On the Implicit Bias of Gradient Descent for Temporal Extrapolation

Edo Cohen-Karlik* **Avichai Ben David*** **Nadav Cohen** **Amir Globerson**

Blavatnik School of Computer Science
Tel Aviv University

Abstract

When using recurrent neural networks (RNNs) it is common practice to apply trained models to sequences longer than those seen in training. This "extrapolating" usage deviates from the traditional statistical learning setup where guarantees are provided under the assumption that train and test distributions are identical. Here we set out to understand when RNNs can extrapolate, focusing on a simple case where the data generating distribution is memoryless. We first show that even with infinite training data, there exist RNN models that interpolate perfectly (i.e., they fit the training data) yet extrapolate poorly to longer sequences. We then show that if gradient descent is used for training, learning will converge to perfect extrapolation under certain assumptions on initialization. Our results complement recent studies on the implicit bias of gradient descent, showing that it plays a key role in extrapolation when learning temporal prediction models.

1 INTRODUCTION

Practical deep neural networks are often larger than necessary for perfectly fitting the data they are trained on. This "over-parametrized" regime could potentially result in severe overfitting, but in practice neural networks tend to generalize surprisingly well to unseen data. This observation inspired a multitude of works aimed at theoretically understanding the phenomenon. It quickly became clear [Zhang et al., 2017] that the generalization arises from an "implicit bias" — a tendency towards certain solutions that generalize well — induced by variants of gradient descent (GD) and their initialization schemes. Characterizing this implicit bias is a key goal in the theory of deep learning (e.g., see [Gunasekar et al., 2018] and [Woodworth et al., 2020]).

Most of the works studying implicit bias in deep learning consider the setting where train and test data are drawn from the same distribution, and focus on bounding the gap between train and test errors. An equally interesting and complementary problem is that of "extrapolation," which deals with how the learned function behaves outside the training distribution. A recent work has begun to explore this question in the context of fully connected neural networks [Xu et al., 2020], showing that in the "ultra wide" regime, also known as Neural Tangent Kernel (NTK; [Jacot et al., 2020]) regime, learned functions extrapolate linearly.

In this paper we focus on a setting where extrapolation is especially important: learning temporal models. In particular, we focus on recurrent neural networks (RNN), which form a standard tool for this task. It is common practice to train RNNs with sequences up to a certain length, and then apply them at inference time to longer sequences. The fact that this approach often works well in practice suggests that RNNs perform successful temporal extrapolation by virtue of the implicit bias of GD and its initialization scheme. Our aim is to theoretically understand this phenomenon.

Non-linear neural networks are notoriously difficult to analyze when operating outside the NTK regime, i.e. for realistic model sizes. In order to make progress towards their theoretical understanding, researchers have turned to the simplified model of linear neural networks (e.g., see [Arora et al., 2018, Ji and Telgarsky, 2019]). Linear neural networks are trivial in terms of expressiveness (they realize only lin-

Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151. Copyright 2022 by the author(s).

ear input-output mappings), but not so in terms of optimization and generalization: they induce highly non-convex training objectives, and exhibit phenomena akin to their non-linear counterparts. We accordingly base our analysis of temporal extrapolation on linear recurrent neural networks, also known in the literature as linear dynamical systems (LDS) [Antsaklis and Michel, 2006].

The setting we study is geared specifically towards understanding the temporal extrapolation implicitly brought forth by GD. We consider the case where training data is generated by a memoryless teacher, so as to avoid an explicit bias towards non-trivial extrapolation. Learning from the training data via GD on an over-parametrized RNN, the extrapolation question boils down to whether a model trained on sequences of length k will realize a memoryless mapping for time steps greater than k. We show that in this setup, there exist RNN weights that perfectly fit the training data (regardless of how many training sequences were collected), and yet extrapolate poorly. Empirically however, we find that training via GD with standard initialization schemes yields solutions that extrapolate well. This clearly demonstrates an implicit bias towards good extrapolation. Interestingly, learned solutions do not comprise a zero state transition matrix, but rather one which is far from zero, yet still results in good extrapolation.

Any result establishing implicit bias towards specific solutions must entail assumptions on initialization (otherwise, one may initialize at any solution that perfectly fits the training data, and GD will remain there). In the context of linear neural networks, it is common to assume that initialization admits certain balancedness properties [Arora et al., 2018]. In the same spirit, our analysis focuses on an initialization where model weights satisfy certain symmetries. Under such initialization, we prove that if GD converges to a perfect fit of the training data, it will do so with an extrapolating (i.e., memoryless) solution. We show that this extrapolation is due to a certain form of "complementary slackness" between components of the model. To the best of our knowledge, this result is the first to provide formal evidence for an implicit bias of GD towards temporal extrapolation outside the NTK regime.

The remainder of the paper is structured as follows. In Section 2 we discuss related work. Section 3 describes the setup and required definitions. In Section 4 we analyze the case of learning with sequences whose length k is larger than the hidden dimension of the model d. Section 5 shows that when $k < d$ there exist solutions that perfectly fit the training data but do not extrapolate to longer sequences. Section 6 analyzes the solutions obtained by GD in the latter regime, showing they do extrapolate. Finally, section 7 provides experiments supporting our theoretical findings.

2 RELATED WORK

Linear RNNs, also known as linear dynamical systems (LDS) have been studied for decades [Kalman, 1963, Ghahramani and Hinton, 1996]. The aspect most related to this work is *system identification* [Ljung, 1999], which studies the conditions under which the exact parameters of an LDS can be recovered. This is related to the question of temporal extrapolation because recovering the correct system parameters from training sequences of finite length will lead to perfect extrapolation. Note however, that works along this line do not analyze the dynamics of GD, but rather provide characterizations of the conditions under which system identification is possible.

Two related properties of an LDS that are necessary for unique identification are *controllability* and *observability* [Kalman, 1960]. In the memoryless setting the learned LDS is neither controllable nor observable, and therefore its identification is an ill-posed problem not treated by classic approaches. One common approach to identification are subspace methods [Ho and Kálmán, 1966], which perform an SVD of the Hankel matrix (i.e., the matrix representing the linear input-output mapping realized by an LDS) in order to extract an approximation of the system which has a low dimensional state-space. Other approaches to low rank Hankel matrices are based on rank relaxations such as nuclear norm [Fazel et al., 2001, Liu and Vandenberghe, 2010]. See also [Glover, 1984] for additional approximation notions.

This paper focuses on the question of how GD learns a "simple" system from observed data. It may thus be viewed as the LDS analogue of works studying the implicit bias in matrix factorization [Arora et al., 2019]. Such works ask whether GD over matrix factorization fits observations with solutions that minimize a complexity measure, and what that complexity measure is. At present, these questions are largely open.

Our work relates to the recent results of [Hardt et al., 2016] showing that GD can optimize the loss for LDS. Specifically, they showed that when training on sequences up to length k, GD will converge to a dynamical system that approximates the impulse response up to time k. However, this result does not imply extrapolation in the sense we consider here, as it does not guarantee approximation of the impulse response for times beyond k.

Another recent work [Xu et al., 2020] studies extrapolation of deep learning beyond the support of the training data. They show that feed forward neural networks with ReLU activation extrapolate to linear functions outside the support, and further provide principles to construct graph neural networks which exhibit bias towards specific extrapolation. We study extrapolation in the temporal domain, providing insights into the solutions found by GD over linear RNNs.

Another study of implicit bias in linear RNNs is [Emami et al., 2021], who examine the correlation between such models and convolutions in the asymptotic (NTK) regime of an infinite-dimensional hidden state. They show that GD tends to short-term memory solutions. Our work differs from [Emami et al., 2021] in several aspects. First, we consider the realistic case of finite-dimensional hidden states (i.e. we operate outside the NTK regime). Second, we directly prove an extrapolation result, whereas [Emami et al., 2021] provide results on structural biases of learned impulse responses.

