Winter 12-2022

A Hybrid Model of Event Comprehension Predicts Human Activity at Human Scale

TAN NGUYEN

WASHINGTON UNIVERSITY IN ST. LOUIS

Department of Psychological and Brain Sciences

A Hybrid Model of Event Comprehension Predicts Human Activity at Human Scale

Tan Nguyen

A thesis presented to

The Graduate School

of Washington University in

partial fulfillment of the

requirements for the degree

of Master of Arts

December 2022

St. Louis, Missouri

Table of Contents

List of Figures

List of Tables

Acknowledgements

This research was supported by a grant from the Office of Naval Research (N00014-17-1-2961).

Thank you Dr. Matt Bezdek and Dr. Jeffrey Zacks, this thesis is a product of a close collaboration with them.

Thank you Dr. Todd Braver, Dr. Aaron Bobick, and Dr. Samuel Gershman for helpful advice on developing the model and the manuscript.

Thank you Malcom Tobias for tremendous support on the high-computing clusters.

Thank you Dr. Maverick Smith and DCL lab members for helpful comments on the manuscript.

Thank you Garrett Cunningham, Sarah Hale, Ryan Kahle, Duy Pham, and Chong Wang for performing the chapters used in the stimulus set. Thank you Cory Fox, Emma Lavetter-Keidan, Sierra Revels, Matt Steinhaus, Grace Zhou, and Vi Nguyen for assistance in creating annotations.

Thank you the thesis committee: Dr. Jeffrey Zacks, Dr. Todd Braver, Dr. Wouter Kool.

Tan Nguyen

Washington University in St. Louis

December 2022

ABSTRACT

A Hybrid Model of Event Comprehension Predicts Human Activity at Human Scale

by

Tan Nguyen

Master of Arts in Psychological and Brain Sciences

Brain, Behavior, and Cognition Psychology

Washington University in St. Louis, 2022

Professor Jeffrey M. Zacks, Chair

To act effectively, humans store event schemas and use them to predict the near future. How are schemas learned and represented in memory, and used in online comprehension? One means to answer these questions is modeling event comprehension. What are, then, computational principles of event comprehension? We proposed three candidate properties: 1) abstract representation of visual features, 2) predictive mechanism and prediction error as feedback, and 3) contextual cues to guide prediction, and adapted a computational model embodying these properties. The model learned to predict activity dynamics from one pass through an 18-hour corpus of naturalistic human activity. Evaluated on another 3.5 hours of activities, it updated at times corresponding with human segmentation and formed human-like event categories—despite being given no feedback about segmentation or categorization. These results establish that a computational model embodying the three proposed properties can naturally reproduce two important features of human event comprehension.

Introduction

To make sense of and act effectively in the world—whether making a cup of tea or writing an email—humans store knowledge and use it to make predictions about what is going to happen[1,2]. How is this type of knowledge represented in memory? How do humans acquire this type of knowledge over time and use this knowledge in online comprehension? This type of knowledge has been historically described as "schemas"[3,4] or "scripts"[5,6]. Schemas or scripts represent generic (probabilistic) knowledge, accumulated by experiencing various examples of a class of events, about how a type of event typically unfold4. For example, a "sandwich-making" schema can comprise of sequence of events "slice tomatoes", "cut the bread", "put vegetables in", and "add condiments," among other sequences. During online comprehension, humans select a schema and construct a working event model—stable representations of the current situation—to predict the near future[7,8]. When the working event model is no longer relevant to the current situation—once the person has done preparing a sandwich and is about to make a cup of coffee—humans should be able to update the working event model to represent "coffee making." Previous models of event comprehension tried to formalize these mechanisms using different types of architecture and showed the correspondence between models' output and human behavior[9–12].

What do models of event comprehension have in common? Examining that question might reveal core computational principles governing human event comprehension. Smith et al[13] built a model of intuitive physics that tracks objects' positions in scenarios typically used in developmental psychology to test core object knowledge[14]. The model has two main modules: 1)

"inverse graphics" module and "physical simulation" module. The "inverse graphics" module uses deep recognition networks to segment and identify objects from raw frames. The "physical simulation" module predicts objects' location in the current frame, using its belief about their locations in the past. The model was tested on physically implausible scenarios (e.g. an object disappears behind an occlude), and the model's surprise scores aligned with surprise level rated by humans more than previous models of intuitive physics. Another study[10] trained a feedforward neural network to predict sequences of discrete states. The stimulus comprised[15] states that formed 3 "communities"; transitions between states within a community were common, whereas transitions that crossed communities were rare. This structure can be viewed as an abstraction of situations where larger events such as making a sandwich, making coffee, and making juice each includes collections of sub-actions such that transitions amongst sub-actions are more common than transitions between events. The network's hidden unit representations mirror the community structure and the similarity relations found in left IFG, insula, left ATL, and left STG[10]. One study[11] that trained a recurrent neural network to predict human motion showed that gating information to the network by detecting context changes (peaks in prediction error) improved prediction performance. These studies suggest that models of event comprehension should be able to: 1) represent visual features in an abstract space, 2) predict the future and use prediction error as feedback, and 3) use contextual cues to guide prediction.

Structured Event Memory (SEM)[9] embodies these three features: 1) SEM compresses and abstracts the high-dimensional visual input (pixels) into a lower-dimensional vector space by a variational autoencoder[15] (VAE), 2) it uses recurrent neural networks to learn to predict activity sequences and back-propagate errors, 3) it detects contextual change and updates working event

model to guide prediction via approximate Bayesian clustering. SEM has been used to model how activities are segmented into events, how working memory is updated at event boundaries, how a category of event generalizes to a new instance, and how event structure organizes long-term memory on short activities.

Previous models of event comprehension have been subject to two key limitations. First, though abstraction can help models learn generalized event dynamics, those representations were abstracted to the point that they cannot be compared meaningfully with human perceptual representations. Previous simulations have used arbitrary localist codings[12], highly simplified pose representations[11], or unstructured scene representation[9]. Second, the activities used for training and evaluation have been too brief to capture the naturalistic structure of human action performance and comprehension. These limitations have precluded answering a key question about the modeling of event comprehension: can a predictive system that updates event models based on contextual change and operates on the feature space that is comparable to human vision learn event representations that lead to human-like event segmentation and categorization?

To answer that question, we adapted the SEM architecture so that it could be trained and evaluated on rich representations of naturalistic human goal-directed activity, which were recorded and processed such that the model's outputs and internal states could be directly compared to human performance on a moment-by-moment basis. The model was trained on over 19 hours of recordings of actors completing extended naturalistic activities. Each activity was recorded with three video cameras and an infrared time-of-flight depth sensor that captured the three-dimensional pose of the body. The identities and positions of objects were tracked over time using semi-automated object tracking, and semantic information about objects was incorporated using a large-scale language model16. These recordings were subjected to a

dimension reduction process that preserved interpretable information about the dynamics of body movement and the semantics of the objects with which the actor interacted. The computational model was trained to take in the running sequence of these reduced representations and to predict the next timepoint in the sequence, 1/3 of a second later. Here, we report how a model embodying three key computational principles learned, segmented activity, and categorized scene vectors, and we compare the model's segmentation and categorization to human judgments.

Methods

Model architecture and implementation

The core architecture of the Structured Event Memory (SEM) model has two main components: a library of recurrent neural network (RNN) schemas, and a generative model of event labels9. The specific RNN architecture was a four-layer, fully-connected neural network with gated recurrent units (GRU), a leaky rectified linear activation function (leaky ReLU), and 50 percent dropout for regularization. The generative model clusters each incoming scene vector to an event schema by inferring which latent state (event schema) generates the scene vector, and it infers the latent state via local maximum a posterior (MAP) estimation (Fig.1). For each incoming vector, SEM computes likelihoods that the vector belongs to each event schema by comparing event schemas' predictions with the scene vector, with higher similarity indicating higher likelihood. SEM-1.0 generated priors for the vector belonging to event schemas through the sticky Chinese Restaurant Process (sCRP). In SEM-2.0, we replace this process with a sticky uniform process (sUP), which is equivalent to an sCRP modified to treat the size of all visited clusters as equal to a constant. As in the sCRP, sUP has a hyperparameter called stickiness that

controls the tendency to remain in the currently active event, and a hyperparameter called concentration that controls the likelihood of spawning new event schemas. Priors and likelihoods are used to compute posterior probabilities for all event schemas, and the incoming scene vector is assigned to the event schema with the highest posterior probability. Consequently, in this architecture, event boundaries are by-products of switches between different event schemas. In principle, this inference over the latent states by estimating local-MAP is not exactly Bayesian inference, which requires computations for all past clustering outcomes. However, comparisons between local-MAP and more exact forms of Bayesian inference have shown their performance to be highly similar33.

