Early Detection and Reduction of Memorization for Domain Adaptation and Instruction Tuning

Dean L. Slack **Noura Al Moubayed**
Durham University, UK
{dean.l.slack,noura.al-moubayed}@durham.ac.uk

Abstract

Although large language models excel across many tasks, they can memorize training data and thereby expose private or copyrighted text. Most defenses target the pre-training stage, leaving memorization during fine-tuning–especially for domain adaptation and instruction tuning–poorly understood. We fine-tune Pythia, Llama3, and Mistral models spanning 1.4B–70B parameters on common evaluation datasets and track verbatim memorization throughout training. We find that memorization increases dramatically in the first few epochs, often significantly before either validation perplexity or evaluation performance is optimized. We use a simple but effective n-gram memorization score which reliably precedes verbatim memorization; using it as an early-stopping criterion mitigates memorization with minimal performance loss. Further, we introduce an n-gram–aware loss regularizer and show that it reduces memorization across all model families tested by up to 40% while minimizing evaluation performance trade-offs when compared to an existing memorization mitigation strategy. These results yield practical, scalable insights into memorization dynamics during language model fine-tuning.

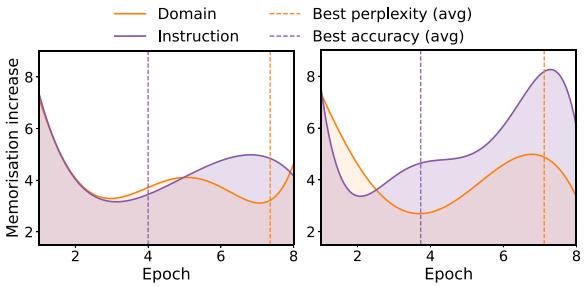
1 Introduction

Large Language Models (LLMs) have become increasingly powerful, achieving remarkable performance across diverse tasks and domains as they scale from millions to trillions of parameters (Brown et al., 2020; Fedus et al., 2022). Transformer-based architectures have propelled significant advancements in Natural Language Processing (NLP), setting new benchmarks in various applications (Vaswani et al., 2017; Devlin et al., 2019; Radford et al., 2019). However, alongside these achievements, concerns have emerged about the extent to which these models memorize their training data rather than genuinely understanding and generalizing linguistic patterns (Khandelwal et al., 2020; Tänzer et al., 2022). Memorization in LLMs poses serious privacy and security risks, where models have been shown to reproduce verbatim passages from their training data, including sensitive personal information and copyrighted materials (Patil et al., 2024). This not only presents ethical challenges and potential legal issues, but can potentially undermine user consent when deploying models in a generative environment. Training data extraction attacks (Carlini et al., 2021) demonstrate that adversaries can recover spans of pretraining sample data, highlighting the practical threat of generative model deployment.

Most existing mitigation efforts focus on unlearning strategies and regularization techniques applied during pre-training (Carlini et al., 2023; Cheng et al., 2023). While valuable, these approaches often lack scalability and are not easily deployable in practice, especially given the immense computational resources required to retrain large models or apply differential privacy methods (Anil et al., 2022). Moreover, on large datasets, exhaustive extraction tests are infeasible, making it challenging to assess and mitigate memorization effectively. Fine-tuning pre-trained LLMs on domain-specific and instruction-specific data is a common practice to adapt models to new domains and tasks, often utilising datasets with private and sensitive information. Despite this widespread application, there is a gap in understanding how fine-tuning for domain adaptation or instruction tuning impacts memorization dynamics.

Our preliminary observations, illustrated in Figure 1, show significant memorization occurring early during fine-tuning, before the model achieves optimal validation perplexity or task evaluation performance. This suggests that LLMs rapidly memorize new information before reaching typical early stopping criteria, potentially exposing sensitive information. Owing to this, we

Transactions of the Association for Computational Linguistics, vol. 13, pp. 1459–1473, 2025. https://doi.org/10.1162/TACL.a.49
Action Editor: Sebastian Ruder. Submission batch: 5/2025; Revision batch: 7/2025; Published 10/2025.



(a) Pythia 1.4B (b) Pythia 12B

Figure 1: Memorization increases (number of new samples memorized at a given epoch) at successive fine-tuning epochs, comparing fine-tuning for domain adaptation (orange) and instruction tuning (purple) on the same data. Dashed vertical lines mark the average epoch for which validation perplexity (orange) and task evaluation accuracy (purple) are achieved, showing high memorization before for both (a) Pythia 1.4B and (b) Pythia 12B models.

conduct an empirical investigation into memorization in LLMs during fine-tuning, focusing on fast, deployable mitigation strategies and insights applicable during both domain adaptation and instruction tuning, leveraging widely used memorization metrics. We perform fine-tuning experiments with the *Pythia* (Biderman et al., 2023) models family across multiple parameter scales (1.4B - 12B), as well as *Llama2* 7B (Touvron et al., 2023), *Llama3* 8B and 70B (Dubey et al., 2024), and *Mistral* 7B (Jiang et al., 2023) models for both domain adaptation and instruction tuning across a range of common LLM evaluation datasets.

Our key contributions are:

- **Memorization dynamics during fine-tuning paradigms**: We examine how memorization manifests during common fine-tuning approaches; domain adaptation and instruction tuning, across a wide range of both model sizes and datasets.
- **N-gram memorization as a precursor to verbatim memorization**: We use an n-gram based partial memorization metric as an early indicator of longer phrase memorization, finding high-rates in samples prior to becoming memorized, across the majority of datasets, model scales, and fine-tuning methods.
- **Optimal stopping criteria**: We identify optimal stopping criteria during fine-tuning that significantly reduce memorization with minimal impact on performance, providing a generalizable heuristic to mitigate memorization risks in real time.
- **Comparison of mitigation techniques**: We explore loss-based regularization approaches, demonstrating further reductions in memorization that are scalable, generalizable, and competitive with an existing approach.

2 Related Work

Prior research on memorization in LLMs spans three main areas: measurement, characterization across pre-training versus fine-tuning, and mitigation, each covered in the following section.

2.1 Measuring Memorization

Evaluating the extent of memorization in LLMs necessitates robust metrics and evaluation techniques. Carlini et al. (2023) introduce the concept of *k-extractable memorization*, which measures a model's tendency to reproduce training data when provided with specific input prefixes, representing a stringent test for data leakage. Complementary approaches include membership inference attacks aimed at classifying pretraining samples (Shokri et al., 2017). Memorization and generalization have been shown to carry some interdependent relationships (Tänzer et al., 2022; Khandelwal et al., 2020; Yeom et al., 2018), with memorization dynamics in large scale LLMs studied in Tirumala et al. (2022); Carlini et al. (2023).

