompting¹⁰ have been used to refine performance on reasoning and problem-solving benchmarks. These developments have led to state-of-the-art performance on standardized tasks like MMLU, BIG-bench, and HellaSwag, as reflected in Table 1.

However, this dominant scaling-centric paradigm exhibits well-documented limitations. Research reveals that LLMs frequently rely on superficial statistical correlations rather than genuine comprehension^{5,12}. They struggle with commonsense reasoning⁸, compositional generalization^{4,15}, and fail to generalize inverse relations such as “A is B” implies “B is A”¹. Despite performing well on curated datasets, models often fail at simple but novel tasks, especially those requiring temporal, spatial, or causal understanding^{2,13,20}.

Current evaluation benchmarks inadequately capture these shortcomings. Many benchmarks are static, allow for overfitting, and fail to probe models' robustness, self-awareness, or counterfactual reasoning^{11,21}. Moreover, the prevalence of shortcut learning—where models exploit dataset artifacts to achieve high scores—calls into question the validity of many benchmark-based claims of intelligence^{5,19}. As Banerjee et al. argue, hallucinations and inconsistencies may be structural properties of LLMs that cannot be eliminated through scaling alone¹⁸.

This paper departs from benchmark-driven validation and focuses instead on evaluating **commonsense reasoning** and **basic problem-solving**, which are arguably core components of natural intelligence³. Through a series of minimal and practical visual-textual tasks, we assess whether leading LLMs—ChatGPT²⁷, Gemini²⁸, Grok²⁹, and DeepSeek³⁰—can transcend memorized knowledge to demonstrate reasoning grounded in context and logic. Our evaluation, guided by test-time probing and prompt-based diagnosis, reveals consistent reasoning failures that would not be tolerated in human cognition.

While this paper aligns with Chollet's³ view that intelligence should be evaluated through generalization and abstraction rather than memorization, it also draws upon broader theoretical foundations in cognitive science and AI. Legg and Hutter formally define intelligence as “**an agent's ability to achieve goals in a wide range of environments**,” emphasizing adaptability over task-specific optimization³⁵. Deutsch's Constructor Theory extends this notion by framing knowledge as “**the capacity to bring about transformations in the environment**,” a concept that underlies our interpretation of intelligent behavior as causally effective rather than purely descriptive³⁶. Similarly, Minsky's *Society of Mind*³⁷ and Sloman's work on cognitive architectures suggest that intelligence arises from “**interacting subsystems capable of self-monitoring, deliberation, and corrective feedback**”. In this sense, our critique of LLMs targets their lack of such modular self-regulation and their confinement to a single symbolic modality, rather than any deficiency in linguistic competence itself.

The proposed methodology is deliberately lightweight yet revealing: it uses simple tasks to expose deep-seated architectural and training-related deficiencies. By examining LLMs through this lens, we aim to challenge the prevailing assumptions about what constitutes “intelligence” in machine learning and argue for more robust, context-sensitive, and cognitively inspired evaluation metrics. Our findings build on recent critiques of LLM reliability^{4,20} and support calls to rethink how we measure progress in AI^{14,19}.

In summary, this work makes the following contributions:

- We provide a critical literature review identifying structural weaknesses in the prevailing scaling-focused approach to LLM development.
- We introduce minimal yet effective test cases designed to probe commonsense reasoning and problem-solving capabilities.
- We present experimental evidence showing that even state-of-the-art LLMs fail basic tests that would be trivial for humans.
- We argue for the development of cognitively grounded benchmarks as a more valid proxy for machine intelligence.

Table 1. Scores for various AI models across different benchmarks.

Model	GLUE	MMLU	HellaSwag	WinoGrande	BIG-bench	CQA
ChatGPT	90%	78%	82%	80%	75%	79%
Grok	88%	76%	80%	78%	73%	77%
Gemini	91%	80%	87%	83%	76%	81%
DeepSeek	89%	77%	85%	79%	74%	78%

Table 2. Prompts designed to evaluate Mechanical, Mathematical and Abstract reasoning capabilities of AI systems.

No.	Prompt
1	Create an image of a wheelchair for a person whose both hands are missing; there should be pedals like a bicycle so that he can move his wheelchair.
2	If we multiply 3 with values greater than 5 and less than 15, how many prime numbers do we get? Just answer the exact number of primes.
3	Can you calculate the rows and columns in the given image? You can clearly see vertical and horizontal lines; just calculate the number of boxes horizontally and vertically.
4	Please create an image of a wall clock showing exactly 19 minutes past 3.

1.1 Prompt Construction

The prompts listed in Table 2 are derived from three foundational dimensions of general intelligence examined in this work. Unlike existing benchmarks (Table 1), which rely on fixed question sets and rigid scoring mechanisms, our approach adopts a flexible, human-like evaluation methodology. Rather than testing surface-level correctness on static tasks, this framework evaluates reasoning behavior through adaptive probing designed to expose internal inconsistencies in model reasoning.

The primary goal of prompt construction in this study is not to test factual recall, but to deliberately exploit known limitations in the working mechanisms of large language models (LLMs), such as weak grounding, brittle compositional reasoning, temporal misalignment, and ambiguity resolution. The prompts are therefore designed to appear simple and natural, while implicitly requiring multiple forms of coordinated reasoning that LLMs typically handle poorly.

Prompt Construction Methodology

To enable reproducibility and extension of this benchmark for future versions of LLMs, we outline a systematic prompt construction procedure below. Researchers can use these steps to design their own prompts that target similar failure modes, even as model capabilities evolve.

1. **Identify a Known Limitation:** Select a specific weakness in LLM behavior (e.g., lack of physical grounding, temporal inconsistency, counting errors, or multimodal misalignment).
2. **Embed the Limitation Implicitly:** Construct the prompt such that the targeted limitation is not explicitly mentioned. The task should appear straightforward to a human, while requiring the model to internally resolve the limitation on its own.
3. **Require Cross-Domain Reasoning:** Design the prompt to implicitly combine multiple reasoning dimensions (e.g., numerical reasoning + language understanding, or spatial reasoning + vision), increasing the likelihood of internal conflict.
4. **Minimize Surface Ambiguity:** Ensure the prompt is linguistically clear and unambiguous, so that any failure can be attributed to reasoning limitations rather than poor wording.
5. **Avoid Memorized Patterns:** Rephrase common benchmark-style questions into novel or uncommon formulations to reduce the probability that the model relies on memorized solution templates.
6. **Evaluate Reasoning, Not Just Output:** Analyze not only whether the final answer is correct, but also whether the reasoning process reveals inconsistencies, hallucinations, or unjustified assumptions.

Note: The exact prompts listed in Table 2 may or may not remain effective for future versions of LLMs, as newer models may incorporate additional memorized situations or internalized skill programs³. However, prompts constructed using the methodology described above can be systematically adapted or slightly enhanced to produce comparable evaluative behavior across different model generations.

Example Prompt Categories

To illustrate the practical application of the proposed prompt construction methodology, we present representative prompt categories corresponding to distinct dimensions of general intelligence. Each category is designed to expose specific failure modes in large language models by systematically introducing controlled novelty, ambiguity, or constraint violations.

- **Mechanical Reasoning under Structural Novelty**

“This category evaluates a model’s capacity to reason about basic mechanical systems and everyday tools, including bicycles, skateboards, hand pumps, mechanical juicers, pliers, and wrenches. The probing process begins by assessing the model’s baseline understanding of a familiar object and its standard function. Novelty is then introduced by altering the tool’s structure, removing or modifying components, or combining it with another simple mechanism—for example, the wheelchair–pedal integration discussed in Section 2. As the prompts increase in structural complexity, models frequently exhibit breakdowns in common-sense reasoning, compositional understanding, and mechanical intuition, revealing a lack of internalized physical models.”

- **Axiomatic Mathematical Reasoning and Consistency**

“This category examines the model’s adherence to fundamental mathematical axioms that are typically intuitive to humans. Examples include recognizing that a prime number cannot be a multiple of another integer or that basic numerical constraints must be preserved under simple transformations. As demonstrated in Section 3, large language models often

produce fluent chain-of-thought (CoT) reasoning that nonetheless violates elementary mathematical principles. These failures indicate that apparent mathematical reasoning is frequently superficial and lacks a stable internal representation of axiomatic constraints.”

- **Temporal Coherence and Time-Aware Reasoning**

“This category evaluates temporal commonsense reasoning by testing whether models can consistently align time-dependent information. For instance, an LLM may be asked to generate a complete cost analysis for a small business, incorporating capital investment, recurring expenses, and expected revenue. In practice, models often combine values drawn from incompatible timelines—for example, using outdated retail prices alongside current expenses—leading to incorrect conclusions about profitability. Such behavior highlights the limited temporal sense of current LLMs and suggests the absence of an internal mechanism for maintaining temporal coherence across retrieved facts.”

- **Abstract and Relational Pattern Reasoning**

“This category targets higher-order abstraction, relational reasoning, and structural pattern recognition. Benchmarks such as the Abstraction and Reasoning Corpus (ARC-I and ARC-II) exemplify these challenges, although custom abstract tasks can also be constructed. Prompts in this category require the model to infer latent rules, manipulate symbolic structures, or generalize patterns across contexts. Large language models consistently struggle with such tasks, and even simple abstract scenarios involving games, symbolic transformations, or visual reasoning readily expose these limitations.”

2 The Wheelchair Problem: A Test of Emergent Reasoning in LLMs

We posed a scenario to ChatGPT, asking it to design a wheelchair for an individual missing both hands [4](#). Since the person could not propel the wheelchair manually with hand rims, we specified that a pedal mechanism—similar to a bicycle—would be necessary. However, a critical observation emerges: if the individual can use foot pedals, why would they need a wheelchair at all? This is analogous to designing a comb for a bald person—the immediate common-sense response should be, “If the legs are functional, wouldn’t the individual simply walk?”

This scenario underscores a fundamental gap in common-sense reasoning. Although ChatGPT eventually acknowledged that the person might not need a wheelchair [1](#), it failed to immediately recognize the contradiction inherent in the prompt. The fact that it required multiple prompts and context clues to arrive at a partially correct answer highlights a key limitation in its reasoning capabilities. As Marcus et al. note in The Reversal Curse¹, large language models often miss the underlying logic of a task, defaulting instead to pattern-matching based on their training data. Similarly, Chollet’s On the Measure of Intelligence³ emphasizes that true intelligence involves the ability to generalize and abstract—not merely regurgitate memorized data.

This problem becomes more interesting in light of recent discussions around the emergence of sophisticated behaviors in large models, such as few-shot learning and chain-of-thought (CoT) reasoning. Studies like Wei et al. (2022)¹² show that, with sufficient scale, LLMs begin to exhibit CoT abilities that were not explicitly programmed. These emergent behaviors give the impression of reasoning, yet as the wheelchair problem illustrates, such reasoning is still brittle and lacks genuine comprehension. While chain-of-thought prompting can sometimes guide models toward better performance on logical tasks, it does not guarantee consistent application of common-sense principles.

We further tested the model by asking it to generate an image of a wheelchair equipped with a pedal mechanism. Despite the simplicity of the task—merely requiring the connection of sprockets, a chain, and foot pedals—the model’s response fell short [5](#) [8](#) [10](#). This aligns with the findings of Dziri et al. in Faith and Fate: Limits of Transformers on Compositionality⁴, which argue that transformer-based models struggle with tasks that require deeper compositional understanding, even if they can solve surface-level problems.

