StableSSM: Alleviating the Curse of Memory in State-space Models through Stable Reparameterization

Shida Wang [1] **Qianxiao Li** [1]

Abstract

In this paper, we investigate the long-term memory learning capabilities of state-space models (SSMs) from the perspective of parameterization. We prove that state-space models without any reparameterization exhibit a memory limitation similar to that of traditional RNNs: the target relationships that can be stably approximated by state-space models must have an exponential decaying memory. Our analysis identifies this "curse of memory" as a result of the recurrent weights converging to a stability boundary, suggesting that a reparameterization technique can be effective. To this end, we introduce a class of reparameterization techniques for SSMs that effectively lift its memory limitations. Besides improving approximation capabilities, we further illustrate that a principled choice of reparameterization scheme can also enhance optimization stability. We validate our findings using synthetic datasets, language models and image classifications.

1. Introduction

Understanding long-term memory relationships is fundamental in sequence modeling. Capturing this prolonged memory is vital, especially in applications like time series prediction (Connor et al., 1994), language models (Sutskever et al., 2011). Since its emergence, transformers (Vaswani et al., 2017) have become the go-to models for language representation tasks (Brown et al., 2020). However, a significant drawback lies in their computational complexity, which is asymptotically $O(T^2)$, where T is the sequence length. This computational bottleneck has been a critical impediment to the further scaling-up of transformer models. State-space models such as S4 (Gu et al., 2021), S5 (Smith et al., 2023), RWKV (Peng et al., 2023), RetNet (Sun et al., 2023) and Mamba (Gu & Dao, 2023) offer an alternative approach. These models are of the recurrent type and excel in long-term memory learning. Their architecture is specifically designed to capture temporal dependencies over extended sequences, providing a robust solution for tasks requiring long-term memory (Tay et al., 2021). One of the advantages of state-space models over traditional RNNs lies in their computational efficiency, achieved through the application of parallel scan algorithms (Martin & Cundy, 2018). Traditional nonlinear RNNs are often plagued by slow forward and backward propagation, a limitation that state-space models circumvent by leveraging linear RNN blocks.

As traditional nonlinear RNNs exhibit an asymptotically exponential decay in memory (Wang et al., 2023), the SSMs variants like S4 overcome some of the memory issues. The previous empirical results suggest that either (i) the "linear dynamics and nonlinear layerwise activation" or (ii) the parameterization inherent to S4, is pivotal in achieving the enhanced performance. Current research answers which one is more important. We first prove an inverse approximation theorem showing that state-space models without reparameterization still suffer from the "curse of memory", which is consistent with empirical results (Wang & Xue, 2023). This rules out the point (i) as the reason for SSMs' good long-term memory learning. A natural question arises regarding whether the reparameterizations are the key to learn long-term memory. We prove a class of reparameterization functions f, which we call stable reparameterization, enables the stable approximation of nonlinear functionals. This includes commonly used exponential reparameterization and softplus reparameterization. Furthermore, we question whether S4's parameterizations are optimal. Here we give a particular sense in terms of optimization stability that they are not optimal. We propose the optimal one and show its stability via numerical experiments.

We summarize our main contributions as follow:

1. We prove that similar to RNNs, the state-space models without reparameterization can only stably approximate targets with exponential decaying memory.
2. We identify a class of stable reparameterization which

[1]Department of Mathematics, National University of Singapore. Correspondence to: Qianxiao Li <qianxiao@nus.edu.sg>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

achieves the stable approximation of **any nonlinear functionals**. Both theoretical and empirical evidence highlight that stable reparameterization is crucial for long-term memory learning.

3. From the optimization viewpoint, we propose the gradient boundedness as the criterion and show the gradients are bounded by a form that depends on the parameterization. Based on the gradient bound, we solve the differential equation and derive the **"best" reparameterization in the stability sense** and verify the stability of this new reparameterization across different parameterization schemes.

Notation. We use the bold face to represent the sequence while then normal letters are scalars, vectors or functions. Throughout this paper 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.

2. Background

In this section, we first introduce the state-space models and compare them to traditional nonlinear RNNs. Subsequently, we adopt the sequence modeling as a problem in nonlinear functional approximation framework. Specifically, the theoretical properties we anticipate from the targets are defined. Moreover, we define the "curse of memory" phenomenon and provide a concise summary of prior theoretical definitions and results concerning RNNs.

2.1. State-space models

State-space models (SSMs) are a family of neural networks specialized in sequence modeling. Unlike Recurrent Neural Networks (RNNs) (Rumelhart et al., 1986), SSMs have layer-wise nonlinearity and linear dynamics within their hidden states. This unique structure facilitates accelerated computing using FFT (Gu et al., 2021) or parallel scan (Martin & Cundy, 2018). With trainable weights $W \in \mathbb{R}^{m\times m}, U \in \mathbb{R}^{m\times d}, b, c \in \mathbb{R}^m$ and activation function $\sigma(\cdot)$, the simplest SSM maps d-dimensional input sequence $\mathbf{x} = \{x_t\}$ to 1-dimensional output sequence $\{\hat{y}_t\}$. To simplify our analysis, we utilize the continuous-time framework referenced in Li et al. (2020):

$$\begin{aligned} \tfrac{dh_t}{dt} &= Wh_t + Ux_t + b, \quad h_{-\infty} = 0, \\ \hat{y}_t &= c^\top \boldsymbol{\sigma}(h_t), \qquad\qquad t \in \mathbb{R}. \end{aligned} \tag{1}$$

As detailed in Appendix A, the above form is a simplification of practical SSMs in the sense that practical SSMs can be realized by the stacking of Equation (1).

It is known that multi-layer state-space models are universal approximators (Wang & Xue, 2023; Orvieto et al., 2023). In particular, when the nonlinearity is added layer-wise, it is sufficient (in approximation sense) to use *real diagonal* W (Gu et al., 2022; Li et al., 2022). In this paper, we only consider the real diagonal matrix case and denote it by $\Lambda = \mathrm{Diag}(\lambda_1, \ldots, \lambda_m)$.

$$\frac{dh_t}{dt} = \Lambda h_t + Ux_t + b. \tag{2}$$

Compared with S4, the major differences lie in initialization such as HiPPO (Gu et al., 2020) and parameters saving method such as DPLR (Gu et al., 2022) and NPLR (Gu et al., 2021).

2.2. Sequence modeling as nonlinear functional approximations

Sequential modeling aims to discern the association between an input series, represented as $\mathbf{x} = \{x_t\}$, and its corresponding output series, denoted as $\mathbf{y} = \{y_t\}$. The input series are continuous bounded inputs vanishing at infinity: $\mathbf{x} \in \mathcal{X} = C_0(\mathbb{R}, \mathbb{R}^d)$ with norm $\|\mathbf{x}\|_\infty := \sup_{t\in\mathbb{R}} |x_t|_\infty$. It is assumed that the input and output sequences are determined from the inputs via a set of functionals, symbolized as

$$\mathbf{H} = \{H_t : \mathcal{X} \to \mathbb{R} : t \in \mathbb{R}\}, \tag{3}$$

through the relationship $y_t = H_t(\mathbf{x})$. In essence, the challenge of sequential approximation boils down to estimating the desired functional sequence $\mathbf{H}$ using a different functional sequence $\widehat{\mathbf{H}}$ potentially from a predefined model space such as SSMs.

In this paper we focus on target functionals that are bounded, causal, continuous, regular, time-homogeneous (time-shift invariant). Formal definitions are given in Appendix B.1. The continuity, boundedness, time-homogeneity, causality are important properties for good sequence-to-sequence models to have. Linearity is an important simplification as many theoretical theorems are available in functional analysis. Without loss of generality, we assume that the nonlinear functionals satisfy $H_t(\mathbf{0}) = 0$. It can be achieved via studying $H_t^{\text{adjusted}}(\mathbf{x}) = H_t(\mathbf{x}) - H_t(\mathbf{0})$.

2.3. Memory function, stable approximation and curse of memory

The concept of memory has been extensively explored in academic literature, yet much of this work relies on heuristic approaches and empirical testing, particularly in the context of learning long-term memory (Poli et al., 2023). Here we study the memory property from a theoretical perspective.

Our study employs the extended framework proposed by Wang et al. (2023), which specifically focuses on nonlinear

RNNs. However, these studies do not address the case of state-space models. Within the same framework, the slightly different memory function and decaying memory concepts enable us to explore the approximation capabilities of nonlinear functionals using SSMs.

Definition 2.1 (Memory function)**.** For bounded, causal, continuous, regular and time-homogeneous nonlinear functional sequences $\mathbf{H} = \{H_t : t \in \mathbb{R}\}$ on $\mathcal{X}$, define the following function as the *memory function* of $\mathbf{H}$: Over bounded Heaviside input $\mathbf{u}^x(t) = x \cdot \mathbf{1}_{\{t \geq 0\}}$

$$\mathcal{M}(\mathbf{H})(t) := \sup_{x \neq 0} \frac{\left|\frac{d}{dt} H_t(\mathbf{u}^x)\right|}{|x|_\infty + 1}. \tag{4}$$

We add 1 in the memory function definition to make it more regular. The memory function of the target functionals is assumed to be finite for all $t \in \mathbb{R}$.

Definition 2.2 (Decaying memory)**.** The functional sequences $\mathbf{H}$ has a *decaying memory* if $\lim_{t\to\infty} \mathcal{M}(\mathbf{H})(t) = 0$. In particular, we say it has an *exponential (polynomial) decaying memory* if there exists constant $\beta > 0$ such that $\lim_{t\to\infty} e^{\beta t}\mathcal{M}(\mathbf{H})(t) = 0$ ($\lim_{t\to\infty} t^{\beta}\mathcal{M}(\mathbf{H})(t) = 0$).

Similar to Wang et al. (2023), this adjusted memory function definition is also compatible with the memory concept in linear functional which is based on the famous Riesz representation theorem (Theorem B.3 in Appendix B). As shown in Appendix C.1, the nonlinear functionals constructed by state-space models are point-wise continuous over Heaviside inputs. Combined with time-homogeneity, we know that state-space models are nonlinear functionals with decaying memory (see Appendix C.2).

Definition 2.3 (Functional sequence approximation in Sobolev-type norm)**.** Given functional sequences $\mathbf{H}$ and $\widehat{\mathbf{H}}$, we consider the approximation in the following Sobolev-type norm (Appendix B.2):

$$\left\|\mathbf{H} - \widehat{\mathbf{H}}\right\|_{W^{1,\infty}} := \tag{5}$$

$$\sup_t \left(\|H_t - \widehat{H}_t\|_\infty + \left\| \frac{dH_t}{dt} - \frac{d\widehat{H}_t}{dt} \right\|_\infty \right). \tag{6}$$

Definition 2.4 (Perturbation error)**.** For target $\mathbf{H}$ and parameterized model $\widehat{\mathbf{H}}(\cdot, \theta_m), \theta_m = (\Lambda, U, b, c) \in \Theta_m := \{\mathbb{R}^{m\times m} \times \mathbb{R}^{m\times d} \times \mathbb{R}^m \times \mathbb{R}^m\}$, we define the *perturbation error* for hidden dimension m:

$$E_m(\beta) := \sup_{\tilde{\theta}_m \in \{\theta : |\theta - \theta_m|_2 \leq \beta\}} \|\mathbf{H} - \widehat{\mathbf{H}}(\cdot; \tilde{\theta}_m)\|_{W^{1,\infty}}. \tag{7}$$

In particular, $\widetilde{\mathbf{H}}$ refers to the perturbed models $\widehat{\mathbf{H}}(\cdot; \tilde{\theta}_m)$. Moreover, $E(\beta) := \limsup_{m\to\infty} E_m(\beta)$ is the asymptotic perturbation error. The weight norm for SSM is $|\theta|_2 := \max(|\Lambda|_2, |U|_2, |b|_2, |c|_2)$.

Based on the definition of perturbation error, we consider the stable approximation as introduced by Wang et al. (2023).

Definition 2.5 (Stable approximation)**.** Let $\beta_0 > 0$. A target functional sequence $\mathbf{H}$ admits a β_0-*stable approximation* if the perturbed error satisfies that: (i) $E(0) = 0$; (ii) $E(\beta)$ is continuous for $\beta \in [0, \beta_0]$.

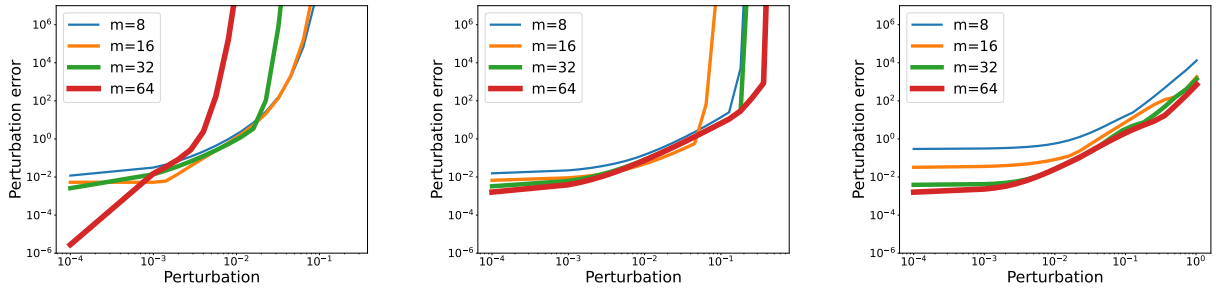
Equation $E(0) = 0$ means the universal approximation is achieved by the hypothesis space. Stable approximation strengthens the universal approximation by requiring the model to be robust against perturbation on the weights. As the stable approximation is the necessary requirement for the optimal parameters to be found by the gradient-based optimizations, it is a desirable assumption.