3 LINEAR RECURRENT NEURAL NETWORKS

In this section we describe our model and the analyzed setting. We consider a single layer linear RNN, and for simplicity present our analysis for the single-input single-output (SISO) case. The analysis readily extends to the more general multiple-input multiple-output (MIMO) case, as shown in the Supplementary (Section C). The dynamics of interest are defined by:

$$\hat{y}_t = Cs_t, \qquad s_{t+1} = As_t + Bx_{t+1}, \tag{1}$$

where $A \in \mathbb{R}^{d\times d}$, $B \in \mathbb{R}^{d\times 1}$ and $C \in \mathbb{R}^{1\times d}$ are learned parameters (weights), and $\{s_t\}_{t=1}^{\infty} \subset \mathbb{R}^{1\times d}$ are the resulting hidden states, where by assumption $s_0 = 0$. We refer to A as the *state transition matrix*, to B as the *input weights*, and to C as the *output weights*.

Given $\mathbf{x} = (x_1, x_2, \ldots, x_k) \in \mathbb{R}^k$, an input sequence of length k, we denote by $RNN(\mathbf{x}) = \hat{y}_k$ the output of the RNN at time step k.[1] The latter can be expressed in terms of the learned parameters and input sequence. Indeed, it results from taking a convolution of the input sequence with the impulse response sequence $(CA^iB)_{i=0}^{\infty}$:

$$\hat{y}_k = \sum_{i=1}^{k} CA^{k-i}Bx_i. \tag{2}$$

We consider the problem of learning the parameters of the RNN from a set of N training sequences and their desired outputs:

$$S = \left\{(\mathbf{x}^{(i)}, y^{(i)})\right\}_{i=1}^{N} \subset \mathbb{R}^k \times \mathbb{R}, \tag{3}$$

via minimization of the empirical squared loss:

$$\frac{1}{2N}\sum_{i=1}^{N}\left(RNN(\mathbf{x}^{(i)}) - y^{(i)}\right)^2.$$

Note that this setup is more challenging (entails less supervision) than the one in which training labels include outputs for all time steps between 1 and k. For convenience, in the remainder of the paper, we omit the subscript of the output at time k, and simply denote $\hat{y} = \hat{y}_k$.

Our interest lies on the impact of implicit bias on extrapolation. To decouple that from generalization (i.e., from the question of how accurate the model is on sequences of length k not seen during training), we assume an unlimited amount of training data, or formally, that GD is applied to the population loss:

$$\mathbb{E}_{\mathbf{x},y}\left[\frac{1}{2}\left(RNN(\mathbf{x}) - y\right)^2\right].$$

We will study the case where training data is generated by a teacher RNN, and to isolate the implicit bias of GD, we avoid any type of explicit bias towards non-trivial extrapolation. That is, we assume the teacher RNN is ***memoryless***, meaning that there exists $w^* \in \mathbb{R}$ such that for any $k \in \mathbb{N}$ and any input sequence $\mathbf{x} = (x_1, \ldots, x_k) \in \mathbb{R}^k$, the corresponding label is given by $y = w^*x_k$ (namely, the output depends on input only via latest time step). We disregard the trivial case of constant zero labels, i.e. we assume $w^* \neq 0$. Using Equation (2), we obtain an expression for the loss induced by the memoryless teacher, in the case where inputs are drawn independently.

Lemma 3.1. *Assume $\mathbf{x} \sim \mathcal{D}$ such that $\mathbb{E}_{\mathcal{D}}[\mathbf{x}\mathbf{x}^\top] = I_k$, where $I_k \in \mathbb{R}^{k\times k}$ is the identity matrix. Then, given a memoryless teacher RNN, the loss for the student RNN satsifies:*

$$\mathbb{E}_{\mathbf{x},y}\left[\frac{1}{2}\left(RNN(\mathbf{x}) - y\right)^2\right] = \tag{4}$$
$$\frac{1}{2}\sum_{i=1}^{k-1}(CA^{k-i}B)^2 + \frac{1}{2}(CB - w^*)^2$$

Proof. The result follows from expanding the population loss and calculating first and second order moments of $\mathbf{x}$. See Supplementary. □

Lemma 3.1 admits a simple interpretation. It states that the population loss will be minimized when the impulse response starts with w^*, and is followed by $k-1$ zeros. This agrees with the fact that the system is trained to be memoryless for the first k time steps. However, as we shall see later, it does not guarantee that it will be memoryless for times greater than k.

[1] We interchangeably use $RNN(\mathbf{x}) = RNN(x_1, \ldots, x_k)$.

We say that the learned RNN ***extrapolates*** with respect to the teacher RNN if it agrees with the latter's output for any input sequence of *any length*, including lengths which exceed that of the training sequences. This amounts to requiring that for any $j \in \mathbb{N}$ (in particular $j > k$) and any $\mathbf{x} = (x_1, \dots, x_j) \in \mathbb{R}^j$, the output of the learned RNN satisfies $\hat{y} = RNN(\mathbf{x}) = w^* x_j$.

Lemma 3.2. *The parameters (A, B, C) for the learned RNN are extrapolating with respect to a memoryless teacher if and only if $CB = w^*$ and $CA^jB = 0$ for all $j \in \mathbb{N}$.*

Proof. The proof follows from Equation (2) and the expected loss in Lemma 3.1 when $k \to \infty$. □

One possible solution that extrapolates to a memoryless teacher is $A = 0$ along with any pair B, C satsifying $CB = w^*$. Surprisingly, we observe empirically (see Section 7) that typically $A \neq 0$ while $CA^jB = 0$ for all $j \in \mathbb{N}$, suggesting a non-trivial alignment between A and B, C. In the next sections we theoretically explore this phenomenon.

4 LONG TRAINING SEQUENCES GUARANTEE EXTRAPOLATION

Theorem 4.1 below shows that when $k > d$ (i.e., when the length of training sequences is larger than the width of the learned model), any solution that minimizes the loss extrapolates.

Theorem 4.1. *Assume that $k > d$, and let (A, B, C) be a solution (parameters for learned RNN) that minimizes the loss in Equation (4). Then, it holds that $CA^jB = 0$ for all $j \in \mathbb{N}$, meaning the learned model extrapolates.*

Proof. Let $p(z) = z^d + \rho_{d-1} z^{d-1} + \dots + \rho_1 z + \rho_0$, be the characteristic polynomial of the matrix $A \in \mathbb{R}^{d \times d}$. By the Cayley-Hamilton theorem [Zhang, 1997, Frobenius, 1877]:

$$p(A) = A^d + \rho_{d-1} A^{d-1} + \dots + \rho_1 A + \rho_0 I = 0,$$

which implies that we may write:

$$A^d = -\sum_{i=0}^{d-1} \rho_i A^i.$$

Multiplying both sides of the above by A, followed by left multiplication by C and right multiplication by B, yields:

$$CA^{d+1}B = -\sum_{i=0}^{d-1} \rho_i CA^{i+1}B.$$

Since the global minimum of the loss in Equation (4) is zero, it necessarily holds that $CA^jB = 0$ for all $j \in \{1, \dots, k-1\}$, and in particular for all $j \in \{1, \dots, d\}$. We therefore have:

$$CA^{d+1}B = -\sum_{i=0}^{d-1} c_i CA^{i+1}B = 0.$$

Continuing in this fashion, we conclude that $CA^jB = 0$ for all $j \in \mathbb{N}$. □

The above result implies that sufficiently long training sequences guarantee extrapolation. In other words, the training data in this case is sufficient to uniquely identify the memoryless teacher. As we shall see next, for shorter training sequences this no longer holds.

