Cognitive Science 31 (2007) 613–643

A Computational Model of Event Segmentation From Perceptual Prediction

Jeremy R. Reynolds,[1,*] Jeffrey M. Zacks,[2] Todd S. Braver[2]

[1] *Department of Psychology, Washington University*
[2] *Departments of Psychology and Radiology, Washington University*

Received 23 September 2005; received in revised form 31 May 2006; accepted 9 June 2006

Abstract

People tend to perceive ongoing continuous activity as series of discrete events. This partitioning of continuous activity may occur, in part, because events correspond to dynamic patterns that have recurred across different contexts. Recurring patterns may lead to reliable sequential dependencies in observers' experiences, which then can be used to guide perception. The current set of simulations investigated whether this statistical structure within events can be used 1) to develop stable internal representations that facilitate perception and 2) to learn when to update such representations in a self-organizing manner. These simulations demonstrate that experience with recurring patterns enables a system to accurately predict upcoming stimuli within an event, to identify boundaries between such events based on transient increases in prediction error, and to use such boundaries to improve prediction about subsequent activities.

Keywords: Event perception; Prediction; Connectionist model; Knowledge structures

We live in a complex world that is defined by change, and we interact with objects and people that follow intricate trajectories through space and time. How do human perceptual systems make sense of the dynamic complexity characteristic of everyday activities such as baseball games, classes, and cooking? Adaptive behavior in dynamic environments would seem to require information processing mechanisms that can identify the relevant spatiotemporal patterns underlying sensory change. Research in psychology, artificial intelligence, and neuroscience has focused on characterizing the perception of spatial structure in the environment, leading to relatively mature cognitive-computational theories of object recognition and scene perception. Although temporal structure may be equally important, much less scientific attention has been given to this problem.

*Present address: Department of Psychology, University of Colorado, Boulder, CO, USA.
Correspondence should be addressed to Jeremy R. Reynolds, Department of Psychology, University of Colorado, 345 UCB, Boulder, CO 80309. E-mail: jeremy.reynolds@colorado.edu

There is good reason to believe that people can utilize temporal structure during perception, and that this affects their memory for events. Computational and psychological studies of word learning show that people can use sequential structure in a speech stream to learn words (e.g., Brent, 1999; Elman, 1990; Saffran, Newport, Aslin, Tunick, & Barrueco, 1997), and studies of narrative memory and artificial intelligence suggest that readers depend on knowledge structures that encode the temporal organization of everyday activities (Schank & Abelson, 1977; Zwaan & Radvansky, 1998). Yet, despite these examples, we know very little about how people encode the temporal structure of everyday activity as it unfolds. In particular, we have a poor understanding of a basic question about perception: How does a person come to perceive that one meaningful event has ended and another has begun? In this set of studies, we present computational simulations that investigate how such a perception may by a self-organizing system. Before turning to the model, we first review the current state of the field in terms of extant experimental data and previous computational approaches.

1. Behavioral and neuropsychological correlates of event structure

Newtson (1973) introduced a simple and powerful method to measure people's perception of event structure. This method requires participants to watch movies of everyday activities (such as a person doing a load of laundry) and to press a button to indicate that one event has ended and another has begun. Using this method, researchers have demonstrated that boundaries between events demonstrate good consistency across participants (Newtson, 1976; Zacks, Tversky, & Iyer, 2001b) and good test-retest reliability within the same participant (Newtson, 1973; Speer, Swallow, & Zacks, 2003). These data provide direct evidence that observers can track the temporal structure of ongoing activity in real time.

The fact that observers can segment activity into meaningful parts does not imply that they do so during normal perception. However, data from functional neuroimaging studies have suggested that a network of cortical regions responds selectively to perceptual boundaries during passive viewing of events. In one experiment (Zacks, Braver, Sheridan, Donaldson, Snyder, Ollinger, Buckner, & Raichle, 2001a), participants passively viewed movies of everyday activities while local brain activity was recorded with functional MRI (fMRI). The segmentation procedure was then explained, and they segmented the same movies into both large and small units. Each individual's segmentation data were used to estimate the brain response evoked by event boundaries during their earlier passive viewing of the movies. The participants were naive to the segmentation procedure during the initial fMRI recording, so they could not have been covertly performing the task. During passive viewing, a network of brain regions increased transiently in activity at the points each observer later identified as event boundaries. This network included a set of posterior occipital, temporal, and parietal regions, and right dorsal frontal cortex. A subsequent study confirmed that the posterior activity included the human MT complex (MT+, an area specialized for processing motion) (Speer, Swallow, & Zacks, 2003). A recent electrophysiological study in monkeys provides converging results. In a different paradigm in which monkeys performed a sequentially structured perceptual task, neurons within dorsal frontal cortex showed selective responses at event boundaries (Fujii & Graybiel, 2003).

Whereas the fact that observers can and do segment activity is well established, the means by which they do so are less well understood. Early evidence indicated that when viewers segment human activity the identification of unit boundaries is related to changes in the actor's body position (Newtson, Engquist, & Bois, 1977). More recent research provides concrete evidence that perceptual event segmentation can be predicted from low-level movement cues under some circumstances (Hard, Tversky, & Lang, 2006), particularly when the activity being segmented appears to observers not to be goal-directed (Zacks, 2004).

This evidence supports three conclusions about perceptual segmentation of ongoing activity. First, humans can segment everyday activities reliably. Second, event segmentation is a normal part of perception. Third, under some circumstances observers may use low-level movement cues as a basis for segmentation. However, a number of central questions remain unanswered: What causes the cascade of neurocomputational processes that result in the perception that one event has ended and another begun? How does the neurocomputational system for event segmentation self-organize? and What implications does the structure of the event segmentation system have for how event knowledge is internally represented in the mind/brain of human perceivers? Satisfactory answers to such questions likely will require formal theoretical models that address the computational mechanisms subserving online event perception. Specifically, we hypothesize that successful event segmentation may involve the extraction and representation of sequential structure, and further, that boundaries between contiguous events may be defined on the basis of violations in such sequential structure, such that boundaries are identified when humans perceive something unpredictable. Whereas there have been several notable attempts to model structured sequential action (Botvinick & Plaut, 2004; Cooper & Shallice, 2000), there have been only limited attempts to develop a computational account of how sequential event structure is extracted (e.g., perceived) from ongoing activity (Cohen, 2001; Hanson & Hanson, 1996). However, there does exist a larger body of computational research examining sequential structure in other temporal domains, such as language.

2. Computational studies of sequential domains

The last 15 years have seen significant progress in the development of computational models in certain temporal domains, most notably language comprehension. Several different approaches have been used to investigate the question of how humans process sequential information. One set of approaches, primarily used by scientists in the computer vision field, has focused on using hidden Markov models (Bregler, 1997) or motion templates (Zhao & Nevatia, 2002) in order to group sequences of pictures into meaningful units. Similar computational techniques have been used to segment arbitrary letter sequences into meaningful episodes (Cohen & Adams, 2001). Whereas these approaches have focused on computational optimization, researchers using more biologically plausible *connectionist* (or *neural-network*) models (Elman, 1991; St. John & McClelland, 1990) have attempted to capture how human cognitive systems might learn sequential structure. A typical task given to such models is a simple one-step prediction task: based on the current input at time t, predict the input to appear at the next time $t + 1$. Two important conclusions have emerged from this research. First, sequential structure can be critical for predicting a near-future state of the environment from

the current state. Second, computational systems that learn to internally represent important sequential structure must have some form of memory. That is, there must be some internal mechanism present that can store previous states, such that they can serve as context for interpreting the current state of environmental input.

The most popular approach utilized for studying sequential problems in a connectionist framework has been to employ a computational architecture known as a simple recurrent network. Simple recurrent networks are similar to standard 3-layer feed-forward neural networks, which contain layers representing the input, an intermediate processing stage (*hidden* layer), and the output, each of which is connected serially. In addition, simple recurrent networks have a *context* layer that maintains a copy of the prior state of the hidden layer. This context layer provides the system with the capability for storing previous states. Because of their architectural simplicity, simple recurrent networks have been the architecture of choice for most research examining learning in sequential domains. Simple recurrent networks have been shown to have substantial computational capabilities for learning sequential dependencies (Cleeremans & McClelland, 1991; Cleeremans, Servan-Schreiber, & McClelland, 1989; Elman, 1990), but they also have a number of well-known limitations. In particular, simple recurrent network-based architectures struggle when faced with environments involving long-term temporal contingencies. Although more complex architectures can be employed (e.g., standard recurrent neural networks), such networks are notoriously difficult to train because standard learning algorithms such as back-propagation are not well-suited for solving the *temporal credit assignment problem*: determining at what point in time an internal representation was inappropriate for the task at hand (Bengio, Simard, & Frasconi, 1994; Hochreiter & Schmidhuber, 1997).

A recently developed architecture that provides a solution to this problem is the gated recurrent network (Hochreiter & Schmidhuber, 1997). In such networks, recurrent layers (which maintain context information) are protected from input signals via a learned gating mechanism, such that context representations are only updated when the gate is open. This mechanism provides more robust storage of context while reducing the complexity of the learning task. This gated context mechanism is precisely the formal analog to current conceptual hypotheses regarding event perception (Zacks, Speer, Swallow, Braver, & Reynolds, 2005): While a human perceives activity within an event, an active set of knowledge structures (an *event representation*) is maintained and influences on-going processing and perception. When that event ends and a new one begins, a new set of representations becomes activated and displaces the old set. The mechanism underlying the updating of such event representations is hypothesized to be the same mechanism by which boundaries are identified: a transient increase in prediction error associated with boundaries between events opens the gate and allows the current event representation to be updated. If the hypotheses regarding the relationship between sequential dependency and event structure are true, then this unsupervised, transient increase in prediction error would serve as an ideal gating signal to update internal representations. In fact, similar signals associated with a violation of expectations have been used in attempts to parse streams of auditory input into either chords (Gjerdingen, 1992) or words (Grossberg, 2003).

