STUDYING MEMORIZATION DYNAMICS IN LARGE LANGUAGE MODELS ACROSS PRE-TRAINING

Kaustubh Ponkshe* **Raghav Singhal*** **Daniele Affinita*** **Martin Jaggi**

École Polytechnique Fédérale de Lausanne (EPFL), Switzerland

ABSTRACT

Large language models are known to memorize training data, raising concerns about privacy, copyright, and unintended data leakage. Yet how memorization evolves across the training lifecycle of modern models remains poorly understood. In this work, we analyze memorization dynamics across pre-training, cooldown, and model merging using intermediate checkpoints from the OLMo-2 13B training trajectory. We distinguish between *theoretical memorization*, measured by sequence likelihood, and *practical extractability*, measured via prefix-based extraction attacks. Our experiments reveal three key phenomena. First, memorization increases with repetition frequency. Second, we uncover a strong *recency bias*: data introduced during the final cooldown phase becomes significantly more extractable than earlier data despite fewer total exposures, indicating effective forgetting in token-rich regimes. Third, we identify a *merging anomaly*: although weight-averaged models exhibit loss values consistent with reduced overfitting, their practical extractability is often higher than that of any individual ingredient model. This divergence shows that extractability and compressibility, typically treated as correlated, can decouple after model merging. Overall, our findings emphasize the need for training-stage-aware evaluation and provide new insights into memorization in modern LLM training pipelines.

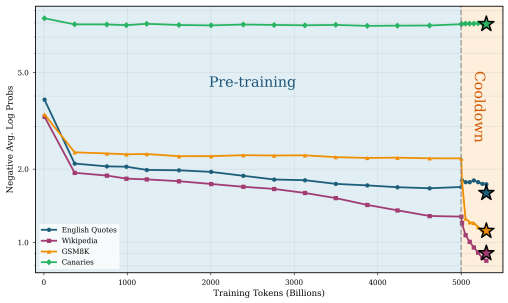
1 INTRODUCTION

As large language models (LLMs) scale (Achiam et al., 2023), their capacity to memorize information increases substantially (Carlini et al., 2021; Biderman et al., 2024; Nasr et al., 2025). While memorization can aid factual recall, it poses serious risks for copyright, as models may reproduce long training sequences verbatim (Karamolegkou et al., 2023; Allen-Zhu, 2025). Understanding the mechanisms driving memorization is therefore essential for responsible deployment.

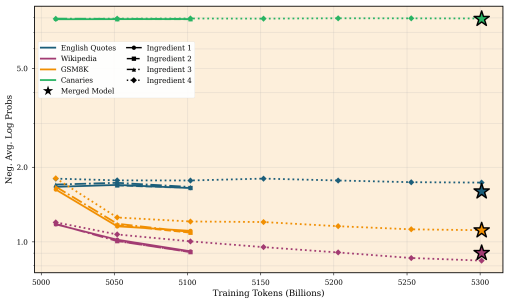
In this work, we study the dynamics of memorization across the training trajectory of modern LLMs. We distinguish between *theoretical memorization*, defined as the model's ability to compress a sequence as measured by negative log-likelihood or training loss, and *practical extractability*, defined as the ability to generate a sequence verbatim given a prefix. Although prior work has linked memorization to data repetition (Carlini et al., 2021), the roles of training stage, such as pre-training versus cooldown, and advanced techniques like model merging remain underexplored. We analyze intermediate checkpoints of **OLMo 2** 13B (OLMo et al., 2024), leveraging its transparent training recipe and data mixture to trace how memorization evolves over time. By probing datasets from different distributions and introduced at different training stages, we decouple the effects of repetition frequency from training recency.

Our experiments yield three key findings. First, higher repetition consistently causes greater memorization. Second, we observe a strong *recency bias*: data introduced during the cooldown phase is memorized significantly more than data seen early in pre-training, despite fewer total exposures. This suggests that for models trained beyond Chinchilla-optimal regimes (Hoffmann et al., 2022), early data is effectively forgotten unless reinforced. Third, we identify an anomaly in model merging. For individual checkpoints, theoretical memorization closely tracks practical extractability.

*Equal contribution. Author ordering decided randomly.



(a) Theoretical memorization across all datasets during pre-training and cooldown.



(b) Impact of cooldown and merging on theoretical memorization across all datasets.

However, this relationship breaks down in the final merged model. While weight averaging smooths the loss landscape, practical extractability remains disproportionately high. This challenges the use of loss-based metrics alone for auditing privacy risks in deployed models.

