Mirage of Mastery: Memorization Tricks LLMs into Artificially Inflated Self-Knowledge

Anonymous Authors[1]

Abstract

When artificial intelligence mistakes memorization for intelligence, it creates a dangerous perception of reasoning. Existing studies treat memorization and self-knowledge deficits in LLMs as separate issues. In our study, we pinpoint an intertwined causal link between the two that undermines the trustworthiness of LLM responses. To investigate this, we utilize a novel framework to ascertain if LLMs genuinely learn reasoning patterns from training data or merely memorize them to assume competence across problems of similar complexity focused on STEM domains. Our analysis shows a noteworthy problem in generalization: LLMs draw confidence from memorized solutions to infer a higher self-knowledge about their reasoning ability, which manifests as an over 45% inconsistency in feasibility assessments when faced with self-validated, logically coherent task perturbations. This effect is most pronounced in science and medicine domains. Our code and results are available publicly. [1]

1. Introduction

For true reliability and trustworthiness, AI tools, especially generative models like large language models (LLMs) must consistently and accurately recognize the boundary of their capabilities, referred to as self-knowledge (Yin et al., 2023). While methodologies studying LLM self-knowledge are rife (Wang et al., 2023; Wen et al., 2024; Ren et al., 2024), only rarely have other challenges like memorization (Slonski, 2024) been attributed as underlying causes for inaccuracies and overconfidence in self-knowledge.

Memorization in LLMs refers to the propensity to store and reproduce training data rather than develop genuine understanding (Satvaty et al., 2025; Prashanth et al., 2024). Numerous techniques to uncover such behaviour have been presented, including methods that utilize injected sequences (Huang et al., 2024), neuron activations (Slonski, 2024), counterfactual reasoning tasks (McCoy et al., 2024) and dynamic, prefix-dependent soft prompts (Wang et al., 2024). As data used for LLM training scales to include more problems and solutions, especially in STEM fields, models may conflate the ability of 'knowing' solutions with genuine reasoning power to solve such problems. This memorization-driven behaviour raises a critical downstream concern: if LLMs mistake recall for reasoning, they may falsely perceive their own knowledge to be deeper than it truly is, leading to unreliable and overconfident responses.

In our methodology, we give language models the freedom to provide a problem in STEM fields that they are confident of solving and evaluate their consistency in feasibility analyses on perturbed problems of similar complexity. If LLMs rely on memorization for self-knowledge, their accuracy and consistency in answering and determining the feasibility of tasks should falter when faced with minor logical perturbations (Xie et al., 2025). Such misplaced trust in AI reasoning in high-stakes fields like science, law, and medicine can lead to critical consequences. Our key contributions can be summarized as follows:

1. We identify a key link between two known problems of language models and present an effective method to analyse the same
2. To quantify the interplay between self-knowledge and memorization, we provide a universally applicable task perturbation pipeline and quantifying metrics
3. We show how overconfidence and a lack of true reasoning awareness in LLMs stems from memorized patterns, and observe a significant lack of consistency in self-knowledge

2. Experimentation Methodology

We present a novel experimentation approach to analyse if LLMs' tendency to memorise problems and corresponding solutions builds an inflated perception of their own knowledge and capabilities. Towards this goal, we design a dynamic experimentation approach presented in Figure 1. Our methodology builds on the self-knowledge evaluation technique given by Kale & Nadadur (2025) and the principle

