

Forgetting Curve: A Reliable Method for Evaluating Memorization Capability for Long-context Models

Xinyu Liu^{1,*}, Runsong Zhao^{1,*}, Pengcheng Huang¹, Chunyang Xiao,
Bei Li², Jingang Wang², Tong Xiao^{1,3}, Jingbo Zhu^{1,3,†}

¹ NLP Lab, School of Computer Science and Engineering, Northeastern University, Shenyang, China

² Meituan Inc. ³ NiuTrans Research, Shenyang, China

{2310737, 2472148, 2201802}@stu.neu.edu.cn, chunyangx@gmail.com

{libei17, wangjingang02}@meituan.com, {xiaotong, zhujingbo}@mail.neu.edu.cn

Abstract

Numerous recent works target to extend effective context length for language models and various methods, tasks and benchmarks exist to measure model’s effective memorization length. However, through thorough investigations, we find limitations for currently existing evaluations on model’s memorization capability. We provide an extensive survey for limitations in this work and propose a new method called forgetting curve to measure the memorization capability of long-context models. We show that forgetting curve has the advantage of being robust to the tested corpus and the experimental settings, of not relying on prompts and can be applied to any model size.

We apply our forgetting curve to a large variety of models involving both transformer and RNN/SSM based architectures. Our measurement provides empirical evidence for the effectiveness of transformer extension techniques while raises questions for the effective length of RNN/SSM based models. We also examine the difference between our measurement and existing benchmarks as well as popular metrics for various models. Our code and results can be found at <https://github.com/lazybug/ForgettingCurve>.

1 Introduction

Arguably driven by large language model (LLM) practical needs, numerous works today aim to extend LLM contexts. The approaches range from modifying existing transformer architectures (Dai et al., 2019; Wu et al., 2022; Munkhdalai et al., 2024), training existing LLMs to extend their contexts (Chen et al., 2023b; Xiong et al., 2023; Chen et al., 2023a), to more recent modelling based on fundamental architectural changes using RNNs or hybrid architectures (Gu and Dao, 2023; Peng et al., 2023; Lieber et al., 2024), etc. All of the

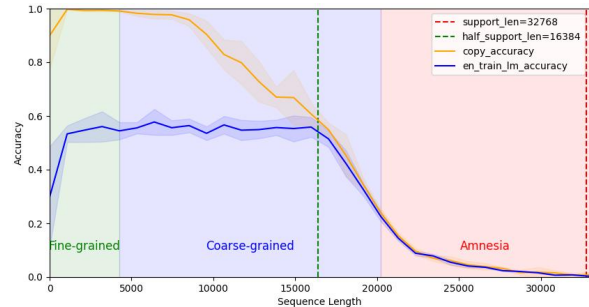


Figure 1: The forgetting curve of Llama-2-base-32k (Together.AI, 2024). The x-axis denotes the prefix length. Green, blue, and red areas respectively indicate fine-grained memory where the model achieves 99% token replication accuracy (except for very short sequences), coarse-grained memory where copy accuracy surpasses LM accuracy, and the amnesia area where the model completely ignores the prefix.

approaches claim to significantly extend the context window of corresponding baseline models with empirical validations. However, to authors’ best knowledge, the community has not established (and thus has not agreed on) a standard metric that can reflect model’s inherent memory with respect to the long context, in other words, model’s effective memory length.

The most commonly used metric for evaluating a model’s natural language generation ability is Perplexity (PPL). It is generally assumed that a superior long-range perplexity suggests a more effective utilization of long context. However, recent findings indicate that a lower perplexity for long context does not necessarily lead to improved performance in downstream long context applications (Hu et al., 2024b). This raises questions about the validity of using perplexity as an indicator of a model’s long-context memorization capability.¹ Alternative metrics have been proposed to evaluate a model’s long-term memory, including “Needle in a

*These authors contributed equally to this work.

†Corresponding author.

¹The same paper suggests that perplexity effectively indicates model performance on short rather than long sequences.

Haystack” (gkamradt, 2023) and its derived metrics (Song et al., 2024). However, these metrics also present their own limitations. For instance, “Needle in a Haystack” is highly sensitive to the prompt used,² and such sensitivity is attributed more to the varying instruction-following capabilities across different models, rather than differences in their memory capabilities. Other metrics (Tay et al., 2020; Bulatov et al., 2022, 2023), which involve toy tasks, face challenges in generalizing to real-world settings and are susceptible to severe overfitting. We survey a significant number of popular metrics and benchmarks which measure model long-term memory in this work.

To address these limitations and provide a reliable measure of a model’s long-term memory, we introduce a new method called *forgetting curve* in this paper. Our method also begins with a language model trained on a real-world corpus. We then exploit the model’s emerging copy capability, assessing whether the trained model can replicate its prefix as part of the completion process. In order to account for instances where the token prediction relies solely on the model’s natural language modelling (LM) capabilities instead of long-term memorization, we additionally plot the LM accuracy curve. Figure 1 illustrates our method, demonstrating the progression of a memory pattern from fine-grained to coarse-grained memory, and ultimately to complete memory absence.

The paper is organized as the following. We start with a critical evaluation of the limitations inherent in the metrics and tasks currently employed to measure a model’s long-term memorization capability (Section 2). Following this, we detail the construction of the forgetting curve, highlighting its dual function in visualizing memory and enabling statistical calculation of memory length. We also establish the reliability of our method in assessing a model’s memorization capability (Section 3). Using our forgetting curve, we comprehensively examine the long-range memorization capabilities of various current models and architectures (Section 4). We demonstrate that the forgetting curves effectively illustrate a model’s memorization capability through visualizations, revealing varying patterns across different architectures and models. We also validate certain context extension techniques, and raise questions about some architectural claims.

²As demonstrated in experiments by Claude (Anthropic, 2023) and Kimi (Moonshot.AI, 2023).

Finally in analysis (Section 5), we show that forgetting curves can be applied to models of different sizes under different settings. Compared to existing effective length measurements (Hsieh et al., 2024) or perplexity, the forgetting curve provides a self-contained, memory-focused measurement.

2 Limitations of Long-Range Memory Measures

Limited Memory Usage (LMU) The first limitation that we identify in the long-range measurement literature is limited memory usage (LMU), where the measurement or metric does not have a strong relationship with model’s capability to leverage long context. Perplexity is a surprising example of LMU. Recent works (Chen et al., 2023b; Yang, 2023; Xiao et al., 2023) typically demonstrate their model’s long-range capability by showing a stably low perplexity given long contexts. However, Hu et al. (2024b) recently found that lower perplexity does not imply an improved downstream performance on long-context tasks. In section 5, we further demonstrate, using forgetting curves, that some models can improve long-range perplexity without enhancing long-range memory. LMU also manifests in popular long-range measurement tasks such as “Needle in a Haystack”(gkamradt, 2023) which assesses the capability to retrieve a specific piece of information from lengthy distracting texts. The memory usage is clearly limited since the model can theoretically achieve great performance by focusing only on the ‘needle’ and ignoring all the long context. Methods derived from “Needle in a Haystack” like Counting Stars (Song et al., 2024) alleviate this drawback but still inherit such limitations.

Prompt Required (PR) The second limitation in existing evaluation methods is their strong dependency on the exact prompts during evaluation. For instance, in the Claude 2.1 experiments (Anthropic, 2023), a single prompt change dramatically improved the model’s ability to find the ‘needle’, boosting Claude 2.1’s task success rate from 27% to 98% in the original evaluation. Such significant dependency on prompts raises the following two issues. Firstly, it makes the comparisons between models extremely difficult due to the large performance variance across different prompts.³

³Treating all prompts as i.i.d experiments allows for theoretical comparison of two models using statistical tests. However, large variance makes achieving statistical significance

Secondly, it is challenging to apply these metrics to unaligned models and smaller ones due to their limited capability to follow instructions to find the ‘needle’. This implies that new architectures need to be scaled up and aligned before these metrics can be applied, preventing rapid and low-cost experimental validation of a model’s long-context capability. Although latest benchmarks such as Ruler (Hsieh et al., 2024) and StreamEval (Xiao et al., 2023) can force a model to produce answers through carefully designed prompts, this approach is still cumbersome and model-specific.

