Investigating Spaced Repetition and Memory Retention in Language Models

Alexander Chin
Allen School of Computer Science and Engineering, University of Washington
alexchin@uw.edu

Abstract

We attempt first an investigation into the field of ***catastrophic forgetting*** *in language models (GPT-2 XL and Qwen2 0.5B). Next, we explore whether spaced repetition, proven effective for human memory retention, can optimize fine-tuning schedules for large language models. Drawing from Ebbinghaus's forgetting curve, we characterize forgetting patterns in GPT-2 XL and develop scheduling strategies that mirror human spaced repetition techniques. Training GPT-2 XL and Qwen2 0.5B on TriviaQA pairs and the AG-News dataset, we conduct two experiments: (1) tracking forgetting curves as the model learns irrelevant information, and (2) comparing three training schedules—shuffled baseline, naive-spaced, and exponential-spaced repetition. Our findings reveal that medium language models do not tend to display an exponential forgetting curve akin to humans within the range of 1.5k update steps, and that exponentially-spaced training schedules slightly outperform both random shuffling and uniform spacing, achieving 50% accucracy versus the 23% achieved after 1.5k fine-tuning steps for shuffled baselines. For domain-specific fine-tuning, this novel approach shows promise for while maintaining performance, offering significant efficiency gains for continual learning in neural networks.*

1. Introduction

Hermann Ebbinghaus discovered in 1885 that human memory follows exponential decay—the forgetting curve—showing rapid information loss without reinforcement [1]. This insight led to spaced repetition, where information is reviewed at increasing intervals to promote long-term retention.

Large language models face catastrophic forgetting when adapting to new tasks—losing previously learned information when acquiring new knowledge [2, 3]. Can principles from human learning psychology inform better training strategies for neural networks?

We systematically apply spaced repetition principles to language model fine-tuning. Unlike previous work focusing on online adaptive curriculum learning [10], which is computationally more expensive, as it requires evaluation steps *during* training, we explore the frontier of *pre-scheduling* training data to mitigate forgetting. Our experiments reveal that GPT-2 XL does not display similar forgetting curves to humans, though models trained using exponentially-spaced repetition schedules initially retain information more effectively than those using uniform spacing or random shuffling.

2. Related Work

Human Memory and Spaced Repetition. Ebbinghaus's work [1] revealed exponential forgetting patterns, showing memory retention drops to 40% within days. Modern algorithms like SuperMemo [4] and Pimsleur's intervals [5] operationalize these insights by scheduling reviews at exponentially increasing intervals.

Catastrophic Forgetting in Neural Networks. McCloskey and Cohen [2] identified catastrophic forgetting in connectionist models. Recent work shows this persists in modern architectures—Luo et al. [6] demonstrated that LLMs from 1B to 7B parameters exhibit catastrophic forgetting during continual fine-tuning, with larger models showing more severe forgetting.

Mitigation Strategies. Elastic Weight Consolidation (EWC) [3] protects important weights using quadratic penalties. Experience replay methods [7] retain samples from previous tasks. Parameter-efficient methods like LoRA [8] update only small parameter subsets, though studies show even these approaches suffer from forgetting [9].

Spaced Repetition in ML. Amiri et al. [10] pioneered applying spaced repetition to neural networks with ”Repeat before Forgetting,” achieving 2.9-4.8x speedup while using only 34-50% of data per epoch. However, dynamic methods rely on online inference while training, since adaptive scheduling requires live feedback from the model. Furthermore, it is not yet clear whether or not this benefit generalizes to transformer-based language models. Recent work [11] applied spaced scheduling to instruction tuning, showing more balanced performance. However, these approaches have not systematically characterized forgetting dynamics or developed principled scheduling algorithms based on measured forgetting curves. To this end, prescheduling using static scheduling methods may still yield improvements in data efficiency and information retention.

3. Methods

3.1. Model and Data Preparation

We use two models of different sizes. The first, GPT-2 XL (1.5B parameters), we use to investigate LM forgetting curves on extensive non-overlapping data sourced from the same dataset. The second, Qwen2 0.5B, we use to investigate different scheduling methods, while incorporating a different dataset to mitigate the risk of dataset-specific generalization.

