APPENDIX

We begin this appendix by stating a precise condition on the subclass of RNNs to which the algorithms OK and KF-RTRL can be applied (Section A.0.1) and show that in a setting as given by the algorithm KTP, the concept of a minimum-variance approximator is not well defined (Section A.0.2). We also give details on how the memory requirement of KTP can be kept at $O(n)$ (Section A.0.3).

In the remaining two sections, we prove Theorems 1 and 2 from the paper (Section A.1) and provide additional experiments (Section A.2).

A.0.1. Subclass of RNNs for OK and KF-RTRL

Recall, that similarly to KF-RTRL (Mujika et al., 2018), we restrict our attention to RNNs for which the term F_t (see description of RTRL in the paper) can be factored as $F_t = h_t \otimes D_t$. We restate the condition given in (Mujika et al., 2018):

Lemma A.1. *Assume that the learnable parameters θ are a set of matrices $W^1, ..., W^r$, let $\hat{h}_{t-1}$ be the hidden state h_{t-1} concatenated with the input x_t and let $z^k = \hat{h}_{t-1}W^k$ for $k = 1, ..., r$. Assume that h_t is obtained by point-wise operations over the z^k's, that is, $(h_t)_j = f(z_j^1, ..., z_j^r)$. Let $D^k \in \mathbb{R}^{n\times n}$ be the diagonal matrix defined by $D^j_{kk} = \frac{\partial (h_t)_j}{\partial z_j^k}$, and let $D = (D^1|\dots|D^r)$. Then, it holds that $\frac{\partial h_t}{\partial \theta} = \hat{h}_{t-1} \otimes D$.*

We refer the reader to (Mujika et al., 2018) for the simple proof. There, it is also shown that this class of RNNs includes standard RNNs and the popular LSTM and RHN architectures.

A.0.2. No Optimal Approximation for 3-Tensors

Here, we show that the concept of a minimum-variance approximator is ill-defined for a situation as encountered by KTP. We also explain how similar problems are related to NP-hardness.

For the next lemma, we slightly adapt an example from (Hillar & Lim, 2013) based on an exercise in (Knuth, 1997). We first recall some notions related to 3-tensors. For $a, b, c \in \mathbb{R}^n \setminus \{0\}$, we call $a \otimes b \otimes c$ a rank-1 tensor. In general, the rank of a tensor X is the minimum number k, so that X can be written as the sum of k rank-1 tensors. The following is related to the fact that the set of rank-2 tensors is not closed.

Lemma A.2. *For $i = 1, 2, 3$, let $x_i, y_i \in \mathbb{R}^n$ so that the pairs x_i, y_i are linearly independent and define $X = x_1 \otimes x_2 \otimes y_3 + x_1 \otimes y_2 \otimes x_3 + y_1 \otimes x_2 \otimes x_2$. Then, there are rank-2 approximators of X with arbitrarily small variance, but no rank-2 approximator of variance 0. Thus, the concept of a 'minimum-variance' rank-2 approximator of X is ill-defined.*

Proof. The statement that there is no rank-2 approximator with variance 0 is equivalent to X having rank larger than 2, the details of which we leave to the reader.

Now, let $s \in \{\pm 1\}$ be a uniformly random sign and for each positive integer n define a random variable

$$X_n = x_1 \otimes x_2 \otimes (y_3 - s \cdot n x_3) + \left(s \cdot x_1 + \frac{1}{n} y_1\right) \otimes \left(x_2 + s \cdot \frac{1}{n} y_2\right) \otimes n x_3$$

A simple calculation shows $X_n = X + s \cdot \frac{1}{n} (y_1 \otimes y_2 \otimes x_3)$. From this we conclude $\mathrm{E}[X_n] = X$ and $\mathrm{Var}[X_n] \to 0$ as $n \to \infty$, finishing the proof of the lemma. □

In addition to the concept of a minimum-variance approximator not being well defined, we note that finding 'good' approximators (which might still be possible given the above lemma) seems to be closely related to finding the rank of a 3-tensor, which is, like many other problems for 3-tensors (Hillar & Lim, 2013), NP-hard.

A.0.3. Memory of KTP

Here, we describe how the memory requirement of KTP can be kept at $O(n)$ despite the need of calculating $H_t b$ (see description of KTP in the paper, $H_t \in \mathbb{R}^{n\times n}, b \in \mathbb{R}^{n\times 1}$). One way to do this was already used in (Tallec & Ollivier, 2017). Recall $h_t = f(x_t, h_{t-1}, \theta)$ and $H_t = \frac{\partial h_t}{\partial h_{t-1}}$. Therefore, $H_t b$ is a directional derivative of h_t in the direction of b, which implies

$$H_t b = \lim_{\epsilon \to 0} \frac{f(x_t, h_{t-1} + \epsilon b, \theta) - f(x_t, h_{t-1}, \theta)}{\epsilon \|b\|}. \quad (1)$$

To evaluate $H_t b$ it therefore suffices to choose a small ϵ and evaluate the expression above. The expression above can be calculated together with the forward pass of the RNN, so that no additional batch-memory is needed.

A.1. Proofs

In this section, we prove Theorems 1 and 2 from the main paper. Their statements and the related algorithms are restated below for convenience.

The outline of this section is as follows. We start in Section A.1.1 by introducing some notation and reviewing the concept of a Singular Value Decomposition along with some of its properties, which will be useful later. In Section A.1.2 we prove Theorem A.1 assuming correctness of Algortihm A.2 and Theorem A.2. These are then jointly proved in Section A.1.3.

Theorem A.1. *Let G be an $(r+1)$-Kronecker-Sum and let G' be the random r-Kronecker-Sum constructed by OK. Then G' unbiasedly approximates G. Moreover, for any random r-Kronecker-Sum Y of the same format as G' which satisfies $\mathrm{E}[Y] = G$, it holds that $\mathrm{Var}[Y] \geq \mathrm{Var}[G']$.*

Theorem A.2. *Given $C \in \mathbb{R}^{m\times n}$ and $r \leq \min\{m,n\}$, one can (explicitly) construct an unbiased approximator C' of C, so that C' always has rank at most r, and so that C' has minimal variance among all such unbiased, low-rank approximators. This can be achieved asymptotically in the same runtime as computing the SVD of C.*

Algorithm A.1 The OK approximation

Input: Vectors $u_1, \dots, u_{r+1}$ and matrices $A_1, \dots, A_{r+1}$
Output: Random vectors $u'_1, \dots, u'_r$ and matrices $A'_1 \dots A'_r$, such that $\sum_{i=1}^{r} u'_i \otimes A'_i$ is an unbiased, minimum-variance approximator of $\sum_{i=1}^{r+1} u_i \otimes A_i$