2.2 Memorization in Pre-training Versus Fine-tuning

The dynamics of memorization exhibit distinct characteristics during the pre-training and fine-tuning stages of LLM development. In the pre-training phase, models are exposed to extensive and often publicly available datasets, where factors such as data redundancy and model size play critical roles in determining the extent of memorization (Tänzer et al., 2022; Khandelwal et al., 2020; Carlini et al., 2023, 2019). Research indicates that larger models are more prone to rapidly memorizing training data (Tirumala et al., 2022; Nasr et al., 2025). Conversely, during fine-tuning on specialized or private datasets, different memorization risks emerge. Studies have demonstrated that specific fine-tuning methodologies,

like adapter-based techniques, can reduce the likelihood of memorizing sensitive information (Mireshghallah et al., 2022; Dodge et al., 2021; Raffel et al., 2020). Additionally, counterfactual memorization assessments (Zhang et al., 2023) aid in distinguishing between memorization arising from pre-training and that from fine-tuning, thereby informing targeted mitigation strategies tailored to each training phase.

2.3 Mitigation Strategies and Regularization

During the training process, regularization methods such as the addition of noise to input embeddings (Jain et al., 2024) are employed to mitigate memorization (Feldman and Zhang, 2020; Tirumala et al., 2022). Post-training techniques include fine-tuning and machine unlearning approaches (Maini et al., 2023), which aim to remove specific data from the model without necessitating a complete retraining. Despite these measures, achieving a balance between preserving model performance and ensuring data privacy remains a significant challenge. Mitigating memorization in language models is critical for preserving privacy and preventing the leakage of sensitive information. Conventional regularization techniques, such as weight decay and dropout, are designed to prevent overfitting and thereby reduce memorization (Feldman and Zhang, 2020). However, these methods have proven inadequate in fully reducing memorization within LLMs (Tirumala et al., 2022). Advanced regularization approaches, including data-dependent token dropout (Hans et al., 2025) and targeted token masking (Jain et al., 2024), offer partial mitigation but often fail to eliminate the risk of memorizing entire data passages, especially when dealing with highly duplicated datasets.

3 Methodology

We begin by defining how we will measure memorization, leveraging an existing approach and introducing a partial measure for fine-grained measurement. We follow this by introducing the experimental setup of our study for fine-tuning for domain adaptation and instruction tuning.

3.1 Memorization Metrics

For an exact and scalable measure of verbatim memorization, we employ the widely used extraction metric introduced in Carlini et al. (2023).

Memorization: Let f be a generative LLM trained on data D, with prefix-suffix pair (p, s) contained within a sample in D. A suffix s is said to be k-*extractable* (*memorized*) if f generates a string containing s exactly when prompted with a prefix of length k using greedy decoding.

Therefore we can compute the percentage of the fine-tuning data memorized at each fine-tuning epoch as:

$$\text{Mem} = \left(\frac{\sum k\text{-extractable suffixes } s}{\text{total samples in data } D} \right) \times 100.$$

This definition provides a directly computable metric on the generated output from our fine-tuned models, allowing fast evaluation at each fine-tuning epoch. We use the above as the definition for *memorization* throughout.

3.2 N-gram Memorization

For fine-grained measure of memorization, we implement a partial memorization metric.

N-gram Memorization: For a set of n-gram sizes $N = \{n_1, n_2, \ldots, n_k\}$, the n-gram memorization score between the model's output $f(p)$ and the target sequence s is defined as the proportion of matching n-grams of sizes in N, where the matches are exact for each n-gram but invariant to ordering of n-grams.

Formally, given M_i as the fraction of matching n-grams in N between $f(p_i)$ and s_i, the n-gram memorization score is then calculated as:

$$n - \text{gram Mem} = \left(\frac{\sum_{d \in D} M_d}{|D|} \right) \times 100.$$

This metric provides a finer-grained measure of memorization that allows for different lengths and number of n-grams, which can be tuned for suitability for specific datasets, sequence length sizes, and granularity of sensitive information.

3.3 Datasets

We leverage datasets taken from three open instruction pools–the Public Pool of Prompts (**P3**) (Sanh et al., 2022), the **FLAN** collection (Longpre et al., 2023), and the **Alpaca-52K** corpus (Taori et al., 2023). We conduct both instruction tuning and domain adaptation experiments by choosing to include/remove the task-specific instruction prompt for each dataset. These datasets encompass a range of core NLP

Category	Dataset	Domain	Input type	Output type
Classification	SST-5	Movie reviews	Single review sentence	5-way sentiment
	QQP	Quora community QA	Question_1, Question_2	Duplicate? (yes/no)
	RTE	News & Wikipedia	Premise, Hypothesis	Entail / Not-entail
	WANLI	MultiNLI-derived genres	Premise, Hypothesis	Entail / Neutral / Contradict
Question-Answering (QA)	SQuAD v2	Wikipedia articles	Paragraph context, Question	Answer span or `[NoAnswer]`
	HellaSwag	WikiHow narratives	Context + 4 candidate endings	Correct ending (MC-4)
	PubMedQA	Biomedical abstracts	Abstract (no conclusion), Question	Yes / No / Maybe
Summarization	XSum	BBC news	Full news article	One-sentence abstractive summary
	CNN/DailyMail	CNN & Daily Mail news	Full news article	Multi-sentence highlights
Instruction Following	Alpaca	Mixed user prompts	Instruction ($\pm$ optional input)	Free-form response
	FLAN v2	Multi-domain tasks	Instruction template	Free-form response

Table 1: Summary and grouping of datasets used for performing fine-tuning.

task types–including classification, Natural Language Inference (NLI), coreference resolution, Question-Answering (QA), and free-form instruction following–and span diverse domains such as encyclopedic, news, clinical notes, biomedical research, and social media text. A summary of all datasets used is outlined in Table 1, with further details in Appendix A. We categorize them into the following:

- **Classification & NLI**: short, label-based prompts (sentiment, paraphrase, entailment).
- **Question-Answering**: a mix of extractive, multiple-choice, and yes/no items.
- **Summarization**: single-sentence (XSum) and multi-sentence (CNN/DailyMail) summaries.
- **Instruction Following**: open-ended prompts from Alpaca and FLAN tasks.

These datasets are chosen to provide task and domain diversity for evaluating how this impacts memorization, as well as providing datasets which can be used for both domain adaptation and instruction tuning.

