Characterizing Mamba's Selective Memory using Auto-Encoders

Tamanna Hossain[*†] **Robert L. Logan IV**[‡] **Ganesh Jagadeesan**[‡]
Sameer Singh[†] **Joel Tetreault**[‡] **Alejandro Jaimes**[‡]
[†]University of California, Irvine [‡]Dataminr Inc.
{tthossai,sameer}@uci.edu aj27@caa.columbia.edu
{rlogan,cjagadeesan,jtetreault}@dataminr.com

Abstract

State space models (SSMs) are a promising alternative to transformers for language modeling because they use fixed memory during inference. However, this fixed memory usage requires some information loss in the hidden state when processing long sequences. While prior work has studied the sequence length at which this information loss occurs, it does not characterize the types of information SSM language models (LMs) tend to forget. In this paper, we address this knowledge gap by identifying the types of tokens (e.g., parts of speech, named entities) and sequences (e.g., code, math problems) that are more frequently forgotten by SSM LMs. We achieve this by training an auto-encoder to reconstruct sequences from the SSM's hidden state, and measure information loss by comparing inputs with their reconstructions. We perform experiments using the Mamba family of SSM LMs (130M–1.4B) on sequences ranging from 4–256 tokens. Our results show significantly higher rates of information loss on math-related tokens (e.g., numbers, variables), mentions of organization entities, and alternative dialects to Standard American English. We then examine the frequency that these tokens appear in Mamba's pretraining data and find that less prevalent tokens tend to be the ones Mamba is most likely to forget. By identifying these patterns, our work provides clear direction for future research to develop methods that better control Mamba's ability to retain important information.[1]

1 Introduction

State space models (SSMs) have emerged as a promising alternative to transformers (Vaswani et al., 2017) for language modeling, with the Mamba (Gu and Dao, 2023) SSM LMs achieving comparable performance to similar sized transformer models while being substantially more memory efficient at inference time. This memory efficiency arises from the fact that SSM next token predictions are computed recurrently using a fixed-size hidden state, whereas transformer next token predictions are computed using a key-value cache that grows linearly with the sequence length.

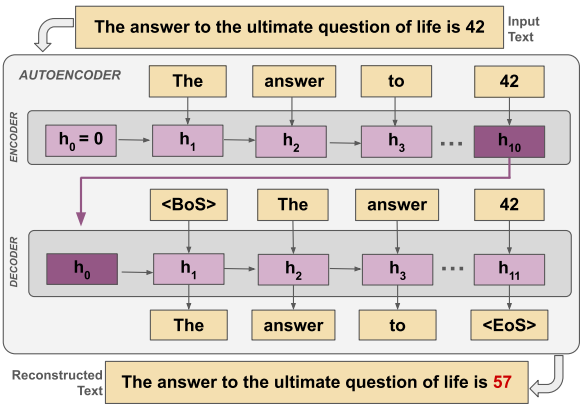

Figure 1: **Identifying Selective Memory Loss.** We train auto-encoders to reconstruct inputs directly from Mamba's hidden states, and measure information loss by comparing inputs to their reconstructions. These reconstructions act as probes of the hidden state's information retention: more faithful reconstruction implies greater information retention.

Recent work has shown that this efficiency comes at the cost of some information loss when processing long sequences. In particular, Jelassi et al. (2024) establish an upper bound on the length of sequences that can be exactly copied by Mamba, while Wang et al. (2025b) prove that the influence of an input token decays exponentially with input length. To provide empirical support for these theoretical results, these works measure Mamba's decline in accuracy as sequence length increases on copying tasks that either use synthetic inputs, or prime the model to memorize information using cues in the prompt. While these results character-

[*]Corresponding author. Work done while an intern at Dataminr Inc.

[1]Our code is available at: https://github.com/dataminr-ai/ouroboros.

Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 2007–2022
December 20-24, 2025 ©2025 Association for Computational Linguistics

ize *when* Mamba begins to forget, they leave open the question of *what* kinds of information is more likely to be forgotten, particularly on natural inputs.

In this paper, we develop a methodology that allows us to better understand the types of tokens and sequences that the Mamba LM is likely to forget in practical settings. The core idea of our approach is to compare input texts with known properties—such as named entity recognition (NER) and parts of speech (POS) tags on the token level, and categorical labels on the sequence level—to reconstructions obtained from Mamba's hidden state. By correlating reconstruction errors (measured using F1 scores and token omission rates) with the input properties we are able to obtain high-level insights about Mamba's *selective memory*. For example, in Figure 1 we see that the number "42" is reconstructed as "57", providing some evidence that Mamba has a tendency to forget numbers.

To obtain input reconstructions, we train an autoencoder that decodes sequences from hidden states produced by a frozen Mamba encoder. By using pretrained checkpoints for our encoder, we are able to examine the impact language model pretraining on Mamba's memory, rather than just the architecture. This approach additionally complements the approaches used by prior works in that it allows us to study reconstructions of *any* piece of text without having to provide memorization cues. In Section 4.2 we measure the performance of this approach for a number of different model sizes and sequence lengths, and validate that it replicates findings from previous works.

In Sections 4.3 and 4.4 we then employ this approach to study token- and sequence-level characteristics of reconstruction errors on a number of datasets. We begin by looking at Mamba's pretraining dataset The Pile (Gao et al., 2020), which allows us to understand how in-distribution errors vary by the input source. Our results show that Mamba is significantly more likely to forget tokens on math-related sequences, even for relatively short sequence lengths (16+ tokens). We also perform experiments on Groenwold et al. (2020)'s SAE/AAVE tweet pair dataset, and find significantly elevated reconstruction errors on texts written using African American Vernacular English.

We then study token-level NER and POS errors on CoNLL-2003 (Tjong Kim Sang and De Meulder, 2003), and find significantly higher reconstruction errors on mentions of organizations, as well as numerical tokens. To better understand this, we examine a 178M token sample from the Pile and find an association between the categories that Mamba tends to forget and the frequency that unique tokens belonging to those categories appear in the Pile, suggesting that their rarity in Mamba's pretraining data may contribute to their omission. This raises the question of whether conventional pretraining pipelines are well suited for the inductive biases of SSM-based language models. In sum, our work establishes clear differences in the types of information the Mamba LM tends to retain versus forget, providing direction for future research into methods to improve the retention of important information in its hidden state.

2 Related Work

State Space Models State space models (SSMs) are linear recurrent architectures with fixed-size hidden states that enable linear-time training and constant-time inference per step, making them a more efficient alternative to transformers for sequence modeling (Gu et al., 2021, 2022a). Recent models like Mamba (Gu and Dao, 2023) have shown that SSMs can match transformers of similar scale on standard NLP benchmarks, aided by long-range initialization techniques from HiPPO theory (Gu et al., 2022b) and time-dependent parameterization that allows selective attention over inputs. Hybrid transformer-SSM architectures have shown gains in both efficiency and accuracy over pure transformers (Wang et al., 2025a), reinforcing the value of SSM-based models—especially as inference-time scaling is increasingly used to enhance LM reasoning through techniques like task decomposition (Wei et al., 2022; Yao et al., 2023) and increased sampling (Wang et al., 2023).

