LongSSM: On the Length Extension of State-space Models

Anonymous Authors[1]

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 (Brown et al., 2020) are usually trained on large corpus with a fixed context length (e.g., 2000 tokens). However, attention-based transformer has an $O(T^2)$ asymptotic growth with respect to the sequence length. The inference cost is even higher when we are working with long sequences. Recently, state-space models (Gu et al., 2020; Gu & Dao, 2023) and linear-attention-based transformers (Katharopoulos et al., 2020; Yang et al., 2023) have shown the potential to replace the attention-based transformers (Brown et al., 2020). SSMs are recurrent models and have length-independent inference cost.

However, 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's length in state-of-the-art models (Gu & Dao, 2023; Yang et al., 2023). 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 finite training sequence length assumption is not only reasonable but also necessary due to the constraints of GPU memory and the relative limitations of the training length when compared against the infinite inference length encountered in real-world applications.

In the standard Transformer model (Vaswani et al., 2017), achieving length extension can also be challenging, often constrained by the limitations of absolute position encoding (Vaswani et al., 2017; Press et al., 2022). Press et al. (2022) 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 (Su et al., 2022; Sun et al., 2022; Chen et al., 2023a). In this paper, we propose a method that is orthogonal to the 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 3).

We summarize our main contributions as follow:

1. We show why the zero hidden states initialization scheme has difficulty doing length extension. It is equivalent to polynomial extrapolation in the simplified setting.
2. Based on the difficulty for zero-initialization case, we introduce the training approach that leverages previous hidden states without batch-level shuffling. This method enables state-space models to achieve what we term "weak length extension", a concept that will be further defined and explored in the background section.
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, but has length extension up to 32768.

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 function(al)s, 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. We use m to denote the hidden dimension and let d be the input dimension.

Table 1. Comparison of asymptotic training/inference cost for (standard) transformers (Brown et al., 2020) and state-space models

	Attention-based transformer	State-space models / Linear-attention (Katharopoulos et al., 2020)
Training step cost	$O(T^2)$	$O(T)$
Inference cost	$O(T^2)$	$O(1)$

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 is diagonal matrix	S5 (Smith et al., 2023), Mamba (Gu & Dao, 2023)	N/A
W is full matrix	Traditional SSM	Gated Linear Attention (Yang et al., 2023)

2. Background

In this section, we first introduce the state-space models. Compared with traditional nonlinear RNNs, it comes with better parallel degree in the sense that fast Fourier transform and associative scan can be used to accelerate the training. Next, we give the definition of three types of length extension capability. In particular, we emphasize 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 (Gu et al., 2021) typically have layer-wise nonlinear activations while the traditional nonlinear RNNs have recurrent nonlinear activations (see the comparison of SSMs and RNNs in Appendix A).

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

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

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

$$\frac{dh_t}{dt} = Wh_t + (Ux_t + b), \tag{3}$$

$$\hat{y}_t = C\boldsymbol{\sigma}(h_t). \tag{4}$$

The solution to the differential equation of h_t is

$$h_t = h_0 + \int_0^t e^{W(t-s)}(Ux_s + b)ds. \tag{5}$$

The convolution form of SSMs Based on the continuous time formulation of the hidden states in Equation (5), hidden state sequences can be rewritten into the following convolution form

$$\mathbf{h} = \rho(t) \ast (U\mathbf{x} + b) \tag{6}$$

The convolution kernel is

$$\rho(t) = e^{Wt}. \tag{7}$$

Based on the convolution form in Equation (6), fast Fourier transform (FFT) or associative scan can be used to accelerate the evaluation of hidden state. Compared with $O(T^2)$ cost to evaluate the attention matrix, it only takes $O(T \log T)$ to evaluate the hidden states $\mathbf{h}$ (and corresponding output $\hat{\mathbf{y}}$).

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 $U(x_k) \in \mathbb{R}^{m \times d}$.

$$h_{k+1} = W(x_k) \odot h_k + U(x_k), \quad h_0 = 0 \in \mathbb{R}^{m \times d} \tag{8}$$

$$\hat{y}_k = C\boldsymbol{\sigma}(h_k), \quad 1 \leq k \leq T. \tag{9}$$

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, achieving a training cost of $O(T \log T)$. In Appendix C.2, we show the associativity of the following binary operator $\circ$ defined over tuple (W, h):

$$(W_1, h_1) \circ (W_2, h_2) = (W_2 \odot W_1, h_1 + W_1 \odot h_2). \tag{10}$$

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

$$(\_, h_k) = (W(x_k), U(x_k)) \circ \cdots \circ (W(x_1), U(x_1)) \circ (I, 0). \tag{11}$$

If matrices W_k are input-independent, this corresponds to the case from S5 in Smith et al. (2023). 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 (Yang et al., 2023) whose hidden states are 2D square matrices. We summarize the differences in Table 2.

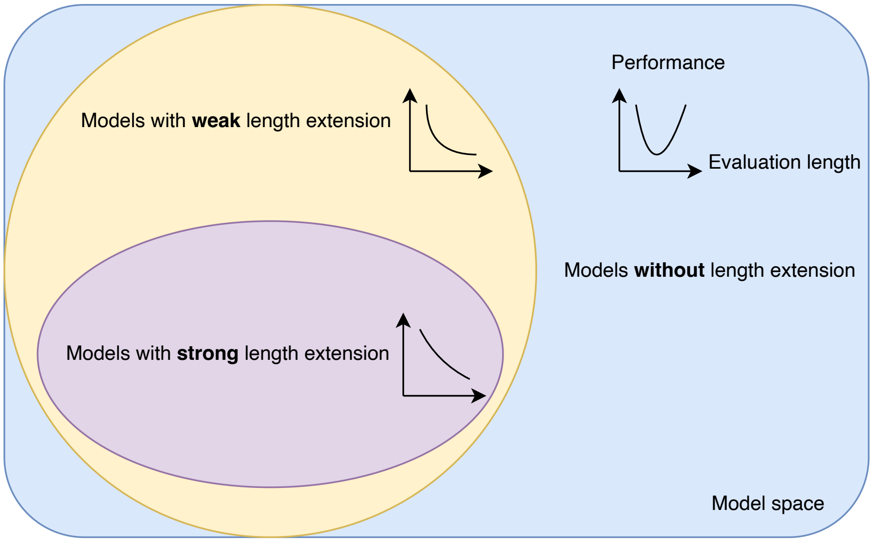

Models with **weak** length extension
Performance
Evaluation length
Models **without** length extension
Models with **strong** length extension
Model space

Figure 1. Graphical demonstration of length extension.

