Comments on presentation: Thank you for the helpful suggestions. We will move some of the "drier" portions of our paper to the supplementary materials and spend more space elucidating and motivating our methods. In addition, as R1 has suggested, we plan to include some new graphics (see Figure 1), in hopes of making our method easier to understand. We will refine and improve these diagrams for the final version of the paper.

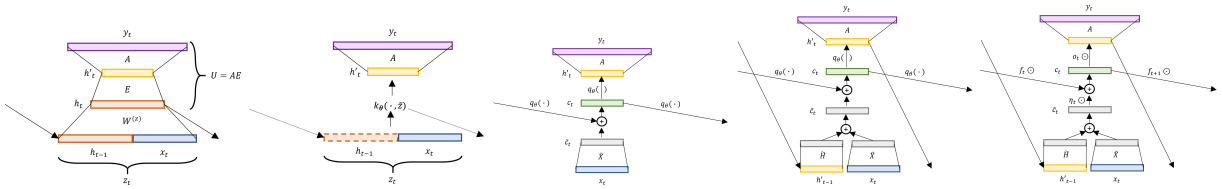

Figure 1: Diagrams for various models. From left to right, the models are: RNN, RKM, RKM with recurrent kernel defined by $q_\theta(\cdot)$, RKM with feedback, RKM-LSTM. We'll make the figures bigger in the final paper (please zoom-in).

Reviewer 1: The factorization $U = AE$ is indeed important for our analysis, but primarily to make the model computationally tractable. As V (which in language models is the vocabulary size) can be quite large, directly modeling y_t can be expensive, as we'd require V anchors $\tilde{x}$. Instead, we use the factorization to get intermediate representation h'_t, which lies in a much smaller dimension j, considerably reducing the number of anchors used. And yes, the memory cell C_t is indeed a vector, not a matrix. We will change this to a lowercase c_t, to reduce confusion.

We focus on Mercer kernel with form $k_\theta(z_t, \tilde{z}) = q_\theta(z_t^\intercal \tilde{z}) = h_t^\intercal \tilde{h}$. As the recurrent hidden variable is of the form $h_t = f(W^{(z)} z_t + b)$ with $z_t = [x_t, h_{t-1}]$, it is natural to choose $e_i = f(W^{(z)} \tilde{z}_i + b)$ with $\tilde{z}_i = [\tilde{x}_i, \tilde{h}_0]$. We do agree that there can be other choices for e_i and $\tilde{z}_i$, which may lead to a RKM model with a formulation different from the standard RNN model. We will add a discussion on this as possible future work in our revision.

Reviewer 2: We'd like to clarify that our claims of a new SOTA were only for the neural LFP task; we did not intend to give the impression that our models for document classification and language modeling were SOTA. We will make this clearer in our revision. Regardless, pushing a new SOTA was not our primary objective. Rather, we seek to connect RNNs with kernel machines, to understand them from a fundamental perspective. Thus, we aimed to compare against strong LSTM-based models, demonstrating that our models derived from kernel methods demonstrate comparative performance. Even so, we obtain SOTA results for recurrent models on all document classification tasks, with the exception of AGNews, for which we're competitive. To the best of our knowledge, the best published transformer-based text classification model Bi-BloSAN [1] performs worse than our model except on AGNews [2].

For language generation, we selected AWD-LSTM as our base because of its popularity, the availability of a reliable implementation, and its relative simplicity. The last factor in particular was important as it allowed us to isolate the impact of different forms of feedback, memory, and gating. We use the official code base of AWD-LSTM, follow their setup exactly, and report the reproducible results in their repository, which are slighty worse than those in the paper.

While LSTM-CNN hybrids have indeed been proposed before, their designs are often somewhat ad-hoc, without much justification. We specifically demonstrate such a construct as a generalization of a recurrent model derived from kernel methods. It also allows us to show that a vanilla 1D CNN (as well as several other proposed models) is in fact a special case (*i.e.*, no feedback or memory) of this generalized RKM-LSTM. We'll add a reference to Quasi-RNNs in our updated version and illustrate the difference with our work. Specifically, Quasi-RNNs can be viewed as a special case of our model by ignoring $\tilde{H} h'_{t-1}$ in eq(17,18) and the $\tilde{W} h'_{t-1}$ terms in all the gates in eq(19), which potentially reduces the capacity to model long-term dependencies.

Reviewer 3: 1. The assumption that e_i lives in the same Hilbert space as the NN output is consistent with prior work on connecting NNs to kernel machines. It is an assumption, but we find it interesting (and elucidating of LSTM mechanisms) that commonly used recurrent models fall out as a result of this assumption, as well as new models. 2. Concerning $q_\theta(C_{t-N})$ being seen as a vector of biases, this is a natural result of the recurrence in the kernel. Such initial biases are often used to initiate a decoder, implemented via a recurrent NN, like an LSTM. Conditions on such biases is worthy of future study, but were deemed beyond the scope of this paper.

References

[1] Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, and Chengqi Zhang. Bi-directional block self-attention for fast and memory-efficient sequence modeling. *International Conference on Learning Representations*, 2018.

[2] Guoyin Wang, Chunyuan Li, Wenlin Wang, Yizhe Zhang, Dinghan Shen, Xinyuan Zhang, Ricardo Henao, and Lawrence Carin. Joint Embedding of Words and Labels for Text Classification. *Association for Computational Linguistics*, 2018.