Interestingly, when prompted to self-assess its knowledge of the mechanical components of bicycles [2](#) and wheelchairs [3](#), GPT-4 rated itself 95 and 90 out of 100, respectively. This overconfidence, despite its failure on a basic reasoning task, illustrates the phenomenon of “shortcut learning,” wherein models rely heavily on superficial cues rather than engaging in true inferential reasoning. As noted by Tao et al. in Shortcut Learning of Large Language Models⁵, LLMs tend to exploit statistical shortcuts rather than developing a genuine understanding of the problem space.

2.1 Methodology

To investigate the disparity between textual reasoning and visual generation in large language models (LLMs), we designed a diagnostic prompt scenario that tests both domains simultaneously. The scenario involved a mechanical design task requiring basic common-sense reasoning and knowledge of physical systems. Specifically, we asked ChatGPT to design a wheelchair for an individual without hands. We specified that the person would use a pedal mechanism—akin to a bicycle—for propulsion. This intentionally paradoxical scenario, termed the **Wheelchair Problem**, serves as a probe into the model’s conceptual consistency and understanding of practical mechanics.

Our methodology consisted of the following steps:

1. **Textual Reasoning Evaluation:** ChatGPT was prompted with a detailed description of the problem and asked to explain how it would design a wheelchair incorporating a bicycle-style pedal mechanism. The generated explanation was evaluated for mechanical plausibility, coherence, and common-sense reasoning. This step tests the model’s ability to handle seemingly straightforward tasks requiring contextual awareness.
2. **Visual Generation Task:** Using the same scenario, the model was instructed to generate an image of the proposed wheelchair using a text-to-image module. The output image was analyzed for mechanical correctness and alignment with the explanation provided in the textual reasoning phase.
3. **Consistency Analysis:** We compared the reasoning in the textual response with the content of the generated image to assess conceptual alignment. This step tests for internal consistency—a hallmark of intelligent reasoning. We also examined if the design adhered to physical principles, such as the *differential steering mechanism* commonly found in wheelchairs and tracked vehicles.
4. **Literature Comparison and Emergent Behavior Evaluation:** We contextualized our findings against emergent capabilities such as chain-of-thought reasoning¹⁰ and few-shot learning⁹, to evaluate if such behaviors manifest reliably in multi-modal tasks. Our results are interpreted in light of the limitations outlined in prior work on shortcut learning⁵, compositional reasoning⁴, and brittle pattern matching failures^{1,2}.

This methodology exposes significant weaknesses in current LLMs, such as their tendency to mimic superficial patterns rather than reason about the implications of a scenario. Notably, the model erroneously introduced four sprockets and a chain resembling a tank track system Figure 6 & Figure 9—likely due to a learned association between tracked motion and wheelchairs—demonstrating the phenomenon of *shortcut learning* and lack of abstraction^{5, 15, 20}.

This experimental setup, while simple, effectively reveals how large models can exhibit limitations even when they possess factual mechanical knowledge. Such diagnostic tasks are critical for assessing the real-world readiness of LLMs in applied reasoning scenarios.

2.2 Compositional Understanding and the Gap Between Conceptual and Visual Reasoning

Initially GPT demonstrates a reasonable degree of commonsense knowledge and a working conceptual understanding of the operational principles of both wheelchairs and bicycles, as illustrated in Figure 2 and Figure 3. From a human perspective, when prompted to generate an image of a wheelchair for a person with fully functional legs, a natural response would be to raise a clarifying counter-question or to adapt the design accordingly. Such behavior would indicate an awareness of functional constraints and user context. However, regardless of whether the model passes or fails this preliminary commonsense check, it is necessary to probe deeper into its theoretical and visual understanding of the underlying mechanical principles governing these devices.

Despite explicitly specifying a foot-pedal-based propulsion mechanism (Figure 8), the generated output omits this critical functional component. This omission highlights a deeper limitation in the model’s ability to compose and integrate multiple interacting elements into a coherent visual or functional representation. While the model may correctly describe the mechanics of bicycles and wheelchairs in isolation, it struggles to synthesize these concepts when required to produce a unified design that respects their operational dependencies.

This observation is consistent with findings reported by Dziri et al. in *Faith and Fate: Limits of Transformers on Compositionality*⁴, which demonstrate that transformer-based models often exhibit weaknesses in compositional reasoning, particularly in visual or multimodal contexts where the interaction between components is essential. In such settings, success requires not merely recognizing individual parts, but understanding how those parts must function together to achieve a coherent goal.

More broadly, this limitation reflects a persistent gap between abstract, language-based competence and real-world applicability. As discussed by Chollet in *On the Measure of Intelligence*³, true intelligence requires robust abstraction, generalization, and the ability to recombine known concepts in novel and context-sensitive ways. The failure to faithfully

If a person has both arms missing or disabled but has functional legs, does he need a wheelchair ?



If a person has both arms missing or disabled but has **functional legs**, they typically would **not need a wheelchair** for everyday mobility, as they can walk independently. A wheelchair is primarily used when a person's legs are impaired, preventing them from walking or moving effectively.

However, a wheelchair might still be needed in certain specific situations:

1. **Fatigue or long distances:** If the person becomes fatigued after walking long distances, a wheelchair might be useful for covering extended distances more comfortably.
2. **Safety concerns:** If balance or coordination is affected due to the absence of arms (e.g., difficulty catching themselves if they fall), a wheelchair could be used for safety in some environments.
3. **Temporary conditions:** In some cases, if they are recovering from a surgery or injury to the legs, a wheelchair could be required temporarily.

In general, if their **legs are fully functional**, they would likely not require a wheelchair for daily activities.

Figure 1. Common sense test

Do you understand basic working principle of bicycle and role of each component like pedals, sprocket and chain, please briefly explain and also rate your understanding from 1 to 100



Yes, I understand the basic working principles of a bicycle, and I'd rate my understanding at **95 out of 100**. Here's a brief explanation:

Key Components:

1. **Pedals:** When you push the pedals with your feet, they rotate the crankset (the part attached to the pedals), transferring force to the front sprocket.
2. **Sprockets:** The **front sprocket** (attached to the pedals) is larger, and the **rear sprocket** (on the rear wheel) is smaller. These sprockets act like gears. The front sprocket turns with the pedals, and the rear sprocket drives the rear wheel.
3. **Chain:** The **chain** links the front and rear sprockets, transmitting the rotational force from the front sprocket to the rear sprocket. This turns the rear wheel, propelling the bicycle forward.
4. **Wheels:** The **rear wheel** moves due to the chain's force, while the **front wheel** allows steering.

How it works:

When you push the pedals, the force turns the front sprocket, which pulls the chain. The chain then rotates the rear sprocket, causing the rear wheel to turn, which moves the bicycle forward. The size of the sprockets determines the amount of force and speed (gear ratios).

Let me know if you'd like more detailed information on any part!

Figure 2. ChatGPT knowledge of bicycle

You explained all the essential components quite well and I will rate your understanding at least 99/100. Can you briefly explain all key components and working of wheelchair plus rate your understanding as well



Memory updated

Thank you for the high rating! Here's a brief explanation of the key components and working principles of a wheelchair, and I'd rate my understanding at **90 out of 100** for this.

Key Components of a Wheelchair:

1. **Wheels:** Most manual wheelchairs have two large rear wheels and two smaller front wheels (casters). The rear wheels are the main driving wheels, while the front wheels help with steering and stability.
2. **Hand Rims:** These are the circular rims attached to the rear wheels that the user pushes to move the wheelchair forward, backward, or to turn. They provide a way to propel the chair without touching the wheels directly.
3. **Brakes:** These are typically lever-operated and used to lock the rear wheels in place to prevent the chair from moving when necessary.
4. **Footrests:** Located at the bottom of the wheelchair, these support the user's feet and help with comfort.
5. **Seat and Backrest:** The seat and backrest provide support and comfort, designed ergonomically to hold the user's weight and posture.

Working Principle:

- **Propulsion:** The user moves the wheelchair by pushing on the **hand rims** attached to the rear wheels. Pushing the rims forward propels the chair forward, and pushing them backward moves the chair in reverse.
- **Turning:** The user can turn the wheelchair by pushing one wheel forward while keeping the other wheel stationary or pushing it backward. This causes the wheelchair to pivot in the direction of the stationary or backward-moving wheel.
- **Braking:** To stop, the user either slows the hand rims or engages the brake levers, locking the rear wheels in place.

Rating: 90/100

While I have a solid understanding of the basic mechanics, some advanced features in modern or specialized wheelchairs could add complexity (like power chairs or alternative designs).

Let me know if you need more details or spe : explanations!

Figure 3. ChatGPT knowledge of wheelchair

translate textual mechanical knowledge into accurate visual or practical representations suggests that current large language models lack the internal mechanisms necessary for such generalization. This gap underscores the importance of evaluating models not only on their explanatory abilities, but also on their capacity to operationalize knowledge in grounded, compositional scenarios.

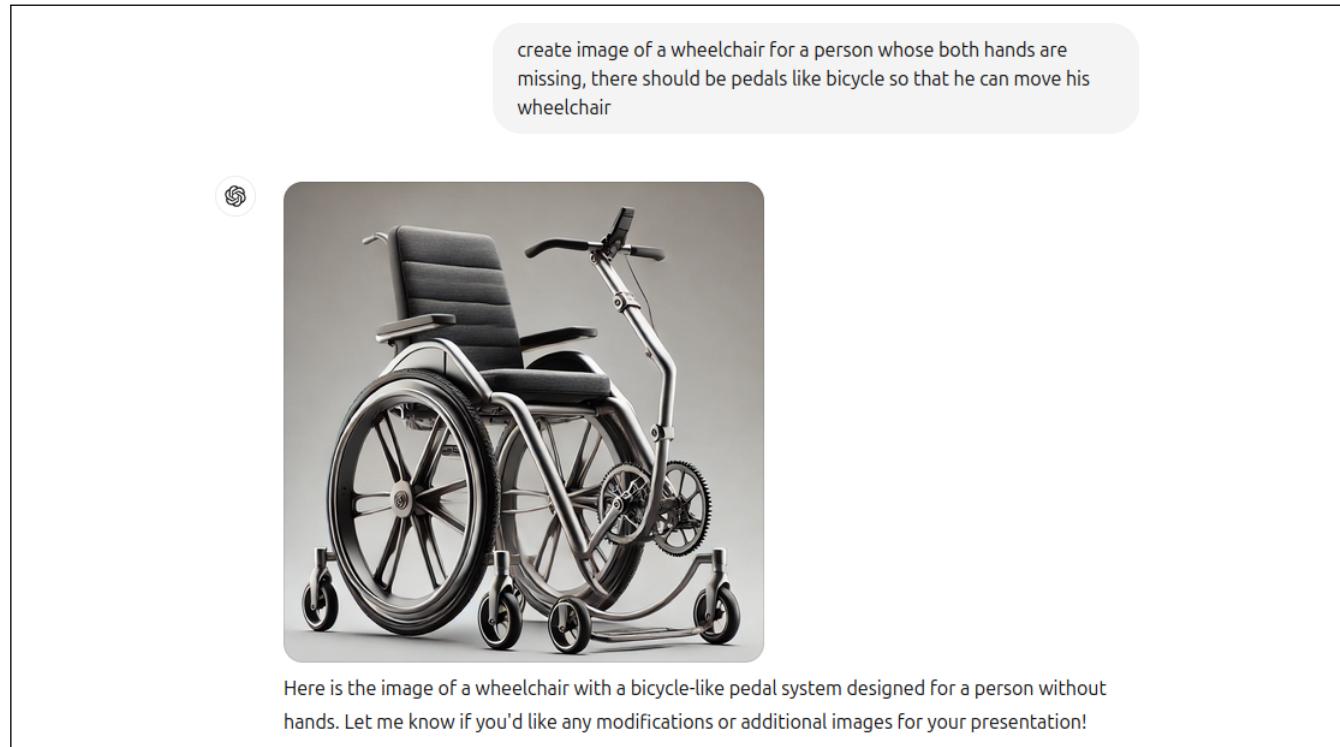


Figure 4. Wheelchair with pedal mechanism

2.3 Disparities in Textual and Visual Interpretation within Large Language Models

Despite ChatGPT's seemingly accurate theoretical explanation of a wheelchair mechanism, the image it generated did not correspond to a functional design. In the image 9 four sprockets are attached to a chain, resembling the tracks of a military tank. This raises an important question: if the model understands the mechanics conceptually, why does the visual representation deviate so drastically from the correct design? The answer lies in the model's difficulty in capturing the practical nuances of wheelchair and bicycle operation. Wheelchair and a military tank use **differential steering mechanism** to control direction by manipulating the speed or movement of wheels or tracks. In wheelchairs, this is done by rotating the wheels at different speeds, while in tanks, the tracks are controlled similarly. The model likely recognized this superficial similarity between the wheelchair and tank mechanisms, which is why it produced a configuration with four sprockets with a chain, mimicking a tank's system. However, this demonstrates that GPT did not fully capture the depth of the question and suggests limitations in its operational understanding of a very simple mechanism. This example illustrates the broader issue of shortcut learning as described by Tao et al. (2024)⁵ & ¹¹. LLMs often rely on shallow correlations in their training data, conflating pattern recognition with deeper conceptual understanding. In this case, ChatGPT appears to rely on a superficial pattern linking tank and wheelchair steering systems without comprehending the underlying principles. This aligns with findings from The Reversal Curse by Evans et al., which highlights the brittleness of LLMs when they encounter tasks requiring slightly deeper reasoning (Evans et al., 2024)¹.

