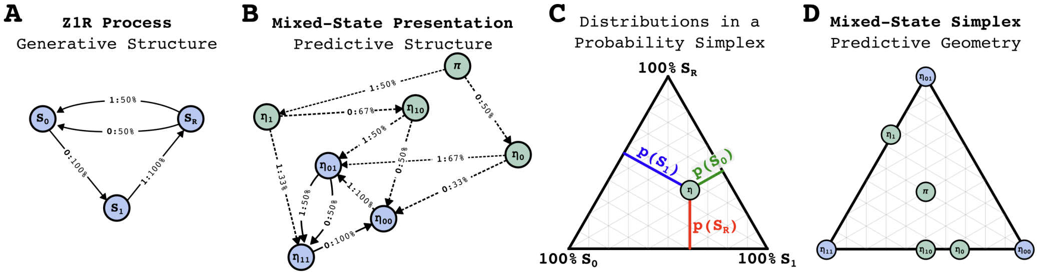

Figure 3: (A) An example generative structure called the zero-one-random process (Z1R), since it generates data of the form `...01R01R01R...` where `R` is a random bit. (B) The generative structure implies a unique metadynamic over belief states as a predictor synchronizes to the hidden state of the world as it observes more context. This predictive structure is called the mixed-state presentation (MSP). We label belief states with $\boldsymbol{\eta}_w$ where w is the shortest string of emissions which leads to that belief state. (C) Belief states are distributions over generator states and can be embedded in a probability simplex. To read off this distribution for a given belief state, $\boldsymbol{\eta}$, one measures the perpendicular distance from the point to each edge of the simplex, shown as blue, green, and red lines in this example. These distances directly give the probabilities for each state. Thus, the vertices represent states of certainty over one generator state, since the perpendicular distance is nonzero to only one of the edges of the simplex. (D) Plotting the belief state distributions in the probability simplex gives the belief state geometry.

2.3 Finding the belief simplex in the residual stream

As described in the previous section, the MSP and belief state geometry are formalizations of the computational structure of the belief updating process an observer must perform in the service of optimal prediction. Consequently, a natural hypothesis would be that transformers, trained on data generated from a given ground-truth HMM, will represent the MSP structure internally. Our procedure for finding the belief states in the residual stream of a pretrained transformer is depicted schematically in Figure 4.

We begin by training a transformer on data generated by a ground-truth HMM. We consider activations of the residual stream (in either a single layer, or a concatenation of layers) induced by all possible input sequences (Figure 4AB), and at all context window positions. Thus our dataset consists of a set of activations, $a \in \mathbb{R}^{d_{\text{resid}}}$, when we consider a single layer at a given position, where d_{resid} is the residual stream dimension. For each input sequence, our framework tells us which belief state the input corresponds to, and the probability distribution over hidden states of the generative process that corresponds to this belief state. These probability distributions can be thought of as points $b \in \mathbb{R}^{|\mathcal{S}|}$, where $|\mathcal{S}|$ denotes the number of hidden states in the generative process.

Since every input has a belief state associated with it[3], we can label (or color, as in Figure 4C) each activation by the associated ground-truth belief state. We then use standard linear regression to find an affine map from a to b, of the form $b \approx Wa + c$, where $W \in \mathbb{R}^{|\mathcal{S}| \times d_{\text{resid}}}$ is a weight matrix and $c \in \mathbb{R}^{|\mathcal{S}|}$ is a bias vector. Parameters W and c are found by minimizing the mean squared error between the predicted belief states and the true belief states. The matrix W projects from the residual stream to (as best as possible) the probability simplex, shown diagrammatically in Figure 4CD.

3 Results

3.1 Belief state geometry is linearly represented in the residual stream

To investigate the computational structure learned by transformers we conducted an experiment using a simple 3-state hidden Markov model (HMM) called the Mess3 Process [20] (Figure 5). We used the data generated by this process (Figure 5) to train a transformer model.

[3] In general, multiple inputs will lead to the same belief state.