Although Mamba models offer improved computational efficiency over transformers, recent studies have shown that this comes at the cost of information degradation over long contexts. Jelassi et al. (2024) derive a theoretic upper bound on the sequence length that Mamba can recall information from, and Wang et al. (2025b) demonstrate that token influence decays exponentially with input distance, i.e., there is a *recency bias* to Mamba's memory. These findings are empirically supported by experiments using input sequences with explicit memorization cues and synthetic inputs (Waleffe et al., 2024; Wang et al., 2025b). However, while these prior works illuminate *when* forgetting occurs in Mamba's hidden state, it still remains unknown

what is forgotten, especially for natural inputs. This is the knowledge gap we address in our paper.

Interpreting Hidden States Our approach relates to a large body of work using learned classifiers to probe LM hidden states for the presence of properties such as syntactic and semantic knowledge (see Rogers et al. (2020) for an overview). Our use of an auto-encoder to probe memorization of input sequences is unique in this literature, however this likely stems from the fact that input compression is not a concern for the transformer-based LMs typically considered in these works. While auto-encoders have a rich history of use to train model hidden states to capture linguistic structures e.g., syntax and semantics (Zhang et al., 2023), negation and uncertainty (Vasilakes et al., 2022), content and style (Li et al., 2022b; John et al., 2018), and tense, verb style and gender (Mercatali and Freitas, 2021), there is little existing literature that uses auto-encoders to study hidden states for existing language models. To our knowledge, the closest comparison to our work is Templeton et al. (2024), who train sparse auto-encoders to learn discrete interpretable features captured by Claude's hidden states,[2] however our work substantially differs from theirs in terms of both method and focus.

3 Method

In this section, we introduce a method for identifying the types of tokens and sequences Mamba is most likely to forget. In our approach, we reconstruct text from Mamba's hidden states using a trained auto-encoder and then compare the output to the original input annotated with features such as part-of-speech tags, named entities, and sequence-level categories. This allows us to assess information retention in the natural usage of Mamba as a language model without relying on memorization prompts like prior works.

3.1 Auto-Encoder

Given an input text, x, the encoder function $f_{enc}(\cdot)$ maps the input into a latent space representation β. The decoder function $f_{dec}(\cdot)$ then takes this latent representation, β, and maps it back to the input space of text as $\tilde{x}$. Formally:

$$\beta = f_{enc}(x),\ \tilde{x} = f_{dec}(\beta)$$

[2]https://www.anthropic.com/claude

Encoder We use pretrained Mamba (Gu and Dao, 2023) language models with frozen weights[3] to encode the input text into a latent space representation with two parts: (i) an *SSM State*, β_S, and (ii) a *Convolutional State*, β_C.

The Mamba architecture passes embeddings of an input sequence $x = [x_1, \ldots, x_n]$ through a convolutional filter, creating a convolutional hidden state, β_C. After applying a non-linear activation we get an intermediate state, $z = [z_1, \ldots, z_n]$, which is passed into an SSM. Within the SSM, recurrent states are computed as follows for time-dependent parameters $\bar{A}_t$ and $\bar{B}_t$ for $t \in \{1, \ldots, n\}$,

$$h_t = \bar{A}_t h_{t-1} + \bar{B}_t z_t$$

The last recurrent state is taken as the SSM hidden representation of the input sequence, β_S. Thus, the encoder maps an input sequence, x, into a latent state $\beta = [\beta_C, \beta_S]$.

Decoder Once the latent representation β is obtained from the frozen Mamba encoder, the decoder produces a reconstruction $\tilde{x}$ of the input text x. In this work, we initialize the decoder using the same-pretrained Mamba architecture as the encoder. The decoder (i) sets the initial state, h_0, of the decoder's SSM to the SSM state from the encoder, β_S, and (ii) similarly initializes the the decoder's convolution with the convolutional state of the encoder, β_C. Given these initializations, the decoder autoregressively reconstructs the input text token by token.

3.2 Measuring Information Loss

To measure information loss in Mamba's hidden state, we compare input texts with their reconstructions produced by a trained auto-encoder. These reconstructions serve as probes into the information retention capacity of the hidden state: the more faithfully the input can be recovered, the more information must have been retained. We assess the fidelity of reconstructions using two metrics: (i) Omission Rate, and (ii) Rouge F1-Score.

Omission Rate is a token-level metric that measures the frequency that specific input tokens are forgotten or omitted in reconstructions. Let $f_{\text{in}}(t)$ and $f_{\text{rec}}(t)$ denote the frequency of token t in the original input and reconstructed output, respectively. The omission rate for token t is:

[3]We use frozen weights in the encoder because our goal is to study the pre-trained Mamba checkpoints directly, evaluating the information preserved in their hidden states through standard language modeling.

$$\text{Omission Rate}(t) = 1 - \frac{f_{\text{rec}}(t)}{f_{\text{in}}(t)}$$

An omission rate of 0 indicates that the perfect retention of a token, while a rate of 1 implies complete information loss.

ROUGE F1-Score (Lin, 2004) provides a measure of reconstruction quality at the sequence level by measuring the overlap between the reconstructed text and the original input. Specifically, it balances *precision*—how many of the reconstructed tokens are correct—with *recall*—how many of the original tokens are reproduced.

Together, these two metrics allow us to measure information loss at the token level (e.g., named entities, parts of speech) and the sequence level (e.g., math problems, emails). By measuring this information loss on subsets of data with known token- and sequence-level metadata (e.g., tokens with certain NER tags, or sequences from a particular source) we are able to extract high-level insights about the kinds of information Mamba is more likely to forget.

4 Experiments and Results

4.1 Training

We train the auto-encoder with Mamba models ranging from 130M-1.4B parameters to reconstruct sequences of lengths $l \in \{4, 8, 16, 32, 64, 128, 256\}$. We initialize the autoencoders with pretrained Mamba checkpoints and freeze the encoder to ensure that we are evaluating the representations produced by the original model.

We train on the Pile—Mamba's pretraining dataset—to mitigate issues of distribution shift. However, since the full Pile is no longer publicly available due to copyright restrictions, we use a version with all copyright-protected content removed, Pile Uncopyrighted.[4] We train auto-encoders to reconstruct texts at various fixed lengths in order to study how information is lost as the sequence length changes. To control for sequence length as a potential confounding factor in auto-encoder training and evaluation, we train separate auto-encoders for each fixed sequence length and evaluate them only on sequences of the corresponding length.

The encoder weights are frozen because the goal of our work is to study how Mamba, off the shelf, compresses information into its hidden state for its pre-training objective of next token prediction, while the decoder weights are trained specifically for the task of input reconstruction from the encoder's hidden states. The state sizes of different models are shown in Table A1. We adopt a constant learning rate of 1×10^{-5}, matching the final learning rate of Mamba pre-training. We train until convergence with a batch size calibrated to the model size and GPU memory constraints. We use 200K instances (282M tokens) from Pile Uncopyrighted to train the decoder.

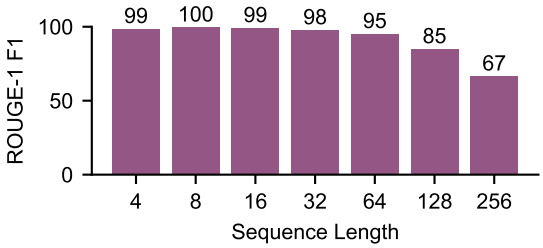

Figure 2: **Reconstruction Performance Across Sequence Lengths**. ROUGE F1-score for reconstructing text using Mamba (130M). Performance declines sharply as sequence length increases.