In addition to having attractive computational properties, the hypothesis that prediction error serves as a gating mechanism is consistent with previous work suggesting that such

a gating mechanism is intimately tied to the appropriate functioning of the dopamine (DA) neuromodulatory system and the prefrontal cortex (PFC), such that phasic DA signals could provide reliable updating signals to PFC (Braver & Cohen, 2000; Rougier, Noelle, Braver, Cohen, & O'Reilly, 2005). The link between a gating mechanism and its potential underlying substrates is relevant for two primary reasons. First, previous work has suggested that prefrontal cortex, particularly the right hemisphere, may actively represent events: Knowledge about events in general and the contents of current events depends on structures in the PFC (Allain, Gall, Etcharry-Bouyx, Aubin, & Emile, 1999; Grafman, 1995; Schwartz, Montgomery, Fitzpatrick-DeSalme, Ochipa, Coslett, & Mayer, 1995; Sirigu, Zalla, Pillon, Grafman, Dubois, & Agid, 1995; Sirigu, Zalla, Pillon, Grafman, Agid, & Dubois, 1996).

Second, there is growing evidence suggesting that DA (and potentially norepinephrine) may represent differences between internal and external states. These systems are thought to broadcast error signals—including prediction errors—through widespread projections to the cortex. For example, DA neurons in the substantia nigra and ventral tegmental area are sensitive to differences between actual and predicted rewards (Schultz, 1998). Moreover, DA effects on PFC neurons are clearly modulatory, and may be multiplicative in nature, consistent with a gating function (Braver & Cohen, 2000; Cohen, Braver, & Brown, 2002).

Thus, converging evidence from neuroscience and computational studies supports the idea that a prediction-error-based gating system may operate in the brain to update actively maintained event representations at appropriate junctures. In the current work, we applied these ideas in developing a computational architecture for event segmentation that utilized prediction error signals in a gated recurrent network.

The simulations described in the following sections applied gated recurrent networks to the perception of event structure. In particular, it was hypothesized that representations of events depend upon stably maintained internal context representations, and that both the learning and updating of these representations occurs within an integrated gating-learning system that is driven by prediction errors. The critical theoretical claim is that during event perception prediction errors will increase transiently at the boundaries between events, providing a natural gating signal. These spikes in prediction error occur as a natural consequence of the sequential structure inherent in the environment, and they constitute an unsupervised mechanism by which the system can learn about event structure.

To preview the results, the current set of simulations provides evidence for five features of the hypothesized relationship between prediction error and event perception:

1. Prediction error is greater during event boundaries relative to time points within an event.
2. Stable contextual information improves the prediction of event sequences.
3. A network can use gating signals occurring at event boundaries to learn and update internal context representations that reflect event knowledge.
4. A network can use gating signals based on prediction error to reliably update internal context representations that carry event information and facilitate subsequent prediction.
5. A network can use gating signals based on prediction error to learn internal context representations for events.

This set of simulations provides strong initial support for the hypothesis that a network can use prediction error to self-organize event representations and, consequently, that prediction

error may form the basis of how humans spontaneously segment continuous activity into meaningful units.

3. General methods

In all simulations reported here, models were trained to perform a one-step prediction task (Elman, 1991). In this task, the model uses the current input ($input_t$) to predict the subsequent environmental input ($input_{t+1}$). All differences discussed were statistically significant at the $\alpha = 0.01$ level. All simulations were run using the bp++ framework in the PDP++ simulation software (O'Reilly, Dawson, & McClelland, 2005).

3.1. Environment

The set of inputs was composed of 3-dimensional motion captures of individuals performing 13 routine tasks, each of which was 3 to 4 s long. These formed the *events* about which the network learned during training (see Appendix A). These motion captures were provided via an animation software package (LifeForms, Vancouver, Canada). The activities were originally captured with a 30 Hz sampling rate and subsampled to 3 Hz for use in the current simulations, resulting in events that lasted between 9 and 12 frames. The 3 Hz sampling rate was used because it balances a trade-off between capturing empirical data suggesting that perceptual variables such as velocity and acceleration are associated with event boundaries (Zacks, 2004), and a concern that boundaries are being sampled from a unique distribution of perceptual changes that is discontinuous from the distribution associated with within-event frames (see Simulation I results). Each input provided to the network consisted of the positions of 18 points of the body in 3-dimensional space, and therefore, each input was represented as a 54 (18 * 3) dimensional vector (see Fig. 1). The sets of inputs used in all simulations, as well as the actual simulations, can be found online at http://dcl.wustl.edu/stimuli.html.

Several steps of pre-processing were applied to these motion captures prior to use in the simulations. First, the coordinate frame of reference of each frame was transformed such that its origin was at the center of the hips of the captured figure (to remove translations across

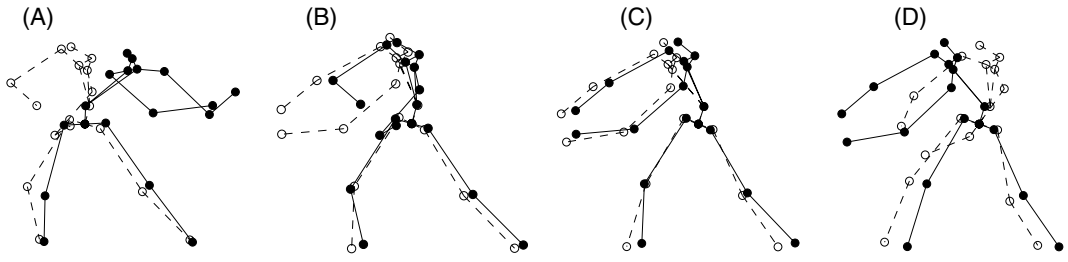

Fig. 1. Graphical representation of the model's input and target output. A–D show four consecutive frames of "chopping down a tree." The target output (dashed lines) on each frame is the model input (solid lines) on the subsequent frame. Frames C and D have similar inputs but dissimilar target outputs, illustrating the need to represent long-term sequential dependencies to achieve accurate prediction.

frames and any differences across events). Second, the figures were scaled to be within the range $\{-1,1\}$ by dividing all points by the largest absolute deviation from origin along any of the three axes across all time-points. Finally, the motion captures were processed to ensure that the orientation of each figure (defined by the vector from the left side to the right side of its hips) was, on average, the same across events. These preprocessing measures were performed in order to eliminate extraneous cues differentiating the events that could have influenced the models' performance.

The networks were trained and tested using a continuous presentation procedure. At the end of each event, a new event was randomly sampled (with replacement) from the pool of events and presented in its entirety to the network. Event presentation was continuous in that the model attempted to make a prediction for every input, such that it attempted to predict the first frame of a new, randomly selected event based on the last frame of the previous event. These frames were considered boundaries between events. All other frames were considered to occur within an event. Each network saw each event multiple times over the course of its training.

3.2. Simulation details

Each model consisted of the same basic structure, with additional components where noted below (see Fig. 2).

The core structure consisted of input (54 units), hidden (100 units), and output (54 units) layers that were fully connected in a feed-forward fashion. The input and hidden units had sigmoidal activation functions, with the activation level of each unit, act_i, determined by the equation:

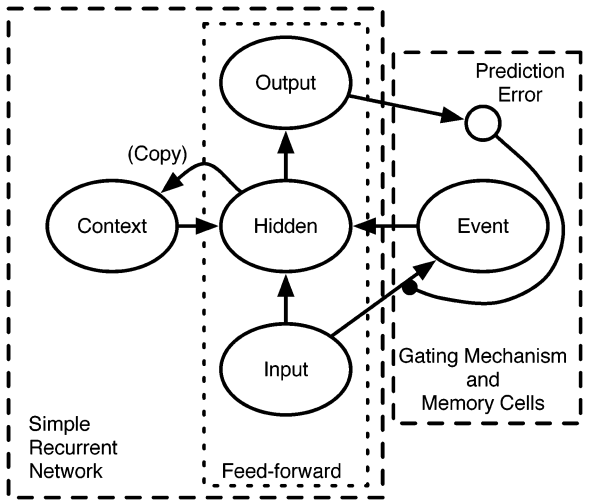

Fig. 2. Architecture of the Model. A feed-forward model was augmented both by a simple recurrent network architecture (Elman, 1991) and by a group of memory cells (Hochreiter & Schmidhuber, 1997) that can maintain information for extended periods of time while still being updated appropriately. The mechanism by which these cells are updated is a transient increase in prediction error.

$$act_{\mathrm{i}} = \frac{1}{(1 + e^{-0.4 \cdot net_{\mathrm{i}}})} \tag{1}$$

where the net input of unit i (net_{i}) was calculated as the sum of the product of the activities of the j units, act_{j}, that project to unit i and the strength of their respective connections (w_{ij}):

$$net_{\mathrm{i}} = \sum_{\mathrm{j}} w_{\mathrm{ij}} act_{\mathrm{j}} \tag{2}$$

The relatively small gain on the activation function (0.4) was used to make the activation function of these sigmoidal units nearly linear over a wide spectrum of input values. The output units had a linear activation function ($act_{\mathrm{i}} = net_{\mathrm{i}}$). This activation function was used rather than the sigmoidal activation function in order to facilitate the use of the sum-squared error (SSE) function in predicting continuous values. SSE was defined as:

$$\mathrm{SSE} = \sum_{\mathrm{i}} (act_{\mathrm{i}} - Target_{\mathrm{i}})^2 \tag{3}$$

where act_{i} is the activity level of unit i in the output layer, $Target_{\mathrm{i}}$ is the value of unit i in the input layer on the next time step, and the sum is carried out over all units in the output layer. The performance of each network was optimized by minimizing the SSE between the predicted and target outputs through back-propagation of the error gradient (Rumelhart & McClelland, 1986). All models had access to error information for only the current input (e.g., the error-gradient was not propagated through time). Although this algorithm is not biologically plausible, it is reasonable to assume that the brain has some capacity for error-driven learning, and we take back-propagation to represent this learning mechanism. The only modification to the basic back-propagation algorithm was the inclusion of tolerance in the computed error, such that an output value within 0.01 of the target value was considered to have an error value of 0; this tolerance was included to insure that weights did not grow infinitely large during training. For each replication of each network, the weights of all connections were initialized by randomly sampling from a uniform distribution with minimum and maximum values of -0.5 and 0.5. These weights were then modified by the back-propagation algorithm at each subsequent time-step with a learning rate of 0.01. Networks saw each event multiple times within each training run, and were trained to asymptotic performance (20,020 events). The stopping point was selected on the basis of visual inspection of the learning curves (e.g., SSE as a function of number of events that the network has seen). After training, each network was tested by presenting 900 events with the learning rate set to 0. The number of events during testing was selected to insure that all events, and all transitions between events, were seen during the testing procedure. All data were produced by training 20 replications of each implemented model. Additional modifications to this basic structure are indicated below.

