LongSSM: On the Length Extension of State-space Models in Language Modelling

Shida Wang[1]

[1]Department of Mathematics, National University of Singapore

April 3, 2024

Abstract

In this paper, we investigate the length-extension of state-space models (SSMs) in language modeling. Length extension involves training models on short sequences and testing them on longer ones. We show that state-space models trained with zero hidden states initialization have difficulty doing length extension. We explain this difficulty by pointing out the length extension is equivalent to polynomial extrapolation. Based on the theory, we propose a simple yet effective method - changing the hidden states initialization scheme - to improve the length extension. Moreover, our method shows that using long training sequence length is beneficial but not necessary to length extension. Changing the hidden state initialization enables the efficient training of long-memory model with a smaller training context length.

1 Introduction

Large language models [1] are usually trained on large corpus with a fixed context length (e.g., 2048 tokens). However, attention-based transformer [1] has an $O(T^2)$ asymptotic growth with respect to the sequence length. The cost for training and inference is even higher when we are working with long sequences. Recently, state-space models [2, 3, 4, 5] and linear-attention-based transformers [6, 7, 8] have shown the potential to replace the attention-based transformers [1]. SSMs are recurrent models characterized by parallelism in sequence length and inference cost that remains independent of length.

Despite state-space models having a recurrent form and thereby inducing an **"infinite-in-time" memory** of the input history, they tend to exhibit limited length extension beyond the training sequence length in mamba [4]. In practical applications, where the target inference context often exceeds the length of the training sequence and can even be infinite, a pertinent question arises: *Is it possible to train a model with the ability to extend its memory beyond the constraints of a finite training sequence length?* The assumption of a finite training sequence length is both reasonable and necessary, given the constraints of GPU memory and the comparatively short training length, especially when compared with the infinite inference length found in real-world applications.

In the earliest transformer model [9], achieving length extension is challenging, usually constrained by the limitations of absolute position encoding [9, 10]. Press et al. [10] have demonstrated that introducing attention with linear bias serves as an effective solution to address this limitation and enable length extension. Apart from the additive bias, another stream of works is constructing relative position embedding [11, 12, 13]. In this paper, we adopt the backpropagation through time method that is orthogonal to these previous approaches, and can be used to improve state-space models' length extension capability. Moreover, our method shows that the length extension capability can be achieved without using a long training sequence (Figure 4).

We summarize our main contributions as follow:

1. We show why the zero hidden states initialization scheme has difficulty doing length extension.
2. Based on the difficulty for zero-initialization case, we introduce the training approach that leverages previous hidden states with no batch-level shuffling.
3. We show the length extension can be achieved without a long training sequence length. In particular, we show the feasibility to train a model with **training sequence length 16** and **truncated BPTT**, but has **length extension up to 32768**.

Table 1: Comparison of asymptotic training/inference step cost for attention-based transformers [1] and state-space models with respect to context length T.

	Attention-based transformer	State-space models/Linear-attention
Training cost	$O(T^2)$	$O(T)$
Inference cost	$O(T^2)$	$O(1)$

Notation We use the bold face to represent the sequence while then normal letters are scalars, vectors or functions. We use $\|\cdot\|$ to denote norms over sequences of vectors, or functions, while $|\cdot|$ (with subscripts) represents the norm of number, vector or weights tuple. Here $|x|_\infty := \max_i |x_i|, |x|_2 := \sqrt{\sum_i x_i^2}, |x|_1 := \sum_i |x_i|$ are the usual max (L_∞) norm, L_2 norm and L_1 norm. Let m be the hidden dimension and d be the input dimension.

2 Background

In this section, we first introduce the state-space models (SSMs). Compared with traditional nonlinear RNNs, they have better parallelism across sequence length in the sense that fast Fourier transform and associative scan can be used to reduce the training latency. Next, we give the definition of three types of length extension capability. The aim of this paper is not to improve the length extension towards a particular length but to achieve the monotonic perplexity decrease for weak length extension.

2.1 State-space models

State-space models [3] have layer-wise nonlinear activations while the traditional nonlinear RNNs have recurrent nonlinear activations (see the comparison of SSMs and RNNs in Appendix A).

$$h_{k+1} = Wh_k + (Ux_k + b), \quad h_0 = 0 \in \mathbb{R}^m \tag{1}$$

$$\hat{y}_k = C\boldsymbol{\sigma}(h_k), \quad 1 \leq k \leq T. \tag{2}$$

Here $h_k \in \mathbb{R}^m, W \in \mathbb{R}^{m\times m}, U \in \mathbb{R}^{m\times d}, b \in \mathbb{R}^m, C \in \mathbb{R}^{d\times m}$. The corresponding continuous-time form is

$$\frac{dh_t}{dt} = Wh_t + (Ux_t + b), \quad \hat{y}_t = C\boldsymbol{\sigma}(h_t). \tag{3}$$

The solution of h_t in convolution form is $h_t = h_0 + \int_0^t e^{W(t-s)}(Ux_s + b)ds$.

The convolution form of SSMs Based on the above continuous-time formulation, hidden states sequences can be written into the following convolution form

$$\mathbf{h} = \rho(t) \circledast (U\mathbf{x} + b) \tag{4}$$

The convolution kernel is $\rho(t) = e^{Wt}$. Given the convolution form in Equation (4), FFT [3] can be used to accelerate the computation. Compared with $O(T^2)$ cost of attention matrix, it only takes $O(T \log T)$ to evaluate the hidden states $\mathbf{h}$ and corresponding outputs $\hat{\mathbf{y}}$.

Scan-based acceleration for models with input-dependent gating Recent advancements in state-space models have significantly enhanced their expressiveness and approximation capabilities through the incorporation of input-dependent gating mechanisms. The input-dependent gating refers to the generalization of W and $Ux_k + b$ to $W(x_k) \in \mathbb{R}^{m\times d}$ and $U(x_k) \in \mathbb{R}^{m\times d}$.

$$h_{k+1} = W(x_k) \odot h_k + U(x_k), \quad h_0 = 0 \in \mathbb{R}^{m\times d} \tag{5}$$

$$\hat{y}_k = C\boldsymbol{\sigma}(h_k), \quad 1 \leq k \leq T. \tag{6}$$

Here $\odot$ is the element-wise product.

