Routledge
Taylor & Francis Group

Cognitive Neuroscience

Current Debates, Research & Reports

ISSN: 1758-8928 (Print) 1758-8936 (Online) Journal homepage: www.tandfonline.com/journals/pcns20

Beyond Markov: Transformers, memory, and attention

Thomas Parr, Giovanni Pezzulo & Karl Friston

To cite this article: Thomas Parr, Giovanni Pezzulo & Karl Friston (2025) Beyond Markov: Transformers, memory, and attention, Cognitive Neuroscience, 16:1-4, 5-23, DOI: 10.1080/17588928.2025.2484485

To link to this article: https://doi.org/10.1080/17588928.2025.2484485

Published online: 15 Apr 2025.

COGNITIVE NEUROSCIENCE
2025, VOL. 16, NOS. 1–4, 5–23
https://doi.org/10.1080/17588928.2025.2484485

Routledge
Taylor & Francis Group

Beyond Markov: Transformers, memory, and attention

Thomas Parr[a], Giovanni Pezzulo[b] and Karl Friston[c]

[a]Nuffield Department of Clinical Neurosciences, University of Oxford, Oxford, UK; [b]Institute of Cognitive Sciences and Technologies, National Research Council, Rome, Italy; [c]Queen Square Institute of Neurology, University College London, London, UK

ABSTRACT

This paper asks what predictive processing models of brain function can learn from the success of transformer architectures. We suggest that the reason transformer architectures have been successful is that they implicitly commit to a non-Markovian generative model – in which we need memory to contextualize our current observations and make predictions about the future. Interestingly, both the notions of working memory in cognitive science and transformer architectures rely heavily upon the concept of attention. We will argue that the move beyond Markov is crucial in the construction of generative models capable of dealing with much of the sequential data – and certainly language – that our brains contend with. We characterize two broad approaches to this problem – deep temporal hierarchies and autoregressive models – with transformers being an example of the latter. Our key conclusions are that transformers benefit heavily from their use of embedding spaces that place strong metric priors on an implicit latent variable and utilize this metric to direct a form of attention that highlights the most relevant, and not only the most recent, previous elements in a sequence to help predict the next.

ARTICLE HISTORY
Received 2 December 2024
Revised 17 February 2025

KEYWORDS
memory; attention; transformers; Markovian; inference; factor graphs; generative

1. Introduction

The success of large language models over the last few years has been attributed to the benefits of the transformer architecture (Bishop & Bishop, 2023) and, specifically, to use of attentional mechanisms (Buckley & Singh, 2024; Niu et al., 2021; Vaswani et al., 2017). However, the use of the term 'attention' is always contentious, and even in psychology has been deployed to mean many different things (Buschman & Miller, 2007; Feldman & Friston, 2010; Gottlieb, 2023; Parr & Friston, 2019). Perhaps the common theme is that attention tells us the degree to which we use (covertly attend, enhance synaptic gain, or foveate) a particular source (location, modality, or feature) of information to update some internal state (implicit belief, representation, or pattern of brain activity). In this paper, we take the success of transformers and their attention mechanisms as our starting point. We will suggest that the primary problem – that attention solves in the context of transformers – is the non-Markovian structure of the sequences they seek to predict (Marković et al., 2019). Markovian processes are often characterized as memoryless sequences (Bakry et al., 2014), where one can forget about everything before the current item in the sequence and lose nothing in predicting the next item. By engaging with non-Markovian processes, we will suggest the problem solved by transformers provides a link between two concepts – often inextricable from one another in cognitive neuroscience – attention and memory (Manohar et al., 2019; Parr & Friston, 2017b). What follows is a somewhat formal treatment in which we deconstruct the computational architectures implied by non-Markovian generative models, asking how these architectures underwrite attention and memory in a computational and cognitive sense; i.e., in the sense of artificial and natural intelligence. These questions speak to some key issues in theoretical neurobiology and cognitive neuroscience.

We take a predictive processing perspective on the issue of when naïve Markovian assumptions break down. To do so, we must try to identify the implicit generative models of non-Markovian processes. Before we do this, we should consider what it means for a process to be non-Markovian (Schütz & Trimper, 2004; van Kampen, 1998). Surely, the past cannot bypass the present to influence the future? Without getting too deep into the philosophy of this question (Lettie, 2024), a simple answer is that a non-Markovian system is one in which we only consider a subset of a more complete Markovian system. More technically, a non-Markovian system can be obtained through a surjective and non-injective (many-to-one) mapping of a Markovian system onto some observable space.

CONTACT Thomas Parr thomas.parr@ndcn.ox.ac.uk Nuffield Department of Clinical Neurosciences, University of Oxford, Level 6, West Wing, John Radcliffe Hospital, Oxford OX3 9DU, UK

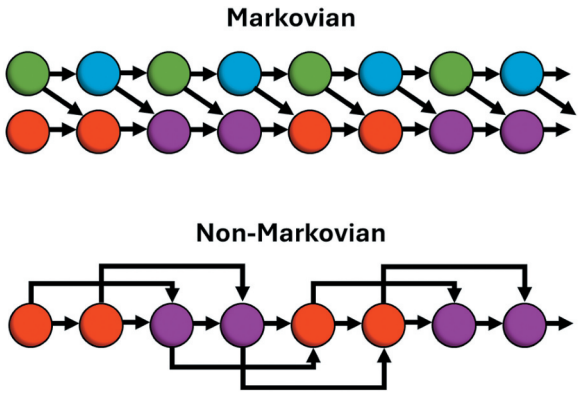

Figure 1. This graphic shows the difference between Markovian and non-Markovian sequences. Specifically, it frames a non-Markovian sequence as being a Markovian sequence with only partial information available. In the upper graphic, pairs of colored balls predict the next pair, and the next pair, and so on. Knowledge of the current pair gives all the information needed to predict the next pair. However, if we only knew about one of the pair, as in the lower graphic, then we can still predict what happens next, but only if we recall previous elements of the sequence. This highlights the importance of memory when the Markovian property is broken.

This idea is illustrated in Figure 1 with a Markovian and a non-Markovian system. Both involve a sequence of colored balls. The Markovian system involves a pair of balls, where the first can be either green or blue, and the second can be either red or purple. There is a predictable sequence here. If the first ball is green, it will be blue next, and vice versa. If the first ball is green, then the second ball will stay the same color (red or purple), and if the first ball is blue, then the second ball will change color (purple or red) at the next time point. Knowing everything about one pair (e.g., green and red) in the sequence tells us everything we need to know about the next pair (e.g., blue and red). However, if we only knew about the second ball, and could not see the first ball, our system would become non-Markovian. Knowing that the ball is currently red is not enough to predict the next color. However, we can make this prediction if we recall the previous color: two reds in sequence predict purple next, and so on.

Even in this simple example, we start to see primitive forms of memory and attention emerge. The move from Markovian to non-Markovian requires us to keep track of events further in the past (memory) by forcing us to update our predictions based upon this selective source of information (attention). While, if we had infinite computational resources, we could maintain a memory of all previous events, we would still need to determine which of these is relevant to predicting the next states. Clearly this could be extended much further, with many additional 'hidden' variables posited to require much longer memory spans or to require specific information about 2 steps ago, 3 steps ago, or some arbitrary point in time. As formulated by transformer systems, the problem is one of determining which previous elements of the sequence to 'attend' to maximize information about the next element (Radford, 2018; Vaswani et al., 2017). This is not unlike the problem of estimating the coefficients of an autoregressive model or temporal autocorrelation function for a continuous time-series (Shumway et al., 2000) or – in dynamical systems theory – applications of Taken's embedding theorem (Deyle & Sugihara, 2011; Freeman, 1987; Takens, 1980). It also resonates with the use of ε-machines (Crutchfield & Young, 1989) to draw inferences about the statistical complexity of a sequence by determining the relevant history of the sequence when attempting to forecast its future.

This example not only highlights the link between attention and memory, it also highlights the dependence on an observer. Implicit in the construction of the non-Markovian sequence is that we have taken the perspective of a (biological or artificial) agent who can only take partial observations from the world. To make this more explicit in what follows, we will assume that there is some process in the world – about which we will be agnostic – generating sequences of data. We will consider the model used by the observer who is attempting to predict the next element in the sequence. These models will include 'hidden states' that are used to try to explain the data available, which may or may not reflect 'real' states of the environment.

A further point that follows from the above is that a model that employs Markovian hidden states may be used to characterize non-Markovian observations. Almost invariably, this is achieved by a (hidden) state-space that includes variables that, if known, are informative about the history of a sequence. For instance, we could model the non-Markovian sequence in Figure 1 using the corresponding Markovian sequence in the same figure as hidden states. Inferences about the blue-green states would then look like memories about the last transition point (red to purple or *vice versa*) in the observable sequence. Finding an appropriate way to structure these state-spaces – i.e., finding an appropriate structure for memory – is particularly important in the context of sequences with long-range dependencies. Maintaining memory of relevant past events – and using them to learn long-term dependencies in sequences – are fundamental problems in many machine learning applications. One of the reasons for the success of transformer architectures is that their attention mechanism resolves these problems better than previous schemes, based upon (for example)

recurrent neural networks and gating mechanisms (Buckley & Singh, 2024; Cho, 2014; Hochreiter & Schmidhuber, 1997; Su et al., 2023).

In what follows, we consider two different approaches to modeling sequences of this sort. Broadly, these include an autoregressive approach, in which we keep track of previous values in the sequence and use these to predict the next step; and a deep temporal approach, in which we explicitly model a separation of timescales (Athanasopoulos et al., 2017; Friston et al., 2024; Friston et al., 2017; George et al., 2009; Pezzulo et al., 2018). The former is the implicit generative model that underwrites the attention mechanisms in a transformer network. The latter is the form of generative model that has been entertained in computational neuroscience to account for our ability to perceive and plan over a range of temporal scales. Crucially, thinking slow involves projecting longer distances into both the past and future, and in doing so replicates some of the features of working memory (Trapp et al., 2021).

