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

Shida Wang & Qianxiao Li

ICML 2024

OptiML 2024 Summer Workshop on State-space Model Theory

Presenter: Hanseul Cho

Optimization & Machine Learning Laboratory

Main References

- Shida Wang and Qianxiao Li. **StableSSM: Alleviating the Curse of Memory in State-space Models through Stable Reparameterization.** ICML 2024. URL.
- Zhong Li, Jiequn Han, Weinan E, and Qianxiao Li. **On the Curse of Memory in Recurrent Neural Networks: Approximation and Optimization Analysis.** ICLR 2021. URL.
- Shida Wang, Zhong Li, and Qianxiao Li. **Inverse Approximation Theory for Nonlinear Recurrent Neural Networks.** ICLR 2024. URL.
- "S4 model": Albert Gu, Karan Goel, and Christopher Ré. **Efficiently Modeling Long Sequences with Structured State Spaces.** ICLR 2022. URL.

RNN's exponentially decaying memory

- Linear RNN (with $h_{-1} = 0$):

$$
\begin{aligned}
h_t &= W h_{t-1} + U x_t \\
&= {\color{red} W^t U x_0} + W^{t-1} U x_1 + \cdots + U x_t
\end{aligned}
$$

- The weight associated with x_0 (may) decay exponentially fast.
- Difficult to approximate or optimize to learn long-term memory

➡ **"Curse of Memory"**

- Non-linear RNNs have the same problem (Wang et al., 2024)

Alternatives Using State-space Models (SSMs)

- E.g., S4, S5, LRU, RWKV, RetNet, Mamba🐍(S6), …
  - Based on the idea of RNN; computational efficiency is much improved (e.g., parallelism of training)
- Are they liberated from the Curse of Memory?
  - **No**, without proper manipulation (called **‘stable’ reparameterizations**)
  - In the sense of both **stable approximation** and **stable optimization**

Overview

1. SSMs without reparameterization can only **stably approximate** targets with exponentially decaying memory.

2. With stable reparameterizations, SSMs can achieve a **stable approximation** of any well-defined targets of sequence modeling.

3. Regarding the gradient-over-weight scale ($=\frac{|\text{gradient}|}{|\text{weight}|}$) as a criterion of **optimization stability**, the "best" reparameterization in terms of optimization stability is derived.

Contents

State-space Models (SSMs) of Our Interest

Continuous-time & Diagonal Λ

- An SSM maps d-dimensional inputs $\mathbf{x} = \{x_t\}$ to 1-dimensional outputs $\{\hat{y}_t\}$:

$$\frac{\mathrm{d}h_t}{\mathrm{d}t} = \Lambda h_t + U x_t, \qquad h_{-\infty} = 0,$$
$$\hat{y}_t = c^\top \sigma(h_t), \qquad t \in \mathbb{R}.$$

- Trainable: $\Lambda = \mathrm{Diag}(\lambda_1, \ldots, \lambda_m) \in \mathbb{R}^{m \times m}, U \in \mathbb{R}^{m \times d}, c \in \mathbb{R}^m$
- Activation function $\sigma(\cdot)$
- Solution: $\hat{y}_t = c^\top \sigma\left(\int_0^\infty e^{s\Lambda} U x_{t-s} \mathrm{d}s\right) = c^\top \sigma\left(\int_0^\infty \mathrm{Diag}(e^{\lambda_1 s}, \ldots, e^{\lambda_m s}) U x_{t-s} \mathrm{d}s\right).$
- We can stack SSMs by ℓ-layers. ($h_t^{(1)} \to \cdots \to h_t^{(\ell)} \to \hat{y}_t$, by $\frac{\mathrm{d}h_t^{(l)}}{\mathrm{d}t} = \Lambda_l h_t^{(l)} + U_l \sigma(h_t^{(l-1)})$)

Sequence Modeling = Functional Approximations

- Input seq.: $\mathbf{x} = \{x_t\} \in \mathfrak{X} = C_0(\mathbb{R}, \mathbb{R}^d)$ with norm $\|\mathbf{x}\|_\infty = \sup_{t \in \mathbb{R}} |x_t|_\infty$
- Functional sequence: $\mathbf{H} = \{H_t : \mathfrak{X} \to \mathbb{R} \mid t \in \mathbb{R}\}$ (each element is a "functional")
- Output sequence: $\mathbf{y} = \mathbf{H}(\mathbf{x})$ meaning that $y_t = H_t(\mathbf{x})$
- We want to approximate $\mathbf{H}$ with our model (another functional sequence) $\hat{\mathbf{H}}$.
- H is a linear functional if $H(c\mathbf{x} + c'\mathbf{x}') = cH(\mathbf{x}) + c'H(\mathbf{x}')$.
  - We are interested in more general, non-linear functional sequences

Memory Function of a Functional sequence

Motivation from linear functionals

- (Lemma A.1 of Li et al. (2020)). Let $\mathbf{H} = \{H_t\}$ be a sequence of continuous, **linear**, regular, causal, and time-homogeneous functional on $C_0(\mathbb{R}, \mathbb{R}^d)$. Then, there exists a vector-valued integrable function $\rho : [0,\infty) \to \mathbb{R}^d$ such that

$$H_t(\mathbf{x}) = \mathbf{x} * \rho := \int_0^\infty x_{t-s}^\top \rho(s) ds.$$

  In particular, H_t is a bounded functional.