As input-dependent gating disrupts the convolution structure and negates the speed benefits derived from FFT, it is still feasible to employ scan-based acceleration techniques [14], achieving the $O(T)$ training cost. In Appendix C.1, we show the associativity of the following binary operator $\circ$ defined over tuple (W, h):

$$(W_1, h_1) \circ (W_2, h_2) = (W_2 \odot W_1, h_1 + W_1 \odot h_2). \tag{7}$$

The initialization is $h_0 = 0$ and hidden states h_k can be achieved from

$$(\_, h_k) = (W(x_k), U(x_k)) \circ \cdots \circ (W(x_1), U(x_1)) \circ (I, 0). \tag{8}$$

If we embed $U(x_k)$ in $\mathbb{R}^{m\times m}$ rather than $\mathbb{R}^{m\times d}$, this corresponds to the gated linear attention [8] whose hidden states are 2D square matrices. We summarize the differences in Table 2 of Appendix C.1.

2.2 Length extension

Length extension has been widely studied for transformers [10, 11, 12, 13]. This is an essential attribute for models designed for infinite contexts windows (writing novels [15], autonomous driving [16], online learning [17]). However, it is shown that the state-of-the-art Mamba [4] failed to achieve length extension beyond 4k[1].

Building upon existing research in length extension, we initially establish specific concepts to qualitatively classify models based on their capability to extend length.

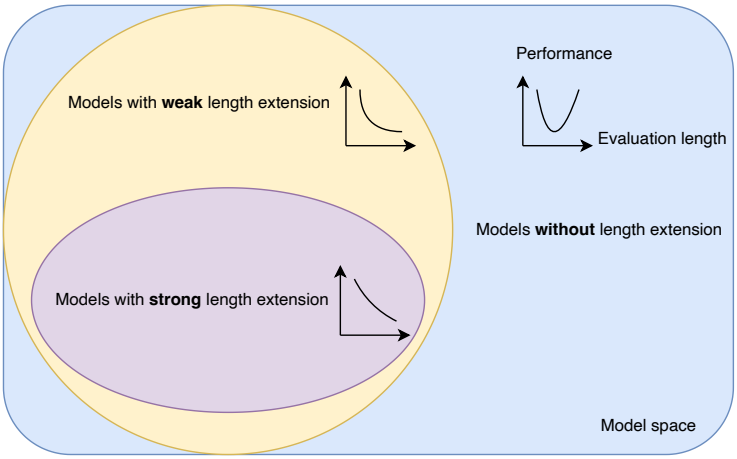


Figure 1: Three types of length-extension capabilities.

Definition 2.1. For auto-regressive language modeling, the entropy $H(p) = -\sum_x p(x) \log p(x)$ of the target language p is fixed. Here we define three types of length-extension capability based on the monotonicity of the perplexity:

1. **(Strong length extension)**: For some $T_0 > 0$, $\forall T > T_0, \text{perplexity}_{T+1} < \text{perplexity}_T$.
2. **(Weak length extension)**: For some $T_0 > 0$, $\forall T > T_0, \text{perplexity}_{T+1} \leq \text{perplexity}_T$.
3. **(No length extension)**: If there does not exists T_0 such that weak length extension holds.

As demonstrated in Figure 1, models with strong length extension are a subset of those with weak length extension.

In Figure 2, we evaluate the length extension difficulty for Mamba across different model sizes. Mamba is trained with sequence length $T = 2048$ and has difficulty maintaining the small perplexity beyond length $T \geq 4096$.

[1] https://openreview.net/forum?id=AL1fq05o7H, https://imgbb.com/XVT0hGJ

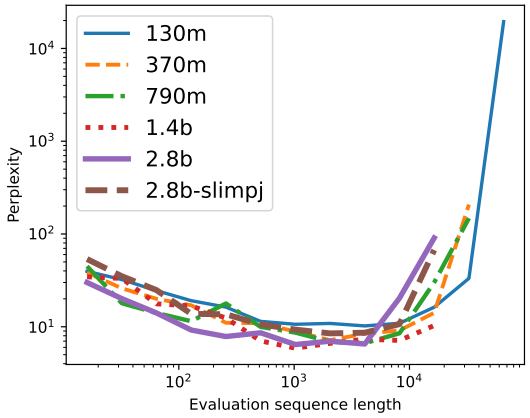


Figure 2: Length extension performance of Mamba evaluated over the Pile dataset [18]. The models are trained with a sequence length of 2048. Although perplexity remains finite for sequences up to 4096, it increases significantly for lengths beyond 8192.

Based on the above definitions, a natural question arises: *Can length extension exist for autoregressive language modeling?* We demonstrate that if a language is viewed as a shift-invariant sequence of random variables, then the weak length extension exists.