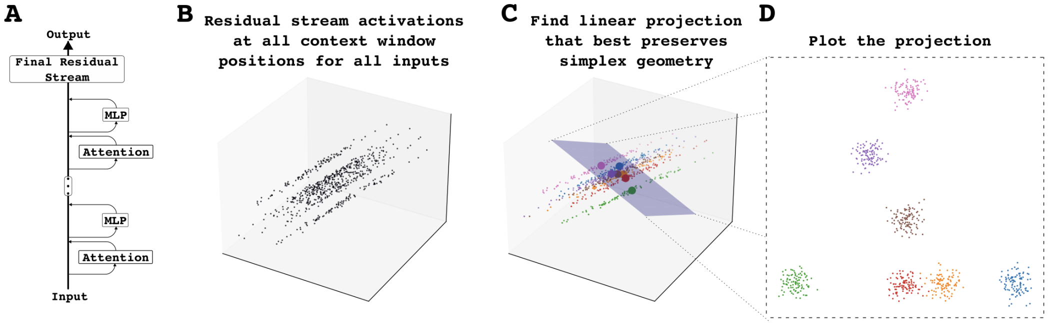
A
Output
Final Residual Stream
MLP
Attention
MLP
Attention
Input
B
Residual stream activations at all context window positions for all inputs
C
Find linear projection that best preserves simplex geometry
D
Plot the projection

Figure 4: To verify if transformers represent belief state geometries in their residual streams, we record (A) residual stream activations at all context window positions over all inputs. (B) These activations live in a high dimensional space. (C) Each input has a ground-truth optimal belief state, which is a probability distribution over states of the data-generating process. In this way we can label, or color, each activation by the ground-truth belief associated with the input. (D) Using linear regression we then find a linear subspace of the activation space that best preserves the belief state geometry of the simplex.

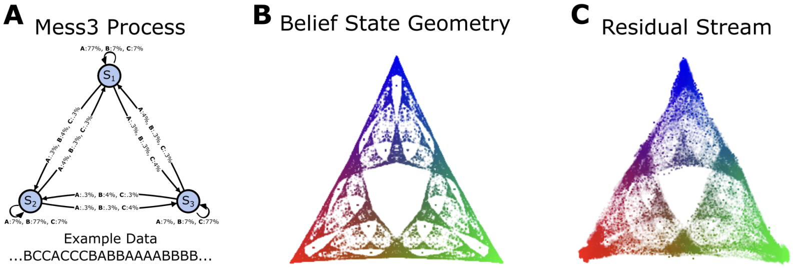

Figure 5: The residual stream of trained transformers linearly represents the belief state geometry of the mixed-state presentation. (A) The Mess3 Process has 3 hidden states and generates sequences in a token vocabulary of $\{\mathtt{A}, \mathtt{B}, \mathtt{C}\}$. (B) The ground truth belief state geometry of the Mess3 Process has intricate fractal structure. Each point in this plot is a belief state—a probability distribution over the hidden states of the Mess3 Process. Points are colored by taking the belief probability distribution and using them as RGB values. (C) We find a linear projection of the final residual stream activations contains a representation of the ground-truth belief geometry. Points are colored according to the ground-truth belief states.

The Mess3 MSP has a fractal structure, shown in Figure 5B, providing a highly non-trivial test of our theory's prediction. Each point in this geometry corresponds to a probability distribution over the hidden states of the data generating process, and thus lies in a 2-simplex.

To test these predictions, we analyzed the final layer of the transformer's residual stream, before the layer norm and unembedding. Using linear regression, we identified a 2D subspace of the 64-dimensional residual activations that best matched the ground-truth belief distributions from the MSP. Remarkably, the geometry of this 2D subspace closely resembled the predicted fractal structure (Figure 5C), providing strong evidence that the transformer had indeed learned to represent the geometry of the belief states in its residual stream[4].

We ran a number of controls, shown in Figure 6, to make sure these results were not artifacts. First, we performed our analysis at multiple points through training and found that the simplex structure emerged gradually (Figure 6A), suggesting that the detailed fractal structure found at the end of training was not a trivial consequence of the model architecture or initialization. Second, quantifying the mean squared error of the regression revealed a decrease in the errors of belief-state geometry representation throughout the course of training (Figure 6D, first 4 bars). In addition, more faithful

[4] These results hold when analyzing the activations after the final layer norm as well, see Figure S1.