5 EXTRAPOLATION MAY FAIL FOR SHORT TRAINING SEQUENCES

In Section 4 we showed that when training sequences have length larger than the width of the trained model ($k > d$), learning guarantees extrapolation. Proposition 5.1 below shows that in stark contrast, when the training sequence length is no greater than model width ($k \leq d$), there exist solutions which minimize the training loss, and yet fail to extrapolate. This implies that an arbitrary loss-minimizing learning algorithm may result in non-extrapolating solutions. In Section 6 we show that despite this fact, under certain conditions, solutions found by GD do extrapolate.

Proposition 5.1. *For any training sequence length $k \geq 2$ and model width $d \geq k$, there exist RNN parameters (A, B, C) that minimize the loss in Equation (4) but do not extrapolate.*

Proof. Assume $d \geq k \geq 2$, and consider the following parameter setting for the learned RNN.

$$A = \begin{pmatrix} 0 & 0 & \dots & 0 & 0 & 1 \\ 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 & 0 \\ & & \ddots & & & \\ 0 & 0 & \dots & 1 & 0 & 0 \\ 0 & 0 & \dots & 0 & 1 & 0 \end{pmatrix} \in \mathbb{R}^{d \times d} \ ,$$

$$B = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \in \mathbb{R}^{d \times 1} \ , \quad C = (w^*, 0, \dots, 0) \in \mathbb{R}^{1 \times d}.$$

Note that A is a permutation matrix, and specifically, multiplying A from the right by a general matrix $M \in$

$\mathbb{R}^{d\times d}$ results in a cyclic shift of rows, i.e.:

$$M = \begin{pmatrix} - & M_1 & - \\ & \vdots & \\ - & M_d & - \end{pmatrix}, \quad AM = \begin{pmatrix} - & M_d & - \\ - & M_1 & - \\ & \vdots & \\ - & M_{d-1} & - \end{pmatrix}.$$

Applying A to itself, we have that for any $n \in \mathbb{N}$, $A^{nd} = I$ and consequently $CA^{nd}B = w^* \neq 0$, which contradicts extrapolation (by Lemma 3.2). On the other hand, for $j \in \{1, \ldots, d-1\}$ we have $CA^jB = 0$ since the first row of A^j is not e_1.[2] To conclude, since $d \geq k$, the loss in Equation (4) is zero and therefore minimized, while the RNN does not meet the necessary condition for extrapolation. □

6 IMPLICIT BIAS OF GRADIENT DESCENT

Section 5 showed that when the length of training sequences is no greater than the width of the trained model ($k \leq d$), there exist solutions which minimize the loss (achieve perfect generalization) and yet do not extrapolate. Despite the existence of such non-extrapolating solutions, we observe empirically (see Section 7) that GD with standard initialization entails an implicit bias towards solutions that do extrapolate. Theorem 6.1 below theoretically grounds this phenomenon, for the case where the input and output weights are initialized to the same value (i.e. $B = C^\top$), and the state transition matrix A is initialized symmetrically. This initialization captures the "residual" setting $A = I_d$, and more generally, allows A to have arbitrary magnitude, implying arbitrary distance from a trivial solution in which $A = 0$. We emphasize that while our analysis assumes symmetric initialization, we observe empirical convergence to an extrapolating solution under non-symmetric initialization as well (see Section 7). Interestingly, we often see that the parameters of the model converge to a symmetric configuration even if not initialized this way (see Subsection 7.1). Theoretically explaining this phenomenon is a promising direction for future work.

Theorem 6.1. *Assume $d \geq k > 2$, $w^* > 0$, and that the learned RNN is initialized such that $B = C^\top$ and $A = A^\top$. Then, if GD converges to a solution minimizing the loss in Equation* (4)*, this solution necessarily extrapolates.*

Proof. The proof proceeds in two steps: we first show that GD preserves a few properties throughout training, and then establish that with these properties in place, any solution (A, B, C) minimizing the loss must satisfy $CB = w^*$ and CA^jB=0 for all $j \in \mathbb{N}$ — conditions equivalent to extrapolation (see Lemma 3.2).

Denote the training loss in Equation (4) by $\mathcal{L}(A, B, C)$. A simple computation of derivatives (provided in Appendix B) yields:

$$\frac{\partial \mathcal{L}}{\partial B} = \sum_{i=1}^{k-1} (A^\top)^i C^\top C A^i B + C^\top (CB - w^*), \quad (5)$$

$$\frac{\partial \mathcal{L}}{\partial C} = \sum_{i=1}^{k-1} C A^i B B^\top (A^\top)^i + (CB - w^*) B^\top, \quad (6)$$

and

$$\frac{\partial \mathcal{L}}{\partial A} = \sum_{i=1}^{k-1} \sum_{r=0}^{i-1} (A^\top)^r C^\top C A^i B B^\top (A^\top)^{i-r-1}. \quad (7)$$

Denote by A_t, B_t and C_t the weights of the learned model at iteration $t \in \mathbb{N} \cup \{0\}$ of GD. We will prove by induction that $B_t = C_t^\top$ and $A_t = A_t^\top$ for all t. By assumption this holds for $t = 0$. Suppose it is true for some t. The GD updates for B and C are given by:

$$B_{t+1} = B_t - \eta \sum_{i=1}^{k-1} (A_t^\top)^i C_t^\top C_t A_t^i B_t - \eta C_t^\top (C_t B_t - w^*), \quad (8)$$

$$C_{t+1} = C_t - \eta \sum_{i=1}^{k-1} C_t A_t^i B_t B_t^\top (A_t^\top)^i + - \eta (C_t B_t - w^*) B_t^\top. \quad (9)$$

Taking transpose of the right-hand side of Equation (9), while noting that by our inductive hypothesis $B_t = C_t^\top$ and $A_t = A_t^\top$, we obtain equality to the right-hand side of Equation (8), where we used the fact that $CB - w^*$ is a scalar. This implies $B_{t+1} = C_{t+1}^\top$.

As for A, its GD update is:

$$A_{t+1} = A_t - \eta \sum_{i=1}^{k-1} \sum_{r=0}^{i-1} \gamma_{t,i,r} \quad (10)$$

where

$$\gamma_{t,i,r} = (A_t^\top)^r C_t^\top C_t A_t^i B_t B_t^\top (A_t^\top)^{i-r-1}$$

For any i between 1 and $k-1$, the internal summation (over r) is symmetric. To see this, let $W = C_t^\top C_t A_t^i B_t B_t^\top$, and note that W is a symmetric matrix (since $B_t = C_t^\top$ and A_t is symmetric by our inductive

[2] The first row of A^j is given by e_{d-j+1}

hypothesis). Now, for every term $(A_t^\top)^r W (A_t^\top)^{i-1-r}$, a corresponding term $(A_t^\top)^{i-1-r} W (A_t^\top)^r$ also appears in the summation, and these two terms together form a symmetric matrix (an exception is the case $i-1-r=r$, which corresponds to itself but is already symmetric). We conclude that the GD update in Equation (10) can be written as a sum of symmetric matrices, and is therefore itself symmetric. That is, $A_{t+1} = A_{t+1}^\top$, and our inductive hypothesis is proven.