Theorem 2.2 (Existence of weak length extension in autoregressive language modeling)**.** *Assume the entropies of language across different sequence lengths are all finite. Consider the autoregressive language modeling as the learning of sequence of random variables* $\{X_k\}_{k=1}^{\infty}$*. The ideal autoregressive language models return the next random variable* $\mathbf{X}_{T+1}$ *based on the previous random variables* $X_{[0,\dots,T]}$*.*

$$\textbf{Model}((X_1,\dots,X_T)) = X_{T+1}. \tag{9}$$

Consider the entropy of this autoregressive language model

$$H((X_1,\dots,X_T)) = -\sum_{x_i\in X_i} p((x_1,\dots,x_T))\log p((x_1,\dots,x_T)) \tag{10}$$

$$= \sum_{i=1}^{T} H(X_i|X_1,\dots,X_{i-1}). \tag{11}$$

By monotonicity of entropy and shift-invariant property, we know $H(X_i|X_1,\dots,X_{i-1}) \leq H(X_i|X_2,\dots,X_{i-1}) = H(X_{i-1}|X_1,\dots,X_{i-2})$*. By the boundedness of* H *we know* $\lim_{i\to\infty} H(X_i|X_1,\dots,X_{i-1}) = 0$*.*

See the proof based on the information theory in Appendix C.3. By the universal approximation property of recurrent state-space models [19], we know the target autoregressive next-word prediction sequence $\Big(X_1, (X_2|X_1), \dots, (X_T|X_1,\dots,X_{T-1})\Big)$

can be approximated by the recurrent model $\left(\mathbf{Model}(\emptyset), \mathbf{Model}(X_1), \ldots, \mathbf{Model}(X_1, \ldots, X_{T-1})\right)$. Therefore the entropy of sequence model $\lim_{T\to\infty} H(\mathbf{Model}(X_1, \ldots, X_{T-1})) = \lim_{T\to\infty} H(X_T|X_1, \ldots, X_{T-1})$ is also decaying to 0.

3 Main results

In this section, we present a theoretical analysis of the challenges associated with length extension in SSMs that have zero-initialized hidden states, as detailed in Section 3.1. We illustrate that doing well in length extension is analogous to performing well in polynomial extrapolation. In Section 3.2, we argue that setting proper initialization for hidden states can transform the extrapolation challenge into an interpolation problem, thereby improving length extension performance.

3.1 Length extension is extrapolation

In approximation theory, state-space models are universal approximators for bounded causal continuous time-homogeneous regular nonlinear functionals, as detailed by Wang and Xue [19]. This outcome ensures the existence of a suitable model capable of learning the sequence-to-sequence relationships over unbounded time horizon $(-\infty, t)$ with any desirable tolerance. In practice, models are trained with a fixed finite length T. Therefore, it becomes crucial to assess whether such finite-window-trained models can effectively capture long-term memory beyond their training scope. In this context, we explore length extension in a simplified linear framework, which can be similarly extended to multi-layer nonlinear SSMs.

Consider the learning of linear functionals [20, 21] by single-layer state-space model, this linear functional target comes with a unique representation: $y_t = \mathbf{H}_t(\mathbf{x}) = \int_0^\infty \rho_s x_{t-s} ds$ with $|\rho|_1 := \int_0^\infty |\rho_s| ds < \infty$ while the single-layer state-space model (without layerwise activation) can be represented by $\hat{y}_t = \widehat{\mathbf{H}}_t(\mathbf{x}) = \int_0^T Ce^{Ws}Ux_{t-s}ds + \hat{y}_0$. The learning of target $\mathbf{H}$ by model $\widehat{\mathbf{H}}$ is equivalent to approximating the memory function $\rho(t) : [0, \infty) \to \mathbb{R}$ with the SSM memory kernel $\hat{\rho}(t) = Ce^{Wt}U$. Consider the following error decomposition

$$
\begin{aligned}
|y_T - \hat{y}_T| &= \left| \int_0^\infty \rho_s x_{T-s} ds - \left(\int_0^T \hat{\rho}_s x_{T-s} ds + \hat{y}_0 \right) \right| \\
&\leq \left| \int_T^\infty \rho_s x_{T-s} ds - \hat{y}_0 \right| + \left| \int_0^T \rho_s x_{T-s} ds - \int_0^T \hat{\rho}_s^* x_{T-s} ds \right| + \left| \int_0^T \hat{\rho}_s^* x_{T-s} ds - \int_0^T \hat{\rho}_s x_{T-s} ds \right|.
\end{aligned}
$$

Here $\hat{\rho}^*$ is the "optimal" model memory function while $\hat{\rho}$ is the achieved model memory function. The three terms in the error decomposition correspond to the **length extension error**, **finite time approximation error**, **optimization error**. For any fixed target $\mathbf{H}$, as the hidden dimension m increases, the finite time approximation error decays to 0. Given sufficient data and abundant computational resources, the optimization error decreases to zero through gradient-based optimization. However,

the length extension error cannot be reduced by simply increasing hidden dimension or improve the training over finite context data.

During the inference, the error decomposition for $t > T$ is

$$|y_t - \hat{y}_t| \leq \left| \int_t^\infty \rho_s x_{t-s} ds - \hat{y}_0 \right| + \left| \int_T^t (\rho_s - \hat{\rho}_s) x_{t-s} ds \right| + \left| \int_0^T (\rho_s - \hat{\rho}_s) x_{t-s} ds \right|. \tag{12}$$

With the first error unobserved and third error minimized in training, the major error for length extrapolation is the second term which be bounded by the form of $\int_T^t |\rho_s - \hat{\rho}_s| ds$. By change of variable $u = e^{-s}$, take $\mathcal{T}\rho_u = \rho_{-\log u}, u \in (0, 1]$.

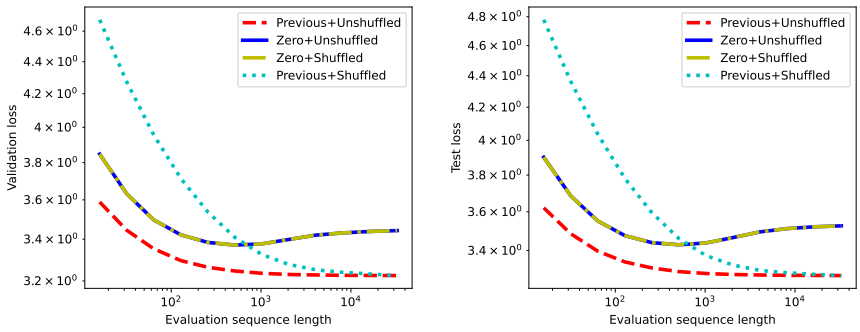
$$\int_T^t |\rho_s - \hat{\rho}_s| ds = \int_{e^{-t}}^{e^{-T}} \left| \mathcal{T}\rho_u - \sum_{k=1}^m c_i u_i^\lambda \right| \frac{1}{u} du. \tag{13}$$

The error between $[T, t]$ is equivalent to evaluate the polynomial extrapolation error of $\frac{\mathcal{T}\rho_u}{u}$ over $u \in [e^{-t}, e^{-T}]$. This is said to be polynomial extrapolation as the coefficient of the polynomials are only fitted over interval $u \in [e^{-T}, 1]$. As the models are usually overparameterized, the minimizer of truncated loss $E_{[0,T]}$ is not the global minimizer for $E_{[0,\infty)}$. In Appendix D.2, we further show the similarity between nonlinear state-space model length extension and polynomial extrapolation. In particular, the overfitting phenomenon gets worse as the number of parmaeters increased.