/*Rewrite in terms of orthonormal basis (onb)*/
$v_1, \dots, v_{r+1} \leftarrow$ onb of span$\{u_1, \dots, u_{r+1}\}$
$B_1, \dots, B_{r+1} \leftarrow$ onb span$\{A_1, \dots, A_{r+1}\}$
for $1 \leq i, j \leq r+1$ **do**
  $L_{i,j} \leftarrow \langle v_i, u_j \rangle, \quad R_{i,j} \leftarrow \langle B_i, A_j \rangle$
end for

/*Find optimal rank r approximation of matrix C */
$C \leftarrow LR^T$
$(L', R') \leftarrow Opt(C)$ {see Algorithm A.2 for $Opt(\cdot)$}

/*Generate output*/
for $1 \leq j \leq r$ **do**
  $u'_j \leftarrow \sum_{i=1}^{r+1} L'_{i,j} v_i, \quad A'_j \leftarrow \sum_{i=1}^{r+1} R'_{i,j} B_i$
end for

A.1.1. Preliminaries

A.1.1.1. Notation

We denote matrices by upper case letters, e.g. $C \in \mathbb{R}^{m\times n}$, and denote their entries by indexing this letter, e.g. $C_{i,j}$ where $1 \leq i \leq m$ and $1 \leq j \leq n$. For vectors $s, z_1, \dots, z_n \in \mathbb{R}^{n\times 1}$, we denote by $s \odot z_i$ the pointwise product and by $Z = (z_1, \dots, z_r) \in \mathbb{R}^{n\times r}$ the matrix whose i-th column is z_i. We write Id_n for the identity matrix of dimension n.

For a random variable $X' \in \mathbb{R}^{n\times m}$ and some fixed value $X \in \mathbb{R}^{n\times m}$, we say that X' is an *unbiased approximator* of X, if $\mathrm{E}[X'] = X$. We further call X' a *rank-r approximator*, if X always (with probability 1) has rank *at most* r. We will usually name random variables by adding a " ' " to the deterministic quantity they represent. The variance of X' is defined as $\mathrm{Var}[X'] = \mathrm{E}[\|X' - \mathrm{E}[X']\|^2]$, where we use the Frobenius norm and the corresponding inner product $\langle X_1, X_2 \rangle = \mathrm{Tr}(X_1^T X_2)$ throughout.

Algorithm A.2 $Opt(C)$

Input: Matrix $C \in \mathbb{R}^{(r+1)\times(r+1)}$
Output: Random matrices $L', R' \in \mathbb{R}^{(r+1)\times r}$, so that $L'R'^T$ is an unbiased, min-variance approximator of C

/* Reduce to diagonal matrix D*/
$(D, U, V) \leftarrow \mathrm{SVD}(C)$
$(d_1, \dots, d_{r+1}) \leftarrow$ diagonal entries of D

/* Find approximator ZZ^T for small d_i ($i \geq m$)*/
$m \leftarrow \min\{i \colon (r-i+1)d_i \leq \sum_{j=i}^{r} d_j\}$
$s_1 \leftarrow \sum_{i=m}^{r+1} d_i, \quad k \leftarrow r - m + 1$
$z_0 \leftarrow \left(\sqrt{1 - \frac{d_m k}{s_1}}, \dots, \sqrt{1 - \frac{d_{r+1} k}{s_1}}\right)^T \in \mathbb{R}^{(k+1)\times 1}$
$z_1, \dots, z_k \leftarrow$ so that $z_0, z_1, \dots, z_k$ is an onb of $\mathbb{R}^{(k+1)\times 1}$
$s \leftarrow$ vector of $k+1$ uniformly random signs
$Z \leftarrow \sqrt{\frac{s_1}{k}} \cdot (s \odot z_1, \dots, s \odot z_k)$ {*pointwise product* $\odot$}

/* Initialise L', R' to approximate D*/
$L', R' \leftarrow \mathrm{diag}(\sqrt{d_1}, \dots, \sqrt{d_{m-1}}, Z)$ {*Block-diagonal*}

/*Approximate $C = UDV^T$*/
$L' \leftarrow UL', \quad R' \leftarrow VR'$

A.1.1.2. Singular Value Decomposition

The Singular Value Decomposition (SVD) is a standard tool from Linear Algebra, and has countless applications in and outside Machine Learning. We refer to the textbook (Golub & Van Loan, 1996) for an introduction and (Cline & Dhillon, 2006) for a review of algorithms to compute the SVD.

To simplify notation, we restrict our attention to square matrices. The concepts straightforwardly generalise to arbitrary matrices, we refer to the above mentioned textbook.

For a matrix $C \in \mathbb{R}^{n\times n}$, the Singular Value Decomposition (SVD) of C is a triple of matrices $U, V, D \in \mathbb{R}^{n\times n}$ satisfying $C = UDV^T$, so that U, V are orthogonal and $D = \mathrm{diag}(d_1, \dots, d_n)$ is a diagonal matrix with non-negative, non-decreasing entries. The existence of a SVD is a standard result in Linear Algebra.

The values d_i are refered to as *singular values* of C and are uniquely determined by C. In fact they are the square-roots of the eigenvalues of CC^T. The number of non-zero singular values of C equals the rank of C. The columns of U, V are called left, respectively right, *singular vectors*. They correspond to eigen-bases of the matrices CC^T and C^TC, respectively. The singular vectors are uniquely determined if and only if the singular values are pairwise distinct. If a singular value d_i appears more than once, the corresponding singular vectors form an orthonormal basis of a subspace uniquely determined by C and d_i (corresponding to an eigen-space of CC^T or C^TC).

One of the important applications of the SVD is the follow-

ing result, known as the Eckart-Young Theorem (Eckart & Young, 1936).

Theorem A.3 (Eckart-Young Theorem)**.** *Let $C \in \mathbb{R}^{n\times n}$ be a matrix with singular values $d_1, \dots, d_n$ and let $X \in \mathbb{R}^{n\times n}$ be a fixed (non-random) matrix of rank at most r.*
Then, $\|C - X\|^2 \geq \sum_{i=r+1}^{n} d_i^2$ and equality is achieved if and only if X is of the form $X = U \cdot \mathrm{diag}(d_1, \dots, d_r, 0 \dots, 0) \cdot V^T$ for an arbirarty singular value decomposition $C = UDV^T$ of C.

Noting that every SVD of the identity matrix Id_n is of the form $\mathrm{Id}_n = U \cdot \mathrm{Id}_n \cdot U^T$ for some orthogonal matrix U, we can deduce the following observation.