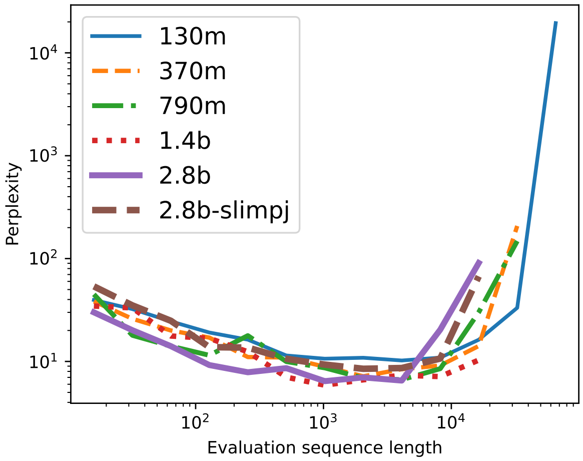


Figure 2. Length extension capability of Mamba evaluated over the Pile (Gao et al., 2020) dataset. The training sequence length for the models is 2048. While the perplexity over 4096 is still reasonable, the perplexities of models beyond length 8192 are increasing significantly.

2.2. Length extension

The length extension property has been widely studied for attention-based transformers (Press et al., 2022; Su et al., 2022; Sun et al., 2022; Chen et al., 2023a). This is an essential attribute for models designed for infinite contexts windows (writing novels (Yuan et al., 2022), autonomous driving (Chen et al., 2023b), online learning (Marschall et al., 2020)). However, it is shown that the state-of-the-art SSMs failed to achieve length extension beyond 4k[1] [2].

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

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. Minimizing the cross entropy loss $CE(p, q) = -\sum_x p(x) \log q(x)$ is equivalent to minimizing the KL-Divergence (relative-entropy) $D_{KL}(p||q) = \sum_x p(x) \log \frac{p(x)}{q(x)}$. As the perplexity of the model is typically evaluated by $\exp(CE(p, q))$, the monotonicity of the perplexity is the same as the cross entropy. Here we define three types of length-extension capability based on the monotonicity of the perplexity:

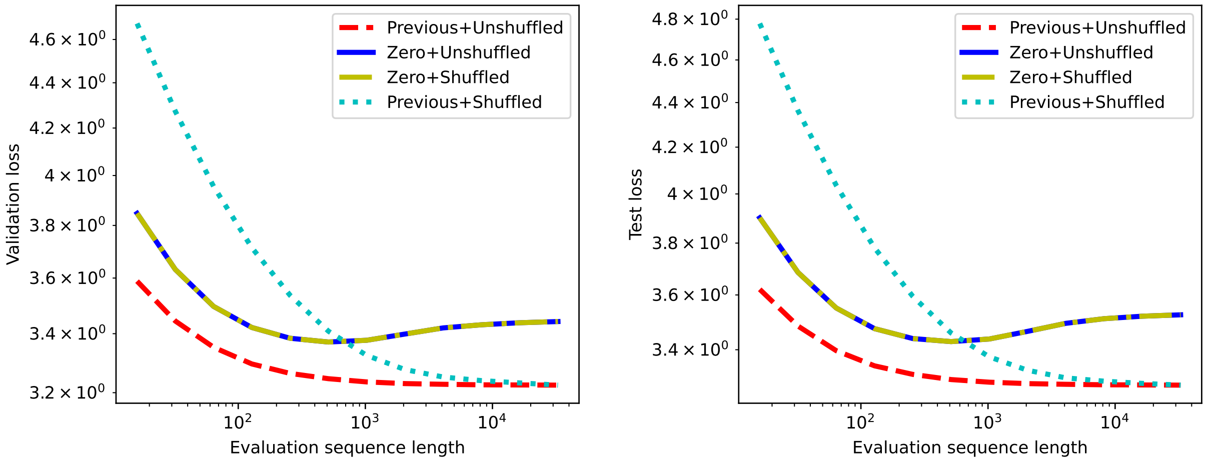
1. **(Strong length extension)**: For some $T_0 > 0, \forall T > T_0, \text{perplexity}_{T+1} < \text{perplexity}_T$.
2. **(Weak length extension)**: For some $T_0 > 0, \forall T > T_0, \text{perplexity}_{T+1} \leq \text{perplexity}_T$.
3. **(No length extension)**: If there does not exists T_0 such 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. As the models are trained with sequence length $T = 2048$, it has difficulty maintaining the perplexity beyond length $T \geq 4096$.

[1] https://openreview.net/forum?id=AL1fq05o7H

[2] https://imgbb.com/XVT0hGJ



(a) Validation loss

(b) Test loss

Figure 3. 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.

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 sequence of random variables (likely not independent), then the weak length extension exists for entropy.

Theorem 2.2 (Existence of weak length extension in autoregressive language modeling)**.** *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* $X_{[0,\dots,T]}$.

$$\mathbf{H}((X_1, \dots, X_T)) = X_{T+1}. \tag{12}$$

Consider the entropy of this autoregressive language model

$$H((X_1, \dots, X_T)) \tag{13}$$

$$= - \sum_{x_i \in X_i} p((x_1, \dots, x_T)) \log p((x_1, \dots, x_T)). \tag{14}$$

See the proof based on the information theory in Appendix C.3. A natural expectation from the monotonically decreasing entropy $\sum p(x) \log p(x)$ is that if the distribution of model is close to the target distribution $p(x)$, then the cross entropy $DL(p, q) = \sum p(x) \log q(x)$ should also be decreasing.