Figure 3: Graphical demonstration of the difference between zero-initialized hidden states and previous-initialized hidden states (truncated backpropagation through time) in training.

3.2 Convert the extrapolation to interpolation

In the training of state-space models, the hidden states are usually zero-initialized between different batches. As shown in Figure 3, we set the initialization of hidden states



(a) Validation loss (b) Test loss

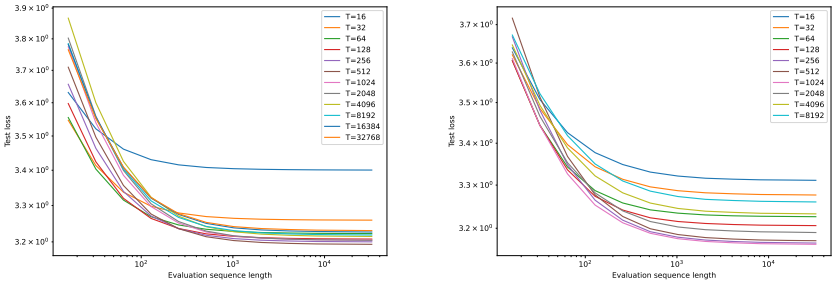
Figure 4: Comparison of two hidden states initialization methods over 6-layer Mamba with 30M parameters. Both the zero-initialized and previous-initialized models are trained over training sequence length $T = 32$. The zero-initialized model has difficulty extrapolating beyond 1024 while the the previous-initialized model has length extrapolation up to $T = 32768$. While the previous hidden state methods achieve the length extension over unshuffled test dataset, when the data is shuffled, models trained with previous hidden state also suffer from the noisy information in the hidden states.

from the previous batch $h_{k+1,0} = h_{k,T}$ instead of zeros $h_{k,0} = 0$. This corresponds to the truncated backpropagation through time method [22]. This change of hidden states initialization requires the dataloader to load consecutive text instead of shuffling them in the batch level. We compare the effects of data shuffling on the following two initialization schemes in Figure 4.

The zero-initialized model gives almost the same loss curves over the shuffled dataset and unshuffled dataset. In contrast, the model trained with previous-initialized hidden sates have smaller validation/test loss and monotonically decreasing loss in the length extension sense. As the previous-initialized model suffer when the evaluation dataset is shuffled dataset, it indicates that the model does extract the information from the non-zero previous hidden states.

4 Numerical results

In this section, we first provide the numerical evidence that training with longer context is generally better but not necessary for length extension (Section 4.1). Then, we further demonstrate that with previous-initialized hidden states, the models can achieve even better length-extension performance that general proper-trained zero-initialized models (Section 4.2). The disadvantage of this previous-initialized training is discussed in Section 4.3.



(a) S5 (b) Mamba

Figure 5: Length extension of models trained with different sequence length using previous-initialized hidden states. We train 6-layer S5 [23] up to training length $T = 32768$ and 6-layer Mamba [4] up to training length $T = 8192$. Mamba has a larger hidden states dimension therefore the maximum training length is smaller (on the same GPU). It can be seen that training with sequence length $T = 1024$ is slightly better than shorter/longer sequence length.

4.1 Longer training context is beneficial but not necessary for length extension

We show in Figure 5 that since inheriting the previous hidden states are approximating the gradient with longer training sequence length, the performance of models trained over longer sequence generally have better length extension capability. Since the models with previous-initialized hidden states have monotonically decreasing perplexity, therefore the length extension property can be achieved without long training sequence length.

4.2 Previous initialized hidden states improve the length extension capability

In Figure 6, we present the length extension curves for the 180M Mamba model, trained using various sequence lengths and different schemes for (training) hidden states initializations. The model with previous initialization, when trained on sequences of length $T = 16$, outperforms the zero-initialized model trained on both $T = 16$ and $T = 32$ sequence lengths. Furthermore, the model trained with a sequence length of $T = 2048$ demonstrates superior length extension performance across both short and long sequences compared to all models with zero initialization. All these models are trained in the same hyperparameter setting.

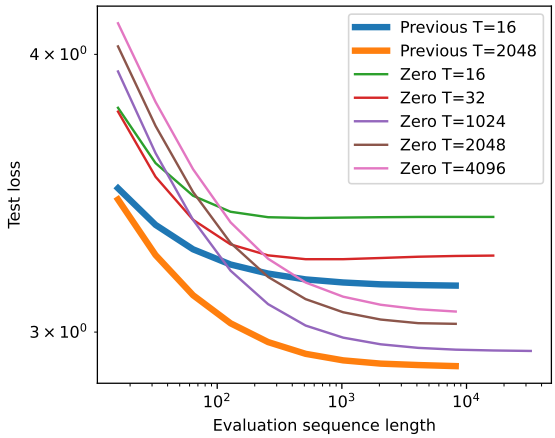
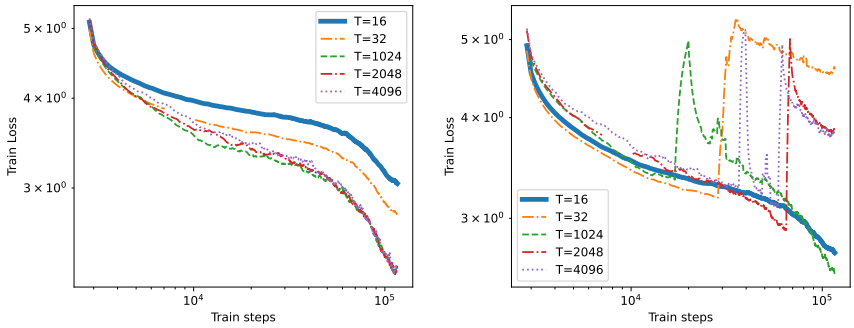