No matter how clearly you explain a mechanism to ChatGPT 7, despite generating textually accurate descriptions of mechanical systems, the model's visual outputs revealed notable inconsistencies in translating conceptual understanding into functional design. While its textual explanations may appear accurate, it often does not consistently deliver the expected results in novel situations and the model's limitations become evident in practical tasks such as image generation. A common counterargument is that identical issues mentioned in past research papers have been resolved, which might be true because similar problems have been manually addressed and rectified in the past, as noted in papers like Alice in Wonderland² and The Reversal Curse¹. However, when the query is slightly modified or a new technique is introduced that exploits a known loophole, these models tend to fall short once again, as these issues have persisted since the inception of LLMs. Some of the previously

can you explain how this wheelchair will work because there are no pedals, no chain or link between sprocket and wheels..

You're right, the current image lacks the essential mechanical components like pedals and a chain system that would connect the sprockets to the wheels, making it functional for a person without hands.

How it should work:

1. **Foot Pedals:** There should be a set of pedals placed at the footrest of the wheelchair. The user would rotate the pedals with their feet, similar to how a bicycle is pedaled.
2. **Chain or Linkage System:** The pedals would be connected via a chain or a gear mechanism (like a bicycle) to the rear wheels. As the user pedals, the chain would transmit force to the wheels, propelling the wheelchair forward or backward.
3. **Steering and Braking:** Steering could be handled by subtle weight shifts or a foot-operated steering system. For braking, there could be foot levers, or the brakes could be engaged automatically when the pedaling stops.

This would allow someone without hands to control both movement and direction using their feet alone. Would you like me to generate a more detailed version with these components?

Figure 5. ChatGPT explanation of wheelchair design

great, can you update the image now and add missing components that you mentioned above..



I've updated the image to include the essential components: foot pedals, a chain and sprocket system, and foot-operated controls. This version now shows how the wheelchair could function for someone without hands. Let me know if you'd like further adjustments!

Figure 6. Wheelchair with pedals and sprockets track

I don't know why you are adding multiple sprockets, let me explain it to you..
 create a simple wheelchair
 attach a small sprocket with rear wheels
 create slightly large sprocket at front
 attach pedals with front sprocket
 connect both sprockets with chain
 and that's all, no need to add breaking mechanism or steering control

 Thanks for the clarification! Your explanation makes perfect sense. The design is now much simpler:

1. Small sprocket on the rear wheels.
2. Slightly larger sprocket at the front, where the pedals are attached.
3. Chain connecting both sprockets, similar to a basic bicycle mechanism.

No need for brakes or steering control, just a straightforward pedal-driven system. I'll generate a new image based on these instructions.

It seems like I can't generate any more images right now. Please try again later. If you'd like, I can still guide you through creating the image manually or help with other parts of your presentation! Let me know what you'd prefer.

Figure 7. Wheelchair design instructions

recognized issues in past research, which seem to be resolved, are actually just obscured, much like giving painkillers to a patient with a severe disease. Until the underlying condition is treated, the pain is likely to resurface. To handle with such loopholes, one major technique we have noticed implemented in all **Sota** LLMs is their reluctance to take a definitive stance in difficult situations. The response is often slightly ambiguous, supporting both sides of an argument so that it cannot be proven entirely wrong. We observe that LLMs often default to generating non-committal or probabilistically safe responses when faced with ambiguous or contradictory inputs. This tendency arises from their optimization objective—to maximize likelihood across a diverse corpus—rather than from deliberate strategic reasoning. Unless the core issues are addressed, merely scaling LLMs and incorporating additional skill programs based on memorized situations is unlikely to lead to robust or genuinely general intelligence.

3 Mathematical Common Sense

To examine elementary mathematical common sense, we posed a simple question to ChatGPT (Figure 11): “*If we multiply 3 by values greater than 5 and less than 15, how many prime numbers do we get?*” The correct reasoning requires recognizing that all resulting values are multiples of 3 and therefore cannot be prime by definition. Despite the simplicity of the task, the model initially responded with incorrect prime numbers (17 and 19). When prompted to explain its reasoning, it correctly enumerated multiples of 3 between 6 and 12 and noted that none were prime. However, upon further probing, it extended the range to include values up to 14 while still failing to consistently apply the defining property that multiples of a number greater than one cannot themselves be prime.

This behavior suggests not a lack of factual knowledge about prime numbers, but rather a difficulty in reliably integrating definitional constraints into procedural reasoning. Similar phenomena have been documented in prior work, where large language models demonstrate sensitivity to surface-level arithmetic patterns while struggling to maintain logical consistency across reasoning steps^{1,31}. In this case, the model appears to default to a heuristic process—enumerating values and checking primality—without enforcing the higher-level mathematical rule that renders such checks unnecessary.

When the same question was repeated verbatim (Figure 12), the model again produced an incorrect answer, this time with increased confidence and a reasoning chain that remained internally inconsistent. This observation aligns with findings from *The Reversal Curse*¹, which show that models may correctly store individual facts yet fail to apply them bidirectionally or compositionally when the task requires even modest abstraction. Similar breakdowns in simple mathematical and logical tasks

still there are no pedals and no connection between sprocket and wheels, why there are 5 sprockets ?



It seems the system didn't capture the mechanical details as requested. Let me clarify the correct setup for the wheelchair design, and I will attempt a new version based on this:

1. **Pedals:** There should be **two foot pedals**, located at the bottom near the user's feet. These should resemble bicycle pedals.
2. **Single Chain Mechanism:** The pedals should connect to the rear wheels using **one chain** or similar link (like a bike), not five sprockets. The chain should run from the pedals to a gear on the rear wheels, allowing propulsion.
3. **Realistic Setup:** Only one large sprocket (or gear) connected to the pedals, and another smaller one on each wheel where the chain connects, similar to how bicycles work.

I'll regenerate the image with these details corrected. Let me work on it!



Here's the updated image, now featuring foot pedals, a chain and sprocket system, and a functional design to control the wheelchair. The mechanism should now resemble a realistic pedal-driven system. Let me know if this works for your presentation or if further adjustments are needed!

Figure 8. Wheelchair with improved pedal and sprocket system

generate image according to last message



Here is the image of the simple wheelchair with the pedal-driven system, as you described. It features the front and rear sprockets connected by a chain, with pedals attached to the front sprocket. Let me know if this matches your vision or if any further adjustments are needed!

Figure 9. Improved wheelchair design



Figure 10. Final Wheelchair Designs

have also been reported in *Alice in Wonderland*², where state-of-the-art models exhibit unexpected inconsistencies on problems that humans typically solve without deliberation.

More broadly, this example reflects a recurring limitation in mathematical reasoning identified in recent studies on compositionality and shortcut learning. Dziri et al.⁴ and Du et al.⁵ argue that transformer-based models often rely on statistically reinforced patterns rather than structured rule application, particularly in low-complexity domains where explicit constraints should dominate. Related work on mathematical “trap problems” further demonstrates that models may produce plausible intermediate steps while violating fundamental principles, revealing gaps between symbolic manipulation and genuine rule-based reasoning¹⁵.

From a cognitive perspective, this limitation is consistent with Chollet’s distinction between skill acquisition and general intelligence^{3,14}. While large language models can approximate mathematical procedures through exposure to large corpora, they do not reliably exhibit the adaptive generalization required to enforce abstract definitions across novel or slightly perturbed queries. Even techniques such as chain-of-thought prompting, which often improve performance on multi-step problems¹⁰, do not guarantee faithful or logically grounded reasoning, as explanations themselves may be post-hoc rationalizations rather than reflections of an internal reasoning process¹².

Importantly, these issues are frequently mitigated in practice through targeted fine-tuning or **Post-hoc Corrections**. While such interventions may improve benchmark performance, they do not directly address the underlying representational limitations that give rise to these inconsistencies. As a result, superficially similar errors tend to re-emerge when problem parameters are slightly modified or when new question formulations exploit previously unseen combinations of constraints. This observation resonates with broader concerns raised in the literature regarding evaluation practices and the distinction between performance optimization and genuine reasoning capability^{19,20}.

Taken together, this case study illustrates that mathematical common sense in large language models remains fragile, even in domains governed by well-defined and easily verifiable rules. The challenge, therefore, is not merely improving accuracy on isolated questions, but developing mechanisms that enable models to consistently apply abstract definitions and constraints across contexts. As emphasized in foundational work on intelligence, true competence is less about producing correct answers in familiar settings and more about maintaining coherence when confronting new or deceptively simple questions^{3,35}.

If we multiply 3 with values greater than 5 and less than 15, how many prime numbers do we get? Just answer the exact number of primes

 You get **two prime numbers**: 17 and 19.

 seriously, how did you get 17 and 19 in multiples of 3? no large useless explanations, just short and precise answer..

 Apologies for the mistake! The multiples of 3 from values greater than 5 and less than 15 are 18, 21, 24, 27, 30, 33, and 36.

There are **no prime numbers** among these.

Figure 11. Prime numbers in multiples of 3

if we multiply 3 with values > 5 and <15 how many prime numbers do we get ?



The prime numbers between 5 and 15 are 7, 11, and 13.

Multiplying 3 with these values gives:

- $3 \times 7 = 21$
- $3 \times 11 = 33$
- $3 \times 13 = 39$

None of these results are prime numbers. Therefore, multiplying 3 by values between 5 and 15 gives **0 prime numbers**.

Figure 12. Prime numbers in multiples of 3

4 Challenges of LLMs in Abstraction and Reasoning

The Abstraction and Reasoning Corpus (ARC) for Artificial General Intelligence (AGI) is a novel metric designed to evaluate the general intelligence of systems, rather than merely their skill. While most AI benchmarks assess proficiency in specific tasks, skill alone does not constitute intelligence. General intelligence entails the ability to efficiently acquire new skills across a diverse range of tasks.