In SEM-1.09, there were three sources of bias (modeling assumptions) that created an imbalance in the relative activation of event schemas. One source of bias was that newly spawned event schemas were initialized to random weights. This initialization disadvantages new event schemas for the learning of naturalistic activities like the META corpus, because the environment is rich with general features and dynamics, such as where objects are typically found and how bodies can move, as well as event-specific information. To address this imbalance, we initialized newly spawned event schemas with weights from a model that was trained on all scene vectors up to that point in time. In addition, the process used to assign priors to event schemas was the sCRP, which assigns higher prior probabilities to latent states (event schemas) that have more frequently been activated in the past. This led to the activation of a small number of event schemas for most time points and rarely activated newly-spawned event schemas. Removing this "rich-get-richer" property helped SEM-2.0 to use event schemas more evenly. Furthermore, SEM-1.0 asks active schemas to make predictions about the current scene by feeding them scene vectors from previous timepoints while asking inactive schemas to make predictions by feeding

them a random vector. This approach helped the authors circumvent a computational challenge because predictions from inactive schemas could be cached (because the random vector was constant, predictions were also constant) and used to compute likelihoods for the inactive event schemas. However, that approach placed inactive event schemas at a disadvantage because the input vectors to these event schemas were not informative to predict the current scene vector, while the input to the active event schemas was the scene vector from previous timesteps. Consequently, inactive event schemas were less likely to be selected and update their weights (since only the active event schema updates its weights at a specific timestep), resulting in only some initial event schemas activating and updating their weights most of the time. Relatedly, because event schemas in SEM-1.0 were trained to predict current scene vectors from either previous scene vectors or random vectors, their predictive power was compromised. We therefore modified SEM-2.0 so that both active and inactive schemas were provided with the previous scene vectors as input. To retain efficient processing, we parallelized the calculation of predictions from active and inactive schemas. These changes led to more even use of event schemas and reduced prediction error (see SI).

All code was implemented in Python, using the Keras library for neural network implementation [link to github repo]. Hyperparameters were chosen by performing a grid-search across several potential values and selecting the configuration of values that minimized prediction error and most closely matched the mean number of human event boundaries (see Supplementary Information).

Materials

SEM-2.0 was trained on the Multi-angle Extended Three-dimensional Activities (META) stimulus set17. This stimulus set contains over 25 hours of performances of everyday activities

of about 10 minutes each, performed in realistic environments (see SI for more information). Performances were captured with a Kinect V2 device, which includes a video camera and a time-of-flight depth sensor34, and two other video cameras. The Microsoft Kinect for Windows SDK 2.0 was used to infer the three-dimensional positions of the actors' skeletal joints from the recorded depth image stream. The three-dimensional joint positions for each frame were translated to place the mid-spine joint at the origin (0,0,0). Joint coordinates were then rotated about the Y-axis to align the left and right shoulder joints on a common plane in the Z-axis. Raw features were smoothed with a rolling mean of seven frames (three frames before and three frames after). From the joint position data, we calculated joint velocity and acceleration, as well as the inter-hand distance, velocity, and acceleration.

Semantic information about objects that the actors touched was captured using the following method. For a subset of video frames at ten seconds intervals, human annotators marked the positions and identities of objects with bounding boxes. Then, using the subset of labeled frames, a computer vision tracking model was used to track the positions of objects both forward and backward in time between the labeled frames (Siam Region Proposal Network35). Each tracker was dropped when the model's confidence fell below a threshold and the Hungarian algorithm was used to match forward and backward tracks. Object appearance and disappearance features were binary, taking values of ones for frames in which at least one object begins to be tracked and frames in which at least one object is no longer tracked, respectively. For each frame, the name of each object present in the scene was converted to a 50-dimensional vector using the GloVe language model16 trained on the large collection of Internet encyclopedia entries and news articles of the Wikipedia 2014 and GigaWord 519 corpora. A 50-dimensional feature set was created as an average of the embeddings of all objects present in the current frame. In

addition to these features representing the semantic meanings of objects in the environment, we created features of the semantic meaning of objects nearest to the actor's right hand. The three-dimensional Euclidean distance between the actor's right hand joint and the average depth in Z axis of the pixels in the object's bounding box was calculated. Then, a weighted average of the object vectors was calculated, scaled by the inverse of the distance to the actor's hand. In total there were 102 object-related features, comprising object appearances and disappearances, 50 features for the average embedding of all the objects present in the scene, and 50 features of the nearest objects to the actor's hand.

We performed principal component analysis to reduce the dimensionality of the feature vectors. Dimensionality reduction was performed separately on body motion and semantic features, to allow for modality-specific calculations of predictions and errors. The resulting set of features contained 30 dimensions (14 body motion dimensions, 13 semantic dimensions, 2 dimension for object appearances and disappearances, and 1 dimension for the correlation of pixel luminance between successive video frames). This dimensionality reduction preserved 76 percent of the original variance of the full feature set.

Training and Testing regimen

The Kinect body tracking produced large-scale errors for some of the activities (for example, in activities where the actor's lower body was occluded, the Kinect mistook the actor's upper body as the whole actor). To prevent activities with large tracking errors from having an undue influence on the model's performance, we developed a filtering algorithm to select high-quality activities for training and validation. First, we calculated the range of all possible values for each derived body motion feature. The algorithm has two thresholds to: 1) mark bad features and 2) filter bad activities. For each feature in each activity, the feature was considered good if y% of

the values for that feature fell inside of the ninetieth percentile of the possible range, otherwise it was marked. If more than x% of all features of the activity were marked, the activity was filtered. A grid-search was used to determine values for x and y. See Fig. 7 for the proportion of activities that are left after applying each pair of x and y thresholds. The x-axis and y-axis in the plot correspond to threshold x and threshold y respectively, and the annotated values indicate the proportion of activities left. Because SEM-2.0 showed learning saturation after watching approximately 40 to 60 activities, we can afford to apply stringent thresholds to maintain high quality without worrying about depleting the training dataset. As a result, we ended up with 128 activities out of the 149 activities in the META stimulus set (86%); an activity is selected if more than 80% if its features are considered good, and a feature is considered good if more than 80% of its value fall within the ninetieth percentile of the possible range.

Filtering thresholds for activity inclusion

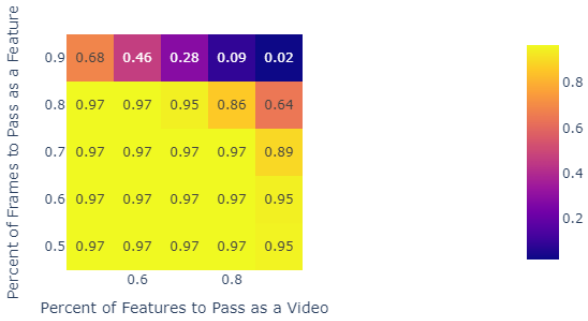

Figure 1: The proportion of activities passing the filtering algorithm with different combinations of two thresholds. Y-axis represents the proportion of frames falling within the ninetieth percentile so that a feature can be marked as good. X-axis represents the proportion of good features so that the video can be used for training and testing. Each annotated number is the proportion of activities passing the filtering algorithm with each combination of threshold

We split the 128 activities into a training set (108 activities) and validation set (20 activities). SEM-2.0 watched activities from the training set, and at each time step the weights in the active RNNs in SEM-2.0 were updated by back-propagating the squared error between SEM-2.0's

predictions and the input scene vectors. In this way, SEM-2.0 was only permitted to learn from each training activity once, to match the uniqueness of experience of humans. We evaluated SEM-2.0 after it watched each training activity by presenting SEM-2.0 with all validation activities, while freezing SEM-2.0's parameters to prevent weight updating.

Generic model architecture

The SEM model builds a library of RNN event schemas and chooses between them using a Bayesian inference process over latent states. For comparison with SEM-2.0, we created a generic model consisting of a single RNN and removed the Bayesian inference mechanism. In this way, we tested a system that applies a single RNN event schema to predict the next time step against the full SEM-2.0 model which selects between separate RNNs to predict the next time step. Apart from this key difference, the parameters were matched between SEM-2.0 and the generic model. SEM-2.0's library of RNNs adds a larger number of parameters above the generic model's but the same amount of weight updating (learning) on each time step; the exact number of weights depends on the number of event schemas created. To address this issue, we also created variations of the generic model with double and triple the number of units in the hidden layer. This increases the number of weights to be more comparable with the number of weights in SEM-2.0—but gives the generic model a large advantage in the amount of weight updating it experiences.

Evaluation Metrics

Whole prediction error

In order to measure how a model learns to predict over the course of training, we calculated its prediction error for each pass through the validation set. The prediction error for each timestep is the Euclidean distance between the active model's prediction and the input scene vector at that

timestep, and the summarized prediction error for the validation set is the average of the prediction errors calculated at all timesteps for all activities.

$$PE = \frac{1}{|activities|} \sum_{a \in activities} \frac{1}{|timesteps|} \sum_{t \in timesteps_a} \sqrt[2]{\left(v(t)_{predicted} - v(t)_{input}\right)^2};\ v \in R^{30} \quad (1)$$

Parcellated prediction error

Because we performed dimensionality reduction separately on body motion features, object semantics, object appearance or disappearance, and optical features, we could calculate predictions and prediction errors specifically for each of those. The resulting set of features contained 30 dimensions (14 body motion dimensions, 13 semantic dimensions, 2 dimensions for object appearances and disappearances, and 1 dimension for the correlation of pixel luminance between successive video frames). Prediction error for a specific modality at each timestep is the Euclidean distance between the model's prediction for that modality and the input scene vector for that modality. For example, prediction error at each timestep for body motion features is the Euclidean distance between a 14-dimensional prediction vector, which is a subset of the model's prediction, and a 14-dimensional input vector, which is a subset of the input scene vector.