Remark 2.3. Based on above definitions for length extension, it can be seen attention with linear bias (ALiBi) (Press et al., 2022) can be regarded as models with weak length extension.

3. Main results and method

In this section, we begin by outlining our method for extending the context length of SSM-based language models (Section 3.2). Following this, we delve into a theoretical analysis focusing on the issue of zero-initialized hidden states (Section 3.1). We demonstrate that extending the length of models with zero-initialized states is analogous to polynomial extrapolation. Our argument posits that adopting appropriate initialization for hidden states enhances length extension.

3.1. Length extension is extrapolation

In approximation theory, state-space models are universal approximators for bounded causal continuous time-homogeneous regular nonlinear functionals (Wang & Xue, 2023). 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. A natural question arises regarding whether such finite-window-trained models can effectively capture long-term memory beyond the training window. Here we first study the length extension in a simplified linear setting and then show how this result generalize to multi-layer nonlinear models.

Consider the learning of linear functionals (Li et al., 2020; Jiang et al., 2023) by single-layer state-space model, this linear functional target comes with a unique representation:

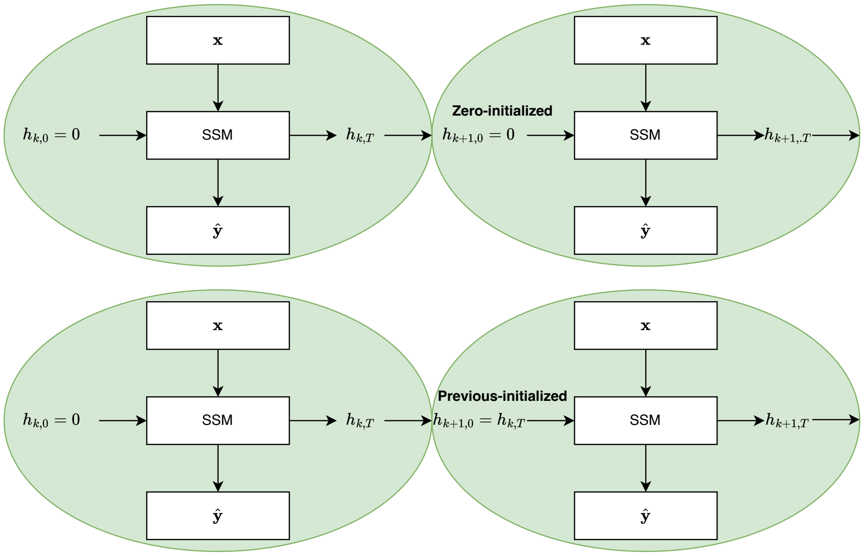


Figure 4. Graphical demonstration of the difference between zero-initialized hidden states and previous-initialized hidden states in training.

$y_t = \mathbf{H}_t(\mathbf{x}) = \int_0^\infty \rho_s x_{t-s} ds$ with $|\rho|_1 := \int_0^\infty |\rho_s| ds < \infty$ while the single-layer state-space model (without layer-wise activation) can be represented by $\hat{y}_t = \widehat{\mathbf{H}}_t(\mathbf{x}) = \int_0^T C e^{Ws} U x_{t-s} ds + y_0$. 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 $\hat{\rho}(t) = C e^{Wt} U$. Consider following error decomposition

$$|y_T - \hat{y}_T| := \tag{15}$$

$$\left| \int_0^\infty \rho_s x_{T-s} ds - \left(\int_0^T \hat{\rho}_s x_{T-s} ds + y_0 \right) \right| \tag{16}$$

$$\leq \left| \int_0^\infty \rho_s x_{T-s} ds - \int_0^T \rho_s x_{T-s} ds - y_0 \right| \tag{17}$$

$$+ \left| \int_0^T \rho_s x_{T-s} ds - \int_0^T \hat{\rho}^*_s x_{T-s} ds \right| \tag{18}$$

$$+ \left| \int_0^T \hat{\rho}^*_s x_{T-s} ds - \int_0^T \hat{\rho}_s x_{T-s} ds \right| \tag{19}$$

Therefore

$$|y_T - \hat{y}_T| \leq \left| \int_T^\infty \rho_s x_{T-s} ds - \hat{y}_0 \right| \tag{20}$$

$$+ \left| \int_0^T \rho_s x_{T-s} ds - \int_0^T \hat{\rho}^*_s x_{T-s} ds \right| \tag{21}$$

$$+ \left| \int_0^T \hat{\rho}^*_s x_{T-s} ds - \int_0^T \hat{\rho}_s x_{T-s} ds \right| \tag{22}$$

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** (Equation (20)), **finite time approximation error** (Equation (21)), **optimization error** (Equation (22)). For any fixed target $\mathbf{H}$, as the hidden dimension m increases, the finite time approximation error decays to 0. If the data are sufficient and computation cost are abundant, the optimization error decays to 0 via gradient-based optimization. However, the length extension error cannot be reduced by simply increasing hidden dimension and training with data.

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

$$|y_t - \hat{y}_t| \leq \left| \int_t^\infty \rho_s x_{t-s} ds - \hat{y}_0 \right| \tag{23}$$

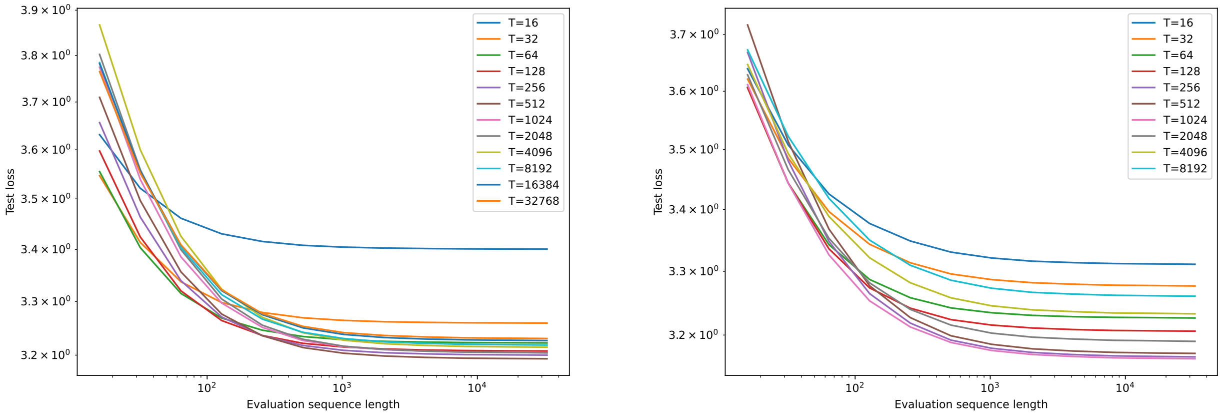
$$+ \left| \int_T^t (\rho_s - \hat{\rho}_s) x_{t-s} ds \right| \tag{24}$$