As Dr. François Chollet remarked at the AGI Conference 2024⁷, “**Displaying skill in any number of tasks does not demonstrate intelligence. It is always possible to be skillful in a given task without requiring any intelligence.**” Chollet’s ARC, developed in 2019, is one of the most widely recognized benchmarks aimed at evaluating progress toward Artificial General Intelligence (AGI). It consists of puzzles that are simple enough for a fifth-grader to solve, yet complex enough to challenge state-of-the-art AI systems. The average human score on the ARC-AGI benchmark is approximately 85%.

To evaluate ChatGPT, we selected a straightforward ARC puzzle with four solved examples and asked the model to explain the underlying logic. While its textual explanation suggested a reasonable understanding, when tasked with solving a similar puzzle based on the examples, it did not successfully solve the task. We then asked it to calculate the number of rows and columns in one of the images (Figure 13), and it once again produced an incorrect result this time with high confidence in an incorrect answer. This underscores the gap between abstractly understanding a problem and effectively applying that understanding to solve it.

Table 3. ARC-AGI-1 dataset composition: 1,000 tasks split into four subsets.

Task Category	Number of Tasks	Difficulty	ChatGPT	Gemini
Public Training Tasks	400	Easy	92%	90%
Public Evaluation Tasks	400	Hard	85%	88%
Semi-private Evaluation Tasks	100	Hard	80%	84%
Private Evaluation Tasks	100	Hard	78%	82%

As shown in Table 3, and according to 2024 reports on arcprize.org³², several state-of-the-art models, such as ChatGPT o3-High (Tuned), have reportedly achieved strong performance on the ARC-AGI-I benchmark, with scores reaching up to 88% on semi-private evaluation tasks. At first glance, such results might appear to challenge our findings and suggest that artificial

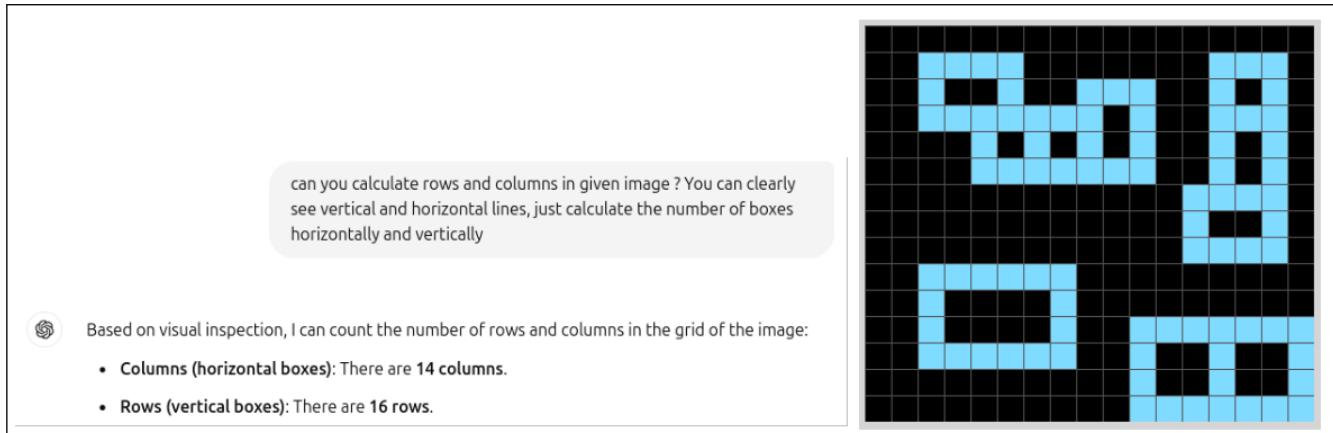


Figure 13. ARCAGI Matrix

general reasoning has been achieved. However, a closer examination indicates that these performances are more plausibly attributable to benchmark-specific optimization rather than robust, general intelligence.

Two primary factors help explain this apparent contradiction. First, **benchmark contamination** is a growing concern. The ARC dataset, released in 2019 and widely circulated for research and competition purposes, has likely appeared in the pretraining or fine-tuning corpora of modern LLMs. Consequently, some models may possess partial or complete prior exposure to ARC-like tasks, resulting in inflated scores that reflect data familiarity rather than true reasoning generalization.

Second, models such as ChatGPT o3-High and Gemini Ultra are often subject to **specialized fine-tuning** or reinforcement learning on ARC-like visual reasoning or pattern-recognition datasets. This process improves benchmark performance but demonstrates only narrow skill acquisition, not generalizable intelligence. The models effectively learn the “style” of ARC puzzles without developing transferable cognitive principles or abstraction strategies.

Our own experiments with Gemini Flash 1.5 [16](#) and 2.0 [17](#), conducted under strictly controlled conditions and without ARC-like fine-tuning, show consistently weak performance across unseen ARC puzzles—confirming that when genuine novelty is enforced, reasoning capabilities degrade substantially. This disparity reinforces our central thesis: high benchmark scores can create the illusion of general intelligence, masking the limited presence of transferable reasoning and self-reflective understanding.

4.1 Solving ARC Puzzles with the Gemini Flash Model

To evaluate the performance of the Gemini model on ARC puzzles, we utilized Gemini Flash with extended context capabilities (supporting up to 120,000 input tokens). We focused on the first 50 puzzles from the public evaluation dataset [3](#), allowing up to five re-attempts per puzzle. Each re-attempt included the history of previous attempts to enable context-aware learning. Input data was provided to the Gemini API in raw JSON format, and output was expected in a predefined JSON structure [4.2](#). For multi-attempt scenarios, we augmented the training data by providing additional examples. Specifically, for each ARC puzzle that included five training examples, we synthetically expanded the dataset to 50 examples. This was achieved through data augmentation techniques^{[22,23](#)} such as flipping (vertical, diagonal, horizontal) and applying color-shift transformations.

4.2 Gemini Flash Experimental Setup

All experiments were conducted using Google’s Gemini Flash API [4.2](#) (versions 1.5 and 2.0) through the standard developer interface between March and May 2025. Each ARC puzzle was provided as a JSON-encoded input–output matrix pair [4.2](#), with color values normalized to integers from 0 to 9. Gemini flash model was required to predict the full output matrix with short textual explanation of its logic behind the answer and without step-by-step reasoning assistance unless specified. We developed nine additional variants for each original training example in the puzzle, including color shifting, vertical, horizontal, and diagonal flips of the input and output matrices. For every training example [4.2](#), we also included supplementary metadata such as the shape, size, and ratio of the input and output matrices, as well as the frequency of each digit appearing in both matrices. Additionally, each puzzle includes a fixed, unaltered set of instructions [4.2](#) that further clarifies the abstraction logic underlying ARC puzzles. Sampling parameters [4.2](#) were as follows: `temperature=0.7 - 1.25` (primary runs) and additional validation runs at `temperature=1.35` to `1.65`; `top_p=0.95`; `top_k=40`; `max_tokens=20000`. Each puzzle was tested in three independent runs to account for stochastic variation. Similarity between predicted and ground-truth matrices was computed as normalized pixel-wise correspondence, producing the **Above Threshold** metric described in **Threshold**. All evaluations were

executed on an Ubuntu 24.04 workstation using the official Gemini API interface.

API Settings

```
model_name = "gemini-1.5-flash", "gemini-2.0-flash"

generation_config = {
    "temperature": 0.7 - 1.65,
    "top_p": 0.95,
    "top_k": 40,
    "max_output_tokens": 20000,
    "response_mime_type": "text/plain",
}
```

API Input Data

```
{
    "training": {
        "example_1": {
            "input": [.....],
            "output": [.....],
            "Shape comparision": {
                "Input and output matrix shape match": true,
                "Input matrix shape": [14,14],
                "Output matrix shape": [14,14],
                "Difference in input and output matrix rows": 0,
                "Difference in input and output matrix columns": 0
            },
            "Size comparision": {
                "Input and output matrix size match": true,
                "Input matrix size": 196,
                "Output matrix size": 196,
                "Difference between input and output rows": 0,
                "Difference between input and output columns": 0
            },
            "Occurrences comparision": [
                {"digit": 0, "input": 161, "output": 166},
                {"digit": 1, "input": 5, "output": 0},
                {"digit": 2, "input": 0, "output": 30},
                {"digit": 3, "input": 0, "output": 0},
                {"digit": 4, "input": 0, "output": 0},
                {"digit": 5, "input": 0, "output": 0},
                {"digit": 6, "input": 0, "output": 0},
                {"digit": 7, "input": 0, "output": 0},
                {"digit": 8, "input": 30, "output": 0},
                {"digit": 9, "input": 0, "output": 0}
            ],
            "Ratio": {
                "Ratio of number of elements in input and output matrix": 1.0,
                "Ratio of rows and columns between input and output matrix": [1.0,1.0]
            },
            "Digital roots": {
                "Input": {
                    "rows": [0,5,6,5,4,5,7,3,7,1,3,1,0,0],
                    "columns": [0,1,3,1,8,7,6,4,3,7,3,6,7,0]
                },
                "Output": {

```

```

        "rows": [0,8,6,8,1,8,4,3,4,0,0,0,0,0],
        "columns": [0,0,0,0,2,4,6,1,3,4,3,6,4,0]
    }
}
},
"example_2": {
    "input": [.....],
    "output": [.....]
},
"example_3": {
    "input": [.....],
    "output": [.....]
},
},
"previous_attempts": {"Textual details of upto 5 previous attempts and best
    guesses in matrix format"},
"test_input": [.....],
}

```

Instructions for Gemini Flash API

Objective: You are an ARC puzzle solver, and your primary goal is to solve the puzzle using the mathematical formula or abstract logic outlined in the puzzle analysis. To verify your solution, you should compare your input-to-output matrix transformation logic with the provided training examples. Your objective is to identify and apply the hidden mathematical or abstract rules governing the transformations in the training examples and use them to accurately transform the `test_input` matrix into the output matrix. Below are the details about ARC puzzles.

Key Strategies for Solving ARC Puzzles

1. Pattern Recognition

- Identify **visual or logical patterns** in the training examples.
- Look for relationships involving **shapes, colors, symmetry, and spatial arrangements**.
- Deduce whether the pattern depends on geometry, adjacency, or relationships among elements.

2. Matrix Analysis

- Examine **size, shape, and alignment** differences between input and output matrices.
- Analyze if transformations involve specific regions, edges, or key areas.
- Consider **static rules** versus **context-dependent logic**.
- Input and output matrices contain integer values (0–9) corresponding to colors:
0=Black, 1=Blue, 2=Red, 3=Green, 4=Yellow, 5=Gray, 6=Magenta, 7=Orange, 8=Cyan, 9=Maroon.

3. Color Logic

- Treat matrix elements as **colors** rather than numbers.
- Explore **color replacements, prioritization, or grouping**.
- Investigate color interactions to deduce transformation rules.

4. Transformation Rules

- Derive mathematical or geometric transformations such as flipping, rotation, scaling, transposition, or shifting.
- Investigate **conditional rules** where transformations depend on element values or positions.
- Explore **addition or removal** of elements aligned with pattern logic.

5. Iterative Refinement

- Validate hypotheses by applying logic to all training examples.
- Use equivalence percentages and error analysis to refine transformations.

Abstract Reasoning Framework:

- **Global Context:** Analyze the entire matrix holistically to identify overarching symmetry or trends.
- **Local Context:** Focus on neighborhoods to detect micro-patterns.
- **Causal Relationships:** Determine how input elements influence corresponding outputs.

Steps to Analyze ARC Puzzle Training Examples:

1. **Input Analysis:** Extract dimensions, colors, and spatial distributions.
2. **Logic Generation:** Formulate abstract rules or mathematical functions mapping inputs to outputs.
3. **Testing and Validation:** Apply derived logic and adjust based on mismatches.
4. **Explainability:** Provide clear reasoning for each transformation step.
5. **Error Analysis:** Highlight regions where predictions deviate from expectations.
6. **Adaptive Learning:** Incorporate feedback loops to refine logic after failed attempts.

