Large Language Models Could Be Rote Learners

Anonymous ACL submission

Abstract

Multiple-choice question (MCQ) benchmarks are widely used for evaluating Large Language Models (LLMs), yet their reliability is undermined by benchmark contamination. In this study, we reframe contamination as an inherent aspect of learning and seek to disentangle genuine capability acquisition from superficial memorization in LLM evaluation. First, by analyzing model performance under different memorization conditions, we uncover a counterintuitive trend: LLMs perform worse on memorized MCQs than on non-memorized ones, indicating the coexistence of two distinct learning phenomena, *i.e.*, rote memorization and genuine capability learning. To disentangle them, we propose **TrinEval**, a novel evaluation framework that reformulates MCQs into an alternative trinity format, reducing memorization while preserving knowledge assessment. Experiments validate TrinEval's effectiveness in reformulation, and its evaluation reveals that common LLMs may memorize by rote 20.5% of knowledge points (in MMLU on average).

1 Introduction

The rapid advancement of Large Language Models (LLMs), driven primarily by large-scale pretraining on massive datasets, has endowed these models with remarkable proficiency across diverse tasks (Ouyang et al., 2022; OpenAI, 2024; Touvron et al., 2023). As LLMs continue to improve, evaluating their genuine capacities has emerged as a fundamental challenge, necessitating proper methodologies to ensure fairness and robustness (Ganguli et al., 2023; Liu et al., 2023b).

Among the developed methods, multiple-choice question (MCQ) benchmarks have become a standard approach for evaluation. Typically, LLMs are presented with a question and a fixed set of answer choices, requiring them to select the most appropriate option (see Fig. 1 for illustration). This format enables straightforward performance measurement through accuracy metrics and could cover a wide range of subjects. However, despite their widespread adoption, MCQ-based evaluation raises concerns about reliability due to benchmark contamination (Li and Flanigan, 2024; Kim et al., 2024), *i.e.*, test data unintentionally appears in training corpora and models may exploit memorized content rather than demonstrating genuine understanding, inflating their apparent capabilities. For instance, Zhou et al. (2023) discovers that smaller models with deliberate pre-exposure could outperform their larger counterparts, thereby contradicting widely accepted scaling laws.

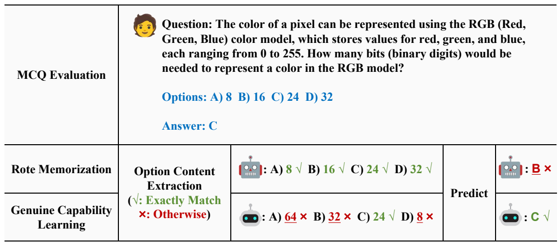


Figure 1: MCQ-based LLM evaluation. We observe that LLMs tend to underperform on memorized MCQs.

To mitigate the issue, Zhou et al. (2023) advocates the removal of benchmark datasets from pretraining corpora. However, this strategy conflicts with the fundamental objective of large-scale pretraining, which aims to maximize model performance by exposing LLMs to as much data as possible. From a broader perspective, human learning also involves problem-solving through practicing on similar questions, *e.g.*, exam preparation. While rote memorization of specific questions and answers merely lead to short-term success, repeated practicing can also facilitate deeper conceptual understanding. Inspired, rather than viewing benchmark contamination as a flaw to be eradicated, which is a nearly impossible task at scale (Sainz et al., 2023; Bordt et al., 2024), we argue that it is an inherent aspect of learning and should be accounted for in evaluation. Therefore, this study shifts its focus to *evaluating LLMs in the presence of contamination, aiming to distinguish genuine capability gains from superficial memorization effects*. The explicit disentangling of these two learning effects remains largely unexplored in MCQ-based evaluation, yet we believe it marks a crucial step towards developing more rigorous and unbiased evaluation methodologies.

To investigate the effects of superficial memorization in LLM evaluation, we compare model performance under different memorization conditions. Inspired by membership inference attacks (MIA) (Carlini et al., 2022a, 2021), we define superficial memorization as an LLM's ability to verbatim reproduce content, *e.g.*, MCQs in our case. Using this criterion, we partition the MMLU benchmark (Hendrycks et al., 2020)[1] into memorized and non-memorized subsets and evaluate three open-source LLMs[2] on both. Surprisingly, results reveal a consistent yet counterintuitive trend: LLMs perform worse on memorized MCQs than on those not (see Fig. 1 for illustration and Fig. 2 for results). This challenges the assumption that memorization improves model performance and suggests the co-existence of two distinct learning phenomena in LLMs: *rote memorization*, where models recall content verbatim without true understanding, and *genuine capability learning*, where they internalize underlying knowledge.

The preliminary investigation has several limitations. First, the binary classification of MCQs as either memorized or non-memorized oversimplifies the nuances of memorization, potentially overlooking intermediate cases. Second, we rely on accuracy to measure performance, which is inherently unreliable. Third, our analysis could not reveal the mutual effects between rote memorization and capability learning. To address these challenges, we propose **TrinEval**, a novel evaluation framework designed to provide a more reliable measure of LLM performance by minimizing the influence of rote memorization. TrinEval employs a query-based probing (q-probing) mechanism (Allen-Zhu and Li, 2023) that reformulates MCQs into an alternative trinity format, *i.e.*, entity-attribute-context. This could prevent direct content recall while preserving knowledge assessment.

Through experiments, we demonstrate that TrinEval's reformulation is knowledge-preserving, *i.e.*, maintaining testing problems' inherent knowledge requirements without introducing extra cues, and could effectively reduce memorization. Combined with a continuous superficial memorization quantification metric, TrinEval reveals the in-robustness of LLMs' capability learning, *e.g.*, with MMLU, tested open-sourced LLMs only mastered 19.6% of knowledge points while 20.5% are memorized by rote in the meanwhile, shedding light on the necessity for further optimization.

2 Related Work

2.1 LLM Evaluation on MCQ Benchmarks

The rapid advancement of LLMs has driven their expansion into diverse domains, necessitating robust and fair evaluation methodologies (Zheng et al., 2023b; Hu et al., 2025) and platforms (Contributors, 2023; Chiang et al., 2024). Among these, evaluating on MCQ benchmarks emerges as a widely adopted approach due to the ease of validation and standardized comparison across models (Hendrycks et al., 2020; Wang et al., 2024; Zhong et al., 2023; Huang et al., 2024).

However, MCQ-based evaluations are not without limitations. Biases in LLM responses have been extensively studied (Dai et al., 2024), revealing issues such as social biases (Salewski et al., 2024; Liu et al., 2023a) and order sensitivity (Akter et al., 2023). To mitigate the latter, Pride (Zheng et al., 2023a) estimates the option positional bias after option permutation. To examine mastery of knowledge, Zhao et al. (2023) applies a hypothesis testing method and checks rephrased-context consistency for a given question. Benchmark contamination is arguably the most severe challenge for MCQ-based evaluations, which may result in misleadingly inflated performance (Zhou et al., 2023; Li and Flanigan, 2024). To address this, prior studies have explored data filtering, frequently-updated test sets (White et al., 2025), and data perturbation (Li et al., 2024).

In this paper, instead of attempting to eliminate contamination, we evaluate LLMs under its presence, aiming to distinguish genuine capability gains from superficial memorization effects. This marks a new perspective of LLM evaluation, revealing the extent to which models truly understand concepts rather than merely memorizing data.

[1] Selected for its popularity and documented data contamination in widely used LLMs (Sainz et al., 2023).

[2] Llama2-7B (Touvron et al., 2023), Mistral-7B-v0.2 (Jiang et al., 2023) and Vicuna-v1.5-7B (Zheng et al., 2023b).

2.2 LLM Memorization

Membership inference attacks (MIA) are commonly used to determine whether a specific sample was present in a model's training data. Initially studied in smaller models, Carlini et al. (2022b) investigates deep learning memorization mechanisms by identifying and removing easily detectable memorized samples. In the context of LLMs, MIA has been employed to assess privacy risks, revealing that both open- and closed-source models can leak sensitive personal data when provided with related prompts (Kim et al., 2024).