$$+ \left| \int_0^T (\rho_s - \hat{\rho}_s) x_{t-s} ds \right|. \tag{25}$$

While the first error term can be minimized and third cannot be observed, the second term can be bounded by the form of $\int_T^t |\rho_s - \hat{\rho}_s| ds$. With change of variable $u = e^{-s}$, take $\mathcal{T}\rho_u = \rho_{-\log u}, u \in (0, 1]$.

$$\int_T^t |\rho_s - \hat{\rho}_s| ds = \int_{e^{-t}}^{e^{-T}} \left| \mathcal{T}\rho_u - \sum_{k=1}^m c_i u_i^\lambda \right| \frac{1}{u} du. \tag{26}$$

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 extrapolation as the coefficient of the polynomials are only fitted over interval $u \in [e^{-T}, 1]$. As deep learning usually works in overparameterized setting, the



(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 (Smith et al., 2023) up to training length $T = 32768$ and 6-layer Mamba (Gu & Dao, 2023) up to training length $T = 8192$. Mamba has a generally 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.

minimizer of truncated loss $E_{[0,T]}$ is not the global minimizer for $E_{[0,\infty)}$. In Appendix D.3, we further show the similarity between nonlinear state-space model length extension and polynomial extrapolation.

3.2. Methods

In the training of state-space models, the hidden states are usually zero-initialized between different training batches. As shown in Figure 4, we propose a new hidden state initialization method by setting the hidden states from the previous batch $h_{k+1,0} = h_{k,T}$ instead of resetting it by $h_{k,0} = 0$. 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 3.

1. (Zero-initialized hidden states) $h_{k,0} = 0$ for $k \in \mathbb{N}$.
2. (Previous-initialized hidden states) Let $h_{k,T}$ be the last hidden states from batch k, then the initial hidden state for batch $k + 1$ will be set to $h_{k+1,0} = h_{k,T}$ for $k \in \mathbb{N}$.

It can be seen that 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. However, 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.

Remark 3.1. However, the hidden state can cause the training instability. As demonstrated in Ainslie et al. (2023), the training of long context can suffer from stability issue. We expand the discussion on instability in Section 4.3.

4. Numerical results

In this section, we first provide the numerical evidence that training the model 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 model failed to achieve (Section 4.2).

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. However, we also notice that training the model with exceedly larger context gives a worse training and testing performance. One likely reason for this phenomenon is the text length in the Wikitext-103 dataset[3].

[3] In Dai et al. (2019), the average sequence length from Wikitext103 is around 3.6K.

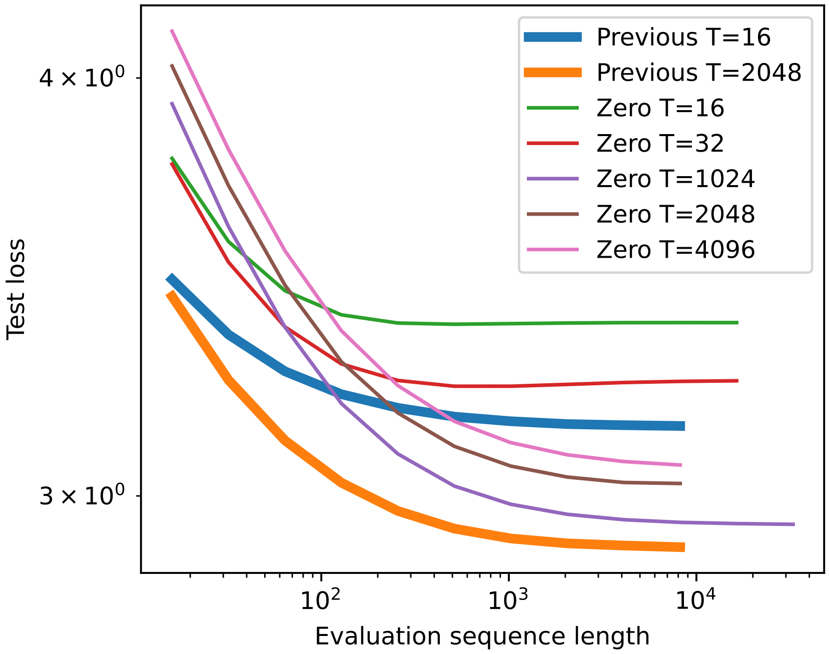


Figure 6. Models are 180M mamba. Dataset is Wikitext103. The training setting of the models are exactly the same. We train the model over different sequences length. And we show that even when the zero-initialized are properly trained with suitable length extension capability, the length extension of the models trained over previous hidden states are still generally better than the models trained with zero-initialized hidden states over all evaluation sequence length.