[1] `https://anonymous.4open.science/r/LLM-Memorization_SK_Eval--543D/`

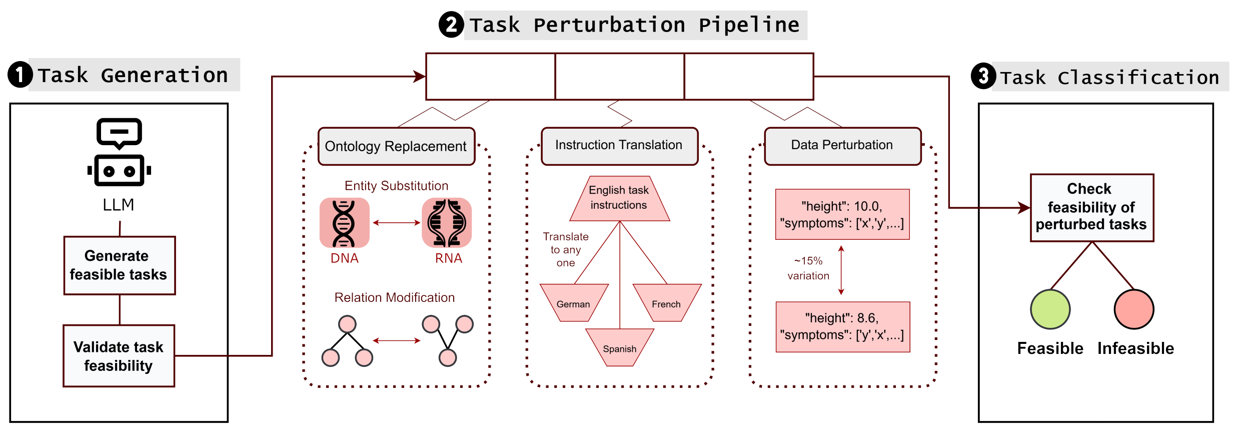

Figure 1. Methodology to analyze how LLM memorization inflates self-perception using minor task perturbations

from Xie et al. (2025) that memorization is characterized by high accuracy on familiar problems but a sharp decline with slight variations.

2.1. Task Generation

In the initial stage, we use a structured prompt to generate tasks in STEM fields (refer to the prompt in Figure 4 in Appendix A) with task instructions and associated data, if any, with the constraint that the LLM finds them completely feasible. During the generation phase, we allow LLMs the flexibility to set their own feasibility boundaries and use the QAP (Yugeswardeenoo et al., 2024) approach to ensure introspection. Since LLMs are prone to inconsistencies in self-knowledge assessments (Kale & Nadadur, 2025), we incorporate a separate prompt-based validation step (refer to the prompt in Figure 5 in Appendix A) within the task generation phase to ensure confidence in feasibility. We feed such validated feasible tasks to the task perturbation pipeline ahead.

2.2. Task Perturbation Pipeline

In order to check whether LLMs' self-knowledge about reasoning capabilities is indeed memorization-driven, we design a pipeline to add minor perturbations to the feasible tasks generated in the previous step while maintaining complete logical cohesion, similar computational complexity and domain relevance. Our pipeline consists of 3 main modules:

Ontology replacement module consists of substituting key domain-specific terms in the original task instructions and data with equivalent terminology from the same STEM domain. If possible, we also modify relationships between entities in the task while preserving logical coherence. This step minimizes keyword-driven memorization while preserving the task's core formulation and structure. We implement this module by setting up a Gemini 1.5 Flash model (Team, 2024) with the prompt shown in Figure 6 in Appendix A.

The **instruction translation** module involves changing the language of the English task instructions to any other high-resource language like German, Spanish or French using the Google Translate API. By only translating instructions and not data, we avoid inconsistencies in domain-specific knowledge, which is often best understood by models in English.

The **data perturbation** module uses a simple rule-based approach to introduce an approximately 15% variation in all numerical values present in the task data. Since LLMs are prone to memorization of numerical patterns and data (Bordt et al., 2024), this step is crucial to dissociate tasks from such patterns. We also reorder all unordered data elements, including lists and arrays in the task data, to minimize pattern matching while responding.

2.3. Task Classification

In the final stage, we feed all perturbed tasks generated through the pipeline to the LLM to attempt (refer to the prompt in Figure 7 in Appendix A). For each task, the LLM is prompted to either generate a conclusive answer (and thus classify the task as feasible) or mark it as infeasible. Since all tasks are validated to be feasible and are perturbed in a way that maintains domain relevance and complete logical cohesion, an inconsistency in feasibility assessments strongly implies that the LLMs relied on memorization of the data or solution steps of the original task to draw confidence for an inflated sense of self-knowledge and reasoning capacity in that domain.