Segmentation

An event boundary for SEM occurs when it switches from one event schema to another event schema, or when it decides to restart the current event schema. The generic model only has one event schema, so this definition of event boundaries is not applicable. To simulate event boundaries for the generic model, we identified peaks in prediction error. Peaks were local

maxima selected in descending order of height. The number of peaks was matched to SEM-2.0's number of event boundaries for that validation activity. To have a fairer comparison between the generic model and SEM-2.0, we applied the same algorithm to SEM-2.0's prediction errors to derive event boundaries. The generic model's boundaries, SEM-2.0's actual boundaries, and SEM-2.0's derived boundaries were compared against human boundaries to calculate point-biserial correlations.

Mutual Information

To quantify SEM-2.0's categorization agreement with human categorization, and how SEM-2.0's event schemas generalize to actions performed by the same actor in different instances and by different actors at different locations, we computed the mutual information score (with adjustment to account for chance36) between the model event labels and script action labels every time the model completes watching a training activity. We treated script action labels as one clustering of input vectors, and model's event labels as another clustering of input vectors. Formally, given a stimulus set Corpus with N input scene vectors (the total number of input scene vectors across all validation activities), and two partitions of Corpus, namely Schemas (SEM-2.0's categorization) and Actions (humans' categorization):

$$Corpus = \{v_1, v_2, \ldots, v_n\};\ v_i \in R^{30} \quad (2)$$

$$Schemas = \{schema_1, schema_2, \ldots, schema_n\};\ schema_i = \{v_i, v_j, \ldots, v_k\} \quad (3)$$

$$Actions = \{action_1, action_2, \ldots, action_m\};\ action_i = \{v_{i'}, v_{j'}, \ldots, v_{k'}\} \quad (4)$$

The mutual information score between Schemas and Actions is computed as:

$$I(Schemas;\ Actions) = \sum_{y \in Schemas} \sum_{x \in Actions} P(schema,\ action) \log\left(\frac{P(schema, action)}{P(schema)P(action)}\right) \quad (5)$$

Concretely, equation 5 is a summation over all schema-action pairs. An example of a schema-action pair is shown in Fig. 8, and the calculation for that pair is below:

$$P(Event\ 4;\ Jumping\ Jacks) \log\left(\frac{P(Event\ 4;\ Jumping\ Jacks)}{P(Event\ 4)P(Jumping\ Jacks)}\right) \quad (6)$$

Equation 6 was repeated and summed for all pairs of SEM-2.0 event schemas and scripted action labels to derive mutual information score.

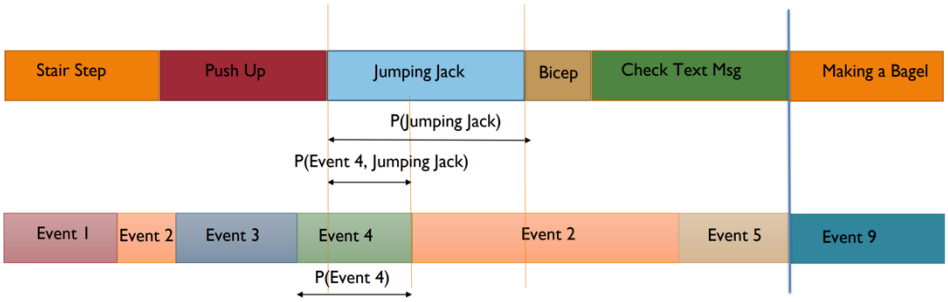

Figure 2: Conceptual example of the calculation of mutual information. Top: a sequence of human actions. Orange vertical line indicates the start of one action and the end of another action. Blue vertical line indicates the transition between two activities (since all validation activities were concatenated to compute mutual information score). P(Jumping Jack) and P(Event 4) are marginal probabilities of jumping jack scenes and scenes that were assigned to SEM's event 4. P(Event 4, Jumping Jack) is the joint probability of scenes assigned to event 4 and jumping jack. In this illustration, the result of equation 6 will be high since there is a high correspondence between event 4 and jumping jack.

For each pair of $Schema_i$ (e.g. SEM-2.0 event schema "4," meaning the 4th schema created by SEM-2.0 during training) and $Action_j$ (e.g. script action label "jumping jack"), P(event 4, jumping jack) is the probability that an input scene vector belongs to both clusters event 4 and jumping jack. If cluster event 4 corresponds with cluster jumping jack (a large number of timesteps are labeled as both SEM-2.0 event schema 4 and script action label jumping jack), the ratio within the logarithmic function will be high and the term for this pair will also be high. The

mutual information score can also inform us about SEM-2.0's ability to generalize across actors and activities. A given action (e.g. jumping jack) can be performed by different actors in different environments. If SEM-2.0 is able to generalize across actors and environments, it should assign the same event label (e.g. event 4) to those input scene vectors. In that case, the adjusted mutual information score will be high. In contrast, if SEM-2.0 assigns different event labels to the same action performed by a different actor in a different room (event 4 to actor A jumping jack and event 20 to actor B jumping jack), the score will be low. To the degree that SEM-2.0 categorizes input scene vectors in a human-like way, and its event schemas generalize to instances of the same action, adjusted mutual information between two partitioning algorithms will be high.

Permutation testing

We assessed how likely the results would occur by chance via permutation testing. SEM-2.0's event labels for all validation activities were first concatenated. A permutation is generated by shuffling runs of event labels, thus preserving event lengths in the resulted permutation. The shuffling not only changes SEM-2.0's event labels for particular input scene vectors but also changes SEM-2.0's event boundaries. As a result, the shuffling procedure can be used for both segmentation and categorization tests. When we concatenate two validation activities, there is an interval between the onset of SEM-2.0's last event in the first activity and the onset of SEM-2.0's first event in the second activity. Because human event boundaries are less likely to fall into these intervals, and SEM-2.0 never placed boundaries in these intervals, we made sure permutations did not have boundaries within these intervals so that SEM-2.0 would not have an unfair advantage over permutations. Permutations were repeated 100 times and scaled point-

biserial correlation, adjusted mutual information, purity, and coverage were computed for each permutation.

Training and testing input-deprived models

To test the ability of the model to learn representations with limited input, we trained two versions of SEM-2.0, each version with eight random initializations, that generated predictions of the full scene vector from deprived input features: we withheld either the body motion features or the semantic features. The input scene vector is a concatenation of multiple PCA-ed vectors: a 14-dimensional vector for body motion features, a 13-dimensional vector for semantic feature, a 2-dimensional vector for object appearance/disappearance feature, a 1-dimensional vector for optical flow feature. For the semantics-deprived model, we set the 13-dimensional vector for semantic features to zeros before feeding into the model. For the motion-deprived model, we set the 14-dimensional vector for body motion features to zeros before feeding into the model. The two models still had to make predictions for all features: 30-dimensional output vectors. We evaluated deprived models on the aforementioned metrics: prediction error, scaled point-biserial correlation, and adjusted mutual information; we used permutation testing to assess the statistical significance of segmentation and categorization metrics.

Results

Models of event comprehension were trained and tested on the Multi-angle Extended Three-dimensional Activities (META) stimulus set, a corpus of naturalistic activities17. In each activity, an actor performed a series of 6 to 7 scripted actions in a realistic environment. Because the visual and semantic features processed in mid-level human vision may be the building blocks from which event representations are constructed18, we used a combination of human and computer vision methods to generate a rich set of these features. From three-dimensional joint position recordings, we calculated features of body pose, velocity and acceleration, as well as inter-hand distance, velocity, and acceleration. To represent the semantic meanings of interactive objects in the activities, we annotated bounding boxes that tracked the positions of objects, then used a language model (GloVe16) trained on a large text corpus19 and translated the name of each object to a vector embedding. We then computed a weighted vector representation of the objects closest to the actor's right hand, and the mean vector representation of all objects currently present in the scene. Principal component analysis reduced a set of 253 input features (object appearances, object disappearances, mean frame-to-frame change in pixel luminance values, skeletal motion features, object semantic features) to a set of 30 features that we presented as input scene vectors to the event prediction model.

