# Modeling Unsupervised Event Segmentation: Learning Event Boundaries from Prediction Errors

**Katherine Metcalf (metcalka@indiana.edu)**
Department of Computer Science, 901 E. 10th St.
Bloomington, IN 47408 USA

**David Leake (leake@indiana.edu)**
Department of Computer Science, 901 E. 10th St.
Bloomington, IN 47408 USA

## Abstract

Segmenting observations from an input stream is an important capability of human cognition. Evidence suggests that humans refine this ability through experiences with the world. However, few models address the unsupervised development of event segmentation in artificial agents. This paper presents work towards developing a computational model of how an intelligent agent can independently learn to recognize meaningful events in continuous observations. In this model, the agent's segmentation mechanism starts from a simple state and is refined. The agent's interactions with the environment are unsupervised and driven by its expectation failures. Reinforcement learning drives the mechanism that identifies event boundaries by reasoning over a gated-recurrent neural network's expectation failures. The learning task is to reduce prediction error by identifying when one event transitions into another. Our experimental results support that reinforcement learning can enable detecting event boundaries in continuous observations based on a gated-recurrent neural network's prediction error and that this is possible with a simple set of features.

**Keywords:** Event Cognition; Unsupervised Segmentation; Expectation-based Failures; Reinforcement Learning

## Introduction

The ability to derive meaning from complex observations is a skill that has been recognized as vital for "growing" an intelligent agent from a simple starting state through interactions with a complex environment (Brooks, 1995; Cohen, Oates, Atkin, & Beal, 1996). Before the agent is able to reason over a world model, it must first develop one. We present an approach in which an agent, exposed to patterns with temporal dependencies, develops a predictive model of its environment. The agent's expectation failures (i.e. prediction errors) are then used as the basis of its event segmentation mechanism. The resulting segments form the foundation of event representations.

The research we present in this paper builds on the work by Reynolds, Zacks, and Braver (2007) to build a computational model of event segmentation. We extend their model by incorporating a reinforcement learning agent to handle the detection of event boundary locations and trigger the subsequent event segmentation. The prediction mechanism is the gated-recurrent neural network (GRNN) model outlined by Reynolds et al. We evaluated several variations on the state representation presented to the reinforcement learning agent. The representations leverage information about the GRNN's prediction error through time. The first representation evaluated is a simple state representation composed of the ratio of the predictive model's current error to its average error. This simple representation is then expanded to include a measure of input change, the amount of time since the gate was last opened, and the type of event that is expected next. Each state representation we evaluated contained the prediction ratio. We tested the GRNN-RL pair and the state represations on a motion captures dataset representing people executing 13 distinct tasks. Our results support the idea that information about the GRNN's prediction error is sufficient to allow a learned RL policy to appropriately identify event boundaries.

## Motivation

People are able to unconsciously and effortlessly perceive sequences of discrete events from dynamic and continuous sensory input (Radvansky & Zacks, 2014; Ross & Baldwin, 2015). People's ability to recognize temporal structure and patterns frequently observed across environmental contexts facilitates their partitioning of continuous activities into discrete events (Elman, 1990; Cleeremans & McClelland, 1991; Cohen & Adams, 2001; Reynolds et al., 2007). Therefore, people must learn the sequential dependencies that allow them to reason about sequences of observations as single, individual events. Reasoning about both observed events and their associated spatiotemporal patterns allows humans to reason about the underlying cause of the change in sensory observations (Radvansky & Zacks, 2014). Evidence suggests that when people use an inferred event (i.e., spatiotemporal pattern) to guide their sensory expectations, they are able to recognize when transitions between events occur because their observations no longer match that of the current, hypothesized spatiotemporal pattern (Braver & Cohen, 2000; Rougier, Noelle, Braver, Cohen, & O'Reilly, 2005). We model how agents develop spatiotemporal models and use them to to interpret continuous observations as discrete events.

