General response: We would like to thank the reviewers for their comments. We will incorporate all of the suggestions in the final revision.

Responses to comments of reviewer 1:

Comment 1: Are there nice examples where models fail the more restrictive stability condition, but satisfy incremental stability and show approximately finite memory?

Response 1: Additional specific examples were not included in the paper due to lack of space, but a large class of nonlinear systems which, in general, fail the contractive stability condition of Miller and Hardt but satisfy incremental stability and approximately finite memory is linear time-invariant systems connected with a nonlinear feedback element satisfying the so-called circle criterion (see, e.g., M. Vidyasagar, *Nonlinear Systems Analysis*). This includes systems of the form $x_{t+1} = Dx_t + C\sigma(Ax_t + Bu_t)$, where σ is a componentwise application of a Lipschitz-continuous nonlinearity. Here, $f(x, u) = Dx + C\sigma(Ax + Bu)$ is Lipschitz, but it need not be contractive. On the other hand, it can be shown that the circle criterion is sufficient for approximately finite memory and for global exponential stability, without enforcing contractivity of the state transition map.

Comment 2: Is the exponential dependence on depth in Theorem 3.1 inevitable or an artifact of the construction via Hanin and Selke 2018?

Response 2: The exponential width dependence of the depth of a minimal-width ReLU network is a consequence of Hanin and Sellke's construction. This can be seen in the last paragraph of the proof of Proposition 3 in their paper. When the output of the ReLU net is scalar, the minimal width is equal to $d + 1$, where d is the input dimension, so the number of neurons is exponential in d. On the other hand, exponential dependence on d is generally inevitable when approximating continuous functions by deep ReLU nets, as shown by D. Yarotsky (COLT 2018).

Responses to comments of reviewer 2:

Comment 1: On Theorem 3.1 - I think this is a relatively trivial application of [Hanin and Sellke, 2018] ... Have I missed something in why the result does not follow in a relatively straightforward manner from Hanin and Sellke?

Response 1: We agree that it is at least mildly surprising how easily this result follows from an existing result on function approximation by neural nets (modulo a careful application of causality and time-invariance to relate everything to the output of F at time t). However, to the best of our knowledge, all existing results on universal approximation of i/o maps (e.g., by Boyd–Chua or by Sandberg) reduce the problem to universal approximation of continuous functions on an appropriate compact set and then apply a suitable version of the Stone–Weierstrass theorem. A common drawback is that these proofs are nonconstructive. What we were after was a *quantitative* version of Stone–Weierstrass that would allow us to isolate explicitly the dependence of the depth and width of the approximating ReLU TCN on the approxiate memory length and on the modulus of continuity associated to the original i/o map F. Although Hanin and Sellke do not mention this, their result is, essentially, a quantitative formulation of the Kakutani–Krein theorem, which guarantees that any continuous real-valued function on a compact set can be approximated by a finite composition of affine maps and lattice operations. We will emphasize these points in the final version.

Comment 2: On definition of time-invariance.

Response 2: Thank you for pointing out this oversight. The correct definition of time invariance should be as follows (from Sandberg, 1991): $(\mathsf{FR}^k\mathbf{u})_t = 0$ for $t < k$ and $(\mathsf{FR}^k\mathbf{u})_t = (F\mathbf{u})_{t-k}$ for $t \geq k$. The recurrent model of Section 4 will be time-invariant if the initial state ξ satisfies the conditions $f(\xi, 0) = \xi$ and $g(\xi) = 0$.

Response to comments of reviewer 3:

Comment 1: Since this study is heavily motivated by [Miller and Hardt, 2019], I would raise my score if the authors could answer the learnability question: What class of i/o maps a TCN can learn during gradient descent training?

Response 1: The work of Miller and Hardt was concerned with both approximation and learning. For the latter, they showed that any strictly contracting recurrent model can be approximately learned using gradient descent with truncated backpropagation through time. Since the original model can be learned using backpropagation through time, it is meaningful to compare the gradient descent trajectories with and without truncation. Note that one needs an explicit state-space realization in order to write down the gradient update equations. By contrast, our goal was to show that TCNs can approximate a much wider class of i/o maps with approximately finite memory. Since a TCN model applies a fixed feedforward deep ReLU net to shifted copies of the input training sequence, one can use standard gradient descent with backprop for training.