Moving on to the second part of the proof, let (A, B, C) be a minimizer of $\mathcal{L}(A, B, C)$ satisfying $B = C^\top$ and $A = A^\top$. By the structure of $\mathcal{L}(A, B, C)$ (Equation (4)), it holds that $CB = w^*$ and $CA^jB = 0$ for any $j \in \{1, \ldots, k-1\}$. We will show that $CA^jB = 0$ for all $j \in \mathbb{N}$. Recalling that $k > 2$, we have in particular:

$$CA^2B = 0. \tag{11}$$

A is symmetric and therefore orthogonally diagonalizable, meaning there exists an orthogonal matrix $V \in \mathbb{R}^{d \times d}$ and a diagonal matrix $D \in \mathbb{R}^{d \times d}$ such that $A = VDV^\top$. We can thus write $A^2 = VD^2V^\top$, and since $B = C^\top$, Equation (11) implies:

$$CA^2B = B^\top A^2 B = B^\top V D^2 V^\top B = 0.$$

Denoting $\mathbf{u} = (u_1, \ldots, u_d)^\top = V^\top B$, we may write the above as

$$\mathbf{u}^\top D^2 \mathbf{u} = \sum_{i=1}^{d} u_i^2 \lambda_i^2 = 0\ . \tag{12}$$

Since this is a sum of non-negative elements that sum to zero, each of them must be zero, namely:

$$u_i \lambda_i = 0, \quad i = 1, \ldots, d\ . \tag{13}$$

We refer to this as a *complementary slackness* condition, since it implies that either u_i or λ_i should be zero for any i. For arbitrary $j \in \mathbb{N}$:

$$CA^jB = B^\top V D^j V^\top B = \mathbf{u}^\top D^j \mathbf{u} = \sum_{i=1}^{d} u_i^2 \lambda_i^j = 0,$$

where the equality to zero follows from Equation (13). This is precisely the condition we set out to prove. □

Remark 6.2. *The assumption $w^* > 0$ can easily be converted to $w^* < 0$, by modifying the conditions on initialization to include $B = -C^\top$ instead of $B = C^\top$.*

Key to the proof of Theorem 6.1 is the fact that symmetry is invariant under GD, i.e. if the model weights are symmetric at initialization, they remain that way throughout. A natural question which arises is whether non-extrapolating solutions such as those described in Section 5 can be expressed with a symmetric weight configuration. The following lemma shows that there exist symmetric weight configurations with arbitrarily small loss values that do not extrapolate.

Lemma 6.3. *Assume $d \geq k \geq 2$ and $w^* > 0$. For any $\epsilon > 0$, there exists a weight configuration (A, B, C) where $B = C^\top$ and $A = A^\top$, such that the loss in Equation* (4) *is smaller than ϵ yet the model does not extrapolate.*

Proof. We present a proof for $k = 3$ and $d = 4$. Extension to arbitrary values of k and d (satisfying $d \geq k \geq 2$) is straightforward.

Let $C = B^\top = (\sqrt{w^*}, 0, 0, \sqrt{\delta})$, $A = diag(0, 0, 0, 2)$, where $\delta > 0$. The loss in Equation (4) is then:

$$(CA^2B)^2 + (CAB)^2 + (CB - w^*)^2 = 16\delta^2 + 4\delta^2 + \delta^2\ .$$

This is smaller than ϵ if $\delta < \sqrt{\frac{\epsilon}{21}}$. On the other hand, when tested on sequences of length $\tilde{k}$, the loss will be $\sum_{i=0}^{\tilde{k}} 2^{2i}\delta^2$, which diverges with $\tilde{k}$. (A, B, C) therefore do not extrapolate. □

7 EXPERIMENTS

In this section we support our theoretical findings with several synthetic experiments demonstrating an implicit bias of gradient-based optimization towards extrapolating solutions. The experiments cover not only linear RNNs (the subject of our theory), but also non-linear recurrent models including Long-Short Term Memory (LSTM) [Hochreiter and Schmidhuber, 1997] and Gated Recurrent Units (GRU) [Cho et al., 2014]. Unless stated otherwise, in all experiments we use (non-symmetric) Xavier initialization [Glorot and Bengio, 2010]. For optimization we use Adam [Kingma and Ba, 2017] with learning rate 10^{-3} and default momentum parameters of Keras implementation ($\beta_1 = 0.9$ and $\beta_2 = 0.999$). Data is generated from a standard normal distribution with identity covariance, we experiment with many training sequences to ensure good generalization for the training sequence length, thereby decoupling it from the question of extrapolation. Models had width (hidden state dimension) $d = 30$, and were trained with sequences of length $k = 5$. All experiments were run on a simple Colab client.

Section 5 showed that when $d \geq k$, there exist solutions that perfectly fit training data yet fail to extrapolate to longer sequences. In order to demonstrate this empirically we "adversarially" learn models using the length k training sequences, while including erroneous (non-extrapolating) values for later time steps. This strategy is similar to that of learning with noisy labels, see [Zhang et al., 2017]. We refer to models learned in this fashion as "adversarial data" in our figures. Figure 1 reports an extrapolation experiment in the case of a memoryless teacher (in accordance with our theory). Three architectures are learned — linear RNN, LSTM and GRU — and are evaluated on time steps up to 15 (recall that training sequences are of length 5). For each architecture, we compare the result of with vs. without adversarial learning as described above. The results confirm that all architectures extrapolate despite existence of non-extrapolating solutions.

Although our theory applies to a memoryless teacher, the question of temporal extrapolation is relevant for teachers with arbitrarily long memory. Namely, if training data is generated from a teacher with state space of dimension d^* on k time units, will a model learned via gradient-based optimization extrapolate? In light of our findings thus far, one may hope that extrapolation also occurs for non-zero d^*. Figure 2 confirms that this is indeed the case, via an experiment analogous to that of Figure 1 but with $d^* = 3$.

Next, we empirically demonstrate the complementary slackness phenomenon discussed in the proof of Theorem 6.1. The proof suggests that any non-zero eigenvalue of A must align with zero entries of the projections of B, C onto the orthonormal eigen-basis of A. Figure 3 demonstrates that this is indeed the case, for a linear RNN learned via gradient-based optimization from a memoryless teacher.

7.1 Weight Dynamics

Section 6 showed that when initializing a linear RNN symmetrically ($A^\top = A$ and $B = C^\top$), GD is guaranteed to preserve symmetry, and consequently converge to an extrapolating solution. In this experiment we optimize the population loss directly (e.g. Equation 4) using GD as to observe the weight dynamics leading to extrapolation. Figure 4 below suggests that GD exhibits a tendency towards symmetry even when initialization is non-symmetric. It displays the evolution of weights during optimization when A is initialized as $A_0 = \alpha I$ with random $\alpha \in [0, 1]$, and B, C are initialized independently from a random normal distribution. As can be seen, weights converge to an approximately symmetric solution, in the sense that the norms of $A - A^\top$ and $B - C^\top$ are much smaller than those of A, B, C. We hypothesize that this is due to conservation laws of the GD dynamics, akin to those studied in [Saxe et al., 2013, Kunin et al., 2020]. Their derivation is left for future work.

8 CONCLUSIONS

In this paper we studied the implicit bias of gradient descent (GD) in the context of temporal extrapolation. Focusing on linear recurrent neural networks (RNNs), also known as linear dynamical systems, we analyzed the setting of unlimited training data generated from a memoryless teacher network, and proved that when the width of the learned model is greater than the length of training sequences, there exist solutions that do not extrapolate, yet GD will converge to solutions that do. We showed that this is a result of a complementary slackness phenomenon between the eigenvalues of the state transition matrix A and the input and output weights B and C respectively.

Our theory imposes certain assumptions on initialization, and is limited to a memoryless teacher. However, we demonstrate empirically that gradient-based optimization exhibits an implicit bias towards extrapolation even without these restrictions, in particular using standard initialization schemes and teachers with memory. Moreover, our experiments confirm that the phenomenon extends to non-linear RNNs including GRU and LSTM. We believe elements of our theory may prove useful in analyzing non-linear RNNs, and view this pursuit as an direction for future work.