4.2. Previous initialized hidden states improve the length extension capability

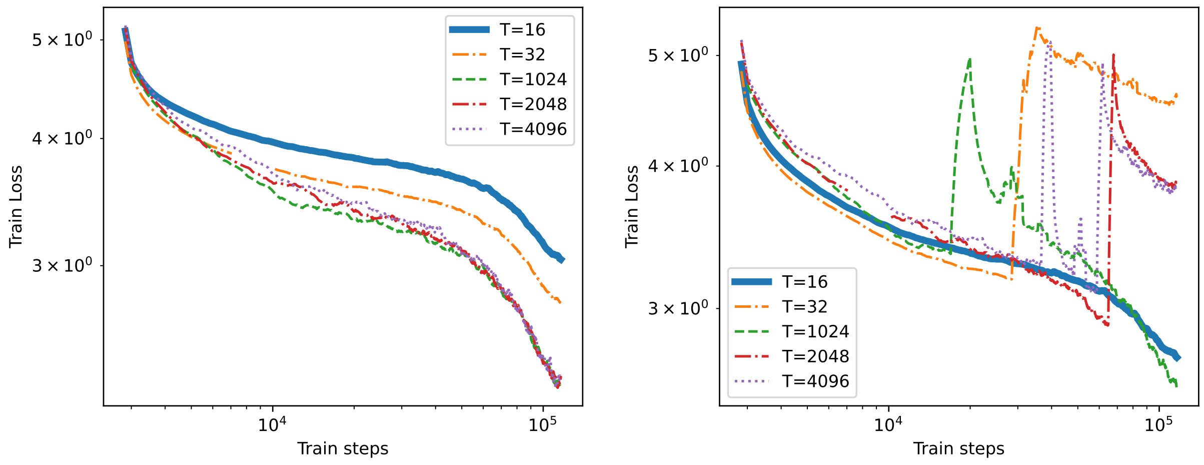
In Figure 6, we present the length extension curves for the 180M Mamba model, trained using various sequence lengths and different schemes for (training) hidden states initializations. The model with previous initialization, when trained on sequences of length $T = 16$, outperforms the zero-initialized 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.

4.3. On the disadvantages of previous-initialized training

In the previous subsections, we discuss the benefits of length extension from the change of hidden states initialization in training. Here we evaluate this previous-initialized method and show that the change of hidden states initialization comes with significant difficulty of training stability. As shown in Figure 7, when the model is relatively large, the training of 140M S5 with previous-initialized hidden states (right) can be significantly unstable than the zero-initialized training (left). As training setting are the same, this result shows the instability of current previous-initialized training. Future direction includes the study of achieving length extension in a more stable manner.

5. Related works

Recurrent neural networks (Rumelhart et al., 1986) are widely used in sequence modeling. Variants such as LSTM (Hochreiter & Schmidhuber, 1997) and GRU (Cho et al., 2014) are efficient to model sequence-to-sequence relationship but they suffer from the problem such as vanishing/exploding gradient (Bengio et al., 1994; Hochreiter, 1998) and exponentially decaying memory (Li et al., 2020; Jiang et al., 2023; Wang et al., 2023). Due to the nonlinear dynamics, it cannot be parallelized in time to accelerate the forward and backward propagation. State-space models (Gu et al., 2021; Smith et al., 2023; Wang & Xue, 2023) relax the training difficulty as the linear RNN layer can be parallelized (in time) via FFT or associative scan (Martin & Cundy, 2018). Being a recurrent model and having an asymptotic exponential memory decay, it has been shown that via suitable reparameterization, state-space models can approximate long-term memory in a stable manner (Wang & Li, 2023). Mamba (Gu & Dao, 2023) and gated linear attention (Yang et al., 2023) show that recurrent models with layerwise activation but without recurrent nonlinearity can



(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 (Smith et al., 2023) come with severe training instability.

have matching performance against transformer over many task while maintaining a significantly smaller inference cost.

In Trinh et al. (2018), they improve the learning of long-term dependencies in RNNs by adding an auxiliary loss. While the methods might be similar, our work study the effects of using previous hidden states on length extension while theirs are on the improvement of model performance within training sequence length.

For attention-based transformers, Rotary Position Embedding (RoPE) (Su et al., 2022) integrates relative positional information into the attention matrix but still cannot achieve reasonable performance beyond the pre-trained context length. Sun et al. (2022) proposes XPos to achieve length extension from 1024 to 4096. Position Interpolation (PI) (Chen et al., 2023c) introduces a linear rescaling in RoPE and achieves the extension from 2048 to 32768. In Chen et al. (2023a), they introduce a trainable neural ODE (Chen et al., 2019) into the position encoding, enabling more fine-grained long context extension. Additive bias (Raffel et al., 2020) is another approach to achieve the length extension. ALiBi (Press et al., 2022; Al-Khateeb et al., 2023) 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. Other additive bias of polynomial form and logarithm form have been explored in Chi et al. (2022) and Chi et al. (2023).

It is well known that (polynomial) extrapolation are ill-conditioned(Demanet & Townsend, 2019) and global minimizers of under-determined system are not unique. Empirical evidence (Chen et al., 2023c) 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. Despite their recurrent structure, we demonstrate that SSMs face significant challenges in extrapolating to longer contexts. Our analysis indicates that these difficulties are fundamentally linked to the intrinsic nature of length extension, which is essentially a problem of polynomial extrapolation. One key finding is that in extrapolation problems, a unique minimizer is not guaranteed, implying that merely increasing the hidden dimension of the model does not sufficiently address length extension issues. Therefore SSMs trained with zero-initialized hidden states exhibit limited generalization capabilities for extended contexts. Based on the above observations, we propose a simple yet effective hidden state initialization scheme for initializing hidden states during training. This method significantly enhances the model's performance on longer contexts without compromising its effectiveness on shorter ones. A model trained with length $T = 16$ can extend to $T = 32K$, showcasing a consistent decrease in perplexity, as illustrated in Figure 3. Our approach is beneficial when the primary goal is length extension, rather than optimal model performance, 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 for **longer contexts is desirable but not necessary** for achieving effective length extension.

Impact Statements

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Acknowledgements

Shida Wang is supported by NUS-RMI Scholarship. We thank Qian Liu, Chao Du, Tianbo Li, Min Lin for helpful discussions.

References

Ainslie, J., Lee-Thorp, J., de Jong, M., Zemlyanskiy, Y., Lebrón, F., and Sanghai, S. GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints, May 2023.

Al-Khateeb, F., Dey, N., Soboleva, D., and Hestness, J. Position Interpolation Improves ALiBi Extrapolation, October 2023.

