We thank the reviewers for their positive and constructive feedback on the paper. There are two main comments which we will address below. We also addressed all the other comments, and they will appear in the revised version.

Power law expansion We thank the reviewers for highlighting this point, as it led to a better formulation of the expansion. We now expand in $\tilde{\tau} = \beta^2\tau$ instead of in τ. Figure R1 shows that for low target values $\hat{z}$ the curves for different g-s collapse, indicating that the relation between g and learning time holds. Note that full convergence occurs for $\tilde{\tau} > 1$, which is beyond the scope of the expansion, but the majority of learning takes place before that time.

For large $\hat{z}$, $\tilde{\tau}$ remains small throughout training, indicating that the expansion is valid. In this case, however, the curves only collapse for the initial training phase. This deviation is due to higher-order terms of the expansion, and is in the direction of decreasing training time for increasing g values – consistent with our main finding. Specifically, expressing the third-order prediction for $z(\tau)$, Eq. (15), in terms of $\tilde{\tau}$ shows this trend:

$$z(\tilde{\tau}) = \hat{z}\left[\tilde{\tau} - \frac{\tilde{\tau}^2}{2} + (1 + 8\hat{z}^2\beta)\frac{\tilde{\tau}^3}{6}\right] . \qquad \text{(R1)}$$

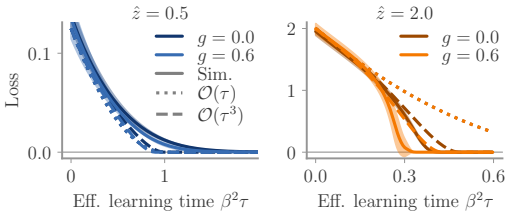

Fig. R1: Loss over effective learning time for the linear problem and different values of initial connectivity strength g and target value $\hat{z}$ (cf. Fig. 3 of the main text). The curves collapse for $\hat{z} = 0.5$. For $\hat{z} = 2$, both the numerical results (full) and our third-order prediction (dashed)) separate at the end of training. The first-order prediction (dotted), $L(\tilde{\tau}) = L_0(1-\tilde{\tau})^2$, has no dependence on g other than through $\tilde{\tau}$.

More complex tasks and network compression We thank the reviewers for suggesting to study more complex tasks and the topic of network compression. We trained an LSTM network on the NLP task of sentiment analysis (Fig. R2). As in our paper, we found that the resulting changes to the network weights are of low rank. Furthermore, when we truncate the *changes* in connectivity, a rank 10 matrix is sufficient to achieve full performance. This is compared to a rank 200 matrix when trying to compress the full connectivity. We are not experts in NLP, and realize that this preliminary result we obtained in a few days is not the end of the story. For the revised version, we will study the effect of higher performing networks, non-binary NLP tasks and different word embeddings among others.

Note that one may not observe this behavior for any task and network off the shelf. In particular, the learning rate is often chosen so high that learning dynamics become highly rugged. In such cases, we repeatedly observed weight changes to be of much higher rank and effectively replacing any initial connectivity. Here we choose a sufficiently small learning rate so that learning dynamics were smooth. Other hyperparameters, such as L2 regularization on the weights, may also change the picture.

For more complex tasks, the emergence of low-rank structure cannot be explained by the small number of in- and output vectors. Instead, we expect the (statistical) task structure to determine the rank, like recently established for feed-forward networks (e.g., Advani & Saxe, 2017, Lampinen & Ganguli, 2019, and the works cited by Reviewer #3).

Other points Very briefly, our results do not depend on the Adam optimizer or model size. Our results rely on BPTT (in particular for the Romo task), and hold for non fixed-point tasks (e.g., producing a periodic output).

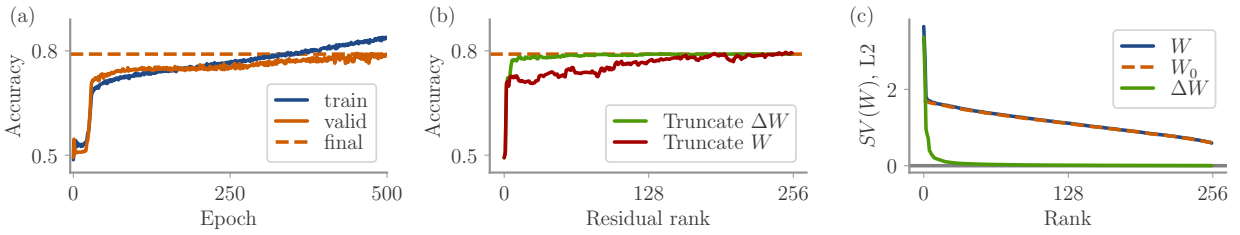

Fig. R2: Low-rank changes for a 2-layer LSTM model trained on a sentiment analysis task, the Standford Sentiment Treebank with binary labels (SST-2). The pretrained word embedding (GloVe) with dimension N_{in} was kept fixed during training, all other parameters were updated with Adam on a binary cross-entropy loss. (a) Training and validation accuracy over epochs. Final validation accuracy marked as a baseline for panel b. (b) Validation accuracy after truncating the lower singular values of connectivity. We either truncate W directly, or apply truncation only to ΔW while keeping W_0. (c) Singular values (SVs) of the recurrent weights in the second layer. The initial, random W_0 is full rank, and the final W visibly differs from it only for the first SVs. The changes, ΔW, are approximately low-rank. Note that the LSTM weights for the four different gates are concatenated ($4N \times N$ matrices). SVs for the other (layer, input) weight matrices look similar. Parameters: Embedding and hidden dimension: $N_{\text{in}} = 100$, $N = 256$, learning rate $0.01/N$, dropout probability 0.5. Weight initialization: $\mathcal{U}(-a, a)$, where $a = \sqrt{1/N}$, except for input weights of layer 1, where $a = \sqrt{1/N_{\text{in}}}$.