Responses to Visual Event Timing in Generative Neural Network Models

Daniel Manns

7016123

Utrecht University

Responses to Visual Event Timing in Generative Neural Network Models

Abstract

Precise estimation of sub-second event timing from visual inputs is a fundamental aspect of human perception, enabling complex coordinative abilities. Early visual cortex areas exhibit monotonically increasing responses to visual event timing, turning into timing-tuned responses beginning in the medial temporal area (MT/V5). Here, we investigate whether such responses can be found in recurrent generative neural network models, unsupervisedly trained to efficiently encode visual event timing. Utilizing biologically plausible learning rules, as well as network structure, we were able to find monotonic and tuned responses to visual event timing in non-hierarchical- but not in hierarchical models. Thus, supporting the emergence of monotonic and tuned responses from inherent components of visual inputs. We showed that unsupervised recurrent generative neural networks can generally be used as models for human visual event timing. Moreover, we propose that advanced models could contribute explaining the response development along the visual hierarchy or the relationship between spatial and temporal abstraction.

Introduction

Quantifying sub-second sensory event timing is crucial to human perception and interaction with a dynamic world. Accurate estimation of timing is essential for discrimination of sensory stimuli (Buonomano & Karmarkar, 2002; Ivry & Spencer, 2004; Shannon et al., 1995) and coordination of motor responses (Medina et al., 2005), enabling complex activities such as speech articulation, musical execution, or sports. Two different hypotheses explaining timing perception have been proposed indicating either a dedicated or an internal model of timing perception (Gibbon, 1977; Hazeltine et al., 1997; Matell & Meck, 2004). A dedicated model assumes one central “internal clock”, composed of a central group of neurons that oscillate with a constant frequency measuring timing for different sensory

channels. Contrary, an internal model for timing perception (Buonomano & Karmarkar, 2002) assumes timing to be an innate property of sensory response patterns. The latter would imply that accurate estimation of timing is emerging independently for each sensory channel and could be represented in an efficient encoding of sensory inputs, thus leading to neurons in sensory channels with response properties tuned to timing. Supporting this hypothesis, investigation on neural responses in primates, carrying out timing-sensitive motor tasks, revealed that responses with both monotonic and tuned characteristics to event duration and period arise in the medial premotor cortex (Merchant et al., 2013). In humans, recent fMRI studies showed that responses in visual cortex areas can be described by monotonic, sub-additive functions of event duration and frequency (Zhou et al., 2018), likely to arise from a summation of transient and sustained neural responses (Stigliani et al., 2017). These monotonous responses are – as the retinotopic organization of the visual cortex implies – presumably to be bound to retinotopic location but tend to gradually change towards tuned, retinotopically invariant responses beginning in area MT/V5 (Hendrikx et al., 2022). A network in subsequent cortical areas has been identified to induce tuned responses to event duration and frequency (Harvey et al., 2020). These areas are widely spread across visual-, multisensory- and action planning areas, internally topographically organized, and largely overlap with cortical areas that induce tuned responses to other quantities such as visual numerosity and visual object size (Harvey et al., 2015, 2020; Harvey & Dumoulin, 2017). Visual numerosity tuned responses in higher cortical areas are invariant to retinotopic position, therefore likely to contribute to an abstract neural representation of numerosity (Harvey et al., 2015). Considering the large overlap of these areas it is hypothesized that tuned responses to visual event timing might similarly contribute to a representation of timing that is invariant to retinotopic position.

Neural response modeling for visually driven quantities has been mostly focused on timing independent cognitive functions. Considering the apparent interleaved relationship between timing and numerosity, we want to highlight several models and techniques regarding numerosity representation. Early research showed that numerosity selective, tuned responses for both symbolic and non-symbolic stimuli emerge simultaneously in an artificial neural network trained to discriminate numerosity (Verguts & Fias, 2004). However, inputs to this network were simplified and the modeling process relied on biologically implausible supervised backpropagation learning rules (Cox & Dean, 2014; Zorzi et al., 2013). Recently, Deep Convolutional Neural Nets (DCNN) have been widely and successfully applied for object recognition from natural images (He et al., 2016; Krizhevsky et al., 2012) and have shown – even when trained to exclusively optimize object recognition performance – to be plausible models for the human visual stream (Yamins et al., 2014). Analogous to the human visual hierarchy, the size of receptive fields in DCNNs tends to increase in higher layers, separating responses to specific objects and features from their spatial position. This formation of abstract, spatially invariant feature maps is supported by lateral inhibition of neurons in the human visual cortex and equivalently local normalization operations of spatial filters in DCNNs (Krizhevsky et al., 2012). Furthermore, global normalization like Batch Normalization has been widely applied to weigh a unit's activation depending on its activation to other elements in a training batch (Ioffe & Szegedy, 2015). Crucially, assuming that activation in a single batch of training data is an approximation for the entire population. Considering numerosity, state-of-the-art DCNNs show an emergence of an innate representation of numerosity without being explicitly trained to discriminate numerosity, peculiarly without being trained at all (Kim et al., 2021).

Unsupervised learning in hierarchical generative models (HGM) (Hinton et al., 2006) seems to be a biologically plausible alternative opposing supervised learning (Zorzi et al.,

2013). HGMs estimate an efficient feature representation according to the probability distribution of presented data without any supervised feedback (Zorzi et al., 2013). This type of model is in line with neurobiological theories implying the mixing of bottom-up and top-down interactions in the brain (Hinton & Ghahramani, 1997). Attempts to model numerosity representation with HGMs revealed that monotonic (Stoianov & Zorzi, 2012) and tuned (Zorzi & Testolin, 2018) responses to numerosity emerge spontaneously in HGMs as an innate property of presented stimuli and without supervised feedback on a specific task. Consecutive modeling suggests numerosity being a salient, emergent property of the visual environment (Testolin et al., 2020), or a feature that closely follows the brain's spatial frequency-based representation of images (Paul et al., 2022).

Modeling approaches investigating neural responses to sensory event timing have shown that an efficient encoding of timing in sensory stimuli can emerge in a biologically plausible fashion by adding paired-pulse facilitation and inhibitory connections between layers of artificial units (Buonomano & Merzenich, 1995). Modeling of top-down time perception recently incorporated feature representations of a DCNN as "sensory" input for cognitively higher reasoning mechanisms enabling the reproduction of biases in human timing perception (Fountas et al., 2022).

Independent of modeling biological responses to event timing, Recurrent Neural Networks (RNN) emerged as a framework to efficiently approximate timing-dependent functions (Rumelhart et al., 1985). Recently, RNNs have been widely applied in complex, timing-dependent tasks such as speech recognition (Graves et al., 2013), machine translation (Cho et al., 2014), or image caption generation (Vinyals et al., 2015). RNNs form a special case of HGM by estimating a feature representation of sequential data and achieve integration of temporal dependencies by repeatedly combining hidden states from previous time-steps with novel input. Comparable to HGMs, RNNs do not utilize explicit feedback but rather

estimate a feature representation that is most beneficial for the realization of inherent learning target. As these hidden representations get passed in a feedforward fashion to higher layers, the content of these representations becomes more and more abstract. Consequently, – in the context of timing – hidden representations in higher layers should be able to contain more complex timing representations and increasingly encode timings with a larger temporal dependency. However, classical RNN architectures were largely limited by the vanishing- or exploding gradient problem which describes exponential decreasing or increasing of backpropagation gradient when dealing with long sequences of data or many hierarchical layers. This often leads to difficulties in learning long-term temporal dependencies (Hochreiter, 1998). The vanishing gradient problem can be circumvented by avoiding hyperbolic activation functions in favor of activation functions with a constant gradient such as the Rectified Linear Unit activation function (ReLu). Nevertheless, the exploding gradient problem might still interfere with encoding capabilities of RNNs, as ReLu is unbounded in the positive domain and function values possibly grow rapidly. Common approaches trying to solve both the vanishing and exploding gradient problem introduced either restriction in connectivity of recurrent connections (Li et al., 2018) or control of information flow within networks through gating mechanisms (Hochreiter & Schmidhuber, 1997). The most notorious example of the latter is known as a Long Short-Term Memory network (LSTM) (Hochreiter & Schmidhuber, 1997) and has been successful at capturing large temporal dependencies by containing relevant information about previous time-steps in a cell state. Considering the biological plausibility of LSTMs, however, different gating mechanisms allow dynamic capturing and storing of any complex temporal information, even when past activity does not follow a monotonic structure. Contrary, neural responses of units in early stages of the visual stream seem to be limited to capturing simpler monotonic patterns, whereas only neural units in later stages inhabit the capability to capture tuned patterns (Hendrikx et al., 2022).

Analogous to DCNNs, RNNs often utilize normalization operations, however, application of Batch Normalization has often been criticized for its computational unfeasibility with growing sequence length and its heavy dependency on the batch size hyperparameter (Ba et al., 2016). As an alternative, Layer Normalization has been proposed (Ba et al., 2016). In Layer Normalization, a hidden unit's value is not normalized depending on values in other training examples, but rather dependent on activity of other units within a layer during presentation of individual stimuli. As a model of biological temporal processing, this approach is supported by fMRI observations in humans: monotonic responses to frequent event timings have been shown to inhabit a stronger sub-additive suppression than responses to infrequent event timing (Zhou et al., 2018). This fact lets us hypothesize that neurons' suppression is dependent on carried-over activation from previous time-steps. Since such activation is primarily dependent on induvial stimulus's structure, a Layer Normalization approach only considering induvial stimulus's activation seems most legitimate.

In the current study, we aim to combine modeling techniques from the numerosity domain with established neural network architectures for temporal-dependent tasks to create a hierarchical recurrent generative model for human sensory processing of visual event timing. Such a model enables exploration of the computational foundations of timing prediction and overcomes limitations concerning the determination of causal relationships between stimuli and responses in fMRI studies. Furthermore, modeling enables maximal control of sensory input, neural connectivity, neural activation, and yields advanced insights into neural dynamics with instantaneous temporal resolution. Within the scope of this model, we want to investigate whether transient and sustained components of visual inputs allow a straightforward computation of monotonic event duration and period, whether such monotonic responses are obligatory for encoding of timing, whether monotonic responses of event duration and period allow a straightforward computation of timing-tuned responses,

and *how* monotonic timing responses turn into timing-tuned responses in such a scenario. Crucially, the model utilizes unsupervised, biologically plausible learning rules to encode timing information from sequential stimuli, necessarily not exploiting any explicit, supervised feedback. Moreover, the model's hierarchically organized architecture conceptually resembles the human visual cortex and successive areas.

Emerging response dynamics in a biologically plausible model for visual event timing allow a comparison to response dynamics in the human visual cortex. More specifically, we want to draw a comparison with monotonic timing response development in early visual areas (Hendrikx et al., 2022; Stigliani et al., 2017; Zhou et al., 2018) and timing-tuned responses in later visual areas (Harvey et al., 2020; Hendrikx et al., 2022).

Methods

Stimuli

The dataset used to train the GNNs consisted of stimuli used in previous fMRI studies (Harvey et al., 2020). Here, a stimulus consisted of a sequence of images with a width and height of 75 pixels. An image contained either of two contents: a centered black dot on a white canvas, or a blank white canvas. In the experiments, the framerate was set to 20 frames per second and the maximal stimulus length was set to 42 images. Therefore, an image was displayed for a duration of 50 ms and a stimulus was displayed for a maximal duration of 42 * 50 ms = 2100 ms. Stimulus timings were defined by a combination of event duration and event period, in which the number of consecutive dot images gives event duration and the number of images between two dot appearances gives event period. An example: a stimulus of repetitive sequences consisting of a dot image followed by two no-dot images corresponds to a stimulus timing with duration = 1 * 50 ms = 50 ms and period = 2 * 50 ms = 100 ms (Figure 1). One of these repetitive sequences corresponded to a single event. Stimuli were intentionally designed to be circular, meaning that the onset and the offset of a stimulus were equal and interchangeable. Thus, a stimulus contained exclusively identical, complete events of specific timing and the length of each stimulus was slightly variable (around 2100 ms) due to differing event lengths. Stimuli with a shorter event length contained a larger number of events, and therefore more information about respective event timing. In the fMRI study, this effect was balanced by accounting for the event frequency in the amplitude prediction (Harvey et al., 2020). Combinations of event duration and period were restricted to a total of 80 combinations in four experimental conditions. Stimuli with similar timings were presented consecutively in a gradual changing manner (for example from low event duration to high

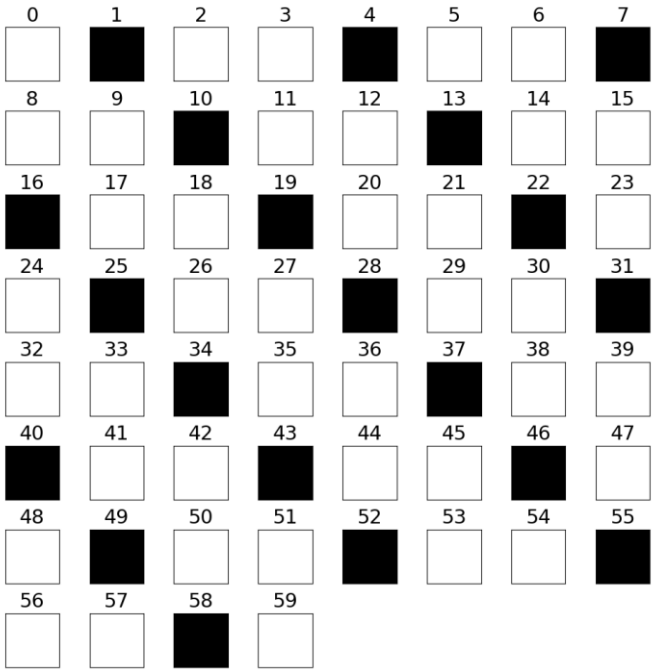


Figure 1. Stimulus with a duration of 50 ms and a period of 150 ms with a phase offset of 1.

event duration), likely introducing carry-over effects of neuronal activation between similar timings.

In the current study, existing stimulus space (Harvey et al., 2020) was adjusted to suit the context of a trainable GNN model. Stimuli for every possible combination of event duration and event period were generated to enhance the resolution of the stimulus space. Here, event duration must be smaller than event period as other configurations would result in an overlap of events leading to multiple dots in one image. All applicable combinations of event duration and event period corresponded to 210 stimulus timings. Stimuli with more events contained more information about respective timing and were inherently easier to learn. To eradicate this internal bias and to add variability to the data, the circular nature of the stimuli was exploited by shifting the start of the stimulus by all possible offsets, resulting in stimuli with the same event duration and period but in different phases of the stimulus. For a specific timing, the number of possible phases equals the event length, because shifting a cyclic stimulus further than the event period would result in a duplicate stimulus. Therefore, in a stimulus with an event duration of 50 ms and an event period of 100 ms, the single dot

image could be in three different positions: in the beginning, after 50 ms, or after 100 ms. All possible phases of stimulus timings were included in the dataset. Construction of the dataset following this logic has the benefit that each stimulus timing had a constant number of events when summed over all different variations of phases. To guarantee efficient training by utilizing a batched training procedure, the stimuli required a fixed length. Therefore, the stimulus length was increased to a fixed length of 60 images, corresponding to a 3000 ms time interval. In case numerous complete events did not exactly 60 frames of a stimulus, several images following the stimulus nature were appended, which introduced a slight imbalance. This effect was counterbalanced by excluding appended images in the evaluation- as well as in the response recording methodology.

Even though images with a width and height of 75 pixels were used for human participants in fMRI studies (Harvey et al., 2020), the reduction of the models' computational complexity, as well as training times, was essential. Therefore, the image size was reduced to a single pixel which was either black or white. This approach was only applicable because pixels in images were perfectly correlated and thus contained equivalent informational content relevant to timing.

Generative Neural Network

The GNN architecture was composed of several elements with a recurrent cell being the major building block. The following section elaborates on two different recurrent cells used in the GNN architecture.