Inference Factors (IF) The third limitation we identify is that current long-context evaluations often fail to distinguish the influence of inherent inference factors, such as natural language understanding ability, etc. For instance, in the “Needle in a Haystack” task where the model is prompted to locate the needle, a failure can be difficult to diagnose, as it could be due to a deficiency in either the model’s long-context memorization capability or its natural language understanding capability. Similarly, failure in tasks such as long context question answering or long mathematical expression calculation may not reflect shortcomings in the model’s long-term memory, but rather, deficiencies in its specific abilities.

Limited Context (LC) The fourth limitation arises from the limited context inherent in benchmarks. Some benchmarks (Li et al., 2024; Zhang et al., 2024; Li et al., 2023b) conduct their tests within a confined context, which restricts their ability to evaluate model performance in longer contexts. Expanding these benchmarks to include extended contexts is typically laborious and time-intensive, posing a challenge to keep pace with the rapid increase in context length that large language models can handle.

Training on Toy Tasks (ToTT) Finally, we notice that many recent studies demonstrate a model’s long-range capability by training on toy tasks (Tay et al., 2020; Gu and Dao, 2023; Bulatov et al., 2023). While we recognize its contributions, we also argue that because long-context memory is a capability that matters in real applications such as long document summarization (Fan et al., 2019), role playing (Li et al., 2023a; Wang et al., 2023) etc., the tests should be performed on real data. In other words, we argue that long-context capability challenging. Furthermore, the i.i.d assumption is questionable.

Metrics/Tests/Tasks/Benchmarks	Limitations
Perplexity	LMU
Needle In A Haystack (gkamradt, 2023)	LMU, PR, IF
Counting Starts (Song et al., 2024)	LMU, PR, IF
StreamEval (Xiao et al., 2023)	LMU, PR
Passkeys (Munkhdalai et al., 2024)	ToTT, LMU, PR, IF
Selective Copy (Gu and Dao, 2023)	ToTT, LMU
Copy (Bulatov et al., 2022)	ToTT
LRA (Tay et al., 2020)	ToTT, LC
BABILong (Kuratov et al., 2024)	LMU, PR, IF
LongICLBench (Li et al., 2024)	PR, IF
Ruler (Hsieh et al., 2024)	LMU, PR, IF
LongBench (Bai et al., 2023)	LC, PR, IF
InfiniteBench (Zhang et al., 2024)	LC, LMU, PR, IF
LooGLE (Li et al., 2023b)	LC, PR, IF
ZeroSCROLLS (Shaham et al., 2023)	LC, PR, IF
L-Eval (An et al., 2023)	LC, PR, IF
BAMBOO (Dong et al., 2023)	LC, PR, IF
LV-Eval (Yuan et al., 2024)	LC, LMU, PR, IF
CLongEval (Qiu et al., 2024)	LC, LMU, PR, IF

Table 1: Overview of long-context measurements and limitations according to the categorization in Section 2.

should be an emerging capability to be measured, which does not involve any specific training, especially not any training on toy datasets. Ideally, the measurement should also be on a real dataset to mitigate issues like overfitting and data leakage (Wu et al., 2023; Balloccu et al., 2024).

Table 1 lists popular methods for assessing a model’s long-context performance along with their limitations. It is important to note that these limitations are often interrelated. Ideally, assessment methods should overcome all the listed limitations to provide an accurate measure of a model’s performance.

3 The Forgetting Curve

In this work, we introduce a novel method called “forgetting curve” to measure a model’s inherent long-context memorization capability. We show that the forgetting curve effectively mitigates the previously listed limitations and remains robust under various experimental settings. This makes it an ideal method for characterizing and visualizing a model’s long-context memorization capability.

3.1 Evaluation Methodology

The forgetting curve is derived from LLM copy tasks and consists of two curves (i.e copy accuracy curve and LM accuracy curve) as shown in Figure 1. For both curves, teacher forcing is applied to measure the next token prediction accuracy. We refer to teacher forcing the setting widely used to measure language model perplexity where the next token prediction probability is taken into account

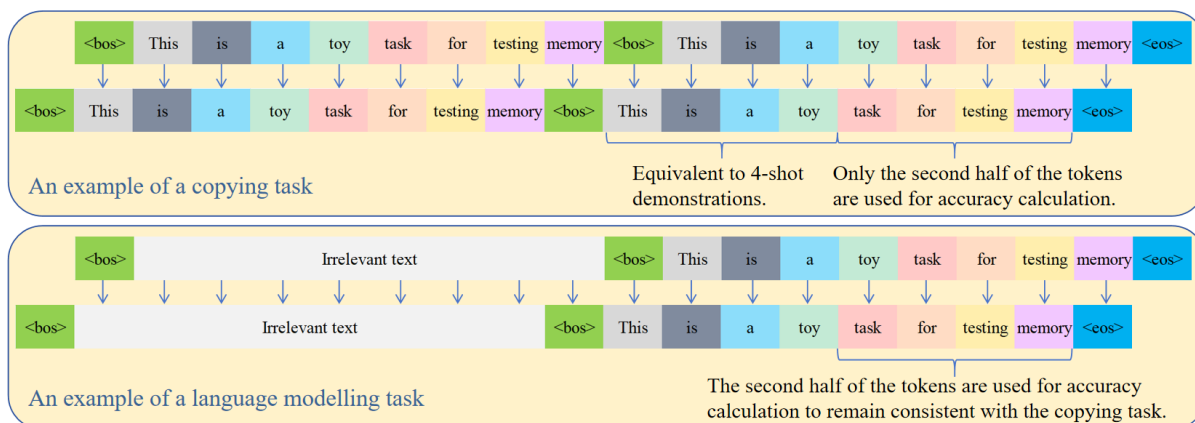


Figure 2: The forgetting curve task measures the LLM prediction accuracy for the target sequence “This is a toy task for testing memory” under two settings. The above figure illustrates the copy setting, while the below one shows the language modelling setting. We calculate the difference between these two settings to obtain the forgetting curve reflecting model’s memory behaviour. As shown in the figure, only the later half of the tokens are taken into account to construct the forgetting curve.

by always assuming a correct prefix derived from a corpus. We only differ from perplexity measurement in that we measure token prediction accuracy (i.e. 0 or 1 for each token) instead of log probability scores for both copy accuracy curve and LM accuracy curve. We further detail the measurement of the two curves below.

3.2 Evaluation Tasks

Copy Accuracy Curve. The first curve, depicted as the yellow curve in Figure 1, is the copy curve. This curve assesses an LLM based on its copy accuracy. We calculate this accuracy by tasking the model to predict the second half of a copied string using its first half, as illustrated at the top of Figure 2. One key difference from the standard copy task is our requirement for the model to be *not trained* on such tasks, thus treating copy as an emergent capability of the LLM. Such constraints avoid trivial solutions which overfit to the copy tasks, as pointed out by Gu and Dao (2023). Secondly, instead of using natural language prompts to trigger the LLM’s copy behavior, we employ teacher forcing and measure the resulting predicted copy accuracy. The initial tokens are disregarded from the measurement as they are mainly used as few-shot learning examples to inform the model of the copy task.⁴ In our experiments, all curve measurements begin with the copy performance for the second half of the target sequence.

⁴Remark that such way of indicating copy task (by showing the model the copied tokens) effectively avoids relying on model’s understanding capability, making our method applicable for unaligned/smaller models as we show in Section 5.

LM Accuracy Curve. In the forgetting curve, we choose real-life natural language texts to avoid trivial or overfitting solutions to the copy task. However, this choice also implies a non-negligible probability of correctly predicting the token even without access to any long-range contexts.⁵ To take this probability into account, the forgetting curve also plots the language modelling accuracy, represented by the blue curve in Figure 1. Specifically, an irrelevant but real text of the same length as the copy target is chosen.⁶ We measure the language model’s accuracy in predicting the later half of the target sequence, using the irrelevant prefix and the first half of the sequence, as depicted in Figure 2.