Figure 6: We change the training sequence length and show that, despite zero-initialized models showing adequate length extension capabilities, models trained with previous hidden states consistently surpass them across all evaluated sequence lengths. Throughout our experiments, Mamba models with 180M parameters trained on the Wikitext103 dataset maintain consistent training settings.



(a) Zero-initialized training

(b) Previous-initialized training

Figure 7: Under the same training setting (learning rate is 0.001), we show that the training of 140M previous-initialized S5 [23] come with severe **training instability**.

4.3 On the disadvantages of previous-initialized training

In the previous subsections, we explore the advantages of length extension through modifications in hidden states initialization during training. Here we evaluate the efficacy of the previously-initialized (truncated-BPTT) method and uncover the significant challenges it presents in terms of training stability. As illustrated in Figure 7, particularly with relatively large models, training the 140M S5 model with previously-initialized

hidden states (on the right) exhibits notable instability compared to zero-initialized training (on the left). As training setting are the same, this result shows the instability of current previous-initialized training. In Appendix C.2, we further examine stability through the lens of both hidden state bounds and weight precision in the setting of long-term memory learning. Future direction includes the study of achieving length extension in a more stable manner.

5 Related works

Recurrent neural networks [24] are widely used in sequence modeling. Variants such as LSTM [25] and GRU [26] are efficient to model sequence-to-sequence relationship but they suffer from problems such as vanishing/exploding gradient [27, 28] and exponentially decaying memory [21, 29, 19]. As nonlinear RNNs cannot be parallelized in time, back propagation through time (BPTT) [22] is widely used to speed up the training of long sequences. State-space models [3, 30] relax the training difficulty as the linear RNN layer can be parallelized (in time) via FFT or associative scan [14]. Mamba [4] shows that recurrent models without recurrent nonlinearity can have matching performance against transformer over many tasks while maintaining a low inference cost.

For transformers, Rotary Position Embedding (RoPE) [11] integrates relative positional information into the attention matrix but still cannot achieve reasonable performance beyond the pretrained length. Position Interpolation (PI) [31] introduces a linear rescaling in RoPE and achieves the extension from 2048 to 32768. In Chen et al. [13], they introduce a trainable neural ODE [32] into the position encoding, enabling more fine-grained long context extension. Additive bias [33] is another approach to achieve the length extension. ALiBi [10, 34] is the first effective method to do length extensions, it has been shown to have monotonically decreasing perplexity up to length 3072 for models trained over 64.

It is well known that polynomial extrapolation are ill-conditioned[35] and global minimizers of under-determined system are not unique. Empirical evidence [31] shows the difficulty of extrapolation in the sense that almost every learned curve has the extrapolation issue.

6 Conclusion

In this paper, we investigate the length extension problem in language modeling, particularly focusing on state-space models. We emphasize the challenge faced by zero-initialized SSMs in achieving length extension, which essentially boils down to a problem of polynomial extrapolation. Building upon the above observation, we adopt a simple yet effective hidden states initialization scheme during training. This method significantly enhances the model's performance on longer contexts without compromising its effectiveness on shorter ones. A model with training length $T = 16$ can extend to $T = 32K$, showcasing a consistent decrease in perplexity, as illustrated in Figure 4. Contrary to the common believe that backpropagation is restricted to training lengths of 10-20x [22], our approach is beneficial when the primary goal is length

extension, leading to a dramatic reduction in GPU memory requirements—by up to 2000 times (from 32768 to 16). This discovery suggests that training state-space models with **longer training contexts is desirable but not necessary** for achieving effective length extension.

References

[1] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, and Amanda Askell. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[2] Albert Gu, Tri Dao, Stefano Ermon, Atri Rudra, and Christopher Ré. HiPPO: Recurrent Memory with Optimal Polynomial Projections. In *Advances in Neural Information Processing Systems*, volume 33, pages 1474–1487. Curran Associates, Inc., 2020.

[3] Albert Gu, Karan Goel, and Christopher Re. Efficiently Modeling Long Sequences with Structured State Spaces. In *International Conference on Learning Representations*, October 2021.

[4] Albert Gu and Tri Dao. Mamba: Linear-Time Sequence Modeling with Selective State Spaces, December 2023.

[5] Soham De, Samuel L. Smith, Anushan Fernando, Aleksandar Botev, George Cristian-Muraru, Albert Gu, Ruba Haroun, Leonard Berrada, Yutian Chen, Srivatsan Srinivasan, Guillaume Desjardins, Arnaud Doucet, David Budden, Yee Whye Teh, Razvan Pascanu, Nando De Freitas, and Caglar Gulcehre. Griffin: Mixing Gated Linear Recurrences with Local Attention for Efficient Language Models, February 2024.

[6] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention, August 2020.

[7] Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and Furu Wei. Retentive network: A successor to transformer for large language models. *arXiv preprint arXiv:2307.08621*, 2023.

[8] Songlin Yang, Bailin Wang, Yikang Shen, Rameswar Panda, and Yoon Kim. Gated Linear Attention Transformers with Hardware-Efficient Training, December 2023.

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

[10] Ofir Press, Noah A. Smith, and Mike Lewis. Train Short, Test Long: Attention with Linear Biases Enables Input Length Extrapolation, April 2022.

[11] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced Transformer with Rotary Position Embedding, August 2022.