RNN Cell

The RNN cell – as the name implies – contains a recurrent, as well as an outgoing connection (Figure 2). The outgoing connection feeds into succeeding layer's RNN cell and is part of the higher RNN cell's inputs. The recurrent connection feeds into the same RNN cell and contains the cell's output from the previous time-step. Therefore, at a given time-step

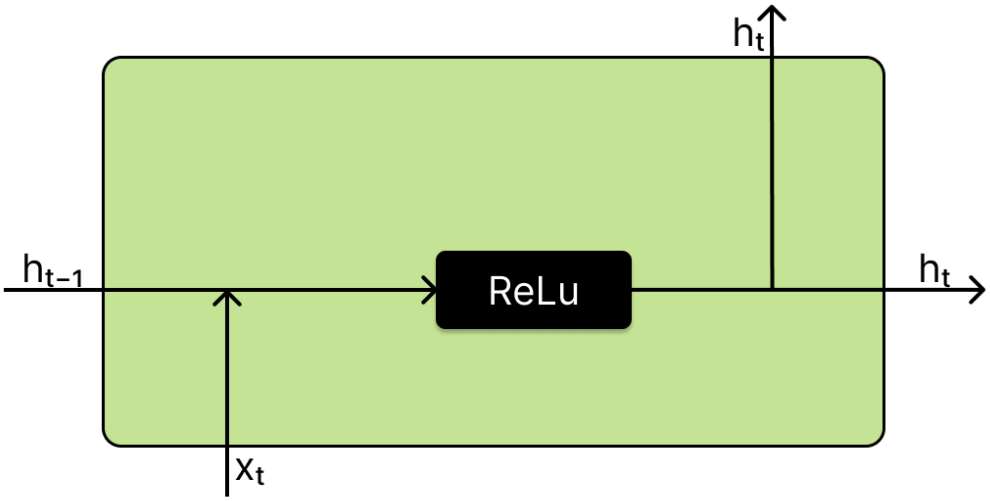


Figure 2. RNN cell.

t, each cell's inputs are composed of its output at time-step t-1 as well as the output of the previous layer's RNN cell at time-step t. Internally, each RNN cell has two sets of dense neural weights for the input- and the recurrent component. Both inputs get multiplied with respective weights (matrix multiplication) and both feature vectors get summed (pointwise summation). Resulting feature vectors are fed into an activation function, producing the hidden state at time-step t (Equation 1). This hidden state can be interpreted as an abstract representation of the input at time-step t combined with its temporal context based on previous hidden states (recurrent connection from t-1).

Equation 1

$$h_t = ReLu(W_{ih}x_t + b_{ih} + W_{hh}h_{(t-1)} + b_{hh})$$

Where h_t is the hidden state at time-step t, x_t is the input at time-step t, and h_{t-1} is the hidden state at time-step t-1. W_{ih}, b_{ih}, W_{hh}, and b_{hh} are weight matrices and bias vectors for input and hidden state respectively.

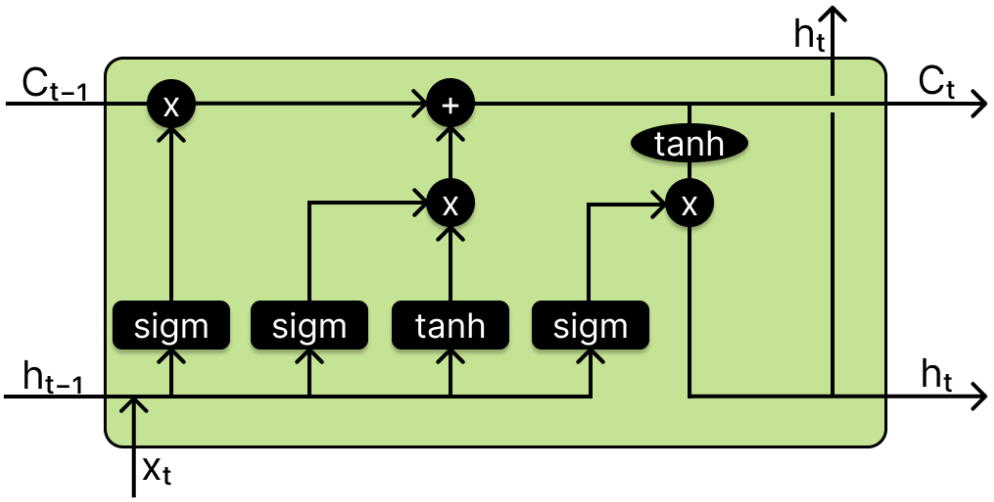


Figure 3. LSTM cell.

LSTM Cell

The LSTM cell is a more specialized version of an RNN cell and archives increased performance in many tasks due to its inherent gating mechanisms (Figure 3). The core structure of the LSTM cell is equal to the RNN's (recurrent connections, outgoing connections); however, the LSTM cell not only contains a set of weights for the input- and the recurrent component, but likewise additional sets of weights called forget gate, output gate, and cell gate (Equation 2). These gates allow the cell to extract additional information from the inputs and independently adjust gates following gradient descent. The forget gate, for example, allows certain properties of the inputs to be dynamically weighted at any given time-step t to improve prediction performance. Furthermore, the LSTM cell extends the RNN cell by having the ability to pass information about previous time-steps not only via the hidden state but also via a cell state. Importantly, information in the cell state can be contained indefinitely through time and therefore memory capacities can be achieved in a biologically implausible manner. Even though LSTM cells do not seem to be biologically plausible, GNN models with LSTM cells were used as a performance benchmark.

Equation 2

$$i_t = \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi})$$

$$f_t = \sigma\big(W_{if}x_t + b_{if} + W_{hf}h_{t-1} + b_{hf}\big)$$

$$g_t = tanh\big(W_{ig}x_t + b_{ig} + W_{hg}h_{t-1} + b_{hg}\big)$$

$$o_t = \sigma(W_{io}x_t + b_{io} + W_{ho}h_{t-1} + b_{ho})$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t$$

$$h_t = o_t \odot \tanh(c_t)$$

Where h_t and c_t is the hidden- and cell state at time-step t, x_t is the input at time-step t, h_{t-1} is the hidden state at time-step t-1, and i_t, f_t, g_t, o_t are the input, forget, cell, and output gates respectively. σ is the sigmoid function and ⊙ is the Hadamard product. W_{i*}, b_{i*}, W_{h*}, and b_{h*} are weight matrices and bias vectors for respective gates, inputs, and hidden states.

Architecture

Following the idea to induce biological plausibility by resembling the structure of the human visual cortex, GNNs with one to six hidden layers and a ReLu activation function were implemented. Networks were composed of multiple layers following the structure in Figure 4 (left): a recurrent cell (i.e., RNN-cell or LSTM-cell) followed by a normalization operator, a ReLu activation function, and a dropout operator. As inputs for a single time-step were naturally chosen to be a single image, the inherent task of the GNN was to predict the *next* image in the sequence. Thus, the last layer of the model feeds into a linear layer, constructing an image from the given hidden state (Figure 4, right). This prediction was evaluated by a criterion and resulting error was fed back into the network, adjusting weights following the gradient descent scheme. Importantly, feedback was *not* explicitly supervised as the obtained error was not determined by an external label but was rather inherent to the

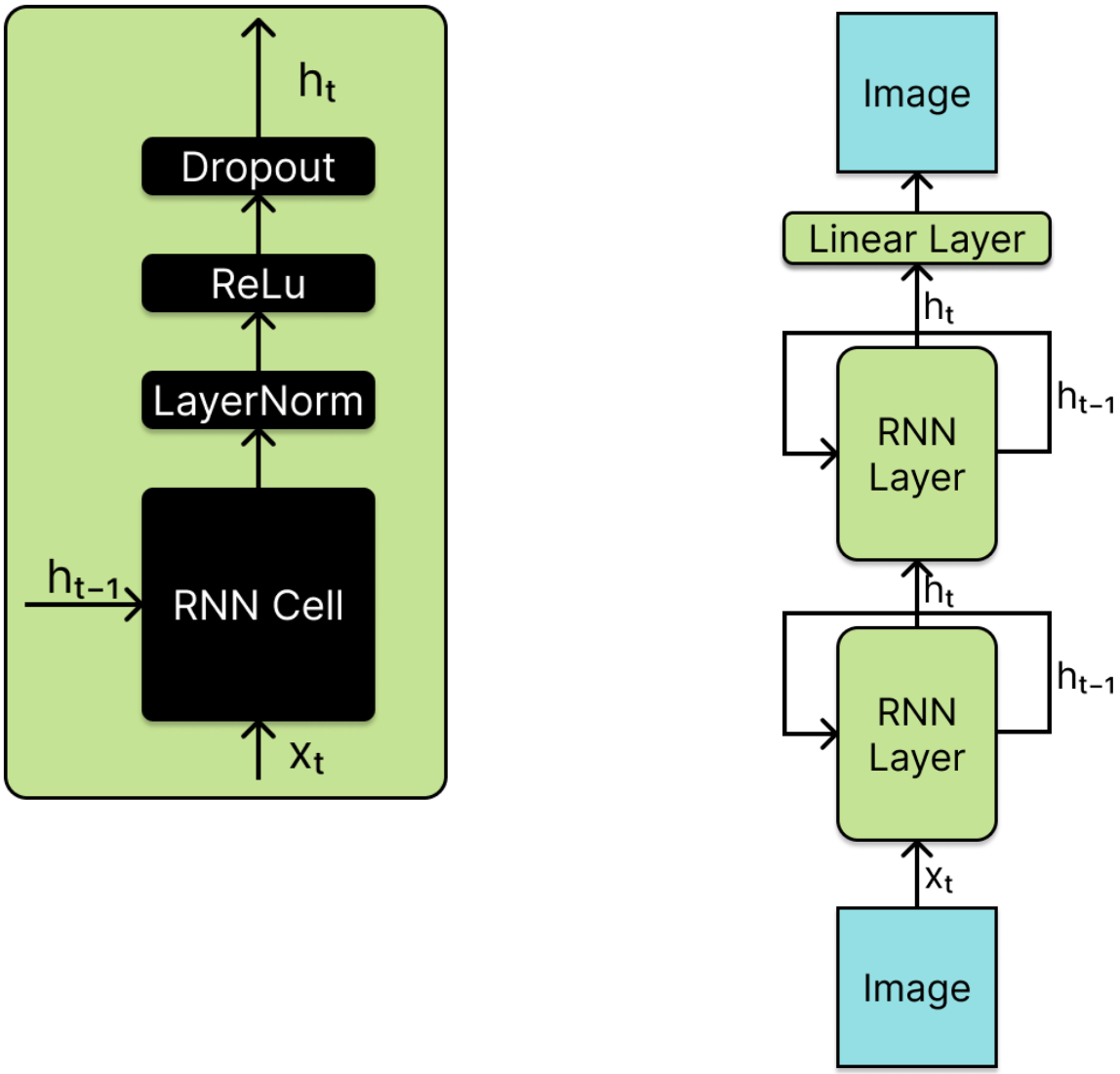


Figure 4. Structure of a single GNN layer (left) and architecture of a two-layer GNN (right).

data (self-supervised). The GNN's predictions were not limited to integer values in which zero would correspond to a white pixel and one would correspond to a black pixel. Instead, predicted floating point values corresponded to shades of grey.

The normalization operation was implemented before the application of the activation function following common practice from DCNNs with a ReLu activation function (Ioffe & Szegedy, 2015). Further, a dropout operator was used in intermediate layers to induce variability in learning of feature representations, often leading to better generalization performance (Srivastava et al., 2014). Consequently, networks with a single hidden layer did not utilize dropout.

In total, six Elman RNNs with one to six layers and two LSTM RNNs with one or two layers were implemented. LSTM RNNs were limited to two layers as the initial structure of the LSTM is designed to capture long-term relationships without many layers. Adding layers would introduce a vanishing gradient due to hyperbolic activation functions at different gates. Moreover, hidden weights of the networks were initialized using He initialization (He et al., 2015). This initialization method ensured that the sum of weights is equal even though the number of total weights differs between architectures enabling a fair comparison of the training process. All models were implemented in Python 3.7 using the PyTorch library.

Hyperparameters

The GNNs used the same set of hyperparameters to ensure comparability (Table 1). Here, a mean squared error (MSE) loss criterion, an Adam optimizer, a dropout probability of 0.2, as well as a batch size of 30 was used. MSE loss for an image was calculated by averaging the pixel-wise loss of a predicted image, and MSE loss for a stimulus was calculated by averaging MSE loss for all predicted images in a stimulus. The Adam optimizer was initialized with PyTorchs default values (learning rate = 0.001, $beta_1$ = 0.9, $beta_2$ = 0.999, weight decay = 0). Dropout probability was set to a moderate value of 0.2.

Table 1. Fixed Hyperparameters of the GNN architecture.

Loss function	MSE
Optimizer	Adam
Dropout probability	0.2
Batch size	30
Hidden neurons	64

To ensure the hyperparameters were generally suitable for the given stimulus space, several models were trained on a so-called “perfect information” setup. Here, a separate model for each of the stimulus timings was generated and each of the models was trained independently on a single stimulus timing. Therefore, a total of 210 stimulus timings, as well

as 210 models were used. Assuming suitable hyperparameters, each of the trained models was expected to reach a loss value close to zero. This would mean that the hyperparameters were feasible to encode the temporal structure of every stimulus timing in independent models. Consequently, a single model with identical hyperparameters could encode the temporal structure of multiple stimulus timings. The number of hidden neurons was tuned by training the models with 256 hidden neurons and halving the size of hidden layers in case all independent models were able to encode respective stimulus timing under perfect information. This ensured that the minimal model size was used to prevent unnecessarily long training times. Finally, the lowest number of hidden neurons, successfully encoding every stimulus timing, was 64.

Training

The GNNs were trained for 1500 epochs using a randomized train-test split of 0.8. A commonly used n-fold cross-validation approach for training and validation was intentionally avoided, as the evaluation performance of the GNNs was not the focus of this study. Contrary to the original fMRI experiments (Harvey et al., 2020), the stimuli were presented in random order and independent of experimental conditions. As the random presentation order of the stimuli implies, neural carry-over effects were not considered to be relevant for GNN unit's tuning, and therefore, carry-over was eliminated by resetting the GNNs hidden state before presentation of a new stimulus batch. To optimize the training procedure, the training was carried out on an Nvidia RTX 3070Ti GPU utilizing PyTorchs CUDA GPU support.

Evaluation

The GNNs were tested on the ability to predict successive images in a presented sequence of images from the holdout evaluation dataset. Contrary to the training procedure, a per-event loss was used as a performance measure. Per-event loss was obtained by dividing the summed stimulus loss by the number of stimulus events. The reason for this being that the

per-movie loss measure is biased to benefit stimuli with fewer total events. Considering that a single event contains exactly one stimulus onset and one stimulus offset, thus a single image in which a dot is appearing and disappearing, fewer events correspond to fewer transitions from no-dot images to dot images. A per-movie loss measure does not account for such a discrepancy in stimulus difficulty. Predicting any timing before the presentation of an entire event is impossible. Therefore, all images beginning from the start of a stimulus until the presentation of an entire event were excluded from performance measurements. Appended images at the end of a stimulus to match a fixed image length were equally excluded.

Timing Response Recording

GNN unit's responses were recorded on the entire training dataset to obtain the most reliable responses and evaluate the units' responses to all event timings. During this process, dropout was disabled to prevent noisy, not reproducible results. Per-event, as well as per-movie responses to every stimulus timing, were recorded. Per-movie responses were expected to produce larger response amplitudes for stimulus timings which contain a larger number of events. Therefore, the per-event responses were of major interest. To obtain a comparable score to neural response model's predictions, responses for movies with the same stimulus timing were averaged. Following this methodology, a single response amplitude of each unit for every stimulus timing was recorded.

Neural Response Models

As the main interest of this study was to investigate GNN unit's response properties to event timing, encodings of event timing in hidden units in different layers of the trained GNNs were investigated. For this purpose, responses of GNN units to different stimuli were recorded and a parameterized neural response model was fit (Harvey et al., 2020). As the nature of the responses was expected to be either monotonic or tuned to event timing, two variants of the neural response model were used: one variant capturing monotonic responses,

and one variant capturing tuned responses. Both models assume a linear relationship between the response amplitude and the actual responses, therefore incorporating linear scaling parameters.

Monotonic Response Model

In the monotonic model, event duration and event period are assumed to influence the response amplitude independently and linearly. Consequently, the model incorporates the two predictors duration and period, each with a compressive exponent to account for sub-additive increases in response amplitude. Accounting for a linear relationship, each predictor is scaled by an independent scaling parameter, which can be combined to form a single "AmplitudeRatio" parameter (Equation 3).

Equation 3

$$Amplitude \propto Duration^{expDur} * AmplitudeRatio + \frac{Period}{Period^{expPer}}$$

Where expDur and expPer are the respective compressive exponents for duration and period and AmplitudeRatio is the linear scaling parameter.

Tuned Response Model

In the tuned model the response amplitude is assumed to have an anisotropic two-dimensional Gaussian tuning to event duration and period, together with a compressive increase in response amplitude with event rate (Harvey et al., 2020) (Equation 4). The tuned response model can be either tuned to event duration, event period, or both.

Equation 4

$$X = \left(Duration - Duration_{pref}\right) * \cos(\theta) - \left(Period - Period_{pref}\right) * \sin(\theta)$$

$$Y = \left(Duration - Duration_{pref}\right) * \sin(\theta) - \left(Period - Period_{pref}\right) * \cos(\theta)$$

$$Amplitude \propto e^{-0.5*\left(\left(\frac{Y}{\sigma_{maj}}\right)^2+\left(\frac{X}{\sigma_{min}}\right)^2\right)} * \frac{Period}{Period^{expPer}}$$

Where Θ is the preferred angulation of the Gaussian functions major axis, $Duration_{pref}$ is the preferred duration, $Period_{pref}$ is the preferred period, σ_{maj} is the standard deviation along the major axis, σ_{min} is the standard deviation along the minor axis, and expPer is the compressive exponent for period.