3.3 An Illustrative Example

We run Llama-2-base-32k model (Together.AI, 2024) using test text from PG-19 (Rae et al., 2019) to produce a forgetting curve (Figure 1). In these experiments, we compute an accuracy score every 1K tokens for both the copy curve and the language model curve. Each accuracy is calculated 10 times, allowing us to plot the mean accuracy and the variance in the curve.

We visualize the corresponding forgetting curve in Figure 1. From the first accuracy data point (1K), we observe that Llama-2-base-32k performs the copy task perfectly without explicit instruc-

⁵One might notice that this is the same issue we highlighted in Section 2 regarding perplexity.

⁶In Section 3.4, we demonstrate that the distribution used to sample the random string does not impact the language model accuracy.

Setting	Text source	Distribution
en_train	PG19-train	ID(En)
en_test	PG19-test	ID(En)
zh	LooksJuicy/ruozhiba	ID(Zh)
random	Random	OOD

Table 2: We extract both irrelevant and target copying text from various sources. All texts from PG-19 and Ruozhiba, which includes English and generated Chinese responses by GPT-4 respectively, are natural language and hence considered in-distribution. In contrast, a random token sequence, not adhering to any natural language distribution, is deemed out-of-distribution.

tion, a phase lasting until 4K.⁷ We refer to this phase as *fine-grained memory*, as the model can perfectly memorize past tokens. After this phase, the model can no longer precisely remember all the previous context, as shown by the copy accuracy curve. However, the model still exhibits significantly higher copy accuracy. We refer to this phase as *coarse-grained memory*.⁸ For Llama-2-base-32k, we notice that during the coarse-grained memory phase, the copy performance gradually degrades to the level of language modeling (LM) accuracy.⁹ Finally, we observe no difference in the model’s performance based on whether the target sequence has been previously seen by the model. We refer to this as the model entering the *amnesia* phase.

3.4 In-depth Analyses on Forgetting Curves

In this subsection, we critically assess the reliability of the forgetting curve as an indicator of a model’s memorization capacity. Our evaluation focuses on two primary aspects: ❶ the effect of using various irrelevant prefixes when measuring language modeling accuracy, and ❷ the impact of testing on different text corpora. Moreover, we investigate whether the forgetting curve mitigates the limitations highlighted in Section 2.

⁷We remark that the copy accuracy is not 100% for the first measurement. The phenomenon is universal (see Appendix C for the forgetting curves of other models we measure) and we believe it is because at this stage, with relatively few examples demonstrating the copy behaviour, the model does not learn to copy perfectly yet.

⁸In our setting, fine-grained memory length is where the model can accurately copy more than 99% tokens, while coarse-grained memory length is where copy accuracy exceeds language model accuracy by more than 1%.

⁹Note that in ideal case, one would expect coarse-grained memory to last much longer than model’s claimed context length.

LM Accuracy with Various Irrelevant Prefixes

To examine the effect of various irrelevant texts on the language model accuracy, we experiment with varying irrelevant text prefixes, ranging from in-distribution (ID) to out-of-distribution (OOD) relative to the Llama-2-base model. We source these irrelevant texts from four distinct text corpora, as detailed in Table 2, with the PG-19 test texts serving as the copy target in all experiments.

The results are illustrated in Figure 3a. Except for the random prefix setting, all other settings yield similar language modeling accuracy, showing a significant downturn around half the claimed context length. These trends remain statistically indistinguishable up to the memorization threshold, as confirmed by our statistical analysis in Appendix A. Importantly, these observations suggest that the model’s coarse-grained memory length is consistent across different prefix settings, demonstrating the robustness of the forgetting curve as a measure of memorization.

Forgetting Curves with Various Copy Targets

We then assess the impact of copy text on the forgetting curve, using different copy targets sourced according to Table 2. The forgetting curves, depicted in Figure 3b, show uniform trends across various settings, except when random targets are used. This uniformity in forgetting curves underscores a consistent coarse-grained memory length, highlighting the measure’s reliability across different textual contexts.

Our results confirm that the forgetting curve is a reliable measure of memorization, even when we change the irrelevant text at the beginning or the texts used in the copying task. This consistency holds as long as the texts come from well-organized sources and not just random samples. Essentially, the forgetting curve proves to be more dependable than expected, consistently showing similar patterns and lengths of memorization across different tests.

Analysis of Reliability Our approach effectively addresses the limitations aforementioned in Section 2. By requiring the language model to replicate all seen content in the copy task, we overcome the limitation of **LMU**. The inherent copy capabilities of language models, leveraged in our methodology, allow for the completion of the copy task using the copy text itself as a contextual cue, thereby circumventing the need for task-specific training **ToTT** or specialized prompts **PR**. Furthermore, the

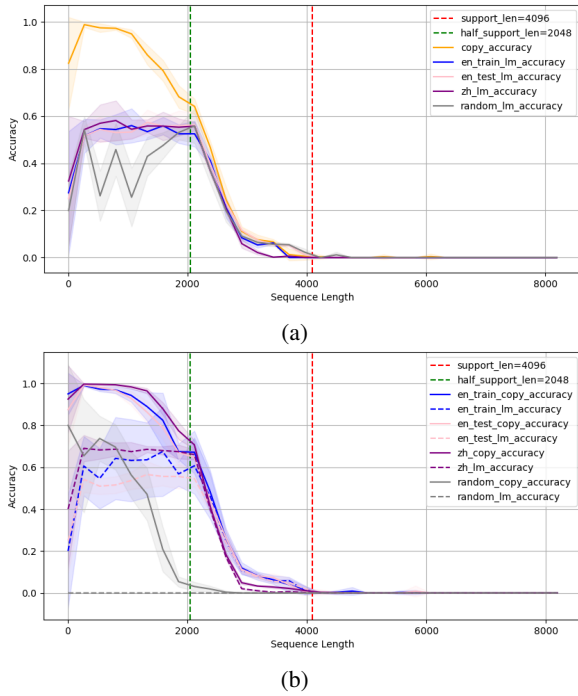


Figure 3: Llama-2-7b forgetting curve with various text sources. (a) Various irrelevant text sources, and copy text is sourced from pg19 test set. (b) Various copy text sources, and irrelevant text is sourced from pg19 test set.

ability to construct a forgetting curve of arbitrary length eliminates the LC limitation. Finally, by comparing the language modeling accuracy and excluding prompt words from our evaluations, we minimize the impact of inherent inference factors **IF**.

4 Experiments

Experimental Settings We experiment with in total 14 open-source long-context LLMs and trace their forgetting curves to examine their memorization capability; the detailed information about the experimented models is included in Appendix B. The forgetting curve does not require natural language understanding capability and can be applied to any language models, allowing us to cover diverse model type (aligned or not aligned), model architecture (transformers or RNNs), and claimed context lengths (32k to 1M). For all models, we pull the model from their huggingface releases and run the inference in BFloat16 on 8 NVIDIA A100 GPUs. Both the copy accuracy curve and the LM accuracy curve are based on token prediction accuracy which is derived from the teacher forcing setting explained in subsection 3.1.

We utilize the PG19 test set (Rae et al., 2019)

containing a collection of 100 books. Each book is individually tokenized based on the tokenizer settings of the model under test. To measure context of arbitrary length, we concatenate all books into an extremely long sequence. Subsequently, we divide the maximum supported length l , as claimed by the model, into n test lengths where n is a fixed granularity (i.e we test model on length $(\frac{l}{n}, \frac{2l}{n}, \dots, l)$ respectively).¹⁰ For each test length, we randomly select a continuous sequence from the concatenated text as a copy target while the remaining part is used to sample the irrelevant text to measure LM accuracy curve. As shown in Figure 2, the copy task is presented to LM model in the form of $[bos]S[bos]S[EOS]$ where S is the target sequence to copy. For the language modeling task, LM accuracy is computed based on $[bos]I[bos]S[EOS]$ where I represents the irrelevant text. For each accuracy datapoint of certain context length, we sample 10 different copy target so that the curve contains mean and variance.