The "curse of memory" phenomenon, which was originally formulated for linear functionals and linear RNNs, is well-documented in prior research (Li et al., 2020; 2022; Jiang et al., 2023). It describes the phenomenon where targets approximated by linear, hardtanh, or tanh RNNs must demonstrate an exponential decaying memory. However, empirical observations suggest that state-space models, particularly the S4 variant, may possess favorable properties. Thus, it is crucial to ascertain whether the inherent limitations of RNNs can be circumvented using state-space models. Given the impressive performance of state-space models, notably S4, a few pivotal questions arise: Do the model structure of state-space models overcome the "curse of memory"? In the subsequent section, we will demonstrate that the model structure of state-space models does not indeed address the curse of memory phenomenon.

3. Main results

In this section, we first prove that similar to the traditional recurrent neural networks (Li et al., 2020; Wang et al., 2023), state-space models without reparameterization suffer from the "curse of memory" problem. This implies the targets that can be stably approximated by SSMs must have exponential decaying memory. Our analysis reveals that the problem arises from recurrent weights converging to a stability boundary when learning targets associated with long-term memory. Therefore, we introduce a class of stable reparameterization techniques to achieve the stable approximation for targets with polynomial decaying memory.

Beside the benefit of approximation perspective, we also discuss the optimization benefit of the stable reparameterizations. We show that the stable reparameterization can make the gradient scale more balanced, therefore the optimization of large models can be more stable.



(a) SSM (b) SoftplusSSM (c) S4

Figure 1. State-space models without stable reparameterization cannot approximate targets with polynomial decaying memory. In (a), the intersection of lines are shifting towards left as the hidden dimension m increases. In (b), SSMs using softplus reparameterization has a stable approximation. In (c), S4 can stably approximate the target with better stability.

Table 1. Impact of stable reparameterizations in approximation and stable approximation. As the reparameterization does not change the hypothesis space of SSMs, both vanilla SSMs and StableSSM are universal approximators. Vanilla SSMs can only stably approximate targets with exponential decay while StableSSM can stably approximate any targets with decaying memory.

	Approximation	Stable approximation
Without reparameterization (Vanilla SSM)	Universal (Wang & Xue, 2023)	Not universal (Thm 3.3)
With stable reparameterization (StableSSM)	Universal (Wang & Xue, 2023)	Universal (Thm 3.5)

3.1. Curse of memory in SSMs

In this section, we present a theoretical theorem demonstrating that the state-space structure does not alleviate the "curse of memory" phenomenon. State-space models consist of alternately stacked linear RNNs and nonlinear activations. Our result is established for both the shallow case and deep case (Remark C.3). As recurrent models, SSMs without reparameterization continue to exhibit the commonly observed phenomenon of exponential memory decay, as evidenced by empirical findings (Wang & Xue, 2023).

Assumption 3.1. We assume the hidden states remain uniformly bounded for any input sequence $\mathbf{x}$, irrespective of the hidden dimensions m. Specifically, this can be expressed as

$$\sup_m \sup_t |h_t|_\infty < \infty. \tag{8}$$

Assumption 3.2. We focus on strictly increasing, continuously differentiable nonlinear activations with Lipschitz constant L_0. This property holds for activations such as tanh, sigmoid, softsign $\sigma(z) = \frac{z}{1+|z|}$.

Theorem 3.3 (Curse of memory in SSMs)**.** *Assume* $\mathbf{H}$ *is a sequence of bounded, causal, continuous, regular and time-homogeneous functionals on* $\mathcal{X}$ *with decaying memory. Suppose there exists a sequence of state-space models* $\{\hat{\mathbf{H}}(\cdot, \theta_m)\}_{m=1}^\infty$ *β_0-stably approximating* $\mathbf{H}$ *in the norm defined in Equation (5). Assume the model weights are uniformly bounded:* $\theta_{\max} := \sup_m |\theta_m|_2 < \infty$. *Then the memory function* $\mathcal{M}(\mathbf{H})(t)$ *of the target decays exponentially:*

$$\mathcal{M}(\mathbf{H})(t) \le (d+1)L_0\theta_{\max}^2 e^{-\beta t}, \quad t \ge 0, \beta < \beta_0. \tag{9}$$

Here d is the dimension of input sequences. When generalized to multi-layer cases, the memory function bound induced from ℓ-layer SSM is: For some polynomial $P(t)$ with degree at most $l-1$

$$\mathcal{M}(\mathbf{H})(t) \le (d+1)L_0^\ell \theta_{\max}^{\ell+1} P(t) e^{-\beta t}, \quad t \ge 0, \beta < \beta_0. \tag{10}$$

The proof of Theorem 3.3 is provided in Appendix C.3. The stability boundary is discussed in Remark C.1. Compared with previous results (Li et al., 2020; Wang et al., 2023), the main proof difference comes from Lemma C.10 as the activation is in the readout $y_t = c^\top \sigma(h_t)$. Our results provide a more accurate characterization of memory decay, in contrast to previous works that only offer qualitative estimates. A consequence of Theorem 3.3 is that if the target exhibits a non-exponential decay (e.g., polynomial decay), the recurrent weights converge to a stability boundary, thereby making the approximation unstable. Finding optimal weights can become challenging with gradient-based optimization methods, as the optimization process tends to become unstable with the increase of model size. The numerical verification is presented in Figure 1 (a). The lines intersect and the intersections points shift towards the 0, suggesting that the stable radius β_0 does not exist.

Therefore SSMs without reparameterization cannot stably approximate targets with polynomial decaying memory.

3.2. Stable reparameterization and its advantage in approximation

The proof of Theorem 3.3 suggests that the "curse of memory" arises due to the recurrent weights approaching a stability boundary. Additionally, our numerical experiments (in Figure 1 (c)) show that while state-space models suffer from curse of memory, the commonly used S4 layer (with exponential reparameterization) ameliorates this issue. However, it is not a unique solution. Our findings highlight that the foundation to achieving a stable approximation is the stable reparameterization method, which we define as follows:

Definition 3.4 (Stable reparameterization)**.** We say a reparameterization scheme $f : \mathbb{R} \to \mathbb{R}$ is stable if there exists a continuous function g such that: $g : [0, \infty) \to [0, \infty), g(0) = 0$:

$$\sup_{w} \left[|f(w)| \sup_{|\tilde{w}-w|\leq\beta} \int_0^\infty \left| e^{f(\tilde{w})t} - e^{f(w)t} \right| dt \right] \leq g(\beta). \tag{11}$$

For example, commonly used reparameterization (Gu et al., 2021; Smith et al., 2023) such as $f(w) = -e^w$, $f(w) = -\log(1 + e^w)$ are all stable. Verifications are provided in Remark C.4.

As depicted in Figure 1 (b), state-space models with stable reparameterization can approximate targets exhibiting polynomial decay in memory. In particular, we prove that under a simplified perturbation setting (solely perturbing the recurrent weights), any linear functional can be stably approximated by linear RNNs. This finding under simplified setting is already significant as the instability in learning long-term memory mainly comes from the recurrent weights.

Theorem 3.5 (Existence of stable approximation by stable reparameterization)**.** *For **any** bounded, causal, continuous, regular, time-homogeneous linear functional* $\mathbf{H}$*, assume* $\mathbf{H}$ *is approximated by a sequence of linear RNNs* $\{\widehat{\mathbf{H}}(\cdot, \theta_m)\}_{m=1}^\infty$ *with stable reparameterization, then this approximation is a stable approximation.*

The proof of Theorem 3.5 is in Appendix C.4. The generalization to nonlinear functionals with Volterra-Series representation can be similarly achieved (Remark C.5). Compared to Theorem 3.3, Theorem 3.5 underscores the role of stable reparameterization in achieving stable approximation of nonlinear functional with long-term memory. Although vanilla SSM and StableSSM operate within the same hypothesis space, StableSSM demonstrates better stability in approximating any decaying memory target (Table 1). In contrast, the vanilla SSM model is limited to stably approximate targets characterized by an exponential memory decay.

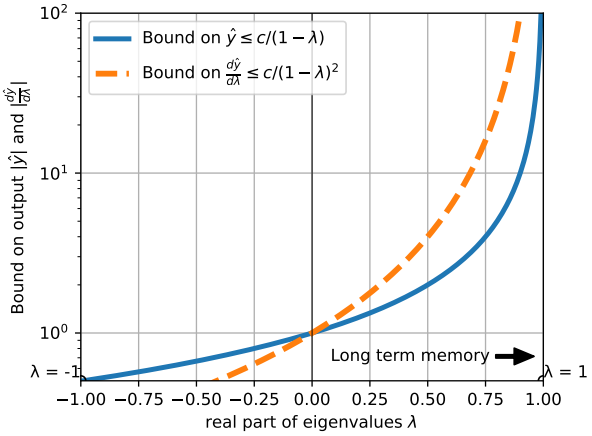


Figure 2. The scaling of layer output bound $|\hat{y}| \leq \frac{c}{1-\lambda}$ and the gradients $|\frac{d\hat{y}}{d\lambda}| \leq \frac{c}{(1-\lambda)^2}$. The stability boundary is $\lambda = \pm 1$.

3.3. Optimization benefit of stable reparameterization

In the previous section, the approximation benefit of stable reparameterizations in SSMs is discussed. Here we study the impact of different parameterizations on the optimization stability, in particular, the gradient scales.

As pointed out by Li et al. (2020; 2022), the approximation of linear functionals using linear RNNs can be reduced into the approximation of L_1-integrable memory function $\rho(t)$ via functions of the form $\hat{\rho}(t) = \sum_{i=1}^m c_i e^{-\lambda_i t}$.

$$\rho(t) \approx \sum_{i=1}^{m} c_i e^{-\lambda_i t}, \quad \lambda_i > 0. \tag{12}$$

Within this framework, λ_i is interpreted as the decay mode. Approaching this from the gradient-based optimization standpoint, and given that learning rates are shared across different decay modes, a fitting characterization for "good parameterization" emerges: *The gradient scale across different memory decays modes should be Lipschitz continuous with respect to the weights scale.*

$$|\text{Gradient}| := \left| \frac{\partial \text{Loss}}{\partial \lambda_i} \right| \leq L|\lambda_i|. \tag{13}$$

The Lipschitz constant is denoted by L. Without this property, the optimization process can be sensitive to the learning rate. We give a detailed discussion in Appendix D. In the following theorem, we first characterize the relationship between gradient norms and recurrent weight parameterization.

Theorem 3.6 (Parameterizations influence the gradient norm scale)**.** *Assume the target functional sequence* $\mathbf{H}$ *is being approximated by a sequence of SSMs* $\widehat{\mathbf{H}}_m$*. If the (diagonal) recurrent weight matrix is parameterized via*

$f : \mathbb{R} \to \mathbb{R} : f(w) = \lambda$. w is the trainable weight while λ is the eigenvalue of recurrent weight matrix Λ. The gradient norm $G_f(w)$ of weight w is upper bounded by the following function:

$$G_f(w) := \left| \frac{\partial Loss}{\partial w} \right| \leq C_{\mathbf{H},\widehat{\mathbf{H}}_m} \frac{|f'(w)|}{f(w)^2}. \tag{14}$$

Here $C_{\mathbf{H},\widehat{\mathbf{H}}_m}$ is independent of the parameterization f provided that $\mathbf{H}, \widehat{\mathbf{H}}_m$ are fixed. The discrete-time version is

$$G_f^D(w) := \left| \frac{\partial Loss}{\partial w} \right| \leq C_{\mathbf{H},\widehat{\mathbf{H}}_m} \frac{|f'(w)|}{(1-f(w))^2}. \tag{15}$$

Refer to Appendix C.5 for the proof of Theorem 3.6. In Appendix E we summarize common reparameterization methods and corresponding gradient scale functions.

Remark 3.7 (Generalization to multi-layer models). We do not prove the gradient bound result for multi-layer case in the paper, here we discuss the idea to genearlize it: Consider a specific layer in a multi-layer model, without loss of generality we also have the boundedness of result from the previous layer and expected inputs for the next layer. If we take the results from previous layer as the inputs and treat the expected inputs for next layer as the outputs, the gradient of recurrent weights for this layer also observe the same gradient norm bound with form in Equation (14). This comes from the fact that the gradient of the selected layer remains unchanged, regardless of whether the remaining layers are frozen or not.

3.4. On the "best" parameterization in stability sense

According to the criterion given in Equation (13), the "best" stable reparameterization should satisfy the following equation for some constant $L > 0$.

$$G_f(w) \leq C_{\mathbf{H},\widehat{\mathbf{H}}_m} \frac{|f'(w)|}{f(w)^2} = L|w|. \tag{16}$$

Based on the criterion, a sufficient condition for the above criterion is to find some function f that satisfies the following equation for some real $a, b \in \mathbb{R}$:

$$\frac{f'(w)}{f(w)^2} = \frac{d(-\frac{1}{f(w)})}{dw} = 2aw, \tag{17}$$

$$\frac{1}{f(w)} = -(aw^2 + b) \Rightarrow f(w) = -\frac{1}{aw^2+b}. \tag{18}$$

The first equation is achieved by integrating the function $\frac{f'(w)}{f(w)^2}$. Therefore the "best" parameterization under the assumption of the Lipschitz property of gradient is characterized by the function with two degrees of freedom: By stability requirement $f(w) \leq 0$ for all w

$$f(w) = -\frac{1}{aw^2+b}, \quad a > 0, b \geq 0. \tag{19}$$

Similarly, the discrete case gives the solution $f(w) = 1 - \frac{1}{aw^2+b}$. The stability of linear RNN further requires $a > 0$ and $b \geq 0$. We choose $a = 1, b = 0.5$ because this ensures the stability of the hidden state dynamics and stable approximation in Equation (11). Notice that $\lim_{w\to 0} 1 - \frac{1}{w^2+0.5} = -1$ which does not cross the stability boundary $\lambda = -1$. It can be seen in Figure 6 that, compared with direct and exponential reparameterizations, the softplus reparameterization is generally milder in this gradient-over-weight criterion. The "best" parameterization is optimal in the sense it has a bounded gradient-over-weight ratio across different weights w (different eigenvalues λ).