In Section 2, we start by using graphical (factor graph) representations to unpack the difference between these two approaches, before focusing in on the autoregressive approach and the role of attention. This will

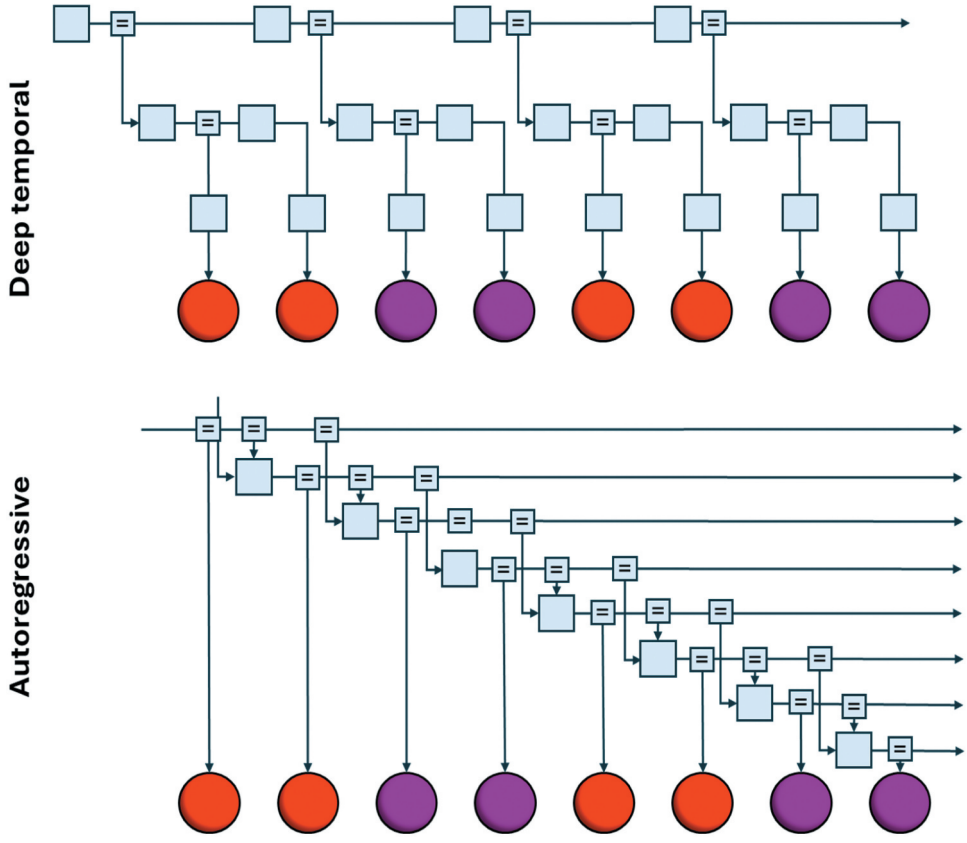

Figure 2. This figure shows two ways in which the sequence from Figure 1 might be modeled. This schematic uses the notation of a Forney factor graph (Dauwels, 2007; Loeliger, 2004; Loeliger etal., 2007) – a graphical depiction of a probabilistic generative model. We will use this notation throughout. Edges (lines) represent variables. Nodes (squares) represent probability distributions. A conditional probability distribution is represented by a square connected to multiple edges, with incoming edges representing the conditioning set, and outgoing edges representing the support of the probability distribution – i.e., a factor representing $p(a|b)$ is a square connected to an edge representing a variable a and an edge representing a variable b. These edges are also connected to the squares representing other probability distributions in which they participate. Inference in a factor graph (Dauwels, 2007; Parr et al., 2019; van de Laar & de Vries, 2019) can be formulated as a process of passing messages between nodes, such that each edge can be associated with bidirectional messages used to update beliefs. Bidirectionality here relates to the messages used to form inferences rather than the way in which data are generated, meaning backwards messages reflect a form of (Bayesian) smoothing; i.e., using future data to resolve uncertainty about past states. These messages refine beliefs about the past, rather than violating temporal causality by influencing the past. Backwards messages are redundant in the transformer architecture as they are formulated such that there is no uncertainty about the past. Squares containing the equals ('=') sign represent Dirac delta distributions that enforce equality of the edges they connect. The upper graphic shows a factor graph detailing a deep temporal solution to the problem of Figure 1. The lower graphic shows a second-order autoregressive approach to the same sequence. See main text for more details.

necessitate a discussion of embedding and of positional encoding. In Section 3, we move to a more mathematical account of transformer attention mechanisms. Section 4 provides an analogous account of deep temporal architectures. In Section 5, we outline some key questions that foreground the difference between artificial and natural intelligence.

2. Autoregression and deep temporal models

Figure 2 illustrates the way in which hierarchical and autoregressive processes can be used to characterize non-Markovian sequences of the sort shown in Figure 1. The format uses that of a Forney factor graph (see caption for details) (Forney, 2001). The autoregressive process effectively caches all previous elements (represented as horizontal lines) and calls upon these previous elements to form empirical priors about future elements in the sequence. The non-Markovian nature of the sequence is evident in the edges (lines) that connect factors (squares) that are separated by more than one horizontal line.

The additional elements we need for a transformer to work are a sensible embedding (Dar et al., 2023; Mikolov et al., 2013; Tennenholtz et al., 2023) that transforms the elements of the sequence into a metric space, and some good prior beliefs about how to choose the relevant previous elements. The last of these will seem familiar to those engaged in active inference (Parr et al., 2022), sensing (Yang et al., 2016), and learning (MacKay, 1992; Settles, 2011) research, where the challenge is one of choosing a means of sampling data that maximize the expected information gain (Lindley, 1956) about some hidden state generating observable outcomes.

So far, we have considered processes that are non-Markovian through a dependence upon a single step back in time. However, the dependence on the past may be arbitrarily complicated. Consider the following rule. If the sequence starts with a red ball, then the 8th ball in the sequence will match the color of the 6th ball. Otherwise, it will match the 7th ball. Here, we see that predicting the 8th ball from the 7th ball depends upon it being contextualized by the 1st and 6th, but that balls 2–5 are irrelevant. This means we can disconnect all the lines connecting balls 2–5 to the 8th ball in the sequence as shown in Figure 3. In other words, we selectively attend to only those sequence elements that are predictive.

The type of contextualization discussed above is fundamental for the practical success of transformers and its applications in neuroscience. Recent studies suggest that the embeddings that best explain linguistic processing are *contextual* embeddings learned by transformer-based language models (e.g., GPT). Embedding models like Word2Vec assign each word a single, fixed embedding that represents its meaning across all contexts. In contrast, transformer-based architectures typically use learned token embeddings (along with positional encodings) as input and refine them through multiple self-attention layers to produce contextual embeddings for each word, adjusting them based on the surrounding words within a given context window. Contextual embeddings enable a more nuanced representation of language and provide a better explanation of brain activity during language processing, plausibly because they capture subtle relations – syntactic, semantic, and pragmatic – among words (Goldstein et al., 2022, 2024; Manning et al., 2020; Pavlick, 2022).

So far, we have assumed that the terms in the sequence are directly observable and that the rules that predict each subsequent term are independently defined for each of the possible sequence elements (i.e., red is treated as distinct from purple, green, or blue). However, what if each element of the observable sequence were in fact generated by some underlying latent variable? If that latent variable had some metric structure, we start to obtain a notion of similarity. Figure 4 illustrates this by associating a 3-dimensional vector of unit length with a color scale. In effect, this means we assume a latent variable that represents a location on the surface of a sphere – where we have wrapped a color-scale around the sphere.

The final ingredient to add is positional encoding (Dufter et al., 2022). This is the incorporation of information not just about what a sequence element is, but where in the sequence it occurs. One way in which this is sometimes achieved is by adding a perturbation to each embedding vector to reflect positions in the sequence (Vaswani et al., 2017). In other words, the likelihoods mapping the latent states to observations include an additive transformation before mapping to the discrete values of sequence elements. This means that, when mapped back into the embedding space through inference, the inferred latent variable represents both identity and sequence position in a single vector value.

3. Transformers and attention

The way in which we have presented the problems above may seem a little unusual to those used to working with transformers, who might view the problem they solve through the lens of function approximation. From this perspective, the challenge is to take some input (e.g., previous text values) and optimize

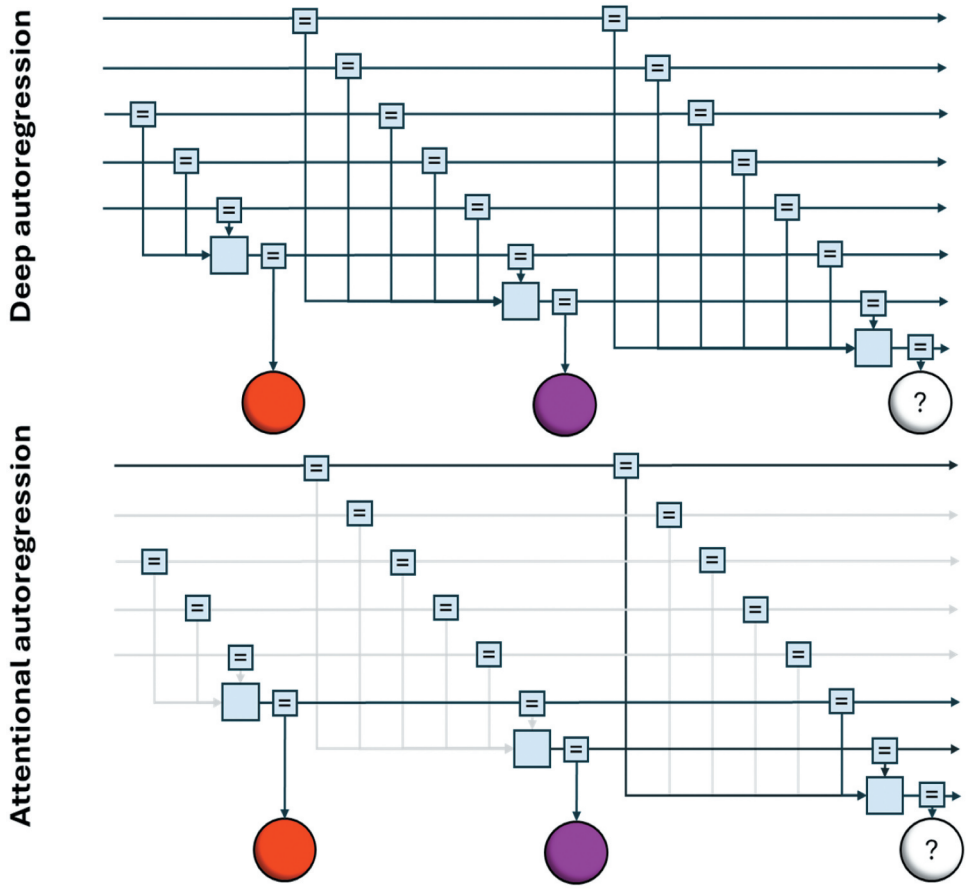

Figure 3. A key problem in designing autoregressive models is determining a 'convolution kernel' that picks out, or 'attends' to, specific relevant elements of the previous sequence. The lower plot shows this sort of attentional processing by highlighting only the connections that are usefully predictive, allowing for an implicit pruning of irrelevant, unattended connections. The sequence in question here determines the next color from knowledge of the first element of the sequence (memory of which is present in the upper horizontal edge) which then contextualizes whether the next element matches the 6th or 7th element. The problem we now face is how to parameterize and optimize the implicit attentional kernel, and which prior beliefs we can bring to this to improve our inferences. Note that the grayed-out edges shown here are not a standard part of the factor graph formalism but are used here – didactically to highlight those conditional dependencies that are imprecise – i.e., for which the conditional probability distribution depends only weakly on the variable in its conditioning set represented by that edge.

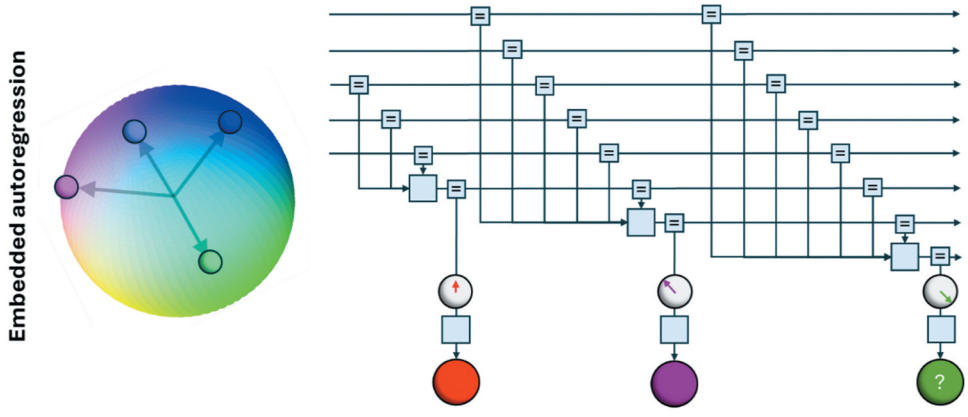

Figure 4. This figure sets out some of the key prior information that is useful in formulating attention. This depends upon equipping the upper graph from Figure 3 with additional factors so that the observable data depend upon some latent variables. Here, the idea is that sequence elements, or tokens, are generated from continuous vectors in some embedding space, and that inference involves projecting back into this embedding space. On the left, an example embedding space is shown in which the discrete colors we were working with to this point become locations on the surface of a sphere. As we will see, the metric structure of this embedding space is an important prior that will be useful in deciding what to attend to.

the weights of a set of linear (interleaved with non-linear) functions, such that the resulting output best matches the next element of a sequence in our data. The reason we have chosen to formulate this explicitly in terms of a probabilistic generative model is that it helps to make the problem more transparent and, hopefully, lets us understand how transformers solve the problem. In this section, our aim is to take the intuitions outlined above and to show how these relate to the mathematics commonly found in the transformer literature.