- The function ρ represents the memory of $\mathbf{H}$
  - if ρ decays fast, it forgets inputs far away from time t

Memory Function of a Functional sequence

Motivation from linear functional sequence

- Quantifying the memory of $\mathbf{H} = \{H_t\}$:

$$
\begin{aligned}
|\rho(t)|_1 &= \sup_{x \in \mathbb{R}^d \setminus \{0\}} \frac{|x^\top \rho(t)|}{|x|_\infty} \\
&= \sup_{x \in \mathbb{R}^d \setminus \{0\}} \frac{|\int_0^\infty x^\top \rho(s) \delta(t-s) \mathrm{d}s|}{|x|_\infty} \\
&= \sup_{x \in \mathbb{R}^d \setminus \{0\}} \frac{|\frac{\mathrm{d}}{\mathrm{d}t} \int_0^\infty (x \cdot \mathbf{1}_{t-s \geq 0})^\top \rho(s) \mathrm{d}s|}{|x|_\infty} \\
&= \sup_{x \in \mathbb{R}^d \setminus \{0\}} \frac{|\frac{\mathrm{d}}{\mathrm{d}t} H_t(\mathbf{u}^x)|}{|x|_\infty}
\end{aligned}
$$

where $\mathbf{u}^x(t) = x \cdot \mathbf{1}_{t \geq 0}$

Memory Function of a Functional sequence

Generalized notion for non-linear functionals

- **Definition 2.1 (Memory Function).** For a bounded, continuous, regular, causal, time-homogeneous, and non-linear functional sequence $\mathbf{H}$ on $C_0(\mathbb{R}, \mathbb{R}^d)$, the memory function of $\mathbf{H}$ is defined as

$$\mathfrak{M}(\mathbf{H})(t) := \sup_{x \in \mathbb{R}^d \setminus \{0\}} \frac{\left| \frac{\mathrm{d}}{\mathrm{d}t} H_t(\mathbf{u}^x) \right|}{|x|_\infty + 1} \quad \text{where } \mathbf{u}^x(t) = x \cdot \mathbf{1}_{t \geq 0}$$

- **Definition 2.2 (Decaying Memory).** The functional sequence $\mathbf{H}$ has a decaying memory if $\lim_{t \to \infty} \mathfrak{M}(\mathbf{H})(t) = 0$.
  - We say the decay is exponential if $\lim_{t \to \infty} e^{\beta t} \mathfrak{M}(\mathbf{H})(t) = 0$ for some $\beta > 0$. (Fast!)
- The slow decay memory function is a necessary condition to build a model with long-term memory.

Stable Approximation of Functional Sequence

- Functional norm: $\|H\|_\infty := \sup_{\mathbf{x}\neq 0} \frac{|H(\mathbf{x})|}{|\mathbf{x}|_\infty + 1} + |H(\mathbf{0})|$
- **Definition 2.4.** Sobolev-type functional sequence norm: $\left\| \mathbf{H} \right\|_{W^{1,\infty}} := \sup_t \left(\|H_t\|_\infty + \left\| \frac{\mathrm{d}H_t}{\mathrm{d}t} \right\|_\infty \right)$
- **Definition 2.5.** A sequence of parameterized models $\{\hat{\mathbf{H}}(\cdot\,;\theta_m)\}_{m=1}^\infty$ is said to be β_0**-stably approximating** the target functional sequence $\mathbf{H}$ if

  1. $E(0) = 0$, (Exact and Strong Universal Approximation)
  2. $E(\beta)$ is continuous for $\beta \in [0, \beta_0]$, (Robustness of Approximation)

  where $E(\beta) = \limsup_{m\to\infty} E_m(\beta)$ and

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

Contents

Curse of Memory in Vanilla SSMs

Theorem 3.3

- Assume $\mathbf{H}$ is a sequence of bounded, continuous, regular, causal, time-homogeneous functionals on $\mathfrak{X} = C_0(\mathbb{R}, \mathbb{R}^d)$ with decaying memory.