Remark 3.8. Apart from the reparameterization method, a simple yet effective method is gradient clipping. However, clipped gradient is biased there the training effectiveness of the gradient descent might be reduced .

4. Numerical verifications

Based on the above analyses, we verify the theoretical statements over synthetic tasks and language models using WikiText-103. The additional numerical details are provided in Appendix F.

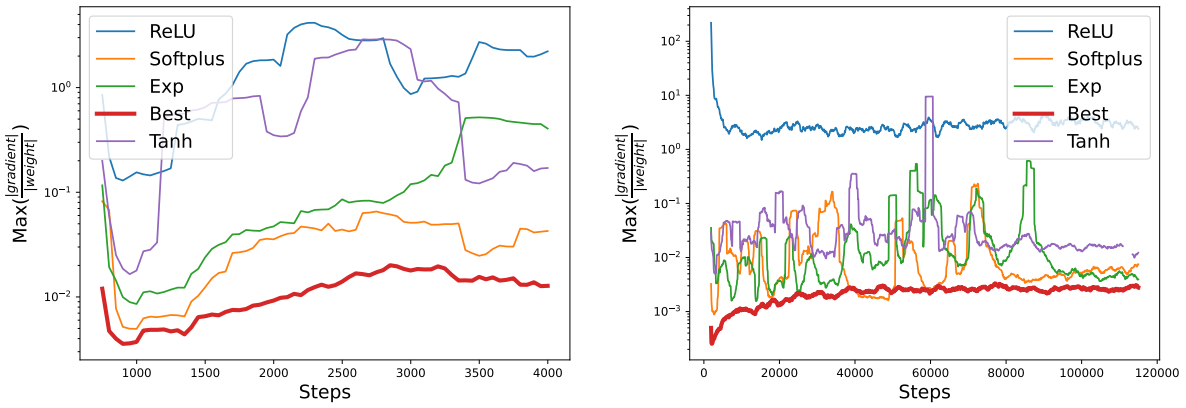
4.1. Synthetic tasks

Linear functionals have a clear structure, allowing us to study the differences of parameterizations. Similar to Li et al. (2020) and Wang et al. (2023), we consider linear functional targets $\mathbf{H}$ with following polynomial memory function $\rho(t) = \frac{1}{(t+1)^{1.1}}$: $y_t = H_t(\mathbf{x}) = \int_{-\infty}^{t} \rho(t-s)x_s ds$. We use the state-space models with tanh activations to learn the sequence relationships. In Figure 3 (a), the eigenvalues λ are initialized to be the same while the only difference is the reparameterization function $f(w)$. Training loss across different reparameterization schemes are similar but the gradient-over-weight ratio across different parameterization schemes are different in terms of the scale.

4.2. Language models

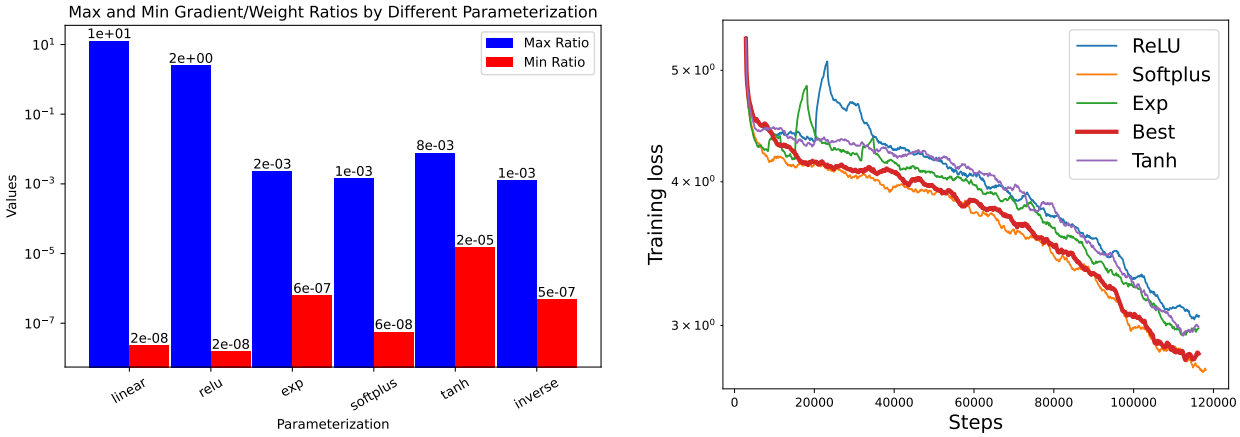
In addition to the synthetic dataset of linear functionals, we further justify Theorem 3.6 by examining the gradient-over-weight ratios for language models using state-space models (S5). In particular, we adopt the Hyena (Poli et al., 2023) architecture while the implicit convolution is replaced by a simple real-weighted state-space model (Smith et al., 2023).

In Figure 4 (a), given the same initialization, we show that stable reparameterizations such as exponential, softplus, tanh and "best" exhibit a narrower range of gradient-over-weight ratios compared to both the direct and relu reparameterizations. Beyond the gradient at the same initialization, in Figure 3 (b), we show the gradient-over-weight ratios during the training process. The stable reparameterization



(a) Linear functionals (b) Language model

Figure 3. In panel (a), in the learning of linear functionals of polynomial decaying memory, the gradient-over-weight scale range during the training of state-space models. It can be seen the "best"discrete parameterization $f(w) = 1 - \frac{1}{w^2+0.5}$ achieves the smallest gradient-over-weight scale. Such property is desirable when a large learning rate is used in training. The "best" reparameterization $f(w) = 1 - \frac{1}{w^2+0.5}$ maintains the smallest $\max(\frac{|\text{grad}|}{|\text{weight}|})$ which is crucial for the training stability. Similar results can be observed in the language modelling task as in panel (b).



(a) Gradient-weight-ratio at initialization (b) Training loss

Figure 4. Language models on WikiText-103. In the left panel (a), we show the gradient-over-weight ratio ranges for different parameterizations of recurrent weights in state-space models. The eigenvalues λ are initialized to be the same while the only difference is the reparameterization function f. In the right panel (b), the "Best" parameterization is more stable than the ReLU and exponential reparameterizations. Additional experiments for different learning rates are provided in Figure 7.

Table 2. Comparison of stability of different parameterizations over MNIST. The experiments conducted on the MNIST and CIFAR10 datasets were replicated three times, with the standard deviation of the test loss indicated in parentheses.

LR	Direct	Softplus	Exp	Best
5e-6	2.314384 (7.19932e-05)	2.241642 (0.001279)	2.241486 (0.001286)	**2.241217** (0.001297)
5e-5	2.304331 (2.11817e-07)	0.779663 (0.001801)	0.774661 (0.001685)	**0.765220** (0.001352)
5e-4	2.303190 (1.66387e-06)	0.094411 (0.000028)	0.093418 (0.000024)	**0.091924** (0.000019)
5e-3	NaN	0.023795 (0.000004)	0.023820 (0.000003)	**0.023475** (0.000002)
5e-2	NaN	0.802772 (1.69448)	0.868350 (1.55032)	**0.089073** (0.000774)
5e-1	NaN	2.313510 (0.000014)	2.314244 (0.000025)	**2.185477** (0.048238)
5e+0	NaN	NaN	NaN	**199.013813** (50690.6)

Table 3. Comparison of stability of different parameterizations over CIFAR10

LR	Direct	Softplus	Exp	Best
5e-6	NaN	1.745752 (0.000006)	1.745816 (0.000009)	**1.745290** (0.000011)
5e-5	NaN	1.220859 (0.000008)	1.218064 (0.000008)	**1.215510** (0.000014)
5e-4	NaN	0.883649 (0.000898)	**0.866817** (0.000328)	0.870412 (0.000442)
5e-3	NaN	1.449352 (0.000414)	1.567662 (0.021489)	**1.364697** (0.013849)
5e-2	NaN	1.942372 (0.011317)	1.846173 (0.007990)	**1.713892** (0.013426)
5e-1	NaN	37.802437 (3776.6383)	**2.296230** (0.000984)	2.554265 (0.168649)
5e+0	NaN	**540.621033** (NaN)	NaN	615.374522 (30795.4)

will give better gradient-over-weight ratios in the sense that the "best" stable reparameterization maintains the smallest $\max(\frac{|\text{grad}|}{|\text{weight}|})$. Specifically, as illustrated in Figure 4 (b) and Figure 7, while training with a large learning rate may render the exponential parameterization unstable, the "best" reparameterization $f(w) = 1 - \frac{1}{w^2+0.5}$ appears to enhance training stability.

4.3. Image classification

Apart from the gradient scale range shown in the language modeling experiments, we further compare the stability of different parameterization schemes over different initial learning rates. As shown in the following Table 2 and Table 3, we found that the "best" parameterization can be trained with a larger learning rates while exp/softplus parameterizations cannot be trained with larger learning rates (lr=5.0). Although the models exhibit comparable performance at lower learning rates, the "best" parameterization consistently outperforms others across a range of learning rates As the training stability issue has been widely reported for larger models [1] [2] , we believe the improved training stability is an important component in the scale-up large language models.

5. Related works

RNNs, as introduced by Rumelhart et al. (1986), represent one of the earliest neural network architectures for modeling sequential relationships. Empirical findings by Bengio et al. (1994) have shed light on the challenge of exponential decaying memory in RNNs. Various works (Hochreiter & Schmidhuber, 1997; Rusch & Mishra, 2022; Wang & Yan, 2023) have been done to improve the memory patterns of recurrent models. Theoretical approaches (Li et al., 2020; 2022; Wang et al., 2023) have been taken to study the exponential memory decay of RNNs. In this paper, we study the state-space models which are also recurrent. Our findings theoretically justify that although SSMs variants exhibit good numerical performance in long-sequence modeling (Gu et al., 2021), simple SSMs also suffer from the "curse of memory".

State-space models (Siivola & Honkela, 2003), previously discussed in control theory, has been widely used to study the dynamics of complex systems. The subsequent variants, S4(Gu et al., 2021), S5 (Smith et al., 2023), RetNet (Sun et al., 2023) and Mamba (Gu & Dao, 2023), have significantly enhanced empirical performance. Notably, they excel in the long-range arena (Tay et al., 2021), an area where transformers traditionally underperform. Contrary to the initial presumption, our investigations disclose that the ability to learn long-term memory is not derived from the linear RNN coupled with nonlinear layer-wise activations. Rather, our study underscores the benefits of stable reparameterization in both approximation and optimization.

6. Conclusion

In this paper, we study the intricacies of long-term memory learning in state-space models, specifically emphasizing the role of recurrent weights parameterization. We prove that state-space models without reparameterization fail to stably approximating targets that exhibit non-exponential decaying memory. Our analysis indicates this "curse of memory" phenomenon is caused by the eigenvalues of recurrent weight matrices converging to stability boundary. As an alternative, we introduce a class of stable reparameterization as a robust solution to this challenge, which also partially explains the performance of S4. With stable reparameterization, state-space models can stably approximate any targets with decaying memory. We also explore the optimization advantages associated with stable reparameterization, especially concerning gradient-over-weight scale. Our results give the theoretical support to observed advantages of reparameterizations in S4 and moreover give principled methods to design **"best" reparameterization scheme in the optimization stability sense.** This paper shows that stable reparameterization not only enables the learning of targets with long-term memory but also enhances the optimization stability.

[1] https://github.com/state-spaces/mamba/issues/6

[2] https://github.com/state-spaces/mamba/issues/22

Impact Statements

This paper study the approximation and optimization properties of parameterization in state-space models. This paper presents work whose goal is to advance the field of Machine Learning. There are minor potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

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.

Boyd, S., Chua, L. O., and Desoer, C. A. Analytical Foundations of Volterra Series. *IMA Journal of Mathematical Control and Information*, 1(3):243–282, January 1984. ISSN 0265-0754. doi: 10.1093/imamci/1.3.243.

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.

Connor, J. T., Martin, R. D., and Atlas, L. E. Recurrent neural networks and robust time series prediction. *IEEE transactions on neural networks*, 5(2):240–254, 1994.

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.

Gu, A., Goel, K., Gupta, A., and Ré, C. On the Parameterization and Initialization of Diagonal State Space Models. *Advances in Neural Information Processing Systems*, 35: 35971–35983, December 2022.

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.

Li, Y., Wei, C., and Ma, T. Towards explaining the regularization effect of initial large learning rate in training neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

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.

Li, Z., Han, J., E, W., and Li, Q. Approximation and Optimization Theory for Linear Continuous-Time Recurrent Neural Networks. *Journal of Machine Learning Research*, 23(42):1–85, 2022. ISSN 1533-7928.

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

Merity, S., Xiong, C., Bradbury, J., and Socher, R. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

Orvieto, A., De, S., Gulcehre, C., Pascanu, R., and Smith, S. L. On the universality of linear recurrences followed by nonlinear projections. *arXiv preprint arXiv:2307.11888*, 2023.

Peng, B., Alcaide, E., Anthony, Q., Albalak, A., Arcadinho, S., Cao, H., Cheng, X., Chung, M., Grella, M., GV, K. K., et al. RWKV: Reinventing RNNs for the transformer era. *arXiv preprint arXiv:2305.13048*, 2023.