Fitting Procedure

GNN unit's responses were fit by obtaining amplitude predictions from given parametrized models and subsequently finding the optimal linear scaling parameter maximizing the explained variance (Least Squares Problem). To match the degrees of freedom of both models (two for the monotonic model and one for the tuned model), a two-fold cross-validation approach was utilized. Here, both neural response models were fit on opposing halves of the data, the best parameters and scaling factors were obtained, and predicted response amplitudes for every timing were computed. Subsequently, explained variance for both models was measured on the complement half of the data. Importantly, both halves of the data were balanced, thus containing an equal number of movies of the same timing. Finally, the best fitting parameters of both models for each of the GNN units were saved. Comparing monotonic- and tuned response models was difficult as tuned response models allow a close approximation of monotonic response models due to their increased complexity. Such an approximation was explicit, whenever the peak of a tuned function was on the edge of investigated timing spectrum. In such a scenario, the observable part of the tuned function looks exactly like a monotonic function (Figure 5). To prevent this effect, preferred duration and period on the edges of the timing spectrum (50 ms duration and 1000 ms period) were excluded. Arguably, this approach still allows tuned functions with a tuning

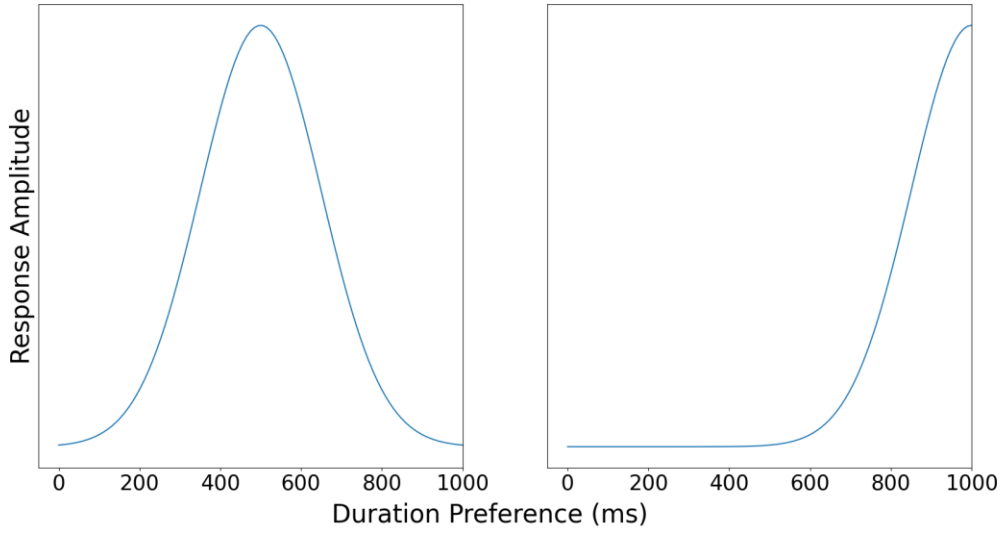


Figure 5. Tuned function with a preferred duration at 500 ms (left) and the same tuned function with a preferred duration at 1000ms (right)

close to the edges of the spectrum, but the monotonic function was assumed to fit the responses better in such a scenario. Models were fit on both the per-event responses and the per-movie responses utilizing PyTorch CUDA GPU support and an Nvidia RTX 3070Ti GPU.

Model Comparisons

After obtaining the best fitting response models for each GNN unit, structural differences between both response models were investigated. However, collected response data, as well as model fits, had to be preprocessed for this purpose. Manually investigating average responses of GNN units revealed that many units were not responsive to any of the presented timings. These units were not relevant for further investigation and were excluded. Likewise, units in which the explained variance of both neural response models was below 0.2 were excluded.

Systematic differences between response model fits could occur on three different levels: within layers of GNNs, within GNNs, and between GNNs. To address all three levels, different test statistics, as well as different measurements, were used. For *within-layer*

comparisons, paired t-tests with monotonic- and tuned fits as the two groups were conducted. For *within GNN* comparisons, the difference between tuned and monotonic fits was calculated, and an n-way ANOVA analysis followed by a Tukey pairwise comparison test was conducted. Here, the number of GNN layers determined the degrees of freedom of ANOVA analysis, and each layer was compared to every other layer in Tukey pairwise comparisons. *For between GNN* comparisons, only the fit of the tuned response model was used as a measurement, all GNN units were pooled, and an n-way ANOVA analysis followed by a Tukey pairwise comparison test was conducted. Here, the number of different GNNs determined the degrees of freedom of ANOVA analysis, and each GNN was compared to every other GNN in Tukey pairwise comparisons. Systematic differences in preferred timings of tuned response models *within GNNs* were tested by measuring the Euclidean distance of each preferred timing to the center of presented timing range (duration = period = 500 ms) and conducting an n-way ANOVA analysis followed by Tukey pairwise comparison test. Again, the number of GNN layers determined the degrees of freedom of ANOVA analysis, and every layer was compared to every other layer in Tukey pairwise comparison test. The significance level for all statistical tests was set to 0.05.

Results

GNN Prediction Performance

Elman RNN

Every trained Elman RNN with one to six hidden layers converged towards the minimal loss after 1500 epochs. Generally, models with more layers except for the six-layer network converged slightly faster than models with fewer layers (Figure 6). Train losses of Elman RNNs with two to six layers were minimally lower compared to losses of the one-Layer Elman RNN. Evaluation of trained models revealed that all stimulus timings could be predicted accurately (Figure 7). Figure 8 shows a prediction of a two-layer Elman RNN for a stimulus with an event timing of 100 ms for duration and 800 ms for period. Here, the

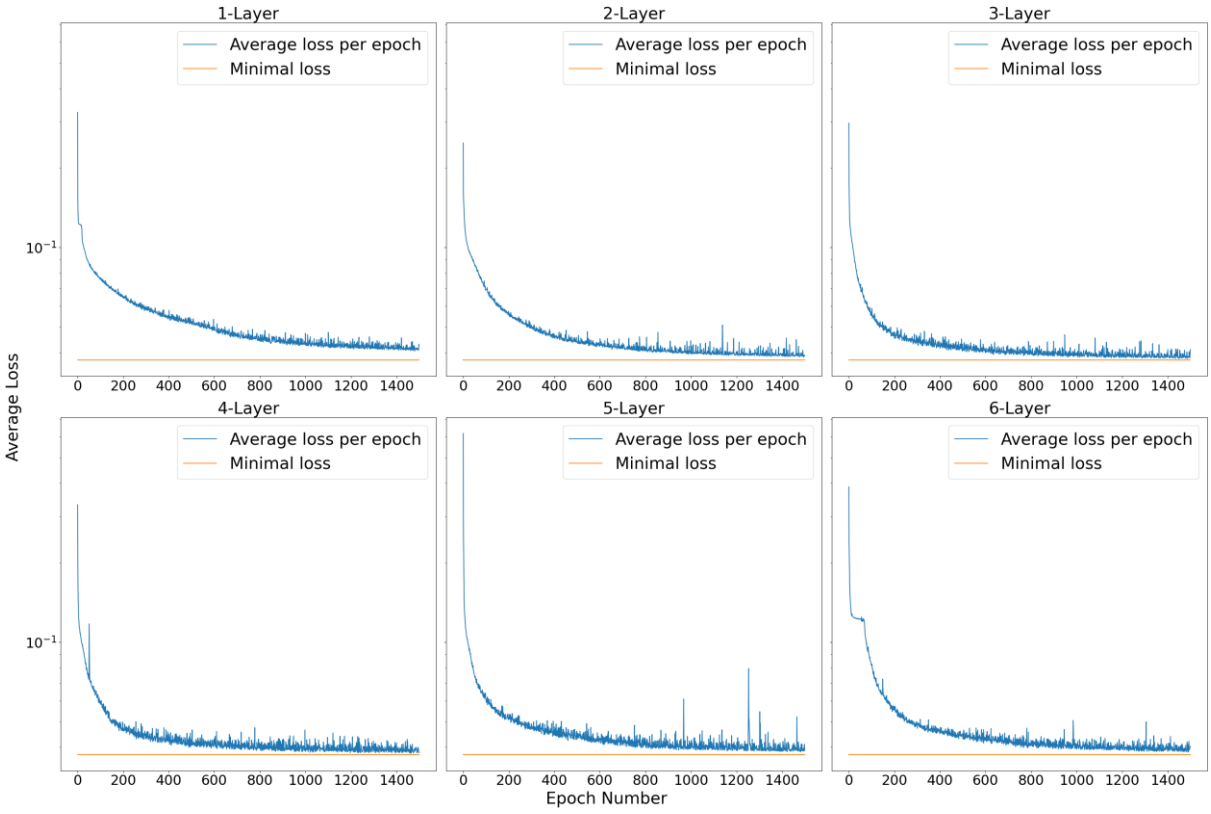


Figure 6. Average training loss per epoch of Elman RNNs with one to six layers.

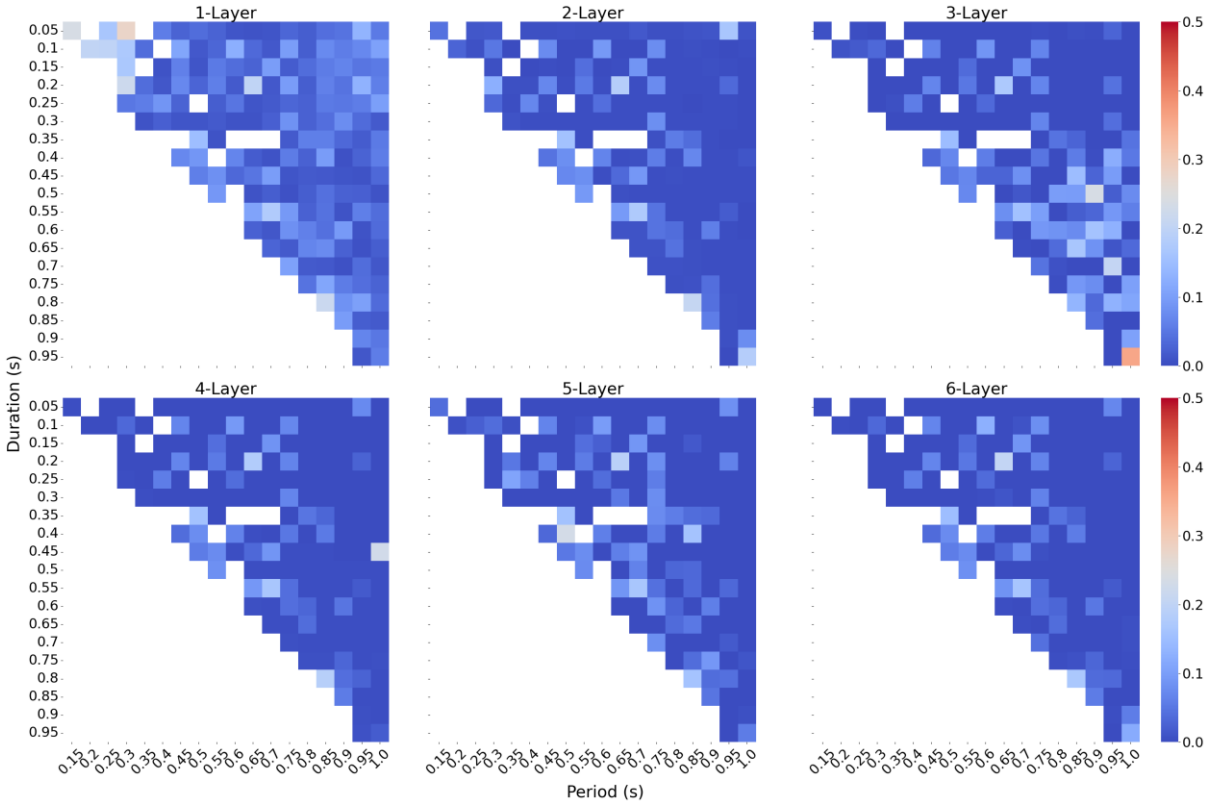


Figure 7. Average evaluation loss of Elman RNNs with one to six layers for different timings.

network produces significant error until a complete event has been presented (frame 0 to 26), subsequent event, however, was predicted precisely (frame 27 to 44).

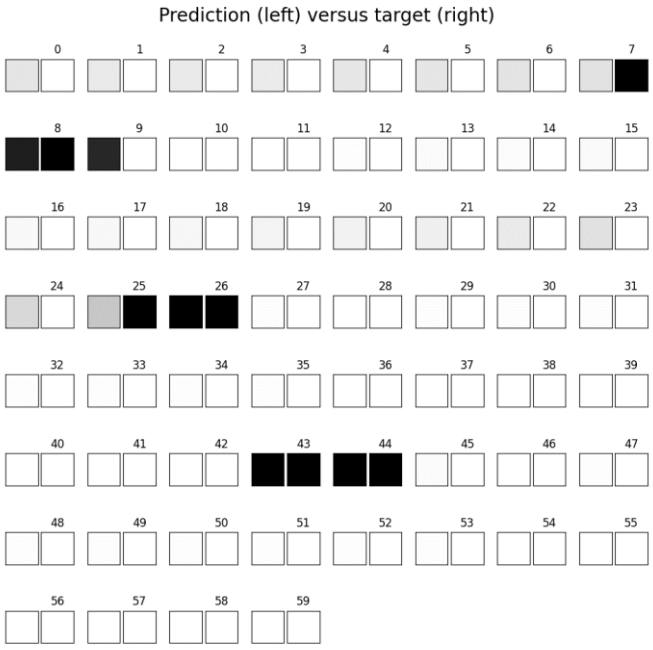


Figure 8. Prediction of a two-layer Elman RNN for a timing with duration of 100 ms and period of 800 ms.

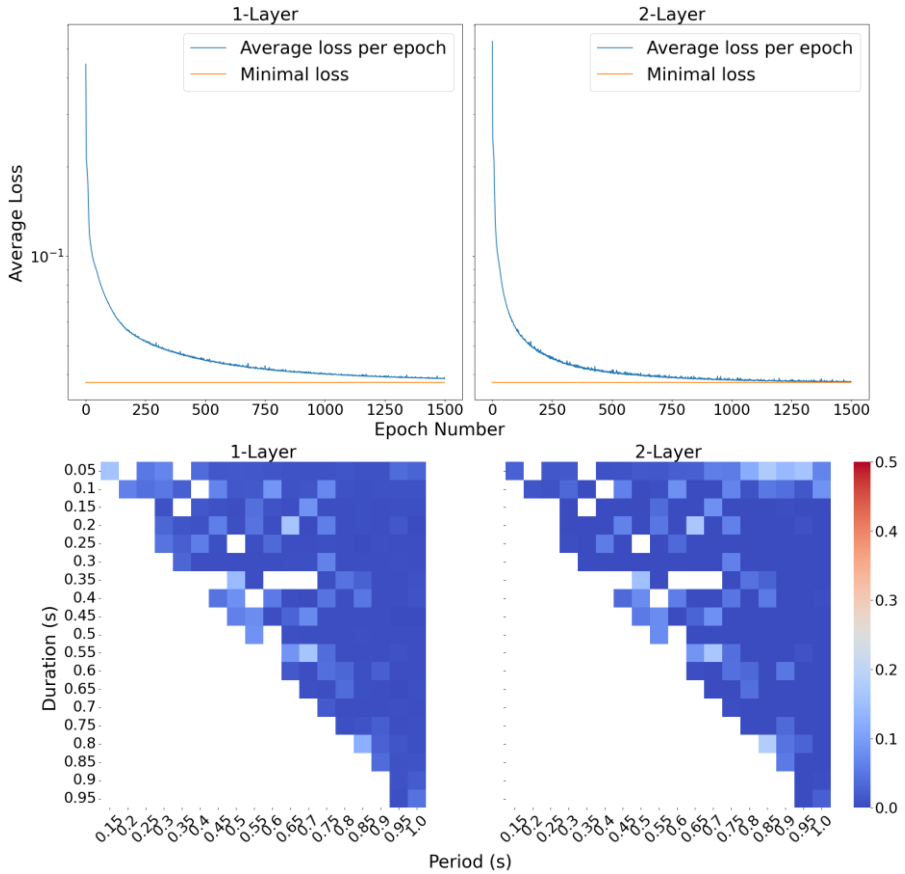


Figure 9. Average training loss per epoch (top) and average evaluation loss (bottom) for different timings of a one-layer LSTM RNN and a two-layer LSTM RNN.

LSTM RNN

As expected, both LSTM RNNs converged towards the minimal loss value quickly. The two-layer LSTM RNN converged slightly faster than the one-layer LSTM RNN (Figure 9, top). Analogous to Elman RNNs, both LSTM RNNs performed well on the evaluation dataset (Figure 9, bottom). Overall performance of LSTMs was only minimally better than Elman RNNs.