Our work extends the rich body of literature studying implicit biases of GD in neural networks, by treating the important class of temporal (recurrent) models, and in particular the question of temporal extrapolation. We believe our results may contribute to a better understanding of when extrapolation fails or succeeds, thereby facilitating learning algorithms that improve time series prediction.

9 Acknowledgements

This work was supported by a Google Research Scholar Award, a Google Research Gift, the Yandex Initiative in Machine Learning, the Israel Science Foundation (grant 1780/21), the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme (grant ERC HOLI 819080), Len Blavatnik and the Blavatnik Family Foundation, and Amnon and Anat Shashua.

References

[Antsaklis and Michel, 2006] Antsaklis, P. J. and Michel, A. N. (2006). *Linear systems*. Springer Science & Business Media.

[Arora et al., 2018] Arora, S., Cohen, N., Golowich, N., and Hu, W. (2018). A convergence analysis of gradient descent for deep linear neural networks. *arXiv preprint arXiv:1810.02281*.

[Arora et al., 2019] Arora, S., Cohen, N., Hu, W., and Luo, Y. (2019). Implicit regularization in deep matrix factorization.

[Cho et al., 2014] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H.,

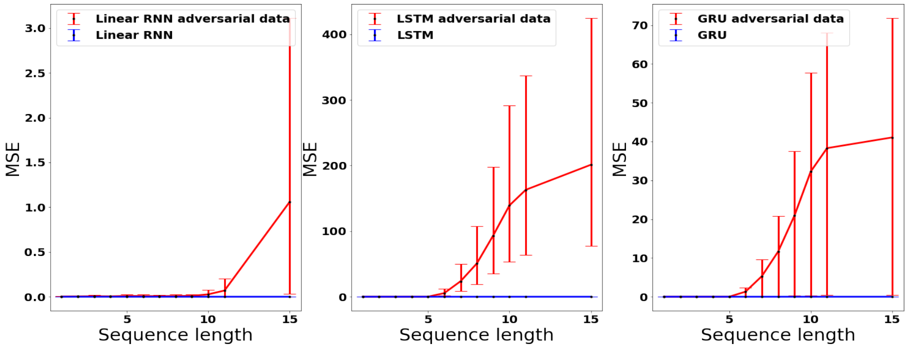

Figure 1: Mean squared error over sequences of different lengths, after learning from length 5 training sequences generated from a memoryless teacher. As can be seen, despite the fact that using adversarial data it is possible to fit training sequences with non-extrapolating solutions, gradient-based optimization leads to extrapolation.

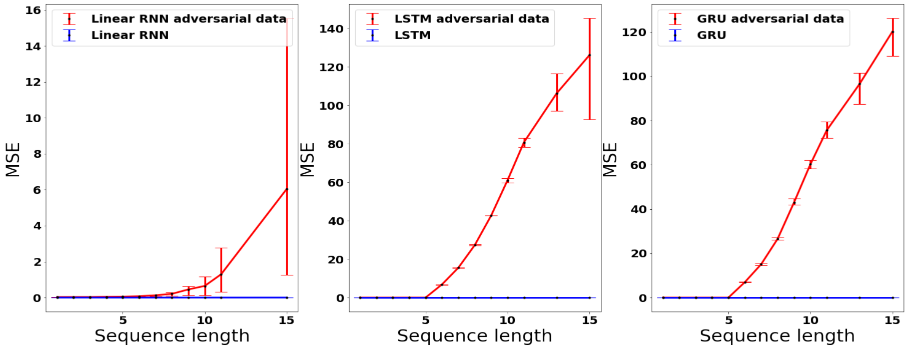

Figure 2: Mean squared error over sequences of different lengths, after learning from length 5 training sequences generated from a teacher with memory (state space of dimension 3). As can be seen, despite the fact that using adversarial data it is possible to fit training sequences with non-extrapolating solutions, gradient-based optimization leads to extrapolation.

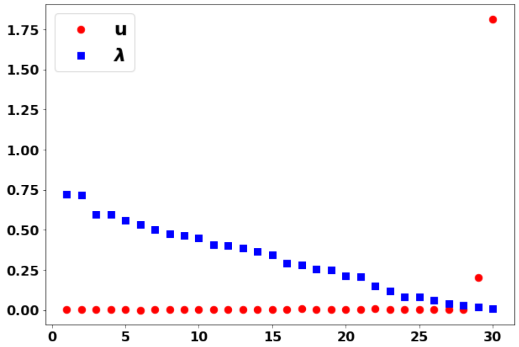

Figure 3: Empirical demonstration of the complementary slackness phenomenon from the proof of Theorem 6.1. In blue are the absolute values of the eigenvalues of the state transition matrix A. The vector $\mathbf{u}$ is the projection of the input and output weights B and C (respectively) onto the orthonormal basis of A, i.e. $\mathbf{u} := V^\top B$ in the notations of the proof of Theorem 6.1. The proof shows that the implicit bias of GD ensures that $\lambda_i u_i = 0$ for all i. The results above validate this phenomenon.

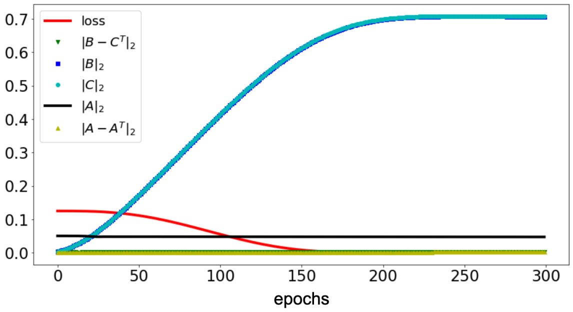

Figure 4: Dynamics of A, B, C — weights of linear RNN — under GD, when B, C are initialized independently (non-symmetrically) from a zero-centered Gaussian distribution with variance $\sigma^2 = 10^{-5}$, and A is initialized as scaled identity. The figure shows the loss, the (Euclidean) norms of the weights, and as a measure of symmetry, the norms of $B - C^\top$ and $A - A^\top$. As can be seen, the norms of B, C grow while the weights remain approximately symmetric.

and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078.*

[Emami et al., 2021] Emami, M., Sahraee-Ardakan, M., Pandit, P., Rangan, S., and Fletcher, A. K. (2021). Implicit bias of linear rnns.

[Fazel et al., 2001] Fazel, M., Hindi, H., and Boyd, S. P. (2001). A rank minimization heuristic with application to minimum order system approximation. In *Proceedings of the 2001 American Control Conference.(Cat. No. 01CH37148)*, volume 6, pages 4734–4739. IEEE.

[Frobenius, 1877] Frobenius, G. (1877). Ueber lineare substitutionen und bilineare formen. *Journal für die reine und angewandte Mathematik*, 84:1–63.

[Ghahramani and Hinton, 1996] Ghahramani, Z. and Hinton, G. E. (1996). Parameter estimation for linear dynamical systems. Technical report, Technical Report CRG-TR-96-2, University of Totronto, Dept. of Computer Science.

[Glorot and Bengio, 2010] Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings.

[Glover, 1984] Glover, K. (1984). All optimal hankel-norm approximations of linear multivariable systems and their $l\infty$ error bounds. *International journal of control*, 39(6):1115–1193.

[Gunasekar et al., 2018] Gunasekar, S., Lee, J., Soudry, D., and Srebro, N. (2018). Characterizing implicit bias in terms of optimization geometry. In *International Conference on Machine Learning*, pages 1832–1841. PMLR.

[Hardt et al., 2016] Hardt, M., Ma, T., and Recht, B. (2016). Gradient descent learns linear dynamical systems. *arXiv preprint arXiv:1609.05191.*

[Ho and Kálmán, 1966] Ho, B. and Kálmán, R. E. (1966). Effective construction of linear state-variable models from input/output functions. *at-Automatisierungstechnik*, 14(1-12):545–548.

[Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

[Jacot et al., 2020] Jacot, A., Gabriel, F., and Hongler, C. (2020). Neural tangent kernel: Convergence and generalization in neural networks.

[Ji and Telgarsky, 2019] Ji, Z. and Telgarsky, M. (2019). Gradient descent aligns the layers of deep linear networks. In *7th International Conference on Learning Representations, ICLR 2019.*

[Kalman, 1960] Kalman, R. E. (1960). On the general theory of control systems. In *Proceedings First International Conference on Automatic Control, Moscow, USSR*, pages 481–492.

[Kalman, 1963] Kalman, R. E. (1963). Mathematical description of linear dynamical systems. *Journal of the Society for Industrial and Applied Mathematics, Series A: Control*, 1(2):152–192.

[Kingma and Ba, 2017] Kingma, D. P. and Ba, J. (2017). Adam: A method for stochastic optimization.

[Kunin et al., 2020] Kunin, D., Sagastuy-Brena, J., Ganguli, S., Yamins, D. L., and Tanaka, H. (2020). Neural mechanics: Symmetry and broken conservation laws in deep learning dynamics. *arXiv preprint arXiv:2012.04728.*

[Liu and Vandenberghe, 2010] Liu, Z. and Vandenberghe, L. (2010). Interior-point method for nuclear norm approximation with application to system identification. *SIAM Journal on Matrix Analysis and Applications*, 31(3):1235–1256.

[Ljung, 1999] Ljung, L. (1999). System identification. *Wiley encyclopedia of electrical and electronics engineering*, pages 1–19.

[Petersen and Pedersen, 2012] Petersen, K. and Pedersen, M. (2012). The matrix cookbook, version 20121115. *Technical Univ. Denmark, Kongens Lyngby, Denmark, Tech. Rep*, 3274.

[Saxe et al., 2013] Saxe, A. M., McClelland, J. L., and Ganguli, S. (2013). Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120.*

[Woodworth et al., 2020] Woodworth, B. E., Gunasekar, S., Lee, J. D., Moroshko, E., Savarese, P., Golan, I., Soudry, D., and Srebro, N. (2020). Kernel and rich regimes in overparametrized models. In Abernethy, J. D. and Agarwal, S., editors, *Conference on Learning Theory, COLT 2020, 9-12 July 2020, Virtual Event [Graz, Austria]*, volume 125 of *Proceedings of Machine Learning Research*, pages 3635–3673. PMLR.

[Xu et al., 2020] Xu, K., Zhang, M., Li, J., Du, S. S., Kawarabayashi, K.-i., and Jegelka, S. (2020). How neural networks extrapolate: From feedforward to graph neural networks. *arXiv preprint arXiv:2009.11848.*

[Zhang et al., 2017] Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. (2017). Understanding deep learning requires rethinking generalization. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

[Zhang, 1997] Zhang, F. (1997). Quaternions and matrices of quaternions. *Linear algebra and its applications*, 251:21–57.

Supplementary Material:
On the Implicit Bias of Gradient Descent for Temporal Extrapolation

A Proof of Lemma 3.1 (Main Paper): Population Risk for SISO

Lemma A.1. *3.1[Main Text] Assume* $\mathbf{x} \sim \mathcal{D}$ *such that* $\mathbb{E}[\mathbf{x}] = 0$ *and* $\mathbb{E}[\mathbf{x}\mathbf{x}^T] = I_k$*, where* $I_k \in \mathbb{R}^{k,k}$ *is the identity matrix. Then, given a memoryless teacher RNN, the loss for the student RNN is given by:*

$$\mathbb{E}_{\mathbf{x},y}\left[\frac{1}{2}\left(RNN(\mathbf{x}) - y\right)^2\right] = \frac{1}{2}\sum_{i=1}^{k-1}(CA^{k-i}B)^2 + \frac{1}{2}(CB - w^*)^2 \tag{14}$$

Proof.

$$\mathbb{E}_{\mathbf{x},y}\left[\frac{1}{2}\left(RNN(\mathbf{x}) - y\right)^2\right] = \frac{1}{2}\mathbb{E}\left[\left(\sum_{i=1}^{k} CA^{k-i}Bx_i - w^*x_k\right)^2\right]$$

The above can be written as

$$\frac{1}{2}\mathbb{E}\left[\left(\sum_{i=1}^{k}\left(CA^{k-i}Bx_i\right)\right)^2 + (w^*x_k)^2 - 2\sum_{j=1}^{k} CA^{k-j}Bx_jx_kw^*\right] \tag{15}$$

Because $\mathbf{x}$ has identity covariance ($\mathbb{E}[x_i^2] = 1$) many terms cancel out and the above is equal to

$$\frac{1}{2}\left[\sum_{i=1}^{k}(CA^{k-i}B)^2 + \left((w^*)^2 - 2CBw^*\right)\right]$$

Removing $i = k$ from the summation, we have,

$$\frac{1}{2}\sum_{i=0}^{k-1}(CA^{k-i}B)^2 + \frac{1}{2}\left((CB)^2 - 2CBw^* + (w^*)^2\right)$$

The above can be written as

$$\frac{1}{2}\sum_{i=0}^{k-1}(CA^{k-i}B - 0)^2 + \frac{1}{2}(CB - w^*)^2$$

which concludes the proof. □

B Details for Proof of Theorem 6.1

Here we the gradient computation for the proof of Theorem 6.1 in the main text.

Consider the expected loss in Lemma A.1. For $\frac{1}{2}(CB-w^*)^2$ the derivative w.r.t B is

$$\frac{\partial \frac{1}{2}(CB-w^*)^2}{\partial B} = C^T(CB-w^*) \tag{16}$$

For $j \geq 1$, the derivative of $\frac{1}{2}(CA^jB)^2$ w.r.t to B is given by

$$\frac{\partial \frac{1}{2}(CA^jB)^2}{\partial B} = (A^j)^T C^T C A^j B = (A^T)^j C^T C A^j B \tag{17}$$

Putting together Equations (16) and (17), the derivative of (4) in the main text w.r.t. B is given by

$$\frac{\partial \mathcal{L}}{\partial B} = \sum_{i=1}^{k-1} (A^T)^i C^T C A^i B + C^T(CB-w^*)$$

A similar derivation w.r.t. C yields:

$$\frac{\partial \mathcal{L}}{\partial C} = \sum_{i=1}^{k-1} C A^i B B^T (A^T)^i + (CB-w^*)B^T$$

For the gradient w.r.t. A, $\forall i \geq 1$, the derivative of $\frac{1}{2}(CA^iB)^2$ is based on Equation (91) from [Petersen and Pedersen, 2012].

$$\frac{\partial \frac{1}{2}(CA^iB)^2}{\partial A} = \sum_{r=0}^{i-1} (A^r)^T C^T C A^i B B^T (A^{i-1-r})^T$$

Using $(A^j)^T = (A^T)^j$ and summing over $i = 1, \ldots, k-1$ results in:

$$\frac{\partial \mathcal{L}}{\partial A} = \sum_{i=1}^{k-1} \sum_{r=0}^{i-1} (A^T)^r C^T C A^i B B^T (A^T)^{i-r-1} \tag{18}$$

C Multiple Input Multiple Output

In this section we discuss the extension of our results to the case of *Multiple Input Multiple Output* (MIMO) systems. In what follows we denote the input dimension by n, and the output dimension, m.