During the data preprocessing step, the training corpus is tokenized, concatenated into a single stream, and then divided into chunks of each sequence length. Additionally, a beginning-of-sentence (BOS) token is appended at the start of each sequence before it is provided to the encoder. For the decoder, an end-of-sequence (EOS) token is added to the end of each sequence.

We optimize our model parameters using a language modeling objective with cross-entropy between the input and reconstructed texts as the loss function. Training is halted using the following early stopping criterion evaluated on a validation set of 128 instances sampled from Pile Uncopyrighted. The encoder maps each validation instance to Mamba's latent space, and the decoder generates tokens autoregressively until an end-of-sequence (EOS) token is produced or the maximum set threshold of 300 tokens is reached. Every 1000 steps, we measure the validation ROUGE F1-score. If it does not improve by at least 0.1 over 5000 training steps, then training is stopped.

4.2 Replicating Existing Findings

We validate our approach on 700 held-out instances from Pile Uncopyrighted, evaluating whether it

[4] https://huggingface.co/datasets/monology/pile-uncopyrighted

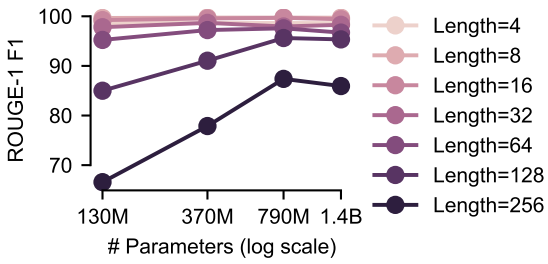

Figure 3: **Performance vs. Model Size**. ROUGE F1-score as a function of model size, broken down by sequence length. Shorter sequences achieve high performance with fewer parameters, while longer sequences benefit significantly from increased model capacity.

replicates existing findings on Mamba's information loss as a function of sequence length, model size, and token position. To measure information loss across sequence length and model size we use the ROUGE F1-score. To measure information loss across token position we use omission rate.

Sequence Length Like prior work (Jelassi et al., 2024; Waleffe et al., 2024), we see a decline in reconstruction fidelity as sequence length increases across model sizes (Figure 3). For instance, for a 130M size model, the reconstruction for the shortest sequence (length 4) is not perfect; it remains very high at 98.6 (Figure 2). However, starting at sequence length 16, the scores begin to drop, with sequences of length 128 and 256 showing notably poor performance, with ROUGE F1-scores of 85.0 and 66.6, respectively.

Model Size As the number of parameters increases, the performance generally improves, with larger sequence lengths benefiting more from increased model size (Figure 3). For smaller sequence lengths ($<$16), even the smallest models learn the reconstruction effectively, achieving high performance. However, for longer sequences (128 and 256), performance improves substantially with an increase in model parameters, showing a much steeper rise in the ROUGE F1-score as the model size grows. This suggests that longer sequence lengths require more capacity in the models to adequately encode and decode the sequence information required for reconstruction, reinforcing prior theoretical results (Jelassi et al., 2024).

For sequence lengths 64 and 256, we observe that Mamba 790M outperforms Mamba 1.4B. Such non-monotonic behavior has been noted in prior work—for example, Lester et al. (2021) report that a T5-Small model can outperform larger T5-Base/Large/XL models under certain prompt-tuning conditions. Nonetheless, the overall trend supports the conclusion that larger models are better equipped to handle longer sequences.

Token Position In line with prior work (Wang et al., 2025b), we observe a *recency bias* in Mamba's memory, i.e., earlier tokens are omitted at higher rates than recent ones. We observe this phenomenon across sequence lengths and model sizes. For example, in Figure 4 we see tokens at the beginning of a sequence being forgotten up to 6 times more than ones at the end of sequences.

4.3 Token-Level Trends

To understand what types of tokens Mamba tends to retain versus forget in its hidden state, we look at (i) tokens with the top omission rates in natural text, and (ii) omission rates across part-of-speech and named entity categories in annotated data.

Top Omitted Tokens in Natural Text To investigate what tokens are most frequently omitted in natural text reconstruction, we again use the Pile evaluation dataset across all sequence lengths, and measure ommission rates per token. To prevent artifacts due to data scarcity, tokens appearing fewer than 100 times in the dataset are excluded.

Table 1 lists the top omitted tokens for a model size of 130M (results for the 1.4B model can be found in Table A5). We find that the top 50 omitted tokens include many numbers, letters, and stop words across model sizes. This provides our first piece of evidence that Mamba has memory issues with math-related texts as numbers and letters correspond to tokens that often take the role of variables.

Part-of-Speech and Named Entities To investigate how well different parts-of-speech (PoS) and Named Entities (NE) are retained by Mamba, we use the test split of CoNLL-2003. It contains 3,453 instances of English newswire text, with token-level annotations for PoS and NE tags. To ensure consistency in sequence length, instances are concatenated up to a fixed length of 256 as it has the highest omission rate. After excluding categories with less than 100 instances, we compute omission rates across PoS and NE categories, and assess differences between categories using t-tests. We apply Bonferroni correction to adjust p-values for multiple comparisons ($\alpha = 0.05$).

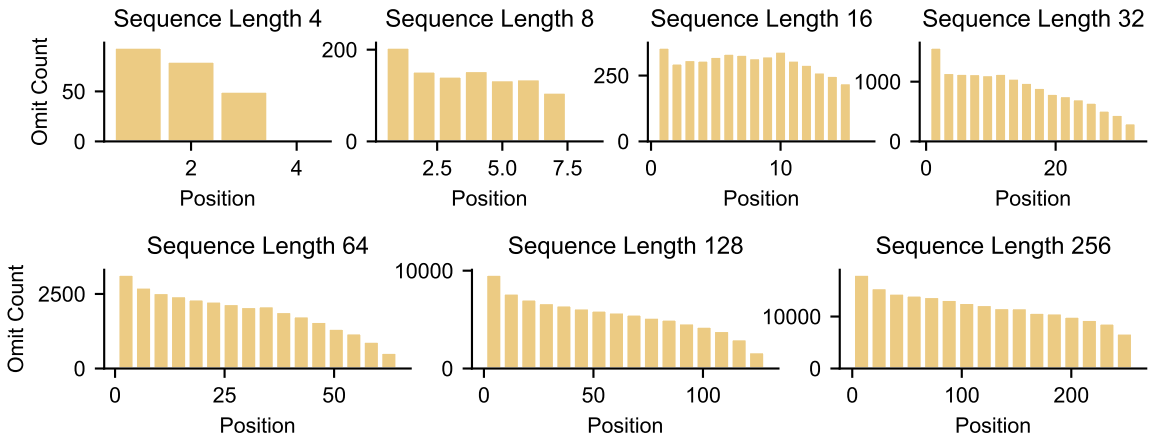

Figure 4: **Error Positions**. We count the number of reconstruction errors per position in a sequence to assess whether token position impacts reconstruction accuracy. We find that later positions have more reconstruction errors than earlier positions. 130M model errors by position are shown here.