## Background

The task of this paper is related to previous works such as the Neo project (Cohen et al., 1996). Neo is a simulated infant that implements a computational model of the *perceptual analysis by image-schema* theory of complex concept formation (Johnson, 1987; Mandler, 1988, 1992; Lakoff & John-

son, 2008). Neo begins with relatively simple configurations and develops/learns complex concepts through interactions with a simulated, complex world, by analyzing occurrences of discrete, symbolic tokens. We agree on the importance of developing complex agents able to learn via a process of perceptual analysis, representations of objects, states, and activities as its foundation for learning conceptual categories. However, a precondition for systems such as Neo is a process for transforming continuous sensory observations of the world into meaningful, discrete units (i.e., "unitization"). Our work addresses the unitization problems.

Reynolds et al. (2007) propose a relatively simple mechanism for event segmentation. Using its experiences with the world, their mechanism refines and hones the way it segments continuous observations without prior knowledge about the events or about the locations of event boundaries. Their mechanism is an implementation of the first component of *Event Segmentation Theory* (EST), a theory of event schema/model creation (Zacks, Speer, Swallow, Braver, & Reynolds, 2007; Kurby & Zacks, 2008). While previous approaches to event segmentation focused on the degree of change between subsequent observations as the key predictive feature (Newtson, 1976; Gibson, 1979), EST emphasizes the role of prediction failures. The importance of prediction error during event segmentation is based on data suggesting that people attempt to predict what they will observe next (Rao & Ballard, 1999; Enns & Lleras, 2008; Niv & Schoenbaum, 2008).

People maintain working models, dynamic representations that facilitate event comprehension and incorporate predictions about what will be observed next, of the events they are observing (Radvansky & Zacks, 2014). Evidence suggests that working models are the result of the segmentation and chunking of experience that are triggered by transient increases in prediction error (i.e., expectation failure driven event segmentation). When an event boundary is detected, people update their working model, thus changing their expectations about what will be observed next. However, there is a key limitation in the approach taken by Reynolds et al. when implementing this process, as their system depends on externally set thresholds to determine when one event ends and another begins. The work we present here extends their prediction model by removing externally set thresholds and examining the impact of incorporating higher-level expectations.

## Modeling Prediction Error-based Segmentation

### Reynolds, Braver, and Zack's Segmentation Model

Reynolds et al. (2007) used a gated-recurrent neural network, with the architecture depicted on the left in Figure 1, to model how people might learn sequential dependencies and perceive discrete event categories from continuous observations. The GRNN identified points at which one activity transitioned into another via an expectation failure based heuristic. They selected the GRNN because they considered it the most biologically plausible model available for capturing how people might learn sequential dependencies with the ability to store a representation of the current event in memory based, on contemporary work in behavioral and neuropsychological correlates of event structure and computational studies of sequential domains (for an extensive literature review see Reynolds et al. (2007)). The GRNN adjusted its event representation by triggering a gating mechanism that allowed the event representation to be directly updated based on the GRNN's observations. The gating mechanism controlled the extent to which the event representation was updated by each new observation and, combined with the network's recurrence, allowed the GRNN to maintain representations of the events through time (Elman, 1990; Hochreiter & Schmidhuber, 1997). In their simulations, the gate was operated either: (1) by ground truth knowledge about the location of event boundaries or (2) by an externally set threshold on the ratio of the model's current sum squared error (SSE) and its average SSE. Their model attempted to predict its next observation; expectation failures were measured as SSE in the model's prediction and the true next observation. Based on the distribution of SSE observed within events versus at event boundaries, the authors concluded that a GRNN with an expectation failure-based gating mechanism is a reasonable approximation of how people might segment sequences of observations into meaningful units.

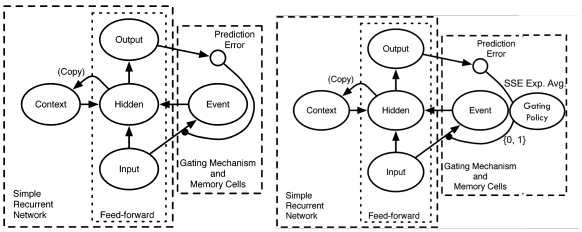


Figure 1: GRNN Model Architecture (Reynolds et al., 2007) on the left. Unsupervised Event Segmentation Model Architecture on the right.

