Supervised Fine-Tuning of Large Language Models on Human Demonstrations Through the Lens of Memorization

Yubin Ge*
University of Illinois at Urbana-Champaign
yubinge2@illinois.edu

Devamanyu Hazarika
Amazon
dvhaz@amazon.com

Yang Liu
Amazon
yangliud@amazon.com

Mahdi Namazifar
Amazon
mahdinam@amazon.com

Abstract

In recent years, the field of natural language processing (NLP) has witnessed remarkable advancements driven by the development of large language models (LLMs). Various techniques, such as instruction tuning, have emerged as crucial approaches, enhancing LLMs' adaptability to new tasks guided by instructional prompts. Meanwhile, the phenomenon of memorization within LLMs has garnered considerable attention. In this work, we delve into memorization within LLMs during supervised fine-tuning on human demonstrations and find a distinct pattern marked by initial memorization growth followed by stabilization, with different degrees of memorization observed across various tasks. An intriguing observation is the increase in validation perplexity, typically indicative of overfitting, does not result in lower generation quality. We probe deeper by examining the entropy derived from LLM's output probabilities, uncovering a consistent trend of decreasing entropy throughout training under both nucleus sampling and teacher forcing scenarios. This implies growing confidence within the LLM in generating output, while such output may deviate from the expected ground truth. Building upon our investigation, we propose a novel Memorization-Based Curriculum (MBC) learning approach. We leverage likelihood as a proxy for measuring memorization and employ it to construct a data distribution for sampling instances with replacement during supervised fine-tuning, emphasizing data with lower degrees of memorization. Evaluations using GPT-4 as a judge demonstrate the effectiveness of MBC in fine-tuning LLMs on human demonstrations.

1 Introduction

Large language models (LLMs), such as PaLM [1] and Llama 2 [2], have brought significant progress in various tasks and applications during recent years. These models are typically pre-trained on massive text corpora, imbuing them with the capacity to predict the next token with unprecedented accuracy. Consequently, LLMs acquire versatile and general-purpose language representations that can be effectively applied to a wide spectrum of language understanding and generation tasks [3]. To harness this immense potential for transfer learning, various techniques have emerged to align LLM outputs with specific task requirements. Notably, *Supervised Fine-Tuning on human demonstrations* (which hereon we refer to as **SFT**) has emerged as a pivotal approach, involving the fine-tuning of

*Work done as an intern at Amazon

Workshop on Instruction Tuning and Instruction Following at NeurIPS 2023.

LLMs on diverse tasks that are characterized by demonstrations of human performance in carrying out these tasks [4]. This technique has proven to be particularly effective in enhancing the adaptability of LLMs, enabling them to excel at previously unseen tasks.

Besides, the phenomenon of memorization has emerged as a significant focal point within the field of LLMs as it relates to privacy and generalization, as underscored by prior research work [5, 6, 7]. These work have highlighted the inherent ability of LLMs to inadvertently memorize portions of their training data, potentially encompassing sensitive information, such as phone numbers and usernames [5, 8, 9]. Beyond the utilization of memorization for crafting attacks aimed at coercing LLMs into revealing training data, empirical investigations have also delved into the various factors that influence the memorization tendencies of LLMs during both pre-training and fine-tuning phases. Factors such as model sizes, learning rates and data duplication have been studied on this front [10, 7, 11]. Despite these efforts, the challenge of harnessing memorization to improve the training of LLMs on new data still remains an open frontier.

In this work, we focus on memorization within LLMs during the SFT process. To initiate our study, we perform an empirical analysis, using *opt-iml-30B* [12]. This analysis entails a quantification of memorization dynamics observed within the LLM during the course of SFT. Our findings reveal a discernible pattern characterized by an initial upsurge in memorization of training records, followed by a subsequent stabilization phase. Furthermore, our investigation uncovers the LLM's different propensity for memorization across various tasks (e.g., classification, summarization, QA, etc.). Additionally, in the course of SFT we observe increasing validation perplexity (going up to 40), which at training time would be interpreted as an indication of overfitting and loss of fluency at generation time. However, we observe that this increase corresponds to no negative change in the quality of LLM's generations. This observation aligns with the findings reported in [13]. Delving deeper into this phenomenon, we examine the entropy derived from the LLM's output probabilities. Our analysis reveals a consistent trend of decreasing entropy throughout the training process, under both nucleus sampling [14] and teacher forcing scenarios. This decrease in entropy implies that the LLM is progressively gaining confidence in its own generation throughout the course of SFT, even when its outputs diverge from the "ground truth" in the validation set (which are also human demonstrations). Given the open-ended nature of the vast majority of the tasks and the corresponding human demonstrations, these divergent outputs, while distinct from the ground truth, often remain valid. Consequently, the observed increase in validation perplexity may be attributed to this divergence, while the overall quality of generations in fact improves.