Our first dataset consists of TriviaQA question-answer pairs. For training, we augment the data to to 10 paraphrased versions each using Together AI's language model. For example, the QA pair:

Q: What is the capital of France?

A: Paris

Is augmented to variations like:

”The capital of France is Paris”

”Paris is the capital city of France”

”France's capital is Paris”

This augmentation ensures that the information learned is not merely specific to the QA format, but robust learning across diverse phrasings.

The second dataset was sourced from AG's News' headlines, serving as a completely unrelated source of text. This was necessary because despite being trained on rephrased versions of the original QA pair, we feared that the model was learning to better answer ”trivia-style” questions by being trained on specific facts.

Training hyperparameters: constant learning rate 5×10^{-5} with no warmup or decay, batch size 8 for both eval and train, AdamW optimizer with weight decay 0.01, evaluation every 10 batches on full eval dataset with few-shot prompting.

3.2. Experiment 1: Characterizing the Forgetting Curve

Phase 1 - Identifying Known Facts: We first establish a baseline of facts the pre-trained models (GPT2-XL) models already know. From TriviaQA, we randomly sample 5,000 question-answer pairs. For each fact, we:

- Format as ”Q: [question] A:” and generate completions with temperature 0.7
- Repeat generation 3 times to account for stochasticity
- Mark as ”known” only if all 3 generations match the correct answer
- This stringent criterion yielded 160 reliably known facts for GPT2-XL, and 200 for Qwen-0.5B.

Phase 2 - Inducing Forgetting: To measure forgetting, we fine-tune the model on a set of ”irrelevant” pieces of data. For GPT2-xl, we train on TriviaQA pairs disjoint from our evaluation set, i.e. ones that the pretrained GPT2-xl model was not evaluated on. For Qwen-0.5B, we train on the completely unrelated AG-News dataset. Training proceeds for 4 epochs (a single epoch is 1000 update steps with batch size 8), and 1 epoch (a single epoch is 1,500 update steps with batch size 8) for GPT and Qwen, respectively.

Evaluation Protocol: Every 10 batches (80 facts), we pause training and evaluate exact-match accuracy on all 160 known facts. We use temperature 0 for deterministic evaluation, measuring the percentage of facts still correctly answered. This creates a time-series of retention values $R(t)$.

3.3. Experiment 2: Spaced Repetition Schedules for Qwen-0.5B

Dataset Construction: We randomly select 400 ”target facts” (information we want the model to learn and retain) and 8000 ”non-target facts” (background training data to simulate continual learning). The target facts serve as both evaluation and training data. We duplicate the target facts five times throughout the scheduling process, yielding a total of 400 * 5 = 2000 repeated target facts. We also select 2000 facts disjoint from the 400 target facts as *primer facts* - ones that the model is initially trained on to reduce the risk of QA-specific model generalization. The primer procedure arose out of an abundance of caution; since we did not observe obvious forgetting in phase one, despite data augmentation, we feared that the model retained its accuracy by virtue of maintaining (or increasing) performance on ”trivia-style” facts. The hope is that through these 2000 initial training examples, the model has time to initially learn the ”trivia-style” questions, meaning that any performance increase observed later on target can be attributed to *specific learning* of those facts rather than general performance on QA-style answers. During schedule creation, each fact is repeatedly sampled from its 10 paraphrases. Each schedule consists of:

$$(400 * 5) \text{ target } + 8000 \text{ non-target } + 2000 \text{ primer}$$

$$= 12000 \text{ total data points.}$$

Schedule Implementations:

1. **Shuffled (Baseline):** Priming on 2000 samples, then the remaining 10000 examples are pooled and shuffled. Training proceeds by sampling without replacement until exhausted. This mimics standard training practices.

2. **Naive-Spaced:** Priming on 2000 samples, then an alternating pattern: present all 400 target examples, then select 1600 non-target examples without replacement. This pattern repeats 5 times. No shuffling occurs within the target examples within blocks.