Poli, M., Massaroli, S., Nguyen, E., Fu, D. Y., Dao, T., Baccus, S., Bengio, Y., Ermon, S., and Re, C. Hyena Hierarchy: Towards Larger Convolutional Language Models. In *International Conference on Machine Learning*, June 2023.

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.

Rusch, T. K. and Mishra, S. Coupled Oscillatory Recurrent Neural Network (coRNN): An accurate and (gradient) stable architecture for learning long time dependencies. In *International Conference on Learning Representations*, February 2022.

Siivola, V. and Honkela, A. A state-space method for language modeling. In *2003 IEEE Workshop on Automatic Speech Recognition and Understanding (IEEE Cat.*

No.03EX721), pp. 548–553, St Thomas, VI, USA, 2003. IEEE. ISBN 978-0-7803-7980-0. doi: 10.1109/ASRU.2003.1318499.

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

Smith, L. N. and Topin, N. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, volume 11006, pp. 369–386. SPIE, 2019.

Sun, Y., Dong, L., Huang, S., Ma, S., Xia, Y., Xue, J., Wang, J., and Wei, F. Retentive network: A successor to transformer for large language models. *arXiv preprint arXiv:2307.08621*, 2023.

Sutskever, I., Martens, J., and Hinton, G. Generating Text with Recurrent Neural Networks. In *International Conference on Machine Learning*, pp. 1017–1024, January 2011.

Tay, Y., Dehghani, M., Abnar, S., Shen, Y., Bahri, D., Pham, P., Rao, J., Yang, L., Ruder, S., and Metzler, D. Long Range Arena : A Benchmark for Efficient Transformers. In *International Conference on Learning Representations*, January 2021.

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 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. and Yan, Z. Improve long-term memory learning through rescaling the error temporally. *arXiv preprint arXiv:2307.11462*, 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.

A. Graphical demonstration of state-space models as stack of Equation (1)

Here we show that Equation (1) corresponds to the practical instantiation of SSM-based models in the following sense: As shown in Figure 5, any practical instantiation of SSM-based models can be implemented as a stack of Equation (1). The pointwise shallow MLP can be realized with two-layer state-space models with layer-wise nonlinearity by setting recurrent weights W to be 0.

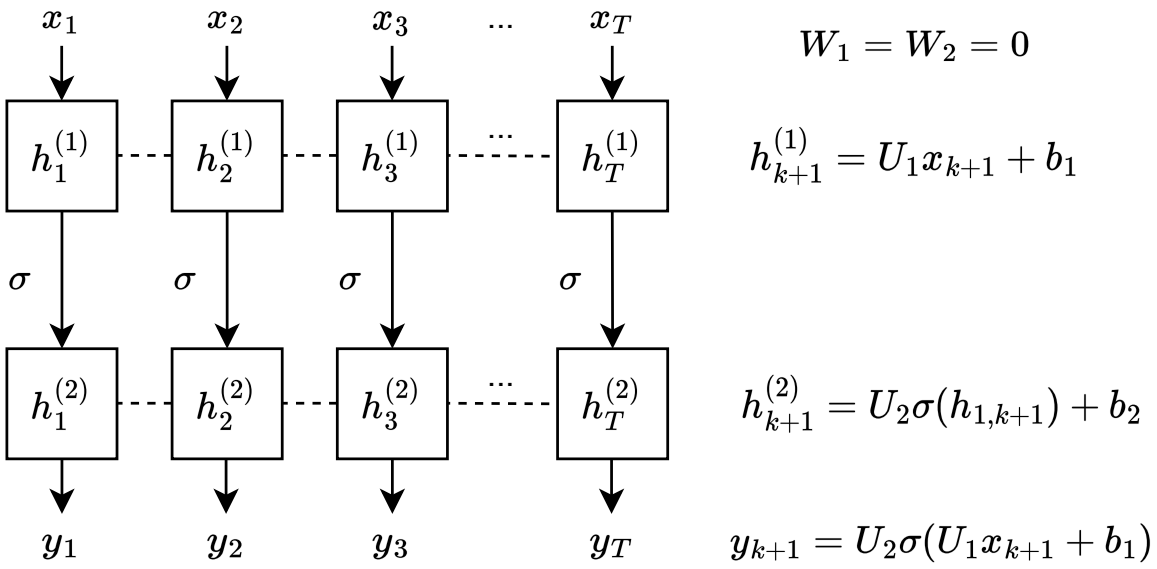


Figure 5. MLP can be realized by two-layer state-space models. The superscript indicates the layers while the subscript indicates the time index. It can be seen MLP is equivalent to SSMs having zero recurrent weights $W_1 = W_2 = 0$.

B. Theoretical backgrounds

In this section, we collect the definitions for the theoretical statements.

B.1. Properties of targets

We first introduce the definitions on (sequences of) functionals as discussed in (Wang et al., 2023).

Definition B.1. Let $\mathbf{H} = \{H_t : \mathcal{X} \mapsto \mathbb{R}; t \in \mathbb{R}\}$ be a sequence of functionals.

1. (**Linear**) H_t is linear functional if for any $\lambda, \lambda' \in \mathbb{R}$ and $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$, $H_t(\lambda \mathbf{x} + \lambda' \mathbf{x}') = \lambda H_t(\mathbf{x}) + \lambda' H_t(\mathbf{x}')$.

2. (**Continuous**) H_t is continuous functional if for any $\mathbf{x},' \mathbf{x} \in \mathcal{X}$, $\lim_{\mathbf{x}' \to \mathbf{x}} |H_t(\mathbf{x}') - H_t(\mathbf{x})| = 0$.

3. (**Bounded**) H_t is bounded functional if the norm of functional $\|H_t\|_\infty := \sup_{\{\mathbf{x} \neq 0\}} \frac{|H_t(\mathbf{x})|}{\|\mathbf{x}\|_\infty + 1} + |H_t(\mathbf{0})| < \infty$.

4. (**Time-homogeneous**) $\mathbf{H} = \{H_t : t \in \mathbb{R}\}$ is time-homogeneous (or time-shift-equivariant) if the input-output relationship commutes with time shift: let $[S_\tau(\mathbf{x})]_t = x_{t-\tau}$ be a shift operator, then $\mathbf{H}(S_\tau \mathbf{x}) = S_\tau \mathbf{H}(\mathbf{x})$.

5. (**Causal**) H_t is causal functional if it does not depend on future values of the input. That is, if $\mathbf{x}, \mathbf{x}'$ satisfy $x_t = x'_t$ for any $t \leq t_0$, then $H_t(\mathbf{x}) = H_t(\mathbf{x}')$ for any $t \leq t_0$.

6. (**Regular**) H_t is regular functional if for any sequence $\{\mathbf{x}^{(n)} : n \in \mathbb{N}\}$ such that $x_s^{(n)} \to 0$ for almost every $s \in \mathbb{R}$, then $\lim_{n \to \infty} H_t(\mathbf{x}^{(n)}) = 0$.

B.2. Approximation in Sobolev norm

Definition B.2. In sequence modeling as a nonlinear functional approximation problem, we consider the Sobolev norm of the functional sequence defined as follow:

$$\left\|\mathbf{H}-\widehat{\mathbf{H}}\right\|_{W^{1,\infty}}=\sup_t\left(\|H_t-\widehat{H}_t\|_\infty+\left\|\frac{dH_t}{dt}-\frac{d\widehat{H}_t}{dt}\right\|_\infty\right). \tag{20}$$

Here $\mathbf{H}=\{H_t:t\in\mathbb{R}\}$ is the target functional sequence to be approximated while the $\widehat{\mathbf{H}}=\{\widehat{H}_t:t\in\mathbb{R}\}$ is the model we use.

In particular, the nonlinear functional operator norm is given by:

$$\|H_t\|_\infty:=\sup_{\mathbf{x}\neq\mathbf{0}}\frac{|H_t(\mathbf{x})|}{\|\mathbf{x}\|_\infty+1}+|H(\mathbf{0})|. \tag{21}$$

As $\mathbf{H}(\mathbf{0})=0$, $\|H_t\|_\infty$ is reduced to $\sup_{\mathbf{x}\neq 0}\frac{|H_t(\mathbf{x})|}{\|\mathbf{x}\|_\infty+1}$. If $\mathbf{H}$ is a linear functional, this definition is compatible with the common linear functional norm in Equation (35).

We check this operator norm in Equation (21) is indeed a norm: Without loss of generality, we will drop the time index for brevity.

1. Triangular inequality: For nonlinear functional H_1 and H_2,

$$\|H_1+H_2\|_\infty:=\sup_{\mathbf{x}\neq\mathbf{0}}\frac{|(H_1+H_2)(\mathbf{x})|}{\|\mathbf{x}\|_\infty+1} \tag{22}$$

$$\leq\sup_{\mathbf{x}\neq\mathbf{0}}\frac{|H_1(\mathbf{x})|}{\|\mathbf{x}\|_\infty+1}+\sup_{\mathbf{x}\neq\mathbf{0}}\frac{|H_2(\mathbf{x})|}{\|\mathbf{x}\|_\infty+1}=\|H_1\|_\infty+\|H_2\|_\infty. \tag{23}$$

   The inequality is by the property of supremum.

2. Absolute homogeneity: For any real constant s and nonlinear functional H

$$\|sH\|_\infty:=\sup_{\mathbf{x}\neq\mathbf{0}}\frac{|(sH)(\mathbf{x})|}{\|\mathbf{x}\|_\infty+1}=|s|\sup_{\mathbf{x}\neq\mathbf{0}}\frac{|H(\mathbf{x})|}{\|\mathbf{x}\|_\infty+1}=|s|\|H\|_\infty. \tag{24}$$

3. Positive definiteness: If $\|H\|_\infty=0$, then for all non-zero inputs $\mathbf{x}\neq\mathbf{0}$ we have $H(\mathbf{x})=0$. As $H(\mathbf{0})=0$, then we know H is a zero functional.

Property of nonlinear functional sequence norm The definition of functional product is by the element-wise product: $(\mathbf{H}_1\mathbf{H}_2)(\mathbf{x})=\mathbf{H}_1(\mathbf{x})\odot\mathbf{H}_2(\mathbf{x})$. As the functional norm satisfies:

$$\|H_1H_2\|_\infty:=\sup_{\mathbf{x}\neq 0}\frac{|H_1(\mathbf{x})H_2(\mathbf{x})|}{\|\mathbf{x}\|_\infty+1}+|H_1(\mathbf{0})H_2(\mathbf{0})| \tag{25}$$

$$\leq\sup_{\mathbf{x}\neq\mathbf{0}}\frac{|H_1(\mathbf{x})|}{\|\mathbf{x}\|_\infty+1}\frac{|H_2(\mathbf{x})|}{\|\mathbf{x}\|_\infty+1}+|H_1(\mathbf{0})|\cdot|H_2(\mathbf{0})| \tag{26}$$

$$\leq\sup_{\mathbf{x}\neq\mathbf{0}}\left(\frac{|H_1(\mathbf{x})|}{\|\mathbf{x}\|_\infty+1}+|H_1(\mathbf{0})|\right)\sup_{\mathbf{x}\neq\mathbf{0}}\left(\frac{|H_2(\mathbf{x})|}{\|\mathbf{x}\|_\infty+1}+|H_2(\mathbf{0})|\right) \tag{27}$$

$$=\|H_1\|_\infty\|H_2\|_\infty \tag{28}$$

Therefore we have

$$\|\mathbf{H}_1\mathbf{H}_2\|_\infty = \sup_t \left(\|H_1H_2\|_\infty + \left\| \frac{d(H_1H_2)}{dt} \right\|_\infty \right) \tag{29}$$

$$= \sup_t \left(\|H_1H_2\|_\infty + \left\| H_1 \frac{dH_2}{dt} \right\|_\infty + \left\| \frac{dH_1}{dt} H_2 \right\|_\infty \right) \tag{30}$$

$$\leq \sup_t \left(\|H_1\|_\infty \|H_2\|_\infty + \|H_1\|_\infty \left\| \frac{dH_2}{dt} \right\|_\infty + \left\| \frac{dH_1}{dt} \right\|_\infty \|H_2\|_\infty \right) \tag{31}$$

$$\leq \sup_t \left(\|H_1\|_\infty + \left\| \frac{dH_1}{dt} \right\|_\infty \right) \sup_t \left(\|H_2\|_\infty + \left\| \frac{dH_2}{dt} \right\|_\infty) \right) \tag{32}$$

$$= \|\mathbf{H}_1\|_\infty \|\mathbf{H}_2\|_\infty \tag{33}$$

B.3. Riesz representation theorem for linear functional

Theorem B.3 (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*

$$H(\mathbf{x}) = \int_{\mathbb{R}} x_s^\top d\mu(s) = \sum_{i=1}^{d} \int_{\mathbb{R}} x_{s,i} d\mu_i(s). \tag{34}$$

In addition, we have the linear functional norm

$$\|H\|_\infty := \sup_{\|\mathbf{x}\|_\mathcal{X} \leq 1} |H(\mathbf{x})| = \|\mu\|_1(\mathbb{R}) := \sum_i |\mu_i|(\mathbb{R}). \tag{35}$$

In particular, this linear functional norm is compatible with the norm considered for nonlinear functionals in Equation (21).

C. Proofs for theorems and lemmas

In Appendix C.1, we show that the nonlinear functionals defined by state-space models are point-wise continuous functionals at Heaviside inputs. In Appendix C.3, the proof for state-space models' exponential memory decaying memory property is given. In Appendix C.4, we prove the linear RNN with stable reparameterization can stably approximate any linear functional. The target is no longer limited to have an exponenitally decaying memory. The gradient norm estimate of the recurrent layer is included in Appendix C.5.