Building upon our investigation of memorization during the SFT process, we introduce a novel curriculum of introducing training records to LLMs during SFT. Given that the principal objective of SFT revolves around acquiring proficiency in understanding and following instructions, we posit that encouraging the LLM to focus on data it is less familiar with, as quantified by memorization, can prove beneficial. To realize this, we adopt perplexity as an approximation for measuring the model's memorization for each training record at regular intervals during SFT. Subsequently, instead of the typical uniform sampling without replacement of training records in SFT, we employ the perplexity values to construct a data distribution from which training records are sampled with replacement. This sampling strategy aims to guide the model to learn more from data that it has not memorized well, thereby enhancing its adaptability and performance. We call this approach *Memorization-Based Curriculum* (MBC) learning. To assess the efficacy of the proposed training record sampling strategy during SFT, we conduct an extensive evaluation by employing GPT-4 as a judge in pairwise comparisons between the outputs generated by two models: one trained with MBC and another by the common uniform sampling without replacement. The results demonstrate the superior performance achieved by the proposed training strategy.

Our contributions can be summarized as follows:

- We conduct an investigation into the memorization dynamics of LLMs during SFT and identify training records from various tasks that LLMs find easy and hard to memorize.

- We observe a noteworthy phenomenon where increasing validation perplexity during training (the SFT process) corresponds to little change in LLM's generation fluency. To shed light on this phenomenon, we delve into the entropy of LLM's output probability and present evidence that this phenomenon could be explained by the increasing "confidence" of the model in its generations during SFT.

- We introduce a simple but effective Memorization-Based Curriculum (MBC) learning approach, which prioritizes training records with low memorization through sampling with replacement.
- We evaluate our proposed training strategy, employing GPT-4 as a judge. The experimental results confirm the efficacy of our training strategy.

2 Memorization During Supervised Fine-Tuning

2.1 Memorization Measurement

The assessment of memorization in LLMs generally falls into two categories: *black-box* and *white-box* methods, depending on whether access to the model's internal information is permitted. Black-box memorization evaluations primarily focus on generation scenarios, assessing LLMs from the perspective of generating training data. These evaluations are often driven by privacy concerns and necessitate the use of prompts involving the model's training data. In line with this approach, k-eidetic memorization has been introduced to gauge whether a given string s can be extracted and if it appears in at most k instances within the training examples [5]. Furthermore, the notion of k-memorized strings is defined as those strings that LLMs are capable of generating when prompted with a context comprising k tokens from the training data [9]. In contrast, white-box methodologies typically leverage the predictions or internal information of LLMs in order to assess memorization. One option is counterfactual memorization, which follows the original definition of memorization in machine learning [15]. It quantifies memorization by measuring the difference in LLM predictions when a target data instance is included or excluded [16]. Furthermore, influence functions have been employed as an approximation to gauge memorization [17], and membership inference has been proposed as a reference-based attack for memorization assessment [6]. However, it is worth noting that these metrics tend to be computationally demanding, often necessitating either the training of multiple models or the computation of Hessian matrices. Hence we turn to the *exact memorization* metric, as it offers computational efficiency while still providing a robust measure of memorization [10]:

$$M(f) = \frac{\sum_{(x,y)\in C} \mathbb{1}\{\text{argmax}(f(x)) = y\}}{|C|}, \tag{1}$$

where f is the LLM we want to measure memorization of, $\mathbb{1}\{.\}$ is the indicator function, C denotes a set of contexts consisting of a list of tuples (x, y) where x is an input context and y is the index of the ground truth token. By this definition, we treat a context $c = (x, y) \in C$ as memorized if $\text{argmax}(f(x)) = y$, and the overall metric can be interpreted as the measurement of how often the argmax of the LLM's prediction matches the indices of the ground truth tokens.

2.2 Quantifying Memorization During Supervised Fine-Tuning

We perform our experiments on *opt-iml-30B* [12] and fine-tune it on a proprietary set of single turn general-purpose human demonstration data. The dataset is a set of $10,000$ single-turn dialogues between a human and an assistant, and each dialogue is labeled with a task category such as *Open Ended QA*. More details on the distribution of instances are provided in Appendix A.

Throughout the SFT, we maintain periodic checkpoints at every 5000 steps. To examine the extent of memorization exhibited by the LLM during SFT, we calculate *exact memorization* of all training records for all checkpoints. For each training record, we separately measure the memorization of the complete dialogue (the human instruction and the assistant response) and the assistant response only. The results of our experiments are depicted in Figure 1. These results reveal that the LLM rapidly memorizes the training data during the initial stages of SFT, with this memorization trend gradually decelerating as the training progresses. This analysis also shows that the degree of memorization of only assistant responses given the human instructions is higher than that of both human instructions and assistant responses (as seen in Figure 1); which is expected.

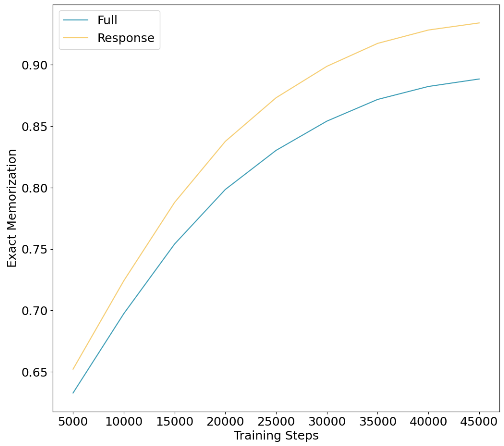

Figure 1: The memorization of LLM during supervised fine-tuning.