3. **Exponential-Spaced:** Target blocks appear with exponentially increasing gaps, with our spaced-repetition gap equation specified as follows:

   $$s_i = \left\lfloor \frac{e^{\alpha i} + b}{\sum_{j=1}^{n}(e^{\alpha j} + b)} \cdot d \right\rceil \quad \text{for } i = 1, 2, \ldots, n$$

   where s_i is the i-th normalized sequence element, α is the exponential growth rate, b is a constant additive bias applied before normalization, d is the number of non-target data points, and n is the number of blocks. For our experiment, our parameters are as follows:

   $$\alpha = \ln(1.5), \quad b = 0, \quad d = 8000, \quad i \in \{1, 2, 3, 4, 5\}$$

   This yields a normalized exponential sequence where the base of the exponent is 1.5, adding to a total of 8000. This equation yielded a sequence of:

   $$[765, 1020, 1402, 1976, 2837]$$

   for our blocks of non-target data sampled without replacement from our pool of 8000, each preceded by a block of 400 target data. The schedule is again primed with 2000 QA examples for a total of 12000 training samples.

Training and Evaluation: Models all start from the same pre-trained checkpoint. During evaluation steps, occurring every 10 batches (every 80 samples with a batch size of 8), we evaluate the model using a few-shot prompt initialized with three QA-pairs disjoint from the training set, for example:

```
Question: In what year did Malaysia
receive its independence?
Answer: 1957
Question: What is the capital of France?
Answer: Paris
Question: What UN secretary went to
Harvard?
Answer: General Ban Ki-moon
Question: Which Japanese island that
has its capital at Sapporo is the
traditional home of the Ainu people?
Answer: {model response}
```

We then calculate the following metrics over all 400 target data points:

1) Exact substring-match accuracy of model generation over 32 tokens (using only the original answer string, not paraphrases). 2) Average per-sample negative log-likelihood of model over exclusively answer tokens.

3.4. Sample size and training logistics:

Due to limited compute resources, we ran each experiment a max of two times, with a different random seed each time. Between experiments comparing different schedules, we keep both the same random seed and the same order of primer and target data to reduce cross-trial variance. Experiments were run on a single RTX 4090 GPU for Qwen-0.5B and an A100 PCIe for GPT2-XL. We did not test scheduling on GPT2-XL due to financial hurdles associated with the larger model size. Both models were fine-tuned with full precision FP32 on experimental schedules.

4. Results and Efficiency Analysis

4.1. Forgetting Curve Analysis

Our analysis of GPT-2 XL's forgetting behavior reveals that at least within the short fine-tuning schedule of 4000 weight update steps (32000 augmented TriviaQA data points), there is little-to-no observable catastrophic forgetting. Rather, the model's performance on the eval set disjoint from the training data seems to converge to around 65% over multiple trials, as seen in Figure 1. This suggests that models are robust to forgetting, at least when fine-tuning on data from a similar source. Despite the data augmentation, which should reduce the degree to which training on other facts helps the model learn QA-style data, we hypothesize that GPT2-xl remains robust to catastrophic forgetting because a) the data it was trained on comes from a similar distribution to the eval data, which means that performance should stay stable, or b) it resists forgetting by generalizing weakly to trivia-style facts. However, we do observe some instability in the first 300 batches, perhaps as the model adjusts to the new distribution of training data. Soon, however, the model converges to constant performance not far from its initial performance.

We also tested a naive-spaced method on GPT2-xl to contrast with training schedules containing exclusively irrelevant data. We see a clear difference; the naive-spaced schedule (exposure to target, eval data every 1k batches) manages to bump the accuracy up to around 85% within 4k updates.