Beyond privacy concerns, Carlini et al. (2022a) formally defines LLM memorization as a model's ability to verbatim generate text sequences following a prefix prompt. Using this definition, several studies (Sainz et al., 2023; Bordt et al., 2024; Carlini et al., 2021) have examined mainstream LLMs, confirming widespread test data leakage across popular benchmarks. To quantify memorization strength, researchers (Shi et al., 2023; Zhang et al., 2024; Oren et al., 2023; Carlini et al., 2019) have further explored methods such as analyzing token probability distributions in generated outputs. However, while these studies extensively analyze LLM memorization, few explicitly investigate how memorization influences an LLM's problem-solving ability. In contrast, our work focuses on their interplay, presenting a more rigorous approach to fair and reliable LLM evaluation.

3 Methodology

3.1 Pre-investigation of LLM Capability w.r.t. Memorization

Benchmark contamination often leads to inflated performance estimate. This phenomenon is commonly attributed to models memorizing specific questions and answers rather than demonstrating genuine problem-solving abilities. However, the extent to which and how memorization influences LLM performance remains unclear. To disentangle genuine capability acquisition from superficial memorization, we conduct a preliminary investigation into how LLMs perform under different memorization conditions. By examining model accuracy on memorized vs. non-memorized subsets, we aim to reveal the role of memorization in LLM evaluation and establish a foundation for more rigorous assessment methodologies.

Formally, we define an MCQ as $x = \{x_Q, x_O, x_W\}$, where x_Q, x_O, and x_W refer to the question, options, and ground-truth answer, respectively. Following the memorization definition from Carlini et al. (2022a), we say an MCQ x is memorized by LLM G if G can extract/generate the content of options x_O exactly given question x_Q. In practice, we incorporate meta-information (*e.g.*, benchmark name) and 5-shot examples to recall memory and use greedy decoding (*i.e.*, temperature fixed to 0) during extraction (Bordt et al., 2024; Sainz et al., 2023) (refer to Appendix A for the complete prompt). Using MMLU (Hendrycks et al., 2020) as the evaluation benchmark, we divide the test set MCQs into memorized and non-memorized subsets, where the memorized subset consists of 909–982 questions (accounting for 6.5%–7.0% of the total 14,006) depending on the tested LLMs Llama2-7B, Mistral-7B-v0.2, and Vicuna-v1.5-7B. The detailed statistics of questions across subsets are given in Table 1 of Appendix A, and we also observe that the majority of memorized questions are those relatively simple, *i.e.*, not in MMLU-PRO (Wang et al., 2024).

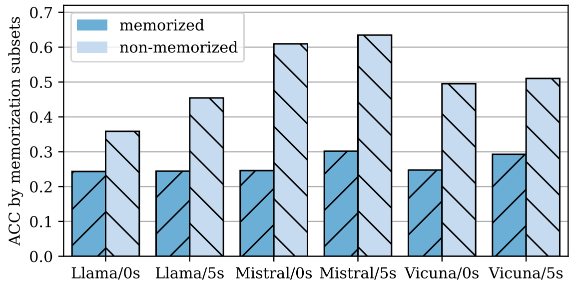


Figure 2: Model performance on memorized and non-memorized subsets of MMLU, where '0s' and '5s' stand for zero- and five-shot prompting, respectively.

We then compute the accuracy (ACC) of tested LLMs by subsets as a proxy of model performance under different memorization conditions. The results of both zero- and five-shot prompting are reported in Fig. 2, from which we observe a consistent yet somehow counterintuitive trend: LLMs exhibit 47.2% lower accuracy on average on memorized MCQs compared to non-memorized ones, regardless of LLMs and prompting techniques. This finding challenges the commonly held assumption that memorization directly improves model performance. In addition, it also implies the coexistence of two distinct learning paradigms within LLMs, which we term rote memorization and genuine capability learning, respectively.

However, our pre-investigation has its limitations. The binary classification of memorization potentially overlooks more nuanced forms of learning. Additionally, using ACC as the performance metric does not truly capture model capacity. We address these two issues in the following subsections, which then ensure a disentangle analysis of rote memorization and capability learning.

3.2 Quantifying LLM Memorization

For quantifying the memorization of LLMs, prior research (Shi et al., 2023; Zhang et al., 2024) suggests that outlier tokens, which exhibit higher generation probabilities, are more likely to be found in memorized samples. Building on this idea, we develop a metric that utilizes the bottom $K\%$ of token probabilities within the generated sequence as a measure of memorization. Formally, the memorization score $F_m(\overline{x}, G)$ of LLM G on text sequence $\overline{x}$ is computed as follows:

$$F_m(\overline{x}, G) = \frac{1}{|\mathcal{M}_K(\overline{x})|} \sum_{\overline{x}_i \in \mathcal{M}_K(\overline{x})} \log p_G(\overline{x}_i | \overline{x}_{1:i-1}), \tag{1}$$

where $p_G(\overline{x}_i | \overline{x}_{1:i-1})$ denotes the generation probability of token $\overline{x}_i$ by G given its prefix subsequence as context, and set $\mathcal{M}_K(\overline{x})$ includes the $K\%$ of tokens with the lowest probabilities. The higher F_m is, the more likely $\overline{x}$ is memorized by the LLM, *i.e.*, the least memorized content could still been extracted with a high probability.

3.3 Measuring LLM Capability with TrinEval

We next present TrinEval, a novel evaluation framework designed to provide a more reliable measure of LLM performance by minimizing the influence of rote memorization.

To understand how LLMs store and manipulate knowledge, Allen-Zhu and Li (2023) created a fictional biography dataset that enumerates various attributes (*e.g.*,, names, jobs, universities) and trained LLMs on this dataset. They employed a linear query-based probing method to uncover correlations between the entity token embeddings and the associated attributes, revealing that where LLMs encode knowledge, *e.g.*, under person names or sequence of the knowledge mention, is crucial for robust mastery of knowledge. This insight leads us to believe that entity tokens, which should ideally store related knowledge, are the target for evaluating an LLM's genuine capability.

However, applying this method to real-world datasets, such as MMLU, presents challenges. Unlike controlled datasets with explicitly defined attributes, real-world data includes a far broader range of possible knowledge. As a result, we cannot enumerate all potential attributes and directly apply linear probing. To this end, we propose TrinEval, a verbal query probing method that reformulates MCQs around a knowledge-centric trinity: knowledge entity, attribute, and context. TrinEval is a pluggable augmentation on any MCQ benchmarks and could expose the genuine capability of LLMs by verifying whether they have correctly encoded knowledge. We next explain the elements in the trinity and how to reformulate.

Knowledge entity. We suppose that if an LLM has mastered some knowledge, the key information pertinent to the knowledge should be encoded within a few subject tokens, namely knowledge entity, to support efficient retrieval. By isolating these tokens, TrinEval ensures that only the essential information is considered.

Attribute. The attribute acts as a verbal probe to guide the model focusing on the specific feature or property of the knowledge entity being inquired. This mechanism allows TrinEval to isolate and assess the model's understanding of the critical aspects of the questioning subject.

Context. In a certain portion of questions, the conditions or background context can significantly influence the solution approach. By explicitly including context in the evaluation process, TrinEval helps the model account for relevant situational details that might otherwise be overlooked, ensuring that the model's answer is based on a comprehensive understanding of the problem.

By extracting the core and necessary question information in this trinity format, the reformulation by TrinEval is knowledge-preserving for the purposes of assessment. In the meanwhile, it completely destructs the original token sequence, effectively reducing the influence of memorization. We will empirically verify these properties through experiments. The reformulation is completed by a two-round reflection-based prompting method, with detailed procedure (Alg. 1) and related prompts available in Appendix B. Given an MCQ $x = (x_Q, x_O, x_W)$, it first queries a capable reformulation LLM to derive the knowledge entity x_E, attribute x_A, and Context x_C from the original x. The LLM is instructed that the triplet should be sufficient for answering the question correctly, without including the answer option itself, ensuring the integrity of the evaluation. The same

LLM then assesses whether the triplet contains all necessary information and no redundant details, in the meanwhile, yields a rationale x_L as reflection (Shinn et al., 2024; Yao et al., 2022). If it does, the triplet is returned as the re-formulated question. Otherwise, the reformulation model refines the extraction, taking as input x_E, x_A, x_C, and x_L, and re-evaluates the updated triplet.

