

## **UC Merced**

### **Proceedings of the Annual Meeting of the Cognitive Science Society**

#### **Title**

A Computational Model for the Dynamical Learning of Event Taxonomies

#### **Permalink**

<https://escholarship.org/uc/item/15q1k95w>

#### **Journal**

Proceedings of the Annual Meeting of the Cognitive Science Society, 39(0)

#### **Author**

Gumbsch, Christian

#### **Publication Date**

2017

Peer reviewed

# A Computational Model for the Dynamical Learning of Event Taxonomies

Christian Gumbsch (christian@gumbsch.de)

Sebastian Otte (sebastian.otte@uni-tuebingen.de)

Martin V. Butz (martin.butz@uni-tuebingen.de)

Chair of Cognitive Modeling, Department of Computer Science and Department of Psychology, Faculty of Science, Eberhard Karls University of Tübingen, Tübingen, Germany

## Abstract

We present a computational model that can learn event taxonomies online from the continuous sensorimotor information flow perceived by an agent while interacting with its environment. Our model implements two fundamental learning biases. First, it learns probabilistic event models as temporal sensorimotor forward models and event transition models, which predict event model transitions given particular perceptual circumstances. Second, learning is based on the principle of minimizing free energy, which is further biased towards the detection of free energy transients. As a result, the algorithm forms conceptual structures that encode events and event boundaries. We show that event taxonomies can emerge when the algorithm is run on multiple levels of precision. Moreover, we show that generally any type of forward model can be used, as long as it learns sufficiently fast. Finally, we show that the developed structures can be used to hierarchically plan goal-directed behavior by means of active inference.

**Keywords:** event models; object interaction; predictive encoding; event segmentation; free energy; active inference; event taxonomy; concept learning

## Introduction

*Event segmentation theory* (EST) (Zacks & Tversky, 2001; Zacks, Speer, Swallow, Braver, & Reynolds, 2007) postulates that humans automatically structure the stream of sensory and sensorimotor information into meaningful events. Events are defined as “a segment of time at a given location that is conceived by an observer to have a beginning and an end” (Zacks & Tversky, 2001, p. 3). This definition is formulated rather broadly, containing long, abstract events (e.g. ‘going on a vacation’), more concrete events (e.g. ‘taking a taxi to the airport’), and very short events (e.g. ‘grasping something’). Zacks and Tversky (2001) suggest that events can be organized in *event taxonomies*, where abstract events consist of multiple, more concrete events.

In Gumbsch, Kneissler, and Butz (2016) we proposed a system that learns events and event boundaries from the sensorimotor stream an agent experiences while interacting with its environment. Following EST, we defined predictive events (Zacks et al., 2007) as sets of forward models, i.e. internal models that predict the sensorimotor consequence of an agent’s actions. Moreover, event boundaries were defined as the determining features of a situation that typically lead to a transition between events, that is, between active forward models (Butz, 2016). Events and event boundaries are encoded in a Bayesian fashion, whereby learning and updating can be closely related to the free energy minimization principle (Friston, 2009). In addition, though, the detection of event

boundaries, and the consequent possible transition to another event model, was biased towards the detection of transient free energy signals. This is closely related to the transient error principle of Zacks et al. (2007), according to which event boundaries are characterized by transient increases in the perceptual prediction error. The result is an algorithm that learns event boundaries from “surprising” perceptions. We have shown that the algorithm learns a compositional conceptual structure of the experienced environment. Algorithmically, the system monitors the current prediction error of the active forward models and registers surprise when the encountered prediction error surpasses an adaptive confidence threshold. As a result, the set of active models is changed, the relevant features that characterize the encountered event boundary are identified, the event model transitions are memorized, and new event models may be learned. We have shown that the developing model can be used to both predict sensorimotor changes and plan in a goal-directed manner.

In this paper, we generalize the established mechanism, making it more noise robust and more generally applicable beyond linear models and recursive least squares updating. Moreover, we show that the surprise signal and the noise level determine the granularity of the event segmentation. As a result, an event hierarchy can emerge naturally during this process. Finally, we explicitly derive the goal-directed planning process from formalizations of active inference (Friston, 2009; Friston et al., 2015), showing that the developing event and event boundary models are very well-suited for the invocation of inverse, hierarchical, goal-directed behavior.