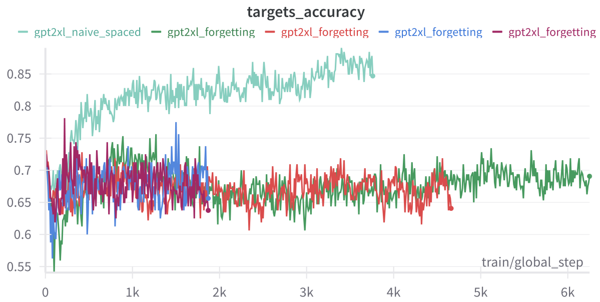

Figure 1. GPT2-xl accuracy over update steps

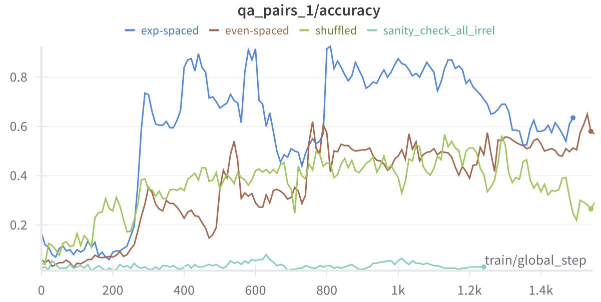

Figure 2. Qwen accuracy over update steps

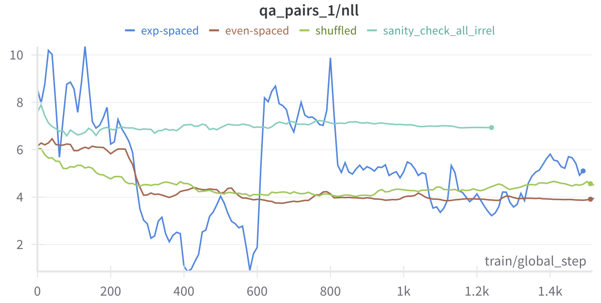

Figure 3. Qwen NLL over update steps

4.2. Scheduling Effectiveness

The three scheduling strategies demonstrate markedly different learning dynamics. The shuffled baseline shows steady but slow improvement in negative log-likelihood throughout training, comparable to that of the naively-spaced schedule, but with a small decrease in accuracy past 1.2k steps, ultimately achieving approximately 30% accuracy on target facts. This is a surprising result that suggests that perhaps spacing out target data has an inherently positive effect on model retention.

The naive-spaced schedule achieves notably better final performance at around 58% end-of-training accuracy. However, its learning curve exhibits characteristic sawtooth patterns that reveal the limitation of uniform spacing. Performance seems to initially spike following each target block presentation, then gradually flatten out or decay during the subsequent non-target training phase. However, this pattern becomes less noticeable as training progresses.

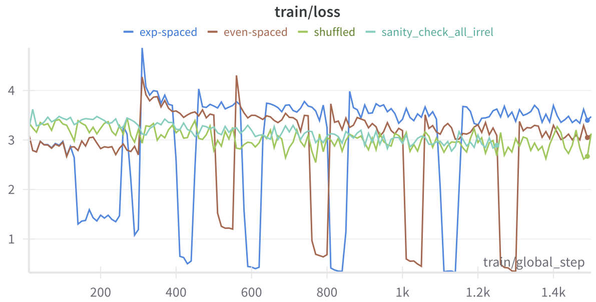

Figure 4. Qwen loss over training steps

Furthermore, we observe a relationship between current performance and learning efficiency in the naive schedule, as seen in Figure 2. Early in training, when baseline accuracy is low, each target block produces substantial improvements. As training progresses and performance increases, the marginal benefit of each repetition seems to decrease. This diminishing returns pattern weakly supports our theoretical model suggesting that learning effectiveness is inversely proportional to current retention. However, the inherent stochasticity in single-trial experiments warrants more investigation into this hypothesis.

Most significantly, the exponential-spaced schedule demonstrates superior retention, especially in the early stages of training, as expected. By the end of training, however, it reaches comparable or slightly better accuracy than the evenly-spaced schedule, reaching a final accuracy of around 64%. Remarkably, we noted that early exposure seems to have a marked increase in accuracy, with the model quickly jumping to an accuracy of 65% after only being exposed to relevant information once. Future exposures, however, seem to have less of an impact, and in fact we observe a later decay in accuracy.