Finally, prompting with the extracted x_E, x_A, and x_C as well as options x_O, we inspect the generation probability of the ground-truth answer x_W as the first token to measure capability:

$$F_c(x, G) = p_G(x_W | x_E, x_A, x_C, x_O). \quad (2)$$

As can be seen, the F_c metric retains the necessary knowledge-centric information while discarding unnecessary biases, especially the rote memorization of LLMs, which leads to the quantification of genuine capability of LLMs.

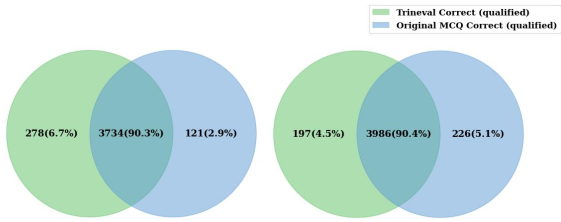
4 Experiments

In this section, we conduct extensive experiments to answer the following questions:
Q1. Is TrinEval knowledge-preserving in order to fulfill knowledge assessment?
Q2. Can TrinEval reduce memorization effects during capability evaluation?
Q3. What does TrinEval reveal about LLMs' rote memorization and genuine capability?

4.1 Experiment Setup

Models. We utilize API-based commercial LLMs, specifically gpt-4o-2024-08-06 (GPT) (OpenAI, 2024) and qwen-max-2024-09-19 (Qwen) (Yang et al., 2024; Team, 2024) for question reformulation by TrinEval. Model evaluation is conducted on open-source LLMs due to limited budgets, and we experiment with three popular LLMs including Llama2-7B (Llama) (Touvron et al., 2023), Mistral-7B-v0.2 (Mistral) (Jiang et al., 2023), and Vicuna-v1.5-7B (Vicuna) (Zheng et al., 2023b). All the three LLMs are accessed from Huggingface and implemented with transformers library, we thus could obtain the log-probability of output token for fine-grained study. Throughout our tests, we use the default generation parameters and adopt greedy decoding to enhance reproducibility.

Benchmarks. We evaluate LLM on the widely used MMLU (Hendrycks et al., 2020) benchmark. MMLU consists of 57 subjects from areas including STEM, humanities, social sciences, and others,



(a) Validation with Qwen (b) Validation with GPT

Figure 3: Knowledge-preserving validation for the reformulation by TrinEval. We obtain 4,343 and 4,645 qualified MCQs with Qwen and GPT, respectively, after reformulation. We then test Qwen and GPT in these qualified subsets and the green and blue circles stand for the correctly answered MCQs in TrinEval and original formats, respectively.

enabling comprehensive evaluation of LLM capacity. As there are duplicated MCQs across different subjects, we eliminate them and obtain 14,006 MCQs as the test set.

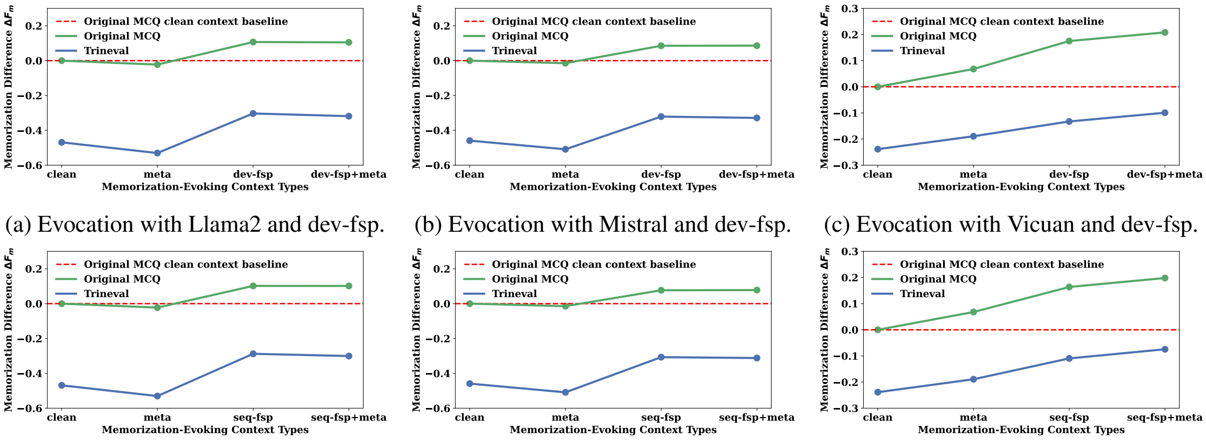
Evaluation. With commercial LLMs, we evaluate model performance by extracting answers with regular expressions. For open-source LLMs, we access the output probability of the first generated token (*e.g.*, option IDs A/B/C/D) to obtain a quantitative performance result.

4.2 Q1. Is TrinEval Knowledge-preserving?

We first verify whether the reformulation by TrinEval is knowledge-preserving in order to fulfill knowledge assessment. To achieve this, our primary objective is to validate that the reformulation approach (1) does not lose key information that results in previous correctly-answered questions being answered incorrectly and (2) does not introduce anomalous or unexpected information that results in inflated performance.

Upon completing the complete TrinEval reformulation process, we ultimately obtained 4,343 MCQs and their corresponding knowledge entities, attributes, and contexts that met our criteria using Qwen, as well as 4,645 qualified MCQs and their respective triplets using GPT. We then instruct Qwen and GPT to answer these respective questions in both the original (baseline) and restated triplet form. The prompts and an MCQ example are available in Table 2 in Appendix and the results are shown in Fig. 3.

We can observe that for Qwen, 92.95% of correctly answered MCQs in the TrinEval format maintain their accuracy in the original format, while for GPT, 90.05% of qualified MCQs are answered



(a) Evocation with Llama2 and dev-fsp. (b) Evocation with Mistral and dev-fsp. (c) Evocation with Vicuan and dev-fsp.

(d) Evocation with Llama2 and seq-fsp. (e) Evocation with Mistral and seq-fsp. (f) Evocation with Vicuan and seq-fsp.

Figure 4: The results of memorization evocation under various dataset-related information context, with blue and green curves referring to the memorization difference ΔF_m in the TrinEval and original formats, respectively. In the x-axis, 'clean', 'meta', 'dev-fsp', and 'seq-fsp' stand for without dataset-related context, with the name of the dataset, with few-shot prompt from the training set, and with few-shot prompt from the test set ahead of the current testing question. These curves indicate the growing memorization metric ΔF_m with the stronger dataset-related information in general. However, the ΔF_m by TrinEval under the strongest memory evocation context are consistently lower than those in the original format, *e.g.*, 'clean'.

correctly, with 95% of these maintaining accuracy in the original format. That is, for both Qwen and GPT, we can infer that the correctly answered MCQs from the qualified ones in TrinEval format can be regarded as a subset of the correctly answered MCQs with the original MCQ format. This proves that the proposed TrinEval reformulation method does not incorporate extra information that leads to additional capability of LLMs. On the other hand, the intersection MCQs between the correctly answered in two formats also make up of around 95% of the MCQs correctly answered in the original MCQ format, which proves that the TrinEval incorporates all the necessary information to answer the question. In conclusion, our TrinEval effectively retains the LLMs' problem-solving capability compared to the original MCQ text.