4. Simulation I—Does prediction error identify boundaries?

The primary goal of Simulation I was to determine whether increased prediction error was associated with event boundaries in a feed-forward network.

4.1. Methods

This simulation used a feed-forward architecture (see Fig. 2) constructed exactly as specified in the general methods section. This model was the simplest type of processing that was investigated in the current set of simulations, and it served as the baseline for all subsequent comparisons.

4.2. Results

After training, the feed-forward network performed the prediction task with a mean SSE across all frames of 0.44. This SSE value is relatively small, as the number of output units (54) and the dynamic range of their activation values (−1 to 1) allowed for a possible SSE range of 0 to 108. To illustrate the performance of the feed-forward architecture, Fig. 3 illustrates a set of target and output values produced by the model.

Critically, within-event frames had an SSE approximately one fourth that of boundaries (mean within-event SSE = 0.35, mean boundary SSE = 1.33, see Table 1), suggesting that event boundaries are associated with increased prediction error, $t(19) = 106$, $p < 0.001$.

4.2.1. Predictors of SSE

One potential concern is that the difference in SSE between within-event and boundary frames could be due to an artificial difference between event-boundaries and within-event time points in the training environment. Specifically, SSE may be related to the dissimilarity

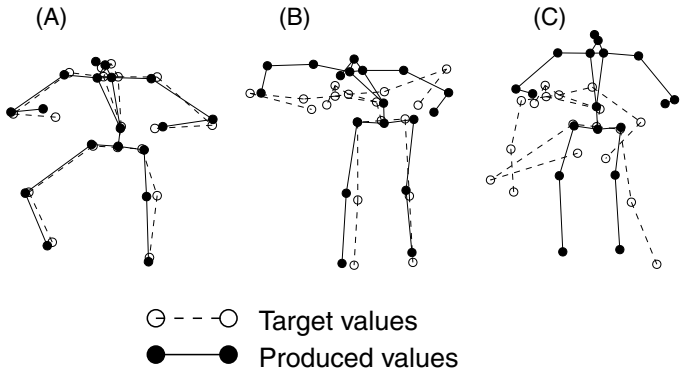

Fig. 3. Performance of the feed-forward architecture. Panel A is a graphical representation of the performance of the feed-forward architecture on its best frame (the ninth frame of "drinking:" SSE = 0.027). Panel B represents the performance of the feed-forward architecture on an average frame (the ninth frame of "spilling a drink:" SSE = 0.440). Panel C represents the model's performance on its worst frame (the boundary between "driving" and "sawing:" SSE = 3.733).

Table 1
Mean SSE and SD for different network architectures

Network Architecture	Mean SSE		Standard deviation	
	Within-event	Boundary	Within-event	Boundary
Feed-Forward	0.352	1.326	0.007	0.036
Event Knowledge	0.272	1.327	0.005	0.029
Perfect Gate	0.274	1.293	0.007	0.024
Prediction Error and Event Knowledge	0.289	1.309	0.008	0.021
Self-Organizing	0.337	1.312	0.016	0.046

Note. The lower the mean SSE, the better the performance.

between the input and the target, such that the more dissimilar the current input and target are, the higher the SSE. Due to autocorrelation within events, this difference may be particularly extreme on event-boundaries; the inputs and targets may be more similar to one another on frames within events relative to frames between events.

This issue was investigated empirically. The dissimilarity between the input and target of each time point was characterized by calculating the squared Euclidean distance between each input and output pair. The squared Euclidean distance was chosen because it is on the same scale as SSE: it is the SSE that the model would generate if it were re-producing the current inputs as its output. Two pieces of data suggested that distance could potentially account for the increase in SSE associated with event boundaries: First, error increased with increasing distance (mean $r = 0.68$). Second, the distance between inputs and targets was greater for event boundaries than time points within an event, $t(19) = 154$, $p < 0.001$ (mean distance on boundaries $= 1.32$, mean distance within an event $= 0.67$). (However, we note the distributions overlapped substantially; in fact, the range of boundary frame distances [0.30–2.22] fell completely within that of within-event distances [0.06–2.78].) More detailed analyses using hierarchical linear regression techniques characterized the contribution of an event boundary to SSE, above and beyond the contribution of distance. In the first step of the regression, distance and squared distance were used to predict the SSE of the model. In the second step, boundary status was included as a predictor in the model. The results of this analysis were clear. Although the distance variables accounted for 47% of the variance in the SSE variable in this feed-forward network, boundary status accounted for an additional 16% of the total variance, indicating that boundary status is associated with increased SSE, above and beyond the effects of distance, minimum incremental $F(1,9475) = 2994$, $p < 0.001$.

4.2.2. Classifying boundaries as a function of SSE

In addition to hypothesizing that SSE is associated with event boundaries, we also hypothesized that it could be used in order to discriminate between time points within an event and event boundaries. Even though there was a large difference between the mean SSE of within-event and boundary frames across multiple replications, inspection of the distributions

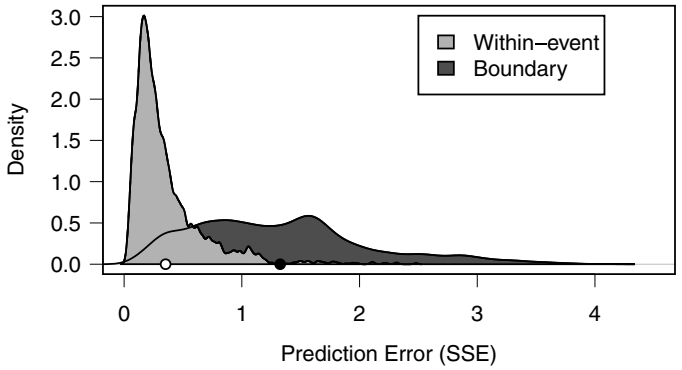

Fig. 4. Distributions of prediction error for the feed-forward network. The light grey distribution reflects the distribution of prediction error for within-event frames. The dark grey distribution reflects the distribution of prediction error for boundaries. The open and closed circles reflect the mean of the within-frame and boundary distributions, respectively.

of SSE on within-event and boundary frames indicated that there was substantial overlap between them (see Fig. 4). This overlap indicates that SSE cannot perfectly classify frames as within an event or as a boundary. In order to investigate how well SSE served this function, linear discriminant analyses were used to predict whether a frame was a boundary or not on the basis of either the SSE or the distance between the input and output at that point in time. All discriminant analyses used the original probabilities of boundary status as the prior probabilities for classification (90.5% within-event frames, 9.5% boundaries). The linear discriminant analysis using SSE as the predictor was able to identify correctly over 54% of the boundaries, while falsely identifying 2% of frames (SSE scaling = 2.72), which was a statistically significant increase relative to the proportional chance criterion, minimum $t(9496) = 23.1$, $p < 0.001$. By comparison, the model using distance as the predictor variable identified only 7% of the true boundaries, but also identified 4% of the within-event frames as boundaries (distance scaling = 2.02). While this was also a significant increase relative to chance, minimum $t(9469) = 11.4$, $p < 0.001$, the discriminant analysis based on SSE consistently outperformed the discriminant analysis based on distance, $t(19) = 32$, $p < 0.001$.

In order to control for distance in a more direct way, we also ran an additional simulation with the feed-forward architecture. In this simulation, linear interpolation was used to create boundaries that lasted two frames, and therefore had an average distance between inputs and outputs that was half of that in the original, un-interpolated simulation. This interpolation resulted in boundaries that had a mean distance between inputs and outputs that was numerically (but not statistically) smaller than the mean distance between inputs and outputs on within-event frames, $t(6712) = -1.27$, $p = 0.20$ (mean distance on boundaries = 0.66, mean distance within an event = 0.67). Even within this environment, boundary status produced an increase in SSE relative to within-event frames, minimum $t(3539) = 9.0$, $p < 0.001$ (mean SSE on boundaries = 0.42, mean SSE within an event = 0.34). This simulation demonstrates that distance between the input and output was not the only driving factor in producing increased SSE on boundaries.

4.3. Discussion

The results from Simulation I suggest that the feed-forward network was able to perform the prediction task quite well, and further, that prediction error was related to boundary status. This increase in prediction error at event boundaries is a direct result of the statistical structure of the environment. Within an event, a perceptual input always predicts exactly the same thing (e.g., frame 2 of "bowing" always comes after frame 1 of "bowing"). However, between events, a perceptual input does not always predict the same thing: the first frame of any event could follow the last frame of a given event (e.g., frame 1 of "bowing," "cheering," or "driving," etc., could occur after the last frame of "sawing"). Consequently, the network must attempt to reduce the global prediction error for all such transitions. The network accomplishes such a global minimization by attempting to produce the mean of all first frames when it reaches a boundary. This solution is the best the network can hope to do, but it results in errors at event beginnings that are larger and more variable than errors within an event. As a result, sudden increases in prediction error form a good but not perfect signal that a new event has begun. A discriminant analysis was able to classify over 50% of the boundary frames appropriately based solely on prediction error, while false-alarming to only 2% of within-event frames.