Observation A.1. *Let $X \in \mathbb{R}^{n\times n}$ be a fixed (non-random) matrix of rank at most r. Then $\|X - \mathrm{Id}_n\|^2 \geq n - r$ and equality is achieved if and only if X is of the form $X = \sum_{i=1}^{r} u_i u_i^T$, where the u_i are orthonormal vectors in $\mathbb{R}^{n\times 1}$.*

A.1.2. Proof of Theorem A.1

Let us first restate the objective encountered by OK. We are given vectors $u_1, \dots, u_{r+1} \in \mathbb{R}^{1\times n}$ and matrices $A_1, \dots, A_{r+1} \in \mathbb{R}^{n\times n}$ and we want to construct random vectors $u'_1, \dots, u'_r$ and matrices $A'_1, \dots, A'_r$ such that the r-Kronecker-Sum $G' = \sum_{i=1}^{r} u'_i \otimes A'_i$ is an unbiased approximator of the $(r+1)$-Kronecker-Sum $G = \sum_{i=1}^{r+1} u_i \otimes A_i$, and such that G' has minimum variance.

Theorem A.1 follows directly from the following lemma.

Lemma A.3. *Assume that Algorithm A.2 gives a minimum-variance rank-r approximation $C' = L'R'^T$ of the matrix $C = L^T R$ as constructed by Algorithm A.1. Then Algorithm A.1 gives a minimum-variance unbiased approximator $G' = \sum_{i=1}^{r} u'_i \otimes A'_i$ of $G = \sum_{i=1}^{r+1} u_i \otimes A_i$.*

Proof. The first important observation is that for an optimal approximator G', the random variables u'_i are always elements of $\mathrm{span}\{u_1, \dots, u_{r+1}\}$ and the A'_i are always elements of $\mathrm{span}\{A_1, \dots, A_{r+1}\}$. Otherwise, we could simply take the u'_i and project them (orthogonally) onto $\mathrm{span}\{u_1, \dots, u_{r+1}\}$ (and similarly for the A'_i) to obtain a new unbiased approximator G'' of G which has less variance than G'.

From this observation, it follows that the u'_i are a (random) linear combination of the u_i (and similarly for A'_i). In order to be able to get simple closed-form expressions for the variance of G', we now choose orthonormal bases of the spaces $\mathrm{span}\{u_1, \dots, u_{r+1}\}$ and $\mathrm{span}\{A_1, \dots, A_{r+1}\}$, let us denote them by $v_1, \dots, v_{r+1}$ and $B_1, \dots, B_{r+1}$ respectively. Now define matrices $L, R \in \mathbb{R}^{(r+1)\times(r+1)}$ by setting $L_{i,j} := \langle v_i, u_j \rangle$ and $R_{i,j} := \langle B_i, A_j \rangle$. Especially, we have $u_j = \sum_{i=1}^{r+1} L_{i,j} v_i$ (and an analogous equation for A_j). Observe that the matrix $C := LR^T$ has coefficients representing G in terms of an orthogonal bases, more precisely

$$G = \sum_{1\leq i,j\leq r+1} C_{i,j} \left(v_i \otimes B_j\right), \tag{2}$$

where it is not difficult to see that the $(v_i \otimes B_j)_{i,j}$ are orthonormal.

As noted above, the u'_i, A'_i forming G' are linear combinations of u_i, A_i respectively, so they can be written in terms of the ONBs u'_i, A'_i. Let us record all the corresponding coefficients in (random) matrices $L', R' \in \mathbb{R}^{(r+1)\times r}$ meaning that $L'_{i,j} = \langle v_i, u'_j \rangle$ (or equivalently $u'_j = \sum_i L'_{i,j} v_i$) and $R'_{i,j} = \langle B_i, A_j \rangle$. The same calculations as above then show that, for the matrix $C' := L'R'^T$, we have

$$G' = \sum_{1\leq i,j\leq r+1} C'_{i,j} \left(v_i \otimes B_j\right). \tag{3}$$

Now, with linearity of expectation and independence of $(u'_i \otimes A'_j)_{i,j}$, we can conclude from (2), (3) that

$$\mathrm{E}[G'] = G \quad \Leftrightarrow \quad \mathrm{E}[C'] = C.$$

From the orthonormality of $(u'_i \otimes A'_j)_{i,j}$ we moreover obtain

$$\mathrm{Var}[G'] = \mathrm{E}\left[\sum_{i,j} \left(C'_{i,j} - \mathrm{E}[C'_{i,j}]\right)^2\right] = \mathrm{Var}[C'].$$

Combined, the last two statements show that finding a minimum variance, unbiased approximator G' of G is equivalent to finding a minimum variance, unbiased rank-r approximator $C' = L'R'^T$ of C. This finishes the proof of the lemma. □

A.1.3. Proof of Theorem A.2, Correctness of Algorithm A.2

Here, we prove Theorem A.2. The calculations carried out in the proof precisely match the ones carried out by Algorithm A.2, so that its correctness is an immediate consequence of the proof given below. Theorem A.2 is a direct consequence of Theorems A.4 and A.5 below.

To simplify notation, we shall assume that C has dimension $C \in \mathbb{R}^{n\times n}$, the more general case $C \in \mathbb{R}^{m\times n}$ can be proved in the same way without additional complications. The outline of the proof is as follows: We will first reduce finding an unbiased rank-r approximator of C to the problem of finding an unbiased, rank-r approximator of a diagonal matrix D using SVD. We will then use a duality argument to give a sufficient condition for an approximator of D to have minimal variance and conclude by constructing an approximator fulfilling this condition.

A.1.3.1. Reducing the problem to diagonal matrices

In this subsection, we give the simple explanation of how finding an optimal rank-r approximator C' of $C \in \mathbb{R}^{n\times n}$ can be reduced to finding an optimal rank-r approximator D' of a diagonal matrix $D = \operatorname{diag}(d_1, \dots, d_n)$ with non-negative entries.

Lemma A.4. *Let C be as above and let $UDV^T = C$ be a SVD of C. Then, given an unbiased rank-r approximator D' of D, it holds that $C' = UD'V^T$ is an unbiased approximator of C. Moreover, C' is optimal if D' is optimal.*

Proof. The proof is almost immediate. The fact that C' unbiasedly approximates C follows from the fact that D' unbiasedly approximates D, linearity of expectation and $C = UDV^T$. Note that given C', we can write $D' = U^T C' V$, so that there is a one-to-one correspondence between C' and D'. Since U, V are orthogonal, it follows that $\operatorname{Var}[C'] = \operatorname{Var}[D']$, so that C' is optimal if and only if D' is optimal. □

A.1.3.2. Optimally approximating diagonal matrices

In this subsection, we construct a minimum-variance, unbiased approximator for diagonal matrices $D = \operatorname{diag}(d_1, \dots, d_n)$ with non-negative, non-increasing entries. The first step is giving a sufficient condition for any such approximator to be optimal. The second step is the construction of an unbiased approximator satisfying this condition.