Bengio, Y., Simard, P., and Frasconi, P. 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.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., and Askell, A. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Chen, G., Li, X., Meng, Z., Liang, S., and Bing, L. CLEX: Continuous Length Extrapolation for Large Language Models, October 2023a.

Chen, L., Wu, P., Chitta, K., Jaeger, B., Geiger, A., and Li, H. End-to-end Autonomous Driving: Challenges and Frontiers, June 2023b.

Chen, R. T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. Neural Ordinary Differential Equations, December 2019.

Chen, S., Wong, S., Chen, L., and Tian, Y. Extending Context Window of Large Language Models via Positional Interpolation, June 2023c.

Chi, T.-C., Fan, T.-H., Ramadge, P. J., and Rudnicky, A. I. KERPLE: Kernelized Relative Positional Embedding for Length Extrapolation, October 2022.

Chi, T.-C., Fan, T.-H., Rudnicky, A. I., and Ramadge, P. J. Dissecting Transformer Length Extrapolation via the Lens of Receptive Field Analysis, May 2023.

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1724–1734, September 2014.

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

Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q. V., and Salakhutdinov, R. Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context. https://arxiv.org/abs/1901.02860v3, January 2019.

Demanet, L. and Townsend, A. 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.

Gao, L., Biderman, S., Black, S., Golding, L., Hoppe, T., Foster, C., Phang, J., He, H., Thite, A., Nabeshima, N., Presser, S., and Leahy, C. The Pile: An 800GB Dataset of Diverse Text for Language Modeling, December 2020.

Gu, A. and Dao, T. Mamba: Linear-Time Sequence Modeling with Selective State Spaces, December 2023.

Gu, A., Dao, T., Ermon, S., Rudra, A., and Ré, C. HiPPO: Recurrent Memory with Optimal Polynomial Projections. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1474–1487. Curran Associates, Inc., 2020.

Gu, A., Goel, K., and Re, C. Efficiently Modeling Long Sequences with Structured State Spaces. In *International Conference on Learning Representations*, October 2021.

Hochreiter, S. 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.

Hochreiter, S. and Schmidhuber, J. Long Short-term Memory. *Neural computation*, 9:1735–80, December 1997. doi: 10.1162/neco.1997.9.8.1735.

Jiang, H., Li, Q., Li, Z., and Wang, S. 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.

Katharopoulos, A., Vyas, A., Pappas, N., and Fleuret, F. Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention, August 2020.

Li, Z., Han, J., E, W., and Li, Q. On the Curse of Memory in Recurrent Neural Networks: Approximation and Optimization Analysis. In *International Conference on Learning Representations*, October 2020.

Marschall, O., Cho, K., and Savin, C. A unified framework of online learning algorithms for training recurrent neural networks. *The Journal of Machine Learning Research*, 21(1):5320–5353, 2020.

Martin, E. and Cundy, C. Parallelizing Linear Recurrent Neural Nets Over Sequence Length. In *International Conference on Learning Representations*, February 2018.

Press, O., Smith, N. A., and Lewis, M. Train Short, Test Long: Attention with Linear Biases Enables Input Length Extrapolation, April 2022.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. 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.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, October 1986. ISSN 1476-4687. doi: 10.1038/323533a0.

Smith, J. T. H., Warrington, A., and Linderman, S. Simplified State Space Layers for Sequence Modeling. In *International Conference on Learning Representations*, February 2023.

Su, J., Lu, Y., Pan, S., Murtadha, A., Wen, B., and Liu, Y. RoFormer: Enhanced Transformer with Rotary Position Embedding, August 2022.

Sun, Y., Dong, L., Patra, B., Ma, S., Huang, S., Benhaim, A., Chaudhary, V., Song, X., and Wei, F. A Length-Extrapolatable Transformer, December 2022.

Trinh, T. H., Dai, A. M., Luong, M.-T., and Le, Q. V. Learning Longer-term Dependencies in RNNs with Auxiliary Losses, June 2018.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Wang, S. and Li, Q. StableSSM: Alleviating the Curse of Memory in State-space Models through Stable Reparameterization, November 2023.

Wang, S. and Xue, B. 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.

Wang, S., Li, Z., and Li, Q. Inverse Approximation Theory for Nonlinear Recurrent Neural Networks. In *The Twelfth International Conference on Learning Representations*, October 2023.

Yang, S., Wang, B., Shen, Y., Panda, R., and Kim, Y. Gated Linear Attention Transformers with Hardware-Efficient Training, December 2023.

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

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

Here we give the typical formulation of single-layer recurrent neural networks (RNNs) (Rumelhart et al., 1986). In nonlinear RNNs the activation σ is applied in the temporal direction.

$$h_{k+1} = \boldsymbol{\sigma}(Wh_k + Ux_k + b), \quad h_0 = 0 \tag{27}$$
$$\hat{y}_k = Ch_k, \quad 1 \leq k \leq T. \tag{28}$$

The corresponding continuous-time form of RNNs is

$$\frac{dh_t}{dt} = \boldsymbol{\sigma}(Wh_t + Ux_t + b), \tag{29}$$
$$\hat{y}_t = Ch_t. \tag{30}$$

B. Theoretical backgrounds

The definitions and theorems are well-known results from information theory. We collect the definition for the completeness (Cover & Thomas, 2006).

B.1. Entropy, conditional entropy and chain rule

Here we let X and Y denote random variable.