We also observed that prediction errors were larger for frames on which the body had moved a larger distance, and that such large jumps were more likely at boundaries between events. Neither finding was unexpected. The relationship between distance and prediction error is a natural consequence of the tendency of PDP models and related models to make regressive (conservative) predictions. The fact that within-event distances were smaller than cross-event distances reflects the fact that actions tend to be characterized by a subset of the possible poses the body can take on. Neither of these relationships was sufficient to account for the transient increases in prediction error at event boundaries, as shown by the simulations using interpolated event boundaries. In the following simulations we chose to retain the uninterpolated stimuli, because there is evidence that in naturally occurring activity there are larger motion changes at event boundaries, and people appear to use such cues for segmentation (Newtson, Engquist, & Bois, 1977; Zacks, 2004).

After verifying that boundary status was related to prediction error, the next question of interest was whether a network could use knowledge about the identity of the current event in order to improve its performance on the prediction task. If such stable representations were useful to a cognitive system, there would be a computational pressure to develop them.

5. Simulation IIa—Does having knowledge about the current event improve performance?

The previous simulation investigated whether prediction error was related to boundary status in a network without explicit event knowledge. The current simulation focused on investigating how additional information regarding the identity and timing of the current event influences prediction.

5.1. Methods

This network, termed the *event knowledge network*, was comprised of the core feed-forward network discussed above with one additional component: this network had an additional input layer with 13 units. Each of these 13 units coded a different event in a localist fashion, such that unit 1 coded "kicking," unit 2 coded "bowing," and unit 3 coded "cheering," etc. The activity states of these 13 units always identified the current event, such that the unit that identified the current event had an activation value of 1.0, and the units identifying all other events had activation values of 0. These activation states were appropriately updated as soon as a new event began.

5.2. Results

This network outperformed the feed-forward network from Simulation I. The event knowledge network had a mean SSE of 0.37. Relative to the feed-forward network in Simulation I, event knowledge facilitated performance on within-event frames, $t(38) = 12.0$, $p < 0.001$ (mean SSE = 0.27, see Table 1), but did not facilitate performance on boundaries, $t(38) = -0.1$, $p = 0.93$ (mean SSE = 1.33). This decrease in SSE on within-event frames reflects a 23% improvement (e.g., decrease in the amount of prediction error) from the feed-forward network (see Fig. 5).

There was a strong positive relationship between distance and SSE ($r = 0.57$), but the strength of this relationship was smaller than that found in the feed-forward network ($r = 0.68$), minimum $t(18946) = 6.4$, $p < 0.001$. As in Simulation I, boundary status accounted

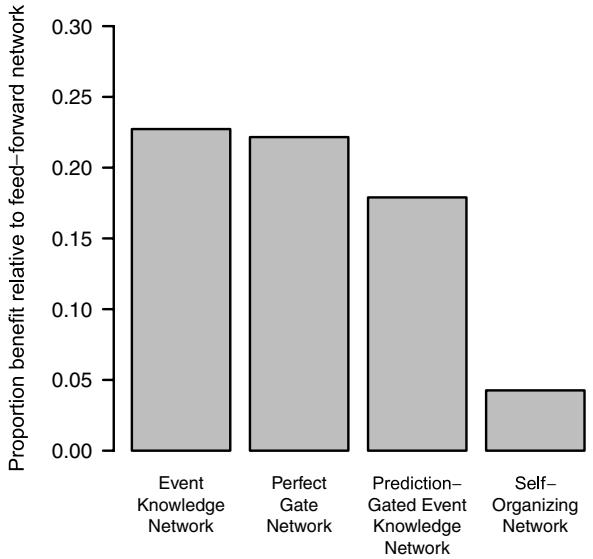

Fig. 5. The proportion improvement of the original feed-forward network's prediction error. This proportion is calculated as the mean (feed-forward SSE—model SSE)/feed-forward SSE. All models demonstrate a significant benefit relative to the feed-forward network.

for a significant proportion of variance above and beyond the variability accounted for by distance, minimum $F(1,9450) = 6250$, $p < 0.001$ ($\Delta R^2 = 0.29$), and across replications, this increase in R^2 was greater than that of the feed-forward network, $t(38) = 29.6$, $p < 0.001$.

5.3. Discussion

Having knowledge of the current event clearly helped the network perform the prediction task. This facilitation occurred solely on within-event frames, and therefore, shifted the distribution of SSE on within-event frames away from the distribution associated with event boundaries.

Event knowledge facilitated performance by disambiguating similar inputs. Analysis of the environment identified a subset of within-event frames that were particularly difficult (note the positive skew in the within-event distribution of SSE in Fig. 4). Close inspection of these difficult (i.e., high SSE) within-event frames revealed that the increased SSE was due to increased similarity to other frames that required different outputs (see panels C and D in Fig. 1). In Fig. 1, two frames with similar inputs and distinct outputs are within the same event. However, if these two frames occurred in different events, then an additional contextual signal (such as the localist representations provided in the current simulation) could disambiguate the two inputs and facilitate the determination of the appropriate prediction.

In addition to showing that stable event representations facilitated performance, adding conceptual event information was shown to reduce the relationship between prediction error and perceptual variables (e.g., distance between the input and target values). This finding was not unexpected. In previous empirical work, perceptual variables such as velocity accounted for more variance in segmentation performance when segmenting at fine relative to coarse boundaries (Zacks, 2004). These data suggest that conceptual knowledge about events interacts in a top-down fashion with how participants use movement features to segment events. In situations where conceptual information may be particularly relevant (e.g., when segmenting at a coarse grain), subtle perceptual changes may not be particularly diagnostic of an upcoming change. This suggests that when conceptual information is available and relevant, the role of perceptual variables may be reduced (although not necessarily). In the model, stable event knowledge has a similar effect. Event knowledge is a robust and relevant cue that facilitates perception and prediction, and therefore, access to event knowledge provides the model with useful top-down information to utilize in generating its prediction. This additional information allows the model to become less dependent upon perceptual changes.

Thus, the simulation results demonstrate that a stable discriminating source of information about the current event (in the form of the event knowledge input units) can facilitate performance within a prediction task, particularly through disambiguating similar inputs. In addition to information regarding the current event, there are other sources of information that may be able to disambiguate two similar inputs. One such piece of information is information regarding its recent history. Such recent history information has been used previously in simple recurrent networks to perform sequential prediction tasks (Elman, 1991), and the question remains as to whether event knowledge, as currently implemented, provides the same type of information as that provided by a *context* layer of a simple recurrent network. This question was addressed in the following two models within Simulation IIb. The first model examines

the performance of a simple recurrent network on the prediction task, and the second model examines the performance of a model with components of both the event knowledge network and the simple recurrent network on the prediction task. To the extent that the benefits of the event knowledge input and the context layer of a simple recurrent network result in additive benefits in the prediction task, one can argue that each processing component is contributing independent and unique information.

6. Simulation IIb—Does event knowledge contribute the same information as a simple-recurrent network?

The primary goal of Simulation IIb was to determine whether event knowledge provided information similar to that provided by a standard in the sequential prediction literature, the simple recurrent network (Elman, 1991). This goal was accomplished through running two simulations: a standard simple recurrent network, and a simple recurrent network with explicit access to event knowledge.

6.1. Methods

Two models were run for this simulation. The first model was a simple recurrent network constructed by adding a context layer of 100 units to the feed-forward architecture used in Simulation I (see Fig. 2). The units in the context layer received one-to-one connections from corresponding units in the hidden layer, and they were updated by copying the previous frame's activation values from the corresponding unit in the hidden layer. Each unit in the context layer sent a connection to every unit in the hidden layer (e.g., it was fully connected). This context layer allowed the simple recurrent network to maintain information across frames: the values of the hidden layer on frame t are copied to the context layer and influence processing on frame $t + 1$.

The second model was a simple recurrent network (as described in the previous paragraph) augmented by an additional input layer that coded each event, exactly as described in the event knowledge network (Simulation IIa).

6.2. Results

The simple recurrent network demonstrated similar patterns to both the feed-forward network and the event knowledge network. The simple recurrent network produced a mean SSE of 0.40 across all frames, which was a significant improvement in performance relative to the feed-forward network, $t(38) = 15$, $p < 0.001$. Similar to both previous networks, within-event frames had lower SSE than event boundaries, $t(19) = 103$, $p < 0.001$ (mean within-event SSE = 0.31, mean boundary SSE = 1.24).

The augmented event knowledge and simple recurrent network also demonstrated similar patterns to previous networks. This network produced a mean SSE of 0.32 across all frames, which was a significant improvement in performance relative to all other networks, minimum $t(38) = 36$, $p < 0.001$. Similar to all previous networks, within-event frames had lower

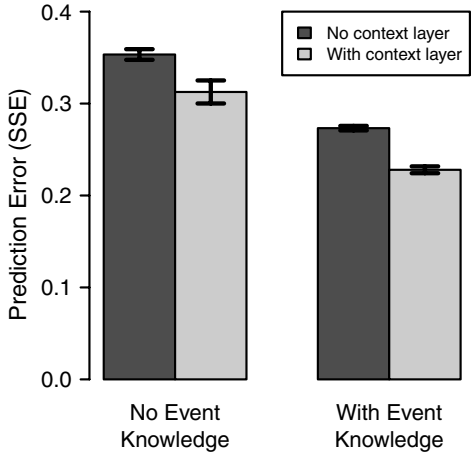

Fig. 6. Respective contributions of context and event-knowledge layers. Both types of layers provide a benefit to prediction (i.e., main effects), but the two types of layers do not interact. Error bars represent 99% confidence interval across 20 replications.