The idea of learning parameters θ to predict sequences can be summarized as the following maximum likelihood estimation problem (or, equivalently, maximum *a posteriori* estimation with uniform priors):

$$\begin{aligned}\Theta &= \arg\max_{\theta} p(\mathbf{o}_1, \ldots, \mathbf{o}_i; \theta) \\ &= \arg\max_{\theta} p(\mathbf{o}_1, \ldots, \mathbf{o}_i, \theta) \Leftarrow p(\theta) = U(\theta; [-\infty, \infty])\end{aligned} \tag{1}$$

One can view this optimization problem as the optimization of a generative model through interpreting the distribution in Equation (1) as the result of a marginalization operation:

$$p(\mathbf{o}_1, \ldots, \mathbf{o}_i; \theta) = \int \cdots \int p([\mathbf{o}_1, \ldots, \mathbf{o}_i], [\mathbf{s}_1, \ldots, \mathbf{s}_i]; \theta) d\mathbf{s}_1 \ldots d\mathbf{s}_i \tag{2}$$

Here, we call upon a set of implicit latent variables ($\mathbf{s}$) which we will assume are continuous vectors. For the model shown in Figure 4, the joint density inside the integral in Equation (2) would factorize into terms that predict the sequence element (or token) $\mathbf{o}$ from its associated latent variable and into terms that predict the next latent variable from preceding values:

$$\begin{aligned} p([\mathbf{o}_1, \ldots, \mathbf{o}_i], [\mathbf{s}_1, \ldots, \mathbf{s}_i]; \theta) &= \prod_{k \leq i} p(\mathbf{o}_k|\mathbf{s}_k) p(\mathbf{s}_k|[\mathbf{s}_1, \ldots, \mathbf{s}_{k-1}]; \theta) \\ p(\mathbf{o}_k|\mathbf{s}_k) &= \delta(\mathbf{s}_k - \mathbf{E} \cdot (\mathbf{o}_k + \boldsymbol{\varepsilon}_k)) \\ p(\mathbf{s}_k|[\mathbf{s}_1, \ldots, \mathbf{s}_{k-1}]; \theta) &= \varphi(\mathbf{s}_k, \phi([\mathbf{s}_1, \ldots, \mathbf{s}_{k-1}])) \end{aligned} \tag{3}$$

The matrix $\mathbf{E}$ is the embedding matrix (i.e., a linear mapping from token vectors – often high-dimensional one-hot vectors – to a lower dimensional metric space), while $\boldsymbol{\varepsilon}$ is a position-specific perturbation. The embedding into a metric space is achieved by the matrix-vector multiplication of $\mathbf{E}$ with the sum of the vectors $\mathbf{o}$ and $\boldsymbol{\varepsilon}$. The third line of Equation (3) is of particular interest for us, in that the choice of function Φ determines which of the previous vectors in the sequence contribute to the next. The Markovian case corresponds to $\Phi([\mathbf{s}_1, \ldots, \mathbf{s}_{i-1}]) = \Phi(\mathbf{s}_{i-1})$. For the non-Markovian case, we need to determine which priors to express in this function. It may seem strange to refer to functions and conditional dependency structures as 'prior beliefs.' However, this is a deliberate and technical use of prior beliefs – in the sense of Bayesian belief updating and propagation – with the aim of making the assumptions of the transformer architecture as transparent as possible. Here, the prior in question is a structural prior that a context sensitive linear combination of previous sequence elements is sufficient to predict future sequence elements. And that the coefficients of this linear combination are functions of metric relations between vector hidden states representing those previous elements. This functional form is imposed on the model *a priori* rather than being learned.

Typically, the φ function used in Equation (3) is the composition of a set of interleaved linear and non-linear functions – as is typical in artificial neural networks. Each of these functions will contain their own set of parameters. In what follows, we will drop the explicit dependence on parameters for simplicity. The parameters for Φ will include a matrix $\mathbf{R}$ that has a role in determining attention and those for φ will include weight matrices $\mathbf{W}$ (see Figure 5) that are involved in the connections between layers in a neuronal network. A probabilistic (factor graph) interpretation of this composition is shown in Figure 5. Our focus will be on the Φ function (the 'attention head' in Figure 5).

Let us say that we wish to express the prior belief – inheriting from the metric embedding – that similar vectors in the embedding space are likely to predict similar next states. Note that the perturbation $\boldsymbol{\varepsilon}$ in Equation (3) means that 'similar' here means both similarity in 'what' the sequence element is and in 'where' it is in the sequence. The notion of characterizing similarity tells us something profound about this approach. It implies the important information is held not in single tokens, but in pairs of tokens. A simple way to measure similarity in an embedding space like that in Figure 4 is to use the dot product between pairs of vectors (Vaswani et al., 2017):

$$[\mathbf{s}_1, \ldots, \mathbf{s}_{i-1}]^T \mathbf{R} [\mathbf{s}_1, \ldots, \mathbf{s}_{i-1}] = \begin{bmatrix} \mathbf{s}_1 \cdot \mathbf{R}\mathbf{s}_1 & \mathbf{s}_1 \cdot \mathbf{R}\mathbf{s}_2 & \cdots \\ \mathbf{s}_2 \cdot \mathbf{R}\mathbf{s}_1 & \mathbf{s}_2 \cdot \mathbf{R}\mathbf{s}_2 & \\ \vdots & & \ddots \end{bmatrix} \tag{4}$$

For generality, the dot products in Equation (4) are augmented by interposing a matrix $\mathbf{R}$—analogous to a metric tensor, but without the constraints that it must be symmetrical. In the deep learning literature, $\mathbf{R}$ is typically decomposed into two matrices (for historical reasons, referred to as key-value matrices) that allow an interpretation in terms of affine transformations of the two vectors being

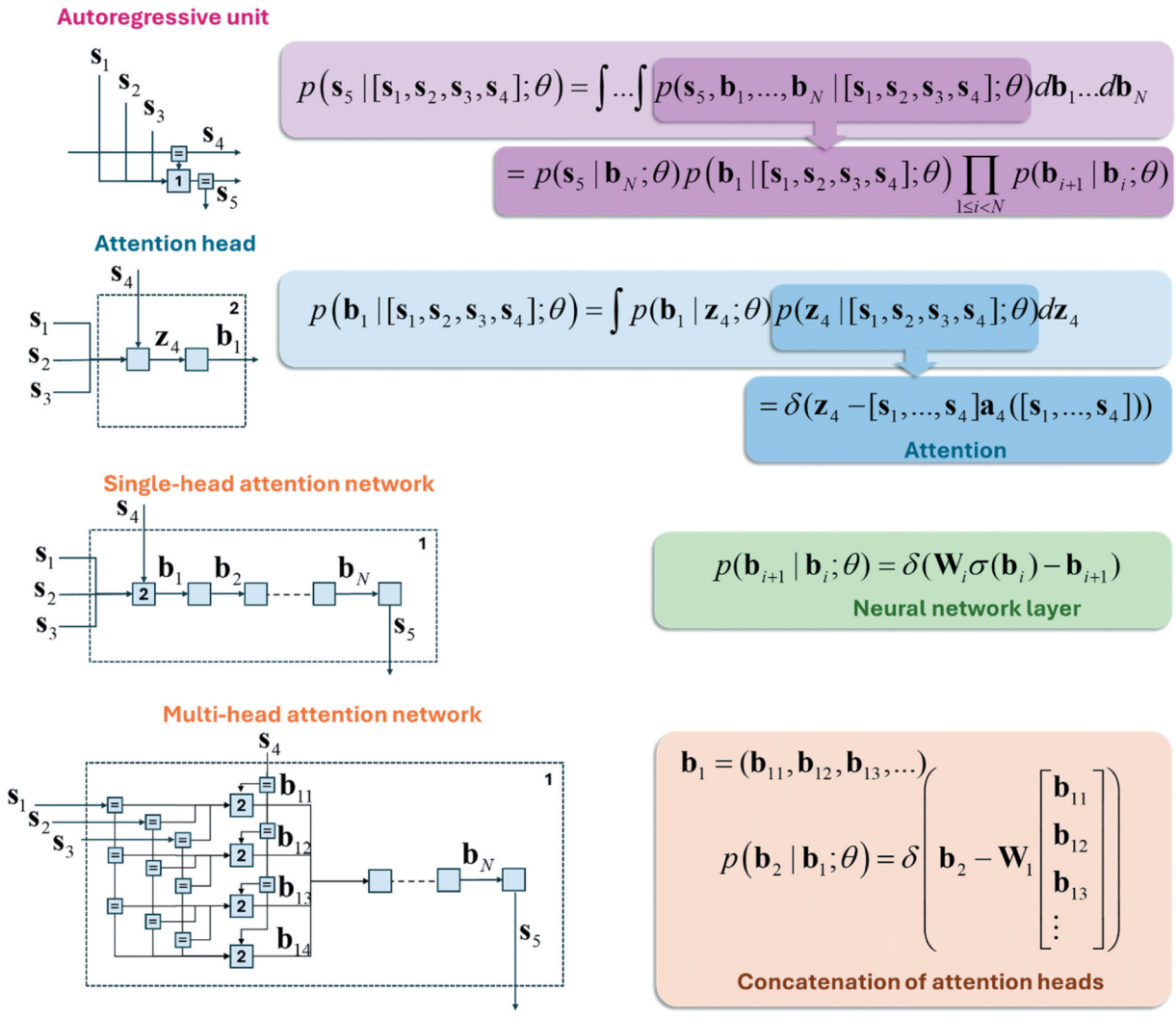

Figure 5. This figure is designed to supplement the previous figures with additional technical detail and can be safely ignored by those seeking a high-level understanding of the ideas set out in the main text. The key point emphasized by this figure is that, as in deep temporal models, there are multiple layers implicit in transformer architectures. However, these layers have a very different interpretation. In transformer architectures, all layers are contained within the single factor labeled **1** in the upper left factor graph (a recurring motif from both Figures 3 and 4). We can understand these layers by treating factor **1** as the result of repeated marginalization operations (i.e., integrating out dummy variables). The single and multi-headed attention networks unpack two ways of constructing the autoregressive unit using an attention head factor (labeled **2**) and a series of factors that represent layers of a deep neural network (collectively implementing the function φ from Equation 3). The activation of each layer of neurons is given by a vector **b** and depends upon weights **W**. The output of the final layer parameterizes the probability distribution over the next state. One can get from the single-head attention network to the autoregressive unit by integrating with respect to all variables on edges contained entirely within the dashed box ('closing the box' (Loeliger et al., 2007)). The attention head factor can be similarly unpacked into a factor generating our **z** variable – that summarizes relevant past information – and another factor that transforms this into the first layer of the neural network. The lower row of the figure shows the more complex organization of factor **1** (the autoregressive unit) when multiple attention heads are used. Here, the inputs (**s**) are replicated so that each attention head (factors labeled **2**) receives the same input. This means a separate $\mathbf{b}_1$ vector for each attention head, labeled with subscripts $\mathbf{b}_{1i}$, that are combined through concatenation and multiplication with the first weight matrix to obtain $\mathbf{b}_2$. Yet more complex configurations can be obtained by stacking together multiple multi-head attention networks, such that the output of one is the input for the next. The two big differences between the layered structure of a deep temporal model compared to a transformer network are: (1) the former are vertically organized, dealing with different temporal scales. In contrast, the latter are arranged horizontally, dealing with intermediate stages in the mapping of past to future. (2) the layers in a transformer are almost all deterministic as indicated by the Dirac delta (δ) functions and therefore admit no uncertainty in these mappings.