The core architecture of the model, depicted in Fig. 1, was modified from the Structured Event Memory (SEM) model9; we will refer to the modified version as SEM-2.0 and the original version as SEM-1.0 (SEM refers to both SEM-1.0 and SEM-2.0). This model uses fully-connected recurrent neural networks (RNNs) to represent event schemas, and an approximate

Bayesian inference (clustering) process to assign incoming scene vectors to event schemas. On each time step (3 Hz), a currently active RNN is presented with an input scene vector and predicts the next scene vector. The clustering process then compares the posterior probability of the active RNN's prediction, relative to that from all other models, and then either 1) retains the current event model, 2) activates a different event schema from the library, or 3) spawns a new event schema. SEM-2.0 includes hyper-parameters for stickiness, the tendency to keep the active model (to ensure temporal coherence in events), concentration, the tendency to spawn new models, and learning rate to update RNNs. The hyper-parameters used for our experiments are shown in Table 1, and hyper-parameter tuning is described in Supplementary Information. In SEM, the set of RNN weights that are learned during training are event schemas, and each activation of a schema constitutes the construction of a working event model.

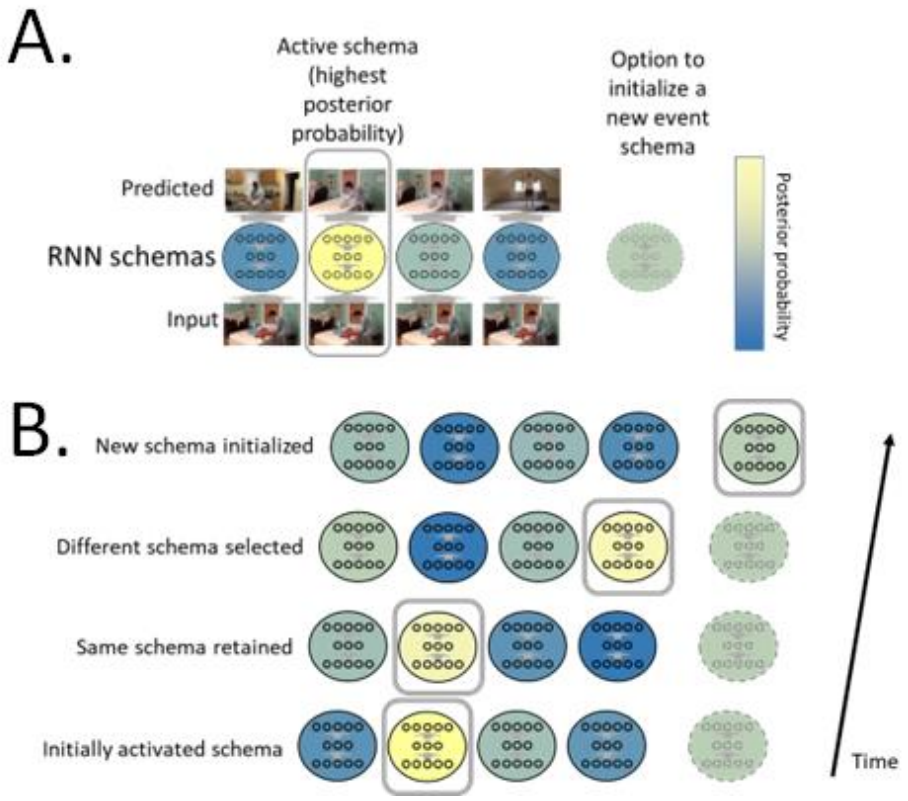

Figure 3: Overview of SEM architecture. (A) In this hypothetical example, SEM is in the process of training and has generated four event schemas (RNNs). At each time step, the event schemas predict the current input scene vector from the previous scene vector. Based on the posterior probability, SEM keeps the active event schema active, switches to another schema in its library, or spawns a new event schema (depicted as an RNN with a dotted outline). In this case, the active schema is retained. The resulting active event schema updates its weights by backpropagating its prediction error. (B) A potential sequence of outcomes. On the first two timesteps, the currently active schema is retained. On the third timestep, SEM switches to a different previously-learned schema. On the fourth timestep, SEM initializes a new schema. (Not shown: SEM separately evaluates the probability of the current schema based on the current RNN hidden unit values and based on re-initializing the RNN. If resetting the hidden unit values is found to be more valuable, they are reset. This allows SEM to model, for example, washing a plate and then immediately washing a second plate.).

Learning Rate	Stickiness	Concentration	Input vector dimensionality	Number of RNN hidden units
1e-3	1e7	1e-1	30	16

Table 1: Model's hyper-parameters.

Applying the original implementation (SEM-1.0)[9] on the corpus dataset, we noticed that the model only used a couple of event schemas to account for most of 22 hours of activities, which wasn't the issue in the original dataset of short videos (average of 4 minutes) that SEM-1.0 was trained and tested on. There were three sources of bias (modeling assumptions) that created an

imbalance in the activation of event schemas when the model interacted with the extended naturalistic activities. First, newly spawned event schemas were initialized to random weights; this disadvantages new event schemas for the learning of naturalistic activities that afford rich "general knowledge" about feature co-occurrence and dynamics. In SEM-2.0, we initialized newly spawned event schemas with weights from a single RNN that was trained on all scene vectors up to that point in time. Second, the process SEM-1.0 used to assign prior probabilities to schemas was the sticky Chinese Restaurant Process20 (a type of Dirichlet process). Dirichlet process is a commonly-used prior distribution and has a "rich-get-richer" property21—a small number of large clusters accounts for most observations. In SEM 1.0, this property caused most timepoints to be assigned to only a small number of event schemas. Although "rich-get-richer" might be appropriate to some clustering applications, this property might not be desirable in certain applications where a more balanced prior distribution is desired21,22. In SEM-2.0, we instead used a uniform prior distribution, while retaining stickiness and concentration parameters. Third, SEM-1.0 asked active schemas to make predictions about the current scene from the current scene vector, but it asked inactive schemas to make predictions from a random vector. This approach is computationally efficient, but it puts inactive schemas at a disadvantage. In SEM-2.0, we feed all schemas scene vectors from previous timepoints (see Methods for details).

From 128 activities (total duration: 21 h 43 m, range: 5 m 35 s to 19 m 16 s, mean: 10 m 11 s), activities were randomly split into a training set of 108 activities (18 h 4 m) and a validation set of 20 activities (3 h 39 m). In contrast to the common practice with deep learning models of interleaving learning with repeated presentation of stimuli23, SEM-2.0 encountered and learned each training activity only once, watching the whole activity before moving to the next activity.

This strategy resembles blocked training regime and imitates the uniqueness of visual stimuli. Both of these features are characteristics of human learning experience, enabling a meaningful comparison of the model's outputs to human's segmentation and categorization. After training on each activity, the validation set was tested with learning turned off.

SEM-2.0 learns to predict naturalistic scene dynamics and outperforms comparison models

The time course of prediction error for the trained SEM-2.0 model contains spikes of high prediction error punctuating periods of stable predictions (Fig. 2A). SEM-2.0's mean prediction error for validation activities decreased over the course of training (Fig. 2B, top), as SEM-2.0 partitioned event knowledge into discrete recurrent neural networks and used a Bayesian updating process to select the active event schema. To assess the impact of event knowledge partitioning, we created a generic model, composed of a single RNN that predicted the incoming scene vector from the last input. This model used the same parameters as SEM-2.0 for its one event schema, but could not switch or spawn new event schemas. The generic model had higher mean validation prediction error across 16 simulations (SEM: 0.575, 95% CI (0.571, 0.579); generic: 0.636, 95% CI (0.622, 0.651); also see Figure 2B and 2C, top panels, for 4 simulations).