## Architecture

The *system architecture*  $\mathcal{S}$  consists of three continuously interacting main components  $\mathcal{S} = \{\mathcal{M}, \mathcal{B}, \mathcal{P}\}$ . The set of *event models*  $\mathcal{M}$  comprises all so-far learned temporal forward models. At every point in time, a subset of forward models is active, simulating and predicting the spatiotemporal changes in the environment. These spatiotemporal simulations are determined in the *predicted perceptual space*  $\mathcal{P}$ . Event transitions are detected based on statistical evaluations of the prediction error, detecting free energy transients. They are stored in the *event boundary models*  $\mathcal{B}$ , where each model attempts to identify the critical sensory features that allow the probabilistic prediction of an event transition.

Figure 1 and Figure 2 show the three components in interaction with the outside environment and the internal moti-

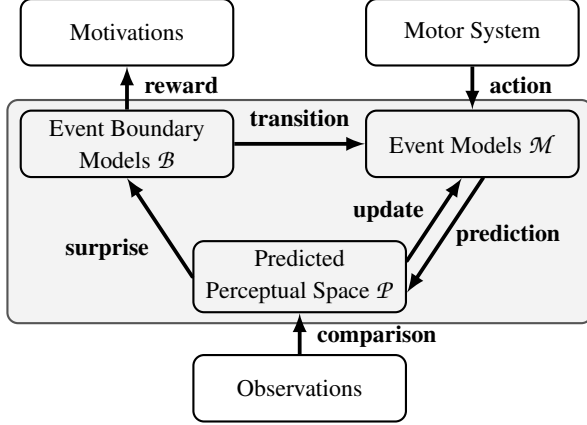


Figure 1: Illustration of the system during forward modeling. An active event model predicts the perception based on the next action. The predictions are compared with the real observations and the event model is updated based on the deviation. If this deviation exceeds a threshold, a surprise is detected, the event boundary models are updated, and a transition in the event models may be triggered. Rewards experienced during transitions are registered in the motivational system and are associated with the respective event boundary.

vational system during prediction and during planning mode, respectively. While interacting with the environment, the architecture generates temporal forward predictions and learns from the registered errors (Figure 1). Inversely, the architecture can generate active inference-driven probabilistic planning by activating event boundaries as goals and inversely propagating these goals into  $\mathcal{P}$  and  $\mathcal{M}$ , leading to the generation of goal-directed actions based on the believed knowledge about the environment (Figure 2). Due to the developing event-based predictive environmental model, hierarchical, conceptual planning becomes possible.

### Model components and functionality

Event models  $m \in \mathcal{M}$  are encoded as sets of  $N$  forward models, given an  $N$ -dimensional observational space. At a certain point in time  $t$  one event model  $m(t)$  is active, with  $m(t) = (m_1^y, m_2^y, \dots, m_N^y)$ , where  $m_i^y$  references the currently active forward model component with respect to a particular dimension  $i$ . Each active forward model  $m_i^y$  predicts the changes in one dimension of the sensory observation  $\Delta o_i^y(t)$ , given a particular action  $a(t)$ . After executing  $a(t)$ , the real observation  $o(t+1)$  is compared with the predicted observation  $o'(t+1) = o(t) + \Delta o^y(t)$ . As a result, the active forward models in  $m(t)$  are updated based on the error signal to improve the respective forward models. To maintain minimal additional statistics, each forward model  $m_i^y$  stores the moving average (currently fixed over the last 100 steps) of its prediction error  $\bar{e}(m_i^y)$ , and the moving average of the variance of that prediction error  $\bar{\sigma}^2(m_i^y)$ , thus estimating the first three moments of the model’s predictions.

Event boundaries  $b \in \mathcal{B}$  are represented as top-down generative predictive models, which probabilistically predict the

perceptual features of the environment that are relevant for enabling or causing the transition from one particular event  $\nu$  to another one  $\mu$ , i.e.  $P(o|m_i^y \rightarrow m_i^\mu)$ . Thus, the models currently assume that all relevant information for the occurrence of an event boundary is observable when an event model transition occurs. In our current implementation, we model these event transitions by means of multivariate Gaussians.

As a result, at any point in time  $t$  the generative model of the system is in a particular state  $s(t) = \{m(t), P(\mathcal{B}), \mathbf{o}(t), \mathbf{o}'(t)\}$ , where  $m(t)$  denotes the current vector of active forward models (winners take all),  $P(\mathcal{B})$  denotes the probability mass over possible event boundaries, and  $\mathbf{o}(t)$  and  $\mathbf{o}'(t)$  denote observation densities.