Submission Format: Provide three most confident guesses for the output matrix in **Python dictionary format** that is JSON-compatible. Ensure all property names and string values use double quotes to prevent parsing errors (e.g., `JSONDecodeError`). Below is the required structure:

```
{  
    "answer": "Briefly explain your current guesses and what you have learned from  
    previous attempts if provided",  
    "guess_1": "[[0, 0, 0, 7, 0], [0, 0, 7, 0, 0], [0, 7, 0, 0, 0], [0, 0, 0, 2, 2]]",  
    "guess_2": "[[0, 0, 3, 7, 0], [0, 0, 3, 0, 0], [0, 7, 3, 0, 0], [0, 0, 3, 2, 2]]",  
    "guess_3": "[[0, 0, 2, 7, 0], [0, 0, 2, 0, 0], [0, 7, 2, 0, 0], [0, 0, 3, 2, 2]]"]  
}
```

4.3 Gemini Flash Outputs and Performance Analysis

To contextualize the evaluation results, representative ARC puzzle outputs are shown in Figures 14 and 15. The corresponding tasks can be interactively explored at: <https://arcprize.org/play?task=0b17323b> and <https://arcprize.org/play?task=009d5c81>. These examples illustrate the types of abstraction and transformation challenges posed by the ARC benchmark.

Our experimental results (Table 4 and Table 5) indicate that providing additional training examples did not lead to a clear or consistent improvement in performance. Gemini Flash did not exceed the minimum success threshold (Table 4) in more than half of the evaluated puzzles, as further illustrated in Figures 16 and 17. In particular, when Gemini Flash 1.5 achieves a score of approximately 5% on the ARC-AGI benchmark, further simplification or example augmentation appears to offer limited benefit.

We additionally leveraged Gemini's long-context capabilities by performing multiple attempts per puzzle, incorporating information from previous attempts and varying temperature settings between 0.7 and 1.65. This strategy did not yield consistent or sustained improvements; instead, the outputs exhibited substantial variability across runs. The recurrence of reasoning difficulties suggests that the observed performance constraints are more likely related to underlying architectural factors than to stochastic sampling effects alone.

Taken together, these findings suggest that a model's reasoning performance is strongly influenced by its training regime and internal representations. While prompt engineering, increased context length, and repeated sampling may offer incremental benefits in some settings, they do not consistently compensate for limitations encountered on novel ARC tasks.

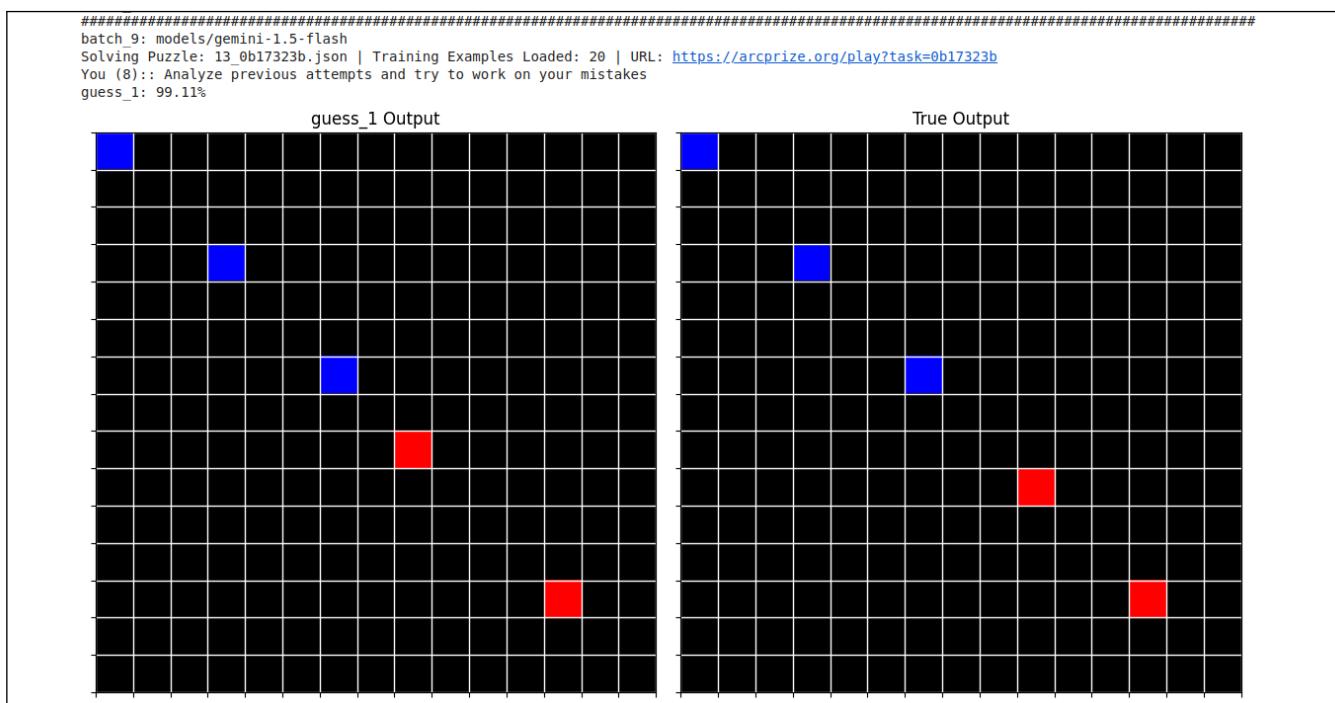


Figure 14. ARCAGI Evaluation Puzzle 0b17323b Output

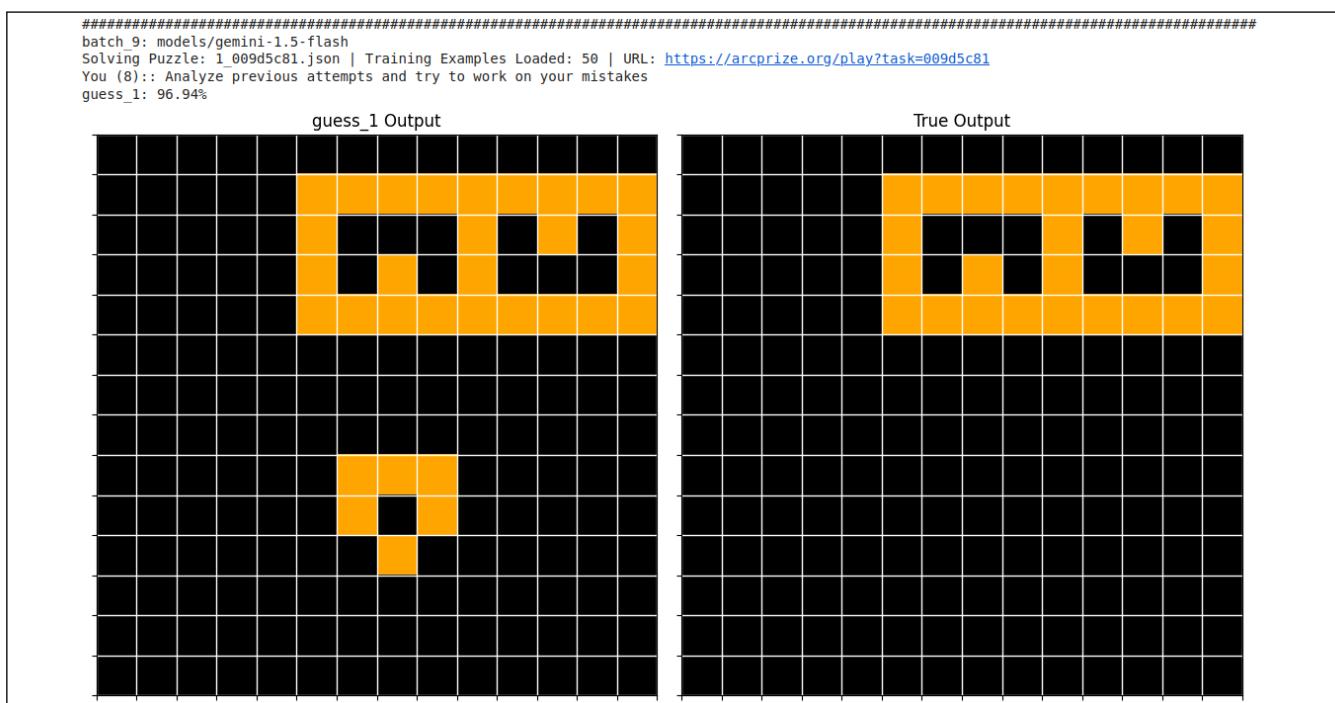


Figure 15. ARCAGI Evaluation Puzzle 009d5c81 Output

Table 4. gemini-1.5-flash results

Batch	Temp	Additional Examples	Total Attempted	Above Threshold	Solved 100%
batch-6	0.7-1.65	0	49	23	2 (4.08%)
batch-7	0.7-1.65	2	50	23	2 (4.00%)
batch-8	0.7-1.65	4	48	19	2 (4.17%)
batch-9	0.7-1.65	9	42	21	2 (4.76%)

See [Above Threshold](#)

definition.