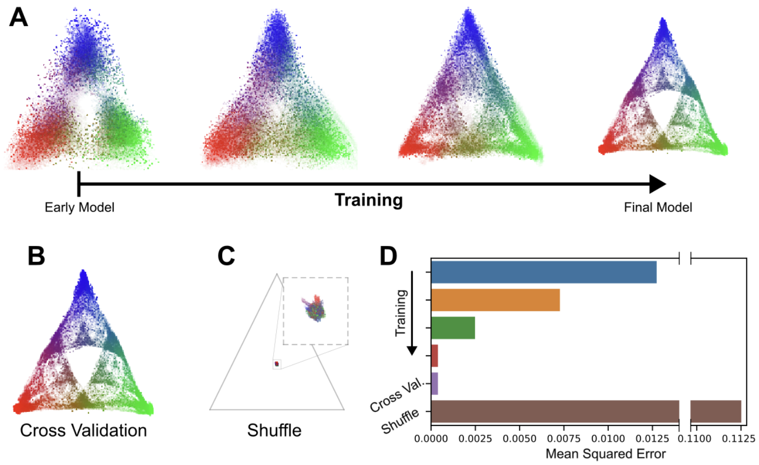

Figure 6: The representation of belief state geometry is nontrivial. (A) Projected activations at different stages of training shows the emergence of belief state geometry. (B) Cross-validation of our main result. (C) We shuffle the belief states in our linear regression procedure, preserving the overall ground-truth fractal shape while getting rid of the associated context. The new projection collapses the data, showing that the fractal's appearance in the residual stream is not an artifact of projecting high-dimensional data to a desired shape. (D) Mean squared error (averaged across input sequences) between (i) the position of projected activations and (ii) the ground-truth position of the corresponding belief state.

representation of the belief state geometry in the residual stream corresponded to lower cross entropy loss (Figure S2). Third, we performed cross-validation, where only 20% of all input-activation pairs were used to train the regression, and then the held out 80% data was used to visualize and analyze the result, and found that the geometry of the belief state was still well represented (Figure 6B) and that the mean squared error was similar to that of the full regression (Figure 6D, red vs. purple bars). Finally, we ran a control that preserved the fractal geometry in the simplex but shuffled the input-point correspondences, resulting in the regression mapping all points to the simplex center due to the lack of discoverable structure (Figure 6C).

These results provide compelling evidence for our central claim: transformers trained on data with hidden generative structure will learn to represent the geometry of belief states in their residual stream. The close match between the predicted fractal structure and the empirical geometry of the residual activations, even for the highly complex Mess3 MSP, suggests that this geometry is a fundamental aspect of how transformers build predictive representations of their input data.

3.2 Belief state geometry represents information beyond next-token prediction

Often, *distinct* belief states will have the *same* next-token prediction associated with them. Our framework suggests that transformers will keep internal distinctions in the representations of these belief states, despite the fact that transformers are trained explicitly on next-token prediction.

The Random–Random–XOR (RRXOR) process, shown in Figure 7A, has these types of degeneracies [28]. The MSP of the RRXOR process has 36 distinct belief states, and can be geometrically represented in a 4-simplex. In Figure 7B we visualize the simplex geometry by projecting down to 2 dimensions. After training, we run our regression analysis to see if the belief state geometry is represented in the residual stream. Unlike our previous results for the Mess3 process, we find that the belief state geometry is not represented in the final residual stream before the unembedding. However, when we take the residual stream activations of all layers and concatenate them, we find a veridical representation (Figure 7C).

The geometry we find is not explainable by next-token predictions. To quantify this, we ask if the pairwise distances between belief state representations in the transformer can be explained by pairwise distances in the next-token predictions, or if they are better explained by ground truth belief state distances. First, we compute the Euclidean distance between pairs of ground truth belief states.