Sufficient Optimality Condition

For stating our condition, we first need some notation. Let $D = \operatorname{diag}(d_1, \dots, d_n)$ be a diagonal matrix such that $d_1 \geq d_2 \dots \geq d_n \geq 0$. Let

$$m = \min\left\{ i \colon (r-i+1)d_i \leq \sum_{j=i}^{n} d_j \right\}, \quad k = r-m+1.$$

We can already give some intuition on the meaning of m. We will see later that it is defined so that the first $m-1$ diagonal entries are so large, that an optimal approximation consists of approximating D by

$$D' = \begin{pmatrix} d_1 & \dots & 0 & \\ 0 & \ddots & 0 & 0 \\ 0 & \dots & d_{m-1} & \\ & 0 & & D'_2 \end{pmatrix} \tag{4}$$

where D'_2 is an optimal, unbiased rank-k approximator of $\operatorname{diag}(d_m, \dots, d_n)$ (note that if the rank of D'_2 was larger then k then the rank of D' would be larger than r). In other words, some large diagonal entries are kept deterministically and only smaller ones are 'mixed' into a matrix of lower rank.

Defining

$$s_1 := \sum_{j=m}^{n} d_j \quad \text{and} \quad s_2 := \sum_{j=m}^{n} d_j^2$$

we can state our optimality condition.

Theorem A.4. *Let D, m, k, s_1, s_2 be as above. Then, any unbiased rank-r approximator D' of D satisfies*

$$\operatorname{Var}[D'] \geq \frac{s_1^2}{k} - s_2.$$

Equality is achieved if and only if, in addition to being unbiased, D' satisfies the following two conditions:

1. *D' is of the form given in equation (4), such that*
2. *D'_2 always (with probability 1) satisfies*

$$\left\| \frac{k}{s_1} D'_2 - \operatorname{Id}_{n-(m-1)} \right\|^2 = n - r.$$

Before proving the theorem, let us explain Condition 2. of the theorem. Note that D'_2 has (square) dimension $n-(m-1)$ and must have rank at most $k = r-(m-1)$. Thus, by the Eckart-Young Theorem

$$\left\| \frac{k}{s_1} D'_2 - \operatorname{Id} \right\|^2 \geq \big((n-(m-1)) - k)\big) = n - r.$$

In other words, the approximator D'_2 is optimal, if and only if $\frac{k}{s_1} D'_2$ is as close to Id as it can be (given its rank).

Proof of Theorem A.4. As mentioned before, we will prove the theorem using a duality argument. Let D' be an unbiased rank-r approximator of D. Observe that for any matrix $B \in \mathbb{R}^{n\times n}$, due to linearity of expectation, it holds that $\mathrm{E}\big[\operatorname{Tr}[(D'-D)B]\big] = 0$. We can therefore write

$$\begin{aligned}
\operatorname{Var}[D'] &= \mathrm{E}\left[\operatorname{Tr}\left[(D'-D)(D'-D)^T \right] \right] \\
&= \mathrm{E}\left[\operatorname{Tr}\left[(D'-D)(D'-D)^T \right] + 2\operatorname{Tr}\left[(D'-D)B^T \right] \right] \\
&= \mathrm{E}\left[\operatorname{Tr}\left[(D'-D+B)(D'-D+B)^T \right] - \operatorname{Tr}\left[BB^T \right] \right] \\
&= \mathrm{E}\left[\|D' - (D-B)\|^2 \right] - \mathrm{E}\left[\|B\|^2 \right] \\
&\geq \min_{\substack{X \in \mathbb{R}^{n\times n}, \\ \operatorname{rank}(X) \leq r}} \Big(\|X - (D-B)\|^2 \Big) - \|B\|^2. \qquad (5)
\end{aligned}$$

Thus, for any $B \in \mathbb{R}^{n\times n}$, we get the lower bound from equation (5) on the variance of D'. We now choose B to maximize the lower bound. Namely, we choose B so that

$$D - B = \begin{pmatrix} d_1 & \dots & 0 & \\ 0 & \ddots & 0 & 0 \\ 0 & \dots & d_{m-1} & \\ & 0 & & \frac{s_1}{k}\mathrm{Id}_{r-(m-1)} \end{pmatrix}.$$

This implies

$$\begin{aligned} \|B\|^2 &= \sum_{j=m}^{n} \left(d_j - \frac{s_1}{k}\right)^2 \\ &= s_2 - 2\frac{s_1^2}{k} + (n-m+1)\frac{s_1^2}{k^2}. \end{aligned} \tag{6}$$

Moreover, note that the diagonal entries of $D - B$ are non-increasing. For $m = 1$ this is immediate, for $m > 1$, the definition of m implies $(r-(m-1)+1)d_{m-1} > \sum_{j=m-1}^{n} d_j$ giving $d_{m-1} > \frac{s_1}{k}$ showing that diagonal entries are indeed non-decreasing. Thus, by the Eckart-Young Theorem, we have

$$\min_{\substack{X\in\mathbb{R}^{n\times n},\\ \mathrm{rank}(X)\le r}} \left(\|X - (D-B)\|^2\right) \ge (n-r)\frac{s_1^2}{k^2}. \tag{7}$$

We now obtain the statement of the theorem by plugging (6) and (7) into (5) and recalling $k = r - m + 1$

$$\begin{aligned} \mathrm{Var}[D'] &\ge \frac{s_1^2}{k^2}\left((n-r) + 2k - (n-m+1)\right) - s_2 \\ &= \frac{s_1^2}{k} - s_2. \end{aligned}$$

Note that equality is achieved if and only if it is always achieved in (7). Since $d_{m-1} > \frac{s_1}{k}$, it follows from the Eckart-Young Theorem that equality in (7) is achieved if an only if the Conditions 1 and 2 from the theorem hold. □

Construction of Approximator fulfilling the Optimality Condition

We now show that the condition from Theorem A.4 can be satisfied by a rank-r approximator D' of D.

Theorem A.5. *In the setting of Theorem A.4, there is an unbiased approximator D' of D satisfying the optimality Conditions 1 and 2 from Theorem A.4.*

To simplify the exposition of the proof of this theorem, we state two lemmas. Their proofs are given in the end of this section.

Lemma A.5. *Let $D = \mathrm{diag}(d_1, \dots, d_n)$ such that $d_1, \dots, d_n \in [0,1]$ with $\sum_{i=1}^n d_i = r$ a positive integer. Moreover, assume there exist orthonormal vectors $z_1, \dots, z_r \in \mathbb{R}^{n\times 1}$ so that the matrix $Z = \sum_{i=1}^r z_i z_i^T$ has diagonal entries $d_1, \dots, d_n$ (in this order). For a vector $s \in \mathbb{R}^{n\times 1}$ of uniformly random signs (i.e. each entry is chosen uniformly and independently from $\{\pm 1\}$), and $z_i' = s \odot z_i$, define $D' = \sum_{i=1}^k z_i' z_i'^T$.*

Then D' is an unbiased rank-r approximator of D satisfying the optimality Conditions 1 and 2 from Theorem A.4.

The pointwise multiplication by random signs s in this lemma can be interpreted as a generalization of the 'sign-trick' from (Tallec & Ollivier, 2017).