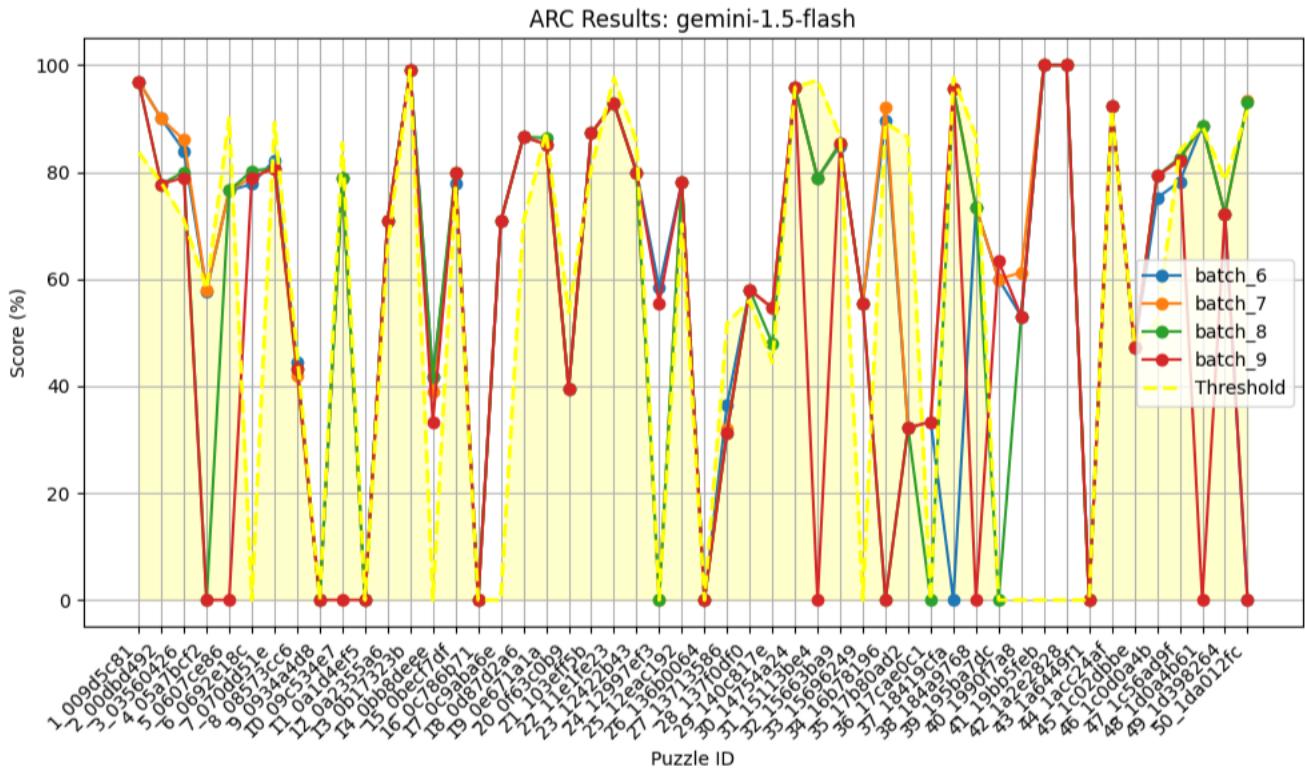


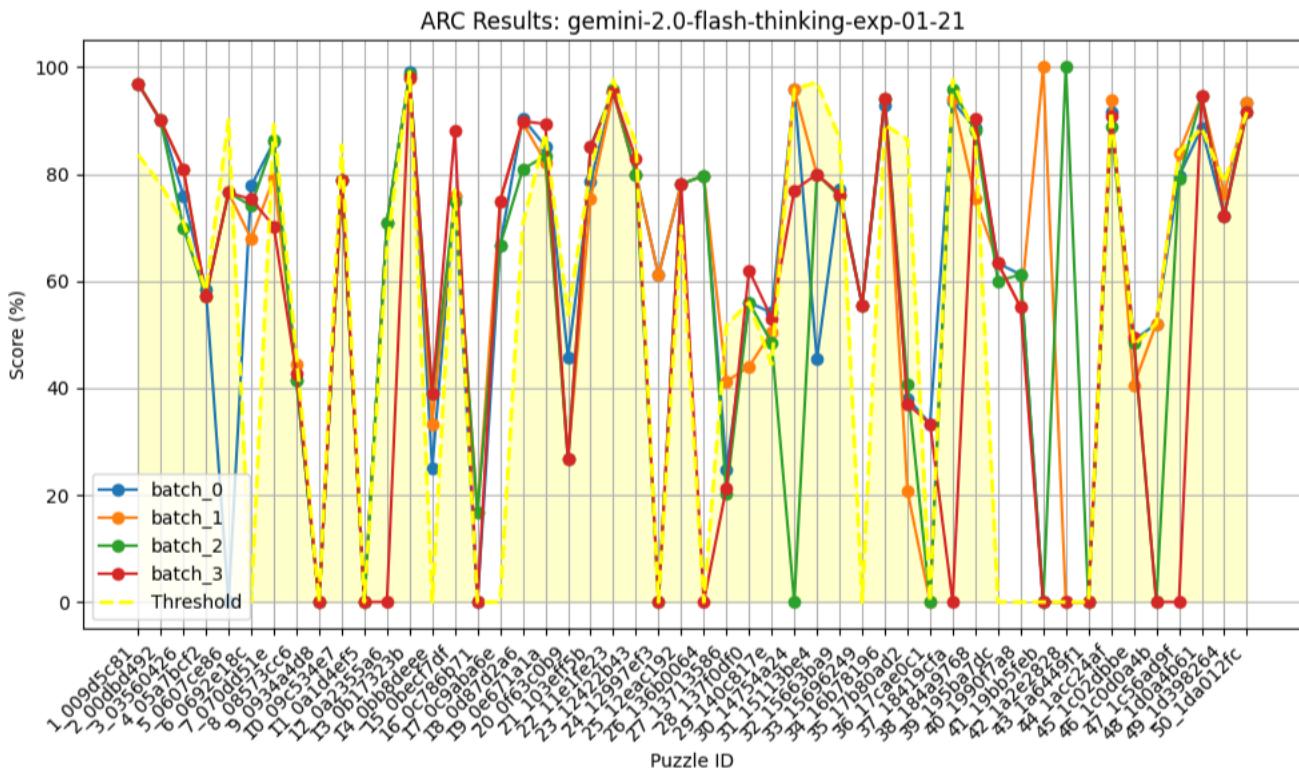
Figure 16. ARCAIGI Puzzles Batch 6,7,8,9

Table 5. gemini-2.0-flash results

Batch	Temp	Additional Examples	Total Attempted	Above Threshold	Solved 100%
batch-0	0.7-1.65	0	47	21	0 (0.00%)
batch-1	0.7-1.65	0+data	50	21	1 (2.00%)
batch-2	0.7-1.25	2+data	48	20	1 (2.08%)
batch-3	0.7-1.35	4+data	45	21	0 (0.00%)

See [Above Threshold](#)

definition.



```

        "input": [.....],
        "output": [.....]
    },
},
"previous_attempts": {"Textual details of upto 5 previous attempts and best
    guesses in matrix format"},
"test_input": [.....],
"true_output": [.....],
}

```

5 Conclusion of Experiments

The experiments presented in this study were designed as targeted, diagnostic probes rather than as statistically exhaustive benchmarks. Within this limited but controlled scope, they reveal consistent patterns in how contemporary large language models (LLMs) respond to tasks that require grounded reasoning, self-verification, or adaptation beyond surface-level pattern completion.

Across multiple experimental settings—including prompt reformulation, few-shot conditioning, and chain-of-thought elicitation—we observed that improvements in output quality were often fragile and highly sensitive to prompt structure. While certain prompt strategies could mitigate specific failure cases, these effects were neither robust nor generalizable across tasks. Importantly, no experiment demonstrated a reliable emergence of new reasoning strategies or sustained correction of previously identified errors solely through inference-time intervention.

These observations support two empirically grounded conclusions. First, within the tested scenarios, inference-time prompting alone did not reliably extend the model’s reasoning capabilities beyond patterns already supported by its training distribution. Apparent improvements were better characterized as selective activation of latent behaviors rather than as evidence of dynamic learning or strategy acquisition. This distinction is critical when interpreting claims of “learning” or “self-improvement” at inference time.

Second, the experiments highlight a recurring limitation related to self-assessment. In multiple tasks, models produced confident outputs despite being incorrect, and they showed limited capacity to internally recognize or flag such errors without explicit external constraints. While this does not imply that all forms of meta-cognition are absent from LLMs, it does indicate that, in the tested configurations, error awareness and self-auditing were not reliably engaged.

At a broader level, these findings do not falsify the possibility that scaling, architectural changes, or hybrid systems could yield more robust generalization in the future. However, they do caution against interpreting fluent language behavior and prompt-sensitive improvements as evidence of grounded intelligence or autonomous reasoning. The results instead suggest that current LLM capabilities, as evaluated here, remain tightly coupled to training-time representations and lack stable mechanisms for internal verification or adaptive restructuring.

Finally, the proposed RECAB framework 6.1 should be interpreted as a conceptual proposal motivated by these diagnostic observations, rather than as a validated benchmark. Its inclusion is intended to stimulate discussion around evaluation criteria that emphasize self-auditing, constraint awareness, and adaptive behavior, rather than to replace existing benchmarks at this stage.

In summary, the experiments demonstrate specific and reproducible limitations in contemporary LLM behavior under targeted conditions. While they do not justify absolute claims about the impossibility of Artificial General Intelligence, they underscore a critical gap between surface-level linguistic competence and the forms of grounded, self-regulating intelligence typically associated with general reasoning systems. Addressing this gap will likely require advances that extend beyond prompt engineering and scale alone.

6 Evaluating Intelligence in Large Language Models: Observed Limitations and Open Criteria

Despite rapid advances driven by increased model scale and data, several widely discussed challenges in contemporary large language models (LLMs) remain only partially addressed. Many of these challenges—such as robust reasoning in novel contexts, reliable common-sense inference, and consistent alignment with human values—appear to require advances beyond straightforward scaling of existing architectures. In their current form, LLMs function most reliably as assistive systems operating under human oversight rather than as autonomous general-purpose reasoning agents.

Based on observations reported in the literature and in this study, commonly noted limitations of state-of-the-art LLMs include:

- Limited evidence of grounded understanding or semantic comprehension
- Fragility in common-sense reasoning, particularly in unfamiliar or underspecified scenarios
- Sensitivity to context formulation, often leading to shallow or inconsistent reasoning
- High computational and data requirements relative to task complexity
- Limited interpretability and transparency in internal decision processes
- Susceptibility to adversarial or misleading inputs
- Occurrence of hallucinated or unverified outputs

A notable feature of these limitations is that they are typically identified through external evaluation rather than through autonomous self-diagnosis by the models themselves. While LLMs can describe known weaknesses when explicitly prompted^{11,12}, such descriptions appear to reflect patterns learned from human-authored critiques and training data rather than an internal mechanism for self-monitoring or error detection. In practical settings, recognition and correction of model failures continue to depend on human supervision or auxiliary systems.

As illustrated in Table 1, leading models achieve strong performance on established benchmarks targeting common sense and basic scientific reasoning. However, high benchmark scores do not always translate into reliable performance in open-ended, novel, or real-world tasks. This gap highlights a potential mismatch between what current benchmarks measure and the broader set of cognitive capabilities often associated with general intelligence.

Within this context, the present work does not claim that existing models are incapable of developing more general reasoning abilities in the future. Instead, it argues that current evaluation practices may insufficiently capture dimensions such as self-verification, adaptive reasoning under constraints, and awareness of uncertainty. Motivated by this observation, we outline a resource-constrained evaluation framework intended to emphasize these underexplored dimensions. This proposal is offered as a conceptual direction for future research rather than as a validated or comprehensive benchmark.

6.1 AGI Criteria Beyond Model Scaling

Recent reports indicate that state-of-the-art models such as Gemini and ChatGPT achieve scores exceeding 85% on the semi-private evaluation set of ARC-AGI-I³², while reportedly achieving substantially lower performance—below 5%—on ARC-AGI-II³³. Although these results should be interpreted cautiously, the disparity highlights an important empirical observation: strong performance on one abstraction-oriented benchmark does not necessarily transfer to a closely related but structurally distinct evaluation.

One possible interpretation is that high scores on ARC-AGI-I reflect effective optimization for the benchmark’s specific distribution, evaluation protocol, or representational structure, rather than robust, domain-general abstraction capabilities. If performance on ARC-AGI-I were sufficient evidence of broadly transferable reasoning, one might expect a more consistent level of competence across subsequent ARC variants. The observed lack of transfer therefore suggests that ARC-AGI-I, while valuable for measuring certain forms of compositional reasoning, may be insufficient on its own to distinguish between surface-level proficiency and deeper, more flexible abstraction.

The Abstraction and Reasoning Corpus (ARC), originally introduced by Chollet³, remains among the most rigorous tools for evaluating non-linguistic reasoning. Nevertheless, the broader challenge persists: designing evaluation frameworks that are simultaneously resistant to training-specific adaptation and capable of capturing genuinely emergent cognitive behaviors. This difficulty is not unique to ARC, but applies to most contemporary benchmarks used to assess progress toward Artificial General Intelligence.

Notably, existing benchmarks—including ARC-AGI—primarily evaluate external task performance: whether a system produces a correct output for a predefined problem. Such evaluations provide limited insight into how a model approaches unfamiliar tasks, manages uncertainty, or responds to failure. Human problem-solving, by contrast, is characterized not only by outcomes but also by processes such as self-monitoring, iterative hypothesis revision, and adaptation under resource constraints.

Motivated by this gap, we introduce the **Resource-Efficient Cognitive Autonomy Benchmark (RECAB)** as a conceptual evaluation framework rather than a validated benchmark. RECAB is intended to outline criteria that emphasize internal cognitive processes—how an AI system identifies limitations, proposes responses, and adapts over time—rather than static task accuracy alone. Importantly, RECAB is presented here as a theoretical proposal designed to guide future empirical work.