Neural Response Model Comparisons

Elman RNN

Figure 10 (left) shows monotonic and tuned fits for a one-layer Elman RNN. Even though many units show a good fit for both monotonic and tuned response models (upper right corner), the tuned response model often fits better than the monotonic model (dots towards the upper left corner). Investigation of preferred timings of the tuned response model

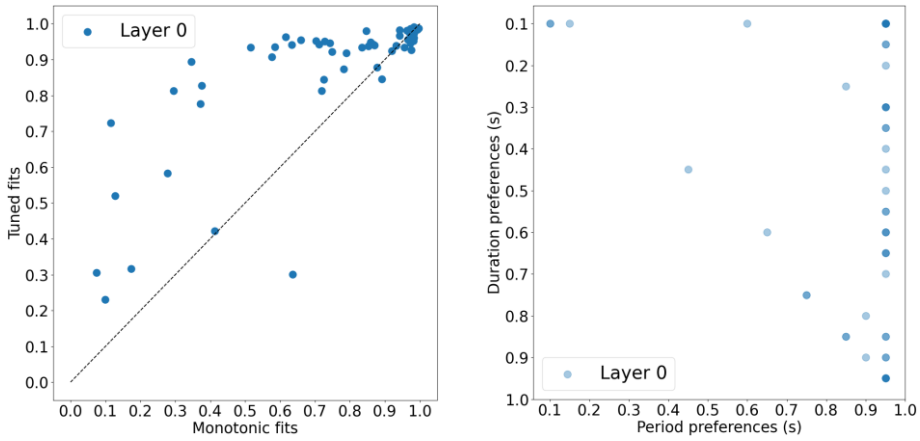


Figure 10. Monotonic and tuned fits (left) and preferred timings of tuned response models where the tuned fits are higher than the monotonic fit (right) of a one-layer Elman RNN.

(Figure 10, right) shows that some of the preferred timings were very close to the corners of investigated spectrum, thus closely approximating a monotonic function. Other preferred timings were located far from the corners and were therefore convincingly tuned. Specifically, a preferred period of 950 ms was found in many units. This indicates that tuning to event period tends to rather approximate a monotonic structure than tuning to event duration. An example of approximation of a monotonic function by the tuned response model can be seen in Figure 11 (top): both monotonic and tuned fits are high (0.97 and 0.95) but the tuned models' prediction strongly mimics a monotonic function. Figure 11 (middle) shows a tuned response to event duration and period with a vertical orientation, and Figure 11 (bottom) shows a tuned response to both event duration and period with an angulation of approximately 45 degrees. Consequently, in both the units with tuned response properties, the fit for the tuned response models is significantly better. Only units with monotonic, as well as two-dimensionally tuned response properties with variable angulation could be found (Figure 12).

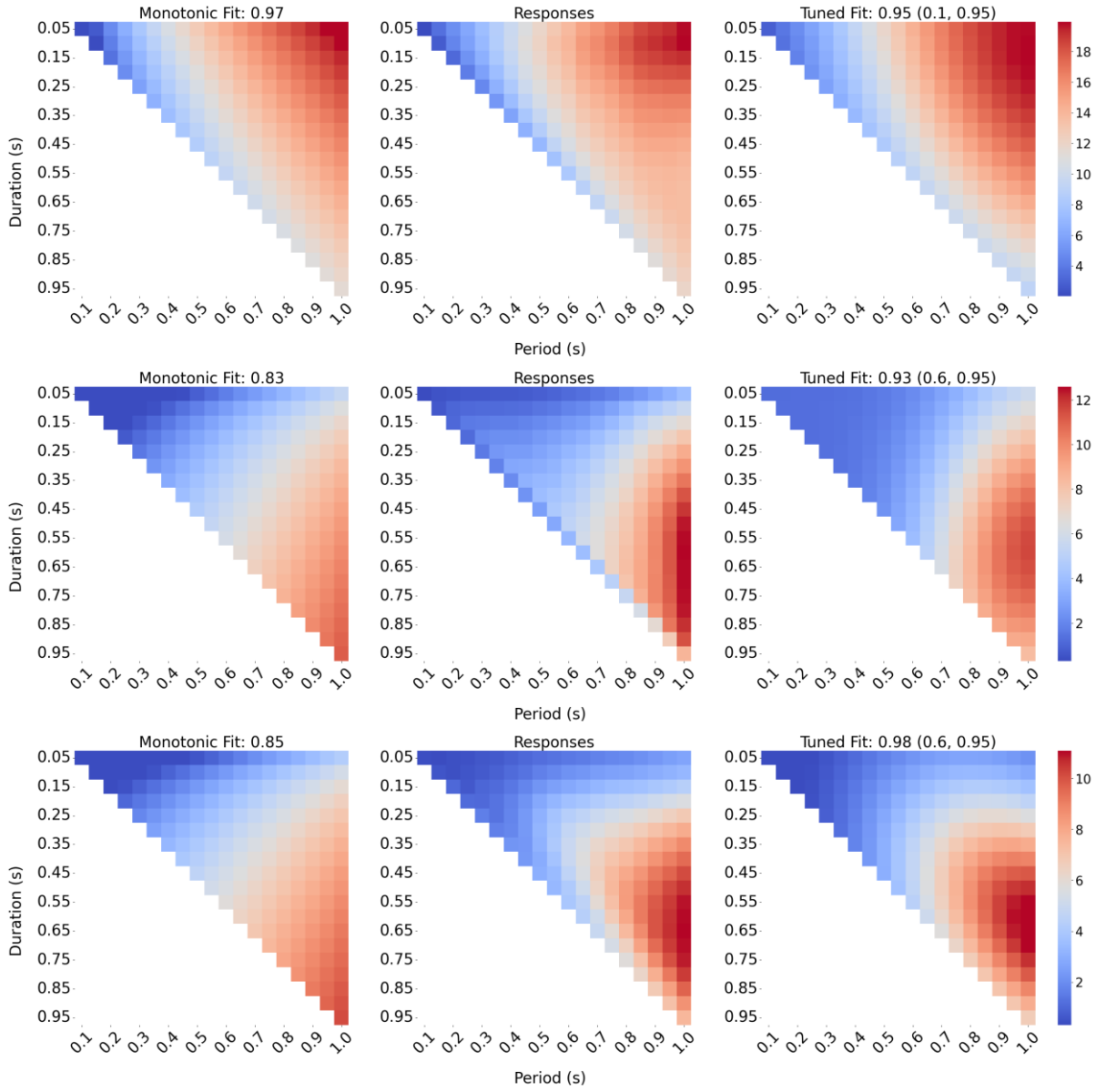


Figure 11. Monotonic fits, responses, and tuned fits for unit 27 (top), unit 59 (middle), and unit 1 (bottom) in a one-layer Elman RNN.

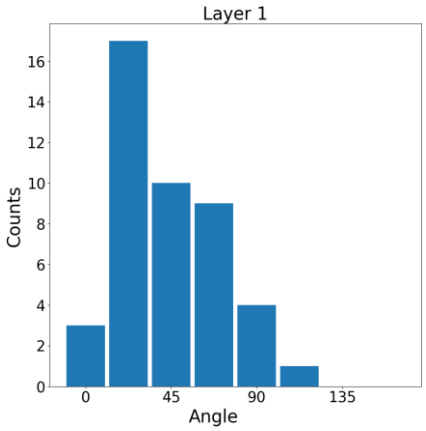


Figure 12. „Angle“ parameter distribution of tuned response models, where the tuned fits are higher than monotonic fits in a one-layer Elman RNN.

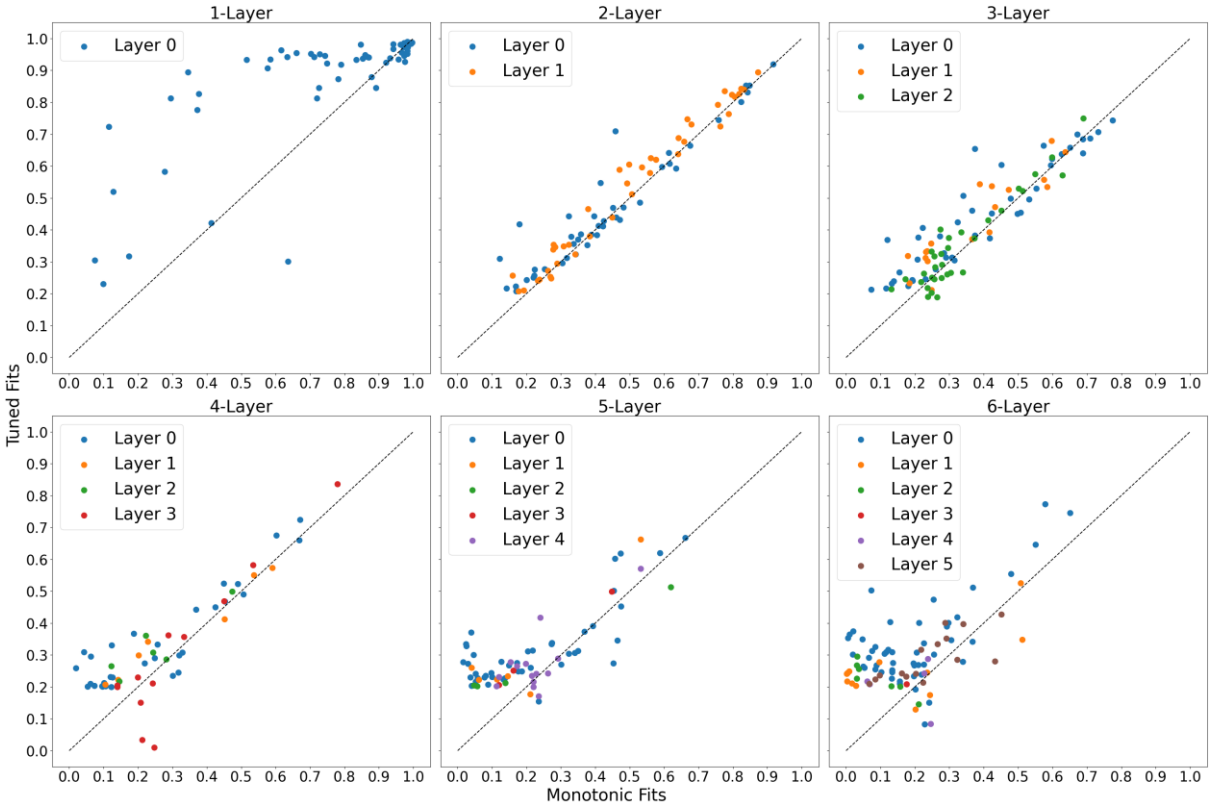


Figure 13. Per-event monotonic and tuned fits for Elman RNNs with one to six layers.

Response properties of units in hierarchical Elman RNNs were different. Figure 13 shows monotonic and tuned response model fits for Elman RNNs from one to six hidden layers. Tuned models' fits were worse in deeper networks compared to a one-layer Elman RNN. To confirm this finding, units from every layer of Elman RNNs were pooled and tuned fits were compared between networks. A six-way ANOVA analysis was significant with $p < 0.001$, $\alpha = 0.05$ and $n_1 = 57$, $n_2 = 84$, $n_3 = 93$, $n_4 = 57$, $n_5 = 71$, $n_6 = 90$. Pairwise Tukey comparison test results can be found in Table 2. Every pairwise comparison except three- versus two-layer, four- versus five-layer, and four- versus six-layer Elman RNNs was significant with $\alpha = 0.05$. Here, especially non-hierarchical Elman RNNs show a large mean difference compared to hierarchical Elman RNNs. Therefore, it can be confirmed that tuning to timing is the strongest in units of the one-layer Elman RNN and gradually decreases with

increasing number of layers. Starting from Elman RNNs with more than three layers, tuning to timing was indifferent.

Table 2. Pairwise Tukey comparison test results of tuned response model fits between pooled units of different Elman RNNs.

Network Type	Network Type	Mean Difference	P	95% Confidence Interval	
				Lower Bound	Upper Bound
1	2	-0.364	< 0.001*	-0.4444	-0.2836
1	3	-0.4544	< 0.001*	-0.5332	-0.3756
1	4	-0.5296	< 0.001*	-0.6173	-0.4418
1	5	-0.5571	< 0.001*	-0.6404	-0.4738
1	6	-0.5623	< 0.001*	-0.6416	-0.483
2	3	-0.0904	0.004*	-0.1609	-0.0199
2	4	-0.1656	< 0.001*	-0.246	-0.0852
2	5	-0.1932	< 0.001*	-0.2687	-0.1176
2	6	-0.1983	< 0.001*	-0.2694	-0.1272
3	4	-0.0752	0.071	-0.154	0.0036
3	5	-0.1028	0.001*	-0.1766	-0.0289
3	6	-0.1079	<0.001*	-0.1772	-0.0386
4	5	-0.0276	0.9339	-0.1109	0.0557
4	6	-0.0327	0.846	-0.112	0.0466
5	6	-0.0051	1.0	-0.0795	0.0692

Next, we were specifically interested in *which layers* of different Elman RNNs units were rather tuned to timing. This was tested by conducting a paired t-test of monotonic- and tuned response model fits within layers (Table 3).

Table 3. Paired t-test results of monotonic- and tuned response model fits within layers.

		Monotonic		Tuned			
Network Type	Layer Number	M	SD	M	SD	N	P
1-layer	1	0.73	0.28	0.86	0.19	57	< 0.001*
2-layer	1	0.44	0.21	0.47	0.19	43	0.008*
	2	0.5	0.22	0.53	0.22	41	< 0.001*
3-layer	1	0.39	0.2	0.44	0.17	43	< 0.001*
	2	0.37	0.15	0.43	0.14	18	0.002*
	3	0.33	0.14	0.35	0.14	32	0.079
4-layer	1	0.25	0.19	0.34	0.15	32	< 0.001*
	2	0.32	0.2	0.37	0.15	7	0.077
	3	0.25	0.13	0.32	0.1	6	0.024*

	4	0.31	0.19	0.3	0.24	12	0.725
5-layer	1	0.22	0.17	0.32	0.12	44	< 0.001*
	2	0.18	0.18	0.3	0.18	6	0.023*
	3	0.22	0.27	0.28	0.15	4	0.35
	4	0.24	0.18	0.32	0.16	3	0.026*
	5	0.23	0.1	0.27	0.1	14	0.029*
6-layer	1	0.2	0.15	0.34	0.13	50	< 0.001*
	2	0.15	0.19	0.26	0.1	12	0.036*
	3	0.09	0.08	0.23	0.05	7	0.025*
	4	0.18	-	0.21	-	1	-
	5	0.17	0.09	0.21	0.08	5	0.522
	6	0.24	0.12	0.29	0.07	15	0.026*

Tuned fits were significantly higher than monotonic fits in the first layer of all Elman RNNs, with $\alpha = 0.05$. Furthermore, tuned fits were significantly higher in layer two of two-, three-, five-, and six-layer Elman RNNs, in layer three of four-, and six-layer Elman RNNs, in layer four and five of the five-layer Elman RNN, and in layer six of the six-layer Elman RNN. Note that N values differ largely throughout layers and architectures. So, especially units in lower layers of all Elman RNNs were rather tuned. Responses of units in higher layers could sometimes be equally well described by a monotonic response model.

From Figure 13 and Table 3, it follows that monotonic and tuned fits might be indifferent between layers of the same Elman RNN. More specifically, it seems like there is no systematic change from lower to higher layers. Thus, the difference between monotonic and tuned response model fits was measured for each unit, differences were grouped by layers, and an n-way ANOVA analysis for different Elman RNNs was conducted (Table 4).

Table 4. N-way ANOVA results for differences in tuned response model fits and monotonic response model fits in between layers of different Elman RNNs.

Network Type	Degrees of Freedom	F	P
2-layer	1	0.052	0.820
3-layer	2	3.245	0.044*
4-layer	3	3.820	0.015*
5-layer	4	0.850	0.499
6-layer	5	1.857	0.111

The only significant differences could be found in the three- and the four-layer Elman RNN, with $\alpha = 0.05$. Consecutive pairwise Tukey comparisons of differences between layers (Table 5), revealed that only units in layer one and layer four in a four-layer Elman RNN differ significantly. Every other pairwise comparison was insignificant. Thus, units in higher layers of different Elman RNNs did not systematically develop stronger monotonic or tuned response properties than units in lower layers.

Table 5. Pairwise Tukey comparison tests for differences of monotonic and tuned fits between layers of Elman RNNs with different depths.