C.1. Proof for SSMs are point-wise continuous functionals

Proof. Let $\mathbf{x}$ be any fixed Heaviside input. Assume $\lim_{k\to\infty} \|\mathbf{x}_k - \mathbf{x}\|_\infty = 0$. Let $h_{k,t}$ and h_t be the hidden state for inputs $\mathbf{x}_k$ and $\mathbf{x}$. Without loss of generality, assume $t > 0$. The following $|\cdot|$ refers to $p = \infty$ norm.

By definition of the hidden states dynamics and triangular inequality, since $\sigma(\cdot)$ is Lipschitz continuous

$$\frac{d|h_{k,t} - h_t|}{dt} = |\sigma(\Lambda h_{k,t} + Ux_{k,t}) - \sigma(\Lambda h_t + Ux_t)| \tag{36}$$

$$\leq L|\Lambda h_{k,t} + Ux_{k,t} - \Lambda h_t - Ux_t| \tag{37}$$

$$= L|\Lambda(h_{k,t} - h_t) + U(x_{k,t} - x_t)| \tag{38}$$

$$\leq L(|\Lambda||h_{k,t} - h_t| + |U||x_{k,t} - x_t|). \tag{39}$$

Here L is the Lipschitz constant of activation σ. Apply the Grönwall inequality to the above inequality, we have:

$$|h_{k,t} - h_t| \leq \int_0^t e^{L|\Lambda|(t-s)} L|U|\ |x_{k,s} - x_s| ds. \tag{40}$$

As the inputs are bounded, by dominated convergence theorem we have right hand side converges to 0 therefore

$$\lim_{k\to\infty} |h_{k,t} - h_t| = 0, \quad \forall t. \tag{41}$$

Let $y_{k,t}$ and y_t be the outputs for inputs $\mathbf{x}_k$ and $\mathbf{x}$. Therefore we show the point-wise convergence of $\frac{dH_t}{dt}$ at $\mathbf{x}$:

$$\lim_{k\to\infty}\left|\frac{dy_{k,t}}{dt}-\frac{dy_t}{dt}\right| = \lim_{k\to\infty}\left|c^\top(\frac{dh_{k,t}}{dt}-\frac{dh_t}{dt})\right| \tag{42}$$

$$\leq \lim_{k\to\infty}|c|L(|\Lambda||h_{k,t}-h_t|+|U||x_{k,t}-x_t|)=0. \tag{43}$$

□

C.2. Point-wise continuity leads to decaying memory

Here we give the proof of decaying memory based on the point-wise continuity of $\frac{dH_t}{dt}$ and boundedness and time-homogeneity of $\mathbf{H}$:

Proof.

$$\lim_{t\to\infty}\left|\frac{dH_t}{dt}(\mathbf{u}^x)\right| = \lim_{t\to\infty}\left|\frac{dH_0}{dt}(x\cdot\mathbf{1}_{\{s\geq -t\}})\right| = \left|\frac{dH_0}{dt}(\mathbf{x})\right| = 0.$$

The first equation comes from time-homogeneity. The second equation is derived from the point-wise continuity where input $\mathbf{x}$ means constant x for all time $\mathbf{x}=x\cdot\mathbf{1}_{\{s\geq-\infty\}}$. The third equation is based on the boundedness and time-homogeneity as the output over constant input should be finite and constant $H_t(\mathbf{x})=H_s(\mathbf{x})$ for all s,t. Therefore $|\frac{dH_0}{dt}(\mathbf{x})|=0$. □

C.3. Proof for Theorem 3.3

The main idea of the proof is two-fold. First of all, we show that state-space models with strictly monotone activation is decaying memory in Lemma C.10. Next, the idea of analysing the memory functions through a transform from $[0,\infty)$ to $(0,1]$ is similar to previous works (Li et al., 2020; 2022; Wang et al., 2023). The remainder of the proof follows a standard approach, as the derivatives of the hidden states follow the rules of linear dynamical systems when Heaviside inputs are considered.

Proof. Assume the inputs considered are uniformly bounded by X_0:

$$\|\mathbf{x}\|_\infty < X_0. \tag{44}$$

Define the derivative of hidden states for unperturbed model to be $v_{m,t}=\frac{dh_{m,t}}{dt}$. Similarly, $\tilde{v}_{m,t}$ is the derivative of hidden states for perturbed models $\tilde{v}_{m,t}=\frac{d\tilde{h}_{m,t}}{dt}$.

Since each perturbed model has a decaying memory and the target functional sequence $\mathbf{H}$ has a stable approximation, by Lemma C.10, we have

$$\lim_{t\to\infty}\tilde{v}_{m,t}=0,\quad \forall m. \tag{45}$$

If the inputs are limited to Heaviside inputs, the derivative $\tilde{v}_{m,t}$ satisfies the following dynamics: Notice that the hidden state satisfies $h_t=0, t\in(-\infty,0]$,

$$\frac{d\tilde{v}_{m,t}}{dt}=\widetilde{\Lambda}_m\tilde{v}_{m,t},\quad t\geq 0 \tag{46}$$

$$\tilde{v}_{m,0}=\widetilde{\Lambda}_m h_0+\widetilde{U}_m x_0+\tilde{b}_m=\widetilde{U}_m x_0+\tilde{b}_m \tag{47}$$

$$\Rightarrow \tilde{v}_{m,t}=e^{\widetilde{\Lambda}_m t}(\widetilde{U}_m x_0+\tilde{b}_m). \tag{48}$$

Notice that the perturbed initial conditions of the $\tilde{v}_{m,t}$ are uniformly (in m) bounded:

$$\begin{aligned}
\tilde{V}_0 &:= \sup_m |\tilde{v}_{m,0}|_2 && (49)\\
&= \sup_m |\widetilde{U}_m x_0 + \tilde{b}_m|_2 && (50)\\
&\leq \sup_m |\widetilde{U}_m x_0 + \tilde{b}_m|_2 && (51)\\
&\leq dX_0(\sup_m \|U_m\|_2 + \beta_0) + \sup_m \|b_m\|_2 + \beta_0 && (52)\\
&< \infty && (53)
\end{aligned}$$

Here d is the input sequence dimension.

Similarly, the unperturbed initial conditions satisfy:

$$\begin{aligned}
V_0 &:= \sup_m |\tilde{v}_{m,0}|_2 && (54)\\
&= \sup_m |U_m x_0 + b_m|_2 && (55)\\
&\leq \sup_m |U_m x_0 + b_m|_2 && (56)\\
&\leq dX_0 \sup_m \|U_m\|_2 + \sup_m \|b_m\|_2 && (57)\\
&\leq (dX_0 + 1)\theta_{max} && (58)\\
&< \infty && (59)
\end{aligned}$$

Select a sequence of perturbed recurrent matrices $\{\widetilde{\Lambda}_{m,k}\}_{k=1}^{\infty}$ satisfying the following two properties:

1. $\widetilde{\Lambda}_{m,k}$ is Hyperbolic, which means the real part of the eigenvalues of the matrix are nonzero.

2. $\lim_{k\to\infty}(\widetilde{\Lambda}_{m,k} - \Lambda_m) = \beta_0 I_m$.

Moreover, by Lemma C.11, we know that each hyperbolic matrix $\widetilde{\Lambda}_{m,k}$ is Hurwitz as the system for $\tilde{v}_{m,t}$ is asymptotically stable.

$$\sup_m \max_{i\in[m]}(\lambda_i(\widetilde{\Lambda}_{m,k})) < 0. \tag{60}$$

This is the stability boundary for the state-space models under perturbations.

Therefore the original diagonal unperturbed recurrent weight matrix Λ_m satisfies the following eigenvalue inequality **uniformly** in m. Since Λ_m is diagonal:

$$\sup_m \max_{i\in[m]}(\lambda_i(\Lambda_m)) \leq -\beta_0. \tag{61}$$

Therefore the model memory decays exponentially uniformly

$$\mathcal{M}(\widehat{\mathbf{H}}_m)(t) := \sup_{X_0} \frac{1}{X_0+1}\left|\frac{d}{dt}\hat{y}_{m,t}\right| \tag{62}$$

$$= \sup_{X_0} \frac{1}{X_0+1}|c_m^\top[\sigma'(h_{m,t})\circ v_{m,t}]| \tag{63}$$

$$\leq \sup_{X_0} \frac{1}{X_0+1}|c_m|_2|\sigma'(h_{m,t})\circ v_{m,t}|_2 \tag{64}$$

$$\leq \sup_{X_0} \frac{1}{X_0+1}|c_m|_2 \cdot \sup_z|\sigma'(z)| \cdot |e^{-\beta_0 t}v_{m,0}|_2 \tag{65}$$

$$\leq \sup_{X_0} \frac{1}{X_0+1}\left(\sup_m|c_m|_2 \cdot \sup_z|\sigma'(z)| \cdot V_0\right)e^{-\beta_0 t} \tag{66}$$

$$\leq \sup_{X_0} \frac{1}{X_0+1}\left(\sup_m|c_m|_2 \cdot L_0 \cdot V_0\right)e^{-\beta_0 t} \tag{67}$$

$$\leq \sup_{X_0}\Bigg(\sup_m|c_m|_2 \cdot L_0 \tag{68}$$

$$\cdot\left(\frac{X_0}{X_0+1}d(\sup_m\|U_m\|_2) + \frac{1}{X_0+1}(\sup_m\|b_m\|_2)\right)\Bigg)e^{-\beta_0 t} \tag{69}$$

$$\leq \left(\sup_m|c_m|_2 \cdot L_0\Big(d\sup_m\|U_m\|_2 + \sup_m\|b_m\|_2\Big)\right)e^{-\beta_0 t} \tag{70}$$

$$\leq (d+1)L_0\theta_{\max}^2 e^{-\beta_0 t} \tag{71}$$

The inequalities are based on vector norm properties, Lipschitz continuity of $\sigma(z)$ and uniform boundedness of unperturbed initial conditions. Therefore we know the model memories are uniformly decaying.

By Lemma C.12, the target $\mathbf{H}$ has an exponentially decaying memory as it is approximated by a sequence of models $\{\widehat{\mathbf{H}}_m\}_{m=1}^\infty$ with uniformly exponentially decaying memory. □

Remark C.1. When the approximation is unstable, we cannot have the real parts of the eigenvalues for recurrent weights bounded away from 0 in Equation (61). As the stability of linear RNNs requires the real parts (of the eigenvalues) to be negative, then the maximum of the real parts will converge to 0. This is the stability boundary of state-space models.

$$\lim_{m\to\infty}\max_{i\in[m]}(\lambda_i(\Lambda_m)) = 0^-. \tag{72}$$

Remark C.2. The uniform weights bound is necessary in the sense that: Since state-space models are universal approximators, they can approximate targets with long-term memories. However, if the target has an non-exponential decaying (e.g. polynomial decaying) memory, the weights bound of the approximation sequence will be exponential in the sequence length T.

$$\theta_{max}^2 \geq e^{\beta_0 T}\frac{\mathcal{M}(\mathbf{H})(T)}{(d+1)L_0}. \tag{73}$$

This result indicates that scaling up SSMs without reparameterization is inefficient in learning sequence relationships with a large T and long-term memory.

Remark C.3 (**On the generalization to multi-layer cases**). We will use the following two-layer state-space models to demonstrate the idea to generalize this result to multi-layer cases.

$$\frac{dh_t}{dt} = \Lambda_1 h_t + U_1 x_t \tag{74}$$

$$y_t = \sigma(h_t) \tag{75}$$

$$\frac{ds_t}{dt} = \Lambda_2 s_t + U_2 y_t \tag{76}$$

$$\hat{z}_t = c^\top\sigma(s_t) \tag{77}$$

We can have the following memory function bounds: For simplicity, we drop the term m in $\Lambda_1, \Lambda_2, U_1, U_2$.

$$\mathcal{M}(\widehat{\mathbf{H}}_m)(t) := \sup_{X_0} \frac{1}{X_0+1} \left| \frac{d}{dt} \hat{z}_{m,t} \right|_2 \tag{78}$$

$$= \sup_{X_0} \frac{1}{X_0+1} \left| c^\top \left(\sigma'(s_{m,t}) \circ \frac{ds_{m,t}}{dt} \right) \right|_2 \tag{79}$$

$$= \sup_{X_0} \frac{1}{X_0+1} \left| c^\top \left(\sigma'(s_{m,t}) \circ \int_0^t e^{\Lambda_2 t_1} U_2 \frac{dy_{m,t-t_1}}{dt} dt_1 \right) \right|_2 \tag{80}$$

$$= \sup_{X_0} \frac{1}{X_0+1} \left| c^\top \left(\sigma'(s_{m,t}) \circ \int_0^t e^{\Lambda_2 t_1} U_2 (\sigma'(h_{m,t-t_1}) \circ v_{m,t-t_1}) dt_1 \right) \right|_2 \tag{81}$$

$$= \sup_{X_0} \frac{1}{X_0+1} \left| c^\top \left(\sigma'(s_{m,t}) \circ \int_0^t e^{\Lambda_2 t_1} U_2 (\sigma'(h_{m,t-t_1}) \circ e^{\Lambda_1 (t-t_1)} v_{m,0}) dt_1 \right) \right|_2 \tag{82}$$

$$\leq \sup_{X_0} \frac{1}{X_0+1} |c|_2 |\sigma'(s_{m,t})|_2 \int_0^t |e^{\Lambda_2 t_1}|_2 |U_2|_2 (|\sigma'(h_{m,t-t_1})|_2 |e^{\Lambda_1 (t-t_1)}|_2 V_0) dt_1 \tag{83}$$

$$\leq L_0^2 \theta_{max}^2 \sup_{X_0} \frac{1}{X_0+1} \int_0^t |e^{\Lambda_2 t_1}|_2 |e^{\Lambda_1 (t-t_1)}|_2 V_0 dt_1 \tag{84}$$

$$\leq L_0^2 \theta_{max}^3 \sup_{X_0} \frac{(dX_0+1)}{X_0+1} \int_0^t |e^{\Lambda_2 t_1}|_2 |e^{\Lambda_1 (t-t_1)}|_2 dt_1 \tag{85}$$

$$\leq L_0^2 \theta_{max}^3 \sup_{X_0} \frac{(dX_0+1)}{X_0+1} \int_0^t |e^{-\beta_0 t_1}|_2 |e^{-\beta_0 (t-t_1)}|_2 dt_1 \tag{86}$$

$$\leq (d+1) L_0^2 \theta_{max}^3 t e^{-\beta_0 t}. \tag{87}$$

The first inequality comes from the Cauchy inequality ($|a \circ b|_2 \leq |a|_2 \cdot |b|_2$). The second inequality comes from the property of activation $\sigma(\cdot)$ and uniform bound on weights. The third inequality comes from the bound of V_0 in Equation (54). The last inequality is the direct evaluation based on the eigenvalues of Λ_1 and Λ_2. As here is a fast decaying term $e^{-\beta_0 t}$, we simplify other polynomial scale components in P.