Consider an RNN with hidden width d, input sequence $\{X_t\}_{t=1}^{\infty} \subset \mathbb{R}^n$ representing a sequence of n-dimensional inputs, denote the i^{th} column of X by X_i. The model produces outputs $\{\hat{\mathbf{y}}_t\}_{t=1}^{\infty} \subset \mathbb{R}^m$ through the following update equations:

$$\hat{\mathbf{y}}_t = Cs_t, \qquad s_{t+1} = As_t + BX_{t+1}, \tag{19}$$

where $A \in \mathbb{R}^{d\times d}$, $B \in \mathbb{R}^{d\times n}$ and $C \in \mathbb{R}^{m\times d}$ are the learned parameters, and $\{s_t\}_{t=1}^{\infty} \subset \mathbb{R}^d$ are the resulting hidden states, where by assumption $s_0 = 0$. Given an input sequence $X \in \mathbb{R}^{n\times k}$, a memoryless MIMO teacher corresponds to $W^* \in \mathbb{R}^{m\times n}$, such that $\mathbf{y} = W^* X_k$.

In the main paper we develop an expression for the population loss for the case of SISO. We provide here a MIMO version of the lemma.

Lemma C.1. *Assume $X \in \mathbb{R}^{n\times k}$, $X \sim \mathcal{D}$ such that $\mathbb{E}_{\mathcal{D}}[XX^\top] = I_n$ and $\mathbb{E}_{\mathcal{D}}[X] = 0$. Then, given a memoryless teacher RNN, the loss for the student RNN satisfies:*

$$\mathbb{E}_{X,\mathbf{y}}\left[\frac{1}{2}\left\|RNN(X) - \mathbf{y}\right\|_F^2\right] = \frac{1}{2}\sum_{i=1}^{k-1}\left\|CA^{k-i}B\right\|_F^2 + \frac{1}{2}\left\|CB - W^*\right\|_F^2 \tag{20}$$

Proof. The proof is given in C.1. □

In the main paper we show that when the sequence length is greater than the hidden dimension, ($k > d$), extrapolation is guaranteed by showing that $\forall j \in \mathbb{N}$ it holds that $CA^jB = 0$. The analysis in the main paper is not dependent on the dimensions of B and C and therefore applies to the MIMO setting as-is.

Following the analysis of extrapolation when learning with long sequences, we show that when $k < d$, there exists solutions that attain zero loss but do not extrapolate w.r.t. a memoryless teacher. The proof uses the following parameters,

$$A = \begin{pmatrix} 0 & 0 & \dots & 0 & 0 & 1 \\ 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 & 0 \\ & & \ddots & & & \\ 0 & 0 & \dots & 1 & 0 & 0 \\ 0 & 0 & \dots & 0 & 1 & 0 \end{pmatrix} \in \mathbb{R}^{d,d} \ , \quad B = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \in \mathbb{R}^{d,1} \ , \quad C = (w^*, 0, \dots, 0) \in \mathbb{R}^{1,d}.$$

In order to apply for MIMO, the parameters B and C need to be padded with zeros to form,

$$B = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{pmatrix} \in \mathbb{R}^{d,n} \ , \quad C = \begin{pmatrix} w^* & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{pmatrix} \in \mathbb{R}^{m,d}.$$

and the same arguments apply.

In the main paper we show GD has implicit bias towards memoryless solutions under standard initialization schemes. Here we show that this result extends to the MIMO case, under the additional conditions that $m = n$ and W^* is symmetric and nonnegative.

The SISO proof follows two steps. The first shows that at any time step of GD, $B_t = C_t^T$ and A_t is symmetric. For the first part of the proof to apply for the MIMO setting, the dimensions of B and C must allow $B_0 = C_0^T$ which implies $n = m$.

The gradient updates in the MIMO case are similar to those of SISO (see Appendix A) and are given by:

$$\frac{\partial \mathcal{L}}{\partial B} = \sum_{i=1}^{k-1} (A^T)^i C^T C A^i B + C^T(CB - W^*), \tag{21}$$

$$\frac{\partial \mathcal{L}}{\partial C} = \sum_{i=1}^{k-1} C A^i B B^T (A^T)^i + (CB - W^*)B^T, \tag{22}$$

$$\frac{\partial \mathcal{L}}{\partial A} = \sum_{i=1}^{k-1} \sum_{r=0}^{i-1} (A^T)^r C^T C A^i B B^T (A^T)^{i-r-1}. \tag{23}$$

The same inductive argument from the main text applies here with the distinction that in order for the RHS of (21) and (22) to satisfy

$$C_t^\top (C_t B_t - W^*) = \left[(C_t B_t - W^*) B_t^\top\right]^\top$$

the matrix W^* must be symmetric.

For the second part of the proof, we use the fact that $B = C^\top$ and $A = A^\top$ to show that at convergence $CA^jB = 0$ for all $j \in \mathbb{N}$. Recalling that $k > 2$, consider the optimized loss:[3]

$$\mathcal{L}(A, B, C) = \sum_{i=3}^{k-1} \|CA^iB\|_F^2 + \|CA^2B\|_F^2 + \|CAB\|_F^2 + \|CB - W^*\|_F^2 \tag{24}$$

[3] The leftmost term is zero by definition if $k = 3$.

Any solution minimizing (i.e., bringing to zero) the above must satisfy:

$$CA^2B = 0 \in \mathbb{R}^{n\times n}. \tag{25}$$

By the assumption of the theorem we have that GD converges to a minimizing solution and therefore satisfies Equation (11). Also, by the first part of the proof, we know that A is symmetric and therefore orthogonally diagonalizable, meaning there exist an orthogonal matrix $V \in \mathbb{R}^{d,d}$ and a diagonal matrix $D \in \mathbb{R}^{d,d}$ such that $A = VDV^\top$. We can thus write $A^2 = VD^2V^\top$, and since $B = C^\top$ (by the first part of the proof),

Equation (11) implies:

$$CA^2B = B^\top A^2 B = B^\top V D^2 V^\top B = 0 \in \mathbb{R}^{n\times n}.$$

Denote $U = V^\top B$, the above can be written as

$$B^\top V D^2 V^\top B = U^\top D^2 U = 0 \in \mathbb{R}^{n\times n}.$$

The above matrix is element-wise zero, in particular its diagonal elements should be zero, implying for all i:

$$\left[U^\top D^2 U\right]_{ii} = \sum_{s=1}^{d} U_{si}^2 \lambda_s^2 = 0 \tag{26}$$

Since Equation (26) is a sum of non-negative elements that sum to zero, each of them should be zero. Furthermore, for any s, it must hold that $U_{si}^2\lambda_s^2 = 0$ and therefore we have the complementary slackness result:

$$U_{si}\lambda_s = 0 \ \ \forall i, s \tag{27}$$

The fact that the model extrapolates follows directly from the observation above. Consider any $p \in \mathbb{N}$. Then the corresponding element in the impulse response is given by:

$$CA^pB = B^\top V D^p V^\top B = U^\top D^p U$$

which can be written as

$$\left[U^\top D^p U\right]_{ij} = \sum_{s=1}^{d} U_{si} U_{sj} \lambda_s^p$$

From Equation (27) we conclude that the above is zero and thus $CA^pB = 0$ (i.e., this part of the matrix impulse response is zero). This is precisely the condition for perfect extrapolation (see main text and recall that $CB - W^* = 0$ because of optimality of GD) and thus the result follows.