- Suppose a sequence of ℓ-layer SSMs $\{\hat{\mathbf{H}}(\,\cdot\,;\theta_m)\}_{m=1}^{\infty}$ (with hidden dimensions m, uniformly bounded weights ($\theta_{\max} := \sup_m |\theta_m|_2 < \infty$), and a strictly increasing L-Lipschitz activation function σ) is β_0-stably approximating $\mathbf{H}$.

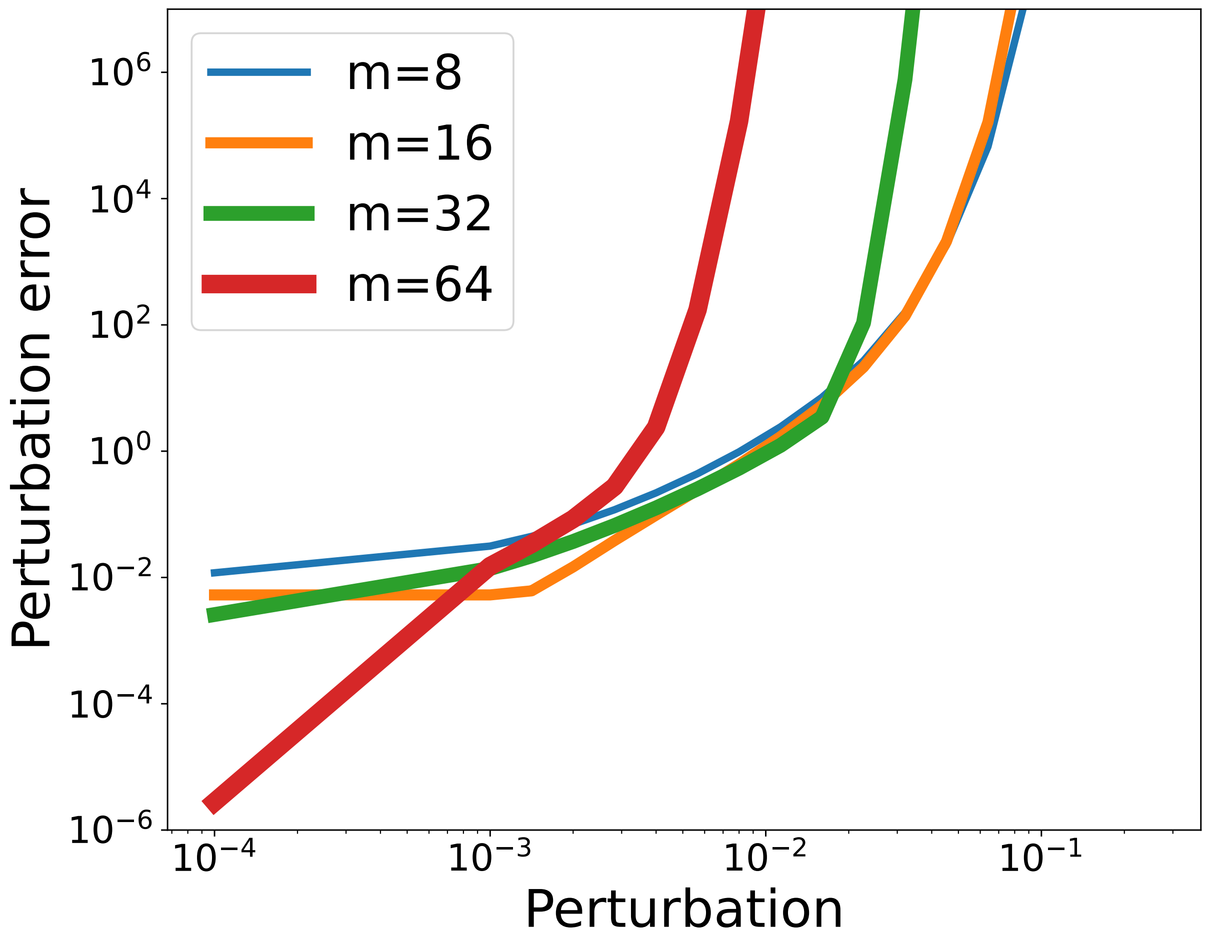
- Then for a nonnegative integer $k \leq \ell - 1$,

$$\mathfrak{M}(\mathbf{H})(t) \leq O\left((d+1)L^{\ell}\theta_{\max}^{\ell+1}t^k e^{-\beta_0 t}\right).$$

- In particular, the memory decays exponentially if $\ell = 1$.

