We wish to thank all of the reviewers for their time and thorough reading of our paper! Specific concerns are addressed:

Reviewer #1 We appreciate the reviewer's suggestions regarding clarity. To improve this, we have: **(1)** Added a preliminaries section which introduces our mathematical notation; **(2)** Highlighted key results and observations that were previously buried stated inside paragraphs in sections 4.2, 4.4, and 4.5; and **(3)** Added more detail to the captions in Figures 1, 3, and 4. We chose not to restructure the mathematical results as theorem/lemma/result (as suggested by the reviewer) as we felt that most of our mathematical statements were largely definitions (such as eq. (3)).

Reviewer #2 Addressing the suggested improvements: **(1)** We have added the suggested summary sentence "the key activity performed by the RNN for sentiment analysis is simply counting the number of positive and negative words used" to the discussion. **(2)** We started with binary sentiment classification, but are actively working on more tasks. For multi-level sentiment classification (e.g. 5-way), our hypothesis is that the networks will still use a 1D line attractor, but that this attractor will be curved such that different readouts will partition different sections of the line attractor (corresponding to different levels of evidence), but still yielding low-dimensional dynamics. We are currently running this experiment and will include its results in the final version of the paper (likely in the supplement, due to space constraints). **(3)** The classification accuracy of the Jacobian linearized model is much worse than the LSTM, due to small errors in the linear approximation that accrue as the network processes a document. Note that if we directly train a linear model, the performance is quite high (only around 3% worse than the LSTM), which suggests that the error of the linearized model has to do with errors in the approximation, not from having less expressive power. We have included a few sentences about this in the discussion. **(4)** We have added a derivation of the expression relating the eigenvalue (λ) to the time constant (τ) in the supplement, along with a corresponding reference in the main text.

Reviewer #3 **(Major point 1.)** We agree with the reviewer that a systematic study of the variability we see in the dynamical structures in our analysis is warranted. Assessing if and how this variability is related to performance differences is something we wish to pursue in future work. We have begun some of these investigations, and have found that the small differences in drift and Q values do not seem to affect the performance (their values are too small to have an effect over typical document lengths in these datasets). **(Major point 2.)** As mentioned in the discussion, we have yet to systematically analyze negation bigrams. We have done some preliminary analysis (see Figure 1, at right) which suggests that RNNs are capable of correctly accounting for 'not' tokens. We have a few ideas for how to uncover these mechanisms (e.g. using switched linear approximations), however, this remains as future work. **(Minor point 4.)** Added more detail to the Figure 1 caption, explaining that it is many neurons for one document. **(Minor points 5. and 3.)** Added a reference to the accuracy table in the appendix in Section 3.1. The bag of words does have performance close to that of RNNs, especially for smaller datasets (providing further support for using linear approximations of the RNNs). **(Minor point 6.)** Regarding the input point around which we linearize: in the paper, we linearized around zero input. We also tried linearizing around the average embedding of all words, this does not change the results (indeed, the average embedding of all words is very close to the zeros vector—the norm of the average embedding is 7.6×10^{-3}). We have added a footnote noting this in the main text. **(Minor point 7.)** Removed the incorrect reference to Fig. 1D. **(Minor point 8.)** Fixed typo. **(Minor point 9.)** We have not correlated the performance with things like the input projections. The projection histograms are over the top positive and negative words, whereas the performance (over test examples) depends on the particular words that show up in those examples; as such, it may not make sense to correlate them.

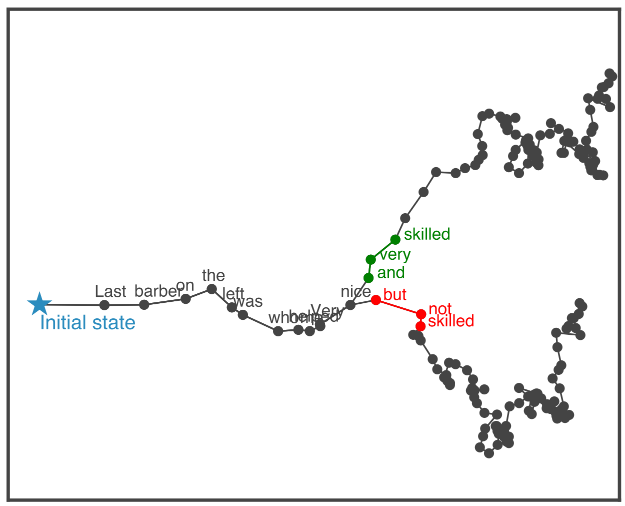

Figure 1: Probing the RNN with negation bigrams. Projection of RNN hidden states onto the top two PCs for two different input sequences that differ only by two tokens (replacing '**and very**' with '**but not**' in the middle of the sequence). The trajectories start out the same as the initial tokens are identical. They then diverge at the critical tokens, moving in opposite directions along the readout (the readout is aligned with the y-axis; not shown). After these two tokens, the rest of the sequence is also identical (tokens not shown to remove clutter). Note how the presence of the negation bigram changes the effect of future tokens on the hidden state.