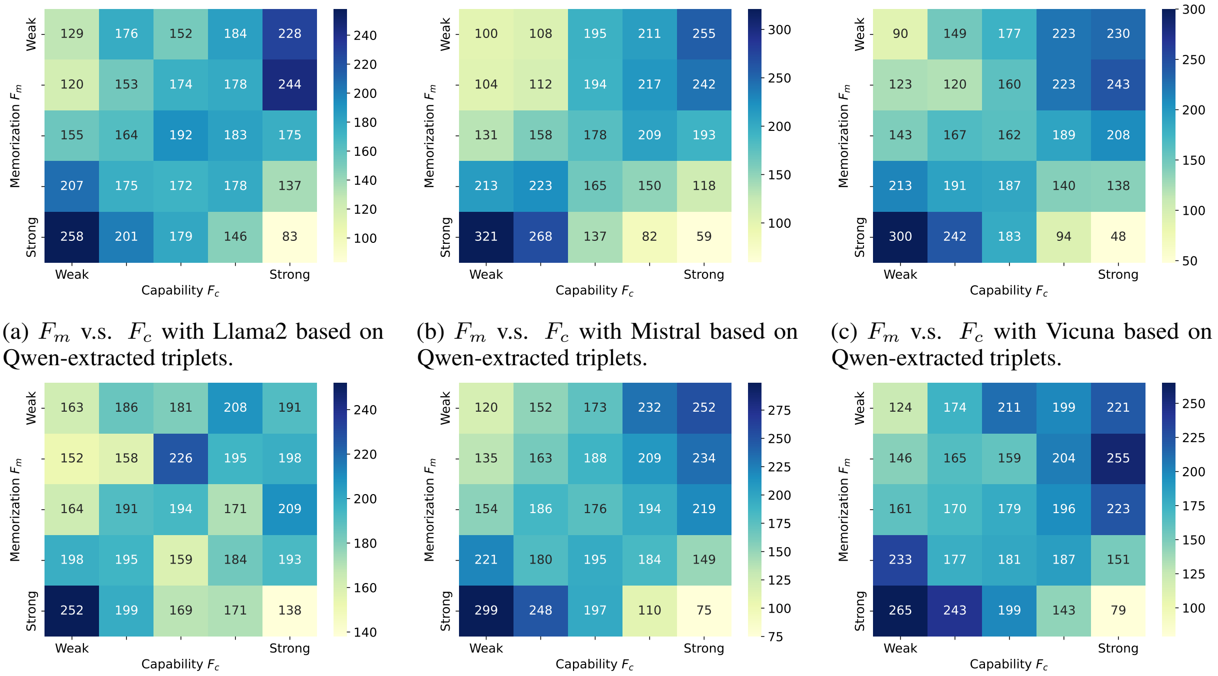
4.3 Q2. Can TrinEval Reduce Memorization?

In this subsection, we aim to validate whether the proposed TrinEval can eliminate the unnecessary memorization of LLMs, and thus demonstrate enhanced robustness against various perturbations. To answer this question, following Bordt et al. (2024), we deliberately incorporate the dataset-related information into the context and evaluate whether the TrinEval reformulation can suppress the growing memorization level with memorization evocation of different extent and can reveal the genuine capabilities of LLMs.

We incorporate the dataset-related information into the context, *i.e.*, the name of the dataset, and the few-shot prompt of samples within the same dataset for the memorization-evocation perturbation. Here we use Llama, Mistral, and Vicuna as the tested LLMs since we access the output probabilities to compute the memorization metric F_m. As there is no specific *zero point* of F_m indicating the absolute-no memorization of MCQs given an LLM, in order to better visualize the difference between the proposed TrinEval and the original MCQ baseline format, we take the F_m with vanilla MCQ (*i.e.*, original MCQ format without any dataset-related prompt) as the baseline and visualize the averaged difference between the F_m of the tested format and the baseline.

Specifically, for the memorization-evocation permutation, we progressively enhance the prompt context for memory evocation, starting from merely providing the dataset name, to offering samples within the same dataset as few-shot prompts (including the training set of the dataset-dev, and the preceding samples adjacent to the test sample-seq), and finally to providing both as context. Here for each tested MCQ, we calculate the difference between the F_m given the corresponding memorization evocation prompt and the F_m with the vanilla MCQ baseline. Fig. 4 shows the curve based on the average difference of each MCQ.

From this figure, as stated by Bordt et al. (2024),



(a) F_m v.s. F_c with Llama2 based on Qwen-extracted triplets.

(b) F_m v.s. F_c with Mistral based on Qwen-extracted triplets.

(c) F_m v.s. F_c with Vicuna based on Qwen-extracted triplets.

(d) F_m v.s. F_c with Llama2 based on GPT-extracted triplets.

(e) F_m v.s. F_c with Mistral based on GPT-extracted triplets.

(f) F_m v.s. F_c with Vicuna based on GPT-extracted triplets.

Figure 5: The distribution of MCQs based on Memorization metric F_m v.s. the Capability metric F_c. According to the values of F_m and F_c, we separate the MCQs equally into five groups and visualize the distribution of MCQs with the heatmap from weak to strong.

we can see that F_m is growing with the stronger dataset-related context. When providing more specific context related with the test dataset, the LLMs tend to exhibit stronger memorization of the MCQs. Specially, for all three open-source LLMs, the ΔF_m curve of TrinEval is below the curve of the original MCQ baseline. More importantly, the F_m of TrinEval with the strongest memorization evocation is still below the vanilla MCQ baseline, which proves that TrinEval can effectively eliminate the memorization from LLMs.

4.4 Q3. TrinEval's Findings on Memorization and Capability

In this subsection, we aim to explicitly study the relationship between the memorization and the capability of LLMs with the metrics F_m and F_c. As the commercial-API-based LLMs do not provide the output probability of the whole vocabulary, we mainly use the open-source LLMs to compute these two metrics. After obtaining the F_m and F_c of each MCQ, we separate all the qualified MCQs into 5 equal groups. Finally, we utilize the heatmap to reveal the relationship between the capability and the memorization of the tested LLMs.

As shown in Fig. 5, most of the MCQs con-

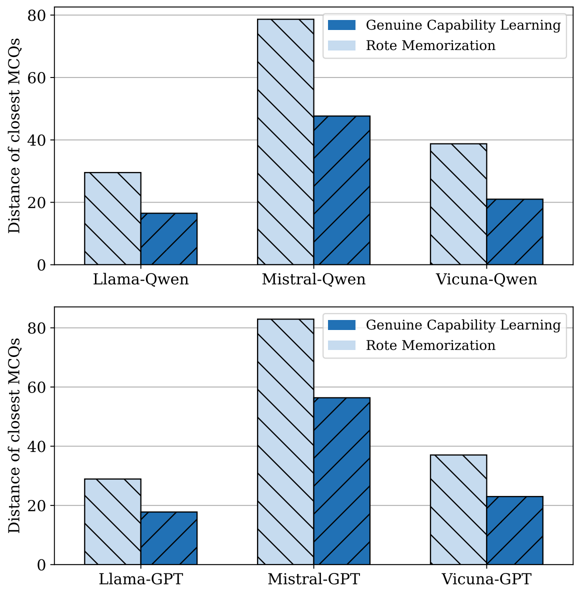


Figure 6: Averaged the distance of each MCQs between the closest 1% MCQs' embeddings. 'Rote Memorization' refers to MCQ within lower left 2×2 squares that typically exhibits high memorization metric F_m and low capability F_c while the 'Genuine Capability Learning' stands for the MCQ lies within the upper right 2×2 squares that has lower probability to be exactly retrieved but can be solved by LLMs.

centrate on the lower left corner and the upper right corner of the heatmap. Specifically, for F_m v.s. F_c with Llama2 based on Qwen-extracted triplets, MCQs within the lower left 2×2 squares and the upper right 2×2 squares make up of the 38.57% of all the tested MCQs with a Pearson-correlation of -0.7755 (p-value < 0.05), while the MCQs within the lower left and upper right 3×3 squares make up of the 74.17% of all the tested MCQs with a Pearson-correlation of -0.8124 (p-value < 0.05). For the results with Mistral based on Qwen-extracted triplets, MCQs within the lower left and upper right 2×2 squares make up of the 44.90% of all the tested MCQs with a Pearson-correlation of -0.8722 (p-value < 0.05), while the MCQs within the lower left and upper right 3×3 squares make up of the 80.82% of all the tested MCQs with a Pearson-correlation of -0.8794 (p-value < 0.05). More results are shown in Tab. 3. This evidence indicates that MCQs with lower memorization levels tend to exhibit better problem-solving capabilities of LLMs, while those with higher memorization levels are associated with reduced performance in solving tasks.