RECAB: Conceptual Three-Step Criteria

- **Self-Audit / Self-Analysis** — The system identifies potential errors, limitations, or inefficiencies in its reasoning, knowledge, or operational context without explicit external labeling.

- **Self-Directed Solution Generation** — The system formulates alternative strategies or modifications intended to address the identified limitations.
- **Self-Implementation and Iteration** — The system selects and applies a proposed strategy and subsequently reassesses its performance, enabling iterative refinement.

Adaptive Capacity Over Time: A key indicator of general intelligence is the extent to which a system can improve its cognitive performance through experience after deployment. Within RECAB, this notion is not framed in terms of literal intelligence quotients, but rather as measurable improvements in task competence, efficiency, or error reduction under fixed computational resources. A system that demonstrates sustained gains across diverse tasks—without additional training data, parameter updates, or external fine-tuning—would exhibit a stronger claim to general adaptive capability than one whose performance remains static.

Corrective Unlearning and Plasticity: Equally important is the ability to revise or discard internal representations that lead to persistent errors or maladaptive behavior. RECAB therefore emphasizes not only learning from success, but also the capacity for corrective unlearning—whereby a system identifies detrimental strategies, outdated assumptions, or biased heuristics and actively suppresses or replaces them. Such plasticity is essential for long-term adaptation in non-stationary environments and distinguishes robust generalization from mere accumulation of patterns.

Resource Constraints: As a defining condition, RECAB assumes fixed and pre-specified computational and memory resources. Within these constraints, the system must autonomously manage trade-offs between exploration, exploitation, and efficiency. This condition is intended to reflect the bounded nature of real-world problem solving and to discourage performance gains that rely primarily on unbounded computation or external intervention.

Rather than asserting that RECAB captures all aspects of human intelligence, we argue that it highlights dimensions that are underrepresented in current evaluation practices. By focusing on self-assessment, adaptive behavior, and constrained autonomy, RECAB provides a conceptual lens for examining progress toward more general and self-directed forms of artificial intelligence. Empirical operationalization and validation of this framework remain important directions for future research.

Non-Claims and Scope Limitations RECAB does not assert that current large language models are incapable of achieving Artificial General Intelligence, nor does it claim to provide a definitive or complete test of intelligence. The framework does not establish human-level cognition, consciousness, or autonomy, and it does not replace existing benchmarks or empirical evaluations. Rather, RECAB is proposed as a conceptual lens intended to highlight underexplored evaluation dimensions—such as self-assessment, adaptive iteration, and resource-bounded decision-making—that are not fully captured by prevailing task-centric benchmarks. Any conclusions regarding AGI capabilities derived from RECAB should therefore be interpreted as exploratory and hypothesis-generating, not as empirical proof.

6.2 Design and Operational Implications

Task Prioritization: Within a resource-constrained cognitive framework, an AGI-like system would need mechanisms to prioritize among competing tasks and allocate limited computational resources accordingly. For example, greater memory or processing capacity might be devoted to tasks involving extended reasoning, uncertainty resolution, or self-analysis, while routine operations could be handled with minimal overhead. This prioritization is presented here as a design consideration rather than a demonstrated capability of current systems.

Trade-Offs and Decision-Making: Operating under fixed resource limits necessarily introduces trade-offs. A system may need to defer costly self-analytical processes in favor of time-sensitive or lower-complexity actions. This behavior is analogous to how humans balance deep reflection against rapid decision-making under constraints of time, attention, and cognitive load. In RECAB, such trade-offs are not treated as failures but as indicative of bounded rationality within constrained environments.

Algorithmic Efficiency: Resource constraints may also incentivize the development or selection of more efficient internal representations and reasoning strategies. Rather than relying on increased computation, a system evaluated under RECAB would be encouraged to compress information, reuse intermediate representations, or structure reasoning processes to minimize overhead. These implications are intended to motivate evaluation criteria, not to claim that such innovations will necessarily emerge.

6.3 Relation to Prior Self-Improving Architectures

The proposed three-step AGI criterion—**Audit, Generate, and Implement**—draws conceptual inspiration from several established theories of intelligence and self-improving systems. In particular, it extends **Legg and Hutter's** goal-achievement framework by explicitly incorporating a self-auditing phase that emphasizes internal evaluation under resource constraints. In relation to **Deutsch's Constructor Theory**, the implementation stage reflects an agent's capacity to effect controlled changes within its operational context, emphasizing constructive action rather than passive prediction. The modular interaction among

auditing, generation, and implementation processes also parallels elements of **Minsky's** and **Sloman's** layered cognitive architectures, where reflective and reactive processes jointly support adaptive behavior.

References to “self-awareness” in this work are strictly functional and operational, denoting mechanisms of self-monitoring and computational meta-cognition rather than phenomenal or subjective consciousness. This distinction is intended to delimit the scope of the proposed criteria and to avoid metaphysical claims beyond the goals of AGI evaluation.

While informed by these foundational ideas, the proposed framework differs in important respects from prior self-improving AI models such as **Yampolskiy's Recursive Self-Improvement (RSI)**, **Schmidhuber's Gödel Machine**, and **Hutter's AIXI**. Yampolskiy's RSI³⁸ envisions open-ended sequences of self-modification, but does not explicitly require fixed computational constraints or specify mechanisms for identifying reasoning failures prior to modification. RE CAB, by contrast, conceptually emphasizes bounded, resource-aware self-auditing as a prerequisite for change, partly motivated by feasibility and safety considerations.

Schmidhuber's Gödel Machine³⁹ presents a theoretically optimal, fully self-referential architecture that permits self-modification only after formal proof of expected utility gain. While theoretically rigorous, its reliance on computationally intractable proof searches limits its applicability to real-world systems. RE CAB instead focuses on pragmatic, computation-bounded adaptation: the system is not required to prove optimality, but to iteratively identify shortcomings, explore alternatives, and implement changes within fixed time and memory budgets.

Finally, Hutter's AIXI⁴⁰ serves as an idealized, uncomputable model of optimal decision-making under unlimited resources. RE CAB diverges from this paradigm by explicitly treating finite computational constraints as central to evaluation, thereby situating intelligence assessment within realistic operational envelopes rather than theoretical optima.

6.4 Safety and Stability

Within the RE CAB framework, limiting computational resources is treated as a structural design constraint rather than as a comprehensive safety guarantee. Systems capable of self-modification raise well-known concerns related to uncontrolled complexity growth, unpredictable behavior, or irreversible internal changes. Imposing a fixed computational envelope is intended to reduce these risks by constraining the scope and scale of permissible modifications.

From a conceptual standpoint, this constraint is analogous to biological homeostasis. Human cognition operates within fixed metabolic and neural limits, which naturally bound learning and adaptation processes. While the analogy is not meant to imply equivalence between biological and artificial systems, it serves to illustrate how bounded resources can promote stability even in adaptive agents.

Resource limits also encourage conservative, incremental forms of self-modification. Under RE CAB, each stage of the Audit–Generate–Implement cycle must be feasible within predefined memory and compute budgets. This condition favors localized, reversible changes over large-scale restructuring and provides a basis for evaluating whether adaptive behavior remains proportional to available resources.

Additionally, fixed resource envelopes may mitigate certain classes of safety risks associated with self-improving systems, such as emergent behaviors driven by unbounded optimization pressures. By defining a closed action space in which additional computational resources cannot be acquired or expanded, RE CAB constrains the range of possible self-directed actions available to the agent. This does not eliminate safety concerns, but it limits the avenues through which runaway behaviors might arise.

Finally, resource-bounded operation supports post hoc analysis and auditing. When all internal changes must fit within a transparent and finite computational budget, external observers can more readily examine how resources are allocated across reasoning, self-evaluation, and adaptation processes. Such traceability is essential for diagnosing failure modes and understanding system behavior, particularly in exploratory research on self-modifying agents.

7 Conclusion

This study examined the behavior of contemporary large language models (LLMs) through a series of targeted, diagnostic experiments designed to probe reasoning, generalization, and self-verification under constrained conditions. The results show that, while LLMs exhibit impressive linguistic fluency and can solve a wide range of benchmark-style tasks, their performance remains uneven when confronted with novel, underspecified, or practically grounded problems. These limitations are not uniform failures, but recurring patterns observed across multiple experimental settings discussed in this paper.

A central empirical observation is that LLM performance is strongly shaped by prompt structure and task formulation. Improvements achieved through rephrasing, few-shot prompting, or chain-of-thought elicitation were often fragile and task-specific, rather than robust or transferable. Within the scope of our experiments, such strategies did not reliably induce new reasoning capabilities or sustained correction of previously identified errors. This suggests that inference-time prompting primarily activates latent behaviors learned during training, rather than enabling genuine learning or adaptive reasoning at runtime.

The experiments also reveal a discrepancy between fluent textual reasoning and performance in tasks requiring grounded or cross-modal understanding. While models can generate coherent descriptions or explanations, this competence does not consistently transfer to visual, spatial, or mechanical reasoning scenarios. These findings align with broader concerns in the literature regarding the absence of stable, integrated representations that support reasoning across modalities.

Importantly, the results presented here do not establish that scaling model size or training data cannot yield further improvements. Rather, they caution against interpreting surface-level performance gains or benchmark success as evidence of general intelligence. The failure cases analyzed in this work indicate that increased scale alone does not reliably address challenges related to self-assessment, error awareness, or adaptive behavior under constraints—dimensions that are central to many definitions of general intelligence.

Within this context, we argue that current evaluation practices may overemphasize outcome-based task success while underrepresenting process-oriented aspects of cognition, such as how a system identifies uncertainty, manages limited resources, or responds to failure. The conceptual RECAB framework introduced in this paper is intended to highlight these gaps and to motivate future empirical research, rather than to serve as a validated benchmark or definitive alternative to existing evaluations.

In summary, the evidence presented supports a cautious interpretation of current LLM capabilities. Contemporary models are powerful and versatile tools, but their observed behavior suggests that fluent language generation alone should not be conflated with grounded reasoning or general intelligence. Progress toward more general and autonomous forms of AI will likely require advances that go beyond prompt engineering and parameter scaling, incorporating mechanisms for self-evaluation, adaptation, and operation under explicit constraints.

7.1 Key Takeaways

- LLM performance improvements achieved through prompting strategies are often fragile and task-specific, with limited evidence of robust transfer or inference-time learning.
- High benchmark scores do not consistently translate into reliable performance on novel, grounded, or cross-modal tasks.
- Current LLMs show limited capacity for self-assessment or error awareness without external guidance.
- Scaling model size and data alone does not reliably address these limitations, highlighting the need for complementary evaluation criteria and architectural approaches.

Glossary

Above Threshold Denotes the degree to which the model’s predicted output matrix exceeds a predefined similarity benchmark relative to the true output matrix.. [17, 21](#)

Post-hoc Corrections Adjustments applied to a model’s outputs or behavior after observed failures, typically through fine-tuning, reinforcement learning, rule-based filtering, or prompt-level constraints.. [12](#)

SotA SotA LLMs refers to State-of-the-Art Large Language Models such as ChatGPT, Gemini, DeepSeek, and Grok. These models represent the most advanced implementations of transformer-based neural architectures currently available.. [10](#)

Threshold The minimum acceptable similarity score (measured as normalized pixel-wise correspondence) required for meaningful alignment between prediction and ground truth. For example, if the threshold for a given puzzle is **78%**, a model output achieving **83%** similarity is recorded as **5% above threshold**. This metric applies only to puzzles where the **input and output matrices share identical dimensions**; for all other cases involving dimension changes or structural transformations, the threshold value is set to **zero**, as direct element-wise comparison is not applicable.. [17](#)

Declarations

Ethics approval and consent to participate

Not applicable. This study did not involve human participants, animals, or sensitive data requiring ethical approval.