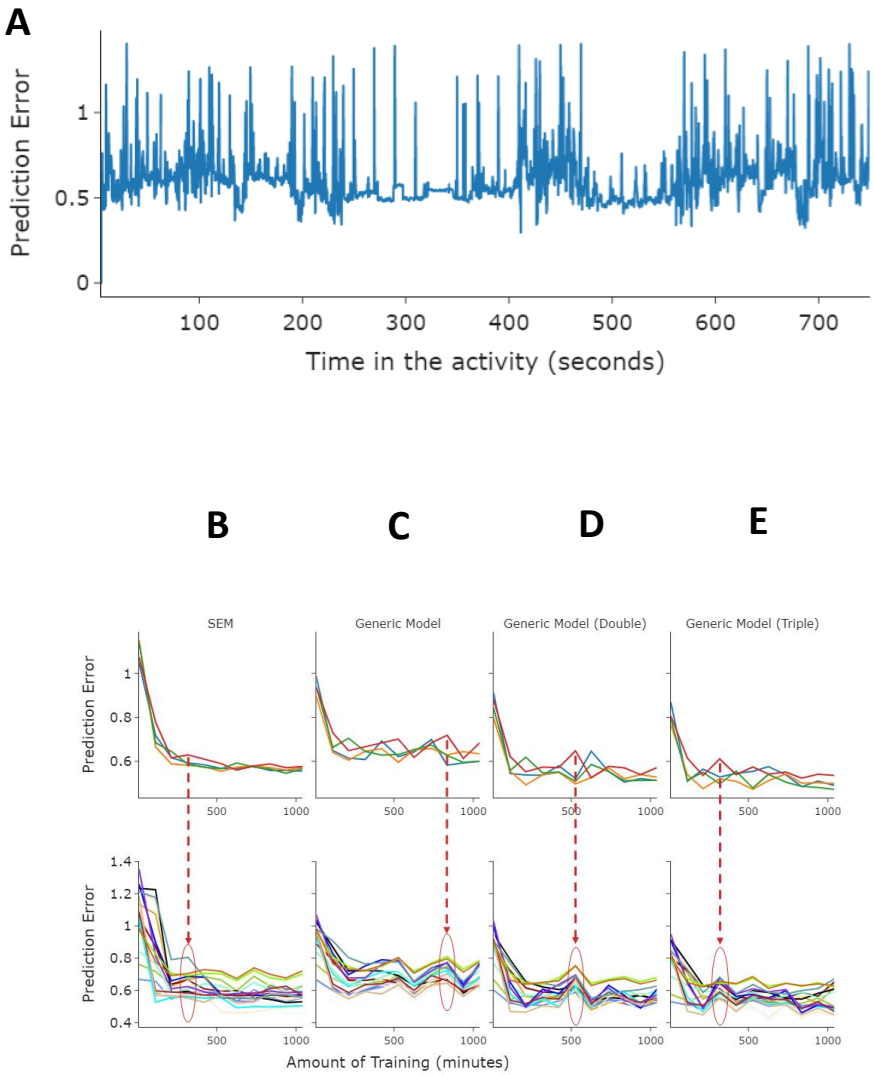

Figure 4: Prediction Error performance. (A) Prediction error of SEM-2.0 shows regular spiking. Illustrated here is prediction error for one cleaning activity in the validation set after training on approximately 450 minutes of other activities in the training set. (B-E) Prediction errors for SEM-2.0 and generic models over the course of training. Each point is the mean prediction error for all validation activities at each evaluation time over the course of training. (B-E) Top: Mean prediction error for all validation activities across training for SEM-2.0, generic model with the same, double, and triple the number of hidden units. Four colors indicate four different simulations with different random weight initializations and different orders of training activities. Using discrete event schemas and a Bayesian updating process, SEM-2.0 reduces prediction error over the course of training. The generic model reduces prediction error over the course of training, to a level of prediction error slightly higher than SEM-2.0. Generic models with double and triple the number of hidden units can reduce errors lower than SEM-2.0; however, all generic models show greater interference from new learning (mean prediction error fluctuates across training). Bottom: Prediction errors for all validation activities across training for SEM-2.0, generic model with the same, double, and triple the number of hidden units, for the "red" simulation. SEM-2.0 shows some interference around minute 300-th, with prediction errors for a couple of validation activities increase. However, the generic model shows catastrophic interference around minute 800-th, with almost all validation activities' prediction errors increase. The same pattern can be observed in the generic models with double and triple number of hidden units.

Compared to the generic model, SEM allocates more storage to representing the results of learning because it accumulates a library of event schemas. For comparison, we considered expanded generic models had double (2x) or triple (3x) the number of units in the hidden layer of the model. Notably, these models experienced much more weight updating than the generic model or SEM; whereas only the active model in SEM is able to update its weights, the original and expanded generic models update all their weights on each time step, which increases the amount of weight updating 2.5 times and 4 times, respectively. As shown in Fig. 2D-E (top panels), expanding the size of the hidden layer in the generic model also reduced the mean validation prediction error across 16 simulations (2x: 0.545, 95% CI (0.537, 0.554); 3x: 0.511, 95% CI (0.505, 0.518)). Thus, whereas adding the ability to switch event schemas reduces prediction error, this set of simulations demonstrate that it is also possible to reduce prediction error by increasing the hidden layer size. However, there are differences in the weight updating processes between SEM and the generic models: whereas the generic models must update its weights with each new input scene, the schema weight updating mechanism in SEM is able to silo data from a newly-encountered event without compromising the integrity of the other event schemas in its library. This reduced interference: As shown in Figure 2C-E (top panels), the generic models' average prediction error for validation activities sometimes increased as they saw more training activities, suggesting that the new learning from recent training activities had interfered with previously acquired learning that was beneficial to predict validation activities. Figure 2C-E (bottom panels) shows prediction errors for all validation activities across training for the "red" simulation. The generic model shows catastrophic interference at many points across training (e.g. around minute 510-th or minute 800-th), with almost all validation activities'

prediction errors increase. The same pattern can be observed in the generic models with double and triple number of hidden units.

To quantify the benefit of modeling scene dynamics with an RNN, we created a pair of very simple comparison models: The last scene model simply used the last scene vector as its prediction for the current scene, instead of generating a prediction as the output of an RNN. The recent scene model used a moving average of the previous three scene vectors as its prediction for the current scene vector. Both models performed poorly compared to SEM (mean prediction errors of 2.15 and 2.33, respectively.)

SEM-2.0 segments activities in a human-like fashion without being reinforced for segmentation

For each timestep, SEM selects an RNN to remain active or become active; this can be interpreted as an event label categorizing that timestep. For each event label en, SEM assumes the event schema en is active and generates a predicted scene vector conditioned on en. The probability of assigning event label en to the input scene vector monotonically decreases with the difference between the input scene vector and the predicted scene vector generated by event schema en. Moreover, for the active event label (the event label assigned to the previous scene vector), SEM calculates two probabilities: the probability of observing the input scene vector if the active event continues, and the probability of observing the input scene vector if the active event restarts. An event boundary is inferred when event labels for subsequent scene vectors are different, or when the probability of restarting the active event is higher than the probability of continuing. Events are the intervals between event boundaries.

A key test of the model is to determine whether it generates human-like event boundaries for naturalistic stimuli. To compare SEM-2.0 to human performance on event updating and

understanding, we used data from the META stimulus set17. Normative event boundaries were collected from an online sample of participants. Participants were instructed to watch a randomly-selected sequence and press a button each time one meaningful unit of activity ended and another began. Each participant was assigned a grain of coarse, defined as the largest meaningful units of activity, or fine, defined as the smallest meaningful units of activity. Participants could segment multiple videos. We collected 30 segmentations per grain per activity.

To quantify model-to-human and human-to-human segmentation agreement, we calculated the proportion of human raters who segmented during each timestep, and computed the point-biserial correlation between that normative human segmentation time series and (a) each individual human rater, and (b) SEM-2.0 models' segmentation. The possible range of this correlation depends on the number of event boundaries; thus, comparing correlations for two segmenters who identify different numbers of event boundaries can be misleading. Therefore, we scaled the correlation24 based on its minimum possible and maximum possible values, given the number of boundaries observed. We also assessed how likely the result would occur by chance by permutation testing: shuffling event boundaries while preserving event lengths (see Methods). As shown in Figure 3A, SEM-2.0's point-biserial correlations across training were much larger than would be expected by chance, and are within the lower end of the distribution of human-to-human segmentation agreement.

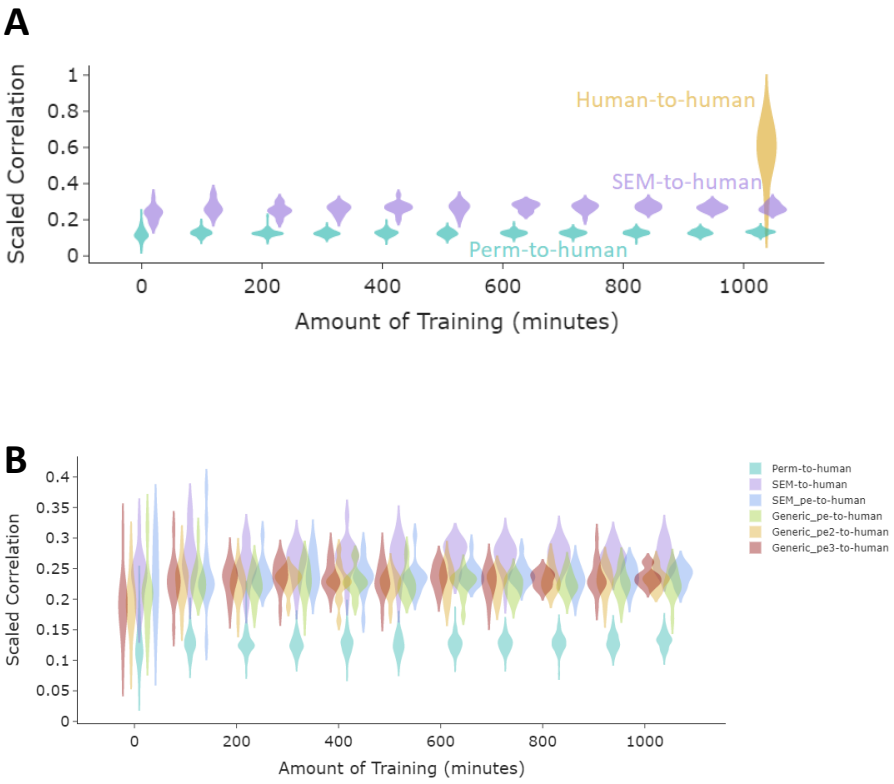

Figure 5: Compare segmentation agreement with a human normative group between SEM-2.0 and individual human segmenters, generic models. (A) Scaled point-biserial correlations across all validation activities for SEM-2.0 simulations, humans, and permutations across training. Each purple violin plot is a distribution of point-biserial correlation for different initializations of SEM-2.0. Each light sea-green violin plot is a null distribution generated by shuffling SEM-2.0's boundaries while preserving SEM-2.0's event lengths. The goldenrod violin plot is a distribution of scaled point-biserial correlation for different human subjects (which doesn't change over the course of training). SEM-2.0's segmentation agreement with human segmenters is bigger than expected by chance, and falls within the lower end of human segmenters. (B) Comparison of agreement with human segmentation between SEM-2.0 and generic models. SEM-to-human denotes SEM-2.0's boundaries derived from Bayesian inference (event label switches). SEM_pe-to-human denotes SEM-2.0's boundaries derived from its prediction error. Generic_pe-to-human, Generic_pe2 -to-human, and Generic_pe3 -to-human denote boundaries derived from prediction errors in the generic model, the generic model with double and triple the number of hidden units. Each violin plot is a distribution of scaled point-biserial correlation for different initializations and training orders. Segmentation agreement between generic models and humans is bigger than expected by chance, though it is smaller than segmentation agreement between SEM-2.0 and humans.