We built on Reynolds et al.'s (2007) work by extending their GRNN to include a RL agent that learns a policy for controlling the gating mechanism.

### A New Approach to Unsupervised, Self-Regulating Event Segmentation

We incorporated a RL agent that learned a policy for controlling the gating mechanism that Reynolds et al. (2007) created for their final simulation (Simulation IIIB), with the architecture depicted on the right in Figure 1. In Simulation IIIB, the gating mechanism was controlled by a simple mathematical function that evaluated whether the ratio of the models' last observed prediction error relative to the observed average error exceeded a threshold (1.5). Before modifying Simulation IIIB to include the RL agent, we tested our implementation in order to replicate their the experimental results, and we observed the same relations between the within event and

boundary observations. Reproducing the author's results (1) allowed us to evaluate the reproducibility of their work before using it as the basis of our system and (2) allowed us to build our model on one already reviewed and evaluated by the scientific community.

In our model, the gating function from Simulation IIIB was replaced with a RL agent that learned a policy for controlling the gating mechanism. An Expected-Sarsa learning algorithm with a linear function approximator (Sutton & Barto, 2015) was used to learn the policy for controlling the gating mechanism. Our GRNN and RL combination is only able to build low-level expectations about what it will observe next. However, it could be used as a component in a larger system to build higher-level expectations that could be used to help guide the actions of the RL agent. In our experiments, while our model only directly builds expectations at the lower level, we incorporate information at higher levels of expectation in the the RL agent's state representation. We distinguish between lower and higher level information based on whether or not the information stems from the RL agent's immediately available observations about the state of the GRNN. Lower-level information is information that is readily available to the RL agent, whereas higher level information is not immediately available. For instance, the degree to which there is change between two subsequent observations is readily observable, whereas knowledge about the likelihood of a the next event transition being from sitting to standing is not.

In our experiments, the combined GRNN and RL agent was presented with a sequence of frames of motion captures of activities carried out by people (i.e. sitting,standing,jumping,etc.). Each activity constituted a single event and contained some number of frames. The frames are what the GRNN observed. The experiments varied the information presented to the RL agent. The RL agent's state representations consisted of both lower and higher level information about the state of the GRNN and RL agent. The lower-level information included a measure of the GRNN's prediction error (as in Reynolds et al. (2007)), the degree of change in the system's subsequent observations (inspired by Newtson (1976); Gibson (1979)), and the amount of time since the RL agent last updated the event representation. The higher level information included was a representation of the next event the agent expected to observe.

## The Models

The GRNN was constructed with the same parameters used by Reynolds et al.: 54 input units, 100 hidden units, 100 event units, 100 recurrent units, and 54 output units. The input and the hidden units had sigmoidal activation functions. The weights were initialized randomly within the range $[-0.5, 0.5]$ and during back-propagation the learning rate was 0.001. When comparing our implementation of the GRNN to that of Reynolds et al. we trained it to asymptotic performance, roughly $20,020$ events, and evaluated it on 900 events. For further specifics about how the GRNN was configured, please refer to details about Simulation IIIB in Reynolds et al. (2007). The GRNN was trained on $50,000$ events with a perfect gating signal prior to incorporating the RL agent as the gating mechanism.

The RL agent was construction according to an $\varepsilon$-greedy Expected-Sarsa with replacing traces policy learning algorithm. The specific state representations the agent learned to operate over can be seen in the experiments section below. Each of the state representations contained at least one continuous feature, therefore a linear function approximator was used to estimate the value of each state-action pair. Tile coding was used to convert the continuous states into binary feature vectors consisting of 32 layers of tilings with 4 tiles for each feature.

The agent had two possible actions: (1) flip the gate and (2) do not flip the gate. The policies were learned according to $-SSE$ computed from the SSE observed in the GRNN's predictions after each action by the RL agent. We chose this reward, because it allows for unsupervised to control the gating mechanism and it is aligned with event segmentation theory (Radvansky & Zacks, 2014). Table 1 shows the learning parameters used to learn the policies for each of the state representation experiments.