Model	**Total ↓**	**Science**	**Technology**	**Engineering**	**Medicine**
GPT-4o	0.79	0.85	0.82	0.66	0.84
DeepSeek-V3	0.79	0.83	0.88	0.64	0.79
Mistral Large 24.11	0.58	0.96	0.42	0.55	0.40
Claude 3.7 Sonnet	0.46	0.73	0.31	0.37	0.42

Table 1. MIRAGE scores for all models distributed across STEM domains

3. Evaluation

3.1. Formulation

To formulate our approach mathematically, we can describe it as follows. First, we prompt an LLM to generate a task τ, where τ is explicitly deemed feasible by the model ($F(\tau) = 1$ for feasible tasks) and confirmed via another self-validation. Upon repeating m times to get m distinct tasks, each generated task is then perturbed n times via the perturbation pipeline that modifies surface-level features. Finally, we get a total of $n * m$ perturbed tasks represented as $\{P_1(\tau), P_2(\tau), \ldots, P_{n*m}(\tau)\}$.

The perturbed task set is then re-evaluated by the LLM, where each task is either classified as feasible or infeasible. An inconsistency in classification, i.e., $F(\tau_i) \neq F(P_j(\tau_i))$, indicates that the model relied on memorization rather than genuine reasoning ability to assess the feasibility of the original task, leading to overconfidence in self-knowledge.

3.2. Metrics

To quantify the proportion of times how often an LLM changes its feasibility stance, we introduce a new metric called MIRAGE (Memorization Induced Reasoning Assumption & GEneralization). It represents the mean infeasibility rate across perturbed tasks, averaged over all sets. A high MIRAGE score shows a high proportion of flipping judgements, implying that models considered the original task to be feasible mainly because of memorized patterns.

$$MIRAGE = \frac{1}{m}\sum_{\tau=1}^{m}\frac{\sum_{i=1}^{n}\mathbb{1}(F(\tau_i)=0)}{n} \tag{1}$$

Similarly, our research findings can also be used to methodically quantify and analyse consistency in self-knowledge of LLMs across various domains. For this purpose, we propose a metric called SKEW (Self KnowlEdge Wavering). SKEW quantifies the inconsistency in feasibility assessment for very similar problems with minor perturbations such that a higher score implies lower agreement, indicating poor self-knowledge about feasibility boundaries. In this case, we also include the original task in our problem set S, meaning each of the m sets has $t = n + 1$ perturbed and original tasks.

$$SKEW = \frac{1}{m}\sum_{S=1}^{m}\frac{\sum_{\tau_i,\tau_j \in S, \tau_j > \tau_i}\mathbb{1}(F(\tau_i) \neq F(\tau_j))}{\binom{T}{2}} \tag{2}$$

3.3. Setup

For a comprehensive analysis, we experiment with a wide range of high-performance models. Since our methodology is universal, we choose 2 closed-source models, GPT-4o (OpenAI, 2024) and Claude 3.7 Sonnet (Anthropic, 2024), and 2 open-source models, Mistral Large 24.11 (AI, 2024) and DeepSeek v3 (DeepSeek-AI, 2024), in our evaluation. For all models, to ensure diversity, we set the temperature to 1 during task generation. On the contrary, to encourage stability, we set the temperature to 0 during task classification. For each STEM domain and model, we generate 34 original tasks and perturb each task 3 times, leading to a dataset of 102 perturbed tasks per domain and 408 across all domains. MIRAGE scores across LLMs for all domains are presented in Table 1, while SKEW values analysing self-knowledge inconsistencies are given in Table 2.

4. Results and Analysis

4.1. Quantifying Memorization-inflated Self-Knowledge

F1.1 Self-knowledge inflation due to memorization is striking. The consistently high MIRAGE scores for all models, as shown in Table 1, point to an alarming flaw of inflated self-knowledge using memorization-driven confidence. Even high-performance models like GPT-4o and Mistral Large 24.11 change their feasibility stance about slightly perturbed tasks over 45% of times, meaning that LLMs are systematically overestimating their ability based on memorized solutions. A likely cause could be the prevalence of STEM benchmarks in training data, which could act more towards reinforcing memorization patterns than fostering robust reasoning in LLMs.

