
RNNs represent belief state geometry in their hidden states¹

Keenan Pepper
Independent

Organized by
Adam Shai, Paul Riechers, PIBBSS

Abstract

Recurrent Neural Networks (RNNs) represent the geometry of the Mixed-State Presentation (MSP) of a Hidden Markov Model (HMM) linearly in their hidden states when trained to predict the HMM's output, similar to how transformers represent this geometry in their residual stream. We train multi-layer RNNs on two simple HMMs, RRXOR and Mess3, and find that a linear projection of the RNN hidden state strongly correlates with the belief state an optimal Bayesian predictor should have about the HMM's underlying state. For RRXOR, the MSP geometry is not faithfully captured by any single RNN layer but is recoverable from the concatenated hidden state across all layers. For Mess3, the geometry is visible even in individual layers. These results demonstrate the generality of Computational Mechanics principles for studying belief state representations across model architectures, with implications for developing flexible interpretability tools as frontier AI systems evolve.

Keywords: Computational Mechanics, Recurrent Neural Networks, AI security, safety infrastructure

1. Introduction

In (Shai, 2024) it was shown that the geometry of the Mixed-State Presentation of a Hidden Markov Model is encoded in the residual stream of a transformer model trained to predict the output tokens, and furthermore it's encoded in a simple, linear way. But the principles of Computational Mechanics are agnostic of model

¹ Research conducted at the Computational Mechanics For AI Safety Hackathon, 2024 (<https://www.apartresearch.com/event/compmech>)

architecture, so there is no reason these findings should be limited to transformers. If a classical Recurrent Neural Network (RNN), for example, learns to predict an HMM’s output near-optimally, it must also have the MSP represented somewhere within it (since that is the set of belief states a Bayesian predictor must occupy when synchronizing to the external process). Is it also encoded in a simple, linear way in this case?

Here we train a multi-layer RNN to predict the output of a simple HMM and do a linear regression on the hidden states to find the subspace of hidden-state-space that is most strongly correlated with the degrees of freedom of the MSP belief states.

At the time of this paper, the current frontier models are all based on the transformer architecture, but that could easily change in the future as algorithmic innovations are made (e.g. Mamba). Mechanistic Interpretability is a popular field of AI interpretability research, but many of the developments are specific to the transformer architecture and may not straightforwardly transfer to other architectures. If Computational Mechanics results are easy to demonstrate for other architectures, that is good news for the hope of general and flexible interpretability tools.

2. Methods

The architecture used here is one that would be familiar from the 1990s (with the exception of ReLU activation). In order to predict the next token generated by the process from the previous tokens, first a unidirectional, multi-layer RNN is run on the known tokens. Then a fully-connected linear layer is run on the hidden state produced by the last layer at the last sequence position, and a final softmax yields the probability distribution over next token predictions.

The RNN was trained via Backpropagation Through Time, by doing a forward pass for a batch of sequences of a given length ($n=6$) and then backpropagating the loss to each layer and each sequence position. Note that Pytorch automatically handles the “unfolding” (multiple contributions to the gradient on each parameter, one for each time it was used in the computation) because of its dynamic computational graph design.

Hyperparameter	Value for RRXOR exp.	Value for Mess3 exp.
Input & output sizes	2 (# of distinct tokens)	3 (# of distinct tokens)
Hidden size	32	128
Layers	3	3
Nonlinearity	ReLU	ReLU
Optimizer	SGD	SGD

Batch size	512	256
Learning rate	0.02	0.02
Sequence length	6	6

The table shows relevant hyperparameters that were used after a brief, informal hyperparameter search.

Next, each trained model was run in a forward pass on a test batch of sequences, and the hidden states were collected. (Technical note: Since the default Pytorch implementation of RNN is heavily optimized such that hidden states at all layers and sequence positions cannot be recovered, the models were run with a custom forward pass implementation that runs it one sequence position at a time in order to preserve all hidden state.)

Finally a linear regression was done where the independent variables are some components of the hidden state (sometimes the whole concatenated hidden state, other times the hidden state at only one layer), and the dependent variables are the predicted probability vectors over the underlying HMM states. In other words, a linear projection of the hidden state is sought which maximizes the correlation with what the model should “believe” about the state of the HMM.

The two HMM examples used are discussed in (Shai, 2024), but briefly, RRXOR (“random-random-XOR”) emits a random bit x , then an independent random bit y , and then $XOR(x,y)$, then repeats. Mess3 is a process with 3 states and 3 tokens where each (token, next state) pair has a nonzero probability, and it has the important property, lacked by RRXOR, that the true probability distribution of next tokens tells you precisely the state (with no “degeneracy”).