Remarks on Closed Source Models The forgetting curve can be applied to closed source models to measure their long context memorization capability. However, as the parameters of commercial models are not accessible, simulating the teacher forcing process requires feeding the prefix to the model and calculating the prediction accuracy for each token, which is too costly and inefficient to perform.

Results Table 3 shows our main results, listing the key statistics derived from forgetting curves of each model. The exact forgetting curves can be found in figures from Appendix C. The models are grouped into three families: Llama models without context extensions, Transformer based models with context extensions and SSM/RNN models.

We observe a clear memorization capability improvement from the Llama2 series to the Llama3 series. While our results on coarse length confirms the claimed length for all four tested models,¹¹ we observe a significant enhancement in fine length where the model perfectly remembers the copy target, notably between Llama-3-base and Llama-2-base (4k vs 0). Considering the similar model structures of Llama3 and Llama2, and the fact that

¹⁰We take $n=32$ for most of the cases to be able to trace a representative curve.

¹¹Since the double of coarse length exceeds the claimed length.

Models	Claimed Length	Fine Length	Coarse Length	Effective Length
Llama-2-base (7B)	4k	0	2.7k	criterion
Llama-2-chat (7B)	4k	1.6k	2.7k	criterion
Llama-3-base (8B)	8k	4k	>9k	-
Llama-3-chat (8B)	8k	4k	6k	-
Llama-2-base-32k (7B)	32k	4k	20k	4k
Mistral-base-v0.2 (7B)	32k	22k	>30k	16k
Mistral-chat-v0.2 (7B)	32k	15k	>33k	16k
Mistral-chat-v0.3 (7B)	32k	22k	>33k	-
Qwen1.5 (7B)	32k	28k	>30k	-
ChatGLM3 (6B)	128k	37k	>62k	4k
Yarn-Llama-2 (7B)	128k	0	82k	<4k
LWM (7B)	1M	>225k	>225k	<4k
Mamba (2.8B)	2k/infinite	0	2.3k	<1k
RWKV (7B)	4k/infinite	265	3.6k	1k

Table 3: Performance of open-source models in Forgetting Curve and Ruler (Hsieh et al., 2024). The Fine-grained Memory Length (Fine Length) is the maximum length at which copy accuracy surpasses 99%. The Coarse-grained Memory Length (Coarse Length) is the maximum length at which the copy accuracy is greater than the language modelling accuracy by at least 1%. In Ruler, the Effective Context Length (Effective Length) is the maximum length where the performance reaches a certain threshold, which is defined by the Llama-2-7b-base/chat performance at 4K. In the Fine length/Coarse Length columns, if there is a “>” symbol, it signifies that the length exceeds the indicated measure.

Llama3’s additional GQA (Ainslie et al., 2023) doesn’t extend the context window, we attribute this improvement to Llama3’s increased training data.

Our experiments validate the effectiveness of context-extending models based on transformer architectures (Peng et al., 2024; Chen et al., 2023a; bloc97, 2023; emozilla, 2023). Their measured coarse lengths meet or exceed the claimed lengths, indicating superior long context modelling capabilities compared to original models. Most of the long-sequence Transformers we examined were produced by increasing the theta in RoPE (Su et al., 2024) and then fine-tuning on long sequences; our empirical results suggest such strategies are generally effective approaches to extend transformer based LLM context. We argue that the main concern for transformer models to extend to long context remain the quadratic time and space complexity.¹²

Our measurement on RNN/SSM models show some negatives results. While RNNs theoretically support infinite context length, they exhibit a short coarse-grained Memory Length and zero fine-

grained Memory Length, indicating an inability to perfectly memorize or to retain memory at any significant length. This highlights current limitations in RNN development for accurate token recall. Furthermore, in our measurement, RNNs like Mamba and RWKV exhibit higher language modeling accuracy than copy accuracy outside training context length (contrary to transformer models where two curves converge after certain length), suggesting an RNN model memory issue which seems to negatively affect correct token prediction at long context, see Appendix C for the Mamba and RWKV forgetting curves for more details.

5 Analysis & Discussion

Forgetting curve can be applied to a small model and is robust to data leakage. We want to see whether the forgetting curve can be used to measure model memory for small models. To this end, we train an 83M parameter model using the Llama architecture on the PG-19 training dataset with a training context size of 1024. Further details are provided in Appendix D.1. We then plot its forgetting curves, measured using both the PG-19 test and training datasets.

Figure 4 shows the corresponding forgetting curves for the Llama-83M model. According to our fine memory and coarse memory definition pre-

¹²Our experiments on LWM (A model with 1M claimed context length and we show the forgetting curve partially at Appendix Figure 18) are very slow and suffers from out-of-memory (OOM) issues in our settings.

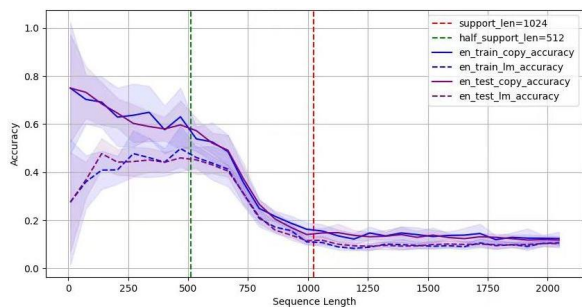


Figure 4: The forgetting curves for the Llama model with 83M parameters. The solid and dashed lines represent copy accuracy and language modeling accuracy respectively. The model is trained on the PG-19 training dataset, and the forgetting curves are plotted using both the PG-19 training and test datasets.

sented in the previous section, the model does not have fine memory but clearly exhibit coarse memory slightly beyond half of the training context size. We remark that the measurement is achieved on such small and arguably under-trained models because our measurement could separate between the *memory capability* and *natural language understanding capability*. In the contrary, it is not possible to measure memory using popular metrics such as “Needle in a Haystack” as the model has not yet acquired understanding at this stage.

We finally remark the closeness between forgetting curves measured on PG-19 test set and those measured on the training set. Obviously, both measurements point to very close coarse memory length. This shows that the forgetting curve is robust to data leakage as the forgetting curve greatly focuses on the difference between copy accuracy and LM accuracy.

Revisiting Perplexity using forgetting curve. Using the forgetting curve, we now revisit the hypothesis that perplexity is not directly related to better memorization or better usage of long context, which is first raised by (Hu et al., 2024b). We test this hypothesis using Transformer-XL (Dai et al., 2019), which has shown perplexity improvement, often attributed to its superior utilization of long context. We train a Transformer-XL style Llama (Llama-XL) on the PG-19 training set (details in Appendix D.2).

We plot the forgetting curve as well as model perplexity curve in Figure 5 measured by PG-19 test set. For the perplexity curve, we confirm the findings of the original Transformer-XL paper where the architecture enables low perplexity for long con-

text. However, the forgetting curve clearly shows that the perplexity performance is not related to model’s memorization (or leverage) of long context, as copy curve becomes indistinguishable from the LM curve from 1k tokens while perplexity continues to decrease until several order of magnitude.¹³ Our empirical results support the hypothesis (Hu et al., 2024b) that perplexity tests model’s short context modelling capability and is not an appropriate measure to test model’s long-context capability including memorization.