F1.2 Science and medicine are memorization hotspots. Almost all models show the highest MIRAGE scores for the science and medicine domains, with Mistral showing an extreme score of 0.96 for science. Since these fields tend to have the most frequent standardized jargon and textbook-style problem formats, models may draw overconfidence in reasoning abilities from such patterns. Future training data

Model	Total ↓	Science	Technology	Engineering	Medicine
GPT-4o	0.51	0.50	0.53	0.50	0.45
DeepSeek-V3	0.51	0.53	0.55	0.51	0.46
Mistral Large 24.11	0.35	0.34	0.51	0.40	0.42
Claude 3.7 Sonnet	0.32	0.42	0.48	0.40	0.36

Table 2. SKEW scores for all models distributed across STEM domains

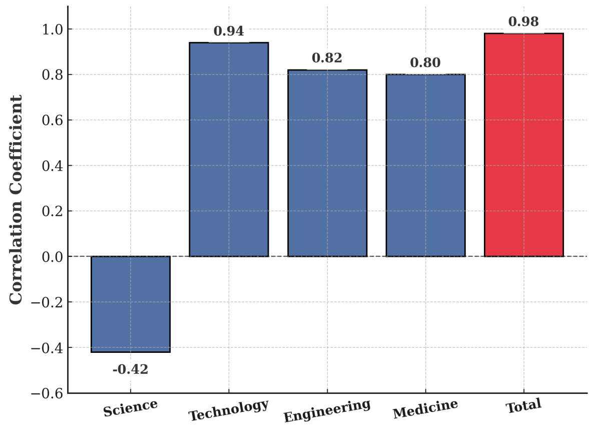

Figure 2. Correlation between SKEW and MIRAGE across STEM domains

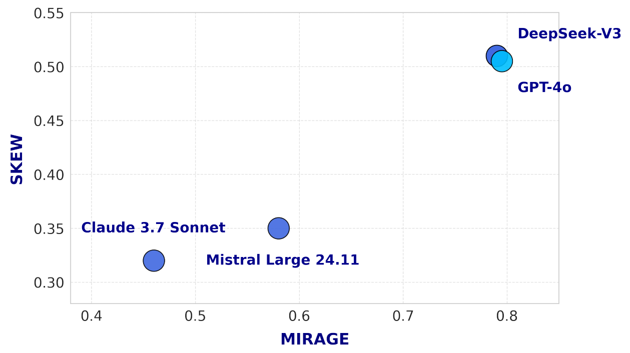

Figure 3. Results showing LLM performance metrics measuring memorization-driven self-knowledge inflation

should diversify science and medicine-related problems to ensure a stable outlook towards reasoning capacity.

F1.3 Architectural choice alone fails to regulate memorization-inflated self-knowledge risk consistently across domains. The general inconsistency in MIRAGE values across both domains and models suggests that a specific model or training architecture is not sufficient to manage risks of artificially increased self-knowledge across STEM domains. This challenges the notion that architectural improvements or scaling alone improve generalization and consistency.

4.2. Impact of Self-Knowledge Inconsistencies

F2.1 LLMs lack the capacity to establish generalizable feasibility boundaries. High SKEW values seen in Table 2, particularly for GPT-4o and DeepSeek-V3, indicate that even minor perturbations disrupt feasibility judgments, highlighting that models lack a generalized, consistent stance of their reasoning ability, especially in STEM domains. Moreover, the near-perfect positive correlation between SKEW and MIRAGE across all domains shown in Figure 2 (except science, which is affected by Mistral's high MIRAGE) provides noteworthy evidence that memorization-driven overconfidence and instability in self-knowledge are entwined problems in STEM-based results.

F2.2 LLMs are not trustworthy enough for critical real-world applications. We demonstrate how different LLMs exhibit domain-specific weaknesses across distinct STEM fields and recommend adaptive LLM routing strategies to be wary of these issues during model selection. Implementing safeguards like confidence thresholds and source markers can help flag uncertain responses, ensuring users are aware of potential inaccuracies before using AI-generated outputs. Due to their inflated self-perception, models are prone to generating responses even when lacking sufficient knowledge, rather than abstaining. Hence, human-in-the-loop fallback strategies are still important in LLM-powered applications for maximum trustworthiness.