[12] Yutao Sun, Li Dong, Barun Patra, Shuming Ma, Shaohan Huang, Alon Benhaim, Vishrav Chaudhary, Xia Song, and Furu Wei. A Length-Extrapolatable Transformer, December 2022.

[13] Guanzheng Chen, Xin Li, Zaiqiao Meng, Shangsong Liang, and Lidong Bing. CLEX: Continuous Length Extrapolation for Large Language Models, October 2023.

[14] Eric Martin and Chris Cundy. Parallelizing Linear Recurrent Neural Nets Over Sequence Length. In *International Conference on Learning Representations*, February 2018.

[15] Ann Yuan, Andy Coenen, Emily Reif, and Daphne Ippolito. Wordcraft: Story Writing With Large Language Models. In *27th International Conference on Intelligent User Interfaces*, IUI '22, pages 841–852, New York, NY, USA, March 2022. Association for Computing Machinery. ISBN 978-1-4503-9144-3. doi: 10.1145/3490099.3511105.

[16] Li Chen, Penghao Wu, Kashyap Chitta, Bernhard Jaeger, Andreas Geiger, and Hongyang Li. End-to-end Autonomous Driving: Challenges and Frontiers, June 2023.

[17] Owen Marschall, Kyunghyun Cho, and Cristina Savin. A unified framework of online learning algorithms for training recurrent neural networks. *The Journal of Machine Learning Research*, 21(1):5320–5353, 2020.

[18] 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. The Pile: An 800GB Dataset of Diverse Text for Language Modeling, December 2020.

[19] Shida Wang and Beichen Xue. State-space models with layer-wise nonlinearity are universal approximators with exponential decaying memory. In *Thirty-Seventh Conference on Neural Information Processing Systems*, November 2023.

[20] Zhong Li, Jiequn Han, Weinan E, and Qianxiao Li. On the Curse of Memory in Recurrent Neural Networks: Approximation and Optimization Analysis. In *International Conference on Learning Representations*, October 2020.

[21] Haotian Jiang, Qianxiao Li, Zhong Li, and Shida Wang. A Brief Survey on the Approximation Theory for Sequence Modelling. *Journal of Machine Learning*, 2 (1):1–30, June 2023. ISSN 2790-203X, 2790-2048. doi: 10.4208/jml.221221.

[22] Herbert Jaeger. Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the echo state network approach. 2002.

[23] Jimmy T. H. Smith, Andrew Warrington, and Scott Linderman. Simplified State Space Layers for Sequence Modeling. In *International Conference on Learning Representations*, February 2023.

[24] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, October 1986. ISSN 1476-4687. doi: 10.1038/323533a0.

[25] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-term Memory. *Neural computation*, 9:1735–80, December 1997. doi: 10.1162/neco.1997.9.8.1735.

[26] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1724–1734, September 2014.

[27] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, March 1994. ISSN 1941-0093. doi: 10.1109/72.279181.

[28] Sepp Hochreiter. The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 06(02):107–116, April 1998. ISSN 0218-4885, 1793-6411. doi: 10.1142/S0218488598000094.

[29] Shida Wang, Zhong Li, and Qianxiao Li. Inverse Approximation Theory for Nonlinear Recurrent Neural Networks. In *The Twelfth International Conference on Learning Representations*, October 2023.

[30] Shida Wang and Qianxiao Li. StableSSM: Alleviating the Curse of Memory in State-space Models through Stable Reparameterization, November 2023.

[31] Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. Extending Context Window of Large Language Models via Positional Interpolation, June 2023.

[32] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural Ordinary Differential Equations, December 2019.

[33] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. ISSN 1533-7928.

[34] Faisal Al-Khateeb, Nolan Dey, Daria Soboleva, and Joel Hestness. Position Interpolation Improves ALiBi Extrapolation, October 2023.

[35] Laurent Demanet and Alex Townsend. Stable Extrapolation of Analytic Functions. *Foundations of Computational Mathematics*, 19(2):297–331, April 2019. ISSN 1615-3375, 1615-3383. doi: 10.1007/s10208-018-9384-1.

[36] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, USA, June 2006. ISBN 978-0-471-24195-9.

A Comparison of state-space models and nonlinear recurrent neural networks

Here we give the formulation of single-layer recurrent neural networks (RNNs) [24]. In nonlinear RNNs the activation σ is applied in the temporal direction.

$$h_{k+1} = \boldsymbol{\sigma}(Wh_k + Ux_k + b), \quad h_0 = 0 \tag{14}$$

$$\hat{y}_k = Ch_k, \quad 1 \leq k \leq T. \tag{15}$$

The corresponding continuous-time form of RNNs is

$$\frac{dh_t}{dt} = \boldsymbol{\sigma}(Wh_t + Ux_t + b), \quad \hat{y}_t = Ch_t. \tag{16}$$

Truncated backpropagation through time Due to the nonlinear dynamics of nonlinear RNNs, backpropagation through time (BPTT) [22] is the standard approach to evaluate the gradient. Due to the vanishing/exploding gradient issue [27, 28], truncated backpropagation through time [22] is widely used to speedup the training. In this paper, our previous-initialized hidden state is similar to this T-BPTT method.

B Theoretical backgrounds

The definitions and theorems are well-known results from information theory. We collect the definition for the completeness [36].

B.1 Entropy, conditional entropy and chain rule

Here we let X and Y denote random variable.