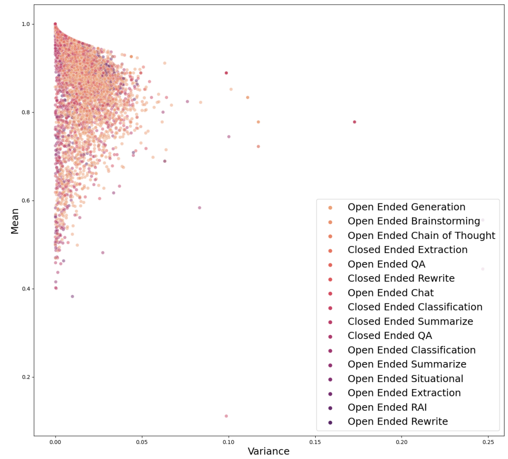

Figure 2: The instance-level memorization for all training data LLM during supervised fine-tuning.

2.3 Instance-Level Memorization Dynamics

Motivated by the concept of Data Maps [18], we extend our investigation to instance-level memorization dynamics during SFT. Specifically, we first calculate the memorization of each training record at various checkpoints throughout the training process, then derive both the mean and variance of memorization values across all checkpoints. These values are depicted in Figure 2 where each dot corresponds to one training record and axes x and y represent variance and mean of the exact memorization metric, respectively. From the figure, we observe a prominent clustering of instances in the upper-left corner. This specific clustering is indicative of instances characterized by high mean memorization values and low variance. Due to this consistent and robust memorization trend throughout the training process, these training records could be considered as *easy-to-memorize*. Upon closer examination, we observe that these instances predominantly pertain to relatively straightforward question-answering tasks, often involving factual knowledge. Conversely, in the bottom-left corner, we identify a subset of instances marked by both low variance and low mean memorization values. This subset is noteworthy for its consistently low memorization levels during the entirety of the SFT process, hence they can be considered as *hard-to-memorize*. These instances predominantly align with more complex natural language generation tasks, such as summarization, as well as challenging question-answering scenarios that necessitate intricate reasoning abilities. We show some *easy-to-memorize* and *hard-to-memorize* examples in Appendix B.

2.4 Memorization vs Generalization

In assessing the performance of a trained LLM, one primary objective is to evaluate its generalization ability to unseen data. To achieve this, we follow [6] to use validation perplexity and calculate perplexity based on a held-out validation set across all saved checkpoints. We show the results in Figure 3. An obvious trend observed during training is the general increase in validation perplexity, typically associated with overfitting and, consequently, is expected to demonstrate diminished model performance. However, upon conducting a thorough manual inspection of the outputs generated by various intermediate checkpoints, it becomes apparent that there exists little discrepancy among them in terms of the generation quality which for pre-trained LLMs is highly and inversely correlated with validation perplexity. This intriguing finding is congruent with the results reported in [13], where the authors conducted a comprehensive evaluation by enlisting ChatGPT as a judge to assess the generation quality of intermediate models exhibiting increasing validation perplexity. The outcomes of this evaluation indicate a positive correlation between increasing validation perplexity and enhanced generation quality. This correlation raises a fundamental question: *Why does the increase of validation perplexity coincide with improved generation quality?*

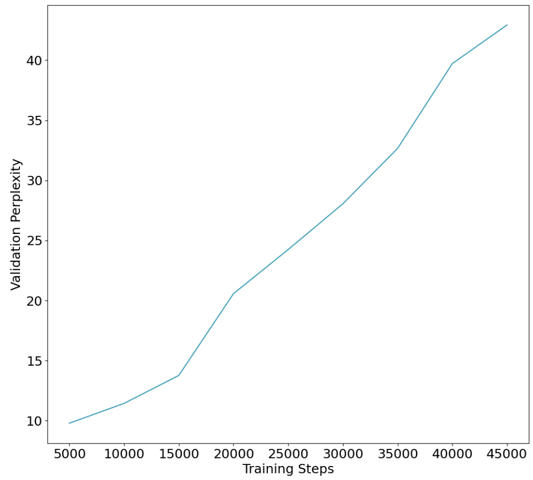

Figure 3: The validation perplexity of LLM during supervised fine-tuning.

Checkpoints	Nucleus Sampling	Teacher Forcing
5000 step	0.3610	0.7675
45000 step	0.1670	0.3638

Table 1: The entropy of LLM's output probability at the first and last checkpoints under two scenarios.

To explore the answer to this question, we look into the mean entropy of the LLM's output probabilities on the validation set, focusing on the initial and final checkpoints under two distinct generation scenarios, namely *nucleus sampling* [14] and *teacher forcing*. Nucleus sampling represents the generation scenario during inference, while teacher forcing is employed to gauge alignment with the ground truth. The results of this analysis are presented in Table 1. Notably, we observe that the entropy under nucleus sampling consistently remains lower than the entropy under the teacher forcing scenario. Furthermore, the entropy values exhibit a diminishing trend from the beginning of training to the end. Note that most of the training and validation records for SFT are human demonstrations of various open-ended tasks which means that there is no single right response for a given prompt in these human demonstrations. The results in Table 1 show that the likelihood of the LLM generating the exact sequence of responses as in the validation set decreases during the SFT process. This phenomenon may be interpreted as an indication of the LLM's increasing confidence in its own generations, even though its output may deviate from the "ground truth", which in this case are human-created responses to some open-ended tasks. As a result, validation perplexity may not be reliable enough as a metric in evaluating and selecting model checkpoints during SFT, and other metrics that truly evaluate the performance of LLMs are required for this purpose.