## Experiments

Experiments evaluated the performance of the RL agent at detecting when the GRNN's event representation should be updated. In each experiment, the RL learning algorithm described above was evaluated according to the quality of the policy it was able to learn given the different state representations. The different state representations incorporated different amounts of low and high-level information. The low-level information described the state of the GRNN and the state of the RL agent. The high-level information described expectations about the next event that would be observed. The RL agent learned over the course of 2,000 episodes. During each episode, the RL agent was exposed to 100 randomly ordered events. For the first 20 events, a perfect gating signal was used before the RL agent began learning. This allowed a reasonable average SSE to be computed before it was used as part of the RL agent's state representation. An overview of the experimental state representations and the learning parameters used by the RL agent to learn a policy for the given state representation can be seen in Table 1.

Each state representation consisted of between 1 and 4 features and always contained a feature describing the GRNN's current prediction error with respect to its historical prediction error, i.e. SSE_Ratio. Each dimension represented different information about the state of the overall system (Table 2):

- SSE_Ratio - the GRNN's current prediction error (i.e. SSE) with respect to a windowed average of the GRNN's historical SSE;
- Obs_Dist - the euclidean distance between two subsequent observations, $X_{t-1}$ and $X_t$;

Table 1: State Representations and Learning Parameters.

| State Representation | Learning Parameters |
|---|---|
| SSE_Ratio | ($\alpha$=0.005;$\gamma$=0.95;$\lambda$=0.9) |
| SSE_Ratio+Obs_Dist | ($\alpha$=0.005;$\gamma$=0.9;$\lambda$=1.00) |
| SSE_Ratio+Next_Event | ($\alpha$=0.001;$\gamma$=0.9;$\lambda$=0.75) |
| SSE_Ratio+Last_Gate | ($\alpha$=0.005;$\gamma$=0.95;$\lambda$=0.8) |
| SSE_Ratio+Obs_Dist+Last_Gate | ($\alpha$=0.005;$\gamma$=0.9;$\lambda$=0.8) |
| SSE_Ratio+Obs_Dist+Next_Event | ($\alpha$=0.001;$\gamma$=0.85;$\lambda$=0.95) |
| SSE_Ratio+Last_Gate+Next_Event | ($\alpha$=0.005;$\gamma$=0.95;$\lambda$=0.95) |
| SSE_Ratio+Obs_Dist+ Last_Gate+Next_Event | ($\alpha$=0.005;$\gamma$=0.9;$\lambda$=0.9) |

- Last_Gate - the distance between the system's current time step, $t$, and the last time step at which the RL agent's action was to opened the gate and update the event representation, $t_{a=1}$;

- Next_Event - the next event the model will observe.

The mechanism by which a system might build higher level expectations is not the focus of this research. Therefore, a perfect version of this mechanism is used in our experiments. Taking this approach is in line with the style of experiments completed by Reynolds et al. (2007) during Simulation IIIA; additionally, it allows for evaluating the performance of the RL agent as the GRNN's gating mechanism and evaluating the benefits higher level expectations can have for lower-level components, without having to tease apart the impact of the performance of the higher level reasoning component.

Table 2: State Representation Features.

| Feature | Definition |
|---|---|
| SSE_Ratio | $\frac{SSEt}{ape_{t-1}}$; $ape_t = ape_{t-1} + 0.05(SSEt - ape_{t-1})$ |
| Obs_Dist | $\sqrt{\sum_{i=1}^{53}(X_t[i] - X_{t-1}[i])^2}$ |
| Last_Gate | $t - t_{a=1}$ |
| Next_Event | $E_{i+1}$ |

For each condition described in Table 1, the RL agent learned its policy by interacting with the GRNN as it operated over a sequence of 20,020 randomly ordered events. The RL agent learned its policy over the course of 500 episodic passes over the event sequence. The performance of each learned policy was evaluated over 50 separate runs where a new randomized sequence of events was generated for each run. The performance of the policy learned for each state representation is described below in Results. The above experiments were run for the $-SSE$ and the distance-based reward functions described in Models above.