SEM-2.0's segmentation agreement out-performs generic models

Because the generic model has only one event schema, it doesn't produce event boundaries. To estimate event boundaries for the generic models, we identified timesteps where its prediction errors were highest. For a fair comparison between the generic models and SEM-2.0, we also

applied the same algorithm to SEM-2.0's prediction errors to obtain event boundaries. We calculated point-biserial correlations for generic model's boundaries and SEM-2.0's boundaries. While SEM-2.0's and generic models' event boundaries derived from prediction errors align with human segmentation significantly more than chance, SEM-2.0's event boundaries derived from the Bayesian inference process (event label switches) had higher agreement with human segmentation than these PE-derived boundaries (Fig. 3B). This result suggests that, although increasing the number of hidden units in the generic model reduces prediction error (Fig. 2C), it does not lead to more human-like event segmentation.

SEM-2.0 produces flurries of updating at some event boundaries

Examining the time course of SEM-2.0's updating reveals that there are moments when SEM-2.0 makes a flurry of rapid updates within a relatively short time window. Figures 4A and 4B show SEM-2.0's boundaries for two example validation activities. Fig. 4C shows the distribution of elapsed durations between consecutive boundaries. The distribution is heavily right-skewed, and approximately 42% of consecutive boundaries have durations below 2 seconds, showing that SEM-2.0 makes flurries of rapid updates within a short time. Even though we know that humans agree where event boundaries are, we don't know if it is the case that the brain experiences one boundary or a series of boundaries. Thus, SEM-2.0 makes the novel prediction that the brain might sometimes experience a series of updates before settling into a new stable event model.

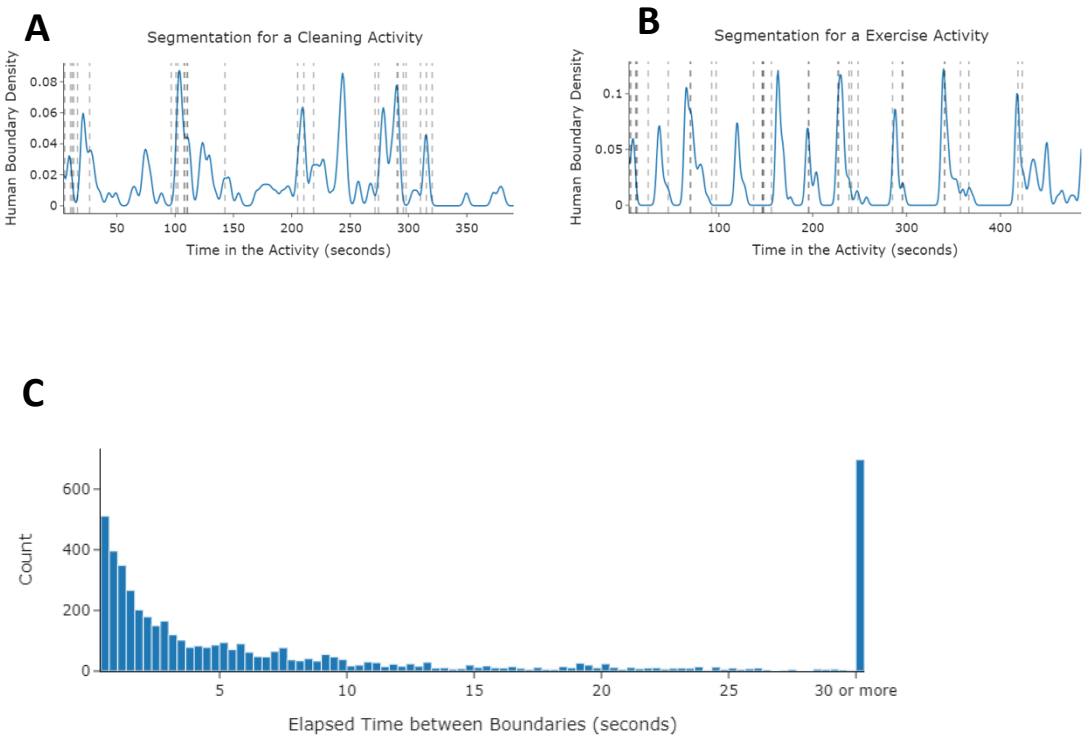

Figure 6: Flurries of updating. Examples of SEM-2.0's boundaries for (A) one "cleaning room" activity and (B) one "exercise" activity. For the cleaning activity, SEM-2.0 made flurries of updates around the 10th and 105th seconds. For the exercise activity, SEM-2.0 made flurries of updates around the 5th and 150th seconds. (C) Distribution of elapsed time between SEM-2.0's consecutive boundaries. Durations longer than 30 seconds were collapsed together. Most of the pairs of consecutive boundaries have small durations, indicating that SEM-2.0 made rapid updates within short intervals.

SEM-2.0 forms schemas that correspond with judges' action categories, and generalizes across actors and environments without being reinforced for categorization

To comprehend an activity, one needs to not only capture its boundaries but also to relate the current activity to previous knowledge. SEM-2.0's event labels model the act of classifying a current moment as an instance of a previously-learned activity. To evaluate SEM-2.0's ability to classify, we used the script action labels that were provided to the actors before recording of each activity. We had two human raters watch videos of the activities and identify the beginning and ending of each of the 6-7 scripted actions per activity. Agreement between raters was high, with a median discrepancy of 1.41 s between raters. Discrepancies were resolved by computing the

mean of the time annotations. We compared human-rated action labels and SEM-2.0's event labels. Fig. 5A shows examples for these script action labels, and Fig. 5B shows SEM-2.0's event labels for one activity.

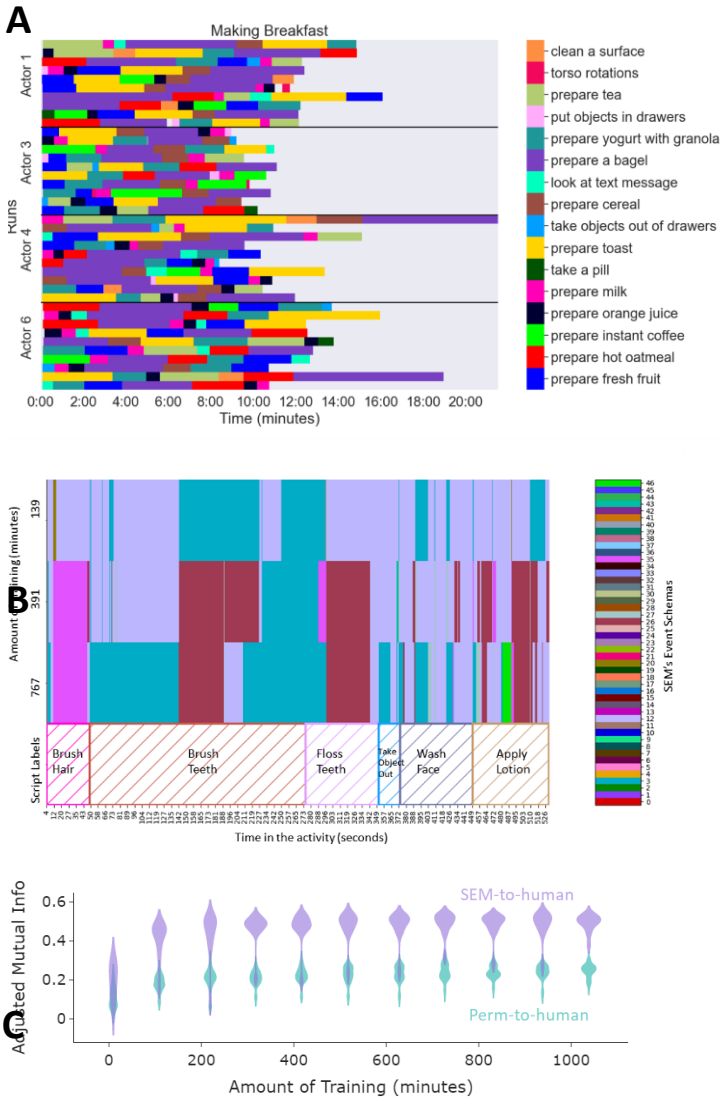

Figure 7: Categorization agreement with script action labels for SEM-2.0. (A) Examples of script action labels for "making breakfast" activities. Each row is an activity, and each group on the y-axis indicates the actor performing these action sequences. Each color represents an action label. X-axis indicates the length of the activity. (B) An example of SEM-2.0's active event schemas for one bathroom grooming activity at three different points (top three rows) in training (139 minutes, 391 minutes, and 767 minutes) and script action labels (last row). (C) Categorization agreement between SEM-2.0 event labels and human action labels. Each purple violin plot is a distribution of adjusted mutual information scores between different simulations of SEM-2.0. Light sea-green violins indicate distributions of adjusted mutual information scores for permutations: each permutation is a shuffle of SEM-2.0's boundaries while preserving event lengths. SEM-2.0's adjusted mutual information scores increase across training, and remains significantly bigger than chance.