compared. For instance, instead of comparing similarities ($\mathbf{R} = \mathbf{I}$), with the right choice of affine transformation we could compare differences ($\mathbf{R} = -\mathbf{I}$). Intuitively, this means synonyms and antonyms could both be similarly predictive. Using a softmax – or 'normalized exponential' – function (σ), the quantity in Equation (4) can be used to construct weightings for any given input vector that is contextualized by previous samples:

$$\mathbf{A}([\mathbf{s}_1, \ldots, \mathbf{s}_{i-1}]) = \sigma\left(\Delta + [\mathbf{s}_1, \ldots, \mathbf{s}_{i-1}]^T \mathbf{R} [\mathbf{s}_1, \ldots, \mathbf{s}_{i-1}]\right)$$
$$\Delta = \begin{bmatrix} 0 & 0 & 0 & \cdots \\ -\infty & 0 & 0 & \\ -\infty & -\infty & 0 & \\ \vdots & & & \ddots \end{bmatrix} \tag{5}$$

The matrix Δ ensures a restriction such that each sequence element can only be contextualized by previous elements of the sequence (i.e., no influences from future to past). The use of a softmax function here effectively suppresses small values in the matrix, rendering them effectively zero. This induces a form of sparsity in that 'attending' to one previous element of the sequence implicitly means 'ignoring' others. Now, say we have the (i-1)-*th* element of the sequence as a prompt and want to know what comes next. We can utilize the attention matrix $\mathbf{A}$ (or its (i-1)-*th* column, $\mathbf{a}_{i-1}$) to modify our prompt such that it takes account of relevant context:

$$\begin{aligned} [\mathbf{z}_1, \ldots, \mathbf{z}_{i-1}] &= [\mathbf{s}_1, \ldots, \mathbf{s}_{i-1}]\mathbf{A}([\mathbf{s}_1, \ldots, \mathbf{s}_{i-1}]) \Rightarrow \\ \mathbf{z}_{i-1} &= \phi([\mathbf{s}_1, \ldots, \mathbf{s}_{i-1}]) \\ &= [\mathbf{s}_1, \ldots, \mathbf{s}_{i-1}]\mathbf{a}_{i-1}([\mathbf{s}_1, \ldots, \mathbf{s}_{i-1}]) \\ \mathbf{A}([\mathbf{s}_1, \ldots, \mathbf{s}_{i-1}]) &= [\mathbf{a}_1([\mathbf{s}_1, \ldots, \mathbf{s}_{i-1}]), \mathbf{a}_2([\mathbf{s}_1, \ldots, \mathbf{s}_{i-1}]), \ldots] \end{aligned} \tag{6}$$

Each $\mathbf{z}$ vector is a context-sensitive summary (or compression) of all elements up to a given point in the sequence: i.e., $\mathbf{z}_4$ is a summary of [$\mathbf{s}_1$, $\mathbf{s}_2$, $\mathbf{s}_3$, $\mathbf{s}_4$] that lives in the same metric space as the vectors of previous sequence elements. When transformers are trained to be used as large language models, they may be used to map a series of inputs to themselves, shifted in time. This motivates the form of the first line in Equation (6). In principle, one can use the above to recursively sample the next element in a sequence. In practice, transformer architectures do not just use a single attention head. They utilize a weighted combination of several attention heads (Vaswani et al., 2017), as highlighted in the factor graph in the lower row of Figure 5. The use of multi-headed attention allows a transformer not only to learn several alternative attentional policies – which might emphasize different features or positions in sequences – but to learn how to select between these attentional sets. While not shown in the figures here, most transformer architectures will stack together architectures like those in Figure 5 so that attentional mechanisms are applied repeatedly.

We conclude this section by highlighting the way in which attention – in the transformer sense – solves non-Markovian problems in which memory is essential. Rather than utilize all elements of the past, in trying to predict the next element of the sequence, attentional mechanisms exploit the spatial structure of the embedding space of tokens (e.g., words). They augment the vector representing the most recent token with relevant (i.e., attended) previous elements. This means that in place of using a complete history to predict the next element, we make use of a latent variable whose value depends upon those elements of the past committed to – or perhaps summarized in – a form of working memory.

4. Temporal hierarchies

In the introduction, we highlighted two possible routes to capturing non-Markovian sequences. In Section 3, we unpacked the implicit generative model that underwrites the attentional autoregressive-like method in transformer architectures. In doing so, we made use of nonstandard – from the perspective of deep learning – formalisms. Our reasons for doing so were that: (1) the probabilistic generative model perspective affords transparency as to the assumptions we make about how data are generated; (2) this formulation makes it easier to compare with deep temporal models that are closer to functional brain architectures; and (3) a common formalism facilitates translation of advances between different formulations.

Here, we unpack the second route to modeling non-Markovian sequences, based upon deep temporal models. Deep temporal models have much in common with the logic of convolutional neural networks (LeCun et al., 2015). The idea in both cases is that in drawing inferences about some data with a (spatio)temporal structure, one can recursively summarize local regions in the dataset. With hierarchical (or convolutional) layers of increasingly coarse resolution, one accounts for longer-range dependencies between sequence elements (Friston et al., 2024).

The basic structure of such models can be expressed recursively as follows:

$$p(\mathbf{o}_1,\ldots,\mathbf{o}_{i_1}) = \int\ldots\int p\left([\mathbf{o}_1,\ldots,\mathbf{o}_{i_1}],\left[\mathbf{s}_1^{(1)},\ldots,\mathbf{s}_{i_1}^{(1)}\right]\right) d\mathbf{s}_1^{(1)}\ldots d\mathbf{s}_{i_1}^{(1)}$$
$$p\left([\mathbf{o}_1,\ldots,\mathbf{o}_{i_1}],\ldots,\left[\mathbf{s}_1^{(n)},\ldots,\mathbf{s}_{i_n}^{(n)}\right]\right) = \int\ldots\int p\left([\mathbf{o}_1,\ldots,\mathbf{o}_{i_1}],\ldots,\left[\mathbf{s}_1^{(n+1)},\ldots,\mathbf{s}_{i_{n+1}}^{(n+1)}\right]\right) d\mathbf{s}_1^{(n+1)}\ldots d\mathbf{s}_{i_{n+1}}^{(n+1)} \tag{7}$$

Often, models are formulated in terms of categorical latent states, so that the integrals become sums. The deep structure emerges from the following factorization:

$$p([\mathbf{o}_1,\ldots,\mathbf{o}_{i_1}],\ldots,[\mathbf{s}_1^{(n)},\ldots,\mathbf{s}_{i_n}^{(n)}]) = \prod_{k=1,\Delta k+2,2\Delta k+2,\ldots} p([\mathbf{o}_k,\ldots,\mathbf{o}_{k+\Delta k}]|[\mathbf{s}_k^{(1)},\ldots,\mathbf{s}_{k+\Delta k}^{(1)}]) \times \prod_m \prod_{k=1,\Delta k+2,2\Delta k+2,\ldots} p\left([\mathbf{s}_k^{(m)},\ldots,\mathbf{s}_{k+\Delta k}^{(m)}]|\mathbf{s}_k^{(m+1)}\right) \tag{8}$$

Figure 6 makes the factor graph representation of this explicit. Note that this factorization means sparsity is built into the architecture of such models, unlike in transformers where the sparsity emerges from the attention mechanism. The factorization in Equation (8) implies a separation of timescales (Friston et al., 2021), such that each element of a sequence at one level summarizes several elements of the sequence at the level below. In principle, this means that one can interpret one level as representing words, the level above as sentences (or perhaps clauses), the level above that as dealing in paragraphs, and so on. Here, it is easy to see that the inferences drawn at the paragraph level, from a given word in a given sentence, can contextualize words in other sentences in that same paragraph.

As an example (Donnarumma et al., 2023), consider a hierarchical language architecture that simultaneously makes inferences and predictions at the levels of words (at the lower level), paragraphs (at an intermediate level), and topic of discourse (at the higher level). When processing the word 'advancing,' the architecture would predict with high probability the words 'medical knowledge' if the inferred topic is 'Medicine' or the words 'AI methods knowledge' if the inferred topic is 'Machine Learning.' This example illustrates that hierarchical architectures – including those based upon Latent Dirichlet Allocation for topic modeling (Blei et al., 2003) – can achieve the same kind of context sensitivity – as in the case of contextual embeddings of words in large language models discussed above – but by different means: specifically, because the activity of higher layers that contextualizes the activity of lower layers, and their next-word predictions. In this perspective, the contextual representation of the word 'advancing' is distributed across all the hierarchical layers, and the next-word predictions become contextual.

But what about attention? In a sense, the deep temporal architecture – while perhaps more interpretable – gets us back to the same problem as autoregressive models. As illustrated in Figure 6, there are routes through the graph that allow any word to be informative about any other word. If this is the case, how does one select the relevant words to contextualize others? There are two answers to this. One of these is more engineered, or based upon a sensible prior, depending upon one's perspective. The other takes a more emergent perspective.

The engineered approach is to recognize that complex sequences, language being a key example, have both a semantic and a syntactic structure (Shain et al.,

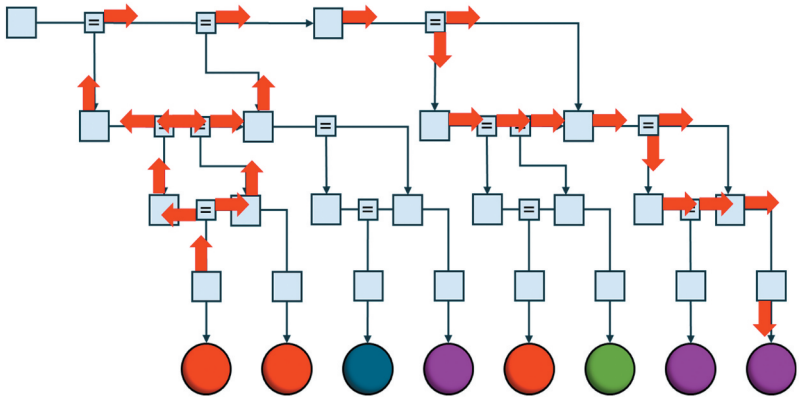

Figure 6. The factor graph shown here is an extension of the deep temporal model in Figure 2 and serves to illustrate the long-range dependencies between sequence elements via higher levels in the hierarchy. The red arrows illustrate those messages, that might be passed during inference, that allow the first token in the sequence to contextualize predictions about the last.

2024). As such, one can factorize the latent states into those representing points in a syntactic trajectory and those representing the semantics that need to be expressed at points in this trajectory. Returning to our example from earlier, shown in Figure 3, we have two possible syntaxes – those syntaxes starting with the red ball, and those starting with another color. The semantic states then jointly determine the colors of either the 6th and 8th balls, or the 7th and 8th balls, depending upon the syntax.