5. Conclusion and Future Scope

Overconfidence in capabilities is a severe problem in LLMs, hampering their trustworthiness. Our research shows how models very likely draw confidence from 'knowing' solutions to develop an inflated perception of their reasoning power. All models alarmingly change their feasibility judgments for slightly perturbed tasks in over 45% of cases.

We hope that researchers can use our method to ensure more trustworthy and dependable LLM-powered applications. We also suggest expanding the pipeline to handle tasks without clear instruction-data separation, or adding languages and multimodal sources as directions for future improvement of this work.

Impact Statement

This work aims to advance the field of Machine Learning. While it may have various societal impacts, we do not identify any specific ethical concerns that require highlighting.

References

AI, M. Mistral large, 2024. URL https://mistral.ai/products/la-plateforme#models. Accessed: 2025-02-20.

Anthropic. Model card claude 3.7 sonnet, 2024. URL https://assets.anthropic.com/m/785e231869ea8b3b/original/claude-3-7-sonnet-system-card.pdf. Accessed: 2025-02-05.

Bordt, S., Nori, H., Rodrigues, V., Nushi, B., and Caruana, R. Elephants never forget: Memorization and learning of tabular data in large language models, 2024. URL https://arxiv.org/abs/2404.06209.

DeepSeek-AI. Deepseek-v3 technical report, 2024.

Huang, J., Yang, D., and Potts, C. Demystifying verbatim memorization in large language models. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N. (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 10711–10732, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.598. URL https://aclanthology.org/2024.emnlp-main.598/.

Kale, S. and Nadadur, V. Line of duty: Evaluating llm self-knowledge via consistency in feasibility boundaries, 2025. URL https://arxiv.org/abs/2503.11256.

McCoy, R. T., Yao, S., Friedman, D., Hardy, M. D., and Griffiths, T. L. Embers of autoregression show how large language models are shaped by the problem they are trained to solve. *Proceedings of the National Academy of Sciences*, 121(41):e2322420121, 2024. doi: 10.1073/pnas.2322420121. URL https://www.pnas.org/doi.

OpenAI. Gpt-4o system card, 2024. URL https://openai.com/index/gpt-4o-system-card/. Accessed: 2025-02-20.

Prashanth, U. S., Deng, A., O'Brien, K., V, J. S., Khan, M. A., Borkar, J., Choquette-Choo, C. A., Fuehne, J. R., Biderman, S., Ke, T., Lee, K., and Saphra, N. Recite, reconstruct, recollect: Memorization in lms as a multifaceted phenomenon, 2024. URL https://arxiv.org/abs/2406.17746.

Ren, R., Wang, Y., Qu, Y., Zhao, W. X., Liu, J., Tian, H., Wu, H., Wen, J.-R., and Wang, H. Investigating the factual knowledge boundary of large language models with retrieval augmentation, 2024. URL https://arxiv.org/abs/2307.11019.

Satvaty, A., Verberne, S., and Turkmen, F. Undesirable memorization in large language models: A survey, 2025. URL https://arxiv.org/abs/2410.02650.

Slonski, E. Detecting memorization in large language models, 2024. URL https://arxiv.org/abs/2412.01014.

Team, G. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context, 2024. URL https://arxiv.org/abs/2403.05530.

Wang, Y., Li, P., Sun, M., and Liu, Y. Self-knowledge guided retrieval augmentation for large language models. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 10303–10315, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.691. URL https://aclanthology.org/2023.findings-emnlp.691/.

Wang, Z., Bao, R., Wu, Y., Taylor, J., Xiao, C., Zheng, F., Jiang, W., Gao, S., and Zhang, Y. Unlocking memorization in large language models with dynamic soft prompting. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N. (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 9782–9796, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.546. URL https://aclanthology.org/2024.emnlp-main.546/.

Wen, Z., Tian, Z., Jian, Z., Huang, Z., Ke, P., Gao, Y., Huang, M., and Li, D. Perception of knowledge boundary for large language models through semi-open-ended question answering, 2024. URL https://arxiv.org/abs/2405.14383.

