We wish to thank all of the reviewers for their time and for their thorough reading of our paper! All minor corrections and typos have been addressed, major concerns/comments are addressed below:

Reviewer #2 We have added text in the discussion about limitations of the fixed point analysis. Other comments addressed in order: **(1)** The reviewer comments that "it is not surprising that the trained RNNs find low-dimensional solutions." We agree! We verified that we never described this fact as 'surprising' in the text. **(2)** All RNNs do indeed achieve close to zero error. Figure 1 (below) has been added to the supplement, it shows the histogram of mean squared error across all networks for each task.

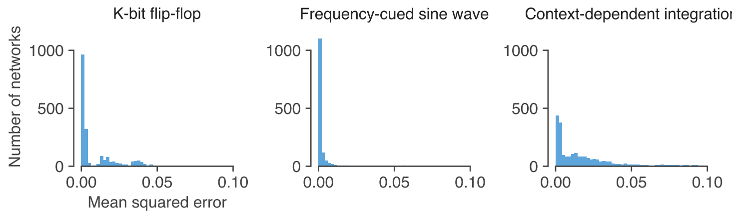

Figure 1: Final losses across all studied networks.

(3) We compare SVCCA and fixed point analyses visually using MDS as it is an intuitive visualization of the corresponding distance matrices (e.g. in Fig. 1c). For any scientific conclusions that necessitate quantitative comparisons, the distance matrix can be used (we have elaborated this point in the text). **(4)** The vanilla RNN is a kind of special case of the gated networks, in that the weights of the gated architectures may be initialized such that the resulting network is effectively a vanilla RNN. After initializing the weights a gated architecture at the solution of a vanilla RNN, we suspect that the weights will not change as the vanilla RNN solves these tasks perfectly well. Indeed, gated architectures were introduced to help with trainability in recurrent networks; rather than increasing the expressivitiy or capacity of these models. **(5)** For the sine wave task, there was always a unique fixed point per fixed input. In general, a limitiation of the topology analysis is that it the same number of fixed points between models (we now elaborate on this limitation in the discussion). Regarding the "key feature being the limit cycle", that is the point of Fig. 2f (see our next response). **(6)** Fig. 2f is an attempt to quantify the limit cycle. It demonstrates that the linearized dynamical system is a good approximation of the full nonlinear dynamics, in that the predicted linear frequency (estimated using the top eigenvalues of the Jacobian of the system) matches the task frequency. This is an important validation of why we should trust the linearization analysis, as such, we think its inclusion is warranted. **(7)** We have added a sentence with our interpretation of Fig 2e after Line 212: the implication is that every network has a 1D manifold of input-dependent fixed points. **(8)** For CDI analysis, we do not presume that there is a line attractor, rather, the topological analysis reveals the presence of the line attractor across networks.

Reviewer #3 **(Major point 1.)** Our analysis has revealed that nearly all of the networks use similar strategies to solve the task, but this was not a foregone conclusion when we began our study. Moreover, we occasionally find networks (typically with extreme values of l2 regularization) with large amplitude, fast oscillations on top of the existing solution (e.g. they still have a line attractor in CDI). These have qualitatively different trajectories, but seem pathological as the these oscillations have no projection on the readout. Note that Sussillo et. al. 2015 similarly reported chaotic solutions under large weight initialization. We are including a depiction of these fast oscillatory networks in the supplement for completeness, along with corresponding discussion in the main text. **(Major point 2.)** Although CCA compares linear projections of two representations, we believe this is a necessary restriction. Under arbitrary nonlinear transformations, any two representations can be made to look identical. Moreover, our motivation in this work is to better understand the surprising similarities found between artificial and biological networks, which use measures such as CCA. Under this motivation, we used CCA in order to be able to compare to the existing literature. In addition, fixed point topology speaks to representation as well and yields identical results for nonlinearly warped, but otherwise similar topologies. **(Minor comment 3.)** We appreciate the pointers to relevant work! We have added these citations to the text. **(Minor comment 4.)** For the MDS graphs, for visualization purposes, we filtered the distance matrix to highlight differences due to architecture (left; formed by taking only tanh networks) or nonlinearity (right; formed by taking only vanilla RNNs). We have generated the requested MDS visualization (comparing all networks) for each task; these will be added in the supplement—the central results do not change. **(Minor comment 5.)** Yes, there are error bars (shaded patches in Fig. 2F); thus it is reproducible. We have not identified the reason for these differences, presumably they arise due to differences in higher order terms of the RNNs. **(Minor comment 11.)** We only ran the analysis for the relu/tanh CDI networks, as the purpose of the figure is just to point out how SVCCA may be misleading.

Reviewer #4 **(1)** Although we would love to develop theory for understanding universality in recurrent networks, we believe that empirical evidence for universality in RNNs is a first step to a broader theoretical understanding. Indeed, a number of studies are beginning to make progress on this front (which are now cited in the main text), but with some simplifying assumptions on the tasks or networks being studied. Given the large diversity of tasks and networks that are actively used in the literature, we feel that the empirical approach taken in this paper (which allows us to explore a large set of tasks/architectures) is a significant contribution on its own. **(2)** We have added a deeper discussion regarding the chosen tasks in both the introduction (motivating why we chose them) and discussion (where we discuss limitations of these particular tasks). We think this particular set of tasks is interesting because they cover fundamental computational primitives: discrete and analog memory, pattern generation, and contextual computation.