SSE than event boundaries, $t(19) = 167$, $p < 0.001$ (mean within-event SSE = 0.23, mean boundary SSE = 1.17).

In order to determine whether the contributions of event knowledge and context layers were different, the prediction error on within-event frames for the feed-forward, event knowledge, simple recurrent, and augmented event knowledge and simple recurrent networks were entered into a 2 (context layer component of simple recurrent network present vs. absent) × 2 (event knowledge units present vs. absent) ANOVA. Although both components exhibited main effects (context layer component: $F(1,76) = 288$, $p < 0.001$; event knowledge units: $F(1,76) = 1054$, $p < 0.001$), the two factors did not show a significant interaction ($F < 1$), suggesting that each component contributes unique information towards solving the prediction problem within events (see Fig. 6).

6.3. Discussion

Similar to Simulation IIa, these data suggest that stable information regarding the current event and information regarding local history can facilitate performance within a prediction task. Importantly, these data demonstrate that the contribution made by the event knowledge input units is distinct from the contribution made by the *context* layer of a simple recurrent network. As discussed in the introduction, simple recurrent networks have difficulties learning long-term temporal contingencies (Bengio, Simard, & Frasconi, 1994; Hochreiter & Schmidhuber, 1997). This difficulty is contrasted with the fact that in the current simulations, events have relatively long durations (9–12 inputs), and therefore, the explicit event information provided by the event knowledge units is able to provide disambiguating information over extended periods of time. The ability of a context layer in a simple recurrent network to capitalize on recent history allows networks with such a context layer to pick up on short-term temporal dependencies, whereas the longer duration of event information allows networks

with access to this information to bridge relatively long-term temporal dependencies. The current set of simulations is focused on understanding how representations that span relatively long time periods (i.e., the length of events) may develop, self-organize, and benefit perception. Because the contributions of each grain of temporal dependency operates independently in the current environment, and because the set of simulations reported here is focused on isolating and understanding the contributions that long-term event knowledge may have on sequential performance, we investigate additional mechanisms in isolation.

The facilitation enabled by sensitivity to long-term temporal dependencies suggests that there may be a computational pressure to develop internal representations for event knowledge. In order for a system to appropriately develop these representations, that system must have two different characteristics: a) the ability to use information available in the perceptual environment to develop stable representations and b) the ability to learn when these representations should be updated. These two properties were teased apart in the following two simulations. Simulation IIIa investigated whether a system provided with appropriate updating signals could use information in the perceptual inputs to develop stable event representations that facilitate performance. Simulation IIIb investigated whether a system with hand-coded event representations could learn when to update these representations based on moment-to-moment fluctuations in prediction error.

7. Simulation IIIa—Can a system with access to event boundaries use the perceptual inputs to develop stable event representations that facilitate performance?

Simulation IIIa asked the following question: If a network is told when new events begin, can it use that information to develop effective representations of what those events are? In this simulation the network was given explicit signals to update its representation of the current event, but was given no information about event identity. Instead, it was outfitted with a pool of units that had the opportunity to form stable event representations based solely on the updating signals and perceptual input.

7.1. Methods

One primary characteristic of event knowledge is that it has to be stable over the course of an event, but it also has to be rapidly updated when the event changes. In order to capture this pattern of temporal dynamics, an additional gated layer was introduced into the network architecture. This additional layer (termed the *event layer*) had 54 units, received input from the input layer, and it projected fully to the hidden layer (see Fig. 2). Following Hochreiter and Schmidhuber (1997), the units within this layer had a linear activation function. Each unit in this layer received two inputs: a one-to-one connection from the corresponding unit in the perceptual input layer and a self-recurrent connection. However, the connections projecting to this layer were modulated by a gating signal. This gating signal permitted the activity of the receiving units to be maintained in the face of additional interfering inputs, but also allowed for the inputs to be quickly and flexibly updated when the signal occurred (Hochreiter

& Schmidhuber, 1997). Therefore, the units' activity states were updated flexibly at event boundaries, but their activity states were maintained over the duration of each event (i.e. until a subsequent boundary was encountered).

In the current simulation, termed the *perfect gate network*, the gating signal provided perfect information regarding the timing of event boundaries:

$$\text{GatingSignal}_t = \begin{cases} 1 & \text{if boundary,} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where $GatingSignal_t$ is the gating signal at frame t.

The gating signal modulated the input and recurrent weights to the event layer, according to:

$$\Delta Input\ Wt = -0.5 \cdot Input\ Wt_t + (1.0 - Input\ Wt_t) \cdot GatingSignal_t \tag{5}$$

$$Recurrent\ Wt_t = 1.0 - Input\ Wt_t \tag{6}$$

where $Input\ Wt_t$ was the weight from the perceptual input layer to the event layer at frame t, $\Delta Input\ Wt$ is the change in the input weight between frames t and $t + 1$, $GatingSignal_t$ was the gating signal at frame t, and $Recurrent\ Wt_t$ was the self-recurrent weight within the event layer at time t. The input weights to the event layer were influenced by two forces. First, the input weight had a constant decay towards 0 (the first half of Equation 5). Second, when a gating signal occurred, the weight was pushed towards 1.0 (the second half of Equation 5). Given the binary status and the timing of the gating signal in the current experiment, this effectively reset the input weight to 1.0 at the beginning of each event. Equation 6 defines a trade-off between the input and self-recurrent weights; when the recurrent weight is near 1.0, the input weight has to approach 0, and vice-versa. Because the units in the event layer have a linear activation function, the sum of the weights feeding into it must be 1 (as required by Equation 6). Therefore, each of these weights represents the proportion of activity that is being represented by that unit. If the external input weight is 1 and the recurrent weight is 0, then the new activity state in the event layer will be a copy of the external input. If both weights are 0.5, then the new activity state will consist of the average of the external input and the unit's previous activity state. Finally, if the external input weight is 0 and the recurrent weight is 1, then the new activity state will be a copy of its previous activity state. The trade-off defined by Equation 6, combined with the relatively slow decay of the input weights, produces a stable pattern of activity in the event layer that effectively represents a weighted average of the first few frames of input that occur after a gating signal. This property of the model is an implicit hypothesis within the model: it suggests that the model will only develop unique event representations if the first few moments of each event are distinct from those of other events. This is likely to be a reasonable assumption, as prediction error (on which this gating signal is hypothesized to be based in subsequent simulations) likely will be disproportionately large on frames that have unique target values, and low on frames that have common target values.

7.2. Results

This network outperformed the feed-forward network from Simulation I, and its performance was strikingly similar to the event knowledge network from Simulation IIa (see Table 1 and Fig. 5). This perfect gate network had a mean SSE of 0.37. Relative to the feed-forward network from simulation I, the perfect gating system facilitated performance on within-event frames, $t(38) = 27$, $p < 0.001$ (mean SSE = 0.27, see Table 1). Relative to the feed-forward network, the perfect gating system facilitated performance on boundaries, $t(38) = 3.9$, $p < 0.001$ (mean SSE = 1.29). This decrease in SSE associated with within-event frames was roughly the same as that appearing when both timing information and event content information was provided, $t(38) = 1.1$, $p = 0.27$ (23% improvement relative to the feed-forward network).

Similar to the event knowledge network, there was a strong positive relationship between distance and SSE ($r = 0.57$), but this relationship was smaller than that found in the feed-forward network, minimum $t(18950) = 11.7$, $p < 0.001$. Boundary status accounted for a significant proportion of variance above and beyond the variability accounted for by distance, minimum $F(1,9433) = 5794$, $p < 0.001$ ($\Delta R^2 = 0.30$). This demonstrates that access to the timing of each event is sufficient to decrease the relationship between perceptual variables and prediction error.

In order to investigate whether this network developed event representations that were stable over time, the event layer's activation patterns were recorded while the network saw one instance of each event. These 137 observations (one for each frame) of the 54 unit activities were then subjected to a clustering algorithm in which the data were classified into 13 clusters, using the partitioning around medoids (PAM) algorithm (Kaufman & Rousseeuw, 1990). The clusters identified matched the original events almost perfectly, with only 1 of the 137 frames not clustered with the other frames of its event (one frame of "bowing" was mistaken as a frame of "cheering"). In contrast, if the same clustering algorithm were applied to the perceptual input values over the same sequence of frames, there is a large degree of confusion, such that 61 out of 137 frames were mis-classified (44.5%), and only 1 event was classified perfectly. The event "driving" had no frames from other events in the empirically defined cluster, and no frames from "driving" were classified within an alternate cluster.

7.2.1. Noisy environments

One potential concern is that the model always sees the exact same instance of each event, whereas humans rarely, if ever, see the exact same sequence of actions more than once. This is particularly a concern for the current simulation, because the network uses snapshots of the perceptual input as stable internal representations that bias on-going processing, and it is unclear whether the network will benefit from these stable internal representations if they are based on noisy inputs. In order to address this concern, the feed-forward and perfect gate networks were run with different amounts of 0-mean gaussian noise applied to the inputs. Results indicated that across a relatively wide spectrum of noise values (up to $\sigma = 0.1$), the perfect gate network continued to outperform the feed-forward network on within-event frames, minimum $t(38) = 17$, $p < 0.001$ (see Table 2).

Table 2
Mean SSE on within-event frames for different levels of noises

SD of noise	Feed-Forward	Perfect Gate
0.005	0.357	0.279
0.01	0.367	0.288
0.02	0.397	0.321
0.03	0.416	0.351
0.04	0.439	0.375
0.05	0.461	0.395
0.1	0.538	0.473

Note. The lower the mean SSE, the better the performance.