Token	Omit Rate	Token	Omit Rate	Token	Omit Rate	Token	Omit Rate	Token	Omit Rate
␣(-	23.5	␣times	14.6	␣z	13.5	␣b	12.2	␣less	11.6
␣13	18.7	␣s	14.3	␣20	13.4	␣further	12.1	21	11.6
␣11	18.3	␣k	14.3	␣15	13.1	␣Suppose	12.0	62	11.5
␣␣4	15.9	␣17	14.3	␣four	13.0	␣however	12.0	␣16	11.4
Suppose	15.4	␣l	14.2	␣12	12.8	␣8	11.9	))	11.4
␣u	15.3	␣7	14.1	31	12.8	␣9	11.9	␣either	11.3
␣w	15.0	with	14.1	␣30	12.7	␣d	11.8	␣18	11.2
␣o	14.9	␣3	13.7	␣List	12.6	␣y	11.7	␣similar	11.0
Category	14.9	␣6	13.6	␣j	12.6	58	11.7	␣above	10.9
␣14	14.6	␣5	13.5	␣n	12.2	␣c	11.7	and	10.9

Table 1: **Top 50 Omitted Tokens.** The 50 most forgotten tokens by Mamba (130M) on the Pile across sequence lengths.

For PoS tags, we find numbers have the highest omission rate across model sizes. For a 130M model, the omission rate is 50.8% and for a 1.4B model the omission rate is 22.7% (Figure 5). We find a statistically significant difference between numbers and all other PoS categories across model sizes (Tables A2 and A6 in the Appendix). This reinforces our result from the previous experiment that Mamba's hidden state struggles to retain numbers. For the 130M model, we also find statistically significant differences between a few other PoS pairs. For the 1.4B model we find statistically significant differences in omission rates for punctuations, particles, and nouns (which are the 3 categories that rank just below numbers as the most frequently omitted) and many other categories.

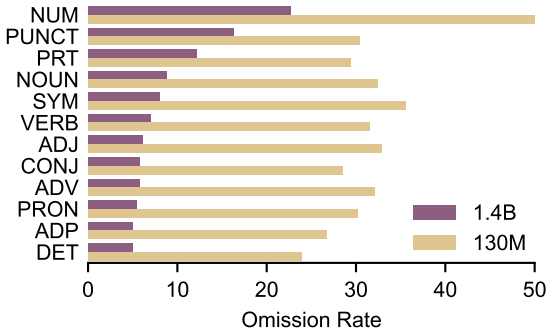

Figure 5: **Part-of-Speech.** Omission rates by PoS type on CoNLL-2003. Numbers have the highest omission rate across model sizes.

For named entities, we find organizations have the highest omission rate across model sizes. For the 130M model, the omission rate is 35.8% while it is 12.3% for the 1.4B model (Figure 6). We find a statistically significant difference between organizations and other NE categories (Tables A3 and A7 in the Appendix). For the 1.4B model, we also find statistically significant differences between Location, which had the lowest omission rate, and non-named entities.

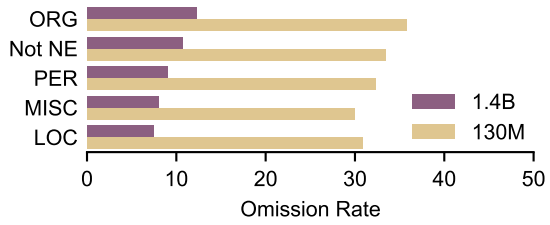

Figure 6: **Named Entities.** Omission rates by NE type on CoNLL-2003. Organizations have the highest omission rate for both models.

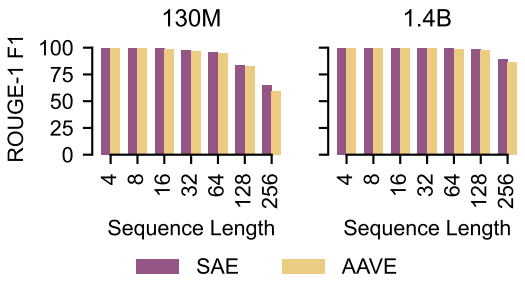

Figure 7: **Dialect.** Compared to Standard American English (SAE), the reconstruction fidelity of African American Vernacular (AAVE) degrades more with increased sequence length.

4.4 Sequence-Level Trends

In order to understand what types of sequences Mamba tends to retain versus forget in its hidden state, we look at reconstruction fidelity across (i) different subsets of Pile Uncopyrighted, (ii) different dialects, and (iii) synthetic datasets.

Pile Subsets We test how well Mamba's hidden state preserves different types of information by evaluating reconstruction performance across diverse textual domains. We sample 700 sequences each from nine subsets of the Pile Uncopyrighted—Common Crawl, ArXiv, NIH, GitHub, PubMed Central, Stack Exchange, Enron Email, Free Law, and DM Mathematics—capturing a broad range of linguistic characteristics, including web text, scientific writing, code, legal documents, and mathematical content.

Across model sizes, we find that the fidelity of reconstruction deteriorates the most for mathematical data as sequence length increases. For the 130M model, the ROUGE F1-score for the DM Mathematics subset is 99.9 for a sequence length of 4, which is on par with the other subsets. At a sequence length of 256, reconstruction performance on DM Mathematics drops sharply, with an F1-score of 41.6—substantially lower than other subsets, such as Common Crawl, which achieves 69.7 (Figure 8). In contrast, the remaining subsets show relatively similar performance to one another. This further supports our finding from token-level experiments that Mamba's hidden state struggles to retain information from mathematical inputs.

Dialects To evaluate if dialect impacts Mamba's retention, we use the AAVE/SAE paired dataset (Groenwold et al., 2020), which contains parallel instances in African American Vernacular English (AAVE) and Standard American English (SAE). For consistency, instances are concatenated to a fixed maximum length, ranging from 4–256 tokens.

For short sequences, ROUGE F1-scores for reconstruction are high and nearly identical for SAE and AAVE. For a 130M model at a sequence length of 4, SAE achieves 99.7 while AAVE is at 99.6 ($\Delta = 0.1$; Figure 7). However, as sequence length increases, the performance gap between the two widens. At a sequence length of 256 tokens, SAE reconstruction F1 declines to 65.3, while AAVE falls further to 59.7 ($\Delta = 5.6$). We see similar trends across model sizes (Figure 7).

Synthetic Numeric Sequences We assess how well purely numerical sequences can be reconstructed by creating a synthetic evaluation dataset. For each sequence length (4–256 tokens), we randomly sample numeric tokens from Mamba's vocabulary to generate 1K instances of the corresponding length.

We found that accuracy declines significantly more with length compared to natural text. For a 130M-parameter model at 256 tokens, the F1-score drops from 66.5 for standard text to just 0.5 for numbers (Figure 9). We see similar trends across model sizes.

Difference Between Reference and Generated Numbers To understand the character-level deviations involved in retaining numeric information, we compute the Levenshtein distance between incorrectly generated numeric tokens and their corresponding references on the CoNLL-2003 and Pile evaluation sets. Across datasets and model sizes, the median reference token length ranged from 2–3, with a median edit distance of 2 (Appendix D). These values show that for errors on numerical tokens, most characters do not match.