Finally, for contrast, we tested the model on a schedule entirely composed of irrelevant data. As expected, it remains both high in negative-log-likelihood and low in accuracy, never pushing past 4% accuracy on the evaluation set.

Results remain inconclusive for whether scheduling also progressively increases stable performance between target block presentations; initial results seem to suggest that there may be benefits to earlier frequent exposures, but forgetting curves for this short fine-tuning process are noisy. Negative log-likelihood, especially on the exponentially spaced schedule seems to display drastic jumps, especially at around 600 steps where accuracy takes a plummet and negative log-likelihood rapidly jumps (Figure 3). Whether this is due to forgetting or mere training stochasticity is not entirely clear; in any case, a distinct pattern in forgetting curves for progressively spaced exposures is unlikely to be found in just one trial.

5. Discussion

5.1. Efficiency for Domain-Specific Fine-Tuning

The implications for practical applications, particularly domain-specific fine-tuning, are substantial. When adapting models to specialized domains using fine-tuning, as is often done with datasets like medical terminology, legal concepts, or proprietary knowledge bases, exponential spacing seems to have a marked impact on initial accuracy while maintaining comparable final performance. This can perhaps be useful for models in which immediate and time-constrained performance is a necessity, or where compute is initially limited. Despite later repetitions becoming more sparse, we find that final accuracy remains unimpaired; a promising result that shows that early accuracy need not come at the cost of later forgetting when trained on more data.

The approach shows exceptional promise for scenarios with limited training data and/or compute. Many domain-specific applications must work with datasets containing fewer than 10,000 examples, like those used in this paper. In these constrained settings, spaced repetition seems to achieve better retention than compared to standard random shuffling by maximizing the learning signal extracted from each training example.

However, our findings demonstrate that medium-sized language models, at least GPT2-xl in initial fine-tuning, do not exhibit catastrophic forgetting on similar training data, indicating that catastrophic forgetting may not be as large of a problem as initially thought. Despite this, traditional fine-tuning approaches often resort to overtraining to ensure adequate retention of critical domain knowledge. Our spaced scheduling shows that repeated redundant exposures over time are not necessary, and in fact even may result in worse final performance. The success of exponentially-spaced repetition requires more research, but is initially promising.

Limitations: We focus on factual knowledge retention; forgetting curves may vary for reasoning or generation capabilities. Furthermore, our data incorporates very few trials and thus is quite noisy. Future work should explore the same methods on larger models and different datasets, perhaps incorporating adaptive spacing functions and integration with other continual learning techniques.

References

[1] Hermann Ebbinghaus. Memory: A Contribution to Experimental Psychology. Teachers College, Columbia University, 1885. 1

[2] Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. Psychology of Learning and Motivation, 24:109–165, 1989. 1

[3] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, et al. Overcoming catastrophic forgetting in neural networks. Proceedings of the National Academy of Sciences, 114(13):3521–3526, 2017. 1

[4] Piotr Wozniak. Optimization of learning. Master's thesis, University of Technology in Poznan, 1990. 1

[5] Paul Pimsleur. A memory schedule. Modern Language Journal, 51(2):73–75, 1967. 1

[6] Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. arXiv preprint arXiv:2308.08747, 2023. 1

[7] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. Advances in Neural Information Processing Systems, 32, 2019. 1

[8] Edward J. Hu, Yelong Shen, Phillip Wallis, et al. LoRA: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685, 2021. 1

[9] Stella Biderman, USVSN Sai Prashanth, Lintang Sutawika, et al. LoRA learns less and forgets less. arXiv preprint arXiv:2405.09673, 2024. 1

[10] Hadi Amiri, Timothy Miller, and Guergana Savova. Repeat before forgetting: Spaced repetition for efficient and effective training of neural networks. In Proceedings of EMNLP, pages 2401–2410, 2017. 1

[11] Anonymous Authors. Spaced scheduling enhances instruction-prompted reasoning in large language models. OpenReview preprint, 2024. 1