3 Memorization-Based Curriculum (MBC) Learning

Building upon our exploration of LLMs' memorization patterns during SFT, our next objective is to devise an efficient approach to harness LLM's memorization for enhancing the training process and attaining better performance. We seek to utilize the varying degrees of memorization exhibited by LLMs across different data records during training, thus encouraging the model to allocate distinct levels of attention to different data records. Our motivation stems from the observation of human learning paradigms, where the acquisition of new knowledge often heavily relies on pre-existing knowledge and typically starts from the knowledge that individuals are more familiar with [19]. Guided by this insight, we posit that, in the context of SFT, a substantial portion of the training data may not constitute entirely new information for LLMs, such as factual knowledge in question-answering tasks, despite being unseen during the model's training. We propose to quantify such kind of familiarity with data through the concept of memorization, wherein a high degree of memorization suggests that LLMs possess a robust understanding of the data, while low memorization indicates the need for LLMs to focus more on these data records during the SFT process.

We design a new training strategy grounded in curriculum learning principles, which leverages the concept of memorization in sampling training batches during SFT. Specifically, at every k training steps, we perform the calculation of LLM's memorization of the entire training dataset. Based on the memorization values of training records, we create a multinomial discrete probability distribution where the probability associated with each training record is inversely proportional to its memorization value. During SFT, training batches are sampled with replacement from this distribution. We use perplexity as a proxy for measuring LLM's memorization of training records, rather than relying on exact memorization. The rationale behind this decision lies in the strong correlation between these two metrics, both of which are substantially dependent on LLM's output probabilities. In this context, lower perplexity values correspond to higher memorization levels, whereas higher perplexity values signify lower memorization. We refer to this approach in creating training batches during SFT as Memorization-Based Curriculum (MBC) Learning and the overall training strategy of MBC is shown in Algorithm 1.

Algorithm 1 MBC Learning for SFT

```
Require: Human demonstrations D, pre-trained model M, a fixed step interval k
  for every training step i do
    if i%k == 0 then
      Calculate perplexity using M for all d ∈ D
      Update distribution w according to the calculated perplexity
    Sample instances with replacement based on w to form a batch b
    Perform a training step on M with the batch data b
```

Note that MBC is significantly different from the traditional sampling of batches from a uniform distribution over the training records without replacement. In MBC there is no concept of training epoch and one training record may appear in the same batch more than once. From one perspective MBC could be viewed as a self-regulating process that tries to bring a balance to the memorization level across all training records. In other words, if at a given stage of SFT the memorization value of a training record is low, MBC increases the likelihood of that training record being sampled. Once that record is sampled enough times that the LLM has a higher memorization of it, MBC reduces that likelihood, in favor of other training records with lower memorization.

It should also be noted that MBC diverges from the conventional curriculum learning [20]. In the traditional curriculum learning paradigm, the emphasis is on quantifying the difficulty levels associated with data instances and initiating the training process with easier samples, subsequently progressing towards more intricate ones. In contrast, MBC samples harder training records (as characterized by low memorization) more frequently than easier ones. The more a training record is sampled, the more it is memorized and, as a result, the less it is likely to be sampled again in future batches.

4 Experiments

4.1 Experimental Setup

We choose *gpt-neo-2.7B* as the pre-trained LLM [21] to conduct our experiments and perform SFT using a proprietary dataset as detailed in Section 2.2. We chose a smaller model for these experiments compared to our previous experiments that were done on *opt-iml-30B* due to the higher computational cost of MBC that is associated with calculating perplexities. For SFT we use PyTorch [22] and Huggingface Transformers [23] libraries. The SFT is done over 15 epochs, uses AdamW optimizer [24], with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and has no weight decay. With no warmup steps, we try two learning rates, $1e-5$ and $5e-5$, and both undergo linear decay. During the inference phase, for each prompt, we generate a single response, using nucleus sampling [14] with parameters $p = 0.9$ and a temperature of $\tau = 0.7$. Furthermore, we apply a repetition penalty to mitigate the recurrence of previously generated tokens, with a value of 1.2 [25]. The maximum allowable length for newly generated tokens is constrained to 256. It is worth noting that, as previously demonstrated in Section 2.4, validation perplexity does not exhibit a significant correlation with generation quality. Consequently, we employ a manual selection process for identifying the best checkpoints as in [13]. This process includes the utilization of a 50-example development set, where we manually assess and compare the outputs generated by different model checkpoints. Subsequently, we select the checkpoint that consistently exhibits the highest generation quality.