2 Preliminaries

2.1 Auto-regressive Language Models

Auto-regressive language models generate text by predicting each token conditioned on all preceding tokens. Let $\mathcal{V}$ denote a vocabulary of size typically between 10^5 and 10^6. Given a prompt $(x_{-l_p}, \ldots, x_{-1}) \in \mathcal{V}^{l_p}$, the model produces a continuation $(x_0, \ldots, x_{l-1}) \in \mathcal{V}^{l}$. The probability of the continuation factorizes as:

$$p(x_0, \ldots, x_{l-1} \mid x_{-l_p}, \ldots, x_{-1}) \;=\; \prod_{i=0}^{l-1} p(x_i \mid x_{-l_p}, \ldots, x_{i-1}),$$

where each term corresponds to the model's prediction at position i. At inference time, the model outputs a distribution over $\mathcal{V}$ for each token position. Since training maximizes likelihood over large corpora, frequently observed continuations can receive disproportionately high probability, leading to memorization and potential verbatim reproduction.

2.2 Memorization and Extractability

Prior work identifies *extractable memorization* as a particularly relevant notion, describing cases where a model can be induced to reproduce training sequences when prompted with informative prefixe (Carlini et al., 2021; 2022).

Definition 1 (Extractable Memorization) *Let p be a language model and $x = (x_0, \ldots, x_{l-1})$ a sequence of length l. The sequence x is* extractable with context length l_p *if there exists a prefix $x_- = (x_{-l_p}, \ldots, x_{-1})$ such that $[\, x_- \,\|\, x \,]$ appears in the training data of p, and greedy decoding from x_- reproduces x. Formally, for all $i \in \{0, \ldots, l-1\}$,*

$$x_i \;=\; \arg\max_{v \in \mathcal{V}} p\big(v \mid x_{-l_p}, \ldots, x_{i-1}\big).$$

This definition is useful because it aligns with realistic threat models involving unintended disclosure of memorized content (Nasr et al.) and provides a clear, testable criterion for empirical evaluation (Carlini et al., 2022).

3 Experiments

To study memorization dynamics in OLMo 2, we use four probing datasets that differ systematically in their exposure during training. Specifically, the datasets vary in whether their examples were seen during pre-training, during the mid-training cooldown phase, or during both. Table 1 summarizes the exposure patterns for each dataset.

Table 1: Datasets used and presence in different training phases.

Dataset	Seen in Pre-training	Seen in Cooldown
GSMSK	No	Yes
Wikipedia	Yes	Yes
English Quotes	Yes	No
Canary	No	No

3.1 Model Training Trajectory and Checkpoints

We analyze intermediate checkpoints across three distinct phases of the OLMo 2 13B training recipe:

- **Phase 1: Pre-training (0 - 5000B tokens).** The model is trained on a diverse corpus with a standard learning rate schedule.
- **Phase 2: The Cooldown (5000B-5300B tokens).** The learning rate is linearly annealed to zero, and the data distribution shifts towards high-quality, domain-specific samples. Crucially, this phase consists of four independent runs ("ingredients"): three trained on 100B tokens and one on 300B tokens.
- **Phase 3: Model Souping.** The final "Base Model" is constructed by averaging the weights of the four Phase 2 ingredients. We evaluate both the individual ingredients and the final merged model to test the effects of model merging on memorization.

3.2 Probe Datasets

We use four probe datasets to test specific hypotheses:

- **High-Repetition Baseline (Wikipedia (Bridge, 2001)):** 2000 Wikipedia sequences (taken from `https://huggingface.co/datasets/allenai/dolmino-mix-1124`) with high repetition counts present throughout both Phase 1 and Phase 2. These serve as a positive control for frequency-driven memorization.
- **The Forgetting Test (English Quotes. `https://huggingface.co/datasets/Abirate/english_quotes`):** A dataset (2.51K samples) present *only* in Phase 1 and absent during Phase 2. Investigating this set allows us to test the "Forgetting Hypothesis", whether early training data is overwritten in token-rich regimes.
- **The Recency Test (GSM8K (Cobbe et al., 2021), `https://huggingface.co/datasets/openai/gsm8k`):** Grade School Math problems (8.79K samples) introduced *only* during Phase 2 (Cooldown). High extractability here would isolate *recency bias*, as these samples have low total repetition compared to the pre-training corpus.
- **Control Group (Canaries, `https://huggingface.co/datasets/swiss-ai/apertus-pretrain-poisonandcanaries`):** Randomized samples (24.6K) strictly absent from all training stages, used to establish a zero-memorization baseline.

3.3 Evaluation Protocol

We operationalize memorization using two distinct classes of metrics:

Theoretical Memorization (Compression) We compute the negative log-probability of the target sequences (with batch size 32 and max. sequence length 256). This measures how well the model has compressed the data distribution, as a proxy for Kolmogorov complexity.