Entropy:

$$H(X) = \mathbb{E}_p \log\left(\frac{1}{p(X)}\right). \tag{31}$$

Joint entropy:

$$H(X,Y) = \mathbb{E}_p \log\left(\frac{1}{p(X,Y)}\right). \tag{32}$$

Conditional entropy:

$$H(Y|X) = \mathbb{E}_p \log\left(\frac{1}{p(Y|X)}\right). \tag{33}$$

Chain rule:

$$H(X,Y) = H(X) + H(Y|X). \tag{34}$$

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.

$$D(p||q) = E_p \log\left(\frac{p(X)}{q(X)}\right). \tag{35}$$

Chain rule for relative entropy:

$$D(p(x,y)||q(x,y)) = D(p(x)||q(x)) + D(p(y|x)||q(y|x)). \tag{36}$$

Mutual information:

$$I(X;Y) = E_{p(X,Y)} \log \frac{p(X,Y)}{p(X)p(Y)} \tag{37}$$
$$= H(X) - H(X|Y) \tag{38}$$
$$= H(X) + H(Y) - H(X,Y) \tag{39}$$

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

$$D(p||q) \geq 0. \tag{40}$$

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,*

$$I(X;Y) \geq 0. \tag{41}$$

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)**.**

$$H(X|Y) \leq H(X). \tag{42}$$

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*

$$H(\mathbf{x}) = \int_{\mathbb{R}} x_s^\top d\mu(s) = \sum_{i=1}^{d} \int_{\mathbb{R}} x_{s,i} d\mu_i(s). \tag{43}$$

In addition, we have the linear functional norm

$$\|H\|_\infty := \sup_{\|\mathbf{x}\|_{\mathcal{X}} \leq 1} |H(\mathbf{x})| = \|\mu\|_1(\mathbb{R}) := \sum_i |\mu_i|(\mathbb{R}). \tag{44}$$

In particular, this linear functional norm is compatible with the norm considered for nonlinear functionals in **??**.

C. Theoretical results and proofs

In Appendix C.1, 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. 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:

$$|h_T|_\infty = \left| h_0 + \sum_{k=1}^{T} \Lambda^k U x_{k-1} \right|_\infty \tag{45}$$

$$\leq |h_0|_\infty + \frac{1-\lambda^T}{1-\lambda} |U|_1 \sup_k |x_k|_\infty \tag{46}$$

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

$$\lambda < 1 - \frac{|U|_1 \sup_k |x_k|_\infty}{M - |h_0|_\infty}. \tag{47}$$

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.2. Input-dependent gating in state-space models is associative

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

$$(W_1, h_1) \circ (W_2, h_2) = (W_2 \odot W_1, h_1 + W_1 \odot h_2). \tag{48}$$

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

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

$$\Big((W_1, h_1) \circ (W_2, h_2)\Big) \circ (W_3, h_3) = (W_1, h_1) \circ \Big((W_2, h_2) \circ (W_3, h_3)\Big) \tag{49}$$

Proof.

$$\begin{aligned}
&\Big((W_1, h_1) \circ (W_2, h_2)\Big) \circ (W_3, h_3) && (50)\\
&= (W_2 \odot W_1, h_1 + W_1 \odot h_2) \circ (W_3, h_3) && (51)\\
&= (W_3 \odot W_2 \odot W_1, h_1 + W_1 \odot h_2 + W_1 \odot W_2 \odot h_3) && (52)\\
&= (W_1, h_1) \circ (W_3 \odot W_2, h_2 + W_2 \odot h_3) && (53)\\
&= (W_1, h_1) \circ \Big((W_2, h_2) \circ (W_3, h_3)\Big) && (54)
\end{aligned}$$

□

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:*

$$H(X_{k+1}|X_1, \dots, X_k) \leq H(X_{k+1}|X_1, \dots, X_k) \leq H(X_{k+1}|X_k) \leq H(X_{k+1}). \tag{55}$$

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

D. Additional numerical results

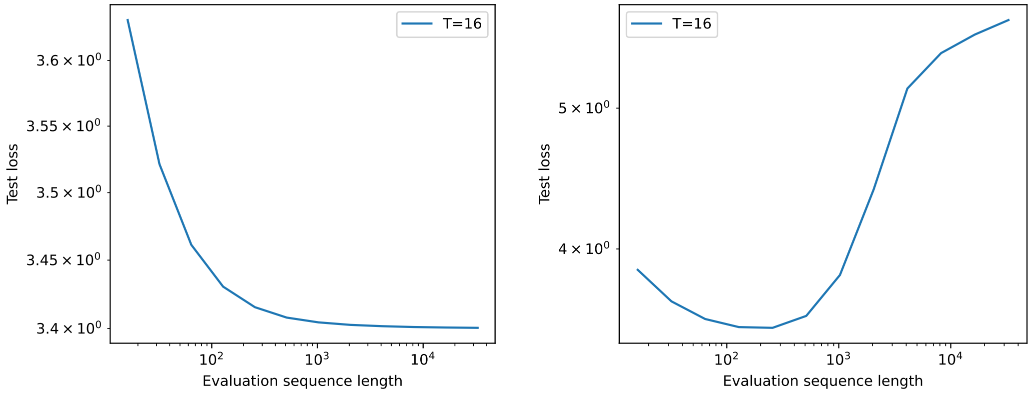
In this section, we provide additional numerical results to show previous-initialized hidden state is also effective for S5 (Appendix D.1), the empirical similarity between length extension and polynomial extrapolation (Appendix D.2) and the benefit of previous-hidden states in training for GRU (Appendix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

D.1. Comparison of different hidden states initialization

In Figure 3 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.