To understand the magnitude of this difference

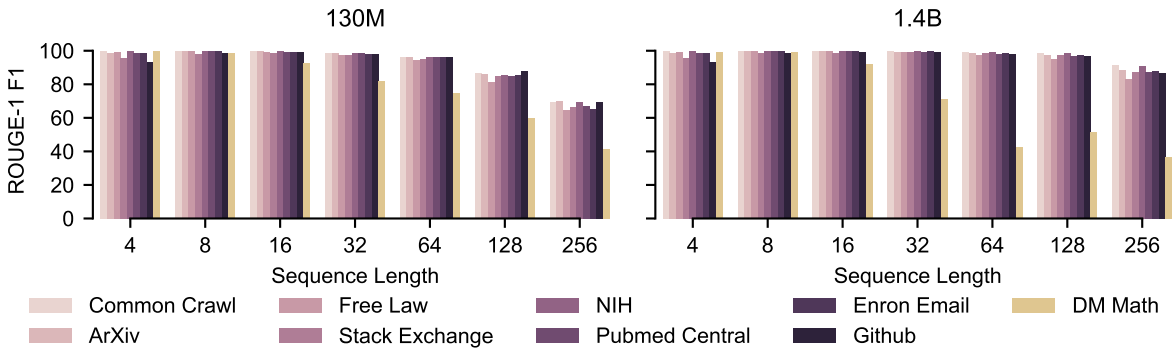

Figure 8: **Reconstruction Errors by Input Source.** Reconstruction fidelity across subsets of Pile for Mamba. As sequence length increases, reconstruction for the DM Math subset deteriorates the most compared to other subsets.

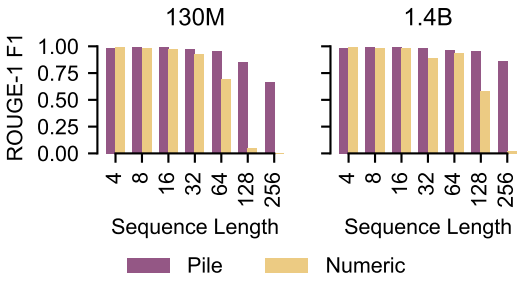

Figure 9: **Numerical Sequences.** Comparison of reconstruction fidelity of randomly sampled numeric sequences versus natural sequences from the Pile.

between reference and generated numbers, we extract numbers from incorrect instances and compute the Mean Absolute Percentage Error (MAPE), which expresses the mean absolute error as a ratio of the reference values, providing a scale-independent metric for error. On the CoNLL-2003 test set, the generated numeric token magnitudes for the 130M model deviate from the reference tokens by an average of 8.4× their magnitude. The MAPE decreases steadily with increasing model size, reaching 1.9 at 1B parameters. When evaluated on the Pile dataset, the MAPE increases with chunk length across model sizes, rising from approximately 0 at a chunk size of 4 to 38.2 at a chunk size of 256 (Appendix D). These results suggest that larger models better preserve the magnitudes of numeric tokens, whereas numeric tokens in longer sequences are retained poorly with high deviation.

Repeated Tokens We test whether Mamba's hidden state retains both token identity and repetition count by creating a synthetic dataset of 2K randomly sampled tokens from its vocabulary. Each token is repeated n times, where n ranges from 4 to 256. For a 130M model, repeated tokens were correctly generated $> 90\%$ of times across chunk sizes (Table A4). However, only at a sequence length of 4 did the model consistently reproduce the correct number of repetitions. For larger sequence lengths, the model instead continuing generation until reaching the maximum token limit of 300.

Memory and Perplexity We then examine whether Mamba's hidden state has more difficulty retaining information about sequences that are less likely under the model, i.e., higher perplexity, by correlating reference perplexity with omission rates in the evaluation subset of the Pile. For a 130M Mamba model at a sequence length of 256, we observe that as the perplexity of input sequences increases, so does the omission rates of their reconstructions (Figure 10). Thus, it is harder for Mamba to accurately store information about higher perplexity sequences.

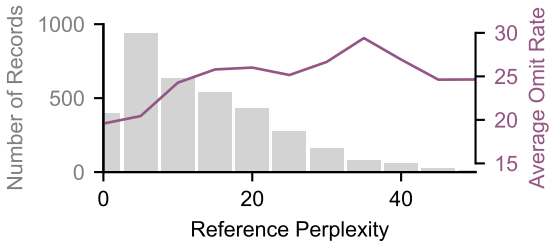

Figure 10: **Perplexity vs. Omission Rates.** We observe that as the perplexity of input sequences increases, so does the omission rate of their reconstructions.

4.5 Training Corpus Analysis

To investigate why certain types of tokens and sequences are more prone to omission in Mamba, we examine its training data. We randomly sample

200K instances from Pile Uncopyrighted by first randomly selecting a partition of the dataset, and then applying reservoir sampling. We annotate this sample using the Stanza part-of-speech tagger.

We find that numerical tokens (NUM) exhibit the lowest count per unique token among all categories evaluated, while, determiners (DET)—which have the lowest omission rates—have the highest count per unique token (Table 2). This shows an association between omission rates and pre-training frequencies.

PoS	Total	Unique	Ratio
DET	14M	887	16K
PRON	7.6M	2.4K	3.1K
ADP	18M	5.9K	3.0K
PUNCT	43M	73K	584
ADV	5.6M	17K	325
SYM	6.0M	21K	293
VERB	16M	79K	205
ADJ	12M	93K	130
NOUN	47M	158K	30
NUM	9.0M	643K	14

Table 2: **Part-of-Speech Distribution in Pile Sample.** In a 200K random sample of the Pile Uncopyrighted, numbers (NUM) have the lowest count per unique token, which may contribute to their poor retention.

5 Discussion

Our main contribution is an evaluation of the selective memory of the pretrained Mamba *language models*. We emphasize this term to highlight our focus on the impact language model pretraining has on information retention, as opposed to fundamental limitations of the Mamba architecture. Our findings show that Mamba LMs are especially prone to forgetting certain types of information (e.g., numeric tokens) and this appears to be driven in part by occurrence in pretraining data. In addition to our analysis, our work contributes a general framework for diagnosing memory limitations in LMs with fixed-size hidden states, offering a tool for future research. As LMs are often used as black-box systems, understanding their limitations is a critical first step toward mitigating them. This paper aligns with existing evaluation-focused research in NLP (Hessel et al., 2023; Lin et al., 2022; Moghe et al., 2023; Selvam et al., 2023) that sheds light on issues and guides future work.

Building on our findings, we suggest that the conventional pretraining pipeline may not be ideally suited to the inductive biases of SSM-based language models. One aspect to reconsider is tokenization. Recent work has demonstrated that byte-level SSM language modeling can outperform transformers on various benchmarks (Wang et al., 2024), indicating that tokenization-free approaches may provide a more natural inductive bias for SSMs (Gu, 2025). Tokenization choice also impacts numerical reasoning; Singh and Strouse (2024) demonstrate that tokenizing numbers as single digits yields better performance on mathematical tasks. Similarly, Yang et al. (2025) show that right-to-left tokenization can enhance numeracy.

The pretraining objective itself is yet another direction to revisit. Auxiliary reconstruction losses and memory-augmenting self-supervised objectives improve memory retention in RNN and LSTM models (Zhang et al., 2021; Trinh et al., 2018). Similarly, arithmetic-aware pretraining strategies, such as contrastive learning and task-specific auxiliary losses, enhance numerical reasoning in LLMs (Petrak et al., 2023; Li et al., 2022a). We hope that future work will address the selective memory issues in SSM language models by exploring pre-training strategies tailored to SSM architectures.