Practical Extractability (Generation) To assess memorization in a realistic black-box setting, we simulate an extraction attack in which the model is conditioned on a prefix of $k = 50$ tokens and required to generate the subsequent $n = 50$ tokens using greedy decoding (temperature $T = 0$, top-$k = 0$, and top-$p = 1$) and batch size 32. The generated continuations are evaluated using a suite of metrics that capture both exact and partial reconstruction:

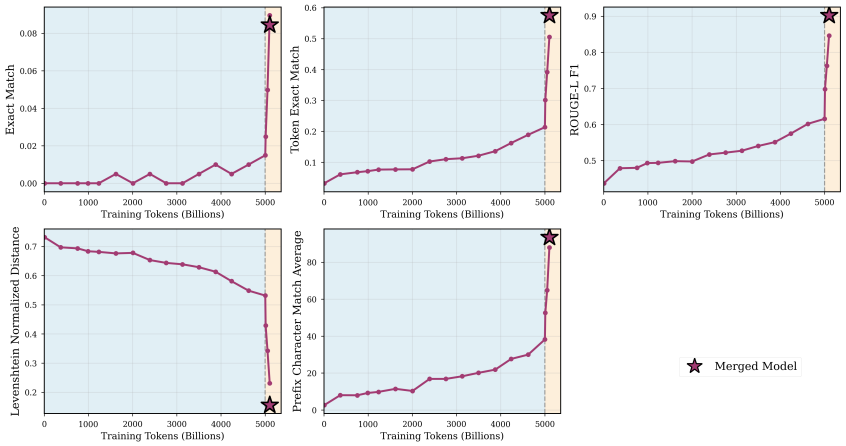


Figure 2: Practical extractability metrics on the highly memorized *Wikipedia* dataset during pre-training and cooldown.

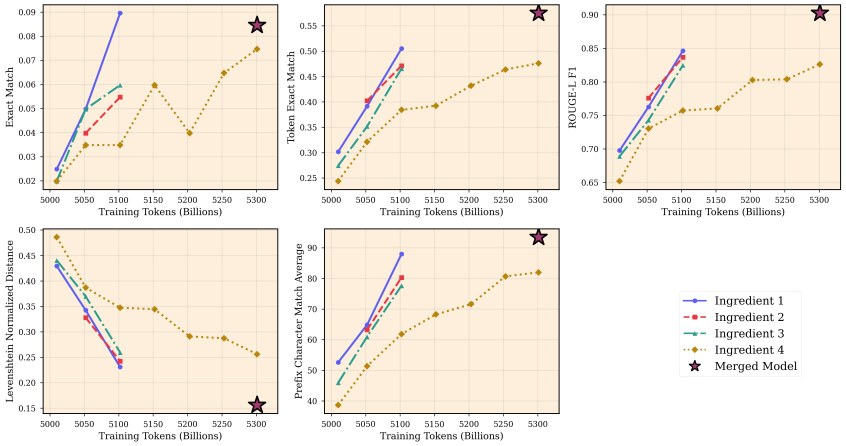


Figure 3: Impact of cooldown and merging on practical extractability metrics on the highly memorized *Wikipedia* dataset.

- **Exact Match (EM) and Token-level EM:** Binary success indicators that determine whether the generated output exactly reproduces the reference sequence at the sequence or token level.
- **Normalized Levenshtein Distance:** A similarity measure that captures near-verbatim recovery, where higher values reflect closer correspondence.
- **ROUGE-L:** A subsequence-based metric that computes the longest common non-contiguous overlap between the generated and reference texts, enabling detection of partial memorization even when exact matches are not achieved.
- **Prefix Character Match Rate:** The normalized length of the longest shared character prefix between the generated output and the reference sequence. This metric is especially sensitive to early memorization effects, where the model correctly generates an initial segment before diverging.

Higher normalized Levenshtein scores indicate greater dissimilarity from the reference sequence, whereas lower values for all the remaining metrics correspond to reduced memorization. Together, these metrics are widely used to quantify both verbatim reproduction and approximately verbatim generation (more details can be found in (Karamolegkou et al., 2023; Hans et al., 2024)).