### Simulation and inference-based learning

While interacting with the environment, the system develops its predictive models  $\mathcal{M}$  and  $\mathcal{B}$ , which are updated and improved based on the comparisons between predicted and actually encountered observations in  $\mathcal{P}$  (illustrated in Figure 1). Event models are updated by standard gradient descent techniques, seeing that we face a self-supervised learning problem. We evaluate recursive least squares as well as delta-rule based gradient descent-based event models.

Event boundaries are detected by a sudden, significant rise of the prediction error above the tolerated uncertainty. The prediction error  $e_i(t)$  of sensory dimension  $i$  at time step  $t$  is considered “surprising” when

$$e_i(t) > \bar{e}(m_i^y) + \theta \bar{\sigma}(m_i^y), \quad (1)$$

with  $m_i^y$  denoting the active forward model of dimension  $i$  and  $\theta$  the confidence-dependent *surprise threshold*. This threshold essentially determines when an error is considered significant depending on the currently estimated standard deviation, i.e. the inverse confidence or precision, of the respective forward model. It modulates the granularity of the event segmentation performed by our system, which in turn strongly influences how accurately each ongoing event is predicted.

If a surprise-signal is detected, the system is allowed to switch the active forward models  $m(t)$ . The system enters a *searching period* during which the next active models are determined. To do so, all existing forward models  $m_i$  of dimension  $i$  are taken into consideration. For a fixed number of time steps (10 time steps in our simulations) each model predicts the change in observation and is updated. After this search period, if the prediction error is still considered “surprising” by all existing forward models (determined by Eq. 1), a new forward model is generated and added to the possible forward models  $m_i$ . On the other hand, if the prediction error is not surprising for at least one existing model, the forward model  $m_i^\mu$  with the smallest mean error is chosen as the new forward model for dimension  $i$ .

To summarize, our generative model space essentially consists of a set of temporal forward models  $\mathcal{M}$  and a set of probabilistic transition models  $\mathcal{B}$ , that is, event boundary models. To minimize free energy, the validity of each temporal forward model is optimized by gradient descent – maximizing

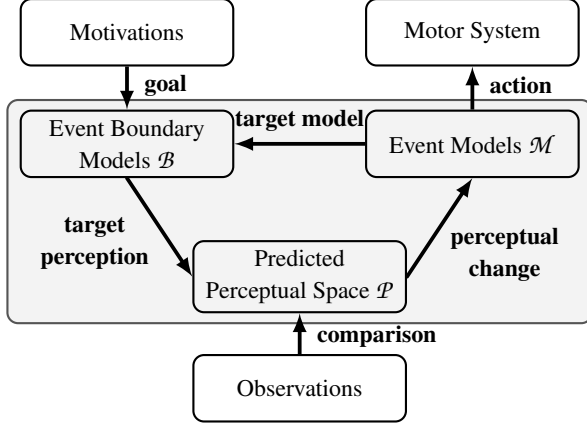


Figure 2: Illustration of the system during planning. Based on the system’s goal, an event boundary is chosen. The required perception to reach this boundary is compared with the currently predicted perception and the necessary perceptual change is computed. If this change can be achieved by the active event model, a suitable motor command is determined. Otherwise, a new model to fulfill this change is chosen and the event boundary to reach this new model is determined, proceeding recursively.

its predictive accuracy in its applicable subspace. Combined with its current minimal statistics (first three moments), the temporal forward models can be viewed as Gaussian densities, which are optimized approximately optimally based on the incoming sensory feedback (Kneissler, Drugowitsch, Friston, & Butz, 2015). Additionally, the approach has – as a structural prior – the assumption that temporal forward models will be typically applicable during an extended period of time and that transitions between forward models can be characterized by event boundaries (Butz, 2016). As a result, event boundaries are optimized such that free energy can be minimized equally well in all subspaces of the environment.

### Goal-directed behavior

To invoke goal-directed behavior, we add a simple “motivational” system to the architecture, which associates model states with changes in its internal motivational state (Butz, Shirinov, & Reif, 2010). In the current implementation, we simplify this part by associating particular motivations with particular event boundaries, such as the disappearance of food due to its consumption. Thus, when goal-directed planning is invoked, a desired event boundary is activated and inversely propagated through the system’s generative model.