## The Data

The training data set for the GRNN and, subsequently, the RL agent was the motion capture data used by Reynolds et al. (2007), which can be found at `http://dcl.wustl.edu/stimuli.html`.

The data set contains 3-dimensional motion captures of people performing 13 distinct tasks. Each motion capture lasted 3-4 seconds and contained between 10 and 13 observations. Each motion capture activity was considered to be one event. Each event observation consisted of 18 (x, y, z) points on the body. We preprocessed each observation following Reynolds et al. (2007); the origin of the coordinate frame was transformed such that the points corresponding to person's hip was the origin, all values were scaled to the range $[-1,1]$, and the orientation of each figure was altered such that it was the same across all events.

Following Reynolds et al. (2007), before each training run for the GRNN or the RL agent, the training set was created by randomly ordering the events from the set of 13 events. A new event was randomly selected and added to the training set until the GRNN reached asymptotic performance. This allowed the GRNN and the RL agent to observe each event multiple times and learn a good predictive model for the frames that fell within a given event. The random ordering of the events provided the learning algorithm with a large variety of transition examples. The same process was used to create the training set of the RL agent, but with a stopping condition of the combined GRNN and RL agent having observed a prespecified number of events.

## Results

The results show that it is possible to use reinforcement learning to identify true event boundaries. Furthermore, it is possible to learn a policy for controlling the GRNN gating mechanism without encoding any knowledge within the reward function about where event boundaries actually exist. This is important, because it provides evidence demonstrating that it is possible for an artificial agent to take the first steps towards learning complex concepts using a bottom-up approach. Additionally, it provides evidence that it is possible for an artificial agent to learn on its own without requiring the painstaking process of handcoding thresholds and decision boundaries on the part of a human.

Table 3 shows the results from training the RL agent with the eight different state representations. *Dist.* describes the average distance between when the RL agent chose to update the event representation and the closest true event boundary. *Reward* is the total reward received by the agent during the episode. *Err.* describes the GRNN's average SSE over the course of the episode. Each value in Table 3 is averaged across 50 independent runs.

For each state representation, it was possible to learn a policy by which the RL agent could control the GRNN's gating mechanism. The learning curves for each state representation can be seen in Figure 2. Each of the state representations was

Table 3: Experimental results after 1000 episodes averaged over 50 runs.

| State | Dist. | Reward | Err. |
|---|---|---|---|
| SSE_Ratio | 0.2 | −12.0 | 0.15 |
| SSE_Ratio+Obs_Dist | 0.13 | −30.72 | 0.18 |
| SSE_Ratio+Next_Event | 0.52 | −31.44 | 0.18 |
| SSE_Ratio+Last_Gate | 1.22 | −65.78 | 0.38 |
| SSE_Ratio+Obs_Dist+Last_Gate | 1.16 | −42.31 | 0.25 |
| SSE_Ratio+Obs_Dist+Next_Event | 0.76 | −48.69 | 0.28 |
| SSE_Ratio+Last_Gate+Next_Event | 1.1 | −103.07 | 0.3 |
| SSE_Ratio+Obs_Dist+ Last_Gate+Next_Event | 0.32 | −52.56 | 0.5 |

able to achieve an average reward between 0.15 -0.5 over the course of 1000 episodes. When run using a perfect gating function, the GRNN is able to achieve an average prediction error within the same range achieved by the RL agent. It is of note that this improvement in performance over that reported by Reynolds et al. (2007) is due, in part, to advances in deep neural network computing libraries. The state representation SSE_Ratio+Obs_Dist was able to learn a policy best able to maximize its received rewards. This indicates that information about the ratio of the current SSE to the average observed SSE and the degree of change between two subsequent observations are critical features for deciding how and when to update the GRNN's event representation.