To make use of the above lemma, we need to construct $z_1, \dots, z_r$ as described in its statement. This is achieved by the following lemma, whose proof uses ideas from (Israel, 2011)

Lemma A.6. *Let $D = \mathrm{diag}(d_1, \dots, d_n)$ such that $d_1, \dots, d_n \in [0,1]$ with $\sum_{i=1}^n d_i = r$ a positive integer. Then, there exists orthonormal vectors $z_1, \dots, z_r \in \mathbb{R}^{n\times 1}$ so that the matrix $Z := \sum_{i=1}^r z_i z_i^T$ has the same diagonal entries $d_1, \dots, d_n$ as D (in this order).*

Note that Z as defined in this lemma is a symmetric idempotent matrix with trace r. It is not difficult to show that every symmetric, idempotent matrix Z with trace r can be decomposed as a sum $Z = \sum_{i=1}^r z_i z_i^T$, where the z_i are orthonormal. So the Lemma can also be interpreted as the following statement about symmetric, idempotent matrices: Symmetric idempotent matrices can have any diagonal up to the constraint that diagonal entries are between 0 and 1 and sum up to an integer. It is easy to check that any symmetric, idempotent matrix also satisfies these two conditions, so that the lemma fully classifies the diagonals of symmetric, idempotent matrices.

We are now ready to give the proof of Theorem A.5.

Proof of Theorem A.5. Note that in order to construct an optimal rank-r approximator D' of D it suffices to find a rank-k approximator D_2' of $D_2 = \mathrm{diag}(d_m, \dots, d_n)$ satisfying condition 2. from Theorem A.4.

Note that $\frac{k}{s_1} D_2$ satisfies the conditions of Lemma A.6, since its diagonal entries sum to k and are in $[0,1]$ by the definition of m. Therefore, there exist orthonormal vectors $z_1, \dots, z_k \in \mathbb{R}^{(n-m+1)\times 1}$ so that $Z = \sum_{i=1}^k z_i z_i^T$ has the same diagonal as $\frac{k}{s_1} D_2$. By Lemma A.5, choosing a vector $s \in \mathbb{R}^{(n-m+1)\times 1}$ of random signs (i.e. each entry is uniformly and independently drawn from $\{\pm 1\}$) gives an optimal unbiased rank-k approximator $\sum_{i=1}^k (s \odot z_i)(s \odot z_i)^T$ of $\frac{k}{s_1} D_2$ satisfying the (rescaled) Conditions 1 and 2 from Theorem A.4. Multiplying this approximator by $\frac{s_1}{k}$ therefore gives an unbiased rank-k approximator of D_2 satisfying the same conditions. □

PROOF OF LEMMA A.5

Proof. We will first check that D' is actually an unbiased approximator of D and then check the condition from Theorem A.4.

In order to show $\mathrm{E}[D'] = D$, consider the (a,b)-th entry $(z_i' z_i'^T)_{a,b}$ of the matrix $z_i' z_i'^T$. Observe that $(z_i' z_i'^T)_{a,b} = s_a s_b (z_i z_i^T)_{a,b}$, so that for $a \neq b$ we have $\mathrm{E}[(z_i' z_i'^T)_{a,b}] = 0$ and for $a = b$ we have $\mathrm{E}[(z_i' z_i'^T)_{a,a}] = (z_i z_i^T)_{a,a}$. From this, it follows that $\mathrm{E}[D']$ has 0 off-diagonal entries and that its diagonal entries equal the ones of Z. In other words, $\mathrm{E}[D'] = D$ as desired.

We now check the conditions given by Theorem A.4. First of all, note that in the notation of the theorem we have $m = 1$ (since $d_1 \leq 1$ by assumption) and therefore $s_1 = r$, so that we just have to show $\|D' - \mathrm{Id}_r\|^2 = (n-r)$. This is immediate from the fact that the z_i' always inherit orthonormality from the z_i and Observation A.1. □

PROOF OF LEMMA A.6

Let us first give a simplified construction for the special case $n = r+1$ which is used by Algorithm A.2. We simply define the unit-norm vector $z_0 = (\sqrt{1-d_1}, \ldots, \sqrt{1-d_{r+1}})^T$. Now, we find $z_1, \ldots, z_r$ completing an orthonormal bases $z_0, z_1, \ldots, z_r$ of $\mathbb{R}^{r+1}$(for example, one can first complete the basis arbitrarily and then apply the (modified) Gram-Schmidt algorithm). We then have $\sum_{i=0}^{r} z_i z_i^T = \mathrm{Id}_n$ and therefore $Z = \sum_{i=1}^{r} z_i z_i^T = \mathrm{Id}_n - z_0 z_0^T$ has the desired diagonal entries.

We now give the full proof of the lemma, it uses ideas from (Israel, 2011).

Proof. First, we may without loss of generality assume that $d_1 \geq \ldots \geq d_n$, since reordering the diagonal entries of Z can be achieved by reordering the coordinates of the z_i.

We will prove the statement by induction on r and we note that the proof can easily be turned into an algorithm constructing the z_i.

For $r = 1$, the statement is trivial: Simply set $z_1 = (\sqrt{d_1}, \ldots, \sqrt{d_n})$ and note that it has norm 1.

Now assume the statement holds for $r-1$. We want to show that it holds for r. Our plan is as follows: We will change two diagonal entries d_m, d_{m+1}, so that the first m diagonal entries sum up to 1 and the remaining ones sum up to $r-1$. We then apply the induction hypothesis to find vectors x_i such that $\sum_i x_i x_i^T$ has the slightly changed values on the diagonal (with new d_m, d_{m+1}) and then apply a rotation R to restore the original diagonal entries d_m, d_{m+1} and giving the desired $z_i = R x_i$.

We now give the details. Set

$$m = \max\left\{ j \in \{1, \ldots, n\} \colon \sum_{t=1}^{j} d_t \leq 1 \right\}$$

and let $\alpha = 1 - \sum_{t=1}^{j} d_t$. Now, set $d_i' = d_i$ for $i \neq m, m+1$ and set $d_m' = d_m + \alpha$ as well as $d_{m+1}' = d_{m+1} - \alpha$ (note that $1 \leq m < r$ so that $m, m+1$ are valid indices). Then we claim that $d_{m+1}', \ldots, d_n'$ satisfy the conditions of the lemma for $r-1$.

This is not difficult to see: Note that $\sum_{i=1}^{m} d_i' = \sum_{i=1}^{m} d_i + \alpha = 1$ by the definition of α. Moreover, $\sum_{i=1}^{n} d_i' = \sum_{i=1}^{n} d_i = r$. Therefore, we get $\sum_{i=m+1}^{n} d_i' = r - \sum_{i=1}^{m} d_i' = r-1$. Moreover, $d_{m+1}' \leq d_{m+1} \leq 1$ and

$$\begin{aligned} d_{m+1}' &= d_{m+1} - \alpha = d_{m+1} - \left(1 - \sum_{i=1}^{m} d_i\right) \\ &= \sum_{i=1}^{m+1} d_i - 1 \geq 0 \end{aligned}$$

by definition of m. For $i > m+1$, the condition $d_i' \in [0,1]$ is trivial. So we have indeed checked that $d_{m+1}', \ldots, d_n'$ satisfy the conditions of the lemma for $r-1$.