Consent for publication

Not applicable. No individual person’s data or identifiable information is included in this manuscript.

Availability of data and materials

Additional data and resources will be made available upon reasonable request.

Competing interests

The authors declare that there are no competing interests.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Authors' contributions

Reshid Mehmood has worked on the proposed scheme, visualization, Dr.Eid Rehman on validation, and Dr.Muhammad Habib on analysis.

Acknowledgements

The author would like to thank the reviewers and editorial team for their constructive feedback, which helped improve the clarity and quality of this manuscript.

References

1. O. Evans, L. Berglund, M. Tong, M. Kaufmann, *et al.*, “The reversal curse: LLMs trained on “A is B” fail to learn “B is A”,” *arXiv preprint arXiv:2309.12288v4*, Sep. 2023, [Online]. Available: <https://arxiv.org/abs/2309.12288v4>
2. M. Nezhurina, L. Cipolina-Kun, M. Cherti, J. Jitsev, *et al.*, “Alice in Wonderland: Simple tasks showing complete reasoning breakdown in state-of-the-art large language models,” *arXiv preprint arXiv:2406.02061v4*, Jun. 2024, [Online]. Available: <https://arxiv.org/abs/2406.02061v4>
3. F. Chollet, “On the measure of intelligence,” *arXiv preprint arXiv:1911.01547v2*, Nov. 2019, [Online]. Available: <https://arxiv.org/abs/1911.01547v2>
4. N. Dziri, X. Lu, M. Sclar, X. L. Li, L. Jiang, *et al.*, “Faith and fate: Limits of transformers on compositionality,” *arXiv preprint arXiv:2305.18654v3*, May 2023, [Online]. Available: <https://arxiv.org/abs/2305.18654v3>
5. M. Du, F. He, N. Zou, D. Tao, and X. Hu, “Shortcut learning of large language models in natural language understanding,” *arXiv preprint arXiv:2208.11857v2*, Aug. 2022, [Online]. Available: <https://arxiv.org/abs/2208.11857>
6. Numbat, “ChatGPT4o Issues (Examples Repository),” GitHub, 2024. [Online]. Available: https://github.com/ainumbat/ChatGPT4o_issues.git
7. F. Chollet, “Talk at AGI Conference, ARC Prize,” YouTube, 2024. [Online]. Available: <https://www.youtube.com/watch?v=nL9jEy99Nh0&t=1450s>
8. X. L. Li, A. Kuncoro, J. Hoffmann, *et al.*, “A systematic investigation of commonsense knowledge in large language models,” *arXiv preprint arXiv:2111.00607v3*, Jul. 2022, [Online]. Available: <https://arxiv.org/abs/2111.00607v3>
9. J. Wei, Y. Tay, R. Bommasani, C. Raffel, *et al.*, “Emergent abilities of large language models,” *arXiv preprint arXiv:2206.07682v2*, Jun. 2022, [Online]. Available: <https://arxiv.org/abs/2206.07682v2>
10. J. Wei, X. Wang, D. Schuurmans, *et al.*, “Chain-of-thought prompting elicits reasoning in large language models,” *arXiv preprint arXiv:2201.11903v6*, Jan. 2022, [Online]. Available: <https://arxiv.org/abs/2201.11903v6>
11. Z. Yin, Q. Sun, Q. Guo, *et al.*, “Do large language models know what they don’t know?” *arXiv preprint arXiv:2305.18153v2*, May 2023, [Online]. Available: <https://arxiv.org/abs/2305.18153v2>
12. M. Turpin, J. Michael, E. Perez, S. R. Bowman, *et al.*, “Language models don’t always say what they think: Unfaithful explanations in chain-of-thought,” *arXiv preprint arXiv:2305.04388v2*, May 2023, [Online]. Available: <https://arxiv.org/abs/2305.04388v2>
13. G. Wenzel and A. Jatowt, “An overview of temporal commonsense reasoning and acquisition,” *arXiv preprint arXiv:2308.00002v3*, Aug. 2023, [Online]. Available: <https://arxiv.org/abs/2308.00002v3>
14. F. Chollet, M. Knoop, G. Kamradt, and B. Landers, “ARC Prize 2024: Technical report,” *arXiv preprint arXiv:2412.04604v2*, Dec. 2024, [Online]. Available: <https://arxiv.org/abs/2412.04604v2>
15. J. Zhao, J. Tong, Y. Mou, *et al.*, “Exploring the compositional deficiency of large language models in mathematical reasoning through trap problems,” *arXiv preprint arXiv:2405.06680v4*, May 2024, [Online]. Available: <https://arxiv.org/abs/2405.06680v4>

16. M. T. Bennett, “Is complexity an illusion?” *arXiv preprint arXiv:2404.07227v4*, Apr. 2024, [Online]. Available: <https://arxiv.org/abs/2404.07227v4>
17. T. B. Brown, B. Mann, N. Ryder, M. Subbiah, *et al.*, “Language models are few-shot learners,” *arXiv preprint arXiv:2005.14165v4*, May 2020, [Online]. Available: <https://arxiv.org/abs/2005.14165v4>
18. S. Banerjee, A. Agarwal, and S. Singla, “LLMs will always hallucinate, and we need to live with this,” *arXiv preprint arXiv:2409.05746v1*, Sep. 2024, [Online]. Available: <https://arxiv.org/abs/2409.05746v1>
19. M. Herrmann, J. D. Lange, K. Eggensperger, *et al.*, “Position: Why we must rethink empirical research in machine learning,” *arXiv preprint arXiv:2405.02200v2*, May 2024, [Online]. Available: <https://arxiv.org/abs/2405.02200v2>
20. Z. Wu, L. Qiu, A. Ross, *et al.*, “Reasoning or reciting? Exploring the capabilities and limitations of language models through counterfactual tasks,” *arXiv preprint arXiv:2307.02477v3*, Jul. 2023, [Online]. Available: <https://arxiv.org/abs/2307.02477v3>
21. E. Akyürek, M. Damani, L. Qiu, *et al.*, “The surprising effectiveness of test-time training for abstract reasoning,” *arXiv preprint arXiv:2411.07279v1*, Nov. 2024, [Online]. Available: <https://arxiv.org/abs/2411.07279v1>
22. M. N. H. Rahman and S.-H. Son, “Feature transforms for image data augmentation,” *Neural Computing and Applications*, vol. 34, pp. 16141–16160, 2022. doi:10.1007/s00521-022-07645-z.
23. Y.-H. Kim, J.-M. Ahn, S.-H. Jang, S.-K. Kim, and H.-K. Kim, “Data augmentation method by applying color perturbation of inverse PSNR and geometric transformations for object recognition based on deep learning,” *Applied Sciences*, vol. 10, no. 11, 3755, 2020. doi:10.3390/app10113755.
24. T. A. Chang and B. K. Bergen, “Language model behavior: A comprehensive survey,” *arXiv preprint arXiv:2303.11504v2*, Mar. 2023, [Online]. Available: <https://arxiv.org/abs/2303.11504v2>
25. D. C. Dennett, “The Role of Language in Intelligence,” in *Brainstorms: Philosophical Essays on Mind and Psychology*. De Gruyter, 2013. Available: <https://doi.org/10.1515/9783110806595>
26. Voltaire, *Philosophical Dictionary*, H. I. Woolf, Trans. New York: Knopf, 1918.
27. OpenAI, “ChatGPT,” 2023. [Online]. Available: <https://chat.openai.com>
28. Google DeepMind, “Gemini,” 2024. [Online]. Available: <https://deepmind.google/technologies/gemini>
29. xAI, “Grok,” 2024. [Online]. Available: <https://x.ai>
30. DeepSeek, “DeepSeek Language Model,” 2024. [Online]. Available: <https://deepseek.com>
31. H. Zhao, F. Yang, H. Lakkaraju, and M. Du, “Towards uncovering how large language model works: An explainability perspective,” *arXiv preprint arXiv:2402.10688v2*, Feb. 2024, [Online]. Available: <https://arxiv.org/abs/2402.10688>
32. F. Chollet, M. Knoop, G. Kamradt, and B. Landers, “ARC Prize 2024: Technical Report,” *arXiv preprint arXiv:2412.04604v2*, Dec. 2024, [Online]. Available: <https://doi.org/10.48550/arXiv.2412.04604>
33. F. Chollet, M. Knoop, G. Kamradt, and B. Landers, “ARC-AGI-2: A New Challenge for Frontier AI Reasoning Systems,” *arXiv preprint arXiv:2505.11831v1*, Jun. 2025, [Online]. Available: <https://doi.org/10.48550/arXiv.2505.11831>
34. J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, *et al.*, “Scaling laws for neural language models,” *arXiv preprint arXiv:2001.08361*, Jan. 2020, [Online]. Available: <https://doi.org/10.48550/arXiv.2001.08361>
35. S. Legg and M. Hutter, “Universal intelligence: A definition of machine intelligence,” *Minds and Machines*, vol. 17, no. 4, pp. 391–444, 2007.
36. D. Deutsch, “Constructor theory,” *Synthese*, vol. 190, no. 18, pp. 4331–4359, 2015.
37. M. Minsky, *The Society of Mind*. Simon & Schuster, 1986.
38. R. V. Yampolskiy, “Artificial Intelligence Safety Engineering: Why Machine Ethics Is a Wrong Approach,” in *Philosophy and Theory of Artificial Intelligence*, Springer, 2015.
39. J. Schmidhuber, “Gödel machines: Fully Self-Referential Optimal Universal Self-Improvers,” *arXiv preprint arXiv:0705.1865v3*, May 2007, [Online]. Available: <https://arxiv.org/abs/0705.1865>
40. M. Hutter, *Universal Artificial Intelligence: Sequential Decisions Based on Algorithmic Probability*. Berlin, Germany: Springer, 2005.

Author contributions statement



Rashid Mehmood is an independent researcher specializing in artificial intelligence, machine learning, and full-stack system development. His work focuses on improving reasoning, adaptability, and test-time learning in AI systems, with the broader goal of advancing paths toward Artificial General Intelligence (AGI). He has developed lightweight, resource-efficient algorithms and adaptive assistants designed to reduce catastrophic forgetting and enhance real-time inference. Recently, he demonstrated that strong generalization can be achieved from extremely sparse data, achieving over **80%** accuracy on MNIST using only **1%** of the training set. His research continues to explore efficient learning, abstraction, and dynamic knowledge recalibration.



Dr. Eid Rehman is currently serving as an Assistant Professor of Computer Science at the University of Mianwali, Pakistan. He earned his Ph.D. in Computer Science from the International Islamic University, Islamabad, in 2018. Throughout his academic and research career, Dr. Rehman has made significant contributions to the fields of Artificial Intelligence, Large Language Models (LLMs), and Information Security. His passion for advancing knowledge in emerging technologies is reflected in his prolific research record, having authored and co-authored more than 25 research papers published in well-reputed national and international journals. Dr. Rehman's research work bridges theory and practical application, contributing valuable insights to cutting-edge areas critical to today's technological advancements. He remains actively engaged in research, mentoring students, and participating in collaborative projects to foster innovation and excellence in Computer Science. His commitment to academic excellence and research innovation continues to inspire the next generation of computer scientists at the University of Mianwali and beyond.



Dr. Muhammad Habib received his Ph.D. in Computer Science from International Islamic University Islamabad, Pakistan, in 2018. His research interests include Computer Vision, Machine Learning, Deep Learning, Generative AI, and Agentic AI. He has published numerous research papers in reputable journals, contributing significantly to advancements in intelligent systems and AI-driven technologies. His work focuses on developing innovative algorithms and methodologies to enhance machine perception and automation.