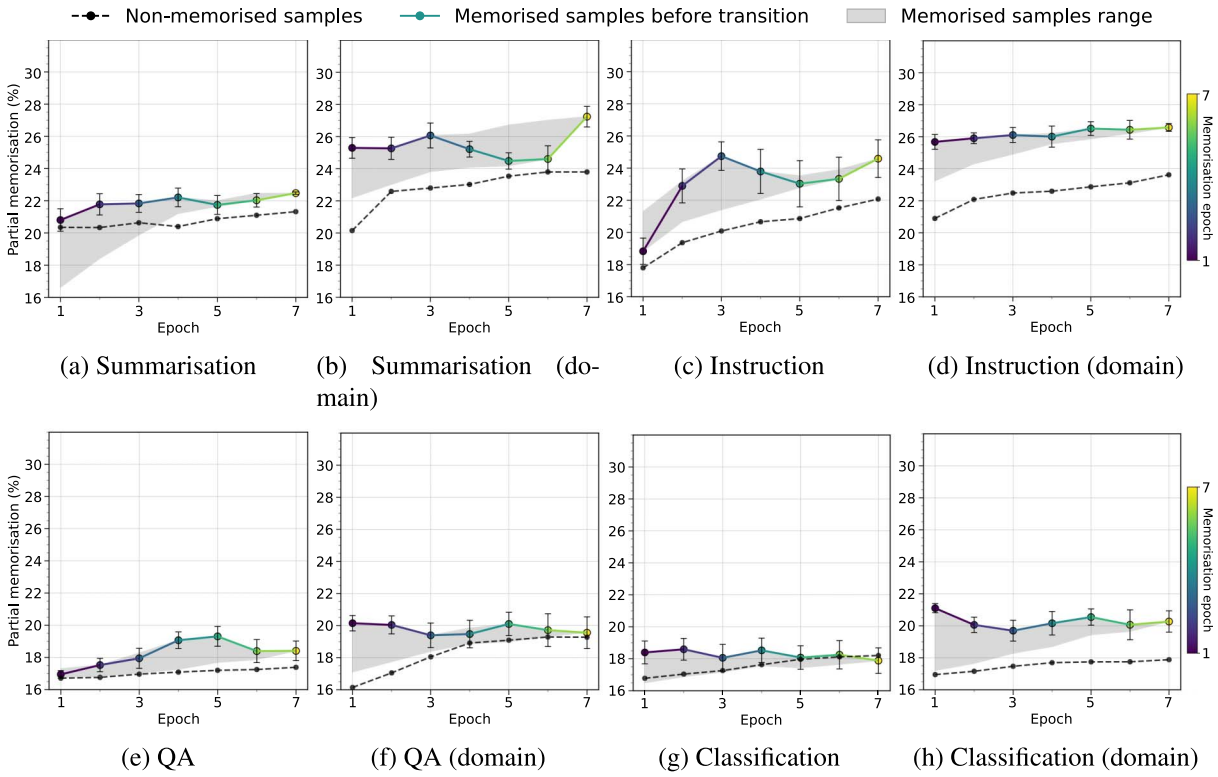
3.4 Pre-trained Models

Experiments are run on the *Pythia* model family (Biderman et al., 2023) using sizes of 1.4B, 2.8B, 6.9B, and 12B parameters. Pythia pre-training keeps hyper-parameters and dataset composition fixed while doubling model size at each step, giving a clean scaling ladder to evaluate on. Additionally, we use *Llama2* 7B (Touvron et al., 2023), *Llama3* 8B and 70B (Dubey et al., 2024), and the *Mistral 7B* model (Jiang et al., 2023). These models are chosen to enable comparisons between architectural variants at similar model sizes. All pre-trained model checkpoints are publicly accessible via HuggingFace (Wolf et al., 2020). Fine-tuning is performed using the Adam optimizer (Kingma and Ba, 2015). We perform full-parameter fine-tuning and, for comparison, perform *partial fine-tuning* in which only the top n transformer layers are updated while the rest remain frozen, enabling us to measure how restricting the trainable subset of parameters alters memorization behavior.

3.5 Fine-Tuning Approach

We employ domain adaptation and instruction tuning by fine-tuning each model for up to 8 epochs on a maximum of 5,000 samples of the target dataset. When performing domain adaptation, we simply remove the task-specific instructions from the input. We evaluate on a held-out validation set for both *validation perplexity* and task-specific *evaluation performance*. For evaluation performance, we use the standard evaluation metrics used to measure performance for that task (details of each can be found in the Appendix A). For our memorization and n-gram memorization metrics, we evaluate on the 5,000 samples of training data used to fine-tune. The small number of fine-tuning samples allows us to rapidly experiment over model scales and datasets while maintaining relevant to typically small and private fine-tuning datasets. Evaluations are performed at each epoch to monitor the progression of memorization relative to validation performance and evaluation performance.

Following Carlini et al. (2023), we test k-extractable memorization with three prefix lengths, $k \in \{12, 16, 20\}$, and a fixed 20-token



(a) Summarisation (b) Summarisation (domain) (c) Instruction (d) Instruction (domain)

(e) QA (f) QA (domain) (g) Classification (h) Classification (domain)

Figure 2: Partial n-gram memorization across fine-tuning epochs for the four dataset categories, with *domain* indicating domain adaptation fine-tuning. In each panel the colored solid line reports, at epoch t, the *median* score of samples that become memorized at subsequent epoch $t+1$; the point color encoding that memorization epoch. The grey shaded region spans the full score range of *all* samples that ever become memorized, irrespective of when the transition occurs. Error bars show the standard deviation over five random seeds, while the black dashed line is the baseline for samples that are never memorized. Results are averages over Pythia model sizes from 1.4 B to 12 B parameters.

suffix. Lengths below 12 tokens collide frequently across corpora, whereas prefixes longer than 20 tokens limits the number of samples we can use from each dataset. We empirically find that 4, 5, and 6-grams for our n-gram memorization metric provides a good signal for highly memorized phrases without being computationally prohibitive. All results are averaged over 10 runs with random seed initializations. For robustness, we use different randomly sampled prefix-suffix pairs for each of the 10 randomly initialized fine-tuning runs.

4 Results and Discussion

We begin by evaluating n-gram memorization results over model scales and domains. After, we discuss epoch selection criteria for minimizing memorization and performance trade-offs. Finally, we compare mitigation strategies across model scales.