We use GPT-4 to serve as a judge for conducting pairwise comparisons between the outputs generated by two LLMs, one leverages MBC during SFT, and the other follows the conventional random sampling method during SFT as a baseline. The evaluation process involves presenting the LLM judge with a prompt from a validation set along with two responses, one from each of the two trained LLMs. The GPT-4's role is to determine which of the two responses is superior or declare a tie between them. To mitigate the potential introduction of biases associated with using GPT-4 as a judge, such as verbosity bias and position bias, we incorporate additional instructions, as outlined in [26]. The specific prompt utilized for this evaluation is provided in Figure 4.

```
Please act as an impartial judge and evaluate the quality of the responses provided by two AI assistants to the user instruction displayed below. You should choose the assistant that follows the user's instruction and responds to the user's instruction better. Your evaluation should consider factors such as the helpfulness, relevance, accuracy, depth, creativity, and level of detail of their responses. Begin your evaluation by comparing the two responses and provide a short explanation. Avoid any position biases and ensure that the order in which the responses were presented does not influence your decision. Do not allow the length of the responses to influence your evaluation. Do not favor certain names of the assistants. Be as objective as possible. After providing your explanation, output your final verdict by strictly following this format: "[[A]]" if assistant A is better, "[[B]]" if assistant B is better, and "[[C]]" for a tie.

[User Instruction]
{instruction}

[The Start of Assistant A's Response]
{response_a}
[The End of Assistant A's Response]

[The Start of Assistant B's Response]
{response_b}
[The End of Assistant B's Response]
```

Figure 4: The prompt for pairwise comparison using GPT-4.

4.2 Experimental Results

The win rates, indicating the frequency at which MBC approach outperforms the baseline judged by GPT-4, are presented in Figure 5. These results illustrate the efficacy of our proposed training strategy in enhancing the generation quality of the LLM. Specifically, the win rate consistently surpasses the loss rate, with a notable contrast of 41.0% versus 26.6% for the win rate versus loss rate when the learning rate is set at $1e-5$. Similarly, at a learning rate of $5e-5$, the win rate stands at 48.3% in comparison to the loss rate of 27.8%. We posit that this improvement is attributable to the strategic utilization of memorization within MBC. This approach encourages the LLM to pay more attention to training records that it does not "remember" well, while often bypassing data instances that it has already memorized. Besides, this may prevent the LLM from learning well-memorized data, which is probable to cause LLMs to overfit.

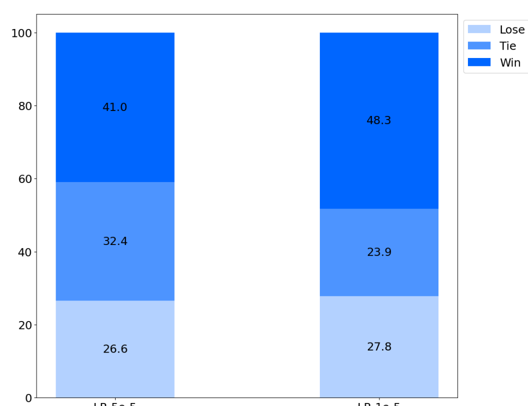

Figure 5: The win rate of our trained model over the baseline under different learning rates.

4.3 Quantifying Memorization Dynamics

We also employ exact memorization to assess the memorization of the training data for the trained LLM with MBC and the baseline, as shown in Figure 6. Notably, when the learning rate is set to $5e-5$, we observe an interesting phenomenon: the memorization for the baseline initially increases but subsequently exhibits a decline, with an associated increase in training loss. We attribute this behavior

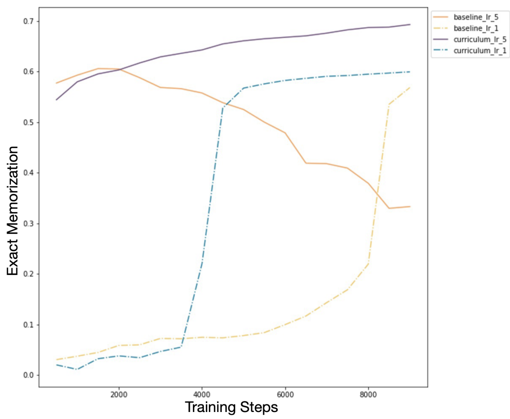

Figure 6: The memorization dynamics observed in the LM trained with our proposed strategy and the baseline model across different learning rates.

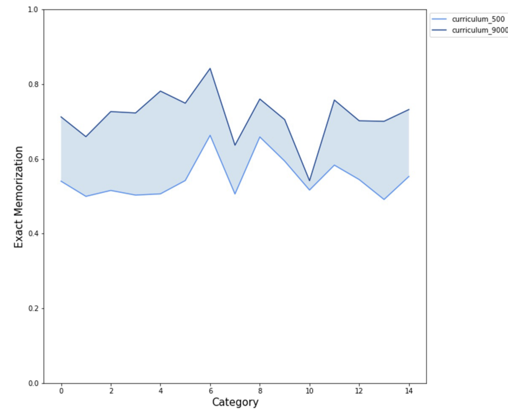

Figure 7: Memorization improvements achieved by the LM trained with MBC across various task categories.

to the learning rate potentially being excessively high. A high learning rate can cause significant momentum accumulation when using AdamW, which may lead to escaping local minima during SFT. In contrast, when training the same LLM with MBC, we observe robustness to the influence of a high learning rate, ultimately resulting in superior performance compared to the baseline. On the other hand, when the learning rate is set to $1e-5$, the baseline exhibits the common training trajectory characterized by a consistent reduction in training loss. However, MBC still shows higher memorization than the baseline and the fast memorization of training data happens earlier.