Figure 7 makes this explicit by illustrating a plausible state space for the model in terms of syntactic and semantic factors over three hierarchical levels (lower left). These are represented as the product of local

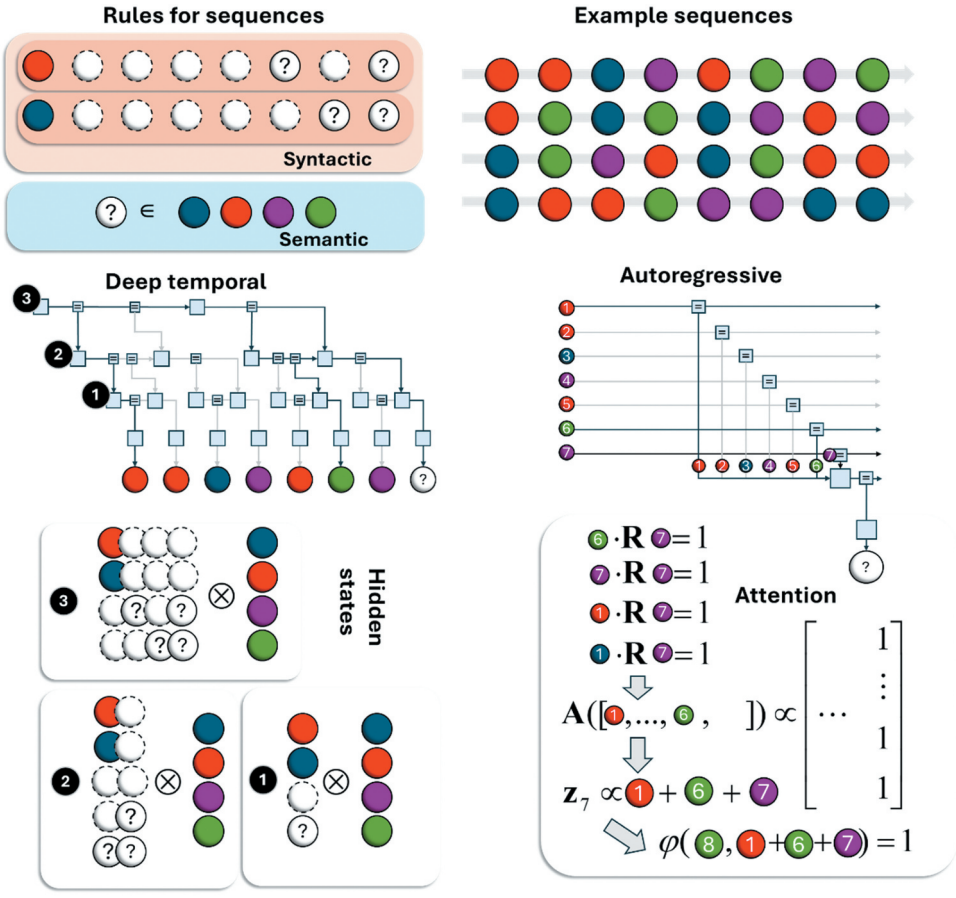

Figure 7. This figure illustrates the links between factorization and attention, while also highlighting the analogy between two forms of attention. For examples of deep temporal models that exploit this structure, see (Friston et al., 2020; Parr & Pezzulo, 2021). The upper left panel shows a set of (semantic and syntactic) rules we might use to generate sequences with long-range dependencies. In the syntactic rules, balls with dashed outlines can be of any color. Those with question marks will take on the same color as one another, and this will be determined by the semantic state. Some example sequences obeying these rules are shown in the upper right. Below this, we reproduce deep temporal and autoregressive models. Some edges could in principle be disconnected with the right attentional rules learned, and these are shown in gray as in previous figures. The three panels in the lower left show the hidden state space that might be learned in a deep temporal model capable of solving this problem. At each level, local syntactic rules are shown to be factorized from local representations of the semantic state (which would be constrained to be the same across all levels). As an example, at the lowest level (labeled 1), when the local syntactic state is '?,' one would predict an observation with the same color as the local semantic state. The lower right panel shows the form of a possible solution that might be learned using a transformer-style attention mechanism. Here, we represent the embedded vectors containing both position and color information as colored circles with numbers representing position. The **R** matrix is assumed to be such that its multiplication with specific combinations of vectors give the (arbitrarily selected) values presented in the panel. For the purposes of this example, we assume that all other combinations would lead to very negative numbers that can be safely ignored following application of the softmax operator to arrive at **A**. The resulting **A** lets us summarize the preceding sequence in terms of a vector proportional to a sum of elements 1, 6, and 7. This vector will be different depending upon whether the vector from position 1 points in the 'blue' or the 'red' direction, and similarly for the 6th and 7th vectors (which could point in the direction of any of the allowable colors). The φ function then uses this vector to parameterize a probability distribution over the possible outcomes – here assigning probability of 1 to a green ball in position 8.

syntaxes and semantic states, respectively. One could imagine that on observing the red ball in the first position, the first-level syntax is inferred to be red at level one. This propagates to level two, at which the local syntax can now be inferred to be the two-step sequence starting with red. This further propagates to the third level, at which we can now infer the local syntax comprising four balls starting with red. As we move from the first sequence of four balls to the second half of the sequence, the highest level will predict a transition to the syntax of four balls with the second and fourth being consistent with the color indicated by the semantic state. This information can now be propagated down the hierarchy; such that the inferred local syntax at the first level – when the 6th ball is observed – informs the semantic state (here, green), which allows for inferences about this state to be propagated via the second level, back to the first level, in time to predict that the 8th ball is green.

For comparison, Figure 7 also highlights how a transformer architecture might deal with the same example. Here, we have represented the embedded tokens as colored circles with numbers indicating their position, noting that these stand in for color-position vectors in some embedding space. Our non-Markovian sequence can be characterized by an $\mathbf{R}$ matrix for which multiplication of the 1st, 6th, and 7th vectors, with $\mathbf{Rs}_7$ give large values (arbitrarily 1 in the figure) but otherwise give very small or negative values. The result of this is that the final (7th) column of the attention matrix will have near-zero values for all but the 1st, 6th, and 7th elements, and it is these three vectors that are linearly combined to give $\mathbf{z}_7$. We then require that the function φ varies depending upon whether the 1st vector points more in the red or the blue direction, such that it favors a prediction of an 8th ball consistent with the 6th color when the 1st is red, and with the 7th color when the 1st is blue. That this is not an entirely satisfactory narrative reflects the 'black box' nature of the φ function – something that is often unavoidable in unpacking deep learning architectures.

As shown in Figure 7, there is an interesting point of connection here with the notion of multi-headed attention. The high-level syntax state effectively sets up two different conditional dependency structures, in which some low-level information is – or is not – propagated to higher levels, which mediate memory. In other words, the two syntaxes imply two different attentional policies and could be seen as analogous to two different attention heads. The selection between the two here would be as a weighted average between the syntaxes, based upon their relative probability given what has been observed so far. This view suggests that multiheaded attention may be simply a form of Bayesian model averaging (c.f., structure-learning of likelihood tensors (Friston et al., 2017)) under models of different attentional sets: see also (Buckley & Singh, 2024).

As alluded to above, another – more emergent – perspective on attention in deep temporal models inherits from recent developments in active learning of a generative model according to the same sort of explorative and exploitative performances when acting on the world (Friston et al., 2024). However, in place of making 'real' movements, actions in the setting of active learning rest on deciding whether to update a given model parameter. In practice, this means that our prior belief as to whether to update depends upon the degree to which a proposed update increases the mutual information (i.e., expected information gain) between edges either side of the factor being updated. In effect, this reinforces the conditional dependencies between edges of a graph that are informative about one another. One could view this as learning to attend selectively.

It is interesting to think about the relationship between the objective function for transformers and that used in active learning of the sort outlined above. Recall from Equation (1) that transformers seek to extremise the (marginal) likelihood of some sequence. On average, for the next element of the sequence, this can be expressed in terms of the following expectation:

$$
\begin{aligned}
&\mathbb{E}_{p(\mathbf{o}_i,[\mathbf{s}_1,\ldots,\mathbf{s}_{i-1}]|[\mathbf{o}_1,\ldots,\mathbf{o}_{i-1}])}[\ln p(\mathbf{o}_i|[\mathbf{o}_1,\ldots,\mathbf{o}_{i-1}])] \\
&\quad = \mathbb{E}[\ln p(\mathbf{o}_i|[\mathbf{s}_1,\ldots,\mathbf{s}_{i-1}])] \\
&\quad + \mathbb{E}[\ln p([\mathbf{s}_1,\ldots,\mathbf{s}_{i-1}]|[\mathbf{o}_1,\ldots,\mathbf{o}_{i-1}])] \\
&\quad - \mathbb{E}[\ln p([\mathbf{s}_1,\ldots,\mathbf{s}_{i-1}]|[\mathbf{o}_1,\ldots,\mathbf{o}_i])] \\
&\quad = \underbrace{\mathbb{H}[p([\mathbf{s}_1,\ldots,\mathbf{s}_{i-1}]|[\mathbf{o}_1,\ldots,\mathbf{o}_i])]}_{=0} \\
&\quad - \mathbb{H}[p(\mathbf{o}_i|[\mathbf{s}_1,\ldots,\mathbf{s}_{i-1}])] \\
&\quad - \underbrace{\mathbb{H}[p([\mathbf{s}_1,\ldots,\mathbf{s}_{i-1}]|[\mathbf{o}_1,\ldots,\mathbf{o}_{i-1}])]}_{=0} \\
&\quad = -\mathbb{H}[p(\mathbf{o}_i|[\mathbf{s}_1,\ldots,\mathbf{s}_{i-1}])]
\end{aligned}
\tag{9}
$$

The final equality uses the fact that the entropies of Dirac delta distributions are zero. In other words, in the transformer architectures outlined in the previous section, our uncertainty about a given latent state is zero, after we have made the corresponding observation. The only uncertainty left is that about the next observation (which inherits from uncertainty about the next latent state). Note that the first equality depends upon a conditional independence between an observation and all previous

observations given knowledge of the hidden states. If we express this same idea in terms of a mutual information, we have:

$$\mathbb{I}\left[\mathbf{o}_i;[\mathbf{s}_1,\ldots,\mathbf{s}_{i-1}]\right] = \mathbb{H}[p(\mathbf{o}_i)] - \mathbb{H}[p(\mathbf{o}_i|[\mathbf{s}_1,\ldots,\mathbf{s}_{i-1}])] \quad (10)$$

In other words, mutual information used in active learning of model structure includes the same conditional negentropy that is the (average) objective function for transformer models. This hints at the notion of attention as increasing the precision (decreasing the conditional entropy) being common to both transformers and active learning in deep temporal models.

Before concluding this section, it is worth mentioning that there are other generative modeling approaches that capture long-term dependencies in the data. For example, the clone-structured cognitive graph (CSCG) is an extension of Hidden Markov Models that is capable of learning context-dependent latent representations, using a variant of variational inference (i.e., expectation-maximization) (George et al., 2021). Interestingly, like transformers, the CSCG shows forms of in-context learning, by rebinding some parts of the learned cloned-structured graphs to novel inputs (Swaminathan et al., 2024).

5. Artificial and natural intelligence

This paper is not just about the technical differences between methods that might be applied to the analysis of complex sequences in machine learning. Our starting point was the notion of attention and how it relates to memory, where it is the need for memory that led us to consider analysis of non-Markovian systems. We outlined two distinct computational architectures for the kinds of generative model one might use to solve such problems. However, the important question – from a neurobiological perspective – is what sort of internal model our brain might use to solve non-Markovian problems. These range from common experimental designs like the delay-period (e.g., N-back) working memory tasks (Funahashi et al., 1989; Fuster, 1973; Zhang et al., 2013), in which humans or animals must make a decision informed by stimuli observed prior to a delay-period, through to language comprehension and generation.