See <https://github.com/keenannepper/epsilon-rnns> for Colab notebooks.

3. Results

These results clearly show that the geometry of the MSP is indeed encoded linearly in the RNN hidden state, just as it is in the residual stream of a transformer.

Where in the hidden state is this geometry mostly embedded? Well, for the RRXOR process, none of the reconstructions from one of the hidden state layers alone resembles the ground truth geometry in detail, but the reconstruction using all three layers together does resemble it. Therefore the belief state is not being stored faithfully in any one layer but instead its representation is distributed across layers.

For the Mess3 process, even though it was trained on much fewer tokens than the RRXOR model was, the reconstruction from any single layer does gain a strong resemblance to the fractal pattern of the MSP, especially (to my eye) the last layer. This makes sense because in this particular process, a correct distribution over next

tokens implies a faithful representation of the belief state, and to achieve low loss the model must be able to obtain that from the last layer.

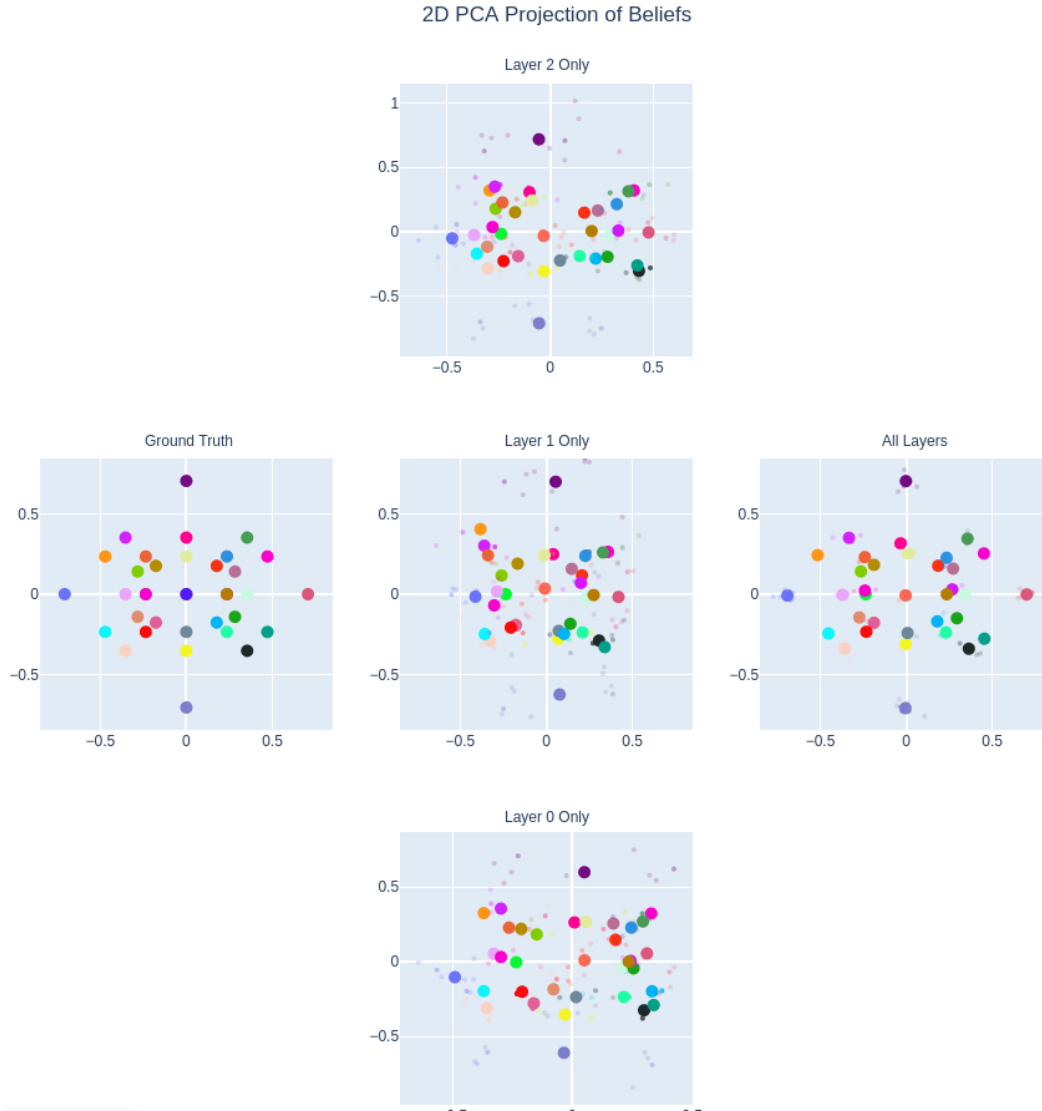
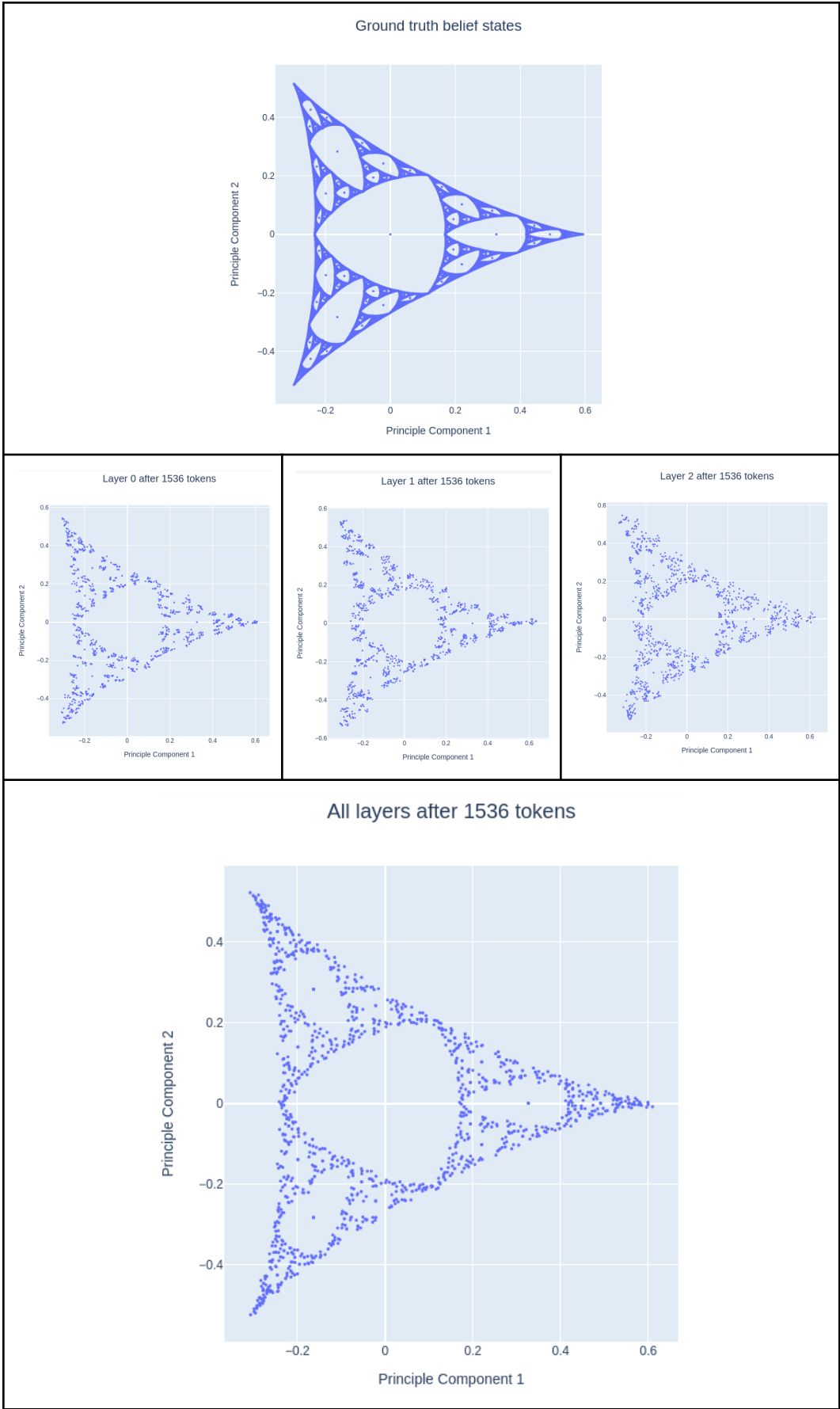


Figure 1 – The geometry of the MSP is a constellation of 36 points (note that some are overlapping in this 2D projection). This geometry is not faithfully recoverable from the hidden state of any layer alone, but it is recoverable from all 3 layers taken together. This model was trained on 986,867,712 tokens.