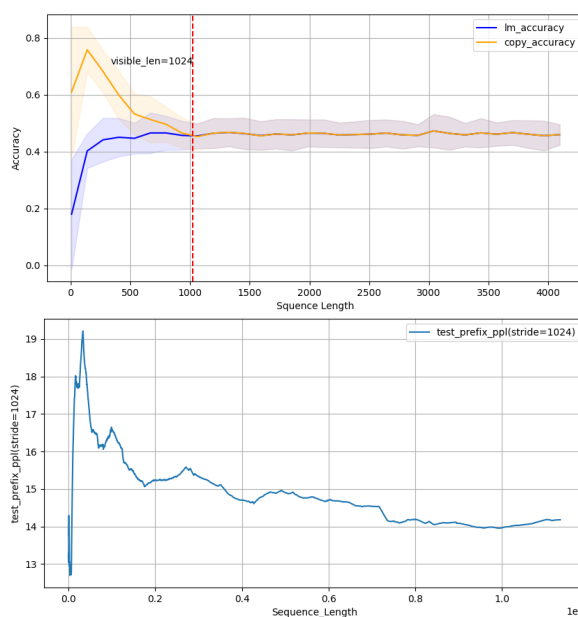


Figure 5: The forgetting curve (top) and perplexity (bottom) for the Llama-XL, which is trained and tested on the PG-19 dataset.

Comparison of forgetting curve and other context length measurement. We show context length measured by forgetting curves as well as by Ruler (Hsieh et al., 2024) (i.e. the Effective Length column) in Table 3. One can see that for models with low context length measured by forgetting curves (e.g. Mamba, RWKV), the effective length measured by Ruler is also low; however, the inverse does not hold, as shown by the experiments on ChatGLM3, Yarn-Llama-2 and LWM where we observe high context length while Ruler effective length remains low (<4k). We think this discrepancy are due to two interleaved factors: ① Ruler uses mainly synthetic tasks fulfilling which requires model’s understanding capability, and ②

¹³Anecdotally, Mistral v0.1 that uses sliding window attention (a technique that improves perplexity at long range) does not perform well on “Needle in a Haystack” at long range (v_JULY_v, 2024).

to be counted as effective length in Ruler, the measured capability needs to exceed Llama-2-base/chat baseline. Contrary to Ruler’s measurement, our measurement does not require understanding capability allowing us to measure model of any size (e.g. see Section 5 for our measurement for a 83M model) and our method does not require a baseline model to define memory length. Fundamentally, we devise our approach trying to disentangle the understanding capability and the memory capability, and uniquely measure the later one.

6 Conclusion

In this work, we identify the limitations present in existing long-sequence language modeling metrics, tasks and benchmarks. To address these limitations, we introduce a new method called forgetting curve, which provides a flexible and robust measure to visualize LLM’s long-range memory capabilities.

Using forgetting curve, we study the memory capabilities of fourteen open-source models with claimed context sizes ranging from 4K to 1M. Our findings empirically validate the effectiveness of existing transformer context extension methods, while raising questions for RNN/SSM architectures. Forgetting curve further reveals no direct correlation between memory capacity and perplexity.

Compared to existing ways to measure long-context memory, forgetting curves mainly differs in trying to effectively decouple the model’s long-term memory and language understanding ability. This provides a novel perspective for evaluating language models at all sizes, potentially guiding future research in model’s long-term memory.

Limitations

There are primarily two limitations in our work. Firstly, as the copy task requires two identical text segments, the maximum token dependency length is only half of the total sequence length, thus not achieving maximum efficiency for measurement. Notably, when the model imposes a strict limit on its maximum input length (e.g., ChatGLM3), our method allows only measuring up to half the maximum context length. Secondly, although we extensively tested the robustness under various sizes (83M, 7B), using different alignment settings (base and chat) and under various corpus settings, we remark that all the tests are performed within the Llama model family, leaving transferability of forgetting curves to other models unexplored.

Acknowledgments

This work was supported in part by the National Science Foundation of China (No.62276056), the Natural Science Foundation of Liaoning Province of China (2022-KF-16-01), the Fundamental Research Funds for the Central Universities (Nos. N2216016 and N2316002), the Yunnan Fundamental Research Projects (No. 202401BC070021), and the Program of Introducing Talents of Discipline to Universities, Plan 111 (No.B16009).

References

- Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. 2023. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. *arXiv preprint arXiv:2305.13245*.
- Chenxin An, Shansan Gong, Ming Zhong, Mukai Li, Jun Zhang, Lingpeng Kong, and Xipeng Qiu. 2023. L-eval: Instituting standardized evaluation for long context language models. *arXiv preprint arXiv:2307.11088*.
- Anthropic. 2023. Long context prompting for claude 2.1. <https://www.anthropic.com/news/claude-2-1-prompting>.
- Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, et al. 2023. Longbench: A bilingual, multitask benchmark for long context understanding. *arXiv preprint arXiv:2308.14508*.
- Simone Balloccu, Patrícia Schmidová, Mateusz Lango, and Ondřej Dušek. 2024. Leak, cheat, repeat: Data contamination and evaluation malpractices in closed-source llms. *arXiv preprint arXiv:2402.03927*.
- bloc97. 2023. Add ntk-aware interpolation "by parts" correction. <https://github.com/jquesnelle/yarn/pull/1>.
- Aydar Bulatov, Yuri Kuratov, Yermek Kapushev, and Mikhail S Burtsev. 2023. Scaling transformer to 1m tokens and beyond with rmt. *arXiv preprint arXiv:2304.11062*.
- Aydar Bulatov, Yury Kuratov, and Mikhail Burtsev. 2022. Recurrent memory transformer. *Advances in Neural Information Processing Systems*, 35:11079–11091.
- Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. 2023a. Extending context window of large language models via positional interpolation. *arXiv preprint arXiv:2306.15595*.
- Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. 2023b. Longlora: Efficient fine-tuning of long-context large language models. *arXiv preprint arXiv:2309.12307*.

- Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*.
- Zican Dong, Tianyi Tang, Junyi Li, Wayne Xin Zhao, and Ji-Rong Wen. 2023. Bamboo: A comprehensive benchmark for evaluating long text modeling capacities of large language models. *arXiv preprint arXiv:2309.13345*.
- emozilla. 2023. Dynamically scaled rope further increases performance of long context llama with zero fine-tuning. https://www.reddit.com/r/LocalLLaMA/comments/14mrgpr/dynamically_scaled_rope_further_increases/.
- Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. **ELI5: Long form question answering**. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy. Association for Computational Linguistics.
- gkamradt. 2023. Needle in a haystack - pressure testing llms. https://github.com/gkamradt/LLMTest_NeedleInAHaystack.
- Albert Gu and Tri Dao. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*.
- Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekeshe, Fei Jia, and Boris Ginsburg. 2024. Ruler: What’s the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*.
- Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, et al. 2024a. Minicpm: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*.
- Yutong Hu, Quzhe Huang, Mingxu Tao, Chen Zhang, and Yansong Feng. 2024b. Can perplexity reflect large language model’s ability in long text understanding? *arXiv preprint arXiv:2405.06105*.
- Yuri Kuratov, Aydar Bulatov, Petr Anokhin, Dmitry Sorokin, Artyom Sorokin, and Mikhail Burtsev. 2024. In search of needles in a 10m haystack: Recurrent memory finds what llms miss. *arXiv preprint arXiv:2402.10790*.
- Cheng Li, Ziang Leng, Chenxi Yan, Junyi Shen, Hao Wang, Weishi Mi, Yaying Fei, Xiaoyang Feng, Song Yan, HaoSheng Wang, et al. 2023a. Chatharuhi: Reviving anime character in reality via large language model. *arXiv preprint arXiv:2308.09597*.
- Jiaqi Li, Mengmeng Wang, Zilong Zheng, and Muhan Zhang. 2023b. Loogle: Can long-context language models understand long contexts? *arXiv preprint arXiv:2311.04939*.
- Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhui Chen. 2024. Long-context llms struggle with long in-context learning. *arXiv preprint arXiv:2404.02060*.
- Opher Lieber, Barak Lenz, Hofit Bata, Gal Cohen, Jhonathan Osin, Itay Dalmedigos, Erez Safahi, Shaked Meirom, Yonatan Belinkov, Shai Shalev-Shwartz, et al. 2024. Jamba: A hybrid transformer-mamba language model. *arXiv preprint arXiv:2403.19887*.
- Hao Liu, Wilson Yan, Matei Zaharia, and Pieter Abbeel. 2024. World model on million-length video and language with ringattention. *arXiv preprint*.
- Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.
- Meta.AI. 2024. Meta.ai. llama3 url <https://ai.meta.com/blog/meta-llama-3/>. <https://ai.meta.com/blog/meta-llama-3/>.
- Mistral.AI. 2023. Mistral.ai. la plateforme, 2023. url <https://mistral.ai/news/la-plateforme/>. <https://mistral.ai/>.
- Moonshot.AI. 2023. Kimi chat publishes the results of the "needle in a haystack" long text pressure test, and also clarifies the essence of the test. <https://mp.weixin.qq.com/s/IC5-FGLVHzHHYqH6x-aNng>.
- Tsendsuren Munkhdalai, Manaal Faruqui, and Siddharth Gopal. 2024. Leave no context behind: Efficient infinite context transformers with infinite attention. *arXiv preprint arXiv:2404.07143*.
- Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Matteo Grella, et al. 2023. Rwkv: Reinventing rns for the transformer era. *arXiv preprint arXiv:2305.13048*.
- Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. 2024. **YaRN: Efficient context window extension of large language models**. In *The Twelfth International Conference on Learning Representations*.
- Zexuan Qiu, Jingjing Li, Shijue Huang, Wanjun Zhong, and Irwin King. 2024. Clongeval: A chinese benchmark for evaluating long-context large language models. *arXiv preprint arXiv:2403.03514*.
- Qwen.Team. 2024. Introducing qwen1.5. <https://qwenlm.github.io/blog/qwen1.5/>.
- Jack W Rae, Anna Potapenko, Siddhant M Jayakumar, and Timothy P Lillicrap. 2019. Compressive transformers for long-range sequence modelling. *arXiv preprint arXiv:1911.05507*.
- Uri Shaham, Maor Ivgi, Avia Efrat, Jonathan Berant, and Omer Levy. 2023. Zeroscrolls: A zero-shot benchmark for long text understanding. *arXiv preprint arXiv:2305.14196*.