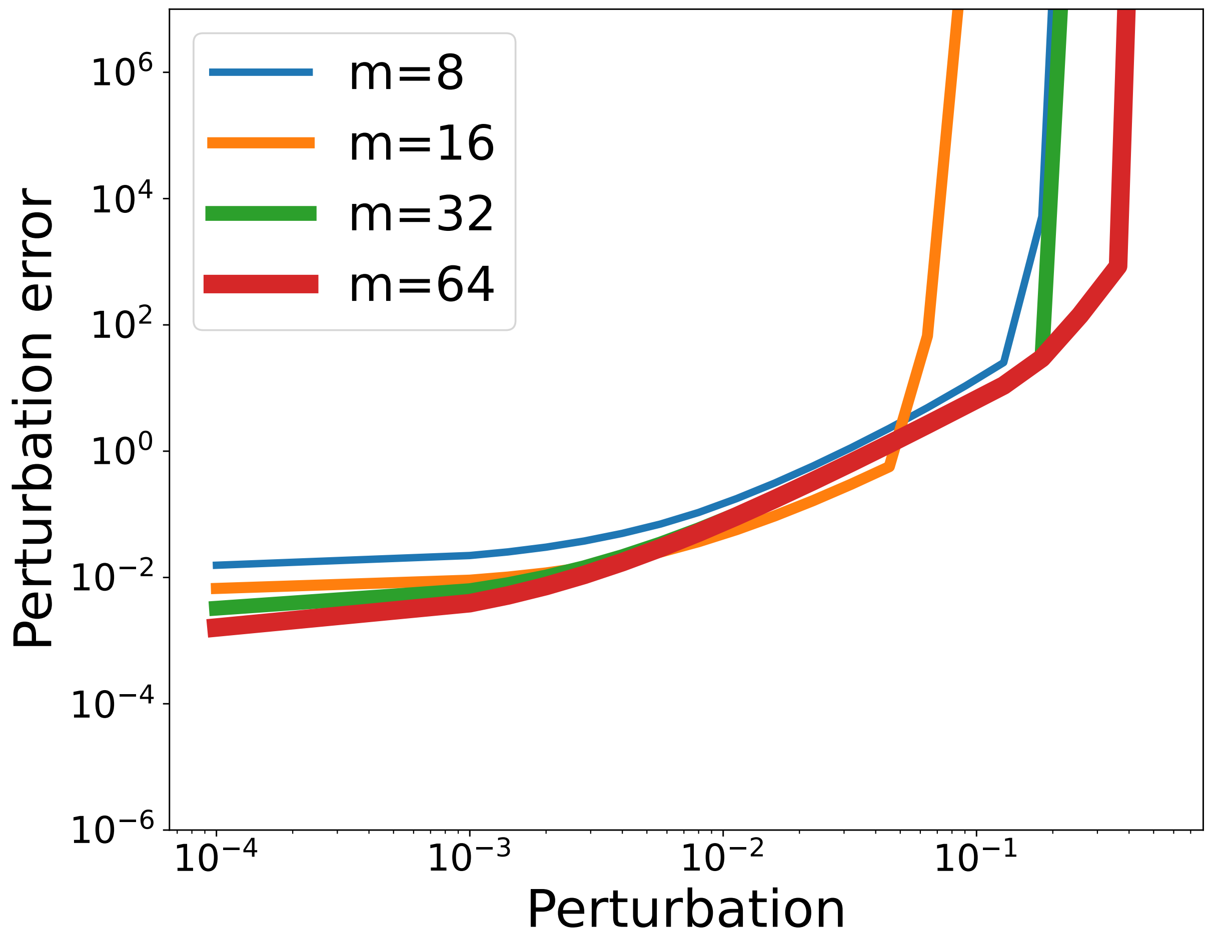
Curse of Memory in Vanilla SSMs

Experiment: approximating polynomial decaying memory



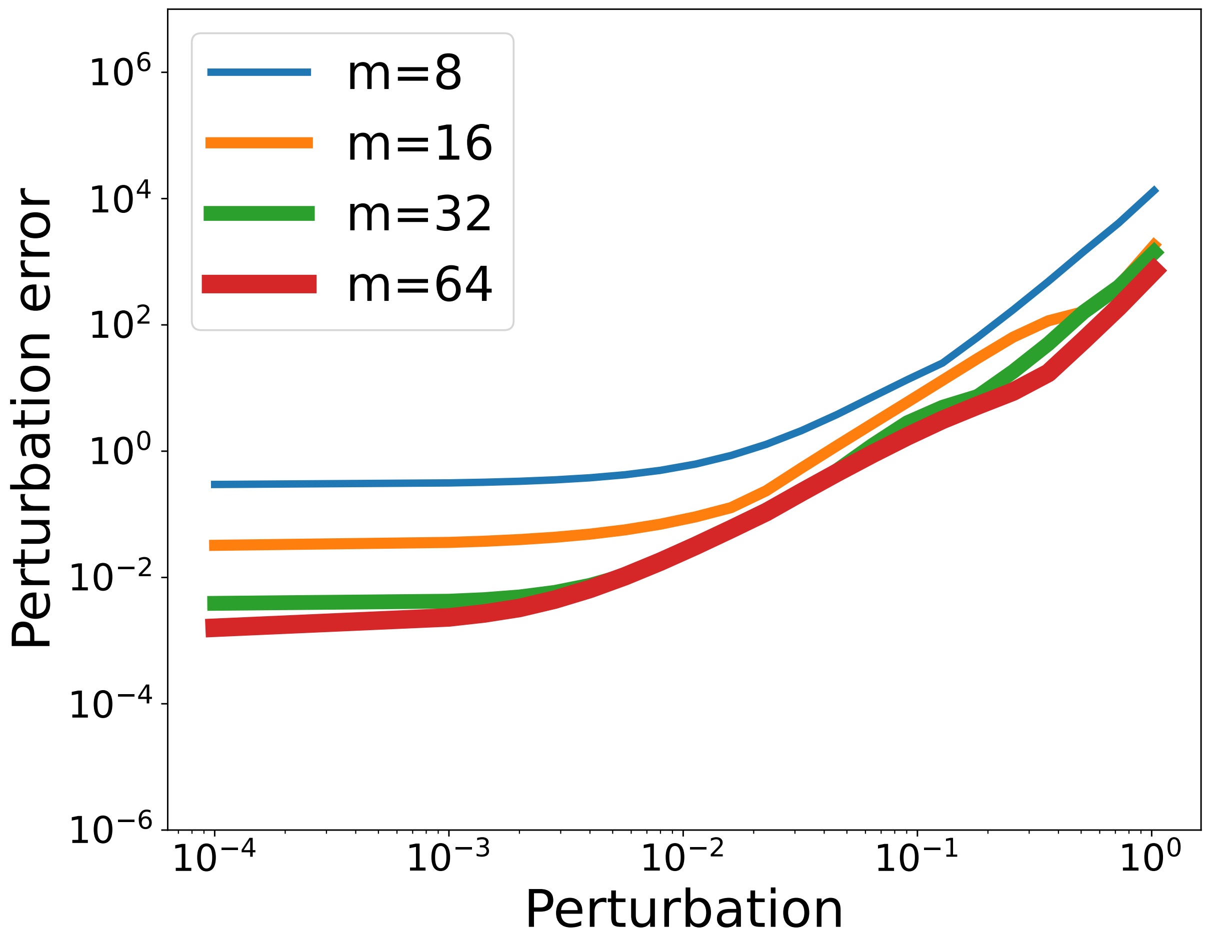
Vanilla SSM

<



SSM with Softplus reparameterization

<



SSM with S4 (Gu et al. 2022) reparameterization

Contents

Reparameterization?

Re-parameterize the matrix $\Lambda = \mathrm{Diag}(\lambda_1, \ldots, \lambda_m)$

- With a reparameterization scheme $\lambda_i = f(w_i)$ (where w_i's are trainable),

$$\hat{y}_t = c^\top \sigma\left(\int_0^\infty \mathrm{Diag}(e^{f(w_1)s}, \ldots, e^{f(w_m)s}) U x_{t-s} \mathrm{d}s \right).$$ (1-layer solution)

Stable reparameterization

The definition (Defn. 3.4) is highly technical.

- **Extension of Theorem 3.5.** There exists a class of stable reparameterizations (See Definition 3.4) satisfying that:
  - For ANY bounded, continuous, regular, causal, time-homogeneous functional sequence $\mathbf{H}$, if SSMs $\{\hat{\mathbf{H}}(\,\cdot\,;\theta_m)\}_{m=1}^{\infty}$ is approximating $\mathbf{H}$ (i.e., satisfying condition #1 of Definition 2.5), then this approximation is a stable approximation.
- Example of stable reparameterizations:
  - $f(w) = -e^{w}, f(w) = -\log(1+e^{w}),\ f(w) = -\frac{1}{aw^2+b}$ for some $a > 0,\ b \geq 0$

Stable reparameterization

The definition (Defn. 3.4) is highly technical.

- **Extension of Theorem 3.5.** There exists a class of stable reparameterizations (See Definition 3.4) satisfying that:
    - For ANY bounded, continuous, regular, causal, time-homogeneous functional sequence $\mathbf{H}$, if SSMs $\{\hat{\mathbf{H}}(\,\cdot\,;\theta_m)\}_{m=1}^{\infty}$ is approximating $\mathbf{H}$ (i.e., satisfying condition #1 of Definition 2.5), then this approximation is a stable approximation.