Network Type	Layer Number	Layer Number	Mean Difference	P	95% Confidence Interval	
					Lower Bound	Upper Bound
2-layer	1	2	0.0027	0.820	-0.0209	0.0263
3-layer	1	2	0.0011	0.998	-0.0441	0.0463
	1	3	-0.0372	0.053	-0.0748	0.0004
	2	3	-0.0384	0.137	-0.0858	0.0091
4-layer	1	2	-0.0336	0.772	-0.1262	0.059
	1	3	-0.0097	0.994	-0.1084	0.0891
	1	4	-0.0947	0.008*	-0.1699	-0.0196
	2	3	0.0239	0.956	-0.0996	0.1474
	2	4	-0.0612	0.424	-0.1667	0.0444
	3	4	-0.0851	0.190	-0.1961	0.0259
5-layer	1	2	0.0155	0.997	-0.1087	0.1397
	1	3	-0.0298	0.980	-0.1788	0.1192
	1	4	-0.0217	0.996	-0.192	0.1485
	1	5	-0.0523	0.456	-0.1399	0.0352
	2	3	-0.0453	0.958	-0.2295	0.1389
	2	4	-0.0373	0.985	-0.239	0.1645
	2	5	-0.0678	0.651	-0.2071	0.0714
	3	4	0.0081	1.0	-0.2099	0.226
	3	5	-0.0225	0.995	-0.1843	0.1393
	4	5	-0.0306	0.990	-0.2121	0.1509
6-layer	1	2	-0.0372	0.920	-0.1469	0.0724
	1	3	-0.0001	1.0	-0.1378	0.1375
	1	4	-0.1059	0.946	-0.4505	0.2386
	1	5	-0.0979	0.481	-0.2579	0.0621
	1	6	-0.0864	0.133	-0.1869	0.014
	2	3	0.0371	0.985	-0.1251	0.1994
	2	4	-0.0687	0.993	-0.4237	0.2864
	2	5	-0.0607	0.925	-0.2423	0.1209
	2	6	-0.0492	0.886	-0.1813	0.0829

3	4	-0.1058	0.958	-0.4705	0.2589
3	5	-0.0978	0.710	-0.2975	0.102
3	6	-0.0863	0.593	-0.2425	0.0698
4	5	0.008	1.0	-0.3657	0.3817
4	6	0.0195	1.0	-0.3329	0.3718
5	6	0.0115	1.0	-0.1647	0.1876

Figure 14 shows the preferred timings of tuned response models where the tuned fit was higher than the monotonic fit for Elman RNNs from one to six layers. Again, preferred timings close to the corners as well as intermediate preferred timings can be found in networks from two to six layers. Compared to results from the one-layer Elman RNN, however, tuned response models of Elman RNNs with more than two layers lack preferred timings in the intermediate range of the right edge which corresponds to timings with variable preferred duration and a preferred period of 950 ms. Further, it appears, especially in Elman RNNs with more than two layers, that preferred timings in layer one (blue dots) are located

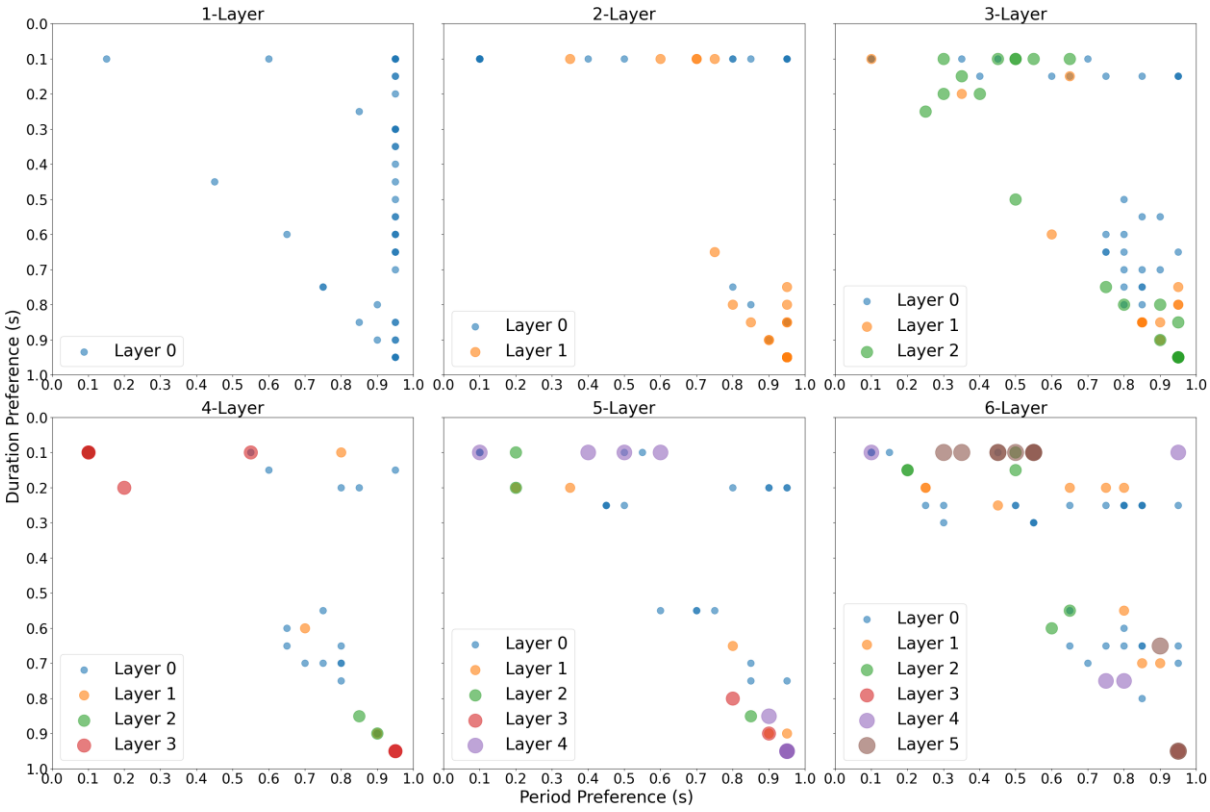


Figure 14. Per-event preferred timings of tuned response models for Elman RNNs with one to six layers, where the tuned fit is higher than the monotonic fit.

closer to the center, whereas units in higher layers tend to be located closer to the corners. To test this, the Euclidean distance of each preferred timing to the center of the timing range (duration = period = 500 ms) was measured, and an ANOVA analysis between layers of different Elman RNNs was conducted (Table 6).

Table 6. N-way ANOVA results of Euclidean distance of preferred timings to the center of the spectrum for Elman RNNs with different depths.

Network Type	Degrees of Freedom	F	P
2-layer	1	0.384	0.543
3-layer	2	0.708	0.499
4-layer	3	2.125	0.131
5-layer	4	1.116	0.371
6-layer	5	2.788	0.031*

Results were only significant for the six-layer Elman RNN, however, following pairwise Tukey comparisons were insignificant for all combinations of layers in every Elmann RNN (Appendix - Table 1). Therefore, the impression that preferred duration and periods of units in early layers of deep Elman RNNs tend to lie closer to intermediate timings cannot be confirmed.

Figure 15 shows an example of a unit with monotonic response properties in layer three of a four-layer Elman RNN in which the tuned response model is approximating the monotonic model. Counterintuitively, the monotonic fit is slightly lower (0.78) than the tuned

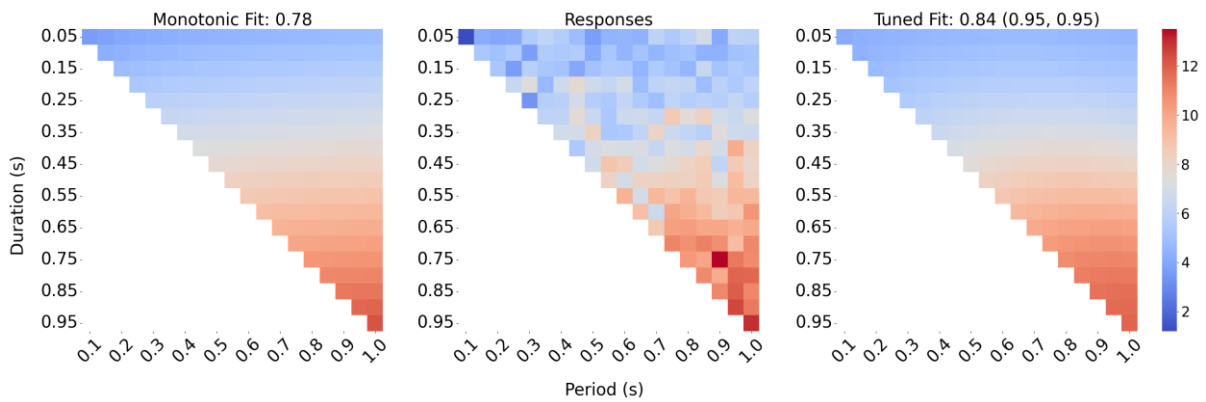


Figure 15. Responses of neuron 216 (layer 3) of a four-layer Elman RNN.

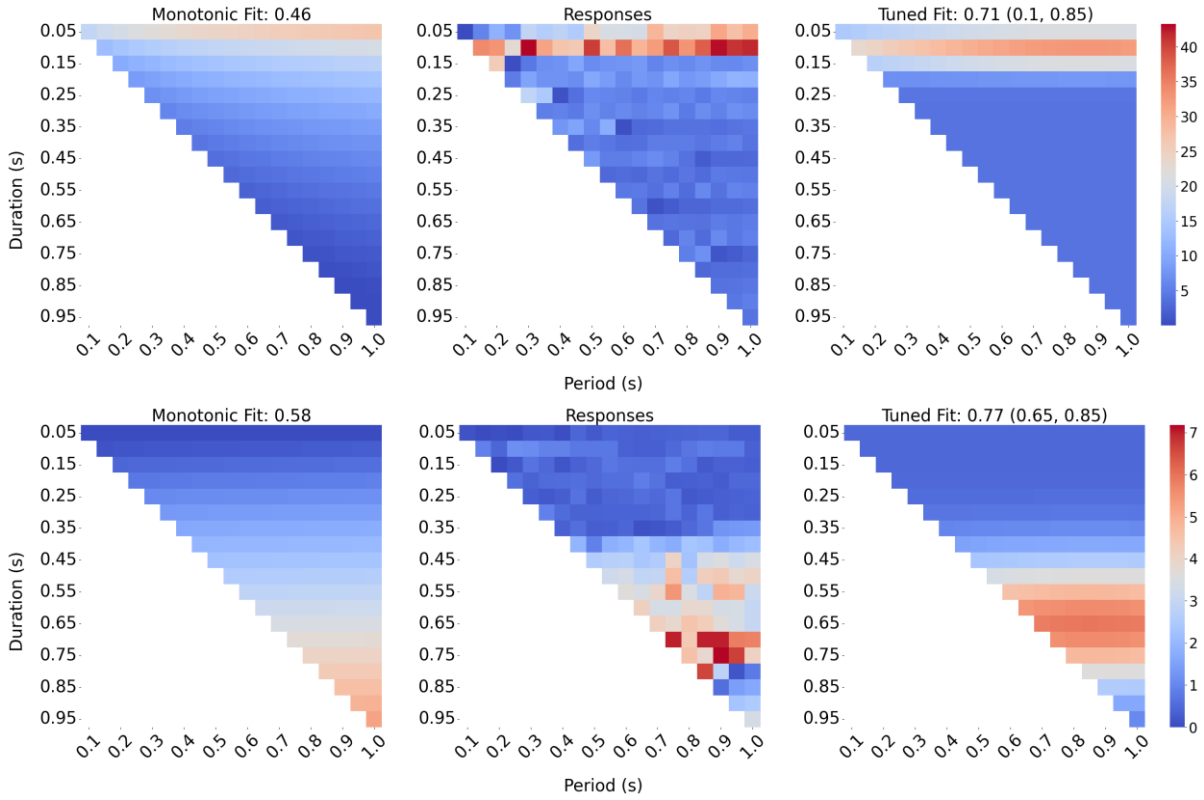


Figure 16. Neuron 44 (layer 1) of a two-layer Elman RNN (top) and neuron 1 (layer 1) of a six-layer Elman RNN (bottom)

fit (0.84). Contrary, Figure 16 shows examples of units with tuned response properties in an Elman RNNs with two layers (top) and six layers (bottom). Found tuned functions were neither tuned to both duration and period nor tuned to only event period. Rather, the angulation of tuned response models was horizontal and therefore possibly tuned to only event duration. Such units could only be found in the first layer of deeper Elman RNNs by manual investigation. Figure 17 shows the distribution of the angle parameter of a six-layer Elman RNN. Here, the absence of other angulations than the horizontal angulation shows that considered tuned response models can never be tuned to only period or both event duration and period. Note that angulation itself does not necessarily imply tuning to event duration as the corresponding model might still approximate a monotonic model or the goodness of fit

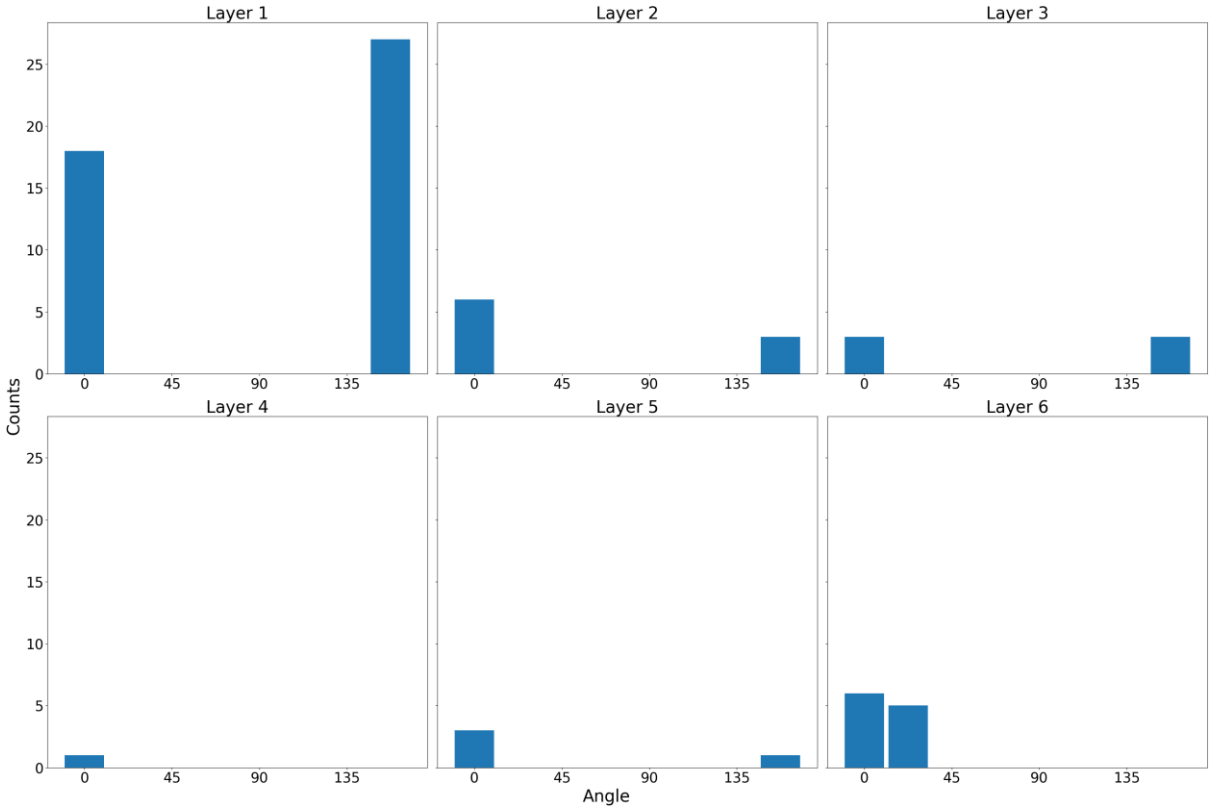


Figure 17. „Angle“ parameter distribution for tuned response models in a six-layer Elman RNN, where the tuned fit is higher than the monotonic fit.

might be very low. A similar pattern could be found in other deeper Elman RNNs (Appendix - Figure 1, Appendix - Figure 2, Appendix - Figure 3).

Figure 18 shows a special response function found in the first layer of the five-layer Elman RNN. Even though the monotonic and the tuned response models both show a mediocre fit, the units' response could be described by a tuned response model with three

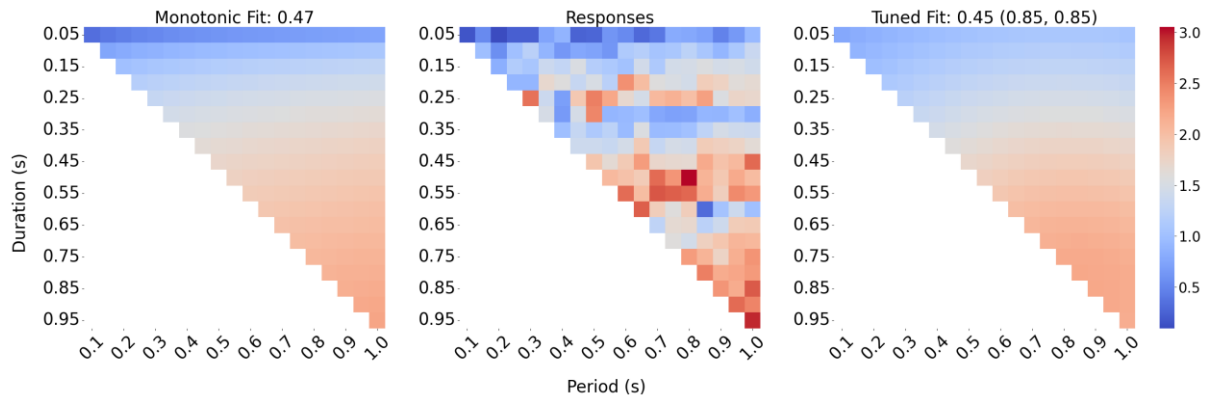


Figure 18. Responses of neuron 27 (layer 1) of a five-layer Elman RNN.