- Mingyang Song, Mao Zheng, and Xuan Luo. 2024. Counting-stars: A simple, efficient, and reasonable strategy for evaluating long-context large language models. *CoRR*, abs/2403.11802.
- Jianlin Su. 2023. The path of transformer upgrade: 15, key normalization facilitates length extrapolation. <https://kexue.fm/archives/9859>.
- Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063.
- Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. 2020. Long range arena: A benchmark for efficient transformers. *arXiv preprint arXiv:2011.04006*.
- Together.AI. 2024. Building llama-2-7b-32k-instruct using together api. <https://github.com/togethercomputer/Llama-2-7B-32K-Instruct>.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.
- v_JULY_v. 2024. The paper-review dataset was used to fine-tune separately mistral, gemma. https://blog.csdn.net/v_JULY_v/article/details/136656918.
- Zekun Moore Wang, Zhongyuan Peng, Haoran Que, Jiaheng Liu, Wangchunshu Zhou, Yuhan Wu, Hongcheng Guo, Ruitong Gan, Zehao Ni, Man Zhang, et al. 2023. Rolellm: Benchmarking, eliciting, and enhancing role-playing abilities of large language models. *arXiv preprint arXiv:2310.00746*.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface’s transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.
- Yonghao Wu, Zheng Li, Jie M Zhang, and Yong Liu. 2023. Condefects: A new dataset to address the data leakage concern for llm-based fault localization and program repair. *arXiv preprint arXiv:2310.16253*.
- Yuhuai Wu, Markus N Rabe, DeLesley Hutchins, and Christian Szegedy. 2022. Memorizing transformers. *arXiv preprint arXiv:2203.08913*.
- Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2023. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*.
- Wenhan Xiong, Jingyu Liu, Igor Molybog, Hejia Zhang, Prajwal Bhargava, Rui Hou, Louis Martin, Rashi Rungta, Karthik Abinav Sankararaman, Barlas Oguz, et al. 2023. Effective long-context scaling of foundation models. *arXiv preprint arXiv:2309.16039*.
- Jianxin Yang. 2023. Longqlora: Efficient and effective method to extend context length of large language models. *arXiv preprint arXiv:2311.04879*.
- Tao Yuan, Xuefei Ning, Dong Zhou, Zhijie Yang, Shiyao Li, Minghui Zhuang, Zheyue Tan, Zhuoyu Yao, Dahua Lin, Boxun Li, et al. 2024. Lv-eval: A balanced long-context benchmark with 5 length levels up to 256k. *arXiv preprint arXiv:2402.05136*.
- Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. 2022. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*.
- Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Khai Hao, Xu Han, Zhen Leng Thai, Shuo Wang, Zhiyuan Liu, et al. 2024. ∞ bench: Extending long context evaluation beyond 100k tokens. *arXiv preprint arXiv:2402.13718*.

A Statistical Analysis

In Figure 6, the one-way ANOVA tests and Kruskal-Wallis H-test were conducted with the copy target setting fixed at en_test, and the irrelevant prefix set across three different settings: en_test, en_train, and zh. It can be seen that within half of the claimed length (2k), the choice of irrelevant prefix has no significant impact on the accuracy of language modeling.

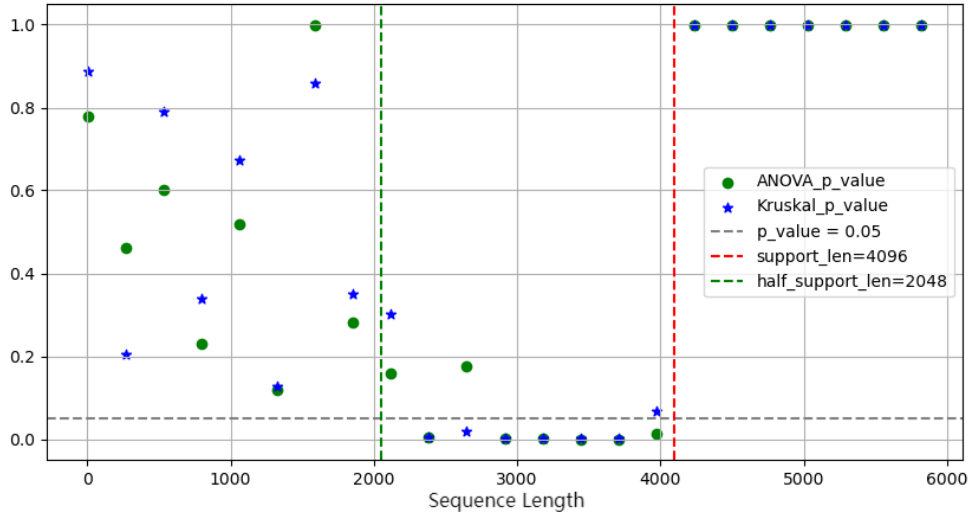


Figure 6: ANOVA and Kruskal tests for LM accuracy.

B Models

The models evaluated in section 4 are listed in Table 4.

Name	Training length	Size	Huggingface (Wolf et al., 2019) Source
Llama-2-base (Touvron et al., 2023)	4k	7B	meta-llama/Llama-2-7b-hf
Llama-2-chat (Touvron et al., 2023)	4k	7B	meta-llama/Llama-2-7b-chat-hf
Llama-3-base (Meta.AI, 2024)	8k	8B	meta-llama/Meta-Llama-3-8B
Llama-3-chat (Meta.AI, 2024)	8k	8B	meta-llama/Meta-Llama-3-8B-Instruct
Llama-2-base-32k (Together.AI, 2024)	32k	7B	togethercomputer/LLaMA-2-7B-32K
Mistral-base-v0.2 (Mistral.AI, 2023)	32k	7B	alpindale/Mistral-7B-v0.2-hf
Mistral-chat-v0.2 (Mistral.AI, 2023)	32k	7B	mistralai/Mistral-7B-Instruct-v0.2
Mistral-chat-v0.3 (Mistral.AI, 2023)	32k	7B	mistralai/Mistral-7B-Instruct-v0.3
Qwen1.5 (Qwen.Team, 2024)	32k	7B	Qwen/Qwen1.5-7B
ChatGLM3 (Zeng et al., 2022)	128k	6B	THUDM/chatglm3-6b-128k
Yarn-Llama-2 (Peng et al., 2024)	128k	7B	NousResearch/Yarn-Llama-2-7b-128k
LWM (Liu et al., 2024)	1M	7B	LargeWorldModel/LWM-Text-1M
Mamba (Gu and Dao, 2023)	2k	2.8B	state-spaces/mamba-2.8b-hf
RWKV (Peng et al., 2023)	4k	7B	RWKV/v5-Eagle-7B-HF

Table 4: Information of evaluated and analyzed models.

