



Figure 7: The belief state geometry can be represented across multiple layers when the belief state structure has next-token degeneracies. (A) The Random-Random-XOR process has zero pairwise correlation, but has interesting higher-order structure. The MSP of this process has 36 distinct states, with many of them degenerate in terms of their optimal next-token predictions. (B) The belief state geometry lies in a 4-simplex. To visualize we project down to 2-dimensions, with each belief state colored uniquely. (C) The transformer linearly represents this belief state geometry. Small dots correspond to individual activations and are colored according to the ground truth belief state. Large dots correspond to the center of mass of all activations associated with a particular ground-truth belief state. (D) We compare Euclidean distances between pairs of ground truth belief states and see if those distances are preserved in the belief state representation in the transformer (left scatter plots), showing good correspondence for both RRXOR and Mess3 processes. The right scatter plots compare distances in ground truth next-token probabilities to distances in the transformer’s belief state representations, showing low correlation for RRXOR ( $R^2 = 0.31$ ) and high correlation for Mess3, indicating that RRXOR belief state geometry is not captured by next-token predictions alone. (E) Mean squared error of the linear regression procedure, applied to individual layers and, at the far right, applied to the concatenation of activations across layers.

For each pair, we also compute distances between the corresponding belief state representations in the transformer. In (Figure 7D Left) we compare the ground truth pairwise distances to the represented pairwise distances, and find that for both the RRXOR and Mess3 process, there is good correspondence.

Next, for every pairwise distances of belief state representations, we compute the corresponding distance in the ground truth next token probabilities. The scatter plots on the right of Figure 7D compares these distances. For the RRXOR process, the correlation with next-token predictions is relatively low ( $R^2 = 0.31$ ) compared to the correlation with the belief state geometry ( $R^2 = 0.95$ ), indicating that belief state geometry captures structure in the residual stream not captured by next-token predictions. For the Mess3 process, the pairwise structure of next-token predictions is preserved in the belief state representation. This difference between the RRXOR and Mess3 results is expected from the theory. Since the RRXOR process contains distinct belief states with the same optimal next-token distribution, the discovered belief state geometry is not well-explained by the next token predictions.

### 3.3 Belief state geometry can be spread across layers of the residual stream

The framework presented here suggests that belief states should be represented in transformers, but does not say that they must be represented in the final layer. In cases of belief states that are degenerate in their next-token predictions, distinctions can be lost before the unembedding. Thus, in the Mess3 process we expect belief state geometry to persist to the final layer of the residual stream, whereas in the RRXOR trained transformer belief state information can be collapsed before the unembedding.

To quantify this, we report the mean squared error of the belief geometry regression over the different layers of the transformer as well as the concatenation of residual activations from all layers in Figure 7E. We find that the RRXOR belief state geometry is not well represented in the final layer of the transformer. In contrast, the Mess3 belief state geometry is well represented in individual layers of the transformer, and persists through to the final layer before the unembedding. Interestingly, the RRXOR belief state geometry is not represented in *any* of the individual layers of the transformer, but is represented in the concatenation of the layers. This suggests that the belief state geometry is spread across multiple layers of the residual stream.

## 4 Discussion

### 4.1 Belief state geometry: Why?

During training, models do not directly see the hidden structure generating the data—they only see data. So how is it that the pretrained model infers belief states in the simplex of the generator states? In fact, there are infinitely many distinct HMMs that can generate the same stochastic process [5]. Despite this, any stochastic process has a canonical vector-space representation in the space of probability densities over all possible future token sequences [33]. This corresponds to a minimal HMM representation, and we should expect that transformers learn belief state geometry in the simplex over these minimal generator states. This canonical geometry for a process provides an explanation for why we were able to find the belief state geometry represented in our transformers.

In the jargon of computational mechanics, it is notable that the recurrent belief states map onto the causal states of the process’  $\epsilon$ -machine [4]. The recurrent states must be distinguished for prediction, even if there exists a generative HMM that is smaller than the  $\epsilon$ -machine. Typically, data can be generated by a much smaller mechanism than is required to predict it [19, 30]. Accordingly, transformers and other pretrained models learn much more than just an accurate generative model of the world—they also learn how to synchronize to the hidden state of the world through observed context.

We expect our main results do not depend strongly on the particular neural network architecture of transformers, except that they (i) utilize a residual stream and (ii) were pretrained on future-token prediction. Indeed, we expect to find the same belief state geometry in other neural network architectures like Mamba [12], and recent work building on our discovery has shown that recurrent neural networks do represent the belief state geometry for the processes tested in our experiments [26]. The generality of our results is thus a powerful architecture-independent insight for interpretability. Pretraining will generally induce mixed-state geometry. We should then be able to work backwards from this knowledge and the knowledge of the neural network architecture to determine both its world model and how it performs Bayesian updating over its hidden states.