By induction, there exist vectors $y_1, \ldots, y_{r-1} \in \mathbb{R}^{(n-m)\times 1}$, so that $Y = \sum_{i=1}^{r-1} y_i y_i^T$ has diagonal entries $d_{m+1}', \ldots, d_n'$. We write $q = (\sqrt{d_1'}, \ldots, \sqrt{d_m'})^T$ and let

$$x_i = \begin{pmatrix} 0 \\ y_i \end{pmatrix} \in \mathbb{R}^{n\times 1} \quad \text{and} \quad x_r = \begin{pmatrix} q \\ 0 \end{pmatrix} \in \mathbb{R}^{n\times 1}.$$

Then, the x_i are clearly orthonormal and the matrix $X = \sum_{i=1}^{r} x_i x_i^T$ can be written as a block diagonal matrix of the form

$$X = \begin{pmatrix} qq^T & 0 \\ 0 & Y \end{pmatrix}.$$

Especially, X has diagonal entries $d_1', \ldots, d_n'$. On top of that, when we restrict the indices of X to be m or $m+1$, we obtain the submatrix

$$\begin{pmatrix} d_m' & 0 \\ 0 & d_{m+1}' \end{pmatrix} = \mathrm{diag}(d_m + x, d_{m+1} - x) =: D_m.$$

Let

$$R(\phi) = \begin{pmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{pmatrix}$$

be a rotation matrix (with angle ϕ) and choose ϕ so that $R(\phi) D_m R(\phi)^T$ has diagonal entries d_m, d_{m+1}, i.e. $\phi = \arcsin\left(\sqrt{\frac{x}{2x + d_m - d_{m+1}}}\right)$, which is well-defined since $d_m \geq d_{m+1}$.

Now, consider the block-diagonal matrix

$$R = \begin{pmatrix} \mathrm{Id}_{m-1} & 0 & 0 \\ 0 & R(\phi) & 0 \\ 0 & 0 & \mathrm{Id}_{n-m-1} \end{pmatrix}.$$

By the choice of ϕ, we then get that $RXR^T = \sum_{i=1}^{r}(Rx_i)(Rx_i)^T$ has diagonal entries $d_1, \ldots, d_n$. Since R is orthogonal, we get that the $z_i = Rx_i$ are orthonormal and we have therefore constructed the z_i as desired. □

A.2. Additional Experiments

Here we include 5 additional experiments complementing the ones presented in the main paper. The first one illustrates that the batch size chosen does not affect the observation that the performance of OK matches that of TBPTT. The next three analyze the cosine between the true gradient and the approximated one. The first of the three shows the cosine for untrained networks on CHAR-PTB. The second of the three shows the cosine on the Copy task after training until the algorithm learns sequences of length 40. The third one shows that the specific set of trained weights does not affect the cosine significantly, by repeating the previous experiment while retraining the network. The last experiment analyzes the quality of r-OK, the optimal unbiased Kronecker rank r approximation, from a different point of view: by comparing it to the 'best' biased rank r approximation of the gradient, that is, the approximation that stores the closest approximation of the gradient as an r-Kronecker-Sum. Intuitively, the performance of the biased version of the algorithm measures how far away the gradient is from a low rank approximation, which also influences how well one can do unbiased low-rank approximations of the gradient.

A.2.1. CHAR-PTB with larger batch size

For the first experiment, we would like to illustrate that the results obtained in Figure 2, regarding 8-OK matching TBPTT-25 did not depend on the batch size. As noted in the paper, this is in principle clear as the batch size b divides the batch noise and the approximation noise by the same factor b. Figure A.1 shows the validation performance of 8-OK and TBPTT-5 and 25 on the Penn TreeBank dataset in bits per character (BPC). The experimental setup is exactly the same as in Figure 2, except the batch size chosen is 64. Table A.1 summarizes the results.

Table A.1. Results on Penn TreeBank with a batch size 64. Standard deviations are smaller than 0.01.

NAME	VALIDATION	TEST	#PARAMS
8-OK	1.69	1.64	$133K$
TBPTT-5	1.72	1.67	$133K$
TBPTT-25	1.68	1.63	$133K$

A.2.2. Cosine analysis between the true gradient and the approximated one

For the second experiment, we pick an untrained RHN with 256 units in the CHAR-PTB task. This contrasts with Figure 3, where we first trained the network weights to assess the gradient estimate at the end of training (which, as indicated there, is more challenging). We compute the cosine of the angle ϕ between the gradient estimates provided by OK

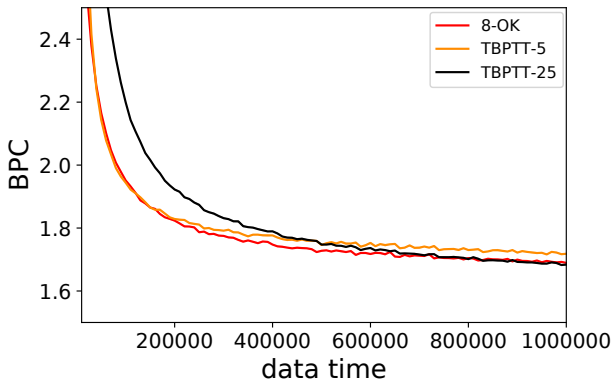

Figure A.1. Validation performance on CHAR-PTB in bits per character (BPC). Even with larger batch sizes, 8-OK matches the performance of TBPTT-25. We trained a RHN with 256 units, with a batch size of 64.

and KF-RTRL and the true RTRL gradient for 10000 steps. We plot the mean and standard deviation for 20 different untrained RHNs with random weights. Figure A.2 shows that the gradient can be almost perfectly approximated by a sum of 2 Kronecker factors, at least at the start of training. This illustrates the advantage of using an optimal approximation, as opposed to the one in KF-RTRL.

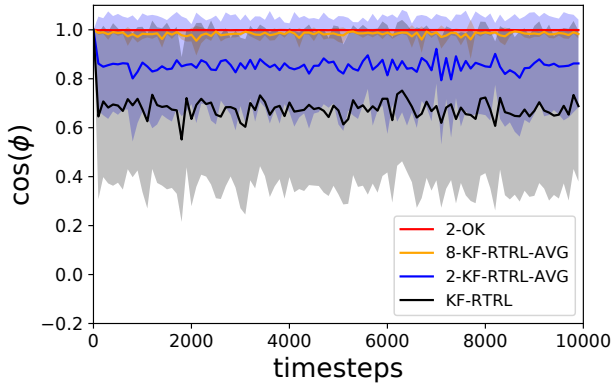

Figure A.2. Variance analysis on an untrained RHN for the CHAR-PTB task. At the start of training, even a sum of 2 Kronecker factors suffices to perfectly capture the information in the gradient.