Entropy is a measure of the uncertainty of a random variable:

$$H(X) = \mathbb{E}_p \log \left(\frac{1}{p(X)} \right). \tag{17}$$

Joint entropy:

$$H(X, Y) = \mathbb{E}_p \log \left(\frac{1}{p(X, Y)} \right). \tag{18}$$

Conditional entropy:

$$H(Y|X) = \mathbb{E}_p \log \left(\frac{1}{p(Y|X)} \right). \tag{19}$$

Chain rule:

$$H(X, Y) = H(X) + H(Y|X). \tag{20}$$

B.2 Relative entropy, mutual information

The **relative entropy** $D(p||q)$ is a measure of the inefficiency of assuming that the distribution is q when the true distribution is p.

$$D(p||q) = E_p \log \left(\frac{p(X)}{q(X)} \right). \tag{21}$$

Chain rule for relative entropy:

$$D(p(x,y)||q(x,y)) = D(p(x)||q(x)) + D(p(y|x)||q(y|x)). \tag{22}$$

Mutual information:

$$\begin{aligned} I(X;Y) &= E_{p(X,Y)} \log \frac{p(X,Y)}{p(X)p(Y)} && (23)\\ &= H(X) - H(X|Y) && (24)\\ &= H(X) + H(Y) - H(X,Y) && (25) \end{aligned}$$

Theorem B.1 (Information inequality)**.**

$$D(p||q) \geq 0. \tag{26}$$

with equality if and only if $p(x) = q(x)$ for all x.

Corollary B.2 (Nonnegativity of mutual information)**.** *For any two random variables, X, Y,*

$$I(X;Y) \geq 0. \tag{27}$$

This comes from the fact by taking p to be $p(X,Y)$ and q to be $p(X)p(Y)$.

Theorem B.3 (Conditioning reduces entropy)**.**

$$H(X|Y) \leq H(X). \tag{28}$$

with equality if and only if X and Y are independent.

B.3 Riesz representation theorem for linear functional

Theorem B.4 (Riesz-Markov-Kakutani representation theorem)**.** *Assume $H : C_0(\mathbb{R}, \mathbb{R}^d) \mapsto \mathbb{R}$ is a linear and continuous functional. Then there exists a unique, vector-valued, regular, countably additive signed measure μ on $\mathbb{R}$ such that*

$$H(\mathbf{x}) = \int_{\mathbb{R}} x_s^\top d\mu(s) = \sum_{i=1}^{d} \int_{\mathbb{R}} x_{s,i} d\mu_i(s). \tag{29}$$

In addition, we have the linear functional norm

$$\|H\|_\infty := \sup_{\|\mathbf{x}\|_\mathcal{X} \leq 1} |H(\mathbf{x})| = \|\mu\|_1(\mathbb{R}) := \sum_i |\mu_i|(\mathbb{R}). \tag{30}$$

C Theoretical results and proofs

In Appendix C.1, we give the proof for input-dependent gating in state-space models is associative. In Appendix C.2, we show the dependency of recurrent weights range with respect to the finite precision range and why the corresponding gradient values might be unbounded.

C.1 Input-dependent gating in state-space models is associative

Table 2: Difference between S5, Mamba and Gated Linear Attention in terms of the recurrent weight and hidden states dimensions.

	h is vector	h is matrix
W diagonal	S5 [23], Mamba	N/A
W full	Traditional SSM	Gated Linear Attention [8]

Consider the following binary operator defined for tuple element (W, h) as follow:

$$(W_1, h_1) \circ (W_2, h_2) = (W_2 \odot W_1, h_1 + W_1 \odot h_2). \tag{31}$$

Notice that W and h can depend on input value x.

Theorem C.1 (Associativity of binary operation in state-space models)**.**

$$\Big((W_1, h_1) \circ (W_2, h_2)\Big) \circ (W_3, h_3) = (W_1, h_1) \circ \Big((W_2, h_2) \circ (W_3, h_3)\Big) \tag{32}$$

Proof.

$$\Big((W_1, h_1) \circ (W_2, h_2)\Big) \circ (W_3, h_3) \tag{33}$$
$$= (W_2 \odot W_1, h_1 + W_1 \odot h_2) \circ (W_3, h_3) \tag{34}$$
$$= (W_3 \odot W_2 \odot W_1, h_1 + W_1 \odot h_2 + W_1 \odot W_2 \odot h_3) \tag{35}$$
$$= (W_1, h_1) \circ (W_3 \odot W_2, h_2 + W_2 \odot h_3) \tag{36}$$
$$= (W_1, h_1) \circ \Big((W_2, h_2) \circ (W_3, h_3)\Big) \tag{37}$$

□

C.2 Sensitivity of recurrent weights

Let M be the maximum value in given finite precision machine. Let $\lambda = \max(\text{diag}(\Lambda))(< 1)$ be the (largest) memory decay mode in state-space models: An estimate for the hidden

states scale as follow:

$$|h_T|_\infty = \left| h_0 + \sum_{k=1}^{T} \Lambda^k U x_{k-1} \right|_\infty \tag{38}$$

$$\leq |h_0|_\infty + \frac{1-\lambda^T}{1-\lambda} |U|_1 \sup_k |x_k|_\infty \tag{39}$$

To prevent the overflow of hidden states $|h_T|_2 \leq M$, as the sequence length increases $T \to \infty$, a **sufficient** condition for the eiganvalue ranges is

$$\lambda < 1 - \frac{|U|_1 \sup_k |x_k|_\infty}{M - |h_0|_\infty}. \tag{40}$$

As the learning process of long-term memory requires the slow decay of information within hidden states, achieving long-term memory implies that the parameter λ gets close to 1. This result shows that if we use low-bit quantization (with small M), the hidden state might be unbounded by M and therefore the training can be unstable due to overflow issues. The overflow issue is **more severe for large models** as $|U|_1$ scale up in $O(m)$ with respect to the hidden dimension m.

C.3 Existence of weak length extension

Consider the language modeling as the learning of sequence of random variables. It can be seen that such language modeling should obey the weak length extension in the information theory framework.