To quantify SEM-2.0's agreement with human action categories, we calculated the adjusted mutual information between SEM-2.0's event labels and the scripted action labels. Mutual information quantifies the information shared by the two partitioning (clustering) algorithms (both SEM-2.0 and humans partition input scene vectors into clusters) and thus can be employed as a categorization similarity measure. If SEM-2.0 categorizes input scene vectors in a human-like way, and SEM-2.0's event schemas generalize across instances of the same action, mutual information between SEM-2.0 and script action labels will be high (see Methods). The adjusted mutual information score corrects for the chance level of expected mutual information between two partitions. To test the significance of the adjusted mutual information between SEM-2.0's event schemas and script action labels, we performed a permutation test by randomly shuffling the order of SEM-2.0's events 100 times and computing the adjusted mutual information between SEM-2.0's event schemas and action labels (see Methods). As shown in Fig. 5C, the adjusted mutual information between SEM-2.0's event schemas and script action labels was significantly higher than the permuted null distributions, indicating that SEM-2.0 can form event schemas that generalize across actions performed by different actors in different environments. Examination of the correspondence between SEM-2.0 categories ad script action labels revealed that SEM's categories generalized across actors and environments: The same schema was often activated for the same script action performed by different actors in different environments (see Supplementary information.)

Note that perfect adjusted mutual information is possible only if two partitions have the same number of categories; if two partitions differ in the number of categories (i.e., if one is finer than the other), the best possible adjusted mutual information score is lower. By the end of training, SEM-2.0 partitioned scenes into a lower level than humans do, with a mean event length of

29.47s compared to a mean scripted action length of 79.90 s, thus worsening the score. To better characterize the relationship between SEM-2.0's evolving partitioning and human categories, we used pair of complementary categorization metrics, purity and coverage, which suggested that SEM-2.0's categorization at the end of training captured sub-units of the action script labels (see Supplementary Information).

The rate of event schema formation slows over learning

A notable aspect of SEM-2.0's behavior is the rate at which it forms new event schemas over the course of its training. Fig. 6 shows the number of SEM-2.0's event schemas over the course of training. Early in training, when SEM-2.0 hasn't learned much, it keeps creating new event schemas to capture the statistics of the stimuli. In the middle and late of training, SEM-2.0 starts to reuse its event schemas to accommodate novel stimuli while using these novel stimuli to update the weights of its existing event schemas. This process might resemble how humans learn and use event schemas: acquiring schemas quickly early in development, then relying more on existing schemas when a library of schemas has been established and encountered situations are similar to those previously encountered.

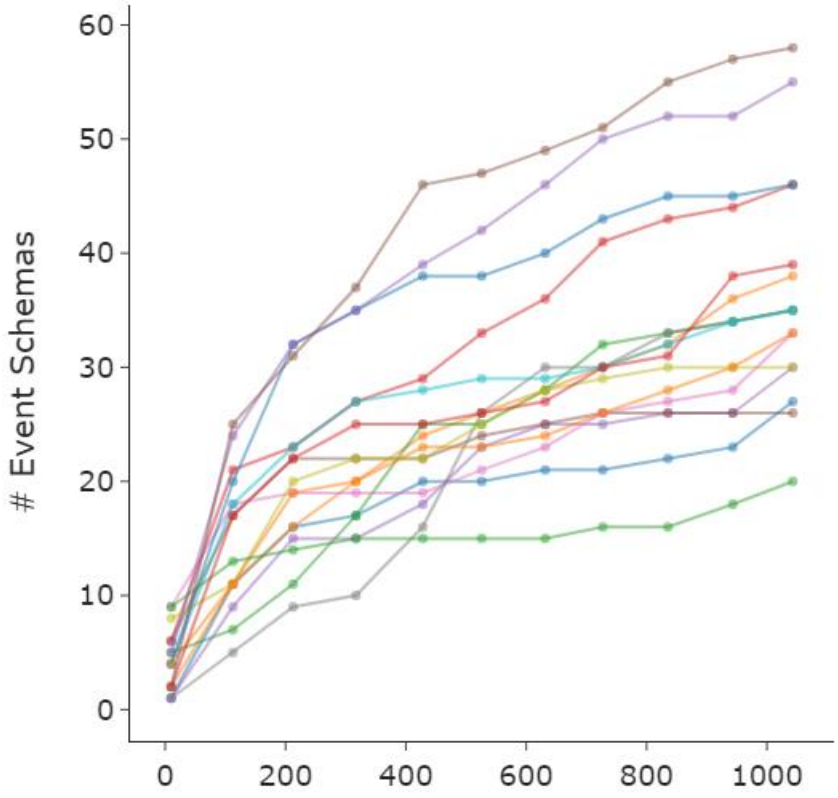

Figure 8: SEM-2.0's number of event schemas across training. Color lines are simulations of SEM-2.0 with different weight initializations and training orders. SEM-2.0 created many new event schemas early in training and reused these event schemas in the middle and late of training.

Comparison with input-deprived models

To quantify how SEM predicts, segment activities, and categorize scene vectors, we created two input-deprived SEM-2.0 versions, semantics-deprived and motion-deprived models (see Methods) and compared them to SEM-2.0. Deprived models have higher prediction error than SEM-2.0, with large contribution coming from respective deprived features. Full SEM-2.0 had higher segmentation correlation with humans and higher categorization agreement with humans than deprived models did (see Supplementary Information).

Discussion

A computational model that can be meaningfully tied to human behaviors and neural mechanisms can provide a framework for understanding human event comprehension. Thus, comparing the model's output and empirical data provides a feasibility test for a computational model of event comprehension. In this research, we tested if a model of event comprehension that embodies three key computational principles—abstraction, prediction, and context—can learn representations to improve predictions, update event models, and categorize scenes in a human-like fashion. We adapted the original SEM architecture[9] so that it can be trained and evaluated on large-scale naturalistic and complex stimuli. Each video is an extended activity that captures the structure and complexity of sequences of everyday activities. SEM-2.0's input features mimic the representations of ventral and dorsal streams of visual perception, which are object categories and biological motion respectively. We have shown that the model can learn to anticipate the next scene, segment events in a human-like manner, form event schemas that correspond to human action labels, and generalize these schemas to new events. The naturalistic dataset provides a rigorous test for SEM-2.0, and these results establish the feasibility of this architecture, and consequently its core computational properties, for event comprehension.

Prediction

SEM-2.0's ability to predict replicated findings that recurrent neural networks are suitable candidate for sequential learning[23,25]. Both the generic model with the same number of hidden units and the generic model with double number of hidden units can learn event dynamics. Notably, we observed that by increasing the number of hidden units in the generic model, effectively increasing its capacity, the generic model can learn to reduce error further. However, the bigger model is more susceptible to catastrophic interference from new data for blocked training scheme[26,27]. In contrast, SEM-2.0's ability to parcel event knowledge into separate RNNs makes it more resistant to interference from new data--it can spawn a new RNN to accommodate new (and possibly noisy) input without messing with the weights of existing

RNNs. This feature of SEM-2.0 distinguishes it with previous models of event cognition[10–12]: in these models, event dynamics was learned by a single neural network instead of a library of neural networks.

Segmentation

The model's segmentation resembled human segmentation on extended naturalistic stimuli. In the segmentation simulation, the input to SEM-1.0 was a high-dimensional vector derived from applying a variational autoencoder[15] (VAE) to video pixels. Even though the features were an advancement from low-dimensional and often artificial stimuli, they do not necessarily represent features that can be meaningfully compared with human perceptual representations. Here, we have demonstrated that the computational principles in SEM-2.0 can account for naturalistic input mimicking human ventral and dorsal streams.

Using peaks in prediction error as event boundaries, the generic model's segmentation can also capture human segmentation behavior better than chance, and poorer than SEM-2.0 does. The finding is unsurprising given that SEM-2.0's segmentation mechanism also relies on prediction errors, and it suggests that human segmentation is sensitive to prediction errors as well.