4 RESULTS

4.1 RECENCY BIAS AND THE FORGETTING HYPOTHESIS

We first analyze the evolution of theoretical memorization, measured by negative log probability, to study how shifts in the data distribution affect retention. Figure 1a shows the training dynamics for the four probe datasets. We observe a clear contrast between early and late training data. The *English Quotes* dataset, included during pre-training but excluded from the cooldown phase, shows an initial loss decrease that slightly worsens and eventually plateaus. This supports our **Forgetting Hypothesis**, suggesting that in token-rich regimes trained beyond Chinchilla optimality, early data is overwritten unless reinforced. In contrast, the *GSM8K* dataset, introduced only during the cooldown phase after 5000B tokens, exhibits strong **recency bias**. Despite far fewer repetitions than pre-training data, its loss drops sharply and quickly reaches levels comparable to highly repeated *Wikipedia* dataset (present in both pre-training and cooldown). This indicates that the cooldown phase actively imprints the final data distribution rather than merely stabilizing training.

4.2 EXTRACTABILITY AS A PHASE TRANSITION

While theoretical memorization, as measured by loss, improves steadily over training, practical extractability exhibits strongly non linear behavior. Figure 2 reports the five attack metrics across the training trajectory. Across all metrics, we observe a clear hockey stick pattern. Throughout most of pre-training, extractability remains low despite steady reductions in loss, followed by a sharp increase during the cooldown phase. Verbatim memorization emerges abruptly and almost entirely after 5000B tokens, indicating a late stage transition in extractability behavior. For example, the *Prefix Character Match Rate* shows that the model starts reproducing the beginnings of sequences slightly before it can generate them in full, while successful completion of the sequence with $k = 50, n = 50$ emerges only at late stages, coinciding with learning rate annealing and replay of high-quality data.

4.3 THE MERGING ANOMALY: DECOUPLING THEORY FROM PRACTICE

Our most novel result arises from the analysis of model souping in Stage 3, where we compare the four individual ingredient runs with the final merged base model. Conventional intuition suggests that weight averaging smooths the loss landscape, acting as a form of regularization that should reduce overfitting and memorization. Consistent with this view, the base model generally lies near the average of the ingredient models in terms of *theoretical memorization* as measured by loss (Fig. 1b). However, this relationship breaks down when considering *practical extractability* (Fig. 3). In most cases, the merged model exhibits higher extractability than any individual ingredient run, not merely exceeding their average. This divergence reveals what we term a **Merging Anomaly**. While weight averaging effectively smooths overall loss, it appears to preserve or even reinforce the specific parameter configurations responsible for verbatim memorization. As a result, a merged model may appear less overfit under loss-based evaluations while still posing elevated privacy risks due to high extractability. This decoupling complicates safety assessments for model ensembles that rely solely on aggregate performance metrics.

5 CONCLUSION

In this work, we analyze how memorization evolves over the training lifecycle of a modern LLM. By separating theoretical memorization from practical extractability, we show that loss-based metrics alone fail to capture privacy risk. Our findings reveal a strong recency bias, with cooldown data being memorized disproportionately to its repetition frequency, and show a non-intuitive merging anomaly due to model souping. Together, these results underscore the need to evaluate memorization across training phases and caution against relying solely on aggregate loss metrics for privacy and copyright auditing.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Zeyuan Allen-Zhu. Physics of language models: Part 3.3, knowledge capacity scaling laws. In *International Conference on Learning Representations (ICLR)*, 2025.

Stella Biderman, Usvsn Prashanth, Lintang Sutawika, Hailey Schoelkopf, Quentin Anthony, Shivanshu Purohit, and Edward Raff. Emergent and predictable memorization in large language models. *Advances in Neural Information Processing Systems*, 36, 2024.

Astoria-Megler Bridge. Wikipedia, the free encyclopedia. *San Francisco (CA): Wikimedia Foundation*, 2001.

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)*, pp. 2633–2650, 2021.

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

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL `https://arxiv.org/abs/2110.14168`.

Abhimanyu Hans, Yuxin Wen, Neel Jain, John Kirchenbauer, Hamid Kazemi, Prajwal Singhania, Siddharth Singh, Gowthami Somepalli, Jonas Geiping, Abhinav Bhatele, et al. Be like a goldfish, don't memorize! mitigating memorization in generative llms. *arXiv preprint arXiv:2406.10209*, 2024.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

Antonia Karamolegkou, Jiaang Li, Li Zhou, and Anders Søgaard. Copyright violations and large language models. *arXiv preprint arXiv:2310.13771*, 2023.

Milad Nasr, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A Feder Cooper, Daphne Ippolito, Christopher A Choquette-Choo, Eric Wallace, Florian Tramèr, and Katherine Lee. Scalable extraction of training data from (production) language models. arxiv 2023. *arXiv preprint arXiv:2311.17035*.

Milad Nasr, Javier Rando, Nicholas Carlini, Christopher A Hayase, Florian Tramèr, and Katherine Lee. Scalable extraction of training data from aligned, production language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, et al. 2 olmo 2 furious. *arXiv preprint arXiv:2501.00656*, 2024.