Proposition C.2. *Let the dataset of autoregressive language modeling sampled from the (potentially infinite) sequence of random variables, then we have the existence of weak length extension:*

$$H(X_{k+1}|X_1, \ldots, X_k) \leq H(X_{k+1}|X_2, \ldots, X_k) \leq H(X_{k+1}|X_k) \leq H(X_{k+1}). \tag{41}$$

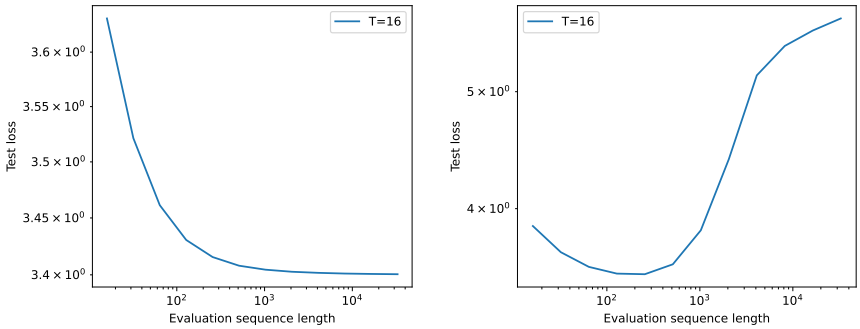
This is a direct result from the conditioning reduces entropy theorem. We just need to repeatedly apply the above Theorem B.3 to $X = (X_{k+1}|X_{i+1}, \ldots, X_k)$ while $Y = X_i$.

D Additional numerical results

In this section, we provide additional numerical results to show previous-initialized hidden state is also effective for S5 (Appendix D.1), the empirical similarity between length extension and polynomial extrapolation (Appendix D.2) and the benefit of previous-hidden states in training for GRU (Appendix D.3).

D.1 Comparison of different hidden states initialization

In Figure 4 we show the effect of previous-initialized hidden state help the Mamba to have length extension. In Figure 8, we further compare the influence of hidden state initialization schemes during the training of the models. It can be seen the S5 trained with previous-initialized hidden states can have length extension from 16 to 32768.



(a) Previous-initialized S5 + unshuffled dataloader (b) Zero-initialized S5 and unshuffled dataloader

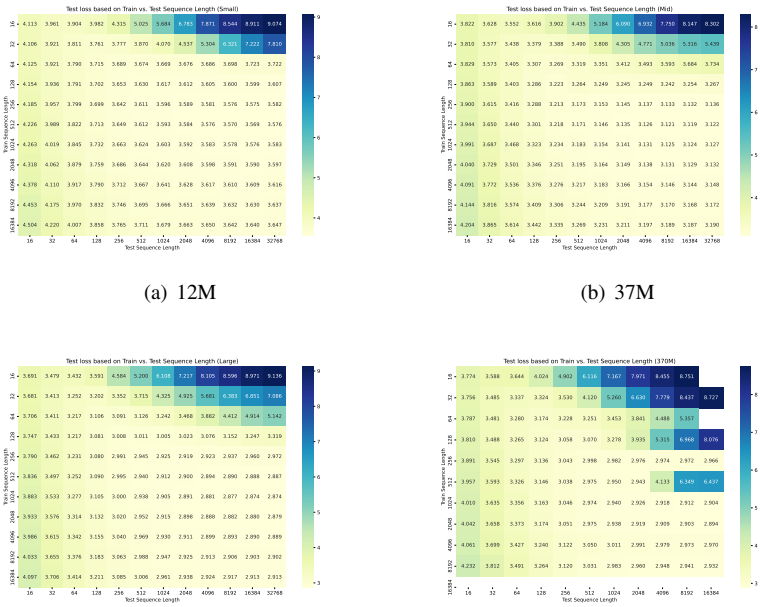
Figure 8: The 30M S5 model trained over Wikitext103 using previous-initialized hidden states has (weak) length extension capability up to sequence length 32768 while the zero-initialized model fails to have monotone length extension capability.

D.2 Overfitting in length extension

We show the length extension for nonlinear state-space models are similar to polynomial extrapolation in the following sense: This escalation in parameters significantly complicates the length extension process, necessitating a proportional increase in the training sequence length for zero-initialized models from 64 and 256 to a substantial 1024. This correspond to the overfitting argument as the larger model size will require larger sequence length (more data for the evaluation of the memory function ρ.) It is anticipated that, for large models trained with zero-initialized hidden states, one cannot use length $T = 2048$ to train a model with length extension capability. The missing last row in 370M is due to the out-of-memory issue. Two missing values come from overflow issue.

D.3 Generalization to other recurrent model

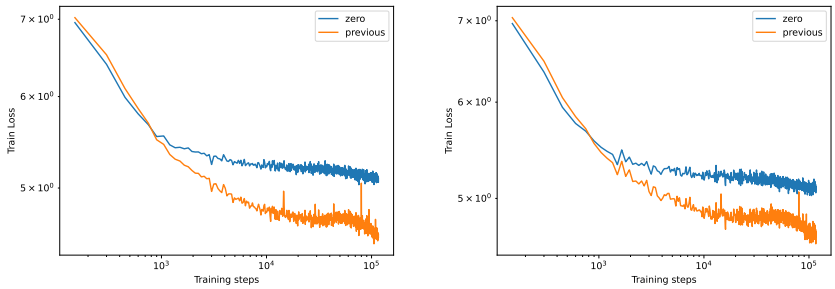
In addition to the state-space models, we extend our comparison to different hidden state initialization schemes for a 6-layer GRU with 30 million parameters, as depicted in Figure 10. The results demonstrate that initializing with previous hidden states can enhance the training performance of the GRU model.



(a) 12M (b) 37M

(c) 127M (d) 370M

Figure 9: Various sizes of zero-initialized S5 models were trained on Wikitext103. The performance observed in the upper triangular area of the graph illustrates their capacity for length extension. To maintain consistency, we scaled up the models while keeping the training settings constant. Notably, as models are initialized with zero-hidden states, larger models necessitate longer training contexts to effectively handle length extension



(a) $T = 16$ (b) $T = 32$

(c) $T = 64$ (d) $T = 128$

Figure 10: Training with previous hidden states initialization on Wikitext103 enhances GRU training performance. Here T is the training sequence length.