Few would argue that transformers are literalist models of how the brain solves these problems. However, it is striking that multilayer transformers are currently state-of-the-art in reproducing human-like linguistic communication (Mitchell, 0000). In contrast, the deep temporal architectures we have outlined have a great deal in common with aspects of brain structure and function; including brain circuits supporting linguistic processing (Caucheteux et al., 2023; Heilbron et al., 2022). In the remainder of this section, we pose a series of questions that we hope will invite comment, and that ask how we might draw from the success of transformers in developing models of human brain function.

5.1. Is language continuous?

Intuitively, it seems that language is one of the most obviously discrete, categorical, forms of cognition and communication. We discretize into phonemes, words, sentences etc. The traditional assumption in linguistics is that language is composed of discrete units. Strikingly, one of the reasons that transformers appear to work so effectively is that their attention mechanism is based upon an underlying metric space. This space emerges from the embedding used to convert tokens, such as words, into continuous vectors with meaningful semantic spaces. The metric structure of this space allows for use of simple priors based upon vector dot products. In contrast, the expressiveness of deep temporal models – particularly in the context of active inference – has benefited from the use of categorical variables. One of the main reasons for this is that expressing variables in terms of their possible alternative values simplifies the problem of formulating and evaluating alternative action plans or trajectories. This is a problem we will return to in our final question. For now, it is interesting to consider whether augmenting these categorical state spaces with continuous variables (Parr & Friston, 2018b) might have benefits in terms of the implicit metric structures one can express in this form. Alternatively, should these metric properties be expressed in terms of the conditional relationships between categorical variables (Friston et al., 2024)? For example, one could impose a metric relationship between three categorical states by setting up a transition function in which the only route from state one to state 3 is via state 2 and *vice versa*.

5.2. Is next-word prediction the most effective way to learn language?

Prediction-based learning is increasingly popular in machine learning as it provides a general and effective way to learn internal representations from data without external supervision. This might be the case because making good predictions requires learning the most informative regularities from data (Bialek et al., 2001). So far, the simplest forms of prediction – next-word

prediction – has proven to be surprisingly effective in training large language models. However, it is possible that biological brains make (and learn from) predictions at multiple levels and timescales simultaneously, as assumed in hierarchical architectures. Predictions made at different levels can reciprocally influence and contextualize each other, potentially making a hierarchical architecture better able to capture regularities that unfold at different timescales, and to support lookahead planning. Multi-level (or multi-token) prediction is now explored in both machine learning as a potentially more effective way to train large language models (Gloeckle et al., 2024) and in computational neuroscience as a potentially more effective way to explain how the brain processes language (Heilbron et al., 2022). It remains to be seen if a capability for long-term prediction is something that needs to be engineered or could emerge from next-word prediction.

5.3. Retention or summarization?

One of the least biologically plausible features of a transformer network is that – although not used to the same extent – all previous elements of the sequence are retained. Memory in this setting is simply the caching of previous observations, which are compressed in the process of predicting the next state. This differs to how deep temporal models account for memory. As one moves to progressively higher levels of a deep temporal model, the variables at each level are summaries of the sequences (i.e., short trajectories) at the level below. This means higher levels deal with variables that change more slowly than the lower levels (Kiebel et al., 2008). It also means that irrelevant information does not make it to the higher levels, thus avoiding the need for – computationally infeasible – direct retention of long records of low-level sensory data. As it is the higher levels that propagate information over longer time spans, this means that memories of the past are treated not as high-fidelity accounts of what has happened but as compressed summaries of what is necessary to contextualize an ongoing narrative (Parr & Friston, 2017b). Both this compression and the compression implicit in the construction of $\mathbf{z}$ in Equation (6) induce an information bottleneck (Still, 2014), although (arguably) deep temporal models make use of a more transparent compression as one ascends hierarchical layers – before passing messages back down the hierarchy – whereas the transformer model has a more transductive approach (Vapnik, 2000).

This also has implications for the way in which we understand attention – whether it is something exclusively applied in time, or whether it is something that can be applied moment-to-moment. As we have framed it here, the logic behind an attention mechanism in a transformer network is that it provides a sophisticated form of (ordinal or temporal) convolution kernel. This has an exclusively temporal aspect to it. In contrast, attention in the context of a temporal hierarchy is more often applied to the vertical rather than the horizontal links (Kanai et al., 2015). In language, some words in a sentence may matter for the semantics of that sentence while others are largely interchangeable. The former are 'attended,' and the latter are 'ignored.' Transformers pay attention to things they have previously observed in their cached memory bank. In other words, transformers can *learn* how to attend by learning contextual dependencies, while hierarchical schemes learn to infer the context that specifies attentional set. Brains pay attention to alternative sources of information currently available in their sensorium. This latter facet of attention is related to the notions of *precision* and uncertainty, which we discuss next.

5.4. Does uncertainty matter, and if so, uncertainty about what?

A follow-on question is whether it is possible to have attention without uncertainty. As formulated by predictive processing theories of brain function, attention is simply differential confidence – or precision – in different probability distributions (Feldman & Friston, 2010). The transformer architecture brings all uncertainty into the dynamics. This means that the past is always known with absolute confidence, even if the future is not fully predictable. This results in significant savings, computationally, because it eliminates the need for any kind of (Bayesian) smoothing or reciprocal message passing when performing inference. Messages can be passed exclusively forwards in time. This raises the question as to whether uncertainty about the past is useful, or whether we should always look forwards (FitzGerald et al., 2020).

However, connections in the brain are not unidirectional. Detailed neuroanatomical observations confirm a network of hierarchically organized regions whose connectivity is highly reciprocal (Felleman & Essen, 1991; Shipp, 2007; Zeki & Shipp, 1988). Provided there is some uncertainty at play, so is the requisite message passing that would invert a deep temporal model (Friston et al., 2017). An interesting feature of this reciprocity is that – to get from a stimulus to predictions about the next stimulus – it is not always necessary to engage all levels of a hierarchy. Sometimes, local information is sufficient, and any uncertainty at the lowest level can be resolved without resorting to message passing all the way up and back down again (Tscshantz et al., 2023). Of note, this

argument has also been used for the benefits of reciprocity in the motor system, with robotics studies showing that local feedback loops result in much faster and more efficient control (Ijspeert & Daley, 2023). Transformers do not work like this. As is evident from Figures 5 and 6, the composition of functions in serial means that one must evaluate every function, and every layer of a neural network, for every query. This may be the cost of sacrificing reciprocity.

5.5. Learning or exploration?

One of the points discussed above was that the objective function, combined with the generative model, for the attention mechanism in transformer architectures is interpretable as one part of a mutual information or expected information gain (Lindley, 1956). Such quantities are used in the study of exploration in computational neuroscience (Manohar & Husain, 2013; Mirza et al., 2018; Pezzulo et al., 2016; Piray & Daw, 2024). They are also used in recent work (Friston et al., 2024) that treats the problem of optimizing a model as a form of active learning or exploration in parameter space. This is particularly interesting from the perspective of attention, where attention is interpreted as deployment of predictions that a particular source of information is highly precise. Choosing to optimize a model in such a way that it has a high mutual information between latent states and observations implicitly means reducing the conditional entropy of observations given latent states. In other words, active learning of this sort is the optimization of attention.

5.6. Is attention just factorization?

In Section 4, we highlighted the interesting parallel between the use of weighted outputs from multiple attention heads and the use of a syntactic hidden state that contextualizes which sequence elements are semantically valuable. Depending upon the syntax in play, different positions in a sentence (or other sequence) provide the most precise information about the semantic content of that sentence (see for example (Friston et al., 2020). The same principle applies to attention in other domains. Perhaps most simply, consider the deployment of overt attention in the visual domain. Overt attention is the term given to the process of actively moving one's eyes to pay attention to a specific location in space (Itti & Koch, 2000). If one were to formulate a generative model of active vision (Heins et al., 2020; Mirza et al., 2016), this might include the position of one's eyes and the semantic content of a visual scene. Here, it is clear that the precision of the mapping between the latent semantic content and the visual data available to our retina will depend upon the position of our eyes (Parr & Friston, 2017a). One factor, the position of the eyes, modulates the gain of the other factor, the semantic content of the scene.

The purpose of these examples is to illustrate the broader point that when one factorizes the latent state space used to predict some data – or to predict the states at a lower level – the different factors contextualize the relationships between each other and the data being predicted. This also means that when the model is inverted – i.e., when we draw inferences from data – the neurons representing beliefs about one factor will appear to direct attention from the perspective of neurons representing beliefs about another factor. Arguably, this means that attention is a ubiquitous feature of a factorized state space.

5.7. What, when, and where?

Drawing from the above, it is striking that positional encoding is something that happens at the first (embedding) step that provides the input to a transformer architecture (Dufter et al., 2022), while our brains seem to go to great lengths to pull apart (i.e., factorize) information about when, where, and what something is (Friston & Buzsaki, 2016). In the visual domain, messages from the primary visual cortex are propagated via two main anatomical pathways (Ungerleider & Haxby, 1994). A dorsal parietal pathway deals with information about where something is. A ventral temporal pathway deals with the identity of that thing. Similar distinctions have been made in the context of language (Bates et al., 1988; Hickok, 2012; Hickok & Poeppel, 2007), arising from the auditory cortices, with a ventral temporal stream dealing in the semantics and a dorsal frontal stream dealing with the organization and production of grammatical speech. The classic neuropsychological observations of Broca and Wernicke are sometimes taken to show a double dissociation between lesions causing speech devoid of grammar (Broca's aphasia) and those causing speech devoid of semantic content (Wernicke) (Bates et al., 1988; Sun & Manohar, 2023). One might conclude from this that our brains exploit factorization – which has computational benefits (through a form of modularization) (Parr et al., 2020) and enables efficient generalization (i.e., the same syntax can be used to express several different semantic ideas).

The question that arises from this is whether transformers, or those who prepare their embedded inputs, have made an error by condensing position and identity

at the earliest processing stage of embedding. Given natural selection appears to have taken the opposite approach, would it be more fruitful to take inspiration from how our brains solve problems of sequences? There are (at least) two possible rebuttals to this question. First, it is important to get the right kind of factorization, and perhaps there are better ways to carve the problem of sequence generation than the naïve position-by-identity decomposition. This may be one of the key benefits of the embedding space, whose dimensions represent a different sort of factorization that brings out other forms of meaningful structure. This answer suggests that as neuroscientists, we might think again about overly simplistic understanding of decompositions in sensory processing streams (Franz et al., 2000; McIntosh & Schenk, 2009; Milner & Goodale, 2008). Is there a more meaningful embedding occurring even at the level of primary sensory cortices whose structure is exploited by subsequent factorizations?

The other possible answer is that the multiheaded attention itself is a form of factorization into syntax and semantics. As in Figure 7, the weighting between attention heads might be a form of (soft) selection between different syntactic structures, which then allow for the semantic content to be interpreted to predict the next word. In other words, is the factorization into the weights for concatenated outputs from each attention head, and the content of each head, the same as the semantic-syntactic factorization often used in smaller scale language models in neurobiology?

5.8. What about agency and purpose?

Our final question is at least partially answerable in the context of deep temporal models (Friston et al., 2023) but is also interesting to consider in the transformer context. How do we understand non-Markovian models in relation to action and agency. At a very basic level, both forms of model – when used to generate sequences – are engaging in a form of active inference. At its simplest, active inference is the process of aligning one's internal model with the world and aligning (observations of) one's world with one's internal model (c.f., self-evidencing (Hohwy, 2016)). In practice, this means action is a process of reflexive fulfillment of predictions (Adams et al., 2013). Using the models outlined above – whose world is effectively the set of available (sequences of) tokens – for generation of novel stimulus streams depends upon precisely this action through fulfillment of prediction.