C Forgetting Curves of Open Source Models

We have plotted the forgetting curves for all the open-source models listed in Table 3.

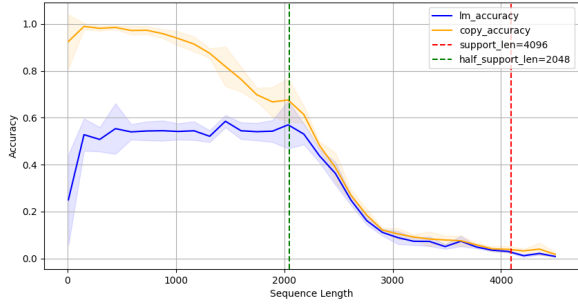


Figure 7: Llama-2-base

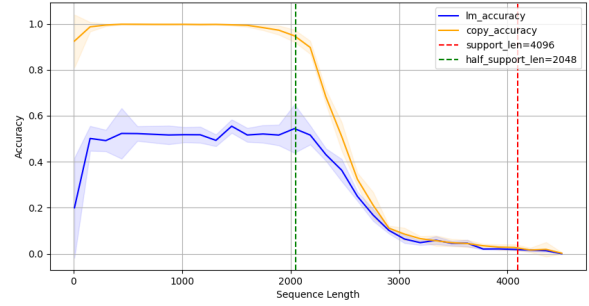


Figure 8: Llama-2-chat

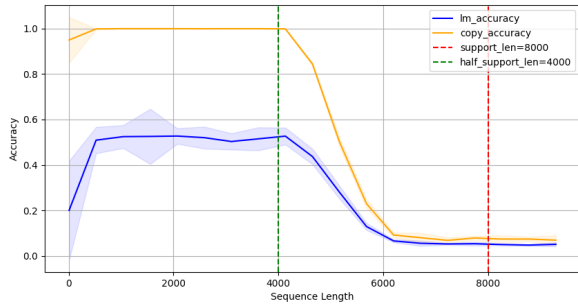


Figure 9: Llama-3-base

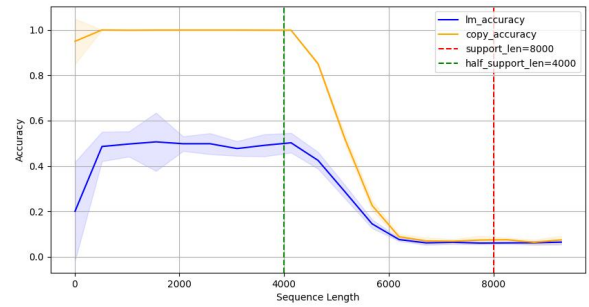


Figure 10: Llama-3-chat

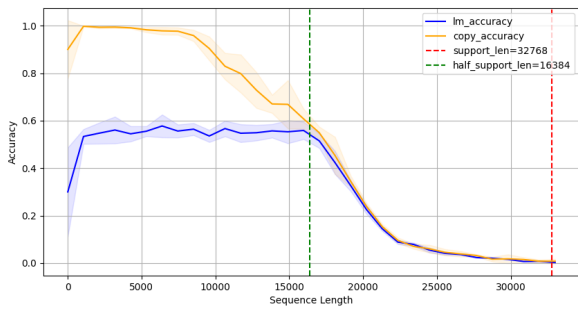


Figure 11: Llama-2-base-32k

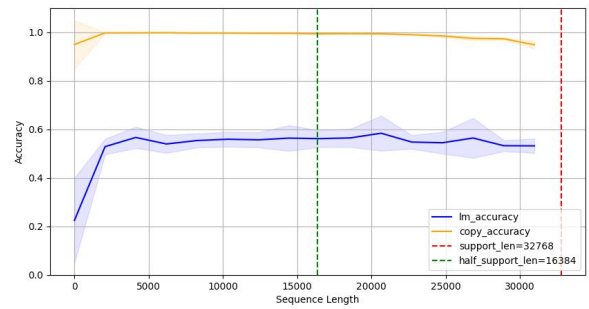


Figure 12: Mistral-base-v0.2

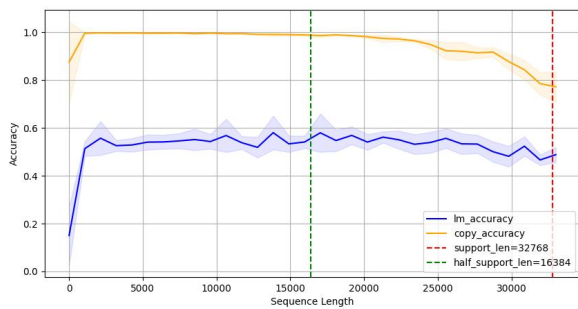


Figure 13: Mistral-chat-v0.2

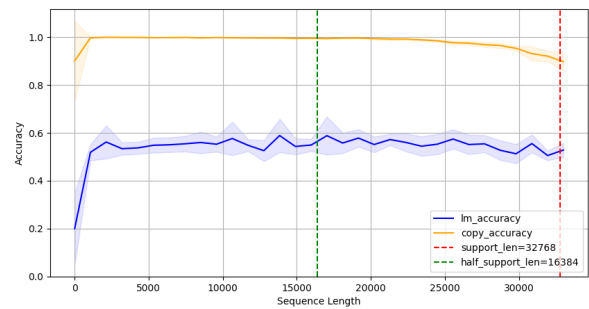


Figure 14: Mistral-chat-v0.3

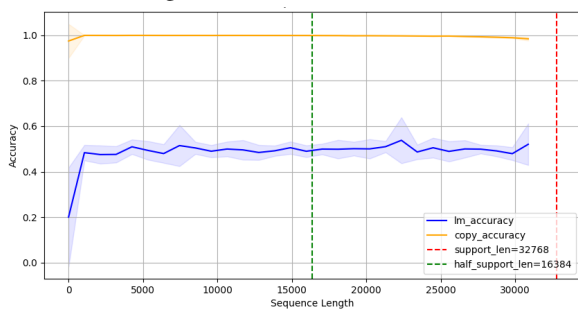


Figure 15: Qwen1.5

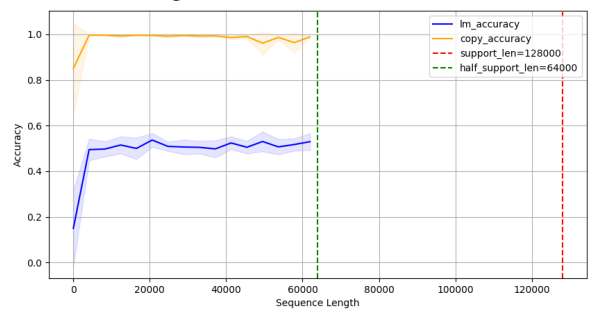


Figure 16: ChatGLM3

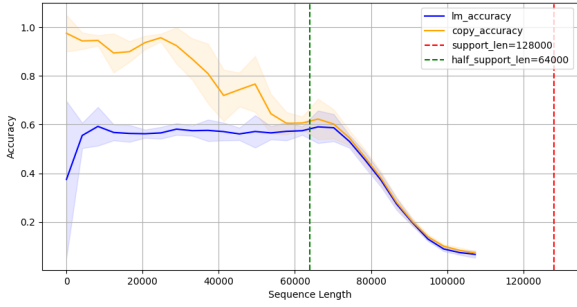


Figure 17: Yarn-Llama-2

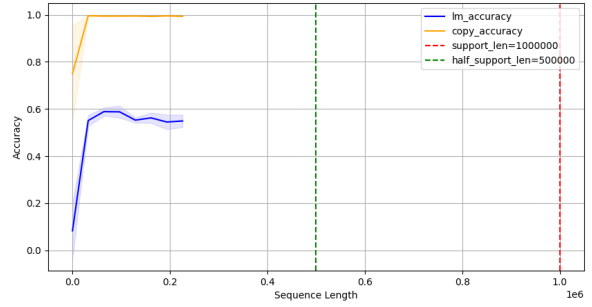


Figure 18: LWM

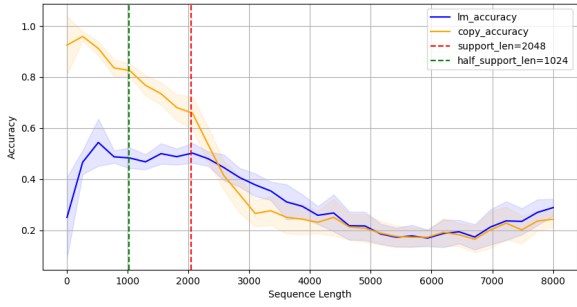


Figure 19: Mamba

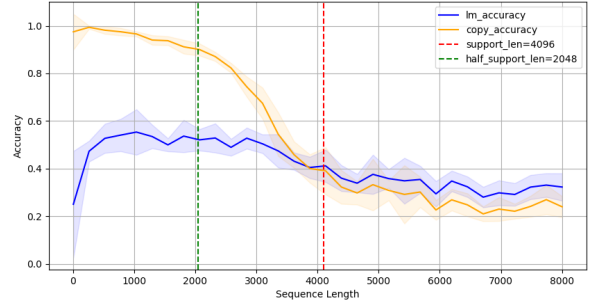


Figure 20: RWKV

D Details of Custom Model Training

D.1 Llama-83M

We train the Llama-83M model for one epoch on the PG-19 training set, with the objective of maximizing the probability of the next token, which is the standard pre-training objective for language models. The model and training hyperparameters are listed in Table 5. We use the WSD scheduler (Hu et al., 2024a) shown in Figure 21. The optimization is performed using the AdamW optimizer (Loshchilov and Hutter, 2017) with parameters $(\text{beta1}, \text{beta2}) = (0.9, 0.95)$, epsilon set to $1e-8$, and a weight decay of 0.1. We do not employ dropout in our model.

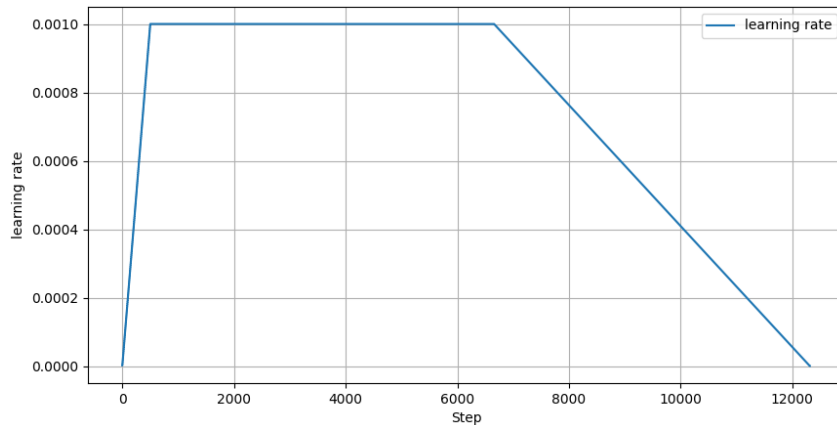


Figure 21: WSD scheduler

Hyperparameters	Value
model_type	llama
hidden_act	silu
initializer_range	0.02
hidden_size	512
intermediate_size	2048
max_position_embeddings	2048
num_attention_heads	8
num_hidden_layers	12
num_key_value_heads	8
pretraining_tp	1
rms_norm_eps	1.00×10^{-5}
tie_word_embeddings	FALSE
torch_dtype	float16
vocab_size	32000
training_len	1024
total_batch_size	256
batch_size_per_device	8
device_count	4
gradient_accumulation_steps	8
learning_rate	0.001
max_grad_norm	1

Table 5: Model Hyperparameters and Training Hyperparameters.

D.2 Llama-XL

We modify the attention module of Llama to make it a model in the style of Transformer-XL (Dai et al., 2019). Here is the attention formula for the n -th layer and the τ -th segment:

$$q_\tau^n, k_\tau^n, v_\tau^n = h_\tau^{n-1} W_q^n, h_\tau^{n-1} W_k^n, h_\tau^{n-1} W_v^n \quad (1)$$