Xie, C., Huang, Y., Zhang, C., Yu, D., Chen, X., Lin, B. Y., Li, B., Ghazi, B., and Kumar, R. On memorization of large language models in logical reasoning, 2025. URL https://arxiv.org/abs/2410.23123.

Yin, Z., Sun, Q., Guo, Q., Wu, J., Qiu, X., and Huang, X. Do large language models know what they don't know?, 2023. URL https://arxiv.org/abs/2305.18153.

Yugeswardeenoo, D., Zhu, K., and O'Brien, S. Question-analysis prompting improves LLM performance in reasoning tasks. In Fu, X. and Fleisig, E. (eds.), *Pro-*

ceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop), pp. 402–413, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-srw.45. URL https://aclanthology.org/2024.acl-srw.45/.

A. Prompt formats

This section presents the format of all the prompts we use in our experimentation. The prompt format used to generate and validate feasible tasks in STEM domains is shown in Figures 4 and 5, respectively. The prompt used for the ontology replacement module is shown in Figure 6. During task classification, the model is guided to answer only if it deems the task to be feasible; otherwise, it is asked to provide an explanation for infeasibility, as described in the prompt in Figure 7.

Prompt used for generation of feasible tasks

```
You are a highly capable model with a strong understanding of your self-knowledge and boundaries.
Generate a task that you find difficult but feasible and you can answer only by extending all your
capability. Ensure the task is just the right side of your feasibility boundary!
The task should only be in the <STEM> domain with formal terminology. Generate such a task with the
following structured metadata:

- **task_instructions**: A precise, complete and structured description of the task.
- **task_data**: An exhaustive, complete and structured dataset containing real-world entities, numeric
variables and whatever might be needed for the task.
- **mathematical_formulation**: Equations, probabilities, or logical constraints relevant to the task.
- **ontology**:
- **entities**: List of key terms and domain-specific concepts.
- **relations**: List of relationships between entities.
```

Figure 4. Prompt format used to generate feasible tasks in any particular STEM domain

Prompt used for validation of feasible tasks

```
You are given a simple task. Analyse the task instructions and task data given to you.
Determine if the given task is feasible or infeasible as you are in your current state.

IMPORTANT: If it is feasible, only return the word FEASIBLE in caps without formatting.
Otherwise, return the word INFEASIBLE in caps without formatting only.

Here is the task:
Task Instructions: <task_instructions>
Task Data: <task_data>
Mathematical Formulation: <math_formula>
```

Figure 5. Prompt format used to validate feasible tasks and ensure confidence in answering self-generated tasks

Prompt used for ontology replacement in task perturbation

Can you perturb the given task JSON in a way in a stepwise instruction pipeline as I describe below? Here are your instructions, only follow these:

1. Substitute the ontology entities with other <domain> related terms or synonyms that fit in the same context and maintain all other relations. Then, substitute this new term in the task instructions accordingly without changing anything else.

2. Change the task data to match the new task ontology entities only. Add task data with appropriate keys and values only if needed to match the new ontology entities or relations. Make sure that the task data is complete and exhaustive for the new task. Otherwise, add in missing data.

3. Modify the mathematical formulation to match the new task ontology entities and relations. Make sure the final task makes sense and is very similar to the original.

4. Return only the modified JSON.

Figure 6. Prompt format used for the ontology replacement module in the task perturbation pipeline

Prompt used for task classification

You are a highly capable model with a strong understanding of yourself and your boundaries.

Analyse the task instructions and task data given to you.
Determine if the given task is feasible or infeasible for you to answer in your current state.

IMPORTANT: If it is feasible, return the word FEASIBLE in caps without formatting, and then give me a brief solution in the next line onwards after the \n character. Otherwise, return the word INFEASIBLE in caps without formatting, and then give me a concise explanation in the next line onwards after the \n character.

Here is the task:
Task Instructions: <task_instructions>
Task Data: <task_data>
Mathematical Formulation: <math_formula>

Figure 7. Prompt format used to classify the perturbed tasks generated by the task perturbation pipeline