6 Conclusion

In this work, we characterized the types of tokens (e.g., parts of speech, named entities) and sequences (e.g., code, mathematical content) that are most susceptible to being forgotten by Mamba LMs. By training auto-encoders to reconstruct inputs directly from Mamba's hidden states, we were able to assess information retention without relying on explicit memorization cues like previous works. Our results showed substantially greater information loss for math-related tokens as well as for organization names and non-standard English dialects. These findings highlight that Mamba's memory is clearly selective, which could pose a problem for applying it to tasks that require exactly recalling the types of tokens it has a tendency to forget, e.g., math reasoning, information retrieval, and open-domain dialogue tasks. By identifying these weaknesses, our work provides motivation and direction for future research into methods that control Mamba's selective memory so that it is able to better retain pertinent information. We release our full implementation at `https://github.com/dataminr-ai/ouroboros` to support future work.

7 Limitations

Decoder Errors A limitation of our methodology is the potential for confounding between decoder errors and information loss in Mamba's hidden states. Since we train a decoder for text reconstruction, it is possible some of the reconstruction omissions we find are due to decoder errors as well as information lacking in Mamba's hidden states. Nevertheless, the successful replication of prior work with our method provides strong validation of our approach, suggesting that any impact from decoder errors is not significant enough to invalidate our core findings.

Single SSM Language Model Family We only evaluate on one SSM language model family, Mamba. The reason we chose these models is that, similar to how ChatGPT has been studied as a representative of state-of-the-art transformer LLMs (Gao et al., 2023; Laskar et al., 2023; Bang et al., 2023; Jang and Lukasiewicz, 2023), the pretrained Mamba models are representatives of state-of-the-art SSM LLMs. Additionally, there are no other reasonable pure SSM language models available for comparison.

Single Linear Architecture Our study focuses solely on one linear recurrent architecture, SSMs, though our approach can be applied to other architectures as well, such as RWKV (Peng et al., 2023). While exploring information retention across these different architectures is an important direction for future research (and if our paper is accepted, we will make our code publicly available to facilitate such work), this paper already required a substantial amount of resources to produce. For instance training the 1.4B decoder required multiple weeks on a 48GB A6000 GPU.

Solutions We do not attempt to solve the selective memory issues we find in Mamba, though we discuss some possible options in Section 5. However, identifying a problem is an essential first step toward developing an effective solution. Developing solutions, particularly for complex behaviors in large language models, is non-trivial and typically requires targeted, follow-up research. Our paper falls within the well-established tradition in NLP of evaluation-focused work (Hessel et al., 2023; Lin et al., 2022; Moghe et al., 2023; Selvam et al., 2023), and aims to shed light on an important issue that can guide future efforts toward mitigate identified issues.

Downstream Task We do not evaluate on downstream tasks. However, we identify a fundamental issue with Mamba: its difficulty in retaining crucial information, such as numbers, from context. This is an essential skill for techniques that are now standard in LLM reasoning, such as task decomposition (Wei et al., 2022; Yao et al., 2023). While downstream evaluation is beyond the scope of this paper, we encourage future researchers to explore this area.

Dataset We use samples from Pile Uncopyrighted to train our autoencoder and estimate the part-of-speech composition of Mamba's training data. However, there might be some distribution shift between Pile Uncopyrighted and The Pile, the original training dataset for the Mamba language models. However, this is unavoidable as the full Pile is no longer available for use.

Training Corpus Analysis While we demonstrate a strong connection between the forgetting of numerical data and the occurrence of numbers in Mamba's training corpus, we are unable to establish a similar connection for all types of information loss, such as non-standard dialects. This is due to the lack of available taggers for dialects, in contrast to the well-established taggers for parts-of-speech.

References

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. 2023. A multitask, multilingual, multimodal evaluation of ChatGPT on reasoning, hallucination, and interactivity. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 675–718, Nusa Dua, Bali. Association for Computational Linguistics.

Jinglong Gao, Xiao Ding, Bing Qin, and Ting Liu. 2023. Is ChatGPT a good causal reasoner? a comprehensive evaluation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11111–11126, Singapore. Association for Computational Linguistics.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. The Pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Sophie Groenwold, Lily Ou, Aesha Parekh, Samhita Honnavalli, Sharon Levy, Diba Mirza, and William Yang Wang. 2020. Investigating african-american vernacular english in transformer-based text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5877–5883, Online. Association for Computational Linguistics.

Albert Gu. 2025. On the tradeoffs of state space models and transformers. https://goombalab.github.io/blog/2025/tradeoffs/. Accessed: 2025-07-28.

Albert Gu and Tri Dao. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *ArXiv*, abs/2312.00752.

Albert Gu, Karan Goel, and Christopher Ré. 2022a. Efficiently modeling long sequences with structured state spaces. In *The International Conference on Learning Representations (ICLR)*.

Albert Gu, Isys Johnson, Karan Goel, Khaled Saab, Tri Dao, Atri Rudra, and Christopher Ré. 2021. Combining recurrent, convolutional, and continuous-time models with linear state-space layers. *Advances in Neural Information Processing Systems*, 34.

Albert Gu, Isys Johnson, Aman Timalsina, Atri Rudra, and Christopher Ré. 2022b. How to train your hippo: State space models with generalized orthogonal basis projections. *ArXiv*, abs/2206.12037.

Jack Hessel, Ana Marasovic, Jena D. Hwang, Lillian Lee, Jeff Da, Rowan Zellers, Robert Mankoff, and Yejin Choi. 2023. Do androids laugh at electric sheep? humor "understanding" benchmarks from the new yorker caption contest. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 688–714, Toronto, Canada. Association for Computational Linguistics.

Myeongjun Jang and Thomas Lukasiewicz. 2023. Consistency analysis of ChatGPT. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15970–15985, Singapore. Association for Computational Linguistics.

Samy Jelassi, David Brandfonbrener, Sham M. Kakade, and Eran Malach. 2024. Repeat after me: Transformers are better than state space models at copying. *ArXiv*, abs/2402.01032.

Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. 2018. Disentangled representation learning for non-parallel text style transfer. In *Annual Meeting of the Association for Computational Linguistics*.

Md Tahmid Rahman Laskar, M Saiful Bari, Mizanur Rahman, Md Amran Hossen Bhuiyan, Shafiq Joty, and Jimmy Huang. 2023. A systematic study and comprehensive evaluation of ChatGPT on benchmark datasets. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 431–469, Toronto, Canada. Association for Computational Linguistics.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhongli Li, Wenxuan Zhang, Chao Yan, Qingyu Zhou, Chao Li, Hongzhi Liu, and Yunbo Cao. 2022a. Seeking patterns, not just memorizing procedures: Contrastive learning for solving math word problems. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2486–2496, Dublin, Ireland. Association for Computational Linguistics.

Zhuang Li, Lizhen Qu, Qiongkai Xu, Tongtong Wu, Tianyang Zhan, and Gholamreza Haffari. 2022b. Variational autoencoder with disentanglement priors for low-resource task-specific natural language generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10335–10356, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Giangiacomo Mercatali and André Freitas. 2021. Disentangling generative factors in natural language with discrete variational autoencoders. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3547–3556, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Nikita Moghe, Tom Sherborne, Mark Steedman, and Alexandra Birch. 2023. Extrinsic evaluation of machine translation metrics. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13060–13078, Toronto, Canada. Association for Computational Linguistics.

Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Leon Derczynski, Xingjian Du, Matteo Grella, Kranthi Gv, Xuzheng He, Haowen Hou, Przemyslaw Kazienko, Jan Kocon, Jiaming Kong, Bartłomiej Koptyra, and 13 others. 2023. RWKV: Reinventing RNNs for the transformer era. In *Findings of the Association for Com-*

putational Linguistics: EMNLP 2023, pages 14048–14077, Singapore. Association for Computational Linguistics.