4.1 N-gram Memorization Predicts Verbatim Memorization

Driven by the observation shown in Figure 1 that high-rate memorization occurs in the early epochs preceding optimal stopping criteria for both validation perplexity and task evaluation performance, we investigate n-gram memorization values as a proxy for fine-grained memorization. To correctly identify early warning signs of samples at high risk of verbatim memorization, we evaluate n-gram memorization after each fine-tuning epoch. Figure 2 shows our results for this evaluation on each of the dataset categories outlined in Table 1, with *domain* indicating the domain-adaptation fine-tuning. For all samples which are identified as memorized during 8 fine-tuning epochs, we track their associated n-gram memorization score on the epochs preceding the transition to verbatim memorization. This allows us to understand if the partial memorization score is higher in the

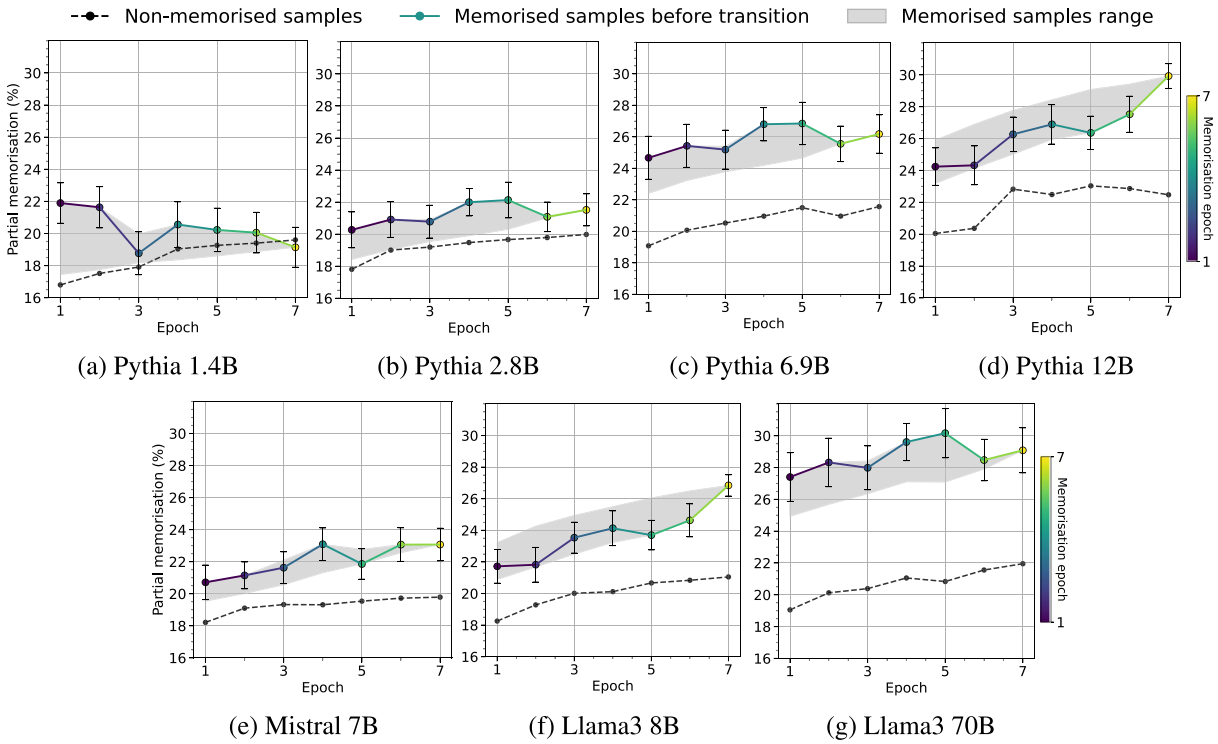


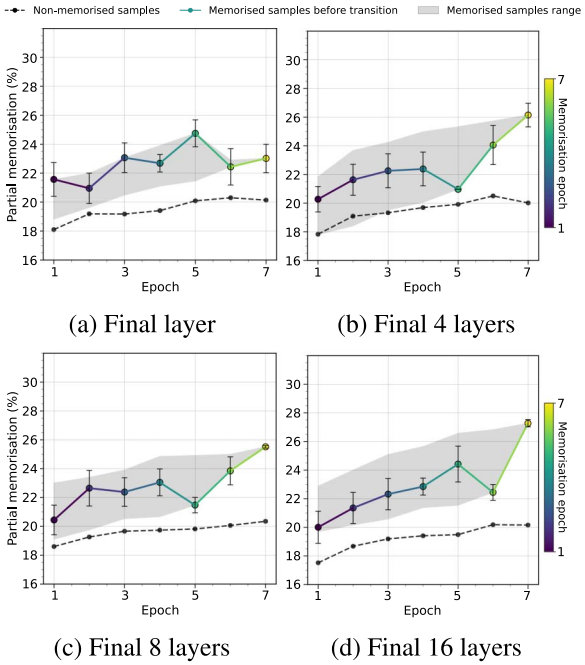
Figure 3: Partial memorization across epochs for different models. The colored solid line gives the median score of samples that will be memorized at epoch $t+1$; point color marks that future epoch. The grey region shows the full range for all eventually memorized samples, while the black dashed line is the baseline for samples never memorized. Error bars denote the standard deviation across five random seeds.

epoch preceding a transition to verbatim memorization, relative to non-memorized phrases. For this, we plot the average n-gram memorization for non-memorized phrases throughout fine-tuning as a baseline.

For each of the dataset categories visualized, we observe a clear distinction in partial memorization between the memorized and non-memorized samples, with the majority of epochs scoring markedly higher than the baseline for non-memorized samples. We see the degree to which this is higher varies significantly between domains, with largest discrepancy observed in Instruction following and Summarization. The News domains used in the summarization tend to include high-frequency stock phrases, as such, these datasets are known to encourage extractive copying (Tejaswin et al., 2021), which our results agree with. Most notably, we find that for all datasets, the domain adaptation version sees a significant increase in partial memorization over the baseline, whereas the baseline scores do not change significantly. Interestingly, there is a large increase in partial memorization scores of samples which are memorized in the early epochs when performing domain adaptation.

We perform the same evaluation but compare model size and architecture, shown in Figure 3. We identify the expected trend that larger model sizes correlate to higher memorization capacity, which is reflected in the partial memorization score increase across the Pythia models. The partial memorization score gap between memorized and non-memorized samples increases significantly with increasing model size, showing a strong indicator that this metric serves as a scalable precursor to verbatim memorization. An unexpected result is that for the smaller 1.4B model, partial memorization decreases for samples memorized in the latter epochs of fine-tuning; a trend which does not follow for the larger model sizes. Comparing different architectures, we find similar gaps to baseline and the same trend of increasing partial memorization gap to baseline over fine-tuning epochs.

Figure 4 repeats the analysis for Llama3 8B when only the top n transformer layers are updated. The non-memorized baseline is unaffected, but unfreezing more layers suppresses partial memorization in the first few epochs and heightens it in later epochs. This is consistent with a capacity-bottleneck view in which extra trainable layers



(a) Final layer (b) Final 4 layers

(c) Final 8 layers (d) Final 16 layers

Figure 4: Final-layer partial fine-tuning comparison of the Llama3 8B model. The final n layers of the model are unfrozen and updated when fine-tuning, with the remaining layers frozen.

delay, yet ultimately amplify, overfitting during fine-tuning.