7.3. Discussion

Telling a network when new events began and providing it with ongoing sensory information was sufficient to allow the network to develop representations that accurately identified the different events. These representations helped perceptual prediction virtually as much as hand-coded event labels. Additionally, adding noise to the environment indicates that this timing information is useful, even if the events the network encounters are never the same across multiple instances. These event representations cluster together to a greater degree than the perceptual inputs do, and therefore, provide a benefit very similar to that seen when explicit information regarding the timing and content of an event is provided (Simulation IIa).

8. Simulation IIIb—Does a system benefit from a gating signal based on prediction error to update event representations?

Whereas Simulation IIIa focused on determining whether a network with perfect access to the timing of the event boundaries could use that information to develop temporally stable, informative representations, Simulation IIIb focused on determining whether a network could learn to use prediction error as a signal from which to update pre-existing event representations.

8.1. Methods

The architecture of this model was identical to that seen in the event knowledge network (Simulation IIa). However, in Simulation IIa, the event representations were forced to appropriately update at the beginning of each event. In the current model, termed the *prediction-gated event knowledge network*, the event representations were updated only after an unsupervised gating signal occurred. This gating signal was based on a relative measure of prediction error. The measure was relative, so that it could be applicable over the entire course of training. The measure was made relative by computing a running average of prediction error via a low-pass filter:

$$AvgPredErr_t = AvgPredErr_{t-1} + 0.05 \cdot (SSE_t - AvgPredErr_{t-1}) \quad (7)$$

where $AvgPredErr_t$ is the average prediction error at frame t and SSE_t is the prediction error (SSE) on frame t. The coefficient of 0.05 was used so that this average prediction error would reflect a relatively stable value (e.g., approximating a running average of the last 20 time-steps). This value was therefore stable enough to be resistant to noise in the data, but fast enough to normalize to the mean SSE throughout training. When the current prediction error was greater than 1.5 times this average, a gating signal occurred:

$$GatingSignal_t = \begin{cases} 1 & \text{if } \dfrac{SSE_t}{AvgPredErr_{t-1}} > 1.5, \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

where $GatingSignal_t$ is the gating signal for frame t, SSE_t is the prediction error on frame t, and $AvgPredErr_{t-1}$ is the average prediction error on frame $t-1$. If there were a transient increase in prediction error, then a gating signal would occur, and the activity states of the event layer would be updated to the appropriate values reflecting the current event. If prediction error were perfectly tied to event boundaries, such that all within-event frames had low SSE, and all event boundary frames had high SSE, then this would result in a special case that would produce the exact same model behavior seen in Simulation II (the event knowledge network). If prediction error were noisy, such that some boundaries were to have relatively low SSE, then the low SSE boundaries would result in perseverative errors: The model would continue to maintain the previous event representation until a transient increase in error was encountered.

8.2. Results

This network outperformed the feed-forward network from Simulation I, and its performance was similar to the models from Simulations IIa and IIIa (see Fig. 5). The prediction-gated event knowledge network had a mean SSE of 0.39. Relative to the feed-forward network, the prediction-gated event knowledge network facilitated performance on within-event frames, $t(38) = 28$, $p < 0.001$ (mean SSE = 0.29, see Table 1), but not on boundaries, maximum $t(38) = 0.26$, $p = 0.79$ (mean SSE = 1.31). This decrease in SSE associated with within-event frames was not quite as large as that appearing when both timing information and event content information was provided, $t(38) = 8.2$, $p < 0.001$ (17% improvement relative to the feed-forward network).

Similar to the event knowledge and perfect gate networks, there was a strong positive relationship between distance and SSE ($r = 0.59$), but this relationship was smaller than that found in the feed-forward network, minimum $t(18979) = 6.7$, $p < 0.001$. Boundary status accounted for a significant proportion of variance above and beyond the variability accounted for by distance, minimum $F(1,9490) = 5521$, $p < 0.001$ ($\Delta R^2 = 0.25$). Similar to results identified in the event knowledge network, this demonstrates that a noisy updating signal coupled with stable event representations is sufficient to decrease the relationship between perceptual variables and prediction error.

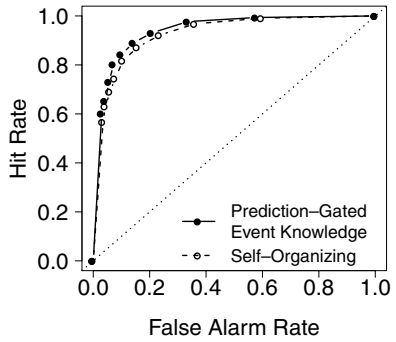

Fig. 7. Receiver Operating Characteristic (ROC) curve plotting the hit rate as a function of false alarm rate for multiple thresholds, for the prediction-gated event knowledge (Simulation IIIb) and self-organizing (Simulation IV) networks.

8.2.1. Transient increases in SSE as a gating signal

The gating system based on prediction error classified 83.9% of the boundary frames correctly as boundaries. The network also misidentified 10.5% of within-event frames as boundaries. Rather than hurting performance, these "false alarms" could be used to update the model to the appropriate event representation if it had previously missed an initial boundary. This was clear in the data, as the probability of identifying a within-event frame as a boundary was 0.289 when the network was currently maintaining the previous (incorrect) event representation (e.g., "perseverating"), but only 0.097 when the network was already maintaining the current event, $t(19) = 17.8$, $p < 0.001$. On the events in which the model did not update its event representations at the appropriate boundary, it took an average of 2.01 frames for it to update its event representations to the appropriate value. The ability to perform delayed updating resulted in the correct event representation being active on 96% of frames.

Further, this ability to correctly label event boundaries was robust to the selection of the threshold value (see Fig. 7). This was investigated by running multiple models and varying the threshold parameter in Equation 8 from 0.5 to 2.5 in increments of 0.25. Plotting the hit rate (e.g., identifying a boundary correctly) as a function of false alarm rate (e.g., identifying a within-event frame as boundary) across multiple thresholds produces a Receiver Operating Characteristic (ROC) curve in which the area under the curve can be used to estimate the ability to discriminate between the two different distributions. The mean area under the curve for this network architecture was 0.94, indicating that this model was able to discriminate between within-event and boundary frames very well (see Fig. 7).

8.3. Discussion

This simulation provided clear evidence that a gating system based on prediction error can be used to update representations in a useful fashion. The network identified 83.9% of the natural event boundaries. If the system missed a natural event boundary, it took, on average, only 2 frames for it to update to the appropriate event. This relatively quick correction was

largely due to the increase in prediction error caused by the perseverating activity of the incorrect event representation. Because of the network's ability to correctly identify most of the boundaries and its ability to correct those that it missed, the correct event representation was active on 96% of the frames. This unsupervised gating signal was able to update internal representations, and these internal representations facilitated performance in the same way as stable event representations present in previous simulations. Taken together, Simulations IIIa and IIIb suggest that a) internal event representations that facilitate performance can develop in networks provided with appropriate timing information and b) prediction error can be used in these networks to update stable internal event representations. The next step was to investigate whether the network could self-organize both of these properties simultaneously. Thus, a final simulation investigated whether a network with access to only prediction error and perceptual inputs could learn to identify event boundaries, and further, use these self-identified event boundaries to update internally developed representations based solely on the perceptual input.

9. Simulation IV—Can a system utilize prediction error to identify events and use this information appropriately to develop useful event representations?

This simulation investigated the final question of interest, namely, whether a system can learn to identify when events start, and further, use that information to develop stable event representations that facilitate performance.

9.1. Methods

To investigate this issue, properties from Simulations IIIa and IIIb were combined into a single model (the *self-organizing network*). Specifically, the current network had exactly the same structure as that described in Simulation IIIa. As in Simulation IIIa, a gating signal was used to modulate the input and recurrent weights to an event layer (see Equations 5 and 6). However, the gating signal itself was not supervised as in Simulation IIIa. Rather, the gating signal was based on relative prediction error, and it was implemented exactly as that described in Simulation IIIb (see Equations 7 and 8). In this way, the network used only relative prediction error to determine whether a frame was a boundary or not, and it had access to only the perceptual inputs in order to develop stable event representations.

9.2. Results

This network outperformed the feed-forward network from Simulation I (see Fig. 5). The self organizing network had an average SSE of 0.43. Relative to the feed-forward network, the additional mechanisms facilitated performance on within-event frames, $t(38) = 3.88$, $p < 0.001$ (mean SSE of 0.34, see Table 1), but did not facilitate performance on boundaries, $t(38) = 1.0$, $p = 0.3$ (mean SSE = 1.31). This decrease in SSE, although smaller than that

seen in the other augmented models, still reflected a significant difference for within-event frames, but not for boundaries.

9.2.1. Transient increases in SSE as a gating signal

The gating system performed similarly to that in Simulation IIIb. The model identified 82.9% of the natural event boundaries successfully and identified 10.5% of the within-event frames as boundaries. Again, the ability to discriminate between within-event and boundary frames was not particularly sensitive to the actual threshold that was used, as the area under the ROC curve was 0.92 (see Fig. 7).

Additional analyses were conducted in order to characterize the sources and consequences of the errors the model did make. The consequences of these errors were not anticipated to be the same as those seen Simulation IIIb, because in that simulation, a gating signal within an event always updated the activity states of the event layer to the same internal representation, regardless of when the gating signal occurred within that event. In the current simulation, the content of the event representation depended on *when* the gating signal occurred within an event, because the updated event representations were dependent upon the perceptual input while the gate was open. Specifically, this means that updating an event's internal representation on the first frame of the event results in a different internal representation than if it were updated on the second frame of the event.