Additionally, in the case that the learning rate is $1e-5$, both models exhibit similar patterns in memorization dynamics during SFT. Initially, there is a gradual increase in memorization, followed by a marked surge within a short time frame, and a final slowdown in memorization. This observed pattern diverges a little from the memorization dynamics observed during SFT on the *opt-iml-30B*, as shown in Section 2.2. This might be due to the size of the models or other factors that need to be further studied. That being said, the overarching trend appears to be one of incremental memorization that eventually stabilizes and converges.

We conduct an analysis of memorization dynamics across distinct task categories based on the SFT of LLM trained using MBC with the learning rate $1e-5$, as illustrated in Figure 7. In this figure, the light blue line corresponds to the memorization values calculated based on the data from the initial saved checkpoint across different task categories, while the dark blue line represents the memorization derived from the final checkpoint. From the figure, it becomes evident that the memorization levels across all task categories exhibit varying degrees of improvement. Significantly, specific tasks that were initially classified among the challenging "hard-to-memorize" categories have exhibited remarkable improvements in memorization, eventually rising to the status of being among the more memorized categories. For instance, the category designated as index 4 in the figure illustrates this noteworthy transition. Moreover, the figure highlights that certain tasks (e.g., the category indexed 10) consistently maintain lower memorization compared to others throughout the training, even when subjected to increased sampling frequency. This observation pertains particularly to tasks that inherently present greater complexities, such as text summarization.

4.4 Sampling Frequency Comparison

To study the difference in the behavior of MBC compared to the conventional batching approach, we have constructed the sampling frequency distribution across all of the training data, as depicted in Figure 8. In this figure, the x-axis corresponds to all training records, while the y-axis shows the frequency at which each individual instance has been sampled during the SFT process. With the number of training epochs set to 15, a divergence in sampling frequency emerges when contrasting MBC with the uniformly random without replacement approach. Specifically, under the latter approach, all training data instances share the same frequency of sampling, 15. However, MBC results in a significantly different sampling frequency distribution. Upon closer examination of the figure, it

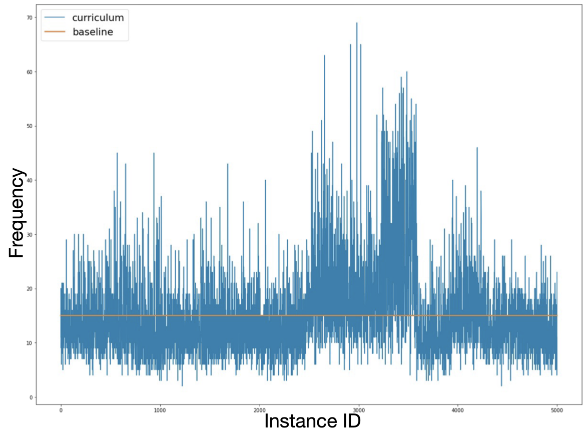

Figure 8: The sampling frequency of training data when training LM using the proposed MBC.

becomes evident that certain instances have been sampled considerably more frequently than others. We attribute this discrepancy to the fact that certain instances are more challenging to memorize and, hence need to be sampled more frequently during SFT.

5 Related Work

Memorization in Language Models: The inadvertent memorization phenomenon poses a well-recognized challenge for language models, as depicted in prior research [27, 28]. This vulnerability to memorization renders language models susceptible to extraction attacks [5, 29] and membership inference attacks [30, 31]. It is noteworthy that great efforts have been made to address and mitigate these vulnerabilities [32, 33]. Recent research, however, has advanced the argument that memorization is not intrinsically detrimental and, in fact, can be of great significance for certain forms of generalization, such as those encountered in question-answering tasks [34, 35, 36]. Additionally, memorization empowers language models to encode substantial reservoirs of worldly or factual knowledge [37, 38, 39]. Furthermore, a growing body of research is dedicated to understanding and leveraging the fundamental properties of memorization within language models [9, 16, 10, 40]. This collective research effort contributes to a deeper understanding of the intricate dynamics of memorization within language models. In this context, our work aligns with this research trajectory, with a specific focus on the role of memorization in the instruction tuning of LLMs.

SFT of Language Models: Prior research has extensively explored the concept of instruction tuning in the context of language models, revealing its potential to facilitate zero-shot task generalization [41, 42, 43, 44]. Various attempts have aimed to enhance instruction tuning by enabling cross-lingual generalization [45], improving label generalization capabilities [46], demonstrating the feasibility of lifelong learning through continual learning [47], and training modular expert language models [48]. This work focuses on memorization dynamics during the process of fine-tuning a language model. In doing so, our research not only elucidates empirical patterns of memorization within the domain of instruction tuning but also introduces a novel approach rooted in memorization-based curriculum learning, contributing to the enhancement of LLMs' performance.

6 Conclusion

In this work, we first investigate the memorization within LLMs during the SFT process. Our empirical analysis has unveiled a pattern characterized by an initial surge in memorization followed by a subsequent stabilization, and uncovered variations in the LLM's memorization across different tasks. We have also observed an increase in validation perplexity corresponding to little change in the LLM's generation quality. To delve into this, we examine the entropy of the LLM's output probabilities and find a consistent trend of diminishing entropy throughout the training process, under both nucleus sampling and teacher forcing scenarios. This result suggests that the LLM is progressively gaining confidence in its own generations, even when its output diverges from the ground truth in human demonstrations. Building upon our investigation, we have introduced a simple yet effective approach called Memorization-Based Curriculum (MBC) learning. Specifically, we measure the model's memorization of each instance at every k steps during SFT and then leverage these memorization values to construct a data distribution from which instances are sampled with replacement. This sampling strategy is intended to guide the model to learn more from data it has not memorized well. To rigorously assess the efficacy of our proposed training strategy, we conduct evaluations, employing GPT-4 as a judge in pairwise comparisons between the outputs generated by two models: one trained with MBC and another using common random sampling. The experimental results demonstrate the superior performance achieved by the LLM trained using MBC.