Corollary 2 of [31] implies that, for a recurrent neural network to perform as well as possible on next token prediction, all information about the past necessary to predict the entire future as well as possible must be present in the latent state of the network. The implication for feedforward networks like transformer architectures is much less obvious, but it motivates the hypothesis that to perform as well as possible on next-token prediction requires that all information about the past necessary to predict the entire future as well as possible must be present across the layers of each context window position’s residual stream. Our results here support this interpretation.

Intuitively, why would a model learn about the entire future if it is only trained on next-token prediction? The simple answer is that, even if different belief states imply the same probability distribution over the next token, any distinction in probability distributions over the entire future may *eventually*, after further context, imply distinct distributions over the next token at some point in the future. If the initial nuance were discarded, then the ability to distinguish next-token probabilities in the future would be compromised. To minimize loss, pretrained models need to not only nail the next-token probability distribution, but also retain all information necessary to get the correct next-token distributions in the distant future. It turns out that this requires distinguishing all pasts that induce distinct probability distributions over the entire future [31].

It is tempting to think that all of this information will persist at the final layer of the residual stream, but this is not strictly necessary nor borne out in practice as we showed in Figure 7. In general, the belief state is spread across the layers at each position. This is consistent with [25], which found that

the predictive accuracy of the residual stream with respect to far-future tokens peaks somewhere in intermediate layers. Performing as well as possible on multi-token prediction, as in [10], would imply the same belief-state structure as next-token prediction, but it should change how this information is spread across the layers and, in practice, performance may be different when loss is not minimized.

It is also interesting to consider the linearity of the belief state representations we found, and if the linearity is strictly necessary for a network to optimize next-token cross-entropy loss. In fact, recent work has constructed a transformer by hand (i.e. a human being selected the weights) that perfectly solves the RRXOR task, but where the belief state geometry *is not* linearly represented [11]. It is thus a non-trivial empirical finding that transformer architectures trained via stochastic gradient descent find solutions in which the belief state geometry *is* linearly represented.

## 4.2 Further implications

Our finding of the belief-state simplex in the residual stream strongly suggests that any model capable of minimal loss requires as many residual-stream dimensions as the number of states in a minimal generative model of the stochastic process sampled by the training data. Moreover, we should be able to use this new understanding to determine the minimal loss with fewer residual-stream dimensions. Likewise, using an adaptation of the rate distortion theory applied to the belief-state simplex [20], we anticipate that our framework should be able to make non-trivial predictions about the evolving geometry of activations during training.

To the extent that different users are represented in the training data, they can be thought of as disconnected ergodic components of a non-ergodic stochastic process. Even after the model has learned, it needs to synchronize (in context) to both (i) the ergodic component representing the user and (ii) the current latent state of that component. Each will show up in different ways in the MSP. Using this framework, we should be able to predict rates of adaptations to different users—i.e., rates of convergence in context—as identified by the reduction in next-token entropy as context position increases.

## 4.3 What are features?

A growing literature in LLM research studies what *features* these systems represent [3]. Importantly, there is no currently agreed upon definition of feature. Consequently, it has been difficult to study this issue in a principled manner in which ground truth features are known. While the MSP provides a ground truth understanding of the computation next-token predictors are asked to accomplish, the relationship between belief states and features is not necessarily one-to-one. A single belief state may encompass multiple features, and the same feature may be represented across multiple belief states. The exact mapping between belief states and features is likely to be complex and dependent on the specific architecture and training data of the transformer model, making it an exciting open problem for future research.

## 4.4 Limitations and applicability to more realistic settings

Our experimental validation focused on small-scale systems, using HMMs with only 3-5 states and vocabularies of 2-3 tokens. While these systems exhibited meaningful complexity through infinite Markov order in both the RRXOR and Mess3 processes, they represent a significant simplification compared to the sophisticated architectures and massive vocabularies (>50,000 tokens) of modern language models trained on natural language data. This scale limitation raises important questions about how our framework extends to larger and more complex settings.

For realistic systems like board games or natural language, the dimensionality of the belief state simplex would exceed the dimension of the residual stream. This suggests that if transformers represent belief state geometry in these domains, they must do so in a compressed form. Understanding the nature of this compression—what information is preserved and what is discarded—represents a crucial direction for future work.

As the size of the minimal generating structure and the context window length grows, explicitly computing ground-truth belief states becomes intractable. We will need new methodologies for validating our framework in settings where we cannot directly compute the MSP structure.