4.2 Selection Criteria as Mitigation

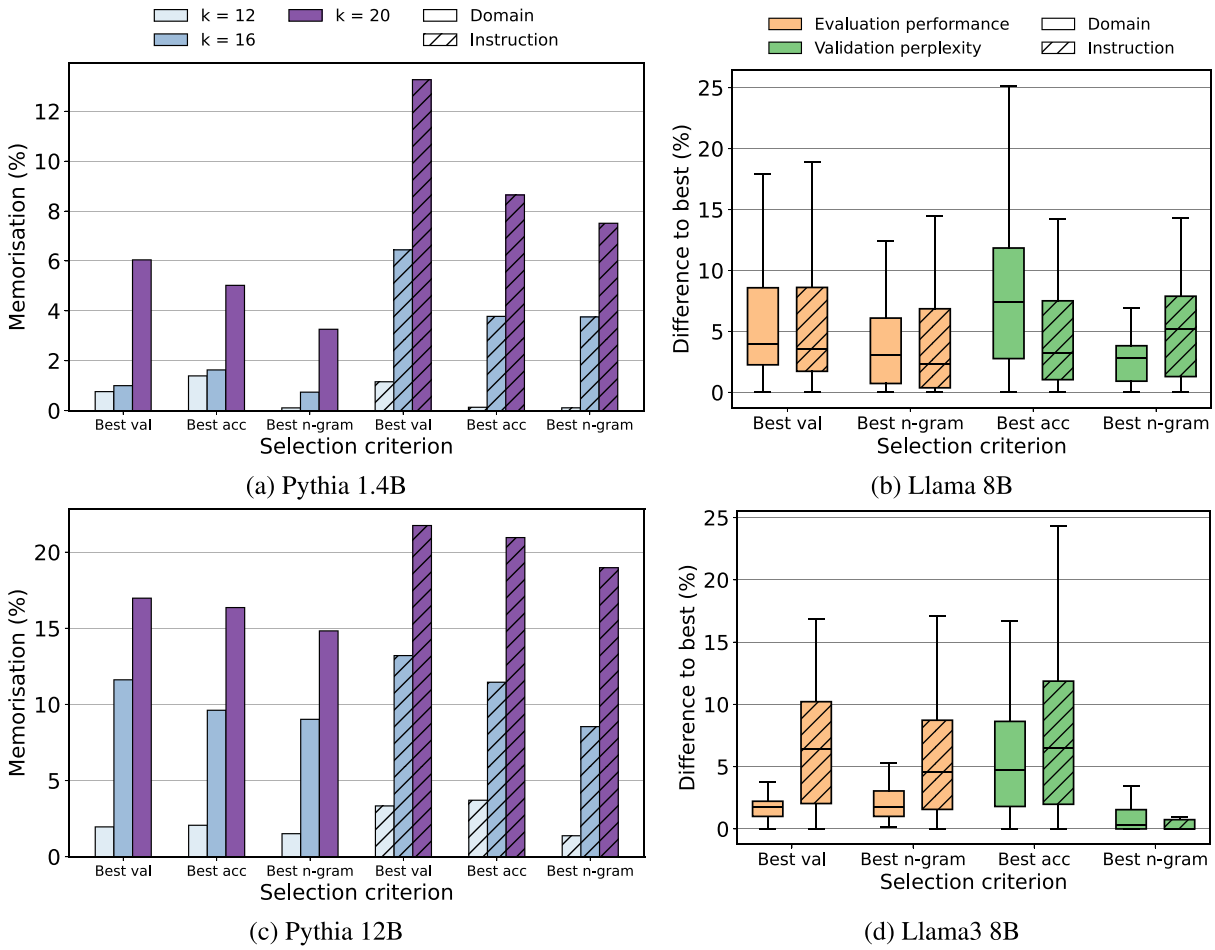
Following our findings that high-rate memorization occurs before optimal validation perplexity or task evaluation performance, and that partial memorization serves as a potential precursor to memorization, we now investigate the efficacy of utilising this as an early stopping criterion. Without resorting to regularization or unlearning strategies, we explore using n-gram memorization as a threshold for early stopping. To adapt n-gram memorization as an early stopping criterion, we test different threshold values for which to stop fine-tuning if exceeded. We find that an average partial memorization threshold score of 20 on the fine-tuning set yields good results. We compare this to the naive selection criterion of validation perplexity and task evaluation for domain adaptation and instruction tuning, respectively, although we experiment with applying validation perplexity and best accuracy to both.

Results for these experiments are shown in Figure 5, highlighting the trade-offs between different early stopping criteria and their impact on both memorization and model performance. Using evaluation performance/accuracy as the selection criterion consistently reduces memorization rates in both domain adaptation and instruction tuning scenarios (Figure 5a and Figure 5c). This could be due to task evaluation performance correlating more highly to the latent capabilities of the pretrained model, rather than validation perplexity on a single domain, and therefore is optimized at lower memorization. However, this comes at the cost of a significant decrease in validation perplexity, as indicated by the high variance and larger differences to the best perplexity scores shown in Figure 5b and Figure 5d. Conversely, when validation perplexity is used as the selection criterion, the models tend to show the opposite behavior through achieving better perplexity scores, but with substantially higher memorization rates, particularly for instruction-tuned models which consistently exhibit the highest memorization levels compared to domain adaptation results.

Interestingly, the n-gram selection criterion strikes a balance, reducing memorization without the steep performance trade-offs observed in the other criteria. It provides a more favorable balance by keeping memorization lower and maintaining better accuracy and perplexity than either of the naive criteria (evaluation accuracy or validation perplexity), as seen by the smaller performance differences at consistently lower memorization percentages. In summary, instruction tuning appears more prone to memorization, particularly under validation-based selection, whereas domain adaptation is relatively less affected by these selection criteria, and n-gram thresholding as a stopping criterion is a simple and effective memorization mitigation strategy.

4.3 Comparing Mitigation Strategies

We test if our n-gram approach can be baked into a loss regularization function by adapting the typical causal LLM loss to include a term to penalize high-confidence n-grams exceeding a tunable confidence threshold, above that of the pretrained model. Intuitively, this penalty is designed to discourage the model from assigning excessively high probabilities to these n-grams as a proxy measure for n-gram memorization. The key limitations of this strategy are in requiring the original model to run inference alongside fine-tuning to acquire the baseline confidence values, and keeping n-gram sizes within practical bounds to not become computationally intensive.