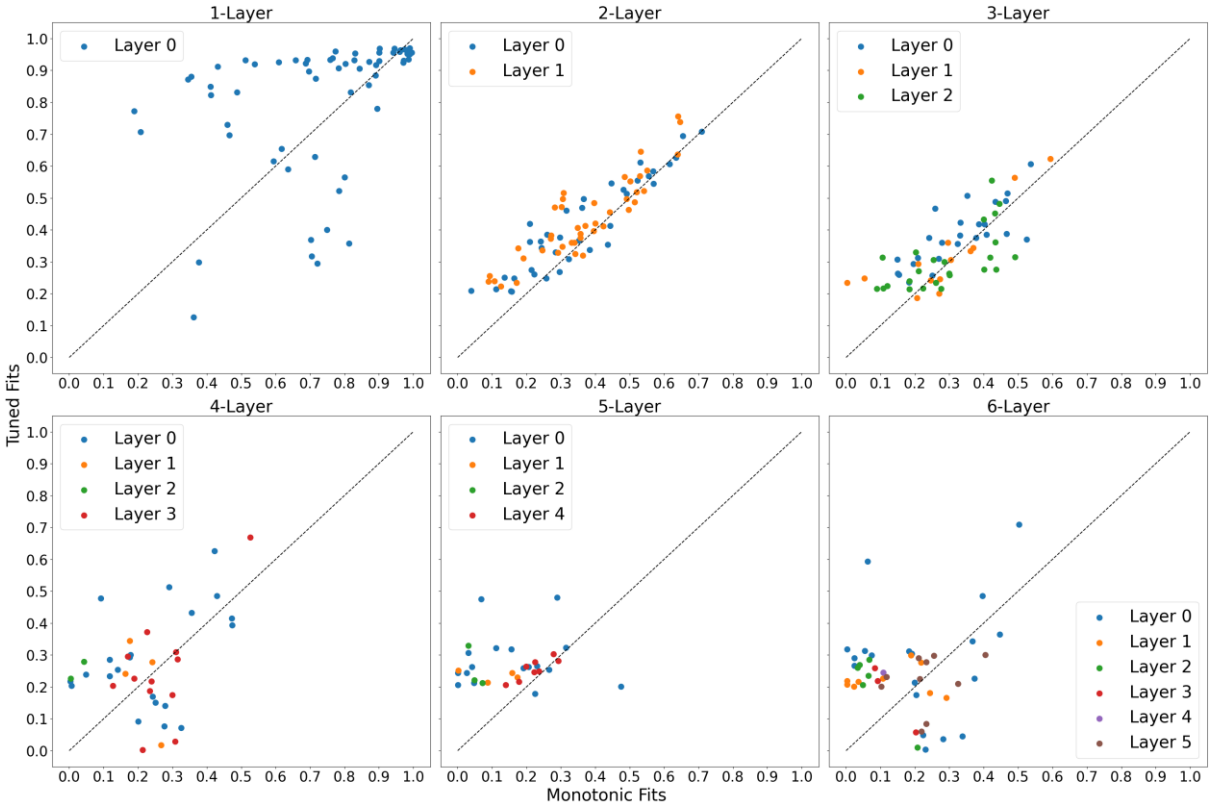


Figure 19. Per-movie monotonic and tuned fits for Elman RNNs with one to six layers.

preferred durations at 200 ms, 500 ms, and 900 ms and horizontal orientation. However, as the tuned response model can only be tuned to a single duration and period, the tuned response model was not able to fit the unit's response better than the monotonic response model. Therefore, units in deeper networks can develop response functions more complex than simple monotonic and tuned profiles allow.

Per-movie fitting of neural response models (Figure 19) showed that units in a one-layer Elman RNN can be clustered into three populations of units: one in which both monotonic and tuned response models fit well (top right corner), one in which only tuned response models fit well (towards the top left corner), and one in which monotonic response models fit well (towards the bottom right corner). Compared to the per-event fitting methodology, overall fits were lower for both monotonic and tuned response models in every Elman RNNs. Confirming this impression, paired t-tests for monotonic as well as tuned fits were significant with $p < 0.001$, $n = 241$ and $p < 0.001$, $n = 241$. Preferred timings of tuned

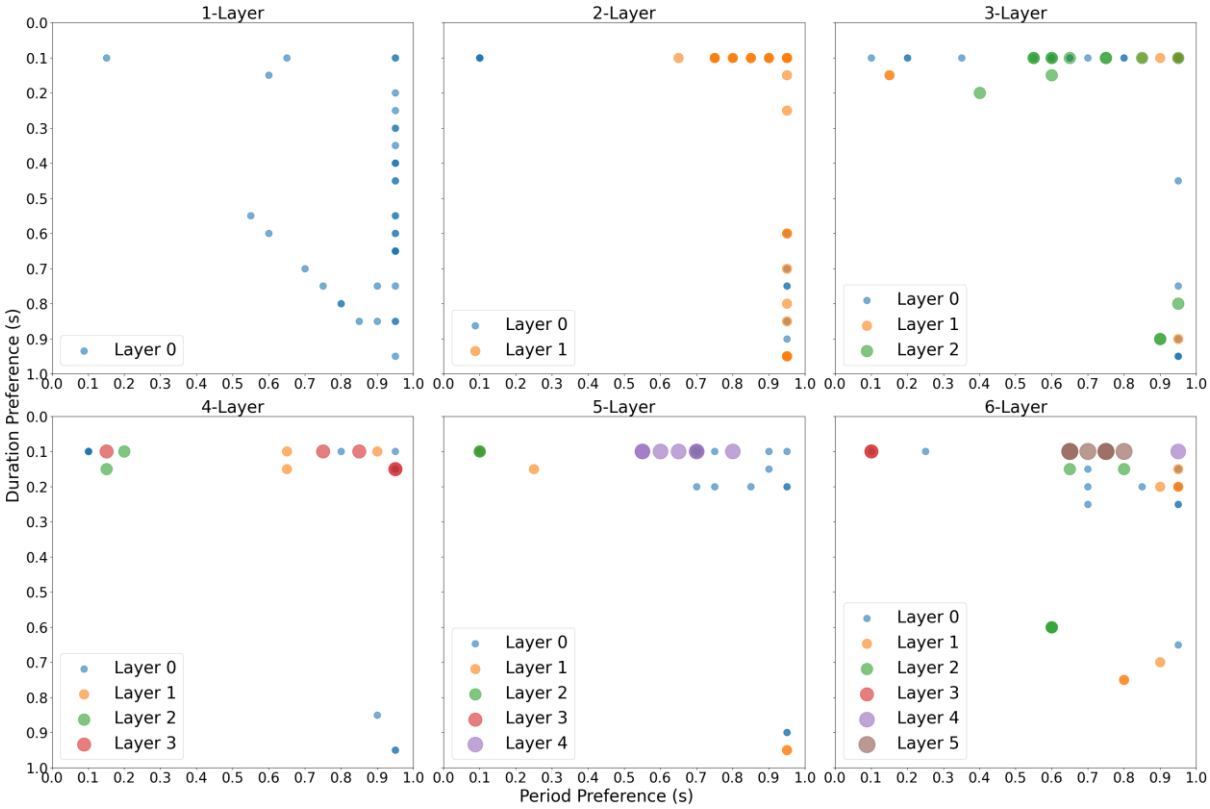


Figure 20. Per-movie preferred timings of tuned response models for Elman RNNs with one to six layers, where the tuned fit is higher than the monotonic fit.

response models for hierarchical Elman RNNs were mostly grouped in the upper right corner contrary to the lower right corner in per-event fitting, respectively corresponding to timings with high period and low duration and timings with high duration and high period (Figure 20).

LSTM RNN

Per-event fitting of LSTM RNNs showed that in a one-layer LSTM RNN monotonic response models fit clearly better (Figure 21, left). In a two-layer LSTM RNN, however, both monotonic and tuned fits were generally lower and the difference between both models is very low (Figure 21, right). Figure 22 shows that preferred durations and periods of tuned response models are mostly located at the edges of the spectrum and that some models for the

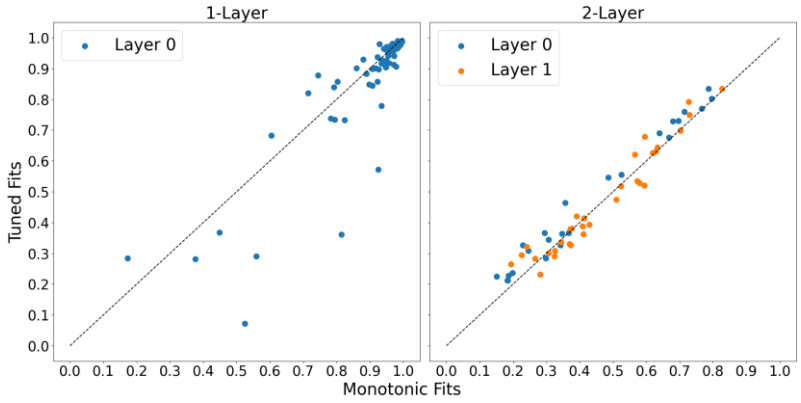


Figure 21. Per-event monotonic and tuned fits for LSTM RNNs with one layer (left) and two layers (right).

one-layer LSTM RNN are tuned in the intermediate range of the right edge analogous to the one-layer Elman RNN.

Again, per-movie fits of LSTM RNNs were generally lower for both monotonic and tuned response models (Appendix - Figure 5) but preferred timings of tuned models for the one-layer LSTM RNN were scattered more to the center of the spectrum (Appendix - Figure 6).

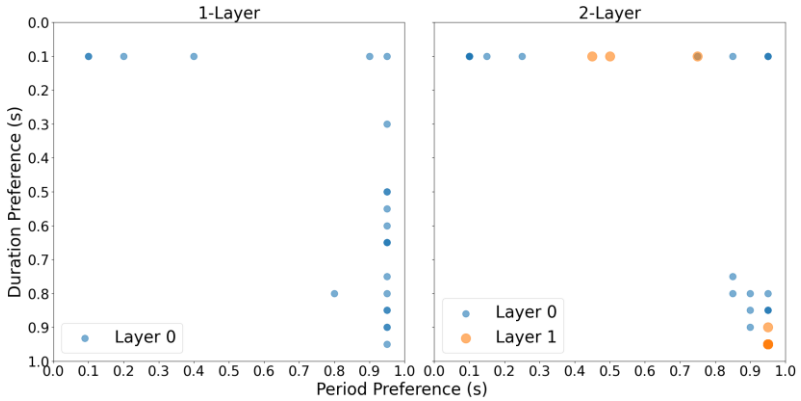


Figure 22. Per-event preferred timings of tuned response models for LSTM RNNs with one layer (left) and two layers (right), where the tuned fit is higher than the monotonic fit.

Discussion

In the current study, we aimed to model sensory responses to visual event timing in the human visual cortex using recurrent GNNs. We could find monotonic responses to event duration and period in different layers of hierarchical GNNs. However, we could not find a gradual change from monotonic to tuned responses, as seen in the human visual hierarchy (Hendrikx et al., 2022). Rather, units in hierarchical Elman RNNs only exhibited tuning to event duration in the first layer, and monotonic response properties in all layers. In units of non-hierarchical Elman RNNs, we found monotonic as well as tuned responses to both event duration and period.

Non-hierarchical Elman RNNs, hierarchical Elman RNNs, and LSTM RNNs contained the ability to efficiently encode and accurately predict timings without supervised feedback. Therefore, an efficient, unsupervised transformation of visual stimuli without explicit feedback suffices as a basis for correct predictions and shows that information necessary for timing prediction is inherent to visual input. This suggests that the human visual cortex might similarly comprise the ability to transform visual inputs into an efficient encoding without supervised feedback. Diverging from the hierarchically organized structure of the human visual cortex, non-hierarchical Elman RNNs archived similar performance levels to hierarchical Elman RNNs with up to six layers. This clearly shows that in Elman RNNs, a hierarchical structure is not required to successfully encode temporal information from visual stimuli. Hierarchical Elman RNNs perform several transformations distributed over multiple layers. These transformations are likely to be redundant, not contributing to an enhanced encoding of timing, as the similar performance between non-hierarchical and hierarchical Elman RNNs shows. The faster training seen in deeper Elman RNNs may simply occur because deeper networks incorporate more weights to encode event timing.

Responses in human early visual cortex areas can be described by monotonic, sub-additive functions of event duration and frequency (Zhou et al., 2018), likely to arise from a summation of transient and sustained neural responses (Stigliani et al., 2017). The existence of monotonic responses to event duration and period in hierarchical as well as non-hierarchical Elman RNNs support these findings. Due to complex top-down and bottom-up interactions as well as interactions between brain areas, fMRI measurements reflect multiple effects on neural responses in any brain area, impeding detection of causal relations from stimulus to response. In GNNs, such interactions are not present and thus we could determine transient and sustained components of visual inputs to causally influence monotonic responses.

First layers of hierarchical Elman RNNs sometimes exhibited tuning to only event duration. A closer investigation of such duration-tuned responses showed that tuning to duration is likely to be an artifact of the per-event fitting methodology. Figure 16 (top) showed a unit with tuning to event duration in the per-event fitting methodology, Figure 23 shows the same unit in the per-movie fitting methodology. Evidently, the unit does not show tuning to event duration in the per-movie fitting methodology. This can be explained by the per-event fitting methodology mostly factoring out the period aspect of timing, sometimes resulting in approximately constant responses for stimuli with a certain event duration.

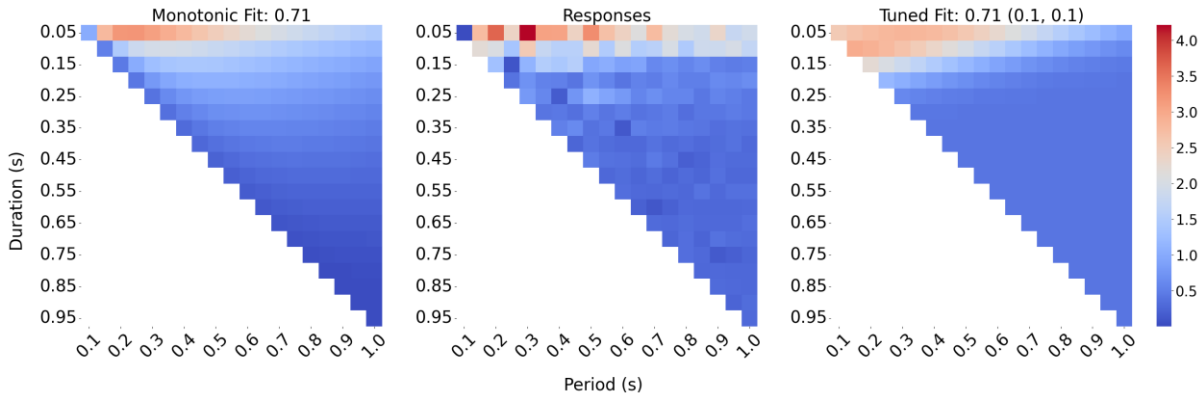


Figure 23. Per-movie responses of neuron 44 (layer 1) in a two-layer Elman RNN.

Human association cortices inhabit a network of topographically organized maps inducing timing tuned response (Harvey et al., 2020). Early visual areas induce monotonically increasing responses to event timing that gradually transition into timing-tuned responses in higher visual areas and finally feed into timing-tuned networks in association cortices (Hendrikx et al., 2022). These responses become increasingly retinotopically invariant in higher visual areas (Hendrikx et al., 2022). In contrast to findings in the human visual system, we could not find tuned responses in higher layers, nor a transition from monotonic to timing tuned responses, in hierarchical Elman RNNs. The reason for this may be found in the architecture of the Elman RNN; connections within layers (recurrent component), as well as connections between layers (input component), are dense, allowing variable interaction within- and between layers. Since dense neural networks with a single layer (and a sufficiently large number of neurons) are universal function approximators (Hornik et al., 1989), both dense connections can approximate complex functions. Hence, already a single layer can integrate different responses to event timing (recurrent component), combine resulting representation with a representation of the current input (input component), and finally capture complex timings. Consequently, timing predictions were very accurate even in a non-hierarchical Elman RNN. Here, we could find monotonic, as well as duration- and period-tuned responses in a single layer suggesting both monotonic- and tuned responses being inevitable for temporal prediction. Contrary, in hierarchical Elman RNNs, additional layers are redundant, and functionality was spaced out over units in different layers, leading to simpler response patterns in individual units. Such simpler response patterns can be combined through complex neural dynamics across layers, yielding complex- and perhaps even tuned functions. Yet, we cannot capture these complex functions, as we only measured single units’ responses.