Dominic Petrak, Nafise Sadat Moosavi, and Iryna Gurevych. 2023. Arithmetic-based pretraining improving numeracy of pretrained language models. In *Proceedings of the 12th Joint Conference on Lexical and Computational Semantics (*SEM 2023)*, pages 477–493, Toronto, Canada. Association for Computational Linguistics.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Nikil Selvam, Sunipa Dev, Daniel Khashabi, Tushar Khot, and Kai-Wei Chang. 2023. The tail wagging the dog: Dataset construction biases of social bias benchmarks. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1373–1386, Toronto, Canada. Association for Computational Linguistics.

Aaditya K. Singh and DJ Strouse. 2024. Tokenization counts: the impact of tokenization on arithmetic in frontier llms. *ArXiv*, abs/2402.14903.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, and 3 others. 2024. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Trieu H. Trinh, Andrew M. Dai, Thang Luong, and Quoc V. Le. 2018. Learning longer-term dependencies in rnns with auxiliary losses. *ICML*.

J. Vasilakes, Chrysoula Zerva, Makoto Miwa, and Sophia Ananiadou. 2022. Learning disentangled representations of negation and uncertainty. *ArXiv*, abs/2204.00511.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Neural Information Processing Systems*.

Roger Waleffe, Wonmin Byeon, Duncan Riach, Brandon Norick, Vijay Anand Korthikanti, Tri Dao, Albert Gu, Ali Hatamizadeh, Sudhakar Singh, Deepak Narayanan, Garvit Kulshreshtha, Vartika Singh, Jared Casper, Jan Kautz, Mohammad Shoeybi, and Bryan Catanzaro. 2024. An empirical study of mamba-based language models. *ArXiv*, abs/2406.07887.

Junxiong Wang, Tushaar Gangavarapu, Jing Nathan Yan, and Alexander M. Rush. 2024. Mambabyte: Token-free selective state space model. *ArXiv*, abs/2401.13660.

Junxiong Wang, Wen-Ding Li, Daniele Paliotta, Daniel Ritter, Alexander M. Rush, and Tri Dao. 2025a. M1: Towards scalable test-time compute with mamba reasoning models. *ArXiv*, abs/2504.10449.

Peihao Wang, Ruisi Cai, Yuehao Wang, Jiajun Zhu, Pragya Srivastava, Zhangyang Wang, and Pan Li. 2025b. Understanding and mitigating bottlenecks of state space models through the lens of recency and over-smoothing. In *International Conference on Learning Representations (ICLR)*. Under review.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed H. Chi, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. *ICLR*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, F. Xia, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *NeurIPS*.

Haotong Yang, Yi Hu, Shijia Kang, Zhouchen Lin, and Muhan Zhang. 2025. Number cookbook: Number understanding of language models and how to improve it. *ICLR*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *NeurIPS*.

Yingji Zhang, Danilo S. Carvalho, Ian Pratt-Hartmann, and André Freitas. 2023. Learning disentangled semantic spaces of explanations via invertible neural networks. In *Annual Meeting of the Association for Computational Linguistics*.

Zhu Zhang, Chang Zhou, Jianxin Ma, Zhijie Lin, Jingren Zhou, Hongxia Yang, and Zhou Zhao. 2021. Learning to rehearse in long sequence memorization. *ICML*.

A Mamba Details

Model	Layers	β_S	β_c
130M	24	[1536, 16]	[1536,4]
370M	48	[2048,16]	[2048,4]
790M	48	[3072,16]	[3072,4]
1.4B	48	[4096,16]	[4096,4]

Table A1: **Hidden State Sizes.** The encoder maps input texts into a hidden state with two parts: (i) an SSM State, β_S, and (ii) a Convolutional State, β_C. The sizes β_S and β_C per layer for encoders of various sizes between 130m and 1.4B are shown.

B Additional 130M Results

Pair	T-Statistic	Adjusted P-Value
NUM - SYM	-4.4	8.8e-04
NUM - ADJ	-15.4	5.1e-51
NUM - NOUN	-25.1	1.8e-135
NUM - ADV	-11.3	1.9e-27
NUM - VERB	-20.8	4.6e-92
NUM - PUNCT	-22.8	1.6e-110
NUM - PRON	-12.3	1.1e-32
NUM - PRT	-8.8	7.8e-17
NUM - CONJ	-11.7	1.3e-29
NUM - ADP	-26.1	3.0e-144
NUM - DET	-24.8	1.1e-129
SYM - DET	-3.8	1.1e-02
ADJ - ADP	-5.6	1.9e-06
ADJ - DET	-7.3	2.1e-11
NOUN - ADP	-7.6	1.7e-12
NOUN - DET	-9.1	4.6e-18
ADV - ADP	-3.5	2.7e-02
ADV - DET	-5.1	1.9e-05
VERB - ADP	-5.3	6.6e-06
VERB - DET	-7.2	3.4e-11
PUNCT - ADP	-4.2	1.9e-03
PUNCT - DET	-6.3	2.4e-08
PRON - DET	-3.9	5.6e-03

Table A2: **Statistical Tests (Part-of-Speech)**. Part-of-speech categories that have statistically different omission rates based on pair-wise t-tests at $\alpha = 0.05$ with Bonferroni corrected p-values for Mamba (130M).

Pair	T-Statistic	Adjusted P-Value
Organization - Location	-3.4	6.3e-03
Organization - Misc	-3.2	1.5e-02

Table A3: **Statistical Tests (Named Entity Recognition)**. Named entity categories that have statistically different commission rates based on pair-wise t-tests at $\alpha = 0.05$ with Bonferroni corrected p-values for Mamba (130M).

Length	Present (%)	Repeat Mode
4	99.9	4
8	99.8	300
16	100.0	300
32	100.0	300
64	99.9	300
128	99.7	300
256	92.1	300

Table A4: **Repeated Tokens.** The percentage of generations that contained the correct repeated tokens and the modal repetition count observed in those generations.

C Additional 1.4B Results

Pair	T-Statistic	Adjusted P-Value
NUM - PUNCT	-8.6	6.5e-16
NUM - PRT	-5.2	1.2e-05
NUM - NOUN	-28.0	1.0e-167
NUM - SYM	-5.0	3.4e-05
NUM - VERB	-23.3	5.4e-115
NUM - ADJ	-18.7	2.4e-74
NUM - CONJ	-11.0	7.4e-26
NUM - ADV	-12.7	5.4e-35
NUM - PRON	-12.8	2.4e-35
NUM - ADP	-26.5	1.2e-148
NUM - DET	-21.5	8.6e-98
PUNCT - NOUN	-15.9	7.7e-55
PUNCT - VERB	-15.1	2.4e-49
PUNCT - ADJ	-12.8	2.5e-35
PUNCT - CONJ	-7.7	9.5e-13
PUNCT - ADV	-9.0	2.6e-17
PUNCT - PRON	-9.1	8.1e-18
PUNCT - ADP	-18.7	2.4e-75
PUNCT - DET	-15.3	1.6e-50
PRT - VERB	-4.0	3.6e-03
PRT - ADJ	-4.7	2.2e-04
PRT - CONJ	-4.0	5.1e-03
PRT - ADV	-4.3	1.1e-03
PRT - PRON	-4.5	4.6e-04
PRT - ADP	-6.4	1.4e-08
PRT - DET	-6.1	9.0e-08
NOUN - VERB	-4.0	4.1e-03
NOUN - ADJ	-4.5	5.1e-04
NOUN - ADV	-3.4	4.7e-02
NOUN - PRON	-3.6	1.8e-02
NOUN - ADP	-8.5	1.1e-15
NOUN - DET	-6.9	3.4e-10
VERB - ADP	-4.2	2.2e-03
VERB - DET	-3.6	2.1e-02

Table A6: **Statistical Tests (Part-of-Speech)**. Part-of-speech categories that have statistically different omission rates based on pair-wise t-tests at $\alpha = 0.05$ with Bonferroni corrected p-values for Mamba (1.4B).