Generalization

In this research, we have tested SEM-2.0's generalizability against a corpus of naturalistic stimuli. We showed qualitatively that SEM-2.0 reused event schemas to action labels performed by different actors in different environments, indicating that it could learn underlying event dynamics while smoothing surface features. We quantified SEM-2.0's generalizability by the adjusted mutual information score between SEM's event instances and ground truth action labels. Adjusted mutual information score was significantly larger than expected by permutation tests, and it increased over the course of training. The finding makes sense because as SEM-2.0 sees more and more training examples, it should be better able

to extract underlying event dynamics and generalize to novel stimuli. In SEM-1.0, the authors have demonstrated that it generalizes a previously learned event schema to novel stimuli with different fillers but similar underlying relations on a toy dataset. A large corpus of extended naturalistic stimuli offered a stronger test of generalizability, and SEM-2.0 was able to generalize event schemas across ground truth action labels on the corpus dataset.

Extensions

One limitation of this model is that it does not combine known event schemas to generate a new event schema. For example, an event "selling stock at a coffee shop" can include elements of event "selling stock" and elements of event "have coffee at a coffee shop." Adding this ability is equivalent to extending SEM's ability to use contextual information to guide prediction more effectively. To do so, SEM might need to represent events compositionally, decomposing events into elements and being able to combine these elements in a rule-like manner. The resulting model could potentially reduce representational demands (representational dimensions) and generalize better thanks to its flexible combination rules. Elman and McRae[12] demonstrated that a single, large neural network, which does not separate event knowledge, can combine elements learned in different events. The authors trained a neural network model on two different events, one in which a person cuts food in a restaurant with a knife and another where the same person cuts themselves with a knife and bleeds. The authors then gave the model an event in which the person was in the restaurant and cut themselves, and the model correctly inferred that the person bleeds, combining elements from the two events. One limitation of this approach is that the network employs localist representation, which has been laborious and infeasible to code for a large corpus of naturalistic stimuli. However, a recent advance[28] in computer vision presents a way to automate the process, opening a promising research direction.

Another limitation of the current model is that it does not model the hierarchical nature of events, smaller events are grouped to form larger events. For example, if "browsing the menu," "call the waiter/waitress," and "order meals" reliably occur sequentially, one could learn this temporal relationship and form a larger

event, “order food at a restaurant.” Behavioral[29,30] and neural[31,32] findings provide evidence that human event comprehension represents such temporal hierarchy. Franklin et al.[9] discussed two strategies to implement a mechanism of learning hierarchical events. The first strategy is to extend the sticky-CRP process to learn transition dynamics between events, effectively grouping smaller events belonging to a larger event together. Another strategy is to make SEM to learn events of multiple hierarchies simultaneously. In principle, by scaling the prior over event noise parameter for each event schema (RNN), one can change the event schema’s sensitivity to prediction errors, effectively changing its granularity.

References

1. Kuperberg, G. R. Tea With Milk? A Hierarchical Generative Framework of Sequential Event Comprehension. *Top. Cogn. Sci.* **13**, 256–298 (2021).

2. Clark, A. Whatever next? Predictive brains, situated agents, and the future of cognitive science. *Behav. Brain Sci.* **36**, 181–204 (2013).

3. Anderson, R. C. Schema-Directed Processes in Language Comprehension. in *Cognitive Psychology and Instruction* (eds. Lesgold, A. M., Pellegrino, J. W., Fokkema, S. D. & Glaser, R.) 67–82 (Springer US, 1978). doi:10.1007/978-1-4684-2535-2_8.

4. Graesser, A. C. & Nakamura, G. V. The Impact of a Schema on Comprehension and Memory. in *Psychology of Learning and Motivation* vol. 16 59–109 (Elsevier, 1982).

5. Schank, R. & Abelson, R. Scripts, plans, goals and understanding. doi:10.1016/0378-2166(79)90031-6.

6. Bower, G. H., Black, J. B. & Turner, T. J. Scripts in memory for text. *Cognit. Psychol.* **11**, 177–220 (1979).

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

8. Zacks, J. M., Speer, N. K., Swallow, K. M., Braver, T. S. & Reynolds, J. R. Event perception: A mind-brain perspective. *Psychol. Bull.* **133**, 273–293 (2007).

9. Franklin, N. T., Norman, K. A., Ranganath, C., Zacks, J. M. & Gershman, S. J. Structured event memory: A neuro-symbolic model of event cognition. *Psychol. Rev.* **127**, 327–361 (2020).

10. Schapiro, A. C., Rogers, T. T., Cordova, N. I., Turk-Browne, N. B. & Botvinick, M. M. Neural representations of events arise from temporal community structure. *Nat. Neurosci.* **16**, 486–492 (2013).

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

12. Elman, J. L. & McRae, K. A model of event knowledge. *Psychol. Rev.* **126**, 252–291 (2019).

13. Smith, K. *et al.* Modeling Expectation Violation in Intuitive Physics with Coarse Probabilistic Object Representations. in *Advances in Neural Information Processing Systems* vol. 32 (Curran Associates, Inc., 2019).

14. Riochet, R. *et al.* IntPhys 2019: A Benchmark for Visual Intuitive Physics Understanding. *IEEE Trans. Pattern Anal. Mach. Intell.* **44**, 5016–5025 (2022).

15. Kingma, D. P. & Welling, M. Auto-Encoding Variational Bayes. Preprint at http://arxiv.org/abs/1312.6114 (2014).

16. Pennington, J., Socher, R. & Manning, C. Glove: Global Vectors for Word Representation. in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* 1532–1543 (Association for Computational Linguistics, 2014). doi:10.3115/v1/D14-1162.

17. Bezdek, M. *et al. The Multi-angle Extended Three-dimensional Activities (META) stimulus set: A tool for studying event cognition*. https://osf.io/r5tju (2022) doi:10.31234/osf.io/r5tju.

18. Richmond, L. L. & Zacks, J. M. Constructing Experience: Event Models from Perception to Action. *Trends Cogn. Sci.* **21**, 962–980 (2017).

19. Parker, Robert, Graff, David, Kong, Junbo, Chen, Ke & Maeda, Kazuaki. English Gigaword Fifth Edition. 9542041 KB (2011) doi:10.35111/WK4F-QT80.

20. Fox, E. B., Sudderth, E. B., Jordan, M. I. & Willsky, A. S. A sticky HDP-HMM with application to speaker diarization. *Ann. Appl. Stat.* **5**, (2011).

21. Wallach, H., Jensen, S., Dicker, L. & Heller, K. An Alternative Prior Process for Nonparametric Bayesian Clustering. in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* 892–899 (JMLR Workshop and Conference Proceedings, 2010).

22. Welling, M. Flexible Priors for Infinite Mixture Models. 8.

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

24. Kurby, C. A. & Zacks, J. M. Starting from scratch and building brick by brick in comprehension. *Mem. Cognit.* **40**, 812–826 (2012).

25. Elman, J. L. Finding Structure in Time. *Cogn. Sci.* **14**, 179–211 (1990).

26. Goodfellow, I. J., Mirza, M., Xiao, D., Courville, A. & Bengio, Y. An Empirical Investigation of Catastrophic Forgetting in Gradient-Based Neural Networks. Preprint at http://arxiv.org/abs/1312.6211 (2015).

27. McCloskey, M. & Cohen, N. J. Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. in *Psychology of Learning and Motivation* vol. 24 109–165 (Elsevier, 1989).

28. Ji, J., Krishna, R., Fei-Fei, L. & Niebles, J. C. Action Genome: Actions As Compositions of Spatio-Temporal Scene Graphs. in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 10233–10244 (IEEE, 2020). doi:10.1109/CVPR42600.2020.01025.

29. Zacks, J. M., Tversky, B. & Iyer, G. Perceiving, remembering, and communicating structure in events. *J. Exp. Psychol. Gen.* **130**, 29–58 (2001).

30. Hard, B. M., Tversky, B. & Lang, D. S. Making sense of abstract events: Building event schemas. *Mem. Cognit.* **34**, 1221–1235 (2006).

31. Baldassano, C. *et al.* Discovering Event Structure in Continuous Narrative Perception and Memory. *Neuron* **95**, 709-721.e5 (2017).

32. Hasson, U., Yang, E., Vallines, I., Heeger, D. J. & Rubin, N. A Hierarchy of Temporal Receptive Windows in Human Cortex. *J. Neurosci.* **28**, 2539–2550 (2008).

33. Wang, L. & Dunson, D. B. Fast Bayesian Inference in Dirichlet Process Mixture Models. *J. Comput. Graph. Stat.* **20**, 196–216 (2011).

34. Corti, A., Giancola, S., Mainetti, G. & Sala, R. A metrological characterization of the Kinect V2 time-of-flight camera. *Robot. Auton. Syst.* **75**, 584–594 (2016).

35. Li, B., Yan, J., Wu, W., Zhu, Z. & Hu, X. High Performance Visual Tracking with Siamese Region Proposal Network. in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* 8971–8980 (IEEE, 2018). doi:10.1109/CVPR.2018.00935.

36. Vinh, N. X., Epps, J. & Bailey, J. Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance. 18.