In human brains, there are two segregated transient and sustained channels from the retina and the lateral geniculate nucleus, feeding into neurons in V1. Through V1's retinotopic organization, connectivity of neurons within layers is spatially restricted to neurons of identical receptive field maps. Connectivity between visual areas is equally restricted by receptive field map size but the spatial connectivity tends to grow along the visual hierarchy. Based on between-layer as well as within-layer connections, a neuron integrates excitatory and inhibitory polarization. Over time, a neuron's integrated polarization is slowly decaying unless it is depolarized through an action potential. Considering this structure, a neuron is primarily dependent on its activation from the previous time step (i.e., state-dependent), activity of connected neurons in the predecessor visual area, as well as activity of nearby, spatially connected neurons (i.e., lateral inhibition). In this study, we did not consider a variable spatial position of visual inputs, therefore constraints regarding spatial connectivity between-, as well as within-layers can be disregarded. In terms of modeling biological neural systems, this means that interactions between layers can be complex, and interactions within layers are more limited and modulatory. The Elman RNN architecture captures complex between-layer interactions through a dense input connection; however, it *additionally* enables complex within-layer interactions through a dense, recurrent connection. Compared to the human visual cortex, however, we think that such dense within-layer interactions (recurrent component) are not biologically plausible. Such connections are conceptionally equivalent to global lateral synapses within layers, which cannot be observed in human visual cortex areas.

Biologically implausible capabilities of Elman RNNs conflict with our aim to compare response development in hierarchical GNNs with response development in the human visual cortex. Here, responses to timing can be described by sub-additive monotonic functions in early visual areas, increasingly turning into tuned functions in higher visual areas

(Hendrikx et al., 2022). Due to the redundancy of hierarchical layers in Elman RNNs, units' responses did not develop systematically in higher layers. Importantly, we want to underline that, conceivably, increasingly sub-additive monotonic functions might not emerge, even in GNNs with enhanced biological plausibility. GNNs follow optimization of a single inherent learning target (e.g., prediction of the next image in a sequence) and consequently, units' response properties will excessively develop to benefit this singular inherent learning target. In the current study, inherent learning target was solitarily concerning the timing aspect of visual movies. Possibly, monotonic response properties do not provide any beneficial features for the realization of timing prediction. In such a scenario, GNN units might immediately approximate more complex functions, which yield a greater reduction in prediction error. Contrary, the human visual cortex does not follow a singular, well-defined learning target. Rather, its neural structure for realization of diverse cognitive capabilities was formed by evolution, implicitly optimizing spatial arrangement by grouping neurons with similar response patterns in close vicinity. Therefore, its functionality simultaneously involves spatially- *or* temporally dependent functions (i.e., numerosity and timing prediction) in intermediate visual areas, among other spatially *and* temporally dependent cognitive functions in higher areas (MT), such as motion detection. In particular, processing strategies for spatial- and temporal visual stimuli seem to be related, because both (spatial and temporal responses) increasingly exhibit temporal or spatial integration in early visual areas (V1 – V3) (Zhou et al., 2018). It remains an open question whether increasingly sub-additive, monotonic functions as seen in the human visual hierarchy are obligatory for solitary temporal prediction in hierarchical GNNs, or if these properties are emergent properties of related cognitive functions such as numerosity prediction.

A possible first step toward investigation of the relation between temporal- and spatial abstraction in hierarchical GNNs, could be the introduction of a variable spatial position in

timing movies. Monotonic responses to visual event timing in early areas of the visual cortex are bound to retinotopic location and monotonic responses in later areas become increasingly invariant to retinotopic position (Hendrikx et al., 2022). Hierarchical GNNs, capable of accurately predicting timing from spatially variable stimuli must contain the capability to encode timing information independent of spatial position. Future research should focus on combining established techniques from image recognition domains with research focused on temporal processing. A possible approach to capture spatial position could be incorporation of convolutional filters in layers of Elman RNNs, resembling DCNN architectures. Due to DCNNs developing increasingly spatially invariant responses in higher layers, we would similarly expect some sort of temporal abstraction for temporal processing. However, such a model should overcome biological implausibility in recurrent components and exhibit a hierarchical structure which is benefitting timing prediction.

A possible candidate for this purpose could be the IndRNN architecture (Li et al., 2018). Here, recurrent connections are linear, inducing independence of units within layers. Therefore, a unit's activation is primarily dependent on its activation of the previous time step as well as output activation of the predecessor layer. Specifically, it has been proven that IndRNNs with two layers and invertible recurrent weights are functionally equivalent to Elman RNNs with a single layer (Li et al., 2018). So, the IndRNN should be able to encode timing equally well as the non-hierarchical Elman RNN. The complexity and prediction capabilities of IndRNNs could be increased by adding hierarchical layers. Responses to visual event timing in these layers and especially the development of these responses remain to be further investigated.

Another open question remains the influence of interactions between stimuli on response patterns of GNN units. In the current study, hidden states of GNNs were reset after the presentation of a single timing movie. However, weighted carry-over of previous

activation could also act as the input hidden state for the next timing movie. This procedure could simulate measurements from fMRI studies in which temporal resolution is very coarse and stimuli are presented in a specific order. Implementing carry-over effects could enhance comparisons of GNN responses to visual cortex's responses and unwanted carry-over effects in fMRI studies could be estimated, possibly leading to adjustments in experimental design.

Despite the limitations by used GNN architectures, this study was in some cases limited by the fitting methodology. Even though we were aware that tuned functions could approximate monotonic functions, we assumed that a monotonic function should fit the responses better in such a case. However, tuned fits were often higher than monotonic fits, regardless of manual inspection suggesting a monotonic response with no clear preferred timing peak. This behavior occurred whenever tuned functions had a preferred duration and period close to the edge of the spectrum and had a large standard deviation. Such tuned functions are in effect a monotonic function but additionally express a slight concavity in intermediate duration- and period range, allowing more complex monotonic relationships between timing and response amplitude and so leading to an improved fit. Implementing cross-validation ensures that complex monotonic relationships (approximated by tuned functions) are repeatable to out-perform simpler monotonic relationships. However, differentiating whether a unit with similar fits was either tuned or monotonic remains difficult. One possible solution to this problem would be to limit the tuned model's major- and minor axis standard deviation parameter range, such that a tuned function with a preferred timing indeed exhibits a perceivable peak. Diverging from restricting parameter values, the goodness of fit of neural response models could be measured with a criterion incorporating the explained variance as well as a regularization term for model complexity. Thus, more complex response models with functionally the same properties would receive

lower goodness of fit compared to simpler models. However, such approaches are often heavily dependent on regularization terms.

By utilizing unsupervised recurrent generative neural networks to encode timing-dependent visual stimuli, this study showed that such artificial neural networks can suffice as a model for sensory visual processing in the human visual cortex. Especially, non-hierarchical GNNs exhibited remarkably similar responses compared to responses seen in the human visual cortex. Thus, this study can be seen as the first step toward modeling visually driven responses in human visual areas in a computational way. However, biological implausible elements of the GNNs limited comparisons to the response development along the human visual hierarchy. Overcoming biological implausibility to enhance hierarchical GNNs explanatory power, remains the most important topic for future work. Such models could be a major component in the investigation of visually driven timing responses and likewise play an important role in examining the relationship between timing- and spatial abstraction in the human visual cortex.

References

Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016). Layer Normalization. *ArXiv Preprint ArXiv:1607.06450*. http://arxiv.org/abs/1607.06450

Buonomano, D. v., & Karmarkar, U. R. (2002). Book Review: How Do We Tell Time? *The Neuroscientist*, *8*(1), 42–51. https://doi.org/10.1177/107385840200800109

Buonomano, D. v, & Merzenich, M. M. (1995). Temporal Information Transformed into a Spatial Code by a Neural Network with Realistic Properties. In *New Series* (Vol. 17, Issue 5200).

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014, June 3). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *Proceedings of the Empiricial Methods in Natural Language Processing*. http://arxiv.org/abs/1406.1078

Cox, D. D., & Dean, T. (2014). Neural networks and neuroscience-inspired computer vision. *Current Biology*, *24*(18), R921–R929. https://doi.org/10.1016/j.cub.2014.08.026

Fountas, Z., Sylaidi, A., Nikiforou, K., Seth, A. K., Shanahan, M., & Roseboom, W. (2022). A predictive processing model of episodic memory and time perception. *Neural Computation*, *34*(7), 1501–1544. https://doi.org/10.1101/2020.02.17.953133

Gibbon, J. (1977). Scalar Expectancy Theory and Weber's Law in Animal Timing. In *Psychological Review* (Vol. 84, Issue 3).

Graves, A., Mohamed, A., & Hinton, G. (2013). Speech Recognition with Deep Recurrent Neural Networks. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 6645–6649. http://arxiv.org/abs/1303.5778

Harvey, B. M., & Dumoulin, S. O. (2017). A network of topographic numerosity maps in human association cortex. *Nature Human Behaviour*, *1*(2). https://doi.org/10.1038/s41562-016-0036

Harvey, B. M., Dumoulin, S. O., Fracasso, A., & Paul, J. M. (2020). A Network of Topographic Maps in Human Association Cortex Hierarchically Transforms Visual Timing-Selective Responses. *Current Biology*, *30*(8), 1424-1434.e6. https://doi.org/10.1016/j.cub.2020.01.090

Harvey, B. M., Fracasso, A., Petridou, N., & Dumoulin, S. O. (2015). Topographic representations of object size and relationships with numerosity reveal generalized quantity processing in human parietal cortex. *Proceedings of the National Academy of Sciences of the United States of America*, *112*(44), 13525–13530. https://doi.org/10.1073/pnas.1515414112

Hazeltine, E., Helmuth, L. L., & Ivry, R. B. (1997). Neural mechanisms of timing. *Trends in Cognitive Sciences*, *1*(5), 163–169. https://doi.org/10.1016/S1364-6613(97)01058-9

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *Proceedings of the IEEE International Conference on Computer Vision*, 1026–1034.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778. http://arxiv.org/abs/1512.03385

Hendrikx, E., Paul, J. M., van Ackooij, M., van der Stoep, N., & Harvey, B. M. (2022). Visual timing-tuned responses in human association cortices and response dynamics in early visual cortex. *Nature Communications*, *13*(1), 3952. https://doi.org/10.1038/s41467-022-31675-9

Hinton, G. E., & Ghahramani, Z. (1997). Generative Models for Discovering Sparse Distributed Representations. In *Philosophical Transactions: Biological Sciences* (Vol. 352, Issue 1358). https://about.jstor.org/terms

Hinton, G. E., Osindero, S., & Teh, Y.-W. (2006). A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation*, *18*(7), 1527–1554. https://doi.org/10.1162/neco.2006.18.7.1527

Hochreiter, S. (1998). The Vanishing Gradient Problem During Learning Recurrent Neural Nets And Problem Solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *6*(2), 107–116.

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9.8*, 1735–1780.

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer Feedforward Networks are Universal Approximators. *Neural Networks*, *2*(5), 359–366.

Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *International Conference on Machine Learning*, 448–456.

Ivry, R. B., & Spencer, R. M. C. (2004). The neural representation of time. *Current Opinion in Neurobiology*, *14*(2), 225–232. https://doi.org/10.1016/j.conb.2004.03.013

Kim, G., Jang, J., Baek, S., Song, M., & Paik, S.-B. (2021). Visual number sense in untrained deep neural networks. In *Sci. Adv* (Vol. 7, Issue 1). https://www.science.org

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems* (Vol. 25). Curran Associates, Inc. https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

Li, S., Li, W., Cook, C., Zhu, C., & Gao, Y. (2018). Independently Recurrent Neural Network (IndRNN): Building A Longer and Deeper RNN. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5457–5466.

Matell, M. S., & Meck, W. H. (2004). Cortico-striatal circuits and interval timing: Coincidence detection of oscillatory processes. *Cognitive Brain Research*, *21*(2), 139–170. https://doi.org/10.1016/j.cogbrainres.2004.06.012

Medina, J. F., Carey, M. R., & Lisberger, S. G. (2005). The Representation of Time for Motor Learning. *Neuron*, *45*(1), 157–167. https://doi.org/10.1016/j.neuron.2004.12.017

Merchant, H., Pérez, O., Zarco, W., & Gámez, J. (2013). Interval tuning in the primate medial premotor cortex as a general timing mechanism. *Journal of Neuroscience*, *33*(21), 9082–9096. https://doi.org/10.1523/JNEUROSCI.5513-12.2013

Paul, J. M., van Ackooij, M., ten Cate, T. C., & Harvey, B. M. (2022). Numerosity tuning in human association cortices and local image contrast representations in early visual cortex. *Nature Communications*, *13*(1), 1–15. https://doi.org/10.1101/2021.03.28.437364

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1985). *Learning internal representations by error propagation*.

Shannon, R. v, Zeng, F.-G., Kamath, V., Wygonski, J., & Ekelid, M. (1995). Speech Recognition with Primarily Temporal Cues. In *New Series* (Vol. 13, Issue 5234).

Srivastava, N., Hinton, G., Krizhevsky, A., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. In *Journal of Machine Learning Research* (Vol. 15).

Stigliani, A., Jeska, B., & Grill-Spector, K. (2017). Encoding model of temporal processing in human visual cortex. *Proceedings of the National Academy of Sciences of the United States of America*, *114*(51), E11047–E11056. https://doi.org/10.1073/pnas.1704877114

Stoianov, I., & Zorzi, M. (2012). Emergence of a “visual number sense” in hierarchical generative models. *Nature Neuroscience*, *15*(2), 194–196. https://doi.org/10.1038/nn.2996

Testolin, A., Dolfi, S., Rochus, M., & Zorzi, M. (2020). Visual sense of number vs. sense of magnitude in humans and machines. *Scientific Reports*, *10*(1). https://doi.org/10.1038/s41598-020-66838-5

Verguts, T., & Fias, W. (2004). Representation of Number in Animals and Humans: A Neural Model. *Journal of Cognitive Neuroscience*, *16*(9), 1493–1504.

Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2015, June). Show and Tell: A Neural Image Caption Generator. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

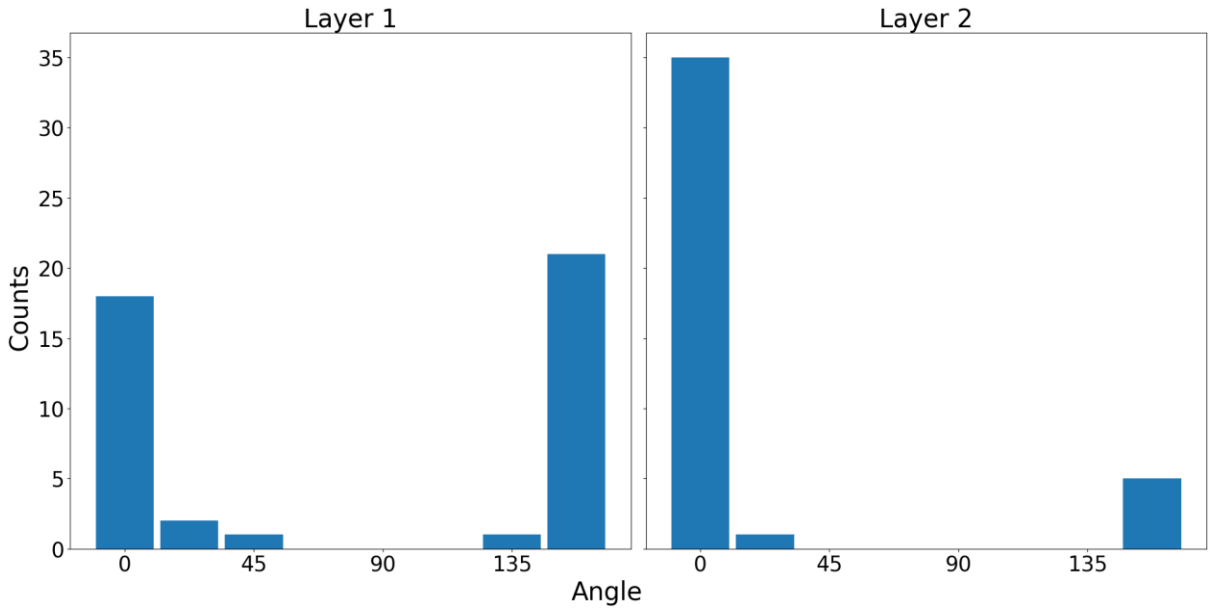
Yamins, D. L. K., Hong, H., Cadieu, C. F., Solomon, E. A., Seibert, D., & DiCarlo, J. J. (2014). Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the National Academy of Sciences of the United States of America*, *111*(23), 8619–8624. https://doi.org/10.1073/pnas.1403112111

Zhou, J., Benson, N. C., Kay, K. N., & Winawer, J. (2018). Compressive temporal summation in human visual cortex. *Journal of Neuroscience*, *38*(3), 691–709. https://doi.org/10.1523/JNEUROSCI.1724-17.2017

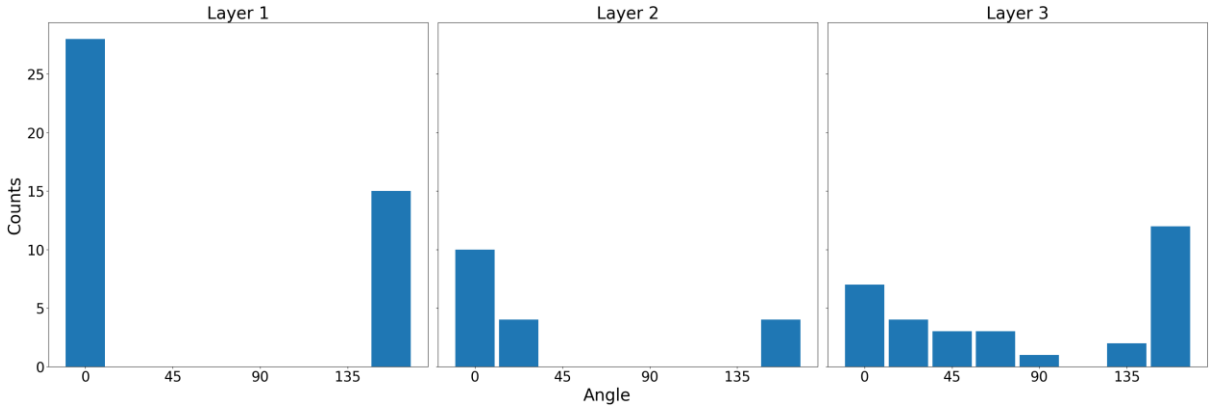
Zorzi, M., & Testolin, A. (2018). An emergentist perspective on the origin of number sense. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *373*(1740). https://doi.org/10.1098/rstb.2017.0043

Zorzi, M., Testolin, A., & Stoianov, I. P. (2013). Modeling language and cognition with deep unsupervised learning: A tutorial overview. *Frontiers in Psychology*, *4*(AUG). https://doi.org/10.3389/fpsyg.2013.00515

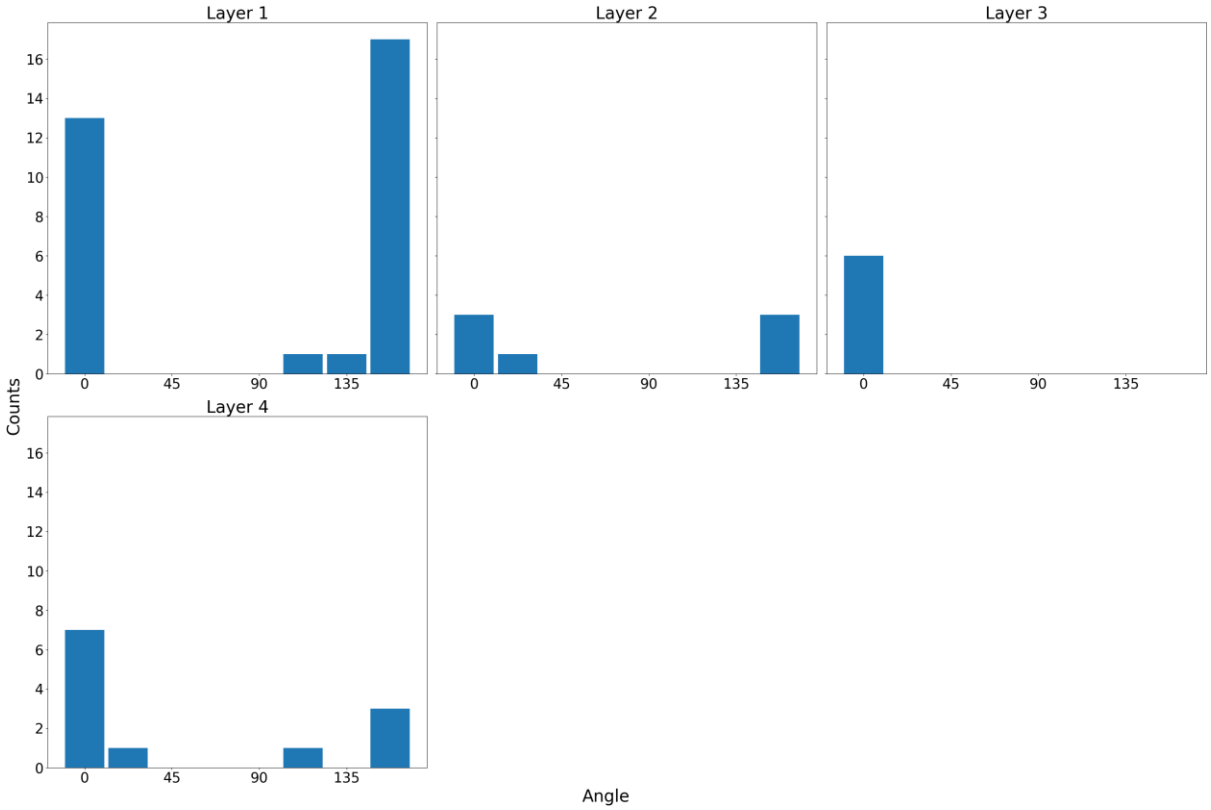
Appendix:



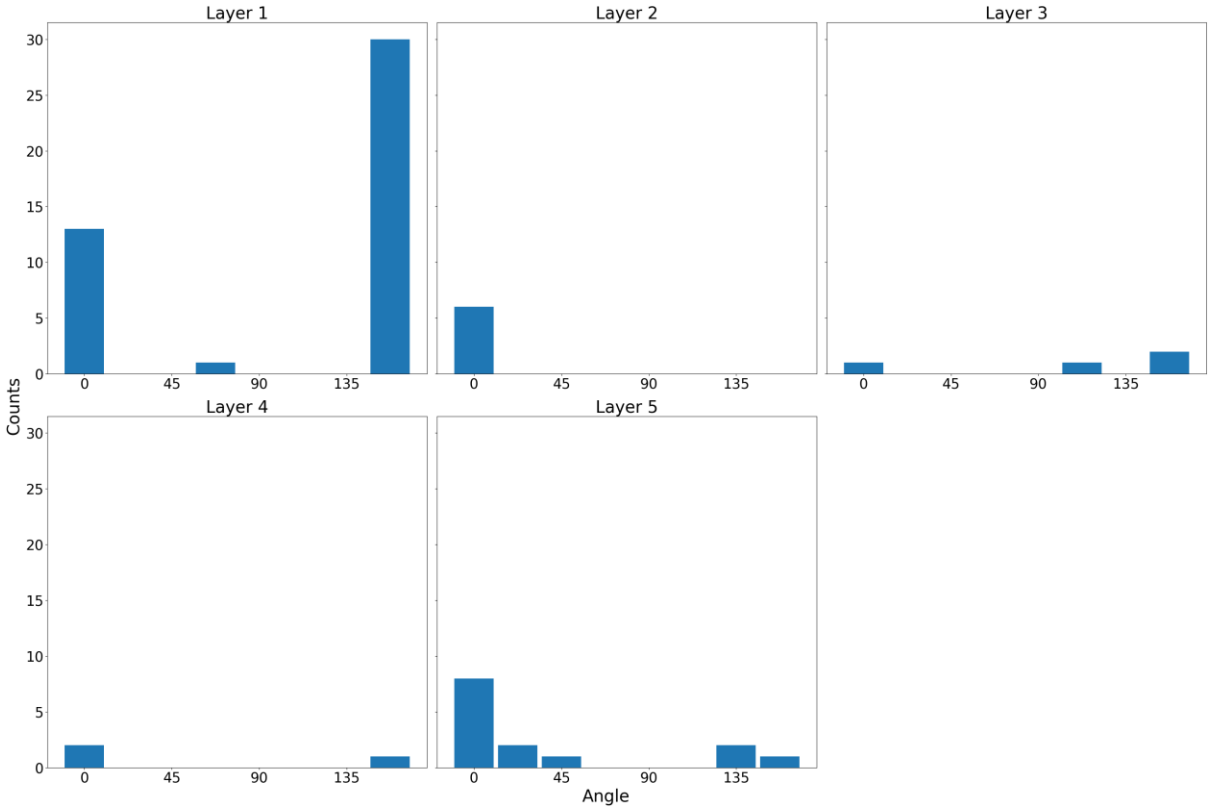
Appendix - Figure 1. „Angle“ parameter distribution for tuned response models in a two-layer Elman RNN, where the tuned fit is higher than the monotonic fit.



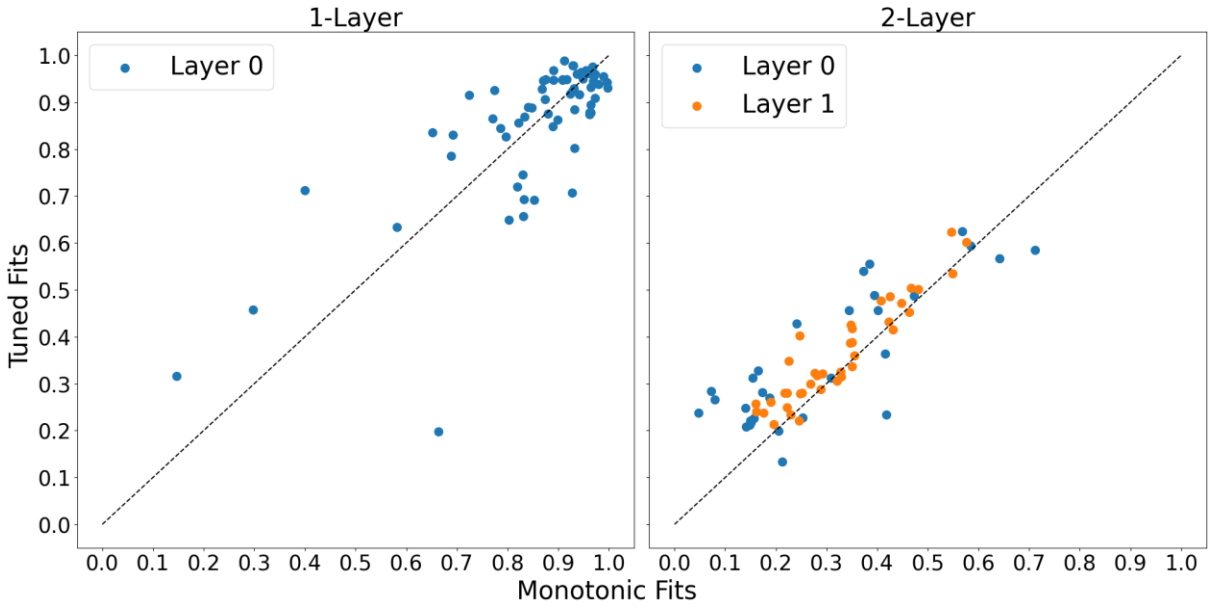
Appendix - Figure 2. „Angle“ parameter distribution for tuned response models in a three-layer Elman RNN, where the tuned fit is higher than the monotonic fit.



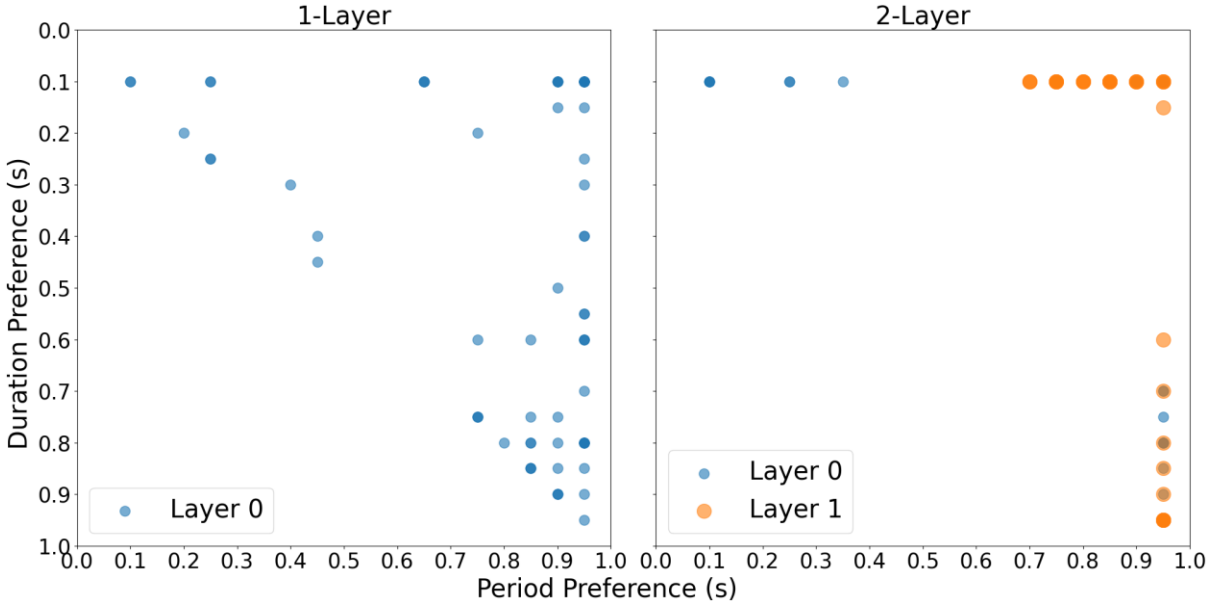
Appendix - Figure 3.„Angle“ parameter distribution for tuned response models in a four-layer Elman RNN, where the tuned fit is higher than the monotonic fit.



Appendix - Figure 4. .„Angle“ parameter distribution for tuned response models in a five-layer Elman RNN, where the tuned fit is higher than the monotonic fit.



Appendix - Figure 5. Per-movie monotonic and tuned fits for LSTM RNNs with one layer (left) and two layers (right).



Appendix - Figure 6. Per-movie preferred timings of tuned response models for LSTM RNNs with one layer (left) and two layers (right), where the tuned fit is higher than the monotonic fit.

Appendix - Table 1. Pairwise Tukey comparison tests for the distance of preferred timings of the tuned response model to the center of the range.

Network Type	Layer Number	Layer Number	Mean Difference	P	95% Confidence Interval	
					Lower Bound	Upper Bound
2-layer	1	2	0.0258	0.232	-0.017	0.0686
3-layer	1	2	0.0539	0.357	-0.0397	0.1475
	1	3	0.0331	0.591	-0.0478	0.114
	2	3	-0.0208	0.872	-0.1206	0.0791
4-layer	1	2	0.0338	0.937	-0.1221	0.1897
	1	3	0.1036	0.236	-0.041	0.2482
	1	4	0.1039	0.153	-0.0252	0.233
	2	3	0.0698	0.769	-0.1236	0.2631
	2	4	0.0701	0.733	-0.1119	0.2521
	3	4	0.0003	1.0	-0.1721	0.1727
5-layer	1	2	0.0155	0.997	-0.1087	0.1397
	1	3	-0.0298	0.980	-0.1788	0.1192
	1	4	-0.0217	0.996	-0.192	0.1485
	1	5	-0.0523	0.456	-0.1399	0.0352
	2	3	-0.0453	0.958	-0.2295	0.1389
	2	4	-0.0373	0.985	-0.239	0.1645
	2	5	-0.0678	0.651	-0.2071	0.0714
	3	4	0.0081	1.0	-0.2099	0.226
	3	5	-0.0225	0.995	-0.1843	0.1393
	4	5	-0.0306	0.99	-0.2121	0.1509
6-layer	1	2	-0.0489	0.88	-0.179	0.0812
	1	3	-0.0915	0.517	-0.2463	0.0634
	1	4	0.2163	0.498	-0.1439	0.5766
	1	5	0.0579	0.942	-0.128	0.2438
	1	6	0.0336	0.963	-0.0863	0.1534
	2	3	-0.0426	0.985	-0.2304	0.1452
	2	4	0.2652	0.315	-0.1103	0.6408
	2	5	0.1068	0.69	-0.1073	0.3209
	2	6	0.0825	0.66	-0.0777	0.2426
	3	4	0.3078	0.191	-0.077	0.6926
	3	5	0.1494	0.409	-0.0806	0.3794
	3	6	0.1251	0.338	-0.0558	0.3059
	4	5	-0.1584	0.852	-0.5568	0.2399
	4	6	-0.1828	0.703	-0.5549	0.1894
	5	6	-0.0243	0.999	-0.2323	0.1837