(a) Pythia 1.4B

(b) Llama 8B

(c) Pythia 12B

(d) Llama3 8B

Figure 5: Memorization and performance comparison for domain adaptation and instruction tuning across different early stopping selection criteria. (a) and (c) show the verbatim memorization percentage for different values of extraction prompt prefix length $k \in \{12, 16, 20\}$ using three early stopping selection criteria: validation perplexity (Best val), evaluation performance (Best acc), and n-gram memorization (Best n-gram) for domain adaptation (solid) and instruction tuning (hatched). (b) and (d) present the difference to the best task evaluation performance (orange) and validation perplexity (green), across the same selection criteria and fine-tuning approaches.

Further details of this approach can be found in Appendix B.

We compare our n-gram regularization to the *Goldfish loss* regularization technique (Hans et al., 2025), incorporating random sampling of dropped tokens from the loss calculation for a given training sample. We test across all models to evaluate transferability and scalability of approach.

We present our results in Table 2, grouped by model size and dataset category, including comparisons to naive baseline results for both domain adaptation and instruction tuning (top row of each model group). We include the stopping criterion *Best n-gram* as a simple non-regularization approach based on the promising findings in Section 4.2. We consider memorization (Mem %) and evaluation performance (Eval %), where Eval is taken as the difference to the best achieved performance—essentially measuring the performance trade-off of the memorization mitigation technique. We group our results by model size, and report the best (bold) within each group.

4.3.1 Impact of Model Size

These results highlight key trends across different model scales and mitigation strategies. Generally, memorization increases with model size, as observed with the unmitigated baseline for Pythia 2.8B of 12.2% rising to 21.8% for Pythia 12B and 25.7% for Llama3 70B. Importantly, the mitigation strategies show consistent reductions in memorization across all models. For example, n-gram regularization reduces memorization from 12.2% to 3.6% in Pythia 2.8B, and from 21.8% to 6.5% in Pythia 12B. We see similar reductions in

	QA		Summarization		Instruction		Average	
Model + Strategy	Mem ↓	Eval ↓	Mem ↓	Eval ↓	Mem ↓	Eval ↓	Mem ↓	Eval ↓
Pythia 2.8B	9.71	–	12.59	–	14.30	–	12.20	–
+ Best n-gram	4.36	7.51	5.63	6.53	6.43	8.09	5.47	7.38
+ n-gram reg	**2.90**	5.08	**3.75**	**4.25**	**4.29**	6.54	**3.65**	5.29
+ Goldfish reg	3.37	**5.04**	4.38	4.31	5.01	**6.07**	4.25	**5.14**
Pythia 6.9B	12.80	–	16.50	–	18.70	–	16.00	–
+ Best n-gram	5.76	7.54	7.42	6.21	8.42	8.30	7.20	7.35
+ n-gram reg	**3.84**	5.15	**4.95**	**4.54**	5.61	**6.30**	**4.80**	**5.33**
+ Goldfish reg	4.48	**5.07**	5.77	4.83	**5.55**	6.59	5.27	5.50
Mistral 7B	13.65	–	17.50	–	19.88	–	17.01	–
+ Best n-gram	6.12	7.55	7.88	6.00	8.91	8.89	7.64	7.48
+ n-gram reg	4.18	5.53	5.25	**4.40**	5.34	**6.08**	4.92	**5.34**
+ Goldfish reg	**4.01**	**5.40**	**5.12**	4.42	**4.97**	6.21	**4.70**	**5.34**
Llama3 8B	14.40	–	18.50	–	20.94	–	17.95	–
+ Best n-gram	6.48	9.21	8.33	6.56	9.41	10.31	8.07	8.69
+ n-gram reg	**4.32**	**4.38**	**5.55**	**3.81**	**6.27**	**5.32**	**5.38**	**4.50**
+ Goldfish reg	5.04	5.02	6.47	4.33	7.32	6.99	6.28	5.45
Pythia 12B	17.66	–	22.50	–	25.30	–	21.82	–
+ Best n-gram	7.92	9.20	10.12	6.41	11.39	8.30	9.81	7.97
+ n-gram reg	**5.28**	3.98	**6.75**	**4.02**	**7.59**	**4.91**	**6.54**	**4.30**
+ Goldfish reg	6.10	**3.90**	7.57	4.36	8.86	5.00	7.51	4.42
Llama3 70B	20.80	–	26.50	–	29.70	–	25.67	–
+ Best n-gram	9.36	9.39	11.93	5.96	13.37	8.45	11.55	7.93
+ n-gram reg	**6.24**	5.54	**7.05**	**3.91**	**8.91**	**5.44**	**7.40**	**4.96**
+ Goldfish reg	7.18	**5.50**	7.27	4.01	10.40	6.11	8.28	5.21

Table 2: Main memorization mitigation results across model scales and mitigation strategies. For each result we report the memorization (*Mem*, lower is better), and Evaluation difference (*Eval*, lower is better) to the best performance achieved for the naive unmitigated strategy (top row of each model group). Bold values indicate the best (lowest) score within each model group (base row excluded). Memorization scores are taken as the average of all prefix lengths $k \in \{12, 16, 20\}$ extractions. Results are averages over 10 randomly initialized fine-tuning runs.

the Llama3 and Mistral models. Goldfish regularization is also effective, though its impact is more pronounced on the Mistral 7B model, whereas our n-gram reg outperforms this on all other models. Across the board, larger models present greater challenges in balancing memorization, validation perplexity, and accuracy. The results suggest that as model size increases, the trade-offs become more pronounced.

4.3.2 Impact of Mitigation Strategy

Averaged over all models, *n-gram regularization* delivers the best trade-off, lowering memorization to 5.45% with a performance evaluation gap of 4.95%; this is a ≈40% relative reduction in memorization and a ≈35% smaller performance hit compared with the simple *Best n-gram* early-stopping rule (8.29%, 7.80%). *Goldfish* is a close second (6.05%, 5.18%), performing best on Mistral 7B. While the early-stopping heuristic of Best n-gram consistently sees higher memorization and worse evaluation performance, it still significantly reduces memorization from the naive baseline—highlighting the importance of a simple non-regularization approach.