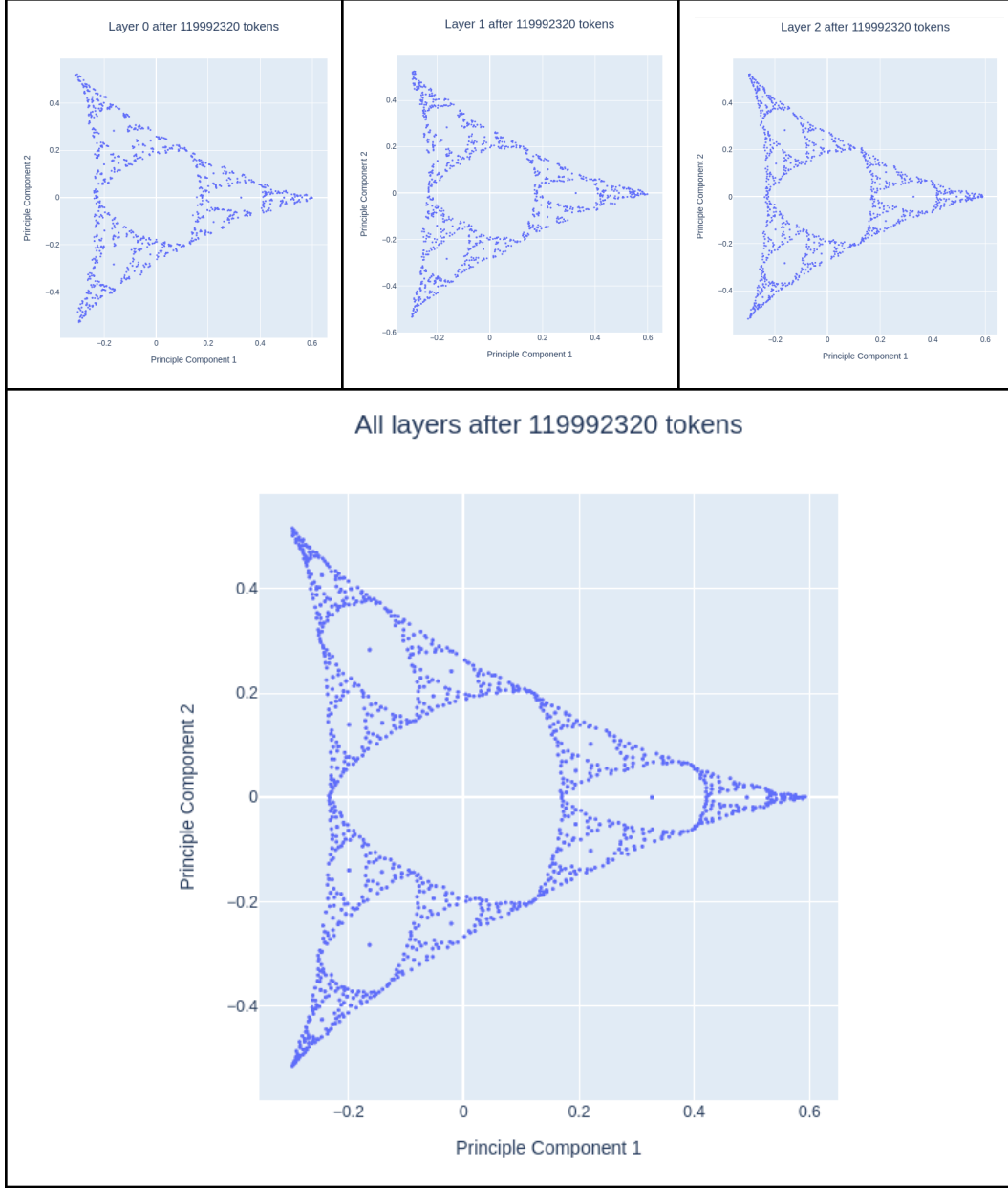


Figure 2 – The belief states for the RNN trained on the Mess3 process, both ground truth and at two different points in training (very early and much later).

4. Conclusion

Belief state geometry is indeed stored linearly in RNN hidden state. In general it does not appear in a single layer but is distributed across layers.

5. References

Shai, A. S., Marzen, S. E., Teixeira, L., Oldenziel, A. G., & Riechers, P. M. (2024). Transformers represent belief state geometry in their residual stream (arXiv:2405.15943). arXiv. <http://arxiv.org/abs/2405.15943>