A further generalization of the memory function for ℓ-layer SSMs would be: For some polynomial $P(t)$ with degree at most $l-1$

$$\mathcal{M}(\widehat{\mathbf{H}}_m)(t) \leq (d+1) L_0^\ell \theta_{max}^{\ell+1} P(t) e^{-\beta_0 t}. \tag{88}$$

C.4. Proof for Theorem 3.5

Proof. Let the target linear functional be $H_t(\mathbf{x}) = \int_{-\infty}^t \rho(t-s) x_s ds$. Here ρ is an L_1 integrable function. We consider a simplified model setting with only parameters c and w. Let c_i, w_i be the unperturbed weights and $\tilde{c}_i, \tilde{w}_i$ be the perturbed recurrent weights. Similar to ρ being L_1 integrable, we note that $\int_0^\infty |c_i e^{f(w_i)t}| dt = \frac{|c_i|}{|f(w_i)|}$. To have a sequence of well-defined model, we require they are uniformly (in m) absolutely integrable:

$$\sup_m \sum_{i=1}^m \frac{|c_i|}{|f(w_i)|} < \infty, \quad \sup_m \sum_{i=1}^m \frac{1}{|f(w_i)|} < \infty. \tag{89}$$

Based $|\tilde{w}-w|_2 \leq \beta$ and $|\tilde{c}-c|_2 \leq \beta$. We know the approximation error is

$$
\begin{aligned}
E_m(\beta) &= \sup_{|\tilde{w}-w|_2\leq\beta,|\tilde{c}-c|_2\leq\beta} \int_0^\infty \left|\sum_{i=1}^m \tilde{c}_i e^{f(\tilde{w}_i)t} - \rho(t)\right| dt && (90)\\
&\leq \sup_{|\tilde{w}-w|_2\leq\beta} \int_0^\infty \left|\sum_{i=1}^m c_i e^{f(w_i)t} - \rho(t)\right| dt && (91)\\
&+ \sup_{|\tilde{w}-w|_2\leq\beta} \int_0^\infty \left|\sum_{i=1}^m c_i e^{f(\tilde{w}_i)t} - \sum_{i=1}^m c_i e^{f(w_i)t}\right| dt && (92)\\
&+ \sup_{|\tilde{w}-w|_2\leq\beta,|\tilde{c}-c|_2\leq\beta} \int_0^\infty \left|\sum_{i=1}^m (\tilde{c}_i - c_i) e^{f(\tilde{w}_i)t}\right| dt && (93)\\
&\leq \sup_{|\tilde{w}-w|_2\leq\beta} \int_0^\infty \left|\sum_{i=1}^m c_i e^{f(w_i)t} - \rho(t)\right| dt && (94)\\
&+ \sup_{|\tilde{w}-w|_2\leq\beta} \int_0^\infty \left|\sum_{i=1}^m c_i e^{f(\tilde{w}_i)t} - \sum_{i=1}^m c_i e^{f(w_i)t}\right| dt && (95)\\
&+ \sup_{|\tilde{w}-w|_2\leq\beta,|\tilde{c}-c|_2\leq\beta} \int_0^\infty \sum_{i=1}^m \beta |e^{f(\tilde{w}_i)t} - e^{f(w_i)t} + e^{f(w_i)t}| dt && (96)\\
&\leq E_m(0) + \sup_{|\tilde{w}-w|_2\leq\beta} \int_0^\infty \sum_{i=1}^m |c_i| \left|e^{f(\tilde{w}_i)t} - e^{f(w_i)t}\right| dt && (97)\\
&+ \sup_{|\tilde{w}-w|_2\leq\beta,|\tilde{c}-c|_2\leq\beta} \int_0^\infty \beta \sum_{i=1}^m |e^{f(\tilde{w}_i)t} - e^{f(w_i)t}| dt + \int_0^\infty \beta \left|\sum_{i=1}^m e^{f(w_i)t}\right| dt && (98)\\
&= E_m(0) + \sup_{|\tilde{w}-w|_2\leq\beta} \int_0^\infty \sum_{i=1}^m (|c_i| + \beta) \left|e^{f(\tilde{w}_i)t} - e^{f(w_i)t}\right| dt && (99)\\
&+ \int_0^\infty \beta \left|\sum_{i=1}^m e^{f(w_i)t}\right| dt && (100)\\
&= E_m(0) + \sum_{i=1}^m (|c_i| + \beta) \sup_{|\tilde{w}-w|_2\leq\beta} \int_0^\infty \left|e^{f(\tilde{w}_i)t} - e^{f(w_i)t}\right| dt + \int_0^\infty \beta \left|\sum_{i=1}^m e^{f(w_i)t}\right| dt && (101)\\
&= E_m(0) + \sum_{i=1}^m (|c_i| + \beta) \sup_{|\tilde{w}_i-w_i|\leq\beta} \int_0^\infty \left|e^{f(\tilde{w}_i)t} - e^{f(w_i)t}\right| dt + \beta \sum_{i=1}^m \frac{1}{|f(w_i)|} && (102)\\
&\leq E_m(0) + \sum_{i=1}^m (|c_i| + \beta) \frac{g(\beta)}{|f(w_i)|} + \beta \sum_{i=1}^m \frac{1}{|f(w_i)|} && (103)\\
&= E_m(0) + \sum_{i=1}^m \frac{g(\beta)(|c_i| + \beta) + \beta}{|f(w_i)|}. && (104)
\end{aligned}
$$

The first and third inequalities are triangular inequality. The second inequality comes from the fact that $|\tilde{w}_i - w_i| \leq |\tilde{w} - w|_2 \leq \beta$. The fourth inequality is achieved via the property of stable reparameterization: For some continuous function $g(\beta) : [0, \infty) \to [0, \infty), g(0) = 0$:

$$
\sup_w \left[|f(w)| \sup_{|\tilde{w}-w|\leq\beta} \int_0^\infty \left|e^{f(\tilde{w})t} - e^{f(w)t}\right| dt \right] \leq g(\beta). \tag{105}
$$

By definition of stable approximation, we know $\lim_{m\to\infty} E_m(0) = 0$. Also according to the requirement of the stable

approximation in Equation (89), we have

$$\lim_{\beta\to 0} E(\beta) = \lim_{\beta\to 0}\lim_{m\to\infty} E_m(\beta) \tag{106}$$

$$\leq \lim_{\beta\to 0}\lim_{m\to\infty} E_m(0) + \left(\sup_m \sum_{i=1}^{m} \frac{|c_i|+\beta}{|f(w_i)|}\right) * \lim_{\beta\to 0} g(\beta) + \lim_{\beta\to 0}\beta * \left(\sup_m \sum_{i=1}^{m} \frac{1}{|f(w_i)|}\right) \tag{107}$$

$$= 0 + 0 + 0 = 0 = E(0). \tag{108}$$

□

Remark C.4. **Here we verify the reparameterization methods satisfy the definition of stable reparameterization.**

For exponential reparameterization $f(w) = -e^w, w \in \mathbb{R}$:

$$\sup_{|\tilde{w}-w|\leq\beta} \int_0^\infty \left| e^{f(\tilde{w})t} - e^{f(w)t} \right| dt = \frac{e^\beta - 1}{|f(w)|}. \tag{109}$$

For softplus reparameterization $f(w) = -\log(1+e^w), w \in \mathbb{R}$: Notice that $\exp(-\beta)\log(1+\exp(w)) \leq \sup_{|\tilde{w}-w|\leq\beta}\log(1+\exp(\tilde{w})) \leq \exp(\beta)\log(1+\exp(w))$,

$$\sup_{|\tilde{w}-w|\leq\beta} \int_0^\infty \left| e^{f(\tilde{w})t} - e^{f(w)t} \right| dt \leq \frac{e^\beta - 1}{|f(w)|}. \tag{110}$$

For "best" reparameterization $f(w) = -\frac{1}{aw^2+b}, w \in \mathbb{R}, a, b > 0$: Without loss of generality, let $w \geq 0$

$$\sup_{|\tilde{w}-w|\leq\beta} \int_0^\infty \left| e^{f(\tilde{w})t} - e^{f(w)t} \right| dt = |a(w+\beta)^2 - aw^2| \tag{111}$$

$$\leq \frac{\frac{a(\beta^2+2\beta w)}{aw^2+b}}{|f(w)|} \tag{112}$$

$$\leq \frac{\frac{a(\beta^2+2\beta w)}{b}}{|f(w)|}. \tag{113}$$

Here $g(\beta) = \frac{a(\beta^2+2\beta w)}{b}$. The famous Müntz–Szász theorem indicates that selecting any non-zero constant a does not affect the universality of linear RNN.

While for the case without reparameterization $f(w) = w, w < 0$: For $0 \leq \beta < -w$,

$$\sup_{|\tilde{w}-w|\leq\beta} \int_0^\infty \left| e^{f(\tilde{w})t} - e^{f(w)t} \right| dt = \frac{\beta}{(-w-\beta)(-w)} = \frac{\beta}{(-w-\beta)|f(w)|}, \tag{114}$$

Here $\lim_{w\to-\beta}\sup_w \frac{\beta}{-w-\beta} = \infty$, therefore the direct parameterization is not a stable reparameterization.

Remark C.5 (On the generalization of existence of stable approximation to nonlinear functionals). The previous results are established for the stable approximation of linear functionals by linear RNNs with stable approximations.

Here we show that this can be further extended to nonlinear functionals. According to the Volterra Series representation, the nonlinear functional has expansion by multi-layer composition or element-wise product (Wang & Xue, 2023). Therefore if the existence of stable approximation is preserved for functional composition and polynomial, then we can generalize the above argument to the nonlinear functionals by working with nonlinear functional representations.

Theorem C.6 (Boyd et al. (1984); Wang & Xue (2023)). *For any continuous time-invariant system with $x(t)$ as input and $y(t)$ as output can be expanded in the Volterra series as follow*

$$y(t) = \rho_0 + \sum_{n=1}^{N} \int_0^t \cdots \int_0^t \rho_n(\tau_1, \ldots, \tau_n) \prod_{j=1}^{n} x(t-\tau_j) d\tau_j. \tag{115}$$

In particular, we call the expansion order N to be the series' order.

Lemma C.7 (Stable approximation induced by polynomials of stable approximation). *Assume* $\mathbf{H}_1$ *and* $\mathbf{H}_2$ *can be stably approximated, let* f *be some polynomial, then* $f(\mathbf{H}_1, \mathbf{H}_2)$ *can also be stably approximated.*

Proof. Let $f(\mathbf{H}_1, \mathbf{H}_2) = \sum_{i,j} c_{i,j} \mathbf{H}_1^i \mathbf{H}_2^j$. The definition of functional product is by the element-wise product: $(\mathbf{H}_1\mathbf{H}_2)(\mathbf{x}) = \mathbf{H}_1(\mathbf{x}) \odot \mathbf{H}_2(\mathbf{x})$.

$$
\begin{aligned}
E_m(\beta) &= \sup_{|\tilde{\theta}-\theta|\le\beta} \|f(\mathbf{H}_1, \mathbf{H}_2) - f(\mathbf{H}_1(\tilde{\theta}), \mathbf{H}_2(\tilde{\theta}))\|_{W^{1,\infty}} && (116)\\
&\le E_m(0) + \sup_{|\tilde{\theta}-\theta|\le\beta} \|f(\mathbf{H}_1(\theta), \mathbf{H}_2(\theta)) - f(\mathbf{H}_1(\tilde{\theta}), \mathbf{H}_2(\tilde{\theta}))\|_{W^{1,\infty}} && (117)\\
&\le E_m(0) + \sup_{|\tilde{\theta}-\theta|\le\beta} \|f(\mathbf{H}_1(\theta), \mathbf{H}_2(\theta)) - f(\mathbf{H}_1(\theta), \mathbf{H}_2(\tilde{\theta}))\|_{W^{1,\infty}} && (118)\\
&\quad + \sup_{|\tilde{\theta}-\theta|\le\beta} \|f(\mathbf{H}_1(\theta), \mathbf{H}_2(\tilde{\theta})) - f(\mathbf{H}_1(\tilde{\theta}), \mathbf{H}_2(\tilde{\theta}))\|_{W^{1,\infty}} && (119)\\
&\le E_m(0) + \sum_{i\ge0, j\ge1} c_{i,j} j \|\widehat{\mathbf{H}}_1(\theta)\|_{W^{1,\infty}}^i (\|\widehat{\mathbf{H}}_2(\theta)\|_{W^{1,\infty}} + E_m^{\mathbf{H}_2}(\beta))^{j-1} E_m^{\mathbf{H}_2}(\beta) && (120)\\
&\quad + \sum_{i\ge1, j\ge0} c_{i,j} i (\|\widehat{\mathbf{H}}_1(\theta)\|_{W^{1,\infty}} + E_m^{\mathbf{H}_1}(\beta))^{i-1} \|\widehat{\mathbf{H}}_2(\theta)\|_{W^{1,\infty}}^j E_m^{\mathbf{H}_1}(\beta). && (121)
\end{aligned}
$$