	Approximation	Stable Approximation
Vanilla SSM	Universal (Wang & Xue, 2023)	Not universal (Theorem 3.3)
StableSSM	Universal (Wang & Xue, 2023)	Universal (Theorem 3.5)

Contents

Parameterization Affects to the Gradient Norm Scale

Theorem 3.6

- Consider a reparameterized SSM $\hat{\mathbf{H}}_m$ with scheme $f(w_i) = \lambda_i$, approximating $\mathbf{H}$.
- Consider a loss function $\mathscr{L}(\mathbf{H}, \hat{\mathbf{H}}_m) = \sup_t \|H_t - \hat{H}_{m,t}\|_\infty = \|H_\tau - \hat{H}_{m,\tau}\|_\infty$.
  - The equality holds for any τ because of time-homogeneity.
- Then the gradient norm is upper bounded as follows:

$$\left| \frac{\partial \mathscr{L}(\mathbf{H}, \hat{\mathbf{H}}_m)}{\partial w_i} \right| \leq C_{\mathbf{H},\hat{\mathbf{H}}_m} \frac{|f'(w_i)|}{f(w_i)^2},$$

where $C_{\mathbf{H},\hat{\mathbf{H}}_m}$ is a constant independent of the parameterization f.

Contents

Optimization stability

in terms of gradient-over-weight scale

- We want the ratio between the gradient and the weight small (in magnitude).
  - …or, bounded by a constant L.
  - It might lead to a stable optimization trajectory.
  - It is desirable when a large learning rate is used in training.

Which parameterization is BEST in stability sense?

Let's derive it from Theorem 3.6!

- Recall: $\left|\dfrac{\partial \mathscr{L}(\mathbf{H}, \hat{\mathbf{H}}_m)}{\partial w_i}\right| \leq C_{\mathbf{H},\hat{\mathbf{H}}_m} \dfrac{|f'(w_i)|}{f(w_i)^2} = L|w_i|$. (Gradient-over-weight scale is bounded by L.)
- A sufficient condition: for some real numbers $a > 0$ and $b \geq 0$,

$$\frac{f'(w)}{f(w)^2} = \frac{\mathrm{d}}{\mathrm{d}w}\left(-\frac{1}{f(w)}\right) = 2aw,$$