4.3.3 Categorical Analysis

We construct coarse semantic categories of n-grams over all datasets, randomly sample 500

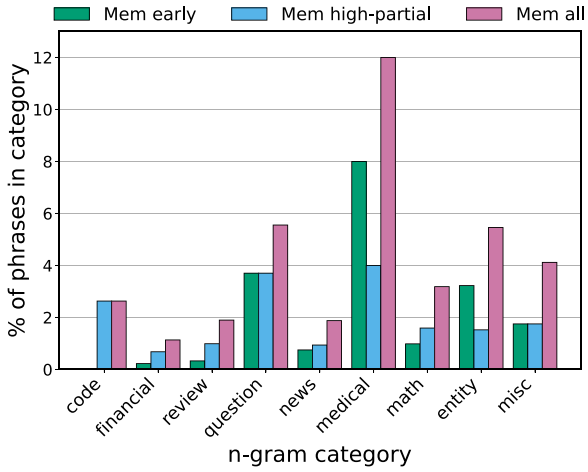


Figure 6: Distribution of memorized n-grams across coarse semantic categories. Bars show the percentage of phrases *within* each category that are: memorized within the first two fine-tuning epochs and remained memorized ('Mem early', green), memorized after exhibiting higher than baseline partial memorization in a preceding epoch ('Mem high-partial', blue), and all verbatim memorized phrases ('Mem all', violet).

unique n-grams *per* category, and plot their memorization outcomes in Figure 6. Memorization differs markedly between categories, with *medical*, *question*, and *entity* forming the highest-risk groups, while *financial*, *news*, and *review* remain lowest. Both *medical* and *question* show high rates of early memorized samples that persist, and most categories include many phrases that enter a high-partial state before verbatim copying, with *code* being the clearest example. Categories whose text is highly templated and repetitive (e.g., medical, questions, and named-entity lists) likely present many identical n-gram patterns for the model to latch onto, whereas free-form prose like news or reviews offers far fewer exact repeats. Qualitative examples of memorized phrases are reported in Appendix C.

5 Limitations

Our study provides insights into memorization during domain adaptation and instruction tuning of causal LLMs, but has limitations. We focused on greedy decoding, while real-world applications often use more complex methods like beam search, which likely influence memorization differently—future research should explore memorization under various decoding strategies.

We used validation perplexity and evaluation performance as metrics, but their trade-offs with memorization aren't necessarily equivalent. Investigating alternative metrics could offer a more nuanced understanding of these relationships. Our experiments were limited to a single high-parameter model (Llama3 70B) due to computational budget limitations—ideally we would evaluate these finding are a larger pool of models and sizes, as well as different fine-tuning protocols.

6 Conclusion and Future Work

This study explores memorization dynamics during both domain adaptation and instruction tuning across eight open-weight LLMs (1.4B–70B parameters). We show that a simple n-gram partial memorization score indicates at-risk samples. The gap between memorized and non-memorized items is widest in domain adaptation and summarization datasets, reflecting repetition and lack of diversity seen with instruction-tuning, whereas classification and QA tasks exhibit a smaller, but still measurable, rise. We also show that our partial memorization metric scales very well with increasing model size, where memorization is more pronounced. Building on these observations, we explore memorization mitigation strategies. A threshold-based early stopping with the n-gram score halves memorization relative to the baseline at low performance cost, but an explicit n-gram penalty in the loss is more effective, averaging 5.45% memorization and a 4.95% performance gap: roughly a 40% reduction in memorization. We show this scales from small models to 70B-parameter models and generalizes across datasets and tasks.

Future work will extend this analysis in two directions. First, alternative decoding strategies such as beam search may surface different leakage patterns and should be audited with the same metrics. Secondly, we will test whether the n-gram regularizer curbs memorization in code generation, mathematical reasoning and multimodal tasks.

Acknowledgments

We would like to express our sincere thanks and appreciation to the reviewers and action editors for their invaluable comments and insights, and to Wordnerds and ERDF for funding.

References

Rohan Anil, Badih Ghazi, Vineet Gupta, Ravi Kumar, and Pasin Manurangsi. 2022. Large-scale differentially private BERT. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6481–6491, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.findings-emnlp.484

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

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2023. Quantifying memorization across neural language models.

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 Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650. USENIX Association.

Hao Cheng, Zhaowei Zhu, Xing Sun, and Yang Liu. 2023. Mitigating memorization of noisy labels via regularization between representations. In *The Eleventh International Conference on Learning Representations*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. 2021. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1286–1305, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.emnlp-main.98

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel

Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, et al. 2024. The Llama 3 herd of models. *CoRR*, abs/2407.21783.

William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(1).

Vitaly Feldman and Chiyuan Zhang. 2020. What neural networks memorize and why: Discovering the long tail via influence estimation. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Abhimanyu Hans, Yuxin Wen, Neel Jain, John Kirchenbauer, Hamid Kazemi, Prajwal Singhania, Siddharth Singh, Gowthami Somepalli, Jonas Geiping, Abhinav Bhatele, and Tom Goldstein. 2025. Be like a goldfish, don't memorize! Mitigating memorization in generative LLMs. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA. Curran Associates Inc.

Neel Jain, Ping Yeh Chiang, Yuxin Wen, John Kirchenbauer, Hong-Min Chu, Gowthami Somepalli, Brian R. Bartoldson, Bhavya Kailkhura, Avi Schwarzschild, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2024. NEFTune: Noisy embeddings improve instruction finetuning. In *The Twelfth International Conference on Learning Representations*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through memorization: Nearest neighbor language models.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings*.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V. Le, Barret Zoph, Jason Wei, and Adam Roberts. 2023. The flan collection: Designing data and methods for effective instruction tuning. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Pratyush Maini, Michael C. Mozer, Hanie Sedghi, Zachary C. Lipton, J. Zico Kolter, and Chiyuan Zhang. 2023. Can neural network memorization be localized? In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Fatemehsadat Mireshghallah, Archit Uniyal, Tianhao Wang, David Evans, and Taylor Berg-Kirkpatrick. 2022. Memorization in NLP fine-tuning methods. In *First Workshop on Pre-training: Perspectives, Pitfalls, and Paths Forward at ICML 2022*.

Milad Nasr, Javier Rando, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A. Feder Cooper, Daphne Ippolito, Christopher A. Choquette-Choo, Florian Tramèr, and Katherine Lee. 2025. Scalable extraction of training data from aligned, production language models. In *The Thirteenth International Conference on Learning Representations*.

Vaidehi Patil, Peter Hase, and Mohit Bansal. 2024. Can sensitive information be deleted from LLMs? Objectives for defending against extraction attacks. In *The Twelfth International Conference on Learning Representations*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M. Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18, Los Alamitos, CA, USA. IEEE Computer Society. https://doi.org/10.1109/SP.2017.41

Michael Tänzer, Sebastian Ruder, and Marek Rei. 2022. Memorization 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, Dublin, Ireland. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.acl-long.521

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following LlaMa model. https://github.com/tatsu-lab/stanford_alpaca

Priyam Tejaswin, Dhruv Naik, and Pengfei Liu. 2021. How well do you know your summarization datasets? In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3436–3449, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.findings-acl.303

Kushal Tirumala, Aram H. Markosyan, Luke Zettlemoyer, and Armen Aghajanyan. 2022. Memorization without overfitting: Analyzing the training dynamics of large language models. In *Advances in Neural Information Processing Systems*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LlaMa: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics. https://doi.org/10.18653/v1/W18-5446

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. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on*

Empirical Methods in Natural Language Processing: System Demonstrations, pages 38–45, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-demos.6

Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. 2018. Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting. In *2018 IEEE 31st computer security foundations symposium (CSF)*, pages 268–282. IEEE. https://doi.org/10.1109/CSF.2018.00027

Chiyuan Zhang, Daphne Ippolito, Katherine Lee, Matthew Jagielski, Florian Tramèr, and Nicholas Carlini. 2023. Counterfactual memorization in neural language models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

A Datasets

The following datasets are used and evaluated according to their respective benchmarks, found in Taori et al. (2023), Longpre et al. (2023), and Wang et al. (2018).