Next, we hypothesize that the LLMs are potential rote learners through the human memory system, which has been characterized by two fundamental components: Long-Term Memorization (LTM) and Short-Term Memorization (STM) Shiffrin (2003). Neurobiological studies reveal that STM relies on transient synaptic protein synthesis with limited temporal persistence and functional scalability. In contrast, LTM is constructed through stabilized neuronal memory traces that constitute an enduring knowledge framework. This neural architecture not only supports STM operations as a cognitive substrate but also enables sophisticated information generalization across diverse contexts. As illustrated in Allen-Zhu and Li (2023) and Ovadia et al. (2023), LLMs trained with multiple rephrased corpus tend to perform better than LLMs trained with only the original corpus. When providing only one format of training corpus, similar to the STM system, LLMs tend to memorize the corpus at token-level rather than knowledge-level. In other words, LLMs encode these corpora at a shallow level with the original format. After questions are rephrased with methods like our proposed TrinEval, the input corpus seems connected with the known knowledge like the LTM for structured storage and enables sophisticated information generalization. We show more detailed results in Appendix C.

To further validate our hypothesis, we compute the embeddings of the MCQs within the qualified MMLU dataset and average the distance between the other closest 1% MCQs. We visualize the mean distance of MCQs within the lower left and upper right 2×2 squares in Fig. 5. The results are shown in Fig. 6. We surprisingly find that the averaged distance of the Genuine Capability Learning MCQs (*i.e.*, MCQs within the upper right 2×2 squares) is almost half as much as the distance of the Rote Memorization MCQs (*i.e.*, MCQs within the lower left 2×2 squares). The result hints that the memorized MCQs are sparsely encoded by MCQs while the unmemorized ones share common embeddings, which is again coincident with the findings of the STM and LTM.

Though it is well believed that memorization may lead to a better but cheating performance of LLMs, we prove that the more LLMs memorize, the worse they are at solving problems.

5 Conclusion

This study provided a novel perspective on benchmark contamination in LLM evaluation, reframing it as an inherent aspect of learning. This perspective leaded us to explore the relationship between memorization and genuine capability in LLMs. Through our empirical investigation, we observed a surprising result: LLMs performed worse on memorized MCQs compared to those not, suggesting that superficial memorization may undermine problem-solving ability rather than enhance it. This finding also implies the existence of two distinct learning paradigms in LLMs: rote memorization and genuine capability learning.

To disentangle them, we proposed TrinEval, a novel evaluation method that reformulates MCQs into a knowledge-centric trinity, thus separating the influence of memorization from genuine knowledge application. Experiments validated both the knowledge-preserving and memorization-reducing properties of this approach. Based on that, TrinEval reveals the in-robustness of LLMs' knowledge learning, *e.g.*, popular open-source LLMs memorize 20.5% of knowledge points by rote without understanding in MMLU. As such, we believe this work lays the groundwork for future studies on improving LLM knowledge robustness and more thorough evaluation.

6 Limitations

Our limitations are mainly two points. First, though our proposed TrinEval retrains the problem-solving ability of the LLMs and obtains stronger robustness, it is not a dynamical re-organizing method that can still be leaked and pre-experienced during training. On the one hand, we appeal to the LLM developers not to use this re-organizing method as part of the training corpus. On the other hand, future works will be focused on developing dynamic evaluation method (Zhu et al., 2023, 2024). Second, we did not give a clear exploration on how and why the more LLMs memorize, the less the capability of the LLMs obtains. In future work, we will also look into the mechanism of the training and structure of LLMs for a thorough study of the phenomenon.

References

Syeda Nahida Akter, Zichun Yu, Aashiq Muhamed, Tianyue Ou, Alex Bäuerle, Ángel Alexander Cabrera, Krish Dholakia, Chenyan Xiong, and Graham Neubig. 2023. An in-depth look at gemini's language abilities. *arXiv preprint arXiv:2312.11444*.

Zeyuan Allen-Zhu and Yuanzhi Li. 2023. Physics of language models: Part 3.1, knowledge storage and extraction. *arXiv preprint arXiv:2309.14316*.

Sebastian Bordt, Harsha Nori, Vanessa Rodrigues, Besmira Nushi, and Rich Caruana. 2024. Elephants never forget: Memorization and learning of tabular data in large language models. *arXiv preprint arXiv:2404.06209*.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2022a. Quantifying memorization across neural language models. *arXiv preprint arXiv:2202.07646*.

Nicholas Carlini, Matthew Jagielski, Chiyuan Zhang, Nicolas Papernot, Andreas Terzis, and Florian Tramer. 2022b. The privacy onion effect: Memorization is relative. *Advances in Neural Information Processing Systems*, 35:13263–13276.

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. 2019. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX security symposium (USENIX security 19)*, pages 267–284.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E Gonzalez, et al. 2024. Chatbot arena: An open platform for evaluating llms by human preference. *arXiv preprint arXiv:2403.04132*.

OpenCompass Contributors. 2023. Opencompass: A universal evaluation platform for foundation models. https://github.com/open-compass/opencompass.

Sunhao Dai, Chen Xu, Shicheng Xu, Liang Pang, Zhenhua Dong, and Jun Xu. 2024. Unifying bias and unfairness in information retrieval: A survey of challenges and opportunities with large language models. *arXiv preprint arXiv:2404.11457*.

Deep Ganguli, Nicholas Schiefer, Marina Favaro, and Jack Clark. 2023. Challenges in evaluating AI systems.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Renjun Hu, Yi Cheng, Libin Meng, Jiaxin Xia, Yi Zong, Xing Shi, and Wei Lin. 2025. Training an llm-as-a-judge model: Pipeline, insights, and practical lessons.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Yao Fu, et al. 2024. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. *Advances in Neural Information Processing Systems*, 36.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Siwon Kim, Sangdoo Yun, Hwaran Lee, Martin Gubri, Sungroh Yoon, and Seong Joon Oh. 2024. Propile: Probing privacy leakage in large language models. *Advances in Neural Information Processing Systems*, 36.

Changmao Li and Jeffrey Flanigan. 2024. Task contamination: Language models may not be few-shot anymore. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18471–18480.

Jiatong Li, Renjun Hu, Kunzhe Huang, Yan Zhuang, Qi Liu, Mengxiao Zhu, Xing Shi, and Wei Lin. 2024. Perteval: Unveiling real knowledge capacity of LLMs with knowledge-invariant perturbations. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Yanchen Liu, Srishti Gautam, Jiaqi Ma, and Himabindu Lakkaraju. 2023a. Investigating the fairness of large language models for predictions on tabular data. *arXiv preprint arXiv:2310.14607*.

Yang Liu, Yuanshun Yao, Jean-Francois Ton, Xiaoying Zhang, Ruocheng Guo Hao Cheng, Yegor Klochkov, Muhammad Faaiz Taufiq, and Hang Li. 2023b. Trustworthy llms: A survey and guideline for evaluating large language models' alignment. *arXiv preprint arXiv:2308.05374*.

OpenAI. 2024. Hello gpt-4o.

Yonatan Oren, Nicole Meister, Niladri Chatterji, Faisal Ladhak, and Tatsunori B Hashimoto. 2023. Proving test set contamination in black box language models. *arXiv preprint arXiv:2310.17623*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Oded Ovadia, Menachem Brief, Moshik Mishaeli, and Oren Elisha. 2023. Fine-tuning or retrieval? comparing knowledge injection in llms. *arXiv preprint arXiv:2312.05934*.

Oscar Sainz, Jon Ander Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. 2023. Nlp evaluation in trouble: On the need to measure llm data contamination for each benchmark. *arXiv preprint arXiv:2310.18018*.