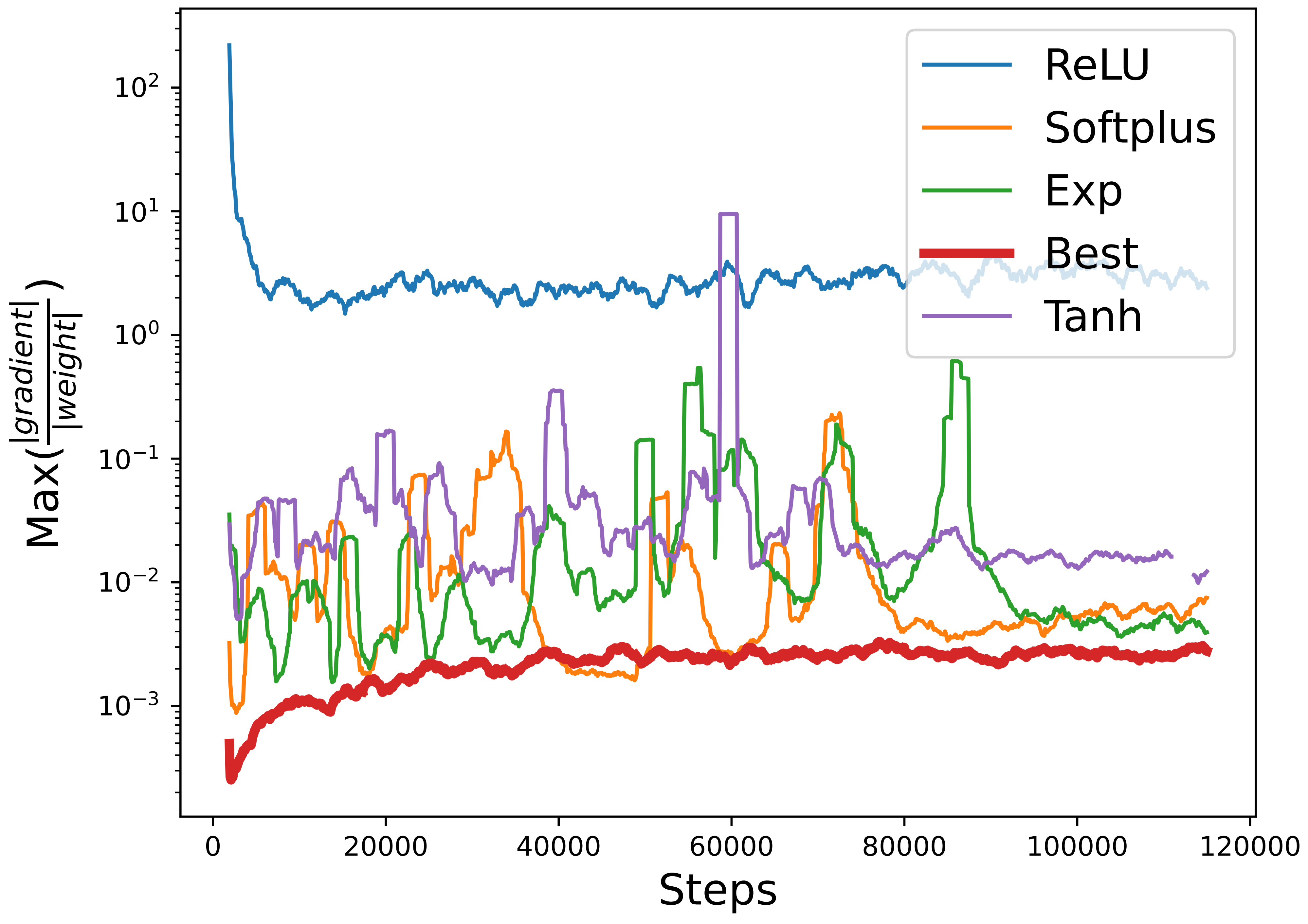
$$-\frac{1}{f(w)} = aw^2 + b,$$

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

Experiments

… with various parameterizations

- The “best” parameterization achieves the smallest (maximum) gradient-over-weight scale over training.



Experiments

… with various parameterizations

- The “best” parameterization allows a large range of learning rates at training.

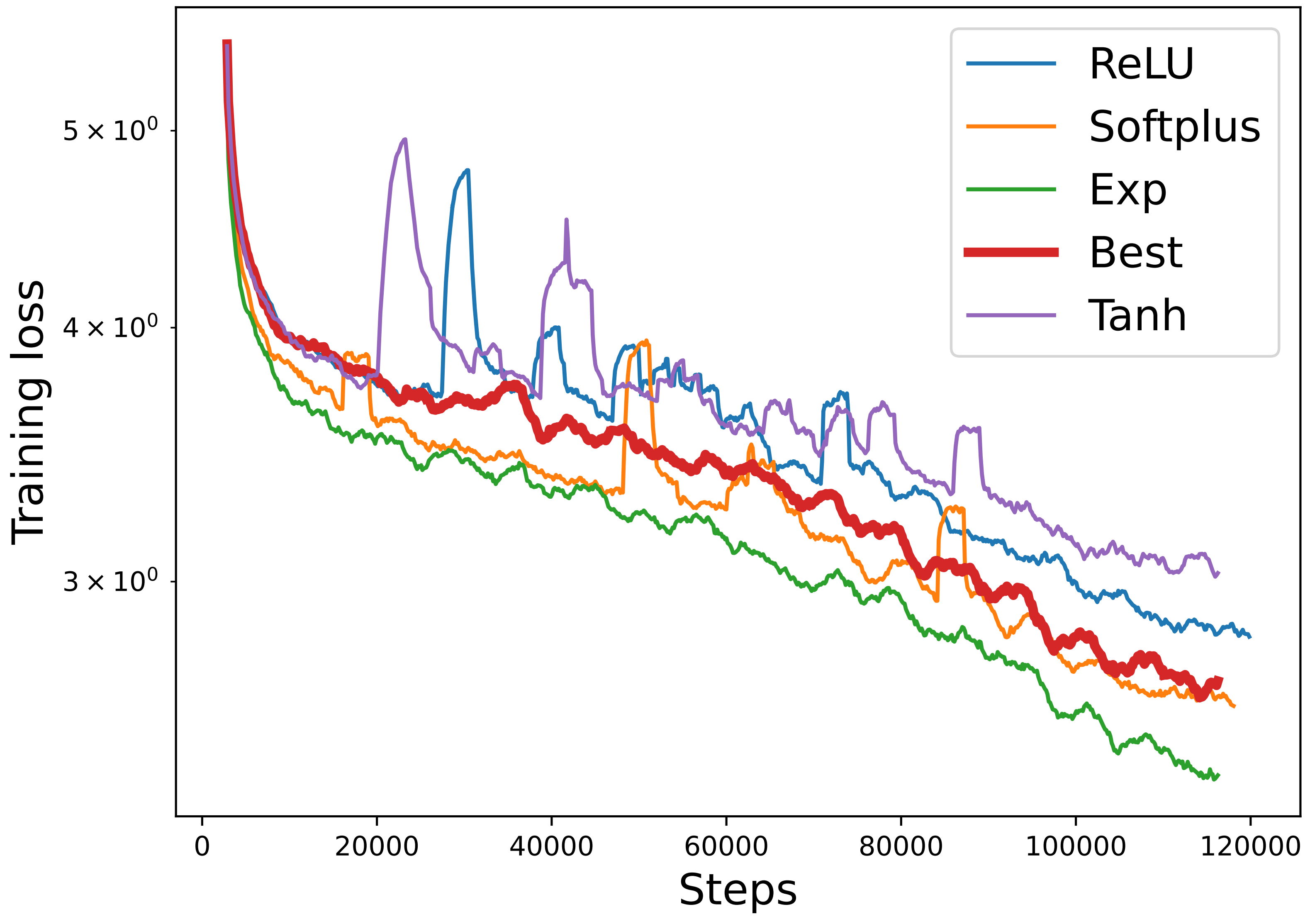
Table 2. Comparison of stability of different parameterizations over MNIST. The experiments conducted on the MNIST and CIFAR10 datasets were replicated three times, with the standard deviation of the test loss indicated in parentheses.