- **SST-5**. Movie-review sentences annotated with five sentiment levels. *Template:* `Sentence: <s> - What is the sentiment?`. **Metric:** accuracy.
- **QQP**. Pairs of Quora questions labelled as duplicates or not. *Template:* `Q1: <q1>\nQ2: <q2> - Duplicate? Yes/No`. **Metric:** accuracy and F1.
- **RTE**. Premise–hypothesis pairs drawn from news and Wikipedia, framed as binary entailment. *Template:* `Premise: <p> Hypothesis: <h> - Entailed? Yes/No`. **Metric:** accuracy.
- **WANLI**. Large-scale adversarial Natural Language Inference corpus generated via human–AI collaboration. *Template:* `Premise:, Hypothesis:, Label: entail / neutral / contradict`. **Metric:** accuracy.
- **SQuAD v2**. Wikipedia paragraphs paired with questions, mixing answerable and unanswerable cases. *Template:* `Context: <para> Question: <q> Answer:`. **Metric:** exact match (EM) and F1.
- **HellaSwag**. Multiple-choice commonsense completion task built from WikiHow and activity narratives. *Template:* `Story: <ctx> Which ending (A–D) is most plausible?`. **Metric:** multiple-choice accuracy.
- **PubMedQA-L**. Biomedical abstracts with yes/no/maybe answers to research questions. *Template:* `Abstract: <abs> Question: <q> Answer (yes/no/maybe):`. **Metric:** accuracy.
- **XSum**. BBC news articles paired with single-sentence abstractive summaries. *Template:* `Article: <doc> nWrite a one-sentence summary:`. **Metric:** ROUGE-1/2/L.
- **CNN/DailyMail**. Long-form news articles with multi-sentence ‘‘highlights’’. *Template:* `Article: <doc> Summarize concisely:`. **Metric:** ROUGE-1/2/L.
- **Alpaca-52k**. GPT-3.5-generated instruction–response pairs covering diverse tasks. *Template:* `Instruction:, Input:, Response:`. **Metric:** GPT-4 preference win-rate.
- **FLANv2**. Composite collection of ∼1.8k tasks (12M examples) in instruction format. *Template:* `Instruction: {task}Input: {x} Answer:`. **Metric:** task-specific (Accuracy, F1, ROUGE, etc.).

B N-gram Regularization Loss

To incorporate n-gram regularization into the standard causal language modelling loss function, we modify the loss function to include a penalty term that discourages the model from assigning excessively high confidence to certain n-grams compared to the pre-trained model. The modified loss function consists of two main components:

Primary Loss Term:

$$\mathcal{L}_{\text{LM}} = -\sum_{t=1}^{T} \log p_\theta(x_t \mid x_{<t})$$

where T is the total length of the token sequence, x_t is the token at position t, $x_{<t} = (x_1, x_2, \ldots, x_{t-1})$ represents all previous tokens before position t, $p_\theta(x_t \mid x_{<t})$ is the probability of token x_t given previous tokens under the current model parameters θ. This is the standard cross-entropy loss used for causal LLM training.

N-gram Regularization Term:

$$\mathcal{L}_{\text{reg}} = \lambda \sum_{g \in \mathcal{G}} \left[\max\{0,\ p_\theta(g) - p_{\theta_0}(g) - \tau\}\right]^2$$

where $p_\theta(g)$ is the probability assigned by the fine-tuned model to the n-gram g, $p_{\theta_0}(g)$ is the probability assigned by the pre-trained model to the same n-gram, $\lambda \geq 0$ is the regularization strength, and $\tau \geq 0$ is the confidence margin. For an n-gram $g = (w_1, \ldots, w_n)$ we compute $p_\theta(g) = \prod_{i=1}^{n} p_\theta(w_i \mid w_{<i})$ and analogously for $p_{\theta_0}(g)$. The penalty is applied only when the fine-tuned model's confidence exceeds the pre-trained model's by more than τ; otherwise the term is zero. Balancing this term prevents the model from over-memorizing while preserving latent pre-trained performance.

C Qualitative Examples

Table 3 contains example model output prefix-suffix pairs evaluated as 'memorized', alongside any incorrect continuations relative to ground truth.

Category	Input prefix, predicted suffix, and ground truth
Instruction	Sophie sat at her desk, staring blankly at the computer screen. Her mind was racing as she weighed the options before deciding to quit her job. [GT: in front of her.]
Review	Feels as if there's a choke leash around your neck so director Nick Cassavetes can give it a good, hard yank and the audience grows impatient. [GT: whenever he wants you to feel something.'']
Question	How can I lose ten pounds in three weeks without exercising? I currently weigh 185 pounds and have an office job that requires a lot of sitting. [GT: but I dislike counting calories–what should I do?'']
News	An American woman died aboard the MS Veendam, owned by cruise operator Holland America Line, after the ship docked in Rio de Janeiro , officials later added she was travelling alone. [GT: on Tuesday, according to the state-run Brazilian news agency Agencia Brasil.'']
Medical	Oral gentamicin as well as oral and intraperitoneal polymyxin B, which binds endotoxin, did not prevent hepatic injury in rats with self-filling blind loops. [GT: <*sentence ends*>]
Medical	However, oral metronidazole and tetracycline therapy continuously administered beginning 1 day after surgery diminished hepatic injury (histology score 3.0 +/− 1.8 and led to unexpected weight gain. [GT: day after surgery diminished hepatic injury...]
Financial	Dow, S&P 500 and Nasdaq futures slipped between 0.7% and 1% as the rise in bond yields weighed and Apple again fell 1.1% in pre-market trading ahead of earnings while Tesla rose on delivery optimism. [GT: <*sentence ends*>]
Financial	The 30-year bond US30YT=RR firmed 26/32, taking its yield to 4.17 percent, after hitting another record low of 4.16 percent as investors fled into equities. [GT: <*sentence ends*>]

Table 3: Example predicted continuations given a 10-token input prefix (grey). Green spans mark ≥10-token verbatim copies; red tokens show divergence. Square-bracketed lines give the Ground-Truth (GT) continuation for each divergent span.