Leonard Salewski, Stephan Alaniz, Isabel Rio-Torto, Eric Schulz, and Zeynep Akata. 2024. In-context impersonation reveals large language models' strengths and biases. *Advances in Neural Information Processing Systems*, 36.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. 2023. Detecting pretraining data from large language models. *arXiv preprint arXiv:2310.16789*.

RM Shiffrin. 2003. Chapter: Human memory: A proposed system and its control processes. *Spence, KW; Spence, JT. The psychology of learning and motivation. New York*.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2024. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*.

Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Benjamin Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Sreemanti Dey, Shubh-Agrawal, Sandeep Singh Sandha, Siddartha Venkat Naidu, Chinmay Hegde, Yann LeCun, Tom Goldstein, Willie Neiswanger, and Micah Goldblum. 2025. Livebench: A challenging, contamination-free LLM benchmark. In *The Thirteenth International Conference on Learning Representations*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Jingyang Zhang, Jingwei Sun, Eric Yeats, Yang Ouyang, Martin Kuo, Jianyi Zhang, Hao Frank Yang, and Hai Li. 2024. Min-k%++: Improved baseline for detecting pre-training data from large language models. *arXiv preprint arXiv:2404.02936*.

Yukun Zhao, Lingyong Yan, Weiwei Sun, Guoliang Xing, Chong Meng, Shuaiqiang Wang, Zhicong Cheng, Zhaochun Ren, and Dawei Yin. 2023. Knowing what llms do not know: A simple yet effective self-detection method. *arXiv preprint arXiv:2310.17918*.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. 2023a. Large language models are not robust multiple choice selectors. In *The Twelfth International Conference on Learning Representations*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023b. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. 2023. Agieval: A human-centric benchmark for evaluating foundation models. *arXiv preprint arXiv:2304.06364*.

Kun Zhou, Yutao Zhu, Zhipeng Chen, Wentong Chen, Wayne Xin Zhao, Xu Chen, Yankai Lin, Ji-Rong Wen, and Jiawei Han. 2023. Don't make your llm an evaluation benchmark cheater. *arXiv preprint arXiv:2311.01964*.

Kaijie Zhu, Jiaao Chen, Jindong Wang, Neil Zhenqiang Gong, Diyi Yang, and Xing Xie. 2023. Dyval: Dynamic evaluation of large language models for reasoning tasks. In *The Twelfth International Conference on Learning Representations*.

Kaijie Zhu, Jindong Wang, Qinlin Zhao, Ruochen Xu, and Xing Xie. 2024. Dyval 2: Dynamic evaluation of large language models by meta probing agents. *arXiv preprint arXiv:2402.14865*.

A Details of the extracting prompts and the extracted (memorized) MCQs

In this section, we introduce the details of the processed dataset and the prompts for extraction. MCQs from some subjects contain similar or identical options[3]. With the provided 5-shot prompt, options of MCQs from these subjects can be easily extracted, leading to a high False-positive ratio. In order to avoid the influence of the few-shot prompt on the option extraction, we eliminate MCQs in which any of the options have appeared twice in the dataset. After deduplication, we obtain 14,006 MCQs for evaluation. The extraction prompt and the detailed statistics are shown in the following text and Tab. 1.

Prompt template for extraction:

You are an expert of multiple choice questions of MMLU dataset. The following are multiple-choice questions (with answers) about [subject].

[examples]

[question]
Options:
A.

B Details of TrinEval

In this section, we introduce the details of the proposed TrinEval. The pseudo-code is shown in the Alg. 1. The prompts used are also shown below. Note that the potential data leakage is often caused by the data crawled on the Huggingface dataset site. Thus, we also provide the original text of MCQs in the format on the Huggingface dataset site to mimic the data contamination with in-context learning.

[3]E.g., the options of MCQs in the subject, moral_scenarios, are all identical (‘Wrong, Wrong’, ‘Wrong, Not wrong’, ‘Not wrong, Wrong’ and ‘Not wrong, Not wrong’).

Prompt template for pre-investigation on LLM Memorization w.r.t. Capability:

You are an expert of multiple choice questions of MMLU dataset. The following are multiple choice questions (with answers) about [subject].

[examples]

[question]
Options:
A. [content for option A]
B. [content for option B]
C. [content for option C]
D. [content for option D]

Answer:

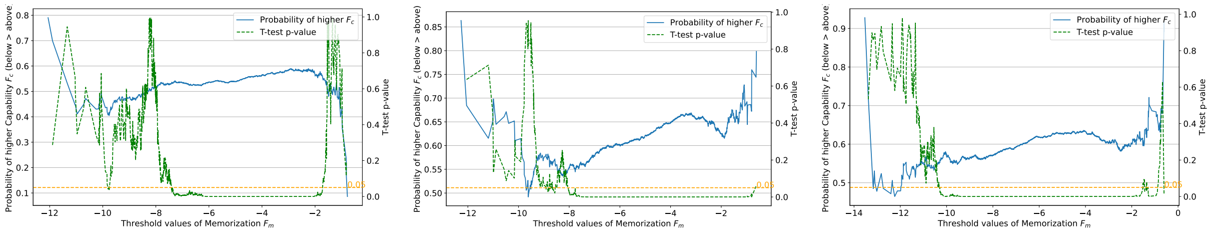
Model	Subset	Simple	Pro	MMLU
Llama	memorized	912	70	982
	non-mem.	6,548	6,476	13,024
	all	7,460	6,546	14,006
Mistral	memorized	879	36	915
	non-mem.	6,581	6,510	13,091
	all	7,460	6,546	14,006
Vicuna	memorized	893	16	909
	non-mem.	6,567	6,530	13,097
	all	7,460	6,546	14,006

Table 1: Statistics of memorized and non-memorized questions by Llama2-7B, Mistral-7B-v0.2, and Vicuna-v1.5-7B in MMLU.

C Detailed results of memorization v.s. capability

In this section, we exhibit the detailed results of the Q3. What does TrinEval reveal about the memorization v.s. the capability of LLMs. We reveal the ratio of MCQs within the upper right and lower left 2×2 and 3×3 squares as well as the Pearson correlations between the F_m and F_c of these MCQs. Our analysis reveals a tendency towards a negative correlation between the capabilities and memorization of LLMs shown in the Tab. 3.

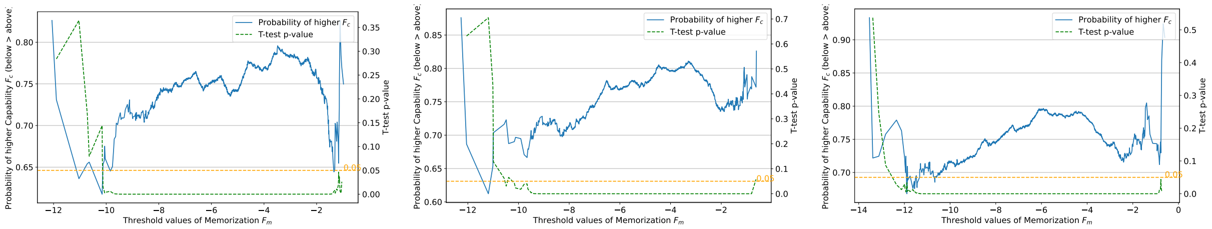
Further, inspired by the Precision-Recall Curve, we take each unique F_m of the qualified MCQs as the threshold to separate them as the Memorized and Capable MCQs. For each separation, we compute the probability of whether the F_c of a randomly selected Capable MCQ exceeds the F_c of a randomly selected Memorized MCQ and plot



(a) Probability and p-value with Llama2 based on GPT.

(b) Probability and p-value with Mistral based on GPT.

(c) Probability and p-value with Vicuna based on GPT.



(d) Probability and p-value with Llama2 based on GPT (21.59 % MCQs filtered).

(e) Probability and p-value with Mistral based on GPT (15.95 % MCQs filtered).

(f) Probability and p-value with Vicuna based on GPT (17.59 % MCQs filtered).