Naturally, as shown in the paper, the most interesting behavior appears later in training. The third experiment in the appendix is equivalent to the one performed to produce Figure 3, except we use the Copy task and a RHN with 128 units trained until it learns a sequence of length 40. The results are similar in spirit to the Figure shown in the main paper and are shown in Figure A.3. Observe that datapoints where the true gradient is smaller than 0.0001 were removed. This is necessary in the Copy task because there are a lot of steps (say when the network is reading the input), where the task is trivial and the performance has already saturated (leading to very small gradients). Of course, small gradient steps are also irrelevant for learning so removing them is justified.

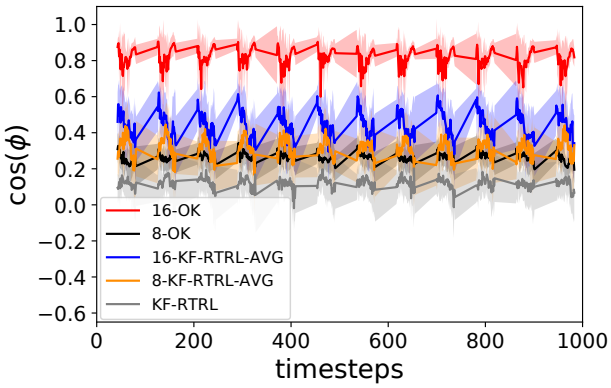

Figure A.3. Variance analysis on the Copy task for a RHN trained until it has learned sequences of length $T = 40$. Even later in training, 16-OK keeps a very good estimate of true gradient. For this plot, we remove datapoints where the true gradient is smaller than 0.0001, as those are irrelevant for learning. In particular, the steps corresponding to the network reading the input are not plotted.

One might wonder whether the behavior observed in Figure A.3 was not specific to the set of trained weights used there. To that end, we retrain the network and repeat the experiment. Figure A.4 shows that the behavior of the cosine does not depend much on the particular set of trained weights used.

Lastly, we analyze the effect of changing the number of units in the RHN. First, we pick untrained RHNs with sizes as powers of 2 from 8 to 512 in the CHAR-PTB task. We compute the cosine of the angle ϕ between the gradient estimates provided by OK and KF-RTRL and the true RTRL gradient after 100 steps. We plot the mean and standard deviation for 10 different untrained RHNs with random weights (in the case of KF-RTRL and 2-KF-RTRL-AVG, we use 100 untrained RHNs). Figure A.5 shows that the number of units does not affect the results seen in Figure A.2, at least for an untrained network.

As mentioned above, the most interesting behavior occurs at the end of training. To this end, we make Figures A.6 and A.7 analogous to Figure A.3 and Figure 3 from the main paper, where we include also an RHN of size 512 for comparison. Observe that there is only a small difference in the performance of both OK and KF-RTRL when the network

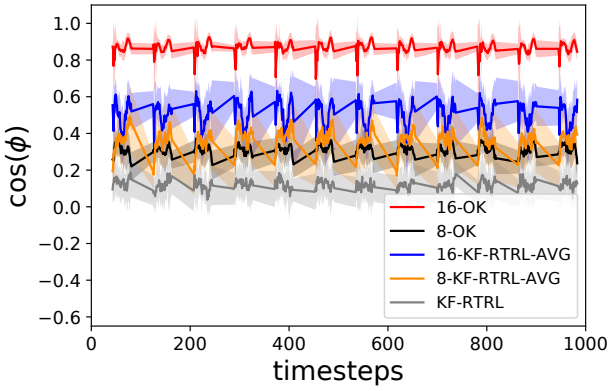

Figure A.4. Variance analysis on the Copy task for a RHN trained until it has learned sequences of length $T = 40$. Repeated experiment with retrained weights.

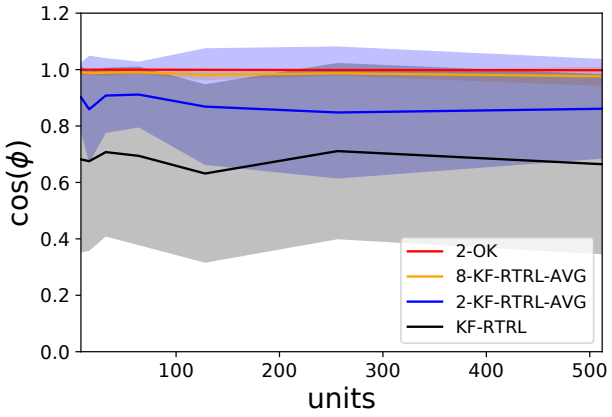

Figure A.5. Variance analysis on an untrained RHN for the CHAR-PTB task, varying the number of units from 8 to 512.

size is increased in Figure A.6. However, in Figure A.7, OK drops more than KF-RTRL, with the advantage of using the optimal approximation almost completely vanishing. We believe that this is due to the gradients in the larger network containing longer term information than the gradients in the smaller network (that is, taking longer to vanish, due to the spectral norm of H_t being closer to 1). In particular, this effect is not present in Figure A.6, as both networks were trained until they learned sequences of length 40. As a result, the gradients probably contain comparable amount of long term information. Naturally, the better test of the quality of the approximations used would be to train a network of larger size in either task. However, due to the computational costs, we have been unable to fully explore the effect of changing the network size experimentally.

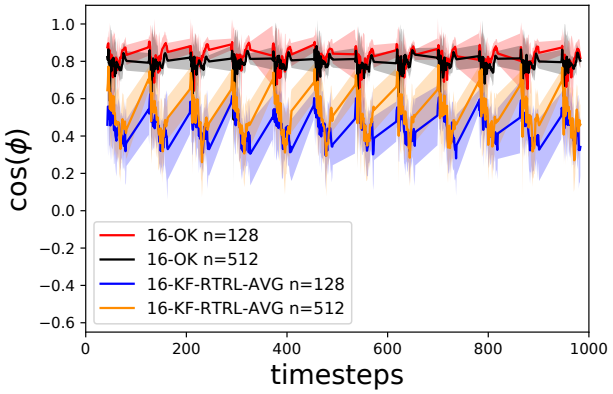

Figure A.6. Variance analysis on the Copy task for a RHN trained until it has learned sequences of length $T = 40$. We vary the size of the RHN used, to show that both the OK and KF-RTRL-AVG approximations do not decay significantly, even later in training, for larger network sizes. As in Figure A.3, we remove datapoints where the true gradient is smaller than 0.0001.

A.2.3. Bias experiments

For the last set of experiments, we perform a Copy task experiment where we compare the optimal unbiased approximations used in OK to the corresponding optimal, biased ones. We first describe the biased approximations and then present the experiments.

