Reviewer #1 In response to the suggestion to control for initialization of $\mathbf{A}$ and $\mathbf{B}$, we explored several alternatives to $\bar{\mathbf{A}}$, including: (a) the identity matrix, (b) the all-zero matrix, (c) the identity shifted down one row (i.e., a queue of length d), (d) only the diagonals of $\bar{\mathbf{A}}$, (e) only the *off*-diagonals of $\bar{\mathbf{A}}$, and (f) the original $\bar{\mathbf{A}}$ with its diagonals perturbed by some small ϵ. We kept all else consistent with the original setup (i.e., no training on $\bar{\mathbf{A}}$ and $\bar{\mathbf{B}}$) and retrained the model for each alternative. We found that (c) achieved 87.58% test accuracy, consistent with RNN results from Chandar et al. Performance for (f) was a cusp centered about $\epsilon = 0$. Although we only had time to collect a few datapoints, we obtained 94.73% for $\epsilon = -0.01$, 96.97% for $\epsilon = -0.001$, and 96.59% for $\epsilon = +0.001$ (compared to 97.15% for $\epsilon = 0$). All other choices (a, b, d, e), as well as larger values of $|\epsilon|$, resulted in chance-level performance (~10%) on every epoch – indicating poorly initialized weights. Setting $\bar{\mathbf{A}}$, $\bar{\mathbf{B}}$ to be trainable did not help. These results empirically support the theory that the LMU's linear dynamical memory is critical for its success.

We have observed that further tuning of $\bar{\mathbf{A}}$, $\bar{\mathbf{B}}$, and θ is unnecessary for these tasks (lines 103–104) and often counter-productive. However, training these parameters improved performance for small θ in the memory capacity task.

The challenge in the neural precision section relates to substituting "rate" neurons that emit 64-bit floating point values every time-step, with spiking neurons that emit temporally sparse 1-bit values (i.e., binary events that are mostly zero). The error mentioned on line 268 is the RMSE between the ideal input to a rate neuron representing a dimension, and the weighted summation of spike events representing that dimension. This scaling is notable because it converges to zero as the number of neurons are increased (this is nontrivially shown by the referenced theorem). This guarantees that, given enough neurons, there is no systematic error that could prevent the system from approaching the ideal. The Poisson neuron was highlighted since it is entirely *memoryless*. For this, or for other neuron models, spikes provide a trade-off between energy (e.g., in neuromorphic hardware) and model accuracy (Blouw et al., 2019). Lastly, we note that Voelker and Eliasmith (2018) show that spiking neural nonlinearities perform a similar role to the hidden nonlinear units in the LMU. We hope to include these clarifications in a rewrite.

Reviewer #2 All experiments were run on GPUs using Keras and TensorFlow. We will clarify this early in the paper and focus on the impact of the LMU earlier in the abstract, in a rewrite. Our GitHub page will mention hardware specs, instructions for running on GPUs or CPUs, and provide examples of how to use the LMU within Keras. We view our work as an important step to realizing the performance benefits of RNNs on low-power spiking neuromorphic chips, which historically have been eclipsed by the accuracy and parallelism of GPUs. To support this, the code for the neuromorphic implementations will also be made available.

Equation 7 was motivated by the need to project all potentially relevant information into a single memory, and is similar to the design of the NRU (line 81). We resume this discussion with reviewer #3. We believe that Figure 3 demonstrates the potential for this theory to scale (note the `1e8` on the t-axis). In this case showing that we can process an input sequence containing a billion elements using limited resources.

Reviewer #3 Phased LSTMs address similar challenges by way of a novel "time gate" designed to filter updates at particular frequencies. By using the `phased-lstm-keras` package, and an otherwise identical experimental setup, we achieved 83.63% test accuracy on psMNIST with phased LSTMs. The model is sized to ~102k parameters, uses the default leak on the time gate ($\alpha = 0.001$), and is consistent with the sequential (non-permuted) MNIST example included in their repository. Reducing α by 10x, which makes it closer to a vanilla LSTM, and using ~165k parameters, improved accuracy to 89.61% (c.f., vanilla LSTM $\implies$ 89.86%). Setting $\alpha = 0$ provided no further improvements. This suggests that although phasing improves LSTMs for the sequential MNIST task, it does not appear to help for the permuted task variant wherein the input sequences no longer contain prominent cyclic features.

Equations 4, 6, and 7 are much closer to the referenced NRU than the LSTM in terms of the design of the module. Only when $d = 1$ does the layer becomes somewhat analogous to a single-unit LSTM, in that its feedback dynamics behave as a controllable leaky integrator for the hidden state. For larger d, a distinguishing feature is the particular choice of $\bar{\mathbf{A}}$ and $\bar{\mathbf{B}}$ which couple linear units to optimally represent a continuously sliding window of u_t. Equation 3 establishes a mathematical link between the Legendre polynomials, the memory vector ($\mathbf{m}$), and the history of u_t.

Multiple dimensions are taken into account by way of the encoding vectors ($\mathbf{e_x}$, $\mathbf{e_h}$, and $\mathbf{e_m}$), which project across all dimensions. But as noted, u_t is one-dimensional, since there is exactly one memory vector in a given layer. Extensions that use multiple memories (each with its own encoders and memory kernel) will be considered in future work (lines 310–311). This would enable the network to maintain multiple windows in parallel within the same layer.

The choice of d, together with the frequency content of the window, determines the adversarial sequences. In the continuous-time domain, polynomial windows of degree $\gg d$ are adversarial (by equation 3 and orthogonality of Legendre polynomials). In the discrete-time domain, the most difficult sequences are those with power at the Nyquist frequency. For example, we find that both Gaussian noise and Brownian motion require approximately $2d$ dimensions in order to accurately maintain a window of d time-steps. Further mathematical analysis is required.