References

[1] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

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

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[4] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

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

[6] Fatemehsadat Mireshghallah, Archit Uniyal, Tianhao Wang, David K Evans, and Taylor Berg-Kirkpatrick. An empirical analysis of memorization in fine-tuned autoregressive language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1816–1826, 2022.

[7] Michael Tänzer, Sebastian Ruder, and Marek Rei. Memorisation versus generalisation in pre-trained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7564–7578, 2022.

[8] Vikas Raunak and Arul Menezes. Finding memo: Extractive memorization in constrained sequence generation tasks. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5153–5162, 2022.

[9] Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations*, 2022.

[10] Kushal Tirumala, Aram Markosyan, Luke Zettlemoyer, and Armen Aghajanyan. Memorization without overfitting: Analyzing the training dynamics of large language models. *Advances in Neural Information Processing Systems*, 35:38274–38290, 2022.

[11] Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR, 2023.

[12] Srinivasan Iyer, Xi Victoria Lin, Ramakanth Pasunuru, Todor Mihaylov, Daniel Simig, Ping Yu, Kurt Shuster, Tianlu Wang, Qing Liu, Punit Singh Koura, et al. Opt-iml: Scaling language model instruction meta learning through the lens of generalization. *arXiv preprint arXiv:2212.12017*, 2022.

[13] Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. Lima: Less is more for alignment. *arXiv preprint arXiv:2305.11206*, 2023.

[14] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2019.

[15] Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. *Advances in Neural Information Processing Systems*, 33:2881–2891, 2020.

[16] Chiyuan Zhang, Daphne Ippolito, Katherine Lee, Matthew Jagielski, Florian Tramèr, and Nicholas Carlini. Counterfactual memorization in neural language models. *arXiv preprint arXiv:2112.12938*, 2021.

[17] Xiaosen Zheng and Jing Jiang. An empirical study of memorization in nlp. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6265–6278, 2022.

[18] Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A Smith, and Yejin Choi. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9275–9293, 2020.

[19] Jeanne Ellis Ormrod. *Human learning*. Merrill Upper Saddle River, NJ, 1999.

[20] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009.

[21] Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. Gpt-neo: Large scale autoregressive language modeling with mesh-tensorflow. *If you use this software, please cite it using these metadata*, 58, 2021.

[22] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[23] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP)*, October 2020.

[24] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2018.

[25] Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*, 2019.

[26] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*, 2023.

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

[28] Congzheng Song and Vitaly Shmatikov. Auditing data provenance in text-generation models. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 196–206, 2019.

[29] Aleena Thomas, David Ifeoluwa Adelani, Ali Davody, Aditya Mogadala, and Dietrich Klakow. Investigating the impact of pre-trained word embeddings on memorization in neural networks. In *Text, Speech, and Dialogue: 23rd International Conference, TSD 2020, Brno, Czech Republic, September 8–11, 2020, Proceedings 23*, pages 273–281. Springer, 2020.

[30] Sorami Hisamoto, Matt Post, and Kevin Duh. Membership inference attacks on sequence-to-sequence models. *arXiv preprint arXiv:1904.05506*, 2019.

[31] Fatemehsadat Mireshghallah, Kartik Goyal, Archit Uniyal, Taylor Berg-Kirkpatrick, and Reza Shokri. Quantifying privacy risks of masked language models using membership inference attacks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8332–8347, 2022.

[32] Xuechen Li, Florian Tramer, Percy Liang, and Tatsunori Hashimoto. Large language models can be strong differentially private learners. In *International Conference on Learning Representations*, 2021.

[33] Om Thakkar, Swaroop Ramaswamy, Rajiv Mathews, and Françoise Beaufays. Understanding unintended memorization in federated learning. *arXiv preprint arXiv:2006.07490*, 2020.

[34] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pages 2206–2240. PMLR, 2022.

[35] Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models. In *International Conference on Learning Representations*, 2019.

[36] Yi Tay, Vinh Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Gupta, et al. Transformer memory as a differentiable search index. *Advances in Neural Information Processing Systems*, 35:21831–21843, 2022.

[37] Badr AlKhamissi, Millicent Li, Asli Celikyilmaz, Mona Diab, and Marjan Ghazvininejad. A review on language models as knowledge bases. *arXiv preprint arXiv:2204.06031*, 2022.

[38] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR, 2020.

[39] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, 2019.

[40] R Thomas McCoy, Paul Smolensky, Tal Linzen, Jianfeng Gao, and Asli Celikyilmaz. How much do language models copy from their training data? evaluating linguistic novelty in text generation using raven. *Transactions of the Association for Computational Linguistics*, 11:652–670, 2023.

[41] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2021.