Token	Omit Rate	Token	Omit Rate	Token	Omit Rate	Token	Omit Rate	Token	Omit Rate
␣(-	43.8	␣w	15.1	␣7	13.2	␣four	10.7	w	9.6
What	23.5	␣Suppose	14.8	␣l	13.0	␣d	10.6	␣t	9.5
Suppose	23.0	␣z	14.6	␣5	12.7	␣20	10.4	␣prime	9.5
␣List	21.9	␣17	14.4	␣b	12.4	␣n	10.3	␣Let	9.4
␣What	17.8	␣6	13.9	21	11.4	␣15	10.3	␣f	9.4
␣13	17.0	␣u	13.7	␣9	11.4	))	10.0	␣g	9.1
␣11	17.0	␣14	13.6	␣12	11.2	␣16	9.9	␣h	9.1
␣4	16.8	␣j	13.3	␣3	11.2	␣2	9.9	?	8.7
␣times	15.6	␣y	13.3	␣c	11.0	␣18	9.7	**	8.7
␣o	15.1	␣s	13.2	␣k	10.8	␣8	9.6	␣1	8.4

Table A5: **Top 50 Omitted Tokens.** The top 50 tokens most forgotten words in natural text across sequence lengths for Mamba (1.4B).

Pair	T-Statistic	Adjusted P-Value
Organization - Person	-3.9	1.0e-03
Organization - Misc	-3.6	3.7e-03
Organization - Location	-5.2	1.8e-06
Not NE - Location	-4.4	9.8e-05

Table A7: **Statistical Tests (Named Entities)**. Named entity categories with statistically different omission rates based on pair-wise t-tests at $\alpha = 0.05$ with Bonferroni corrected p-values for Mamba (1.4B).

D How Different are Reference and Generated Numbers?

Character difference The lengths of reference numeric tokens that are incorrectly reconstructed are shown in Table A9, and the Levenshtein distance between reference and generated tokens are showns in Table A8.

Magnitude difference The Mean Absolute Percentage Error (MAPE) for numbers in incorrectly generated numeric tokens across CoNNL-2003 and Pile evaluation datasets are shown in Table A10.

Dataset	Model	Chunk	Mean	Median
connl	130m	256	1.9	2.0
connl	370m	256	1.9	2.0
connl	790m	256	1.8	2.0
connl	1b	256	1.9	2.0
pile	130m	4	4.1	2.0
pile	130m	8	1.7	2.0
pile	130m	16	1.8	2.0
pile	130m	32	2.0	2.0
pile	130m	64	2.0	2.0
pile	130m	128	2.2	2.0
pile	130m	256	2.2	2.0
pile	370m	4	0.0	0.0
pile	370m	8	1.6	2.0
pile	370m	16	2.0	2.0
pile	370m	32	2.2	2.0
pile	370m	64	1.9	2.0
pile	370m	128	2.1	2.0
pile	370m	256	2.2	2.0
pile	790m	4	2.0	2.0
pile	790m	8	1.7	2.0
pile	790m	16	1.8	2.0
pile	790m	32	1.9	2.0
pile	790m	64	1.9	2.0
pile	790m	128	2.0	2.0
pile	790m	256	2.2	2.0
pile	1b	4	1.8	2.0
pile	1b	8	1.8	2.0
pile	1b	16	2.0	2.0
pile	1b	32	2.1	2.0
pile	1b	64	1.9	2.0
pile	1b	128	2.0	2.0
pile	1b	256	2.2	2.0

Table A8: **Edit Distance.** Levenshtein distance between mismatching reference and generated numeric tokens on the CoNNL-2003 and Pile evaluation datasets

Dataset	Model	Chunk	Mean	Median
connl	130m	256	2.5	3.0
connl	370m	256	2.6	3.0
connl	790m	256	2.5	3.0
connl	1b	256	2.5	2.0
pile	130m	4	2.1	2.0
pile	130m	8	2.0	2.0
pile	130m	16	1.9	2.0
pile	130m	32	2.1	2.0
pile	130m	64	2.1	2.0
pile	130m	128	2.2	2.0
pile	130m	256	2.1	2.0
pile	370m	4	0.0	0.0
pile	370m	8	2.1	2.0
pile	370m	16	2.2	2.0
pile	370m	32	2.2	2.0
pile	370m	64	1.9	2.0
pile	370m	128	2.0	2.0
pile	370m	256	2.1	2.0
pile	790m	4	3.0	3.0
pile	790m	8	2.1	2.0
pile	790m	16	2.1	2.0
pile	790m	32	2.0	2.0
pile	790m	64	1.9	2.0
pile	790m	128	1.9	2.0
pile	790m	256	2.1	2.0
pile	1b	4	2.4	2.5
pile	1b	8	2.1	2.0
pile	1b	16	2.0	2.0
pile	1b	32	2.2	2.0
pile	1b	64	1.8	2.0
pile	1b	128	2.0	2.0
pile	1b	256	2.0	2.0

Table A9: **Length of Error Instance.** Length of incorrectly reconstructed reference tokens in CoNNL-2003 and Pile evaluation datasets

Dataset	Model	Chunk	MAPE
connl	130m	256	8.4181
connl	370m	256	5.9192
connl	790m	256	2.7187
connl	1b	256	1.9930
connl	*Avg*	256	**4.7622**
pile	130m	4	0.0005
pile	370m	4	0.0000
pile	790m	4	0.0000
pile	1b	4	0.0002
pile	*Avg*	4	**0.0002**
pile	130m	8	0.0013
pile	370m	8	0.0032
pile	790m	8	0.0026
pile	1b	8	0.0036
pile	*Avg*	8	**0.0027**
pile	130m	16	0.0439
pile	370m	16	0.0455
pile	790m	16	0.0214
pile	1b	16	0.0620
pile	*Avg*	16	**0.0432**
pile	130m	32	0.2199
pile	370m	32	0.1697
pile	790m	32	0.2208
pile	1b	32	0.1671
pile	*Avg*	32	**0.1944**
pile	130m	64	0.6067
pile	370m	64	0.2211
pile	790m	64	0.2175
pile	1b	64	0.4596
pile	*Avg*	64	**0.3762**
pile	130m	128	2.5763
pile	370m	128	1.0526
pile	790m	128	0.5142
pile	1b	128	0.4900
pile	*Avg*	128	**1.1583**
pile	130m	256	89.7517
pile	370m	256	5.9819
pile	790m	256	3.2479
pile	1b	256	53.8413
pile	*Avg*	256	**38.2057**

Table A10: **Magnitude Differences.** Mean Absolute Percentage Error (MAPE) for numbers in incorrectly generated numeric tokens across CoNNL-2003 and Pile evaluation datasets.