However, this elemental (a.k.a., merely reflexive) form of active inference only takes us so far. An important aspect of memory is the notion of temporal depth in the sense of looking back in time. This also implies one can look forward in time – something we must do to engage in planning and prospective inference (Botvinick & Toussaint, 2012). There is a profound difference between systems that do and do not project forwards in time. When projecting forwards in time, active systems must account for their own actions to predict the data they will observe. Here, we have a form of agency, in which our brains must model themselves – or at least the consequences of their actions (Pezzulo & Rigoli, 2011; Pezzulo et al., 2011) – as an explanation of the data they will solicit.

In deep temporal models, this problem has been articulated in terms of model selection between alternative trajectories; i.e., in terms of alternative transition dynamics (Friston et al., 2017). The resulting message passing has interesting points of contact with the connectivity between cortical and subcortical structures in the brain (Parr & Friston, 2018a). Could the same selection between alternative transitions be implemented for transformers to give a greater degree of autonomy? There appears to be a clear biological relevance to this question, as illustrated by recent work that looks at how trajectories through semantic embedding spaces varies with diagnostic categories (Nour et al., 2023). Should this work be understood in terms of biologically plausible hierarchical models augmented with metric representations, or more transformer-like autoregressive models that must actively select between trajectories? At a more fundamental level, biological systems have a say in what it is they are 'trained' on. Will future large language models be trained on the large datasets currently in use, or will they have the autonomy to select – perhaps by engaging in conversation – those data that they feel to be most informative?

6. Conclusion

This paper has raised more questions than it answers. This is deliberate in that we hope to promote further discussion on the issues raised. Large language models, including those using transformer architectures, have been highly successful in generating samples of plausible prose in a biologically implausible manner. We suggest that the reason they can do this is that they address the issue of how to model non-Markovian sequences – or memory-dependent sequences. The transformer solution to this is to learn attentional priors that make use of the metric space implicit in the embedding of sequence elements as continuous vectors. The primary question this article poses is whether there is something

fundamental about this link between attention and memory, and whether theoretical neurobiology can learn from the solutions artificial intelligence offers to these problems. We look forward to ongoing discussion.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

TP is supported by an NIHR Academic Clinical Fellowship [ref: ACF-2023-13-013]. KF is supported by funding from the Wellcome Trust [Ref: 203147/Z/16/Z]. GP is supported by the European Research Council under the Grant Agreement No. 820213 (ThinkAhead), the Italian National Recovery and Resilience Plan (NRRP), M4C2, funded by the European Union – NextGenerationEU [Project IR0000011, CUP B51E22000150006, "EBRAINS-Italy"; Project PE0000013, "FAIR"; Project PE0000006, "MNESYS"], and the [PRIN PNRR P20224FESY].

ORCID

Thomas Parr http://orcid.org/0000-0001-5108-5743

References

Adams, R. A., Shipp, S., & Friston, K. J. (2013). Predictions not commands: Active inference in the motor system. *Brain Structure & Function*, *218*(3), 611–643.

Athanasopoulos, G., Hyndman, R. J., Kourentzes, N., & Petropoulos, F. (2017). Forecasting with temporal hierarchies. *European Journal of Operational Research*, *262* (1), 60–74. https://doi.org/10.1016/j.ejor.2017.02.046

Bakry, D., Gentil, I., & Ledoux, M. (2014). *Analysis and geometry of Markov diffusion operators* (Vol. 103). Springer.

Bates, E. A., Friederici, A. D., Wulfeck, B. B., & Juarez, L. A. (1988). On the preservation of word order in aphasia: Cross-linguistic evidence. *Brain and Language*, *33*(2), 323–364. https://doi.org/10.1016/0093-934X(88)90072-7

Bialek, W., Nemenman, I., & Tishby, N. (2001). Predictability, complexity, and learning. *Neural Computation*, *13*(11), 2409–2463. https://doi.org/10.1162/089976601753195969

Bishop, C. M., & Bishop, H. (2023). *Transformers, in deep learning: Foundations and concepts*. Springer.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, *3*(Jan), 993–1022.

Botvinick, M., & Toussaint, M. (2012). *Planning as inference*. Trends in cognitive sciences. *Trends in Cognitive Sciences*, *16*(10), 485–488. https://doi.org/10.1016/j.tics.2012.08.006

Buckley, C., & Singh, R. (2024). Attention as implicit structural inference. *Advances in Neural Information Processing Systems* 36 (2023): 24929-24946. https://proceedings.neurips.cc/paper_files/paper/2023/file/4e8a74988bc611495c2d3a5edac8493f-Paper-Conference.pdf

Buschman, T. J., & Miller, E. K. (2007). Top-down versus bottom-up control of attention in the prefrontal and posterior parietal cortices. *Science*, *315*(5820), 1860–1862. https://doi.org/10.1126/science.1138071

Caucheteux, C., Gramfort, A., & King, J.-R. (2023). Evidence of a predictive coding hierarchy in the human brain listening to speech. *Nature Human Behaviour*, *7*(3), 430–441.

Cho, K., Merriënboer, B. V., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv Preprint arXiv: 14061078*.

Crutchfield, J. P., & Young, K. (1989). Inferring statistical complexity. *Physical Review Letters*, *63*(2), 105–108. https://doi.org/10.1103/PhysRevLett.63.105

Dar, G., et al. (2023). Analyzing transformers in embedding space. *arXiv Preprint arXiv: 220902535*. https://doi.org/10.48550/arXiv.2209.02535

Dauwels, J. (2007). *On variational message passing on factor graphs*. In *Information Theory, 2007. ISIT 2007. IEEE International Symposium on* Nice, France, IEEE.

Deyle, E. R., & Sugihara, G. (2011). Generalized theorems for nonlinear state space reconstruction. *PLOS ONE*, *6*(3), e18295.

Donnarumma, F., Frosolone, M., & Pezzulo, G. (2023). Integrating large language models and active inference to understand eye movements in reading and dyslexia. *arXiv Preprint arXiv: 230804941*. https://doi.org/10.48550/arXiv.2308.04941

Dufter, P., Schmitt, M., & Schütze, H. (2022). Position information in transformers: An overview. *Computational Linguistics*, *48*(3), 733–763. https://doi.org/10.1162/coli_a_00445

Feldman, H., & Friston, K. (2010). Attention, uncertainty, and Free-Energy. *Frontiers in Human Neuroscience*, *4*(215). https://doi.org/10.3389/fnhum.2010.00215

Felleman, D. J., & Essen, D. C. V. (1991). Distributed hierarchical processing in the Primate cerebral cortex. *Cerebral Cortex*, *1* (1), 1–47. https://doi.org/10.1093/cercor/1.1.1

FitzGerald, T. H. B., Penny, W. D., Bonnici, H. M., & Adams, R. A. (2020). Retrospective inference as a form of bounded rationality, and its beneficial influence on learning. *Frontiers in Artificial Intelligence*, 3, 1-14. doi:10.3389/frai.2020.00002

Forney, G. D. (2001). Codes on graphs: Normal realizations. *IEEE Transactions on Information Theory*, *47*(2), 520–548.

Franz, V. H. (2000). Grasping visual illusions: No evidence for a dissociation between perception and action. *Psychological Science*, *11*(1), 20–25.

Freeman, W. J. (1987). Simulation of chaotic EEG patterns with a dynamic model of the olfactory system. *Biological Cybernetics*, *56*(2–3), 139–150. https://doi.org/10.1007/BF00317988

Friston, K. (2017). Active inference: A process theory. *Neural Computation*, *29*(1), 1–49.

Friston, K. (2024). From pixels to planning: Scale-free active inference. *arXiv Preprint arXiv: 240720292*, 1–64. https://doi.org/10.48550/arXiv.2407.20292

Friston, K., & Buzsaki, G. (2016). The functional anatomy of time: What and when in the brain. *Trends in Cognitive Sciences*, *20* (7), 500–511. https://doi.org/10.1016/j.tics.2016.05.001

Friston, K. J. (2017). *Active inference, curiosity and insight*. Neural Computation.

Friston, K. J. (2020). Generative models, linguistic communication and active inference. *Neuroscience & Biobehavioral Reviews*, *118*, 42–64. https://doi.org/10.1016/j.neubiorev.2020.07.005

Friston, K. J., Da Costa, L., Tschantz, A., Kiefer, A., Salvatori, T., Neacsu, V., Koudahl, M., Heins, C., Sajid, N., Markovic, D., Parr, T., Verbelen, T., & Buckley, C. L. (2024). Supervised structure learning. *Biological Psychology*, *193*, 108891. https://doi.org/10.1016/j.biopsycho.2024.108891

Friston, K. J., Fagerholm, E. D., Zarghami, T. S., Parr, T., Hipólito, I., Magrou, L., & Razi, A. (2021). Parcels and particles: Markov blankets in the brain. *Network Neuroscience*, *5*(1), 211–251.

Friston, K. J., Salvatori, T., Isomura, T., Tschantz, A., Kiefer, A., Verbelen, T., Koudahl, M., Paul, A., Parr, T., Razi, A., Kagan, B. J., Buckley, C. L., & D, M. J. (2025). Ramstead; active inference and intentional behavior. *Neural computation*, *37*(4), 666–700. https://doi.org/10.1162/neco_a_01738

Friston, K. J., Parr, T., & de Vries, B. (2017). The graphical brain: Belief propagation and active inference. *Network Neuroscience (Cambridge, Mass)*, *1*(4), 381–414.

Friston, K. J., Rosch, R., Parr, T., Price, C., & Bowman, H. (2017). Deep temporal models and active inference. *Neuroscience & Biobehavioral Reviews*, *77*, 388–402. https://doi.org/10.1016/j.neubiorev.2017.04.009

Funahashi, S., Bruce, C. J., & Goldman-Rakic, P. S. (1989). Mnemonic coding of visual space in the monkey's dorsolateral prefrontal cortex. *Journal of Neurophysiology*, *61*(2), 331–349.

Fuster, J. M. (1973). Unit activity in prefrontal cortex during delayed-response performance: Neuronal correlates of transient memory. *Journal of Neurophysiology*, *36*(1), 61–78.

George, D. (2021). Clone-structured graph representations enable flexible learning and vicarious evaluation of cognitive maps. *Nature Communications*, *12*(1), 2392.

George, D., Hawkins, J., & Friston, K. J. (2009). Towards a mathematical theory of cortical micro-circuits. *PLOS Computational Biology*, *5*(10), e1000532.

Gloeckle, F. (2024). Better & faster large language models via multi-token prediction. *arXiv preprint arXiv: 240419737*. https://doi.org/10.48550/arXiv.2404.19737

Goldstein, A. (2022). Shared computational principles for language processing in humans and deep language models. *Nature Neuroscience*, *25*(3), 369–380.

Goldstein, A. (2024). Alignment of brain embeddings and artificial contextual embeddings in natural language points to common geometric patterns. *Nature Communications*, *15*(1), 2768.