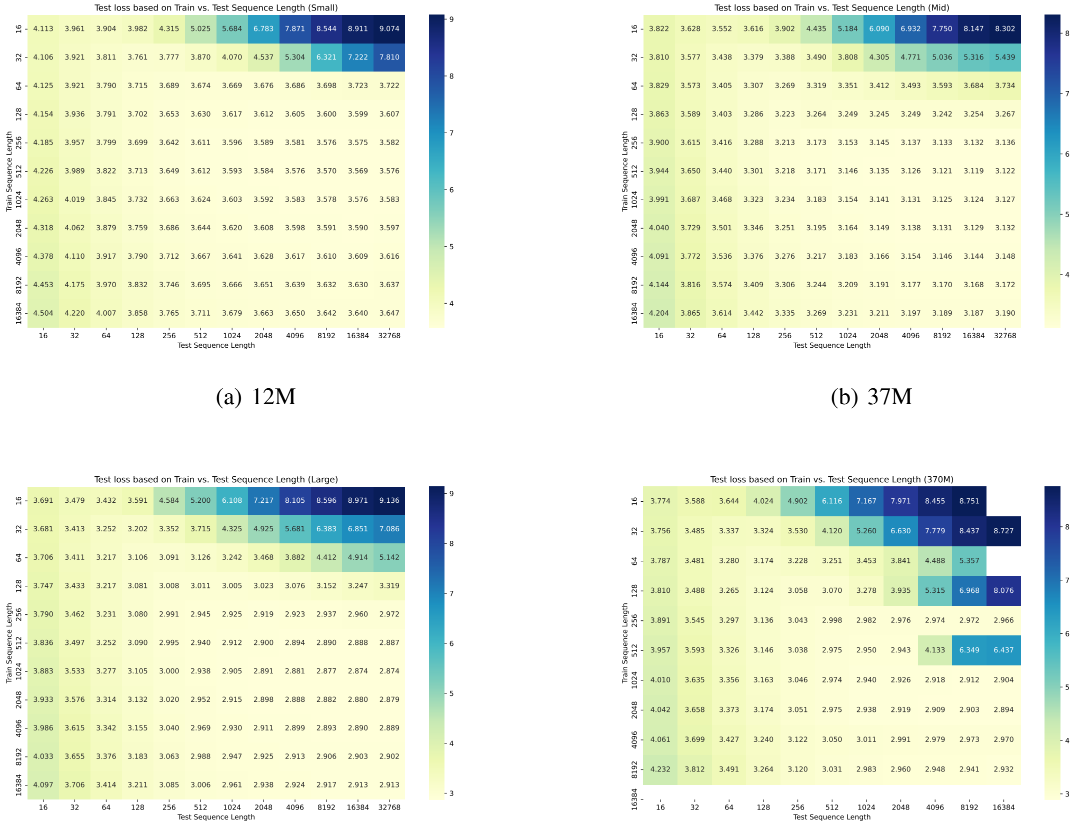
D.2. Overfitting in length extension

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.



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



(a) 12M

(b) 37M

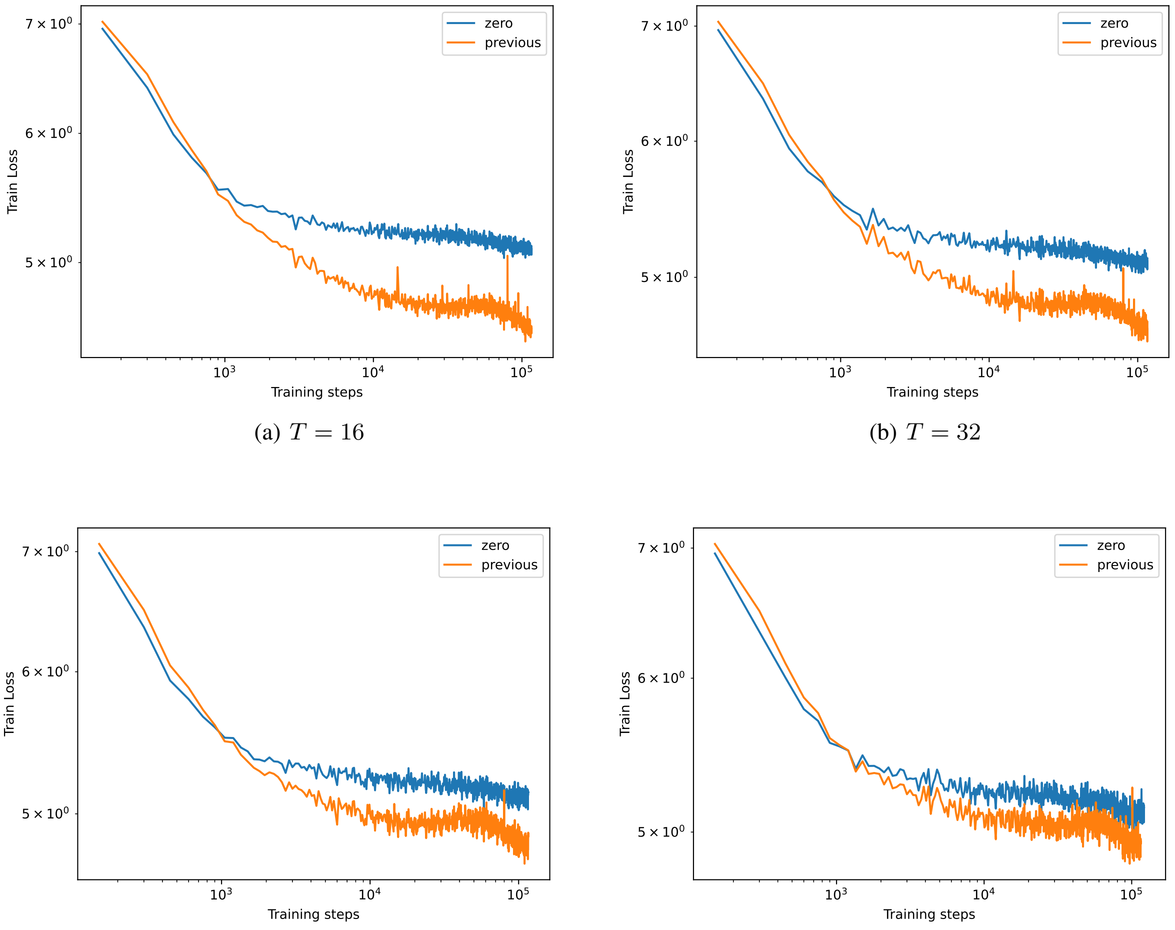
(c) 127M

(d) 370M

Figure 9. Zero-initialized S5 of different sizes trained on Wikitext103. It can be seen the perfomrance in the upper triangular area reflects the capability of length extension. We scale up models and keep the training setting to be the same. As models are trained on zero-initialized hidden states, larger models require longer training context to do well in length extension.

D.3. Generalization to other recurrent models

Beyond the state-space models, we further compare the different hidden states initialization scheme for 6-layer GRU with 30M parameters. It can be seen the previous hidden states can improve the training performance for GRU.



(a) $T = 16$

(b) $T = 32$

(c) $T = 64$

(d) $T = 128$

Figure 10. Training with previous hidden states initialization also improves the training for GRU.