[42] Vamsi Aribandi, Yi Tay, Tal Schuster, Jinfeng Rao, Huaixiu Steven Zheng, Sanket Vaibhav Mehta, Honglei Zhuang, Vinh Q Tran, Dara Bahri, Jianmo Ni, et al. Ext5: Towards extreme multi-task scaling for transfer learning. In *International Conference on Learning Representations*, 2021.

[43] Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. Multitask prompted training enables zero-shot task generalization. In *ICLR 2022-Tenth International Conference on Learning Representations*, 2022.

[44] Seungone Kim, Se June Joo, Doyoung Kim, Joel Jang, Seonghyeon Ye, Jamin Shin, and Minjoon Seo. The cot collection: Improving zero-shot and few-shot learning of language models via chain-of-thought fine-tuning. *arXiv preprint arXiv:2305.14045*, 2023.

[45] Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, et al. Crosslingual generalization through multitask finetuning. *arXiv preprint arXiv:2211.01786*, 2022.

[46] Seonghyeon Ye, Doyoung Kim, Joel Jang, Joongbo Shin, and Minjoon Seo. Guess the instruction! making language models stronger zero-shot learners. *arXiv preprint arXiv:2210.02969*, 2022.

[47] Thomas Scialom, Tuhin Chakrabarty, and Smaranda Muresan. Continual-t0: Progressively instructing 50+ tasks to language models without forgetting. *arXiv preprint arXiv:2205.12393*, 2022.

[48] Joel Jang, Seungone Kim, Seonghyeon Ye, Doyoung Kim, Lajanugen Logeswaran, Moontae Lee, Kyungjae Lee, and Minjoon Seo. Exploring the benefits of training expert language models over instruction tuning. *arXiv preprint arXiv:2302.03202*, 2023.

A Dataset Statistics

Category	Mean Length	Frequency
Open Ended Summarization	815.93	0.49%
Open Ended Extraction	450.79	0.53%
Open Ended Rewrite	441.40	0.38%
Closed Ended Summarization	319.85	6.65%
Open Ended Situational	252.65	0.03%
Open Ended Chat	165.12	7.75%
Open Ended Generation	160.35	34.75%
Closed Ended Extraction	139.44	2.67%
Open Ended Brainstorming	132.98	7.96%
Closed Ended Classification	121.92	2.03%
Open Ended Classification	119.20	3.02%
Open Ended Chain of Thought	117.94	7.40%
Closed Ended QA	109.63	9.26%
Open Ended QA	105.82	7.98%
Closed Ended Rewrite	103.03	3.67%
Open Ended Responsible AI	65.44	5.53%

Table 2: The statistics of our collected dataset.

B *Easy-to-memorize* and *Hard-to-memorize* Examples

Task	Text
Open Ended QA	**Human**: In which country, Niagara Falls are located?\n\n **Assistant**: Niagara Falls are located in North America.
Closed Ended Classification	**Human**: What genre of music do these artists belong to? A) Rock, B) Jazz, C) Pop, D) Tejano, E) Dubstep, F) K-Pop\n1. BTS\n2. EXO\n3. Red Velvet\n\n **Assistant**: The artists belong to the K-Pop genre.
Closed Ended Classification	**Human**: What type of restaurant is McDonald's?\n\nA. Caf\u00e9\nB. Fast Food\nC. Casual Dining\nD. Fine Dining\n\n **Assistant**: B. McDonald's is a fast food restaurant.

Figure 9: *easy-to-memorize* examples.

Task	Text
Open Ended Generation	**Human**: Write a story about a lovely woman who is famous for her beauty in her village.\n\n **Assistant**: A lovely woman who is famous for her beauty in her village because she comes from Europe and wants to live with the people of the village She wants to live the life of a villager and knows the difficulties that they face in their daily lives. One day one of the village boys, whose name is Joy, came to her and asked her why she had come here to live with them, as her beauty would fade in this village because of the pollutants and unhygienic water they drank. This would affect her health as well as her beauty. So the beautiful girl told him that his health was also important and that the village people's health was also important, just like hers, so why were they leaving here? So the boy replied that they are leaving because they don't have any other option. To go anywhere. The girl informed him that she was writing a book. That is why she came here--to know about the difficulties the village people are facing so that she can help them and improve their lives. The boy informed her that nobody knows her name. Everybody calls her the beautiful girl. As a result, she laughed at his words. And tell her that her name is Janne. The girl informed Joy that he would assist her. You know about the lives of the village people, so it will help her to write about them. The boy replied that he would help her. He also wants to help the villagers because they are living a very bad life and nobody takes care of them. She came here to help them, and he will definitely help her. And he also thanked her that she thinks about them; Jane replied you all are as important as we are. I will do everything in my power to improve your standard of living.

Task	Text
Closed Ended Classification	**Human**: Which category best describes the following paragraph?\n1) Entertainment\n2) History\n3) Sports\nThe Statue of Liberty is a 305-foot statue located off the coast of New York City on Liberty Island in Upper New York Bay. The statue represents liberty in the form of a woman. She has a torch in her raised right hand and a tablet clutched in her left. Between 1875 and 1884, the Statue of Liberty was built in France. In 1885, it was disassembled and shipped to New York City.\n\n **Assistant**: History

Figure 10: *hard-to-memorize* examples.