$$\tilde{k}_\tau^n = L2Norm(k_\tau^n) \quad (2)$$

$$\hat{k}_\tau^n, \hat{v}_\tau^n = [k_cache^n; \tilde{k}_\tau^n], [v_cache^n; v_\tau^n] \quad (3)$$

$$k_cache^n, v_cache^n = \tilde{k}_\tau^n, v_\tau^n \quad (4)$$

$$pids = [0, 1, 2, \dots, len(\hat{k}_\tau^n) - 1] \quad (5)$$

$$\bar{q}_\tau^n, \bar{k}_\tau^n = apply_rope(q_\tau^n, pids[-len(q_\tau^n) :]), apply_rope(\hat{k}_\tau^n, pids) \quad (6)$$

$$o_\tau^n = softmax(\bar{q}_\tau^n \bar{k}_\tau^{nT} + Mask) \hat{v}_\tau^n \quad (7)$$

$$\tilde{h}_\tau^n = o_\tau^n W_o^n \quad (8)$$

n denotes the layer index of the attention module. τ denotes the segment index of the data. $[\cdot; \cdot]$ denotes concatenation along the sequence dimension, while $[\cdot]$ denotes slicing along the sequence dimension. h_τ^{n-1} is the hidden state from the previous layer. $W_q^n, W_k^n, W_v^n, W_o^n$ are the weight matrices for query, key, value, and output, respectively. $q_\tau^n, k_\tau^n, v_\tau^n$ are the query, key, and value matrices. \tilde{k}_τ^n is the key matrices after L2 normalization along the embedding dimension. $\hat{k}_\tau^n, \hat{v}_\tau^n$ are the concatenated key and value matrices with the cached ones from the previous segment along the sequence dimension. k_cache^n, v_cache^n are the cached key and value matrices, updated with the current segment. $pids$ is the sequence of position indices from 0 to $len(\hat{k}_\tau^n) - 1$. $apply_rope$ is the function that applies the rotary position embedding (Su et al., 2024). $Mask$ is the attention mask, which is a LowerTriangularFromBottomRightMask as illustrated in Figure 22. o_τ^n is the attention output. \tilde{h}_τ^n is the output hidden state.

The main modification is in Equation (3), where we concatenate the KV cache from the previous



Figure 22: Mask used in Llama-XL. Elements in the green area are 0, and elements in the gray area are $-\infty$.

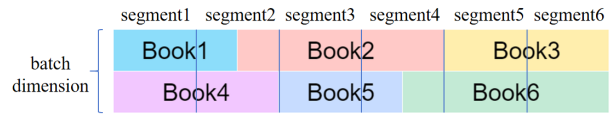


Figure 23: Ordered DataLoader.

segment to the current one, unlike Transformer-XL, which concatenates the hidden state. It is worth noting that our position encoding spans from 0 to $len(\hat{k}_\tau^n) - 1$, thereby guaranteeing that the model employs position encodings encountered during training for inference. Normalizing the k_τ^n with L2Norm helps with the extrapolation of the position encoding (Su, 2023), although we do not use position encoding outside of the training range. We also use an ordered DataLoader to ensure that segments are consecutive, as shown in Figure 23. Other experimental settings are consistent with Llama-83M.