Therefore $E(\beta) \le \lim_{m\to\infty} E_m(\beta) < \infty$. The third inequality comes from Equation (29). □

C.5. Proof for Theorem 3.6

Proof. For any $1 \le j \le m$, assume the loss function we used is the L_∞ norm: Loss $= \sup_t \|H_t - \widehat{H}_{m,t}\|_\infty$. Notice that by time-homogeneity, Loss $= \|H_t - \widehat{H}_{m,t}\|_\infty$ for any t. This loss function is larger than the common mean squared error, which is usually chosen in practice for the smoothness reason.

$$
\begin{aligned}
\left|\frac{\partial \text{Loss}}{\partial w_j}\right| &= \left|\frac{\partial \|H_t - \widehat{H}_{m,t}\|_\infty}{\partial w_j}\right| && (122)\\
&= \left|\frac{\partial \sup_{\|\mathbf{x}\|_\infty\le1} |H_t(\mathbf{x}) - \widehat{H}_{m,t}(\mathbf{x})|}{\partial w_j}\right| && (123)\\
&= \left|\frac{\partial \sup_{\|\mathbf{x}\|_\infty\le1} |\int_{-\infty}^t (\rho(t-s) - \sum_{i=1}^m c_i e^{-f(w_i)(t-s)}) x_s ds|}{\partial w_j}\right| && (124)\\
&= \left|\frac{\partial \int_{-\infty}^t |\rho(t-s) - \sum_{i=1}^m c_i e^{-f(w_i)(t-s)}| ds}{\partial w_j}\right| && (125)\\
&= \left|\frac{\partial \int_{-\infty}^t |(\rho(t-s) - \sum_{i\ne j} c_i e^{-f(w_i)(t-s)}) - c_j e^{-f(w_j)(t-s)}| ds}{\partial w_j}\right| && (126)\\
&= \left|\frac{\partial \int_{0}^\infty |(\rho(s) - \sum_{i\ne j} c_i e^{-f(w_i)s}) - c_j e^{-f(w_j)s}| ds}{\partial w_j}\right| && (127)\\
&\le \int_0^\infty \left|\frac{\partial |(\rho(s) - \sum_{i\ne j} c_i e^{-f(w_i)s}) - c_j e^{-f(w_j)s}|}{\partial w_j}\right| ds && (128)\\
&\le \int_0^\infty \left|\frac{\partial |c_j e^{-f(w_j)s}|}{\partial w_j}\right| ds && (129)
\end{aligned}
$$

The first equality is the definition of the loss function. The second equality equality comes from the definition of the linear functional norm. The third equality expand the linear functional and linear RNNs into the convolution form. The fourth

equality utilize the fact that we can manually select x_t's sign to achieve the maximum value. The fifth equality is separating the term in dependent of variable w_j. The sixth equality is change of variable from $t-s$ to s. The inequality is triangular inequality. The last equality is dropping the term independent of variable w_j.

$$\left|\frac{\partial \text{Loss}}{\partial w_j}\right| \leq \int_0^\infty \left|\frac{\partial |c_j e^{-f(w_j)s}|}{\partial w_j}\right| ds \tag{130}$$

$$= |c_j f'(w_j)| \int_0^\infty e^{-f(w_j)s} s \, ds \tag{131}$$

$$= \left|c_j \frac{f'(w_j)}{f(w_j)}\right| \int_0^\infty e^{-f(w_j)s} ds \tag{132}$$

$$= \left|c_j \frac{f'(w_j)}{f(w_j)^2}\right| (1 - \lim_{s\to\infty} e^{-f(w_j)s}) = \left|c_j \frac{f'(w_j)}{f(w_j)^2}\right|. \tag{133}$$

The first equality is evaluating the derivative. The second equality is extracting $|f'(w)|$ from integral. The third equality is doing the integration by parts.

In particular, notice that c_j is a constant independent of the recurrent weight parameterization f:

$$\widehat{H}_{m,t}(\mathbf{x}) = \int_{-\infty}^t \sum_{i=1}^m c_i e^{-f(w_i)(t-s)} x_s ds. \tag{134}$$

Therefore c_j is a parameterization indepndent value, we will denote it by $C_{\mathbf{H},\widehat{\mathbf{H}}_m}$.

Moreover, in the discrete setting, assume $h_{k+1} = f(w) \circ h_k + U x_k$,

$$\left|\frac{\partial \text{Loss}}{\partial w_j}\right| \leq \sum_{k=0}^\infty \left|\frac{\partial |c_j f(w_j)^k|}{\partial w_j}\right| ds \tag{135}$$

$$= |c_j f'(w_j)| \sum_{k=1}^\infty k f(w_j)^{k-1} \tag{136}$$

$$= |c_j f'(w_j)| \left(\sum_{k=1}^\infty f(w_j)^{k-1}\right)^2 \tag{137}$$

$$= \left|c_j \frac{f'(w_j)}{(1-f(w_j))^2}\right|. \tag{138}$$

So the gradient norm is bounded by

$$\left|\frac{\partial \text{Loss}}{\partial w_j}\right| = \frac{|c_j f'(w_j)|}{(1-f(w_j))^2}. \tag{139}$$

□

Nonlinear functionals Now we show the generalization into the nonlinear functional: Consider the Volterra Series representation of the nonlinear functional.

Theorem C.8 ((Boyd et al., 1984)). *For any continuous time-invariant system with $x(t)$ as input and $y(t)$ as output can be expanded in the Volterra series as follow*

$$y(t) = h_0 + \sum_{n=1}^N \int_0^t \cdots \int_0^t h_n(\tau_1, \ldots, \tau_n) \prod_{j=1}^n x(t-\tau_j) d\tau_j. \tag{140}$$

Here N is the series' order. Linear functional is an order-1 Volterra series.

For simplicity, we will only discuss the case for $N = 2$. When we take the Hyena approach (Poli et al., 2023) and approximate the order-2 kernel $h_2(\tau_1, \tau_2)$ with its rank-1 approximation:

$$h_2(\tau_1, \tau_2) = h_{2,1}(\tau_1)h_{2,2}(\tau_2). \tag{141}$$

Here $h_{2,1}$ and $h_{2,2}$ are again order-1 kernel which can be approximated with linear RNN's kernel. In other words, the same gradient bound also holds for general nonlinear functional with the following form:

$$G_f(w) := \left|\frac{\partial E}{\partial w}\right| = C_{\mathbf{H},\widehat{\mathbf{H}}_m} \frac{|f'(w)|}{f(w)^2}. \tag{142}$$

And the discrete version is

$$G_f^D(w) := \left|\frac{\partial E}{\partial w}\right| = C_{\mathbf{H},\widehat{\mathbf{H}}_m} \frac{|f'(w)|}{(1-f(w))^2}. \tag{143}$$

C.6. Lemmas

Lemma C.9. *If the activation $\sigma(\cdot)$ is bounded, strictly increasing, continuously differentiable function over $\mathbb{R}$. Then for all $C > 0$, there exists ϵ_C such that $\forall |z| \le C_\epsilon$, $|\sigma'(z)| \ge \epsilon_C$.*

Proof. Since $\sigma(\cdot)$ is monotonically increasing, therefore $\sigma'(\cdot) > 0, \forall z \ge 0$. Notice that $\sigma'(\cdot)$ is continuous, for any $C > 0$, we know $\frac{1}{2}\min_{|z|\le C}\sigma'(z) > 0$. Define $\epsilon_C := \frac{1}{2}\min_{|z|\le C}\sigma'(z) > 0$, it can be seen the target statement is satisfied. □

Lemma C.10. *Assume the target functional sequence has a β_0-stable approximation and the perturbed model has a decaying memory, we show that $\tilde{v}_{m,t} \to 0$ for all m.*

Proof. For any m, fix $\widetilde{\Lambda}_m$ and $\widetilde{U}_m$. Since the perturbed model has a decaying memory,

$$\lim_{t\to\infty}\left|\frac{d}{dt}\widetilde{H}_m(\mathbf{u}^x)\right| = \lim_{t\to\infty}\left|c^\top(\sigma'(\tilde{h}_{m,t})\circ\frac{d\tilde{h}_{m,t}}{dt})\right| = \lim_{t\to\infty}\left|c^\top(\sigma'(\tilde{h}_{m,t})\circ\tilde{v}_{m,t})\right| = 0. \tag{144}$$

By linear algebra, there exist m vectors $\{\Delta c_i\}_{i=1}^m$, $|\Delta c_i|_\infty < \beta$ such that $c_m + \Delta c_1, \ldots, c_m + \Delta c_m$ form a basis of $\mathbb{R}^m$. We can then decompose any vector u into

$$u = k_1(c_m + \Delta c_1) + \cdots + k_m(c_m + \Delta c_m). \tag{145}$$

Take the inner product of u and $\tilde{v}_{m,t}$, we have

$$\lim_{t\to\infty} u^\top(\sigma'(\tilde{h}_{m,t})\circ\tilde{v}_{m,t}) = \sum_{i=1}^m k_i \lim_{t\to\infty}(c_m + \Delta c_i)^\top(\sigma'(\tilde{h}_{m,t})\circ\tilde{v}_{m,t}) = 0 \tag{146}$$

As the above result holds for any vector u, we get

$$\lim_{t\to\infty}\left|\sigma'(\tilde{h}_{m,t})\circ\tilde{v}_{m,t}\right|_\infty = 0. \tag{147}$$

As required in Equation (8), the hidden states are uniformly (in m) bounded over bounded input sequence. There exists constant $C_0 > 0$ such that

$$\sup_{m,t}|h_{m,t}|_\infty < C_0. \tag{148}$$

Since σ is continuously differentiable and strictly increasing, by Lemma C.9, there exists $\epsilon_{C_0} > 0$ such that

$$|\sigma'(z)| > \epsilon_{C_0}, \quad \forall |z| \le C_0. \tag{149}$$

Therefore

$$\sup_t\left|\sigma'(\tilde{h}_{m,t})\right|_\infty > \epsilon_{C_0}. \tag{150}$$

We get

$$\lim_{t\to\infty}|\tilde{v}_{m,t}|_\infty = 0. \tag{151}$$

□

Lemma C.11. *Consider a dynamical system with the following dynamics:* $h_0 = 0$

$$\begin{aligned} \frac{dv_t}{dt} &= \Lambda v_t, \\ v_0 &= \Lambda h_0 + \widetilde{U} x_0 + \tilde{b} = \widetilde{U} x_0 + \tilde{b}. \end{aligned} \tag{152}$$

If $\Lambda \in \mathbb{R}^{m\times m}$ *is diagonal, hyperbolic and the system in Equation* (152) *is satisfies* $\lim_{t\to\infty} v_t = 0$ *over any bounded Heaviside input* $\mathbf{u}^{x_0}, |x_0|_\infty < \infty$*, then the matrix* Λ *is Hurwitz.*

Proof. By integration we have the following explicit form:

$$v_t = e^{\Lambda t} v_0 = e^{\Lambda t}(\widetilde{U} x_0 + \tilde{b}). \tag{153}$$

The stability requires $\lim_{t\to\infty} |v_t| = 0$ for all inputs $v_0 = \widetilde{U} x_0 + \tilde{b}$. Notice that with perturbation from $\tilde{U}$ and $\tilde{b}$, the set of initial points $\{v_0\}$ is m-dimensional. Therefore the matrix Λ is Hurwitz in the sense that all eigenvalues' real parts are negative. □

Lemma C.12. *Consider a continuous function* $f : [0, \infty) \to \mathbb{R}$*, assume it can be approximated by a sequence of continuous functions* $\{f_m\}_{m=1}^{\infty}$ *universally:*

$$\lim_{m\to\infty} \sup_{t\geq 0} |f(t) - f_m(t)| = 0. \tag{154}$$

Assume the approximators f_m *are uniformly exponentially decaying with the same* $\beta_0 > 0$*:*

$$\lim_{t\to\infty} \sup_{m\in\mathbb{N}_+} e^{\beta_0 t} |f_m(t)| \to 0. \tag{155}$$

Then the function f *is also decaying exponentially:*

$$\lim_{t\to\infty} e^{\beta t} |f(t)| \to 0, \quad \forall 0 < \beta < \beta_0. \tag{156}$$

The proof is the same as Lemma A.11 from (Wang et al., 2023). For completeness purpose, we attach the proof here:

Proof. Given a function $f \in C([0, \infty))$, we consider the transformation $\mathcal{T} f : [0, 1] \to \mathbb{R}$ defined as:

$$(\mathcal{T} f)(s) = \begin{cases} 0, & s = 0 \\ \frac{f(-\frac{\log s}{\beta_0})}{s}, & s \in (0, 1]. \end{cases} \tag{157}$$

Under the change of variables $s = e^{-\beta_0 t}$, we have:

$$f(t) = e^{-\beta_0 t}(\mathcal{T} f)(e^{-\beta_0 t}), \quad t \geq 0. \tag{158}$$

According to uniformly exponentially decaying assumptions on f_m:

$$\lim_{s\to 0^+} (\mathcal{T} f_m)(s) = \lim_{t\to\infty} \frac{f_m(t)}{e^{-\beta_0 t}} = \lim_{t\to\infty} e^{\beta_0 t} f_m(t) = 0, \tag{159}$$

which implies $\mathcal{T} f_m \in C([0, 1])$.