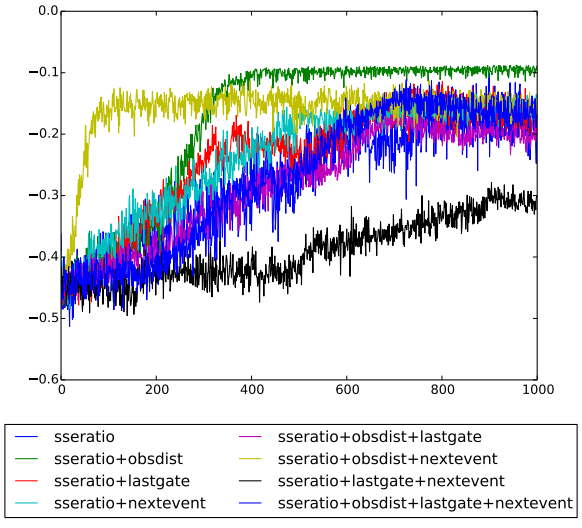


Figure 2: GRNN Model Architecture (Reynolds et al., 2007).

The results show that the RL agent was able to learn a policy for each state representation. The agent was able to reach a reasonable average distance from the true event boundary, approximately 1.5 frames for about half of the state representations. We considered the RL agent's ability to identify an event boundary within 0.67 of the true event boundary to be a reasonable level of performance given that each event lasts for 11 frames on average.

Finding that it is possible for a RL agent to learn when one event ends and another begins using GRNN's the SSE_Ratio alone, while surprising, is encouraging, as an initial step towards learning to unitize continuous observations without the use of higher level information (i.e. Next_Event). It is possible that the SSE_Ratio feature was so powerful on its own because it is correlated with and related to the other lower-level features (i.e. Obs_Dist and Last_Gate). However, it is not surprising that the SSE_Ratio+Obs_Dist both resulted in a high performing policy and the best performaning policy given the evidence in the literature suggesting that the degree of change between two subsquent observations plays a large role in segmenting continuous events and detecting boundary points on physical objects (Newtson, 1976; Gibson, 1979) is considered.

The ability to correctly identify event boundaries does not always have a consistent effect on the GRNNs observed prediction error. This finding indicates that it is not the number of boundaries that are correctly identified that is important, but rather which boundaries are correctly identified. Appropriately handling sub-event boundaries could drive down error in the GRNN while causing the RL agent to trigger gates at non-event boundaries, thus increasing the average distance measure. For example, the SSE_Ratio state representation results in an agent that is better able to detect event boundaries than the SSE_Ratio+Next_Event state represention, but the SSE_Ratio+Next_Event results in more rewards and lower average prediction errors in the GRNN.

## Future Work

That the RL agent was able to learn a policy for controlling the gating mechanism based solely on the GRNN's prediction error supports the potential for prediction error to play a primary role in event segmentation. We hypothesize that prediction error is likely to play an important role in other aspects of complex event segmentation and, possibly, event cognition. For example, if a person is unable to predict the event he/she will observe, then the higher level expectation failures might be propagated back to the segmentation gating mechanism and alter how it is identifying event boundaries. The ability of the agent to learn a gate controlling policy given a state representation that includes higher level expectations (i.e. the likelihood that the current event will transition into a standing event), indicates that the combined GRNN and RL agent model should be able to segment continuous observations into discrete events such that the discrete events are maximally predictive based on higher level expectation errors. We intend to study this in future work.

The research in this paper represents a step towards modeling how an intelligent agent can reason about and manipulate its model of the world in order to develop meaningful representations in an unsupervised way. Given that our system can recognize event boundaries, the next step is to develop a

system that is able to recognize sub-events. We believe that extending our approach to learning a policy for segmenting an event into sub-events will depend on two parts: (1) allowing the RL agent to more finely control the amount of influence each observation has on the subsequent event representation and (2) learning prototypical representations of the events. Giving the RL agent more fine-grained control over the influence incoming observations have on the current event representation should allow the agent to account for sub-events within longer, more complex events. Additionally, by learning the sequential dependencies among the event representations, it should be possible to go beyond identifying event boundaries to predicting which event will be observed next. For example, given an observation that a person is currently seated, represented in the form of the GRNN+RL agents event representation and the learned prototypical agent, it should be possible to predict the likelihood that the person will stand up.