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)

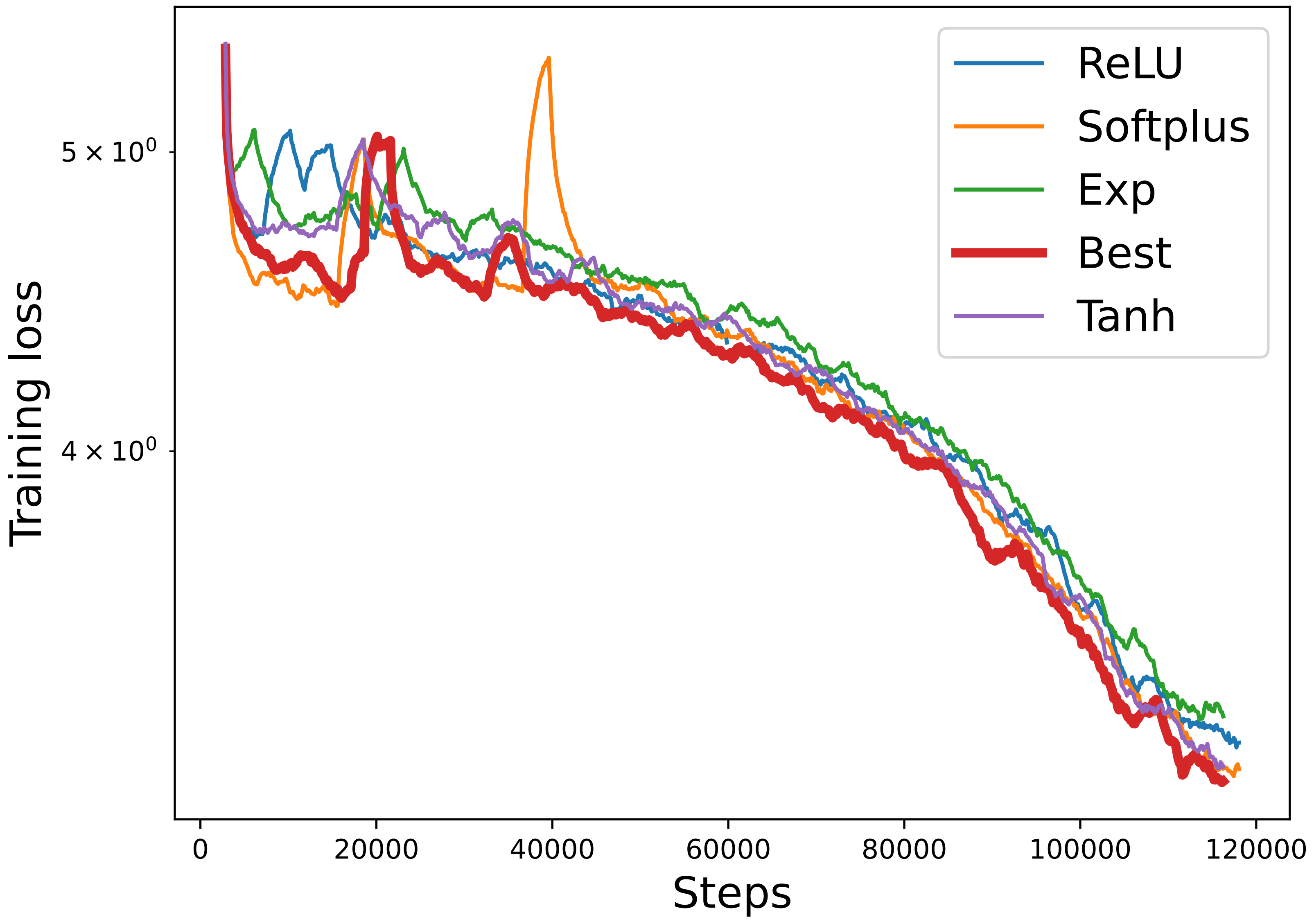
Experiments

… with various parameterizations

- The “best” parameterization achieves better training loss than other parameterizations when the learning rate is large.



LR=0.001



LR=0.01

Q&A

Appendix A. Properties of functionals

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 $\lim_{n \to \infty} H_t(\mathbf{x}^{(n)}) = 0$.