To investigate whether this non-stationary aspect of the event representations was driving performance, each frame was classified as to whether the appropriate event representation was active. The current event representation was considered appropriate if a gating signal occurred on the boundary between the previous event and the current one, and remained appropriate until either an additional gating signal occurred within the event such that the event representation was updated to a new activity state, or a new event began and a gating signal did not occur, at which point the network would perseverate on the previous event. This definition of "appropriate" was used because it paralleled Simulation IIIa, in which a gating signal occurred at event boundaries, but nowhere else. Therefore, either a false alarm (a gating signal on a within-event frame) or a miss (no gating signal at a boundary) would result in inappropriate activity in the event layer. This definition resulted in 60% of the frames being classified as having appropriate event context. Frames with appropriate event context out-performed the within-event frames from the feed-forward network, $t(38) = 15$, $p < 0.001$ (mean SSE = 0.30). However, the frames with inappropriate event context were hurt relative to the feed-forward network, $t(38) = 5.4$, $p < 0.001$ (mean SSE = 0.39). In fact, the frames with inappropriate context representations appeared to be the primary cause underlying why the model underperformed relative to the previous augmented models.

9.3. Discussion

This model suggested that a gating mechanism based on prediction error learned to identify boundaries. This gating signal was used to update event representations based on the perceptual input available to the model, and this combined system was able to facilitate performance relative to a network that had no ability to maintain previous history (the feed-forward network).

10. General discussion

Taken together, these simulations suggest that prediction error is sufficient to identify event boundaries. Coupled with access to perceptual information, prediction error is also sufficient to develop event representations that can be used to facilitate performance. To summarize, five important results were established. First, prediction error was greater during event boundaries than within events. This provides formal evidence suggesting that one mechanism underlying the subjective experience of an event boundary is the inability to predict upcoming stimuli (Zacks, 2004; Zacks, Speer, Swallow, Braver, & Reynolds, 2005). Second, stable information regarding the identity of an event was found to improve the prediction of event sequences. Consistent with previous results (Hochreiter & Schmidhuber, 1997) we found that such event knowledge representations provide unique information about long-term sequential dependencies that are not encoded by simpler architectures such as SRNs. This suggests that parsing continuous activity into discrete units improves perception. Therefore, this facilitation argues that there may be a computational pressure to parse streams into discrete events. Third, gating signals occurring at event boundaries were sufficient to learn and update internal context representations that reflect event knowledge. This suggests that if a system is provided with reliable information regarding the boundaries between relevant events (and appropriate perceptual inputs and memory mechanisms), it can learn to develop appropriate internal event representations. Fourth, gating signals based on prediction error can be used to reliably update internal context representations that carry event information and facilitate subsequent prediction. This provides formal evidence that prediction error can be used to update appropriate event representations without additional instruction. These event representations are sufficient to facilitate performance, even though they are not updated perfectly. Finally, gating signals based on prediction error were sufficient to learn internal context representations for events. This suggests that a self-organizing system is able to identify a majority of natural event boundaries as such, and further, to develop event representations that facilitate performance relative to a simple feed-forward architecture.

Taken together, these simulations provide strong initial support for the hypothesis that prediction error can be used to identify event boundaries, develop internal event representations that guide behavior, and subsequently update these internal representations appropriately. These simulations support the hypothesis that error-based gating may underlie how event representations could self-organize and form the basis of how humans spontaneously segment continuous activity into meaningful units. Further, this approach augments previous computational approaches that use entropy to segment continuous activity by providing a biologically plausible measure (i.e., prediction error) that would track levels of entropy of an environment (Cohen & Adams, 2001).

10.1. Relationship between low-level perceptual variables and event perception

It is clear that low-level perceptual variables (such as the distance between the input and target values) were related to prediction error, and hence, the probability of a gating signal. As

event representations provided disambiguating information about these perceptual inputs, this relationship decreased. This is similar to effects identified in previous behavioral studies, which suggested that participants sometimes use low-level movement cues to segment continuous activity into meaningful units, but that when the activity appears to be goal directed they depend less on these cues, and may rely more on top-down knowledge based on previous experience with similar situations (Zacks, 2004).

10.2. Limitations of the current environment

Two limitations of the current simulations are associated with the environment used. Typically, events are perceived as having a hierarchical part-subpart structure (Hard, Lozano, & Tversky, 2006; Hard, Tversky, & Lang, 2006; Lozano, Hard, & Tversky, 2006; Zacks, Tversky, & Iyer, 2001b). However, the events in the current simulation captured only the two lowest levels in such a hierarchy—the grouping of a set of frames into an event. The hierarchical part-subpart structure of everyday activity could be simulated in future studies by manipulating the probability that one event follows another. For example, a "meta-event" could be defined as the sequence of "opening a door," "sitting," and "drinking." This meta-event could occur in the context of other meta-events which include some of the same component events. Based on previous neuroimaging data (Zacks, Braver, Sheridan, Donaldson, Snyder, Ollinger, Buckner, & Raichle, 2001a), one hypothesis would be that prediction error between meta-events would be larger than the prediction error between component events, which would, in turn, be larger than the prediction error on within-event frames. In addition to investigating this hypothesis, further research will be needed to determine whether the current framework for developing event representations can be extended to account for multiple levels of event abstraction and representation.

10.3. Neural correlates

To this point, the discussion of the neural correlates of processes within the model has been limited to the initial constraints hypothesizing that event representations are subserved by active representations in PFC and that the gating mechanism may be intimately tied to the DA neuromodulatory system. However, it is possible to flesh out some additional hypotheses about the relationship between the additional component processes identified in the current model and their respective neural substrates. At its core, the model argues that early sensory representations are transformed in a perceptual processing stream leading to representations that predict the state of the world a short time hence. Early-stage representations corresponding to the outputs of primary sensory areas, including primary auditory cortex (A1), primary visual cortex, (V1), and primary somatosensory cortex (S1) cascade through multiple steps of processing. In the visual system, these early-stage representations are propagated and transformed into representations in inferotemporal cortex (IT) that code object properties that are invariant over changes in orientation, lighting, etc (Tanaka, 1996), and representations in MT, MST, and the posterior superior temporal sulcus (pSTS) that code features of object and observer movement (Tootell, Reppas, Kwong, Malach, Born, Brady, Rosen, & Belliveau,

1995). These perceptual processing streams are oriented in time such that they not only represent the current state of the world, but represent predictions about what is likely to happen a short time later (Giese & Poggio, 2003). Therefore, the processing stream between the input and output layers may reflect natural on-going transformations in the perceptual processing pathways.

The model proposes that the perceptual predictions are constantly being compared with actual sensory input, providing an evaluation of how well perception is functioning. Several mechanisms have been identified that could compute prediction error in event structure perception (Schultz & Dickinson, 2000). Increases in prediction error lead to a cascade of processing that has been characterized as an orienting response (Sokolov, Spinks, Naeaetaenen, & Lyytinen, 2002), two components of which we hypothesize to be the resetting of event models and transiently increased sensitivity to sensory input. As stated in the introduction, we hypothesize that this reset (and resulting increased sensitivity to input) is implemented by midbrain neuromodulatory systems. In addition to midbrain neuromodulatory systems, the anterior cingulate cortex (ACC) may play a role in adaptively modulating behavior in response to prediction error (Botvinick, Braver, Barch, Carter, & Cohen, 2001). In fact, ACC and nearby regions have been proposed to underlie the learning of sequential structure in cognitive domains (Koechlin, Danek, Burnod, & Grafman, 2002). One possibility is that the ACC is the locus of perceptual prediction representations, and that discrepancy between ACC representations and perceptual inputs triggers the nuclei of the catecholamine neurotransmitter systems (Cohen, Botvinick, & Carter, 2000). These subcortical nuclei have diffuse projections throughout cortex (Schultz & Dickinson, 2000) and receive inputs from the ACC (Holroyd & Coles, 2002). The resetting of event models may be mediated by projections from the substantia nigra to the striatum, which modulate activity in fronto-striatal circuits, or by specific reciprocal connections with lateral PFC (Picard & Strick, 1996). Thereby, this distributed circuit may underly the computation and realization of a prediction error signal (e.g., SSE).

10.4. Further questions

There are several questions that these data raise for future research in addition to those regarding the environment. First, it would be interesting to see whether a different architecture or additional mechanisms would be able to improve the performance of the self-organized event representations in the current simulations. One potential architecture is to include a transformation between the input layer and the gated event layer. An appropriate transformation (such as a running average) may further facilitate perception by minimizing the movement within events, which would in turn minimize the effects of high error, within-event frames (i.e., false-alarms).

An additional question of interest is what other kinds of training signals can be used to facilitate the learning of event representations. One potential piece of information that individuals are aware of, particularly during development, is the actual identity of particular events. Children are frequently told what particular events are (e.g., "we're going to a baseball game"). One possibility is that the gating and prediction mechanism described here could

be used to bootstrap algorithms for learning to identify events, and that identity information could be fed back to improve prediction performance.

Finally, these networks generate testable predictions regarding event perception. Although the current environment is limited in some respects, it captures some key features of the way people move during everyday activities. Moreover, because the networks directly process human movement sequences, it is possible to present both people and networks with the same input and assess the degree to which they segment the input in similar ways on a frame-by-frame basis. It will be important to complement this theoretical work with empirical studies of whether the points of segmentation identified by the networks correspond to those identified by human observers.

10.5. Summary and conclusions

These simulations begin to address the central questions posed in the introduction. First, they suggest that at least one underlying neurocomputational mechanism underlying event segmentation is the sensitivity of a system to sequential structure. Second, they suggest that a network can use prediction error (e.g., sensitivity to sequential structure) to learn when to appropriately update event representations based on perceptual input. These simulations have shown that a signal based on transient increases in prediction error is sufficient to identify boundaries between events, and that information about event boundaries can be used to improve prediction performance. In sum, the simulations suggest that prediction error can be used to identify boundaries between events and, further, that updating event representations based on this signal improves prediction.

Acknowledgments

The authors would like to thank Nicole Speer, the cognitive control and psychopathology lab, and the dynamic cognition lab for helpful comments and suggestions. This research was supported in part by a grant from the NSF (BCS-0353942), and a NDSEG graduate fellowship.