Gottlieb, J. (2023). Emerging principles of attention and information demand. *The Current Directions in Psychological Science*, *32*(2), 152–159. https://doi.org/10.1177/09637214221142778

Heilbron, M., Armeni, K., Schoffelen, J.-M., Hagoort, P., & de Lange, F. P. (2022). A hierarchy of linguistic predictions during natural language comprehension. *Proceedings of the National Academy of Sciences*, *119*(32), e2201968119. https://doi.org/10.1073/pnas.2201968119

Heins, R. C. (2020). Deep active inference and scene construction. *Frontiers in Artificial Intelligence*, *3*(81). doi:10.3389/frai.2020.509354

Hickok, G. (2012). Computational neuroanatomy of speech production. *Nature Reviews Neuroscience*, *13*(2), 135–145. https://doi.org/10.1038/nrn3158

Hickok, G., & Poeppel, D. (2007). The cortical organization of speech processing. *Nature Reviews Neuroscience*, *8*(5), 393–402.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Hohwy, J. (2016). The self-evidencing brain. *Noûs*, *50*(2), 259–285. https://doi.org/10.1111/nous.12062

Ijspeert, A. J., & Daley, M. A. (2023). Integration of feedforward and feedback control in the neuromechanics of vertebrate locomotion: A review of experimental, simulation and robotic studies. *Journal of Experimental Biology*, *226*(15), jeb245784. https://doi.org/10.1242/jeb.245784

Itti, L., & Koch, C. (2000). A saliency-based search mechanism for overt and covert shifts of visual attention. *Vision Research*, *40*(10), 1489–1506.

Kanai, R., Komura, Y., Shipp, S., & Friston, K. (2015). Cerebral hierarchies: Predictive processing, precision and the pulvinar. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *370*(1668), 20140169. https://doi.org/10.1098/rstb.2014.0169

Kiebel, S. J., Daunizeau, J., Friston, K. J., & Sporns, O. (2008). A hierarchy of time-scales and the brain. *PLOS Computational Biology*, *4*(11), e1000209. https://doi.org/10.1371/journal.pcbi.1000209

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444.

Lettie, J. (2024). Explaining Markovian Time. https://philsci-archive.pitt.edu/24076/

Lindley, D. V. (1956). *On a measure of the information provided by an Experiment*. *Annals of Mathematical Statistics*, *27*(4), 986–1005.

Loeliger, H. A. (2004). An introduction to factor graphs. *IEEE Signal Processing Magazine*, *21*(1), 28–41.

Loeliger, H. A., Dauwels, J., Hu, J., Korl, S., Ping, L., & Kschischang, F. R. (2007). The factor graph approach to model-based signal processing. *Proceedings of the IEEE*, *95*(6), 1295–1322.

MacKay, D. J. C. (1992). Information-based objective functions for active data selection. *Neural Computation*, *4*(4), 590–604. https://doi.org/10.1162/neco.1992.4.4.590

Manning, C. D., Clark, K., Hewitt, J., Khandelwal, U., & Levy, O. (2020). Emergent linguistic structure in artificial neural networks trained by self-supervision. *Proceedings of the National Academy of Sciences*, *117*(48), 30046–30054. https://doi.org/10.1073/pnas.1907367117

Manohar, S. G., & Husain, M. (2013). Attention as foraging for information and value. *Frontiers in Human Neuroscience*, *7*, 711. https://doi.org/10.3389/fnhum.2013.00711

Manohar, S. G., Zokaei, N., Fallon, S. J., Vogels, T. P., & Husain, M. (2019). Neural mechanisms of attending to items in working memory. *Neuroscience & Biobehavioral Reviews*, *101*, 1–12. https://doi.org/10.1016/j.neubiorev.2019.03.017

Marković, D., Reiter, A. M. F., & Kiebel, S. J. (2019). Predicting change: Approximate inference under explicit representation of temporal structure in changing environments. *PLOS Computational Biology*, *15*(1), e1006707.

McIntosh, R. D., & Schenk, T. (2009). Two visual streams for perception and action: Current trends. *Neuropsychologia*,

47(6), 1391–1396. https://doi.org/10.1016/j.neuropsychologia.2009.02.009

Mikolov, T. (2013). Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 26. https://doi.org/10.48550/arXiv.1310.4546

Milner, A. D., & Goodale, M. A. (2008). Two visual systems re-viewed. *Neuropsychologia*, *46*(3), 774–785.

Mirza, M. B. (2016). Scene construction, visual foraging, and active inference. *Frontiers in Computational Neuroscience*, *10*(56), 1-16. doi:10.3389/fncom.2016.00056

Mirza, M. B. (2018). Human visual exploration reduces uncertainty about the sensed world. *PLOS ONE*, *13*(1), e0190429.

Mitchell, M. The turing test and our shifting conceptions of intelligence. *Science*, *385*(6710), eadq 9356. doi:10.1126/science.adq9356

Niu, Z., Zhong, G., & Yu, H. (2021). A review on the attention mechanism of deep learning. *Neurocomputing*, *452*, 48–62. https://doi.org/10.1016/j.neucom.2021.03.091

Nour, M. M. (2023). Trajectories through semantic spaces in schizophrenia and the relationship to ripple bursts. *Proceedings of the National Academy of Sciences*, *120*(42), e2305290120.

Parr, T., & Friston, K. J. (2017a). Uncertainty, epistemics and active inference. *Journal of the Royal Society Interface*, *14*(136), 1-10. http://dx.doi.org/10.1098/rsif.2017.0376

Parr, T., & Friston, K. J. (2017b). Working memory, attention, and salience in active inference. *Scientific Reports*, *7*(1), 14678. https://doi.org/10.1038/s41598-017-15249-0

Parr, T., & Friston, K. J. (2018a). The anatomy of inference: Generative models and brain structure. *Frontiers in Computational Neuroscience*, *12*(90). https://doi.org/10.3389/fncom.2018.00090

Parr, T., & Friston, K. J. (2019). Attention or salience? *Current Opinion in Psychology*, *29*, 1–5. https://doi.org/10.1016/j.copsyc.2018.10.006

Parr, T., & Friston, K. J. (2018b). The discrete and continuous brain: From decisions to movement-and back again. *Neural Computation*, *30*(9), 2319–2347.

Parr, T., Markovic, D., Kiebel, S. J., & Friston, K. J. (2019). Neuronal message passing using mean-field, Bethe, and marginal approximations. *Scientific Reports*, *9*(1), 1889. https://doi.org/10.1038/s41598-018-38246-3

Parr, T., & Pezzulo, G. (2021). Understanding, explanation, and active inference. *Frontiers in Systems Neuroscience*, 15. https://doi.org/10.3389/fnsys.2021.772641

Parr, T., Pezzulo, G., & Friston, K. J. (2022). *Active inference: The free energy principle in mind, brain, and behavior*. The MIT Press.

Parr, T., Sajid, N., & Friston, K. J. (2020). *Modules or Mean-Fields? Entropy*, *22*. https://doi.org/10.3390/e22050552

Pavlick, E. (2022). Semantic structure in deep learning. *Annual Review of Linguistics*, *8*(2022), 447–471.

Pezzulo, G. (2011). The mechanics of embodiment: A dialogue on embodiment and computational modeling. *Frontiers in Cognition*, *2*(5), 1–21.

Pezzulo, G., Cartoni, E., Rigoli, F., Pio-Lopez, L., & Friston, K. (2016). Active inference, epistemic value, and vicarious trial and error. *Learning & Memory*, *23*(7), 322–338.

Pezzulo, G., & Rigoli, F. (2011). The value of foresight: How prospection affects decision-making. *Frontiers in Neuroscience*, *5*(79), 1–15. doi:10.3389/fnins.2011.00079

Pezzulo, G., Rigoli, F., & Friston, K. J. (2018). Hierarchical active inference: A theory of motivated control. *Trends in Cognitive Sciences*, *0*(4), 294–306. https://doi.org/10.1016/j.tics.2018.01.009

Piray, P., & Daw, N. D. (2024). Computational processes of simultaneous learning of stochasticity and volatility in humans. *Nature Communications*, *15*(1), 9073.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving Language Understanding with Unsupervised Learning. OpenAI. Retrieved from https://openai.com/research/language-unsupervised

Schütz, G. M., & Trimper, S. (2004). Elephants can always remember: Exact long-range memory effects in a non-Markovian random walk. *Physical Review E*, *70*(4), 045101. https://doi.org/10.1103/PhysRevE.70.045101

Settles, B. (2011). *From theories to queries: Active learning in practice*. In *Active learning and experimental design workshop in conjunction with AISTATS 2010*, Sardinia, Italy, JMLR Workshop and Conference Proceedings.

Shain, C., Kean, H., Casto, C., Lipkin, B., Affourtit, J., Siegelman, M., Mollica, F., & Fedorenko, E. (2024). Distributed sensitivity to syntax and semantics throughout the language network. *Journal of Cognitive Neuroscience*, *36*(7), 1427–1471.

Shipp, S. (2007). Structure and function of the cerebral cortex. *Current Biology*, *17*(12), R443–R449. https://doi.org/10.1016/j.cub.2007.03.044

Shumway, R. H., Stoffer, D. S., & Stoffer, D. S. (2000). *Time series analysis and its applications* (Vol. 3). Springer.

Still, S. (2014). Information bottleneck approach to predictive inference. *Entropy*, *16*(2), 968–989.

Su, Y., Yang, Y., Shi, B., & Zhang, Y. (2023). Bayesian self-supervised learning allying with transformer powered compressed sensing imaging. *Digital Signal Processing*, *140*, 104120. https://doi.org/10.1016/j.dsp.2023.104120

Sun, L., & Manohar, S. G. (2023). Syntax through rapid synaptic changes. *bioRxiv*, 2023.12.21.572018. https://doi.org/10.1101/2023.12.21.572018

Swaminathan, S. (2024). Schema-learning and rebinding as mechanisms of in-context learning and emergence. *Advances in Neural Information Processing Systems*, 36. https://doi.org/10.48550/arXiv.2307.01201

Takens, F. (1980). *Detecting strange attractors in turbulence*. Rijksuniversiteit Groningen. Mathematisch Instituut.

Tennenholtz, G. (2023). Demystifying embedding spaces using large language models. *arXiv Preprint arXiv: 231004475*. https://doi.org/10.48550/arXiv.2310.04475

Trapp, S., Parr, T., Friston, K., & Schröger, E. (2021). The predictive brain must have a limitation in short-term memory capacity. *The Current Directions in Psychological Science*, *30*(5), 384–390.

Tscshantz, A. (2023). Hybrid predictive coding: Inferring, fast and slow. *PLOS Computational Biology*, *19*(8), e1011280.

Ungerleider, L. G., & Haxby, J. V. (1994). What' and 'where' in the human brain. *Current Opinion in Neurobiology*, *4*(2), 157–165.

van de Laar, T. W., & de Vries, B. (2019). Simulating active inference processes by message passing. *Frontiers in Robotics and AI*, *6*(20). https://doi.org/10.3389/frobt.2019.00020

van Kampen, N. G. (1998). Remarks on non-Markov processes. *Brazilian Journal of Physics*, *28*(2), 90–96.

Vapnik, V. N. (2000). *Conclusion: What is important in learning Theory?, in the nature of statistical learning theory*. Springer.

Vaswani, A. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30. https://doi.org/10.48550/arXiv.1706.03762

Yang, S. C.-H., Wolpert, D. M., & Lengyel, M. (2016). Theoretical perspectives on active sensing. *Current Opinion in Behavioral Sciences*, *11*, 100–108. https://doi.org/10.1016/j.cobeha.2016.06.009

Zeki, S., & Shipp, S. (1988). The functional logic of cortical connections. *Nature*, *335*(6188), 311–317. https://doi.org/10.1038/335311a0

Zhang, Z., Cordeiro Matos, S., Jego, S., Adamantidis, A., & Séguéla, P. (2013). Norepinephrine drives persistent activity in prefrontal cortex via synergistic α1 and α2 adrenoceptors. *PLOS ONE*, *8*(6), e66122.