A.2.3.1. Description of the Optimal Biased Approximation

In the paper, we were faced with approximating an $(r+1)$-Kronecker-Sum

$$G = u_1 \otimes (H_t A_1) + \ldots + u_r \otimes (H_t A_r) + h \otimes D \quad (8)$$

by an r-Kronecker-Sum G'. We solved the problem of finding an optimal, *unbiased* approximator G' of G. Instead, one can also construct an optimal biased approximator. Concretely, this means approximating G by a (fixed, non-random) r-Kronecker-Sum G', which minimizes $\|G - G'\|$. To clearly distinguish between unbiased and biased approximations, we refer to the corresponding algorithms as Unbiased Optimal Kronecker-Sum, r-U-OK, and Biased Optimal Kronecker-Sum, r-B-OK.

We now give details of how to construct r-B-OK. Similarly to r-U-OK, we first reduce the problem to approximating a matrix $C \in \mathbb{R}^{(r+1)\times(r+1)}$ optimally by a rank-r matrix C' (which is now deterministic). The steps are exactly the ones given in Section A.1.2 and the matrix C is also the same as the one presented there. Now, we need to minimize $\|C - C'\|$ subject to C' having rank at most r. This is a well known problem and solved by the Eckart-Young Theo-

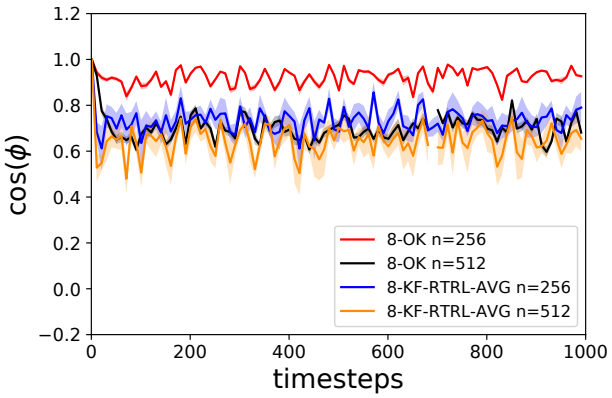

Figure A.7. Variance analysis on the Copy task for a RHN trained for 1 million steps, with sizes either 256 or 512. Observe that 8-OK decays more than 8-KF-RTRL-AVG with the increase in network size. As in Figure A.3, we remove datapoints where the true gradient is smaller than 0.0001.

rem (see Section A.1.1). This finishes the construction of r-B-OK. We also note that almost the same pseudo-code as Algorithm 2 from the paper (Algorithm A.1) can be used. Rather than calling $Opt(C)$, we need to call $OptBias(C)$ as described in Algorithm A.3, which is basically an implementation of the Eckart-Young Theorem.

Algorithm A.3 $OptBias(C)$

Input: Matrix $C \in \mathbb{R}^{(r+1)\times(r+1)}$
Output: Matrices $L', R' \in \mathbb{R}^{(r+1)\times r}$, so that $C' = L'R'^T$ minimizes $\|C - C'\|$.

/* Reduce to diagonal matrix D***/**
$(D, U, V) \leftarrow \text{SVD}(C)$
$(d_1, \dots, d_{r+1}) \leftarrow$ diagonal entries of D
/* Initialise L', R' **to approximate** D***/**
$L', R' \leftarrow 0$
for $1 \le i \le r$ **do**
  $L'_{i,i}, R'_{i,i} \leftarrow \sqrt{d_i}$
end for
/*Approximate $C = UDV^T$***/**
$L' \leftarrow UL', \quad R' \leftarrow VR'$

A.2.3.2. Experiments

The last experiment has essentially two goals. The first is to illustrate that biased approximations are not really desirable when doing gradient descent. This becomes clear in the difference in performance between the biased version of OK and the unbiased ones. The second goal is to show that, throughout training, and not just for specific points as shown in the cosine plots, the gradient can be well approximated by an r-Kronecker-Sum, for small values of r. In particular, this indicates that the noise in r-U-OK is small. Figure A.8 shows that 16-B-OK performs almost as well as 16-U-OK. The performance of 1-B-OK is far worse than the corresponding unbiased OK. For the experiment, we use the same setup as described in Section 4.1.1 of the main paper. Apart from that, the rank 1 algorithms shown in the plot have been run with a batch size of 256. We repeat each experiment 5 times and plot the mean and standard deviation for each.

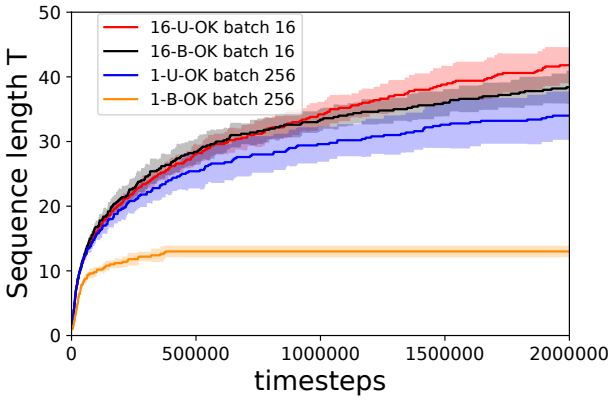

Figure A.8. Analysis of the Kronecker rank of the gradient. The biased rank 16 approximation of the gradient performs almost as well as the 16-OK. This implies the rank of the gradient throughout training can be well approximated by a sum of 16 Kronecker factors.

References

Cline, A. K. and Dhillon, I. S. *Computation of the Singular Value Decomposition*. CRC Press, jan 2006.

Eckart, C. and Young, G. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3): 211–218, Sep 1936. ISSN 1860-0980. doi: 10.1007/BF02288367. URL https://doi.org/10.1007/BF02288367.

Golub, G. H. and Van Loan, C. F. *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996. ISBN 0-8018-5414-8.

Hillar, C. J. and Lim, L.-H. Most tensor problems are np-hard. *J. ACM*, 60(6):45:1–45:39, November 2013. ISSN 0004-5411. doi: 10.1145/2512329. URL http://doi.acm.org/10.1145/2512329.

Israel, R. Constructing idempotent matrices, 2011. URL https://math.stackexchange.com/questions/42283/constructing-idempotent-matrices.

Knuth, D. E. *The Art of Computer Programming, Volume 2 (3rd Ed.): Seminumerical Algorithms*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1997. ISBN 0-201-89684-2.

Merity, S., Keskar, N. S., and Socher, R. An analysis of neural language modeling at multiple scales. *arXiv preprint arXiv:1803.08240*, 2018.

Mujika, A., Meier, F., and Steger, A. Approximating real-time recurrent learning with random kronecker factors. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31*, pp. 6594–6603. Curran Associates, Inc., 2018.

Tallec, C. and Ollivier, Y. Unbiased online recurrent optimization. *arXiv preprint arXiv:1702.05043*, 2017.