References

Allain, P., Le Gall, D., Etcharry-Bouyx, F., Aubin, G., & Emile, J. (1999). Mental representation of knowledge following frontal-lobe lesion: dissociations on tasks using scripts. *Journal of Clinical and Experimental Neuropsychology, 21*(5), 643–665.

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks, 5,* 157–166.

Botvinick, M. & Plaut, D. C. (2004). Doing without schema hierarchies: A recurrent connectionist approach to normal and impaired routine sequential action. *Psychological Review, 111,* 395–429.

Botvinick, M. M., Braver, T. S., Barch, D. M., Carter, C. S., & Cohen, J. D. (2001). Conflict monitoring and cognitive control. *Psychological Review, 108*(3), 624–652.

Braver, T. S. & Cohen, J. D. (2000). On the control of control: The role of dopamine in regulating prefrontal function and working memory. In Monsell, S. & Driver, J. (Eds.), *Attention and Performance XVIII*, (pp. 713–737). Cambridge, MA: MIT Press.

Bregler, C. (1997). Learning and recognizing human dynamics in video sequences. In *Computational Vision and Pattern Recognition*, (pp. 568–574).

Brent, M. R. (1999). An efficient, probabilistically sound algorithm for segmentation and word discovery. *Machine Learning, 34,* 71–105.

Cleeremans, A. & McClelland, J. L. (1991). Learning the structure of event sequences. *Journal of Expecrimental Psychology: General, 120,* 235–253.

Cleeremans, A., Servan-Schreiber, D., & McClelland, J. L. (1989). Finite state automata and simple recurrent networks. *Neural Computation, 1,* 372–381.

Cohen, J. D., Botvinick, M., & Carter, C. S. (2000). Anterior cingulate and prefrontal cortex: Who's in control? *Nature Neuroscience, 3*(5), 421–423.

Cohen, J. D., Braver, T. S., & Brown, J. W. (2002). Computational perspectives on dopamine function in prefrontal cortex. *Current Opinions in Neurobiology, 12,* 223–229.

Cohen, P. R. (2001). Fluent learning: Elucidating the structure of episodes. *Lecture Notes in Computer Science, 2189,* 268–277.

Cohen, P. R. & Adams, N. (2001). An algorithm for segmenting categorical time series into meaningful episodes. *Lecture Notes in Computer Science, 2189,* 198–207.

Cooper, R. & Shallice, T. (2000). Contention scheduling and the control of routine activities. *Cognitive Neuropsychology, 17*(4), 297–338.

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

Elman, J. L. (1991). Distriuted representations, simple recurrent networks, and grammatical structure. *Machine Learning, 7,* 195–225.

Fujii, N. & Graybiel, A. M. (2003). Representation of action sequence boundaries by macaque prefrontal cortical neurons. *Science, 301,* 1246–1249.

Giese, M. A. & Poggio, T. (2003). Neural mechanisms for the recognition of biological movements. *Nature Reviews Neuroscience, 4*(3), 179–192.

Gjerdingen, R. O. (1992). Learning syntactically significant temporal patterns of chords: A masking field embedded in an art 3 architecture. *Neural Networks, 5,* 551–734.

Grafman, J. (1995). Similarities and distinctions among current models of prefrontal cortical functions. *Annals of the New York Academy of Sciences, 769,* 337–368.

Grossberg, S. (2003). Resonant neural dynamics of speech perception. *Journal of Phonetics, 31,* 423–445.

Hanson, C. & Hanson, S. J. (1996). Development of schemata during event parsing: Neisser's perceptual cycle as a recurrent connectionist network. *Journal of Cognitive Neuroscience, 8,* 119–134.

Hard, B. M., Lozano, S. C., & Tversky, B. (2006). Hierarchical encoding: Translating perception into action. *Journal of Experimental Psychology: General, 135*, 588–608.

Hard, B. M., Tversky, B., & Lang, D. (2006). Making sense of abstract events: Building event schemas. *Memory & Cognition, 34*(6), 1221–1235.

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

Holroyd, C. B. & Coles, M. G. (2002). The neural basis of human error processing: reinforcement learning, dopamine, and the error-related negativity. *Psychological Review, 109*(4), 679–709.

Kaufman, L. & Rousseeuw, P. J. (1990). *Finding groups in data: an introduction to cluster analysis*. John Wiley and Sons, New York.

Koechlin, E., Danek, A., Burnod, Y., & Grafman, J. (2002). Medial prefrontal and subcortical mechanisms underlying the acquisition of motor and cognitive action sequences in humans. *Neuron, 35*(2), 371–381.

Lozano, S. C., Hard, B. M., & Tversky, B. (2006). Perspective-taking promotes action understanding and learning. *Journal of Experimental Psychology: Human Perception & Performance, 32*(6), 1405–1421.

Newtson, D. (1976). Foundations of attribution: the perception of ongoing behavior. In Harvey, J. H., Ickes, W. J., & Kidd, R. F., (Eds.), *New Directions in Attribution Research*, Volume 1, (pp. 223–248). Hillsdale, NJ: Lawrence Erlbaum Associates.

Newtson, D., Engquist, G., & Bois, J. (1977). The objective basis of behavior units. *Journal of Personality and Social Psychology, 35*(12), 847–862.

Newtson D. (1973). Attribution and the unit of perception of ongoing behavior. *Journal of Personality & Social Psychology, 28*(1), 28–38.

Randall C. O'Reilly, Chadley Dawson, K., & McClelland, J. L. (2005). The pdp++ software (version 3.1) [computer software and manual]. Retreived from http://psych.colorado.edu/ oreilly/PDP++/ PDP++.html.

Picard, N. & Strick, P. L. (1996). Motor areas of the medial wall: A review of their location and functional activation. *Cerebral Cortex, 6*(3), 342–353.

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, 102*(20), 7338–7343.

Rumelhart, D. E. & McClelland, J. L. (1986). *Parallel distributed processing: Explorations in the microstructure of cognition*, Volume 1 and 2. Cambridge, MA: MIT Press.

Saffran, J. R., Newport, E. L., Aslin, R. N., Tunick, R. A., & Barrueco, S. (1997). Incidental langauge learning: Listening (and learning) out of the corner of your ear. *Psychological Science, 8,* 101–105.

Schank, R. C. & Abelson, R. P. (1977). *Scripts, plans, goals, and understanding: an inquiry into human knowledge structures*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Schultz, W. (1998). Predictive reward signal of dopamine neurons. *Journal of Neurophysiology, 80*(1), 1–27.

Schultz, W. & Dickinson, A. (2000). Neuronal coding of prediction errors. *Annual Review of Neuroscience, 23,* 473–500.

Schwartz, M. F., Montgomery, M. W., Fitzpatrick-DeSalme, E. J., Ochipa, C., Coslett, H. B., & Mayer, N. H. (1995). Analysis of a disorder of everyday action. *Cognitive Neuropsychology, 12,* 863–892.

Sirigu, A., Zalla, T., Pillon, B., Grafman, J., Agid, Y., & Dubois, B. (1996). Encoding of sequence and boundaries of scripts following prefrontal lesions. *Cortex, 32*(2), 297–310.

Sirigu, A., Zalla, T., Pillon, B., Grafman, J., Dubois, B., & Agid, Y. (1995). Planning and script analysis following prefrontal lobe lesions. *Annals of the New York Academy of Sciences, 769,* 277–288.

Sokolov, E. N., Spinks, J. A., Naeaetaenen, R., & Lyytinen, H., (Eds.). (2002). *The orienting response in information processing*. Mahwah, NJ: Lawrence Erlbaum Associates.

Speer, N. K., Swallow, K. M., & Zacks, J. M. (2003). Activation of human motion processing areas during event perception. *Cognitive, Affective, & Behavioral Neuroscience, 3,* 335–345.

St. John, M. F. & McClelland, J. L. (1990). Learning and applying contextual constraints in sentence comprehension. *Artificial Intelligence, 46,* 217–257.

Tanaka, K. (1996). Inferotemporal cortex and object vision. *Annual Review of Neuroscience, 19,* 109–139.

Tootell, R. B., Reppas, J. B., Kwong, K. K., Malach, R., Born, R. T., Brady, T. J., Rosen, B. R., & Belliveau, J. W. (1995). Functional analysis of human mt and related visual cortical areas using magnetic resonance imaging. *The Journal of Neuroscience, 15*(4), 3215–3230.

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), 273–293.

Zacks, J. M., Tversky, B., & Iyer, G. (2001b). Perceiving, remembering, and communicating structure in events. *Journal of Experimental Psychology: General, 123,* 29–58.

Zacks, J. M. (2004). Using movement and intentions to understand simple events. *Cognitive Science, 28,* 979–1008.

Zacks, J. M., Braver, T. S., Sheridan, M. A., Donaldson, D. I., Snyder, A. Z., Ollinger, J. M., Buckner, R. L., & Raichle, M. E. (2001). Human brain activity time-locked to perceptual event boundaries. *Nature Neuroscience, 4*(6), 651–5.

Zhao, T. & Nevatia, R. (2002). 3D tracking of human locomotion: a tracking as recognition approach. In *International Conference on Pattern Recognition*, (pp. 546–551).

Zwaan, R. A. & Radvansky, G. A. (1998). Situation models in language comprehension and memory. *Psychological Bulletin, 123,* 162–185.

Appendix

Event Name	Duration (frames)
Martial arts kick	9
Bow	10
Cheering	10
Drinking	12
Driving	11
Opening a door	9
Sawing	12
Sitting down	11
Using a sledge hammer	10
Spilling a drink	11
Standing up	11
Swatting bees	11
Having a tantrum	10

Appendix A. Left column identifies each event, and the right column identifies how long each event lasted.