Formally, we can infer the optimal behavioral policy  $\pi$  by assigning a value to all imaginable policies given the current beliefs of the model and choosing the best one:

$$Q_\tau(\pi, s_t) = -E_\pi[H(P(o_\tau|s_\tau))] - D_{KL}[P(o_\tau|s_\tau)||P(o_\tau|s_G)], \quad (2)$$

which is based on (Friston et al., 2015). This formulation essentially states that a policy  $\pi$ , starting at believed situational

system state  $s_t$ , will result in the minimization of free energy by minimizing (i) predicted uncertainty over the expected unfolding successive states up to a certain temporal horizon  $T$  into the future  $\tau = \{t, t + 1, \dots, T\}$  measured by the entropy  $H(P(o_\tau|s_\tau))$  over expected states  $s_\tau$  when pursuing policy  $\pi$  and (ii) the divergence, formulated as the Kullback-Leibler divergence  $D_{KL}$ , between expected future observations and desired future observations given a goal state distribution  $s_G$ , i.e.  $P(o_\tau|s_G)$ .

When we fully focus behavior on maximizing reward outcomes, ignoring uncertainties in our expected progression through the environment, it is possible to cancel out the term  $E_\pi[H(P(o_\tau|s_\tau))]$  in  $D_{KL}$ , yielding:

$$Q'_\tau(\pi, s_t) = P(o_\tau|s_t) \ln P(o_\tau|s_G). \quad (3)$$

As a result, the goal is to maximize the overlap between these two probability densities.

In our model’s case,  $P(o_\tau|s_G)$  will correspond to a particular model transition  $m_g^y \rightarrow m_g^h$  and all other model transitions will be set to zero for all future points in time  $\tau$ . As a consequence, the model essentially “wants” to maximize

$$P(m_i^y \rightarrow m_i^h|s_\tau), \quad (4)$$

i.e. the probability of the model transition  $m_i^y \rightarrow m_i^h$ . This corresponds to first maximizing  $P(m_i^y|s_\tau)$ , that is, being in model  $m_i^y$  and then maximizing the likelihood of the transition, which, in turn, corresponds to maximizing the probability of the observation that characterizes the transition, that is,  $P(o_\tau|m_i^y \rightarrow m_i^h)$ . Thus, given the current state of the model  $s(t)$ , the inference process may directly yield motor actions that attempt to maximize  $P(o_\tau|m_i^y \rightarrow m_i^h)$  if this seems possible given the current event model state. However, when the event model  $m_i^y$  is currently not active, then a recursive process must start that selects another model transition  $P(m_i^y \rightarrow m_i^y)$  in order to reach the goal transition by first invoking an intermediate transition (illustrated in Figure 2).

As a result, hierarchical, conceptual goal-directed probabilistic inference-based planning is implemented, which invokes event boundaries as subgoals, given the final goal cannot be reached by the currently available control options. For example, the model can infer that if it wants to drink out of a mug but the mug is not in the hands, the mug first needs to be approached and grasped before it can be transported to the mouth. Note that the recursive planning procedure, in principle, can find any sequence of events and event boundaries, which are believed to lead to the goal. However, time and space as well as precision limitations may generally apply when propagating the active inference signals through the system architecture  $\mathcal{S}$ .

### Evaluation

To investigate the event segmentation and hierarchical planning capabilities of our system, we have chosen a testing scenario, in which a simulated agent, operating in continuous space, can interact with different objects in multiple ways

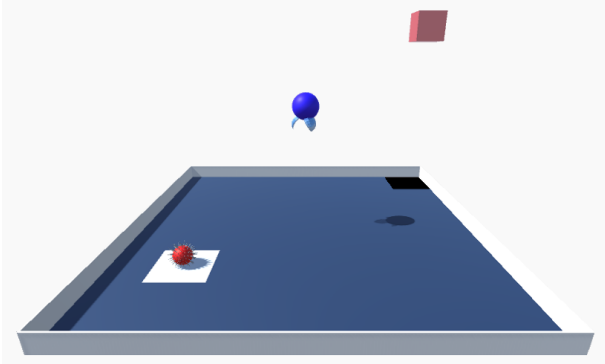


Figure 3: The evaluation scenario: Objects (here a sticky object) are generated in the white area and vanish when they enter the black, rectangular mouth area. The blue hand is able to grasp or attach objects. They are detached from the hand once they are inside the red “release-area” cube.