Figure 7: The over-performing probability curve and p-value curve with different F_m thresholds. In this figure, we take each unique F_m as the threshold to separate the qualified MCQs as the Memorized and Capable MCQs. We compute the probability of a randomly selected Capable MCQ's F_c exceeds a randomly selected Memorized MCQ's F_c under each threshold as the blue curve, and the green curve is the p-value of the T-test between the F_cs of the Capable MCQs and the Memorized MCQs.

Algorithm 1 MCQ reformulation by TrinEval

Input: Question x_Q, options x_O, and answer x_W of an MCQ.

Output: Reformulated question x_Q^R.

```
Preliminarily extract knowledge entity x_E, attribute x_A, and context x_C based on x_Q, x_O and x_W;
Initialize X_Q^R = x_E, x_A, x_C;
Validate the adequacy and necessity of the x_Q^R and give reasons x_L;
if x_Q^R matches the requirement then
    Return x_Q^R;
else
    Re-extract x'_E, x'_A, and x'_C by reflecting with x_E, x_A, x_C and x_L;
    Update x_Q^R = x'_E, x'_A, x'_C;
    Validate the adequacy and necessity of the x_Q^R and give reasons x_L;
    if x_Q^R matches the requirement then
        Return x_Q^R;
    else
        Discard the MCQ, return None;
    end if
end if
```

them as the blue curve. We also compute the T-test p-value between the F_cs of the Memorized MCQs and Capable MCQs as the green curve. The results are shown in Fig. 7. For the second row, we filter out the MCQs within the upper left and lower right 2×2 squares. From the figure, we observe that over a relatively long segment in the middle of the x-axis threshold range, the probability remains at a comparatively high value, while the p-value stays below 0.05. From this, we can conclude that F_m can distinguish between MCQs with high F_c and those with low F_c with a negative correlation at a high confidence level. This further supports that LLMs are potential rote learners, the more the LLMs memorize, the more poorly they perform.

D Use of AI assistants

ChatGPT[4] and Qwen[5] were used purely for the language refinement and polishment during the paper writing process. Any content generated with the LLMs was thoroughly reviewed and approved by the authors. No new content suggested by the AI assistants was used in the paper except the original expression from the authors.

[4] https://chat.openai.com/
[5] https://tongyi.aliyun.com/qianwen/

Prompt template for triplet extraction:

You are an expert of Knowledge Keyword extraction. Analyze and summarize the Question based on the given Fact corpus and extract the Knowledge Keyword, the Attribute and the Context (if necessary) within the Question.

Given a Fact corpus, a Question about the Fact corpus, and the Answer to the Question, analyze the Question corpus as well as the given Answer. Applying the provided steps, extract the Knowledge Keyword, the Attribute of the Knowledge Keyword and the necessary Context to obtain the key information of the Question, ensuring they are sufficient for answering the given Question and obtaining the given Answer.

 # Steps

*1. **Review the Fact corpus:** Read through the entire Fact corpus to understand the context.*

*2. **Identify the Question:** Focus on the given Question to capture which part of the Fact corpus it is asking about.*

*3. **Understand the Answer to the Question:** Compare the given Answer and the identified questioned part within the Fact corpus and understand why this answer was chosen.*

*4. **Write Step-by-Step Reasoning:***
- Identify the asked Knowledge Keyword in the Question that is the subject of the most information in the Fact corpus and the asked Question is about the information among.
- Determine the asked Attribute of the Knowledge Keyword in the Question, which can be used to infer the given Answer.
- Review the identified Knowledge Keyword and Attribute to confirm that only these two parts can be used to obtain the given Answer to the given Question. If not, extract all the necessary Context from the Question that makes it enough to obtain the given Answer to the given Question.

*5. **Determine Outcome:** Based on the reasoning, conclude and extract the Knowledge Keyword, the Attribute and the Context (if necessary) of the Question according to the Question corpus.*

 # Output Format

Provide the outcome in the following format:

*- **Step-by-Step Reasoning:** [Detailed reasoning here]*
*- **Knowledge Keyword:** [Extracted Knowledge Keyword here]*
*- **Attribute:** [Extracted Attribute of the Knowledge Keyword here]*
*- **Context:** [Extracted Context within the Question to make up for the Knowledge Keyword and the Attribute here if necessary]*

 # Examples

[examples]

 # Notes

- Strictly follow the format of the examples and give Knowledge Keywords, the Attribute and the Context (if necessary) anyway.
- The extracted Knowledge Keyword, Attribute and Context (if necessary) should be the original text within the Question and should not incorporate any phrases that cannot be exactly matched in the Question.
- Never include any information from the options of the multiple choice question, especially the content of the answer option.
- The extracted Knowledge Keyword, Attribute and Context (if necessary) should include all the necessary information only within the Question Corpus for answering the Question and obtaining the given Answer.

***Fact:** [question] [option content list] [subject] [answer option index][answer option ID]*

***Question:** [question]*

***Answer:** [content of the answer option]*

Prompt template for triplet validation & reflection:

You are an expert of [subject] and an advanced reasoning agent that can determine whether the given Knowledge Keyword, Attribute of the Knowledge Keyword and the Context present most of the necessary information of the Question for obtaining the given Answer. Suppose you have sufficient background knowledge about subj. Consider the given Knowledge Keyword, Attribute and the Context, then determine whether the given Answer can be directly obtained from them even without the Question.

 # Steps

*1. **Check the Semantic completeness:** Suppose you have sufficient background knowledge about [subject], and you can solve the given Question and obtain the given Answer. Read through the given Knowledge Keyword, Attribute, Context and the given Question. Check if the given Knowledge Keyword, Attribute, Context are the original text within the Question and contain the necessary queried information the Question itself provided (ignore the information the Question did not provided). If not so, check if the missed information is indeed incorporated in the Question (which is not acceptable, but if not, it is acceptable). Point out the information that is within the Question but they have missed. Then in a few sentences, diagnose the possible reason for failure or the phrasing discrepancy, and devise new, concise, high-level improvement suggestions to avoid the same failure.*

*2. **Check the Answer relevance:** Suppose you have sufficient background knowledge about subj, and you can solve the given Question and obtain the given Answer. Read through the given Knowledge Keyword, Attribute, Context and the given Question. Read through the given Knowledge Keyword, Attribute, Context and the given Answer. Check if the Answer can be directly inferred with the given Knowledge Keyword, Attribute and the Context without seeing the Question. If not so, check if the missed information is indeed incorporated in the Question (which is not acceptable, but if not, it is acceptable). Point out the information that is within the Question but they have missed. Then in a few sentences, diagnose the possible reason for failure or the phrasing discrepancy, and devise new, concise, high-level improvement suggestions to avoid the same failure.*

*3. **Check the Semantic Redundancy:** Read through the given Knowledge Keyword, Attribute, Context, the given Question and the given corresponding Answer. Check if the Answer can be directly matched within the given Knowledge Keyword, Attribute and the Context. Check if there are any unnecessary information within the given Knowledge Keyword, Attribute and the Context for obtaining the given Answer to the Question. If not so, point out what is redundant. Then in a few sentences, diagnose the possible reason for failure or the phrasing discrepancy, and devise new, concise, high-level improvement suggestions to avoid the same failure.*

 # Output Format

Provide the outcome in the following format:

*- **Step-by-Step Reasoning:** [Detailed reasoning here]*
*- **Verdict for the given Knowledge Keyword, Attribute and Context:** [Single verdict (Yes/No) here for whether the given Knowledge Keyword, Attribute and Context contain most of the asked information of the Question, can be used to infer the given Answer with only them without the whole Question, and do not contain redundant information for obtaining the given Answer.]*

 # Notes

- Do not deviate from the specified format. Do not generate anything else after the Verdict (only Yes/No) for the given Knowledge Keyword, Attribute and Context.
- Suppose you have sufficient background knowledge about subj, and you can solve the given Question and obtain the given Answer. For Semantic completeness and Answer relevance, it is acceptable to miss information that is also not incorporated in the Question.
- Provide a detailed explanation following the given steps before arriving at the verdict (Yes/No). Provide a final verdict (only Yes/No) in order at the end in the given format.