## Conclusion

This paper has presented a model for learning to segment continuous observations into event units. Additionally, our model is able to learn to identify boundary points without any prior knowledge. The combined GRNN and RL agent proposed in this paper represents an approach to modeling event segmentation that removes the limitation of externally set thresholds and is able to operate in continuous domains. Experimental results support our conclusion that it is possible to use RL to learn a gate controlling mechanism that is able to accurately identify event boundaries independently without incorporating knowledge about the location of event boundaries in the reward function.

## References


Braver, T. S., & Cohen, J. D. (2000). On the control of control: The role of dopamine in regulating prefrontal function and working memory. *Control of cognitive processes: Attention and performance XVIII*, 713–737.

Brooks, R. A. (1995). Intelligence without reason. *The artificial life route to artificial intelligence: Building embodied, situated agents*, 25–81.

Cleeremans, A., & McClelland, J. L. (1991). Learning the structure of event sequences. *Journal of Experimental Psychology: General*, *120*(3), 235.

Cohen, P., & Adams, N. (2001). An algorithm for segmenting categorical time series into meaningful episodes. In *International symposium on intelligent data analysis* (pp. 198–207).

Cohen, P., Oates, T., Atkin, M. S., & Beal, C. R. (1996). Building a baby. In *Proceedings of the eighteenth annual conference of the cognitive science society* (pp. 518–522).

Elman, J. L. (1990). Finding structure in time. *Cognitive science*, *14*(2), 179–211.

Enns, J. T., & Lleras, A. (2008). What's next? new evidence for prediction in human vision. *Trends in cognitive sciences*, *12*(9), 327–333.

Gibson, J. J. (1979). *The ecological approach to visual perception: classic edition*. Psychology Press.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Johnson, M. (1987). The body in the mind: The bodily basis of imagination, reason, and meaning. *The body in the mind: the bodily basis of imagination, reason and meaning*.

Kurby, C. A., & Zacks, J. M. (2008). Segmentation in the perception and memory of events. *Trends in cognitive sciences*, *12*(2), 72–79.

Lakoff, G., & Johnson, M. (2008). *Metaphors we live by*. University of Chicago press.

Mandler, J. M. (1988). How to build a baby: On the development of an accessible representational system. *Cognitive Development*, *3*(2), 113–136.

Mandler, J. M. (1992). How to build a baby: Ii. conceptual primitives. *Psychological review*, *99*(4), 587.

Newtson, D. (1976). Foundations of attribution: The perception of ongoing behavior. *New directions in attribution research*, *1*, 223–247.

Niv, Y., & Schoenbaum, G. (2008). Dialogues on prediction errors. *Trends in cognitive sciences*, *12*(7), 265–272.

Radvansky, G. A., & Zacks, J. M. (2014). *Event cognition*. Oxford University Press.

Rao, R. P., & Ballard, D. H. (1999). Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience*, *2*(1), 79–87.

Reynolds, J. R., Zacks, J. M., & Braver, T. S. (2007). A computational model of event segmentation from perceptual prediction. *Cognitive Science*, *31*(4), 613–643.

Ross, R. A., & Baldwin, D. A. (2015). Event processing as an executive enterprise. In *Emerging trends in the social and behavioral sciences.* John Wiley and Sons, Inc.

Rougier, N. P., Noelle, D. C., Braver, T. S., Cohen, J. D., & O'Reilly, R. C. (2005). Prefrontal cortex and flexible cognitive control: Rules without symbols. *Proceedings of the National Academy of Sciences of the United States of America*, *102*(20), 7338–7343.

Sutton, R. S., & Barto, A. G. (2015). *Reinforcement learning: An introduction* (Vol. 1) (No. 1). MIT press Cambridge.

Zacks, J. M., Speer, N. K., Swallow, K. M., Braver, T. S., & Reynolds, J. R. (2007). Event perception: a mind-brain perspective. *Psychological bulletin*, *133*(2), 273.