For any $\beta < \beta_0$, let $\delta = \beta_0 - \beta > 0$. Next we have the following estimate

$$\sup_{s\in[0,1]} |(\mathcal{T} f_{m_1})(s) - (\mathcal{T} f_{m_2})(s)| \tag{160}$$

$$= \sup_{t\geq 0} \left| \frac{f_{m_1}(t)}{e^{-\beta t}} - \frac{f_{m_2}(t)}{e^{-\beta t}} \right| \tag{161}$$

$$\leq \max \left\{ \sup_{0\leq t\leq T_0} \left| \frac{f_{m_1}(t)}{e^{-\beta t}} - \frac{f_{m_2}(t)}{e^{-\beta t}} \right|, C_0 e^{-\delta T_0} \right\} \tag{162}$$

$$\leq \max \left\{ e^{\beta T_0} \sup_{0\leq t\leq T_0} |f_{m_1}(t) - f_{m_2}(t)|, C_0 e^{-\delta T_0} \right\} \tag{163}$$

where C_0 is a constant uniform in m.

For any $\epsilon > 0$, take $T_0 = -\frac{\ln(\frac{\epsilon}{C_0})}{\delta}$, we have $C_0 e^{-\delta T_0} \leq \epsilon$. For sufficiently large M which depends on ϵ and T_0, by universal approximation (Equation (154)), we have $\forall m_1, m_2 \geq M$,

$$\sup_{0 \leq t \leq T_0} |f_{m_1}(t) - f_{m_2}(t)| \leq e^{-\beta T_0} \epsilon, \tag{164}$$

$$e^{\beta T_0} \sup_{0 \leq t \leq T_0} |f_{m_1}(t) - f_{m_2}(t)| \leq \epsilon. \tag{165}$$

Therefore, $\{f_m\}$ is a Cauchy sequence in $C([0, \infty))$.

Since $\{f_m\}$ is a Cauchy sequence in $C([0, \infty))$ equipped with the sup-norm, using the above estimate we can have$\{\mathcal{T} f_m\}$ is a Cauchy sequence in $C([0, 1])$ equipped with the sup-norm. By the completeness of $C([0, 1])$, there exists $f^* \in C([0, 1])$ with $f^*(0) = 0$ such that

$$\lim_{m \to \infty} \sup_{s \in [0,1]} |(\mathcal{T} f_m)(s) - f^*(s)| = 0. \tag{166}$$

Given any $s > 0$, we have

$$f^*(s) = \lim_{m \to \infty} (\mathcal{T} f_m)(s) = (\mathcal{T} f)(s), \tag{167}$$

hence

$$\lim_{t \to \infty} e^{\beta t} f(t) = \lim_{s \to 0^+} (\mathcal{T} f)(s) = f^*(0) = 0. \tag{168}$$

□

D. Motivation for the gradient-over-weight Lipschitz criterion

Here we discuss the motivation for adopting the gradient-over-weight boundedness as the criterion for "best-in-stability" reparameterization. First of all, the "best" reparameterization is proposed to further improve the optimization stability across memory patterns with different decays. The criterion "gradient is Lipschitz to the weight" is a necessary condition for the stability in the following sense:

1. Consider functions $f(x) = x^4$, the gradient function $\frac{df}{dx}(x) = 4x^3$ does not have a global Lipschitz coefficient for all input values x. Therefore for any fixed positive learning rate η, there exists an initial point x_0 (for example $x_0 = \sqrt{\frac{1}{2\eta}} + 1$) such that the convergence from initial point x_0 cannot be achieved via the gradient descent step

   $$x_{k+1} = x_k - \eta g(x_k). \tag{169}$$

   It can be verified the convergence does not hold as $|x_{k+1}| > |x_k|$ for all k when $x_0 = \sqrt{\frac{1}{2\eta}} + 1$. This comes from the fact that $|x_k| \geq \sqrt{\frac{1}{2\eta}}, \eta g(x) \geq 2x_k$ hold for all k.

2. Consider functions $f(x) = x^2$, the gradient function $g(x) = 2x$ is associated with a Lipschitz constant $L = 2$. Then the same gradient descent step converges for any $\eta \leq \frac{1}{2}$ in Equation (169).

3. As can be seen in the above two examples, **the criterion "gradient is Lipschitz to the weight" is associated with the convergence under large learning rate.** As the use of larger learning rate is usually associated with faster convergence (Smith & Topin, 2019), smaller generalization errors (Li et al., 2019), we believe the Lipschitz criterion is a suitable stability criterion for the measure of optimization stability.

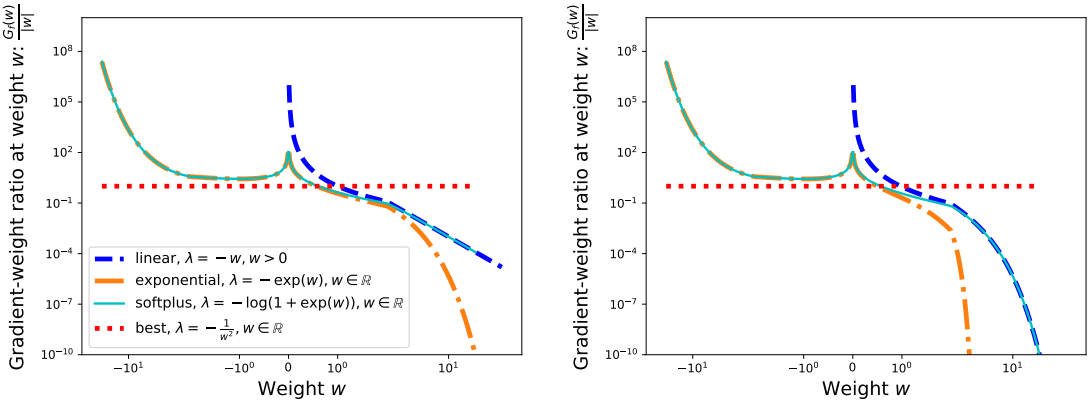
4. The gradient-over-weight ratio evaluated in Figure 4(a) is a numerical verification of our Theorem 3.4. The gradients of stable reparameterizations are less susceptible to the well-known issue of exploding or vanishing gradients (Bengio et al., 1994; Hochreiter, 1998).

E. Comparison of different recurrent weights parameterization schemes

Here we evaluate the gradient norm bound function G_f and G_f^D for different parameterization schemes in Table 4 and Figure 6.

Table 4. Summary of reparameterizations and corresponding gradient norm functions in continuous and discrete time. Notice that the G_f and G_f^D are rescaled up to a constant $C_{\mathbf{H},\widehat{\mathbf{H}}}$.

	Reparameteriations	f	G_f or G_f^D
Continuous	ReLU	$-\mathrm{ReLU}(w)$	$\frac{1}{w^2}\mathbf{1}_{\{w>0\}}$
	Exp	$-\exp(w)$	e^{-w}
	Softplus	$-\log(1+\exp(w))$	$\frac{\exp(w)}{(1+\exp(w))\log(1+\exp(w))^2}$
	"Best"(Ours)	$-\frac{1}{aw^2+b}, a>0, b>0$	$2a\lvert w\rvert$
Discrete	ReLU	$\exp(-\mathrm{ReLU}(w))$	$\frac{\exp(-w)}{(1-\exp(-w))^2}\mathbf{1}_{\{w>0\}}$
	Exp	$\exp(-\exp(w))$	$\frac{\exp(w-\exp(w))}{(1-\exp(-\exp(w)))^2}$
	Softplus	$\frac{1}{1+\exp(w)}$	e^{-w}
	Tanh	$\tanh(w)=\frac{e^{2w}-1}{e^{2w}+1}$	e^{2w}
	"Best"(Ours)	$1-\frac{1}{w^2+0.5}\in(-1,1)$	$2\lvert w\rvert$



(a) Continuous time $\frac{G_f(w)}{|w|}$ (b) Discrete time $\frac{G_f^D(w)}{|w|}$

Figure 6. Gradient norm function G_f and G_f^D of different parameterization methods. The "best" parameterization methods maintain a balanced gradient-over-weight ratio.

F. Numerical details

In this section, the details of numerical experiments are provided for the completeness and reproducibility.

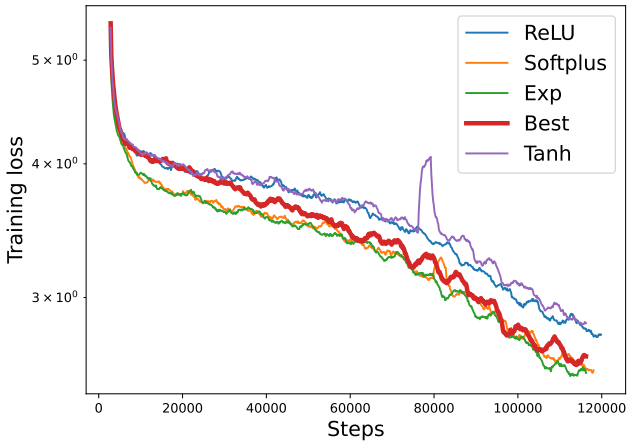
F.1. Synthetic task

We conduct the approximation of linear functional with linear RNNs in the one-dimensional input and one-dimensional output case. The synthetic linear functional is constructed with the polynomial decaying memory function is $\rho(t)=\frac{1}{(t+1)^{1.1}}$. Sequence length is 100. Total number of synthetic samples is 153600. The learning rate used is 0.01 and the batch size is 512.

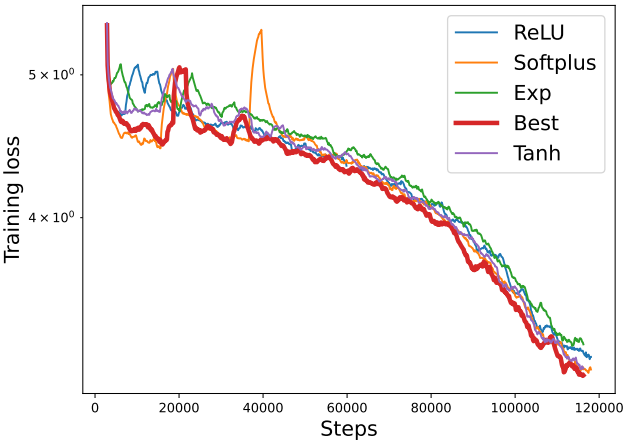
The perturbation list $\beta\in[0,10^{-3},10^{-3}*2^{1/2},10^{-3}*2^{2/2},\ldots,10^{-3}*2^{20/2}]$. Each evaluation of the perturbed error is sampled with 30 different weight perturbations to reduce the variance.

F.2. Language models

The language modeling is done over WikiText-103 dataset (Merity et al., 2016). The model we used is based on the Hyena architecture with simple real-weights state-space models as the mixer (Poli et al., 2023; Smith et al., 2023). The batch size is 16, total steps 115200 (around 16 epochs), warmup steps 1000. The optimizer used is AdamW and the weight decay coefficient is 0.25. The learning rate for the recurrent layer is 0.004 while the learning rate for other layers are 0.005.



(a) lr=0.002, "best" reparameterization is also not optimal, but the final loss is comparable against Exp and Softplus



(b) lr=0.01, "best" reparameterization achieve the smallest loss

Figure 7. The stability advantage of "best" reparameterization (red line) is usually better when the learning rate is larger.

In the main paper, we provide the training loss curve for learning rate = 0.005 as the stability of "best" discrete-time parameterization $f(w) = 1 - \frac{1}{w^2+0.5}$ is mostly significant as the learning rate is large. In Figure 7, we further provide the results for other learning rates (lr = 0.002, 0.010). Despite the final loss not being optimal for the "best" reparameterization, it is observed that the training process exhibits enhanced stability compared to other parameterization methods.

F.3. On the stability of "best" reparameterization for large models

The previous experiment on WikiText-103 language modelling shows the performance of stable reparameterization over the unstable cases. We further verify the optimization stability of "best" reparameterization in the following extreme setting. We construct a large scale language model with 3B parameters and train with larger learning rate (lr=0.01). As can be seen in the following table, the only convergent model is the model with "best" reparameterization. We emphasize that the only difference between these models are the parameterization schemes for recurrent weights. Therefore the best reparameterization is the most **stable** parameterization. (We repeats the experiments with different seeds for three times.)

F.4. Additional numerical results for associative recalls

In this section, we study the performance of of different stable reparameterizations over the extremely long sequences (up to 131k). It can be seen in Table 6 that stable parameterizations are better than the case without reparameterization and simple

	"Best"	Exp	Softplus	Direct
Convergent / total experiments	3/3	0/3	0/3	0/3

Table 5. Experiment to the stability of "best" reparameterization over lr = 0.01. All other reparameterizations diverged within 100 steps while the "best" reparameterizations can be used to train the model.

clipping. The advantage is more significant when the sequence length is longer. The models are trained under the exactly same hyperparameters.

Reparameterizations	Train acc, T=20	Test acc, T=20	Train acc, T=131k	Test acc, T=131k
"Best"	**57.95**	**99.8**	**53.57**	**100**
Exp(S5)	54.55	**99.8**	**53.57**	**100**
Clip	50.0	76.6	13.91	9.4
Direct	43.18	67.0	16.59	5.6

Table 6. Comparison of parameterizations on associative recalls. The first two columns are the train and test accuracy over **sequence length 20**, vocabulary size 10, while the second two columns are the train and test accuracy over **sequence length 131k** and vocabulary size 30.