C.1 Population Loss for MIMO

Proof for Lemma C.1.

$$\mathbb{E}_{X,\mathbf{y}}\left[\frac{1}{2}\left\|RNN(X) - \mathbf{y}\right\|_F^2\right] = \frac{1}{2}\mathbb{E}\left[\left\|\sum_{i=1}^{k} CA^{k-i}BX_i - W^*X_k\right\|_F^2\right]$$

For two general matrices Q, R, the loss in terms of the trace operator is given by

$$\begin{aligned}\|Q-R\|_F^2 &= tr((Q-R)^T(Q-R)) \\ &= tr\left(Q^TQ - Q^TR - R^TQ + R^TR\right) \\ &= tr(Q^TQ) - tr(Q^TR) - tr(R^TQ) + tr(R^TR) \\ &= tr(Q^TQ) - 2tr(Q^TR) + tr(R^TR)\end{aligned} \tag{28}$$

where the transitions rely on the properties of the trace operator. We can now handle each term separately, denote $W_i = CA^{k-i}B$, assigning $Q = \sum_{i=1}^{k} W_i X_i$, the LHS term, $tr(Q^T Q)$, amounts to

$$\begin{aligned} tr\left(\left(\sum_{i=1}^{k} X_i^T W_i^T\right)\left(\sum_{j=1}^{k} W_j X_j\right)\right) &= tr\left(\sum_{i=1}^{k}\sum_{j=1}^{k} X_i^T W_i^T W_j X_j\right) \\ &= \sum_{i=1}^{k}\sum_{j=1}^{k} tr\left(X_i^T W_i^T W_j X_j\right) \\ &= \sum_{i=1}^{k}\sum_{j=1}^{k} tr\left(W_i^T W_j X_j X_i^T\right) \end{aligned}$$

Taking the expectation of IID samples X_i, X_j,

$$\mathbb{E}\left[tr\left(W_i^T W_j X_j X_i^T\right)\right] = tr\left(W_i^T W_j \mathbb{E}\left[X_j X_i^T\right]\right) = \begin{cases} 0 & i \neq j \\ tr(W_i^T W_j) & i = j \end{cases}$$

putting together, the LHS term amounts to

$$\mathbb{E}\left[tr(Q^T Q)\right] = \sum_{i=1}^{k} tr(W_i^T W_i) = \sum_{i=1}^{k} \|W_i\|_F^2 = \sum_{i=1}^{k} \left\|CA^{k-i}B\right\|_F^2 \tag{29}$$

For the second term, $tr(Q^T R)$, we have

$$tr\left(\sum_{i=1}^{k} X_i^T W_i^T W^* X_k\right) = \sum_{i=1}^{k} tr\left(X_i^T W_i^T W^* X_k\right) = \sum_{i=1}^{k} tr\left(W_i^T W^* X_k X_i^T\right)$$

Taking the expectation, for every $i \neq k$, $\mathbb{E}\left[X_k X_i^T\right] = 0$, and for $i = k$, $\mathbb{E}\left[X_k X_k^T\right] = I_n$. Therefore the middle term amounts to

$$\mathbb{E}\left[tr(Q^T R)\right] = tr\left(W_k^T W^*\right) \tag{30}$$

Finally, the RHS is given by

$$\mathbb{E}\left[tr(R^T R)\right] = tr\left((W^*)^T W^*\right) \tag{31}$$

where we again use the linearity and cyclic properties of the trace operator as well as $\mathbb{E}\left[X_k X_k^T\right] = I_n$.

Putting the computed terms, (29) (30) (31), back into Equation (28), we have

$$\mathbb{E}\left[\left\|\sum_{i=1}^{k} CA^{k-i}BX_i - W^* X_k\right\|_F^2\right] = \sum_{i=1}^{k} \left\|CA^{k-i}B\right\|_F^2 - 2tr\left(W_k^T W^*\right) + tr\left((W^*)^T W^*\right)$$

Note that $W_k = CA^{k-k}B = CB$, the above can be written as

$$\sum_{i=1}^{k-1} \left\|CA^{k-i}B\right\|_F^2 + \|CB\|_F^2 - 2tr\left((CB)^T W^*\right) + \|W^*\|_F^2 \tag{32}$$

which can further be written as

$$\sum_{i=1}^{k-1} \left\|CA^{k-i}B\right\|_F^2 + \|CB - W^*\|_F^2 \tag{33}$$

to conclude the proof. □

D Additional Experiments

In the paper we show that GD has an inductive bias towards memory-less models. Namely, if the training data can be fit with a memory-less model, gradient descent with symmetric initialization will extrapolate well. Here we ask the more general question: if data is generated by a low dimensional LinearRNN (i.e., with low dimensional A), will GD extrapolate well. Namely, we ask whether gradient descent with symmetric initialization has an inductive bias towards low-dimensional systems.

Clearly, if the training sequences are shorter than the dimension of the ground-truth A, we should not expect to extrapolate well (since the short sequence does not capture the full behavior of the true model).

In what follows, we use d^* to denote the dimension of A for the ground-truth system. We let k denote the length of the training data. Based on our discussion above, we would expect the following two regimes:

- Good extrapolation for $k \geq d^*$, since in this case there are sufficient observations to identify a low dimensional model and the data can be fit by this model. Moreoever, if GD with the said initialization scheme is indeed biased towards low order models, it will converge to the model with dimension d^*.
- Bad extrapolation for $k < d^*$ since in this case the first k time units are insufficient to uniquely identify the ground-truth model.

We explore the above question using three different models, LinearRNN, GRU and LSTM with standard Xavier initialization. For all experiments, we set $k = 5$, $d = 200$ and $d^* = 1, 2, 4, 6, 8$.

The architecture of the teacher is a LinearRNN with varying d^*. For each trained model, we estimate the extrapolation MSE as the average error of the model on sequence lengths $6, 7, 8, 9, 10$ (i.e., lengths it was not trained on). Figure 5 shows extrapolation error as a function of d^*. It can be seen that results are in line with the two regimes mentioned above. Namely, up to some point (roughly $d^* = k$), the model extrapolates well, and beyond this point extrapolation deteriorates.

These results suggest that gradient descent with standard initialization is indeed biased towards models with smaller dimensionality d. Furthermore, this happens for both linear and non linear models.

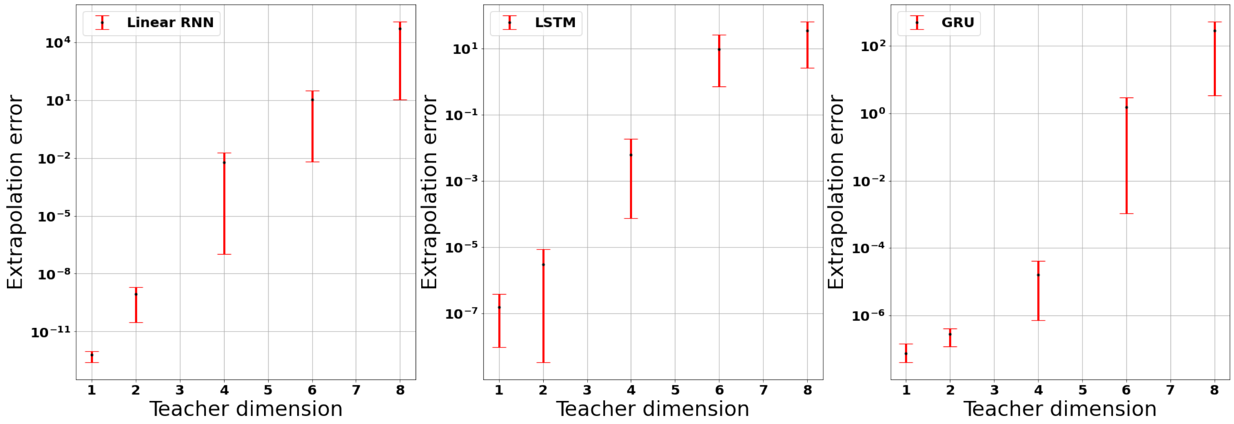

Figure 5: Extrapolation error as a function of the teacher dimension. The figure shows that when $d^* < k$ there is good extrapolation indicating inductive bias towards low dimensional model. On the other hand for $d^* > k$ extrapolation fails, which is expected as the training examples are not long enough to reveal the teacher dynamics for sequences with length greater than k.