*- **Question:** [question]*
*- **Answer:** [answer]*

*- **Knowledge Keyword:** [extracted knowledge entity]*
*- **Attribute:** [extracted attribute]*
*- **Context:** [extracted context]*

Prompt template for the second round triplet extraction:

You are an advanced reasoning agent that can improve through self-reflection and an expert of Knowledge Keyword extraction. Analyze and summarize the Question based on the given Fact corpus and extract the Knowledge Keyword, the Attribute and the Context (if necessary) within the Question.

Given a Fact corpus, a Question about the Fact corpus, and the Answer to the Question, analyze the Question corpus as well as the given Answer. Applying the provided steps, extract the Knowledge Keyword, the Attribute of the Knowledge Keyword and the necessary Context to rephrase the Question, ensuring they are sufficient for answering the given Question and obtaining the given Answer.

 # Steps

*1. **Review the Fact corpus:** Read through the entire Fact corpus to understand the context.*

*2. **Identify the Question:** Focus on the given Question to capture which part of the Fact corpus it is asking about.*

*3. **Understand the Answer to the Question:** Compare the given Answer and the identified questioned part within the Fact corpus and understand why this answer was chosen.*

*4. **Write Step-by-Step Reasoning:***
- Identify the asked Knowledge Keyword in the Question that is the subject of the most information in the Fact corpus and the asked Question is about the information among.
- Determine the asked Attribute of the Knowledge Keyword in the Question, which can be used to infer the given Answer.
- Review the identified Knowledge Keyword and Attribute to confirm that only these two parts can be used to obtain the given Answer to the given Question. If not, extract all the necessary Context from the Question that makes it enough to obtain the given Answer to the given Question.

*5. **Determine Outcome:** Based on the reasoning, conclude and extract the Knowledge Keyword, the Attribute and the Context (if necessary) of the Question according to the Question corpus.*

 # Output Format

Provide the outcome in the following format:

*- **Step-by-Step Reasoning:** [Detailed reasoning here]*
*- **Knowledge Keyword:** [Extracted Knowledge Keyword here]*
*- **Attribute:** [Extracted Attribute of the Knowledge Keyword here]*
*- **Context:** [Extracted Context within the Question to make up for the Knowledge Keyword and the Attribute here if necessary]*

 # Examples

[examples]

You will be given a previous trial. You were unsuccessful in extracting the Knowledge Keyword, Attribute and the necessary that meet the requirements in the previous trial. Given the Reflection below, improve the process. The process is as follows:

 # Previous returns:

*- **Fact:** [question] [option content list] [subject] [answer option index][answer option ID]*

*- **Question:** [question]*

*- **Answer:** [answer option content]*

*- **Knowledge Keyword:** [extracted knowledge entity of the last trial]*

*- **Attribute:** [attribute of the last trial]*

*- **Context:** [context of the last trial]*

*- **Reflection:***
[rational of the last trial]

 # Notes

- Consider the Reflection given above. Improve the extraction of Knowledge Keyword, Attribute and Context (if necessary).
- Strictly follow the format of the examples and give Knowledge Keywords, the Attribute and the Context (if necessary) anyway.
- The extracted Knowledge Keyword should be phrases within the Question and should not incorporate any information of the Fact corpus or the given Answer that is not mentioned in the Question.
- The extracted Attribute and Context (if necessary) should only include information from the Question corpus. Never include information from the options of the multiple choice question, especially the content of the answer option.
- The extracted Knowledge Keyword, Attribute and Context (if necessary) should include all the necessary information only within the Question Corpus for answering the Question and obtaining the given Answer.

***Fact:** [question] [option content list] [subject] [answer option index][answer option ID]*

***Question:** [question]*

***Answer:** [content of the answer option]*

Original MCQ	**TrinEval MCQ**
You are an expert on multiple choice questions of [subject]. Analyze the given question and the given options. Determine the correct answer option to the question. Given a Question and the potential Answer options to the Question, analyze the Question as well as the given options. Generate the option ID of the correct option (answer). - **Question:** [question] - **Options:** A. [option A] B. [option B] C. [option C] D. [option D]	You are an expert on multiple choice questions of [subject]. Analyze the given Knowledge Entity, Attribute of the Knowledge Entity, the Context of a question, and the given options to the question. Determine the correct answer option to the question. The Knowledge Entity is the questioned subject of the question. The Attribute is the questioned attribute of the Knowledge Entity, and the Context is the necessary context information for answering the question. Given a set of Knowledge Entity, Attribute, and Context (which three are extracted as the key information from a question), and the potential Answer options to the Question, analyze the given Knowledge Entity, Attribute, Context as well as the options. Generate the option ID of the correct option (answer). - **Knowledge Entity:** [knwoledge entity] - **Attribute:** [attribute] - **Context:** [context] - **Options:** A. [option A] B. [option B] C. [option C] D. [option D]
Original MCQ Example	**TrinEval MCQ Example**
You are an expert on multiple choice questions of high school computer science. Analyze the given question and the given options. Determine the correct answer option to the question. Given a Question and the potential Answer options to the Question, analyze the Question as well as the given options. Generate the option ID of the correct option (answer). - **Question:** Which of the following is usually NOT represented in a subroutine's activation record frame for a stack-based programming language? - **Options:** A. Values of local variables B. A heap area C. The return address D. Stack pointer for the calling activation record	You are an expert on multiple choice questions of high school computer science. Analyze the given Knowledge Entity, Attribute of the Knowledge Entity, the Context of a question, and the given options to the question. Determine the correct answer option to the question. The Knowledge Entity is the questioned subject of the question. The Attribute is the questioned attribute of the Knowledge Entity, and the Context is the necessary context information for answering the question. Given a set of Knowledge Entity, Attribute, and Context (which three are extracted as the key information from a question), and the potential Answer options to the Question, analyze the given Knowledge Entity, Attribute, Context as well as the options. Generate the option ID of the correct option (answer). - **Knowledge Entity:** subroutine's activation record frame - **Attribute:** usually NOT represented - **Context:** for a stack-based programming language - **Options:** A. Values of local variables B. A heap area C. The return address D. Stack pointer for the calling activation record

Table 2: Template and an example of the Original MCQ template and the TrinEval MCQ template. [·] refers to the blank that should be filled according to the content of each MCQ.

LLMs	Dataset	2×2 squares		3×3 squres	
		Ratio (%)	Pearson correlation	Ratio (%)	Pearson correlation
Llama2-Qwen	All	38.57	-0.7755	74.17	-0.8124
	Simple	37.07	-0.7784	72.63	-0.8121
	Pro	38.66	-0.783	74.51	-0.8109
Llama2-GPT	All	35.22	-0.7835	71.04	-0.7924
	Simple	33.9	-0.777	69.62	-0.7919
	Pro	35.45	-0.7916	71.54	-0.7881
Mistral-Qwen	All	44.9	-0.8722	80.82	-0.8794
	Simple	38.47	-0.8494	74.04	-0.8271
	Pro	44.32	-0.8045	80.08	-0.8682
Mistral-GPT	All	40.37	-0.8042	76.58	-0.8736
	Simple	35.51	-0.8297	72.27	-0.8664
	Pro	38.52	-0.7103	74.91	-0.7969
Vicuna-Qwen	All	42.94	-0.8771	79.23	-0.8365
	Simple	37.86	-0.758	73.85	-0.7168
	Pro	42.01	-0.8609	77.86	-0.886
Vicuna-GPT	All	38.69	-0.8621	74.83	-0.8672
	Simple	34.77	-0.8096	70.71	-0.7775
	Pro	37.37	-0.7794	73.98	-0.8728

Table 3: The ratio and the Pearson-correlation between the F_c and F_m of the MCQs within the upper right and lower left 2×2 and 3×3 squares. For LLMs, 'Llama2-Qwen' refers that the F_c and F_m are calculated with Llama2 based on the Qwen-extracted triplet, and similarly hereinafter. For the Dataset column, 'All' stands for all the qualified MCQs after the triplet extraction, 'Pro' refers to the qualified MCQs that are the members of the mmlupro dataset while 'Simple' refers to the rest of the MCQs that are relatively easier.