(Figure 3). The agent consists of a hand with three shovel-like fingers and a stationary “mouth”. The hand is able to move freely through a limited 3-dimensional workspace with a “release-area”. Three types of differently colored objects appear in our simulation (two types of “sticky objects” and one type of “marble”). Sticky objects are big and spiky. They automatically attach to the hand upon contact. Once attached they are dragged alongside the hand until they enter the release-area, wherein they detach and drop into the mouth. We use two types of sticky objects in our simulation: light objects do not alter the hand movement when attached to it; heavy ones slow down the hand movement by a factor of  $\frac{1}{4}$ . Marbles are small spheres that need to be grasped to be transported by the hand. To grasp a marble the hand has to be positioned directly over a marble for a grasping reflex to activate. The fingers open again when the marble is inside the release-area. Carrying a marble is usually far more difficult to predict than dragging a sticky object, since marbles sit loosely between the fingers and shake while being transported. If an object is dropped into the mouth, it is consumed and a new object is generated. In our scenario the system is rewarded once an object is consumed. Thus, to receive rewards, the agent has to attach or grasp the present object, transport it to the release-area, and drop it into the mouth.

In every time step  $t$ , a motoric action  $a(t)$  is performed and a sensory observation  $o(t)$  is perceived. In particular  $o(t)$  consists of the position of the hand  $x_h, y_h, z_h \in [-100, 100]$ , the position of the object  $x_o, y_o, z_o \in [-100, 100]$ , and the position of the object in a hand-centered frame of reference  $x_{o,h}, y_{o,h}, z_{o,h} \in [-200, 200]$ . Additionally  $o(t)$  contains the color of the object and boolean information whether the object has spikes or not. The motor command  $a(t)$  determines the change in hand position  $\Delta x_h, \Delta y_h, \Delta z_h \in [-1, 1]$ . Furthermore,  $a(t)$  contains the velocity of the object to enable computation of the object’s next position even while it is falling.

To evaluate the system, we investigate how the learning capabilities depend on the underlying learning rule. Therefore,

we learn the forward models, currently linear prediction models, both by means of delta rule based gradient descent (learning rate  $\eta = 0.1$ , momentum term  $\alpha = 0.9$ , linear activation function) as well as by means of recursive least squares (RLS) (forgetting factor  $\lambda = 0.99$ ). RLS essentially implements an adaptive filter that minimizes the sum of squared residuals recursively in an optimal online fashion. Furthermore, we show that both the threshold  $\theta$  and sensory noise influence the granularity of the determined event segmentations – allowing the formation of event taxonomies. We performed every simulation 10 times with a different random initialization.

## Results

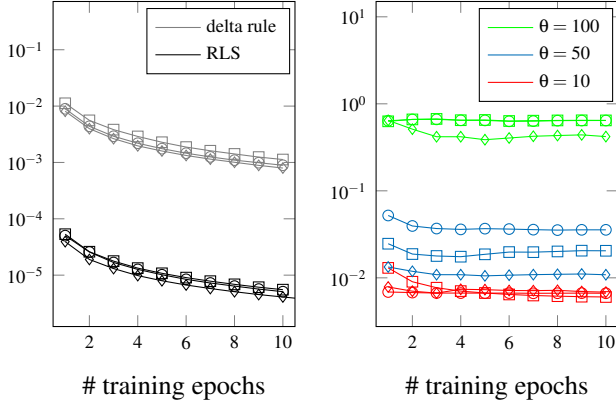
In a first test we analyzed how the underlying learning rule of the forward models influences the event segmentation and learning accuracy of the system, comparing stochastic gradient descent with RLS. During learning, the motoric action  $a(t)$  was determined by an external algorithm, which performed a mixture of random movements and behaviors leading to a new event. Every simulation consisted of 10 training and 10 test epochs, during which no forward model updates took place. In every epoch all three objects were generated once and manipulated by the agent until they were consumed.

The system was able to identify all existing events in the simulation (hand moving normally/slowly, object lying, light/heavy sticky object dragged, marble carried, object dropped) for both types of forward models used. The average prediction error of all active forward models for three different events is shown in Figure 4a. For both learning rules the system improves the prediction accuracy over the training epochs, but RLS-based learning quickly reaches a much better prediction. While this result shows that both forward model learning approaches can be applied, further tests were conducted using RLS-based learning to speed-up the process.

In a second test we analyzed how the surprise threshold  $\theta$  influences the granularity of the event segmentation. Thus we performed this test with three different surprise thresholds ( $\theta \in \{10, 50, 100\}$ ). Since we hypothesized that sensory noise also alters the granularity of segmentation, we additionally varied the amount of Gaussian distributed noise ( $\sigma \in \{0.001, 0.01, 0.05, 0.1\}$ ) that was added to each observation  $o(t)$ . Additionally, a small amount of Gaussian distributed noise ( $\sigma = 0.01$ ) was added to each action  $a(t)$ .

The average prediction error for different events at the lowest level of noise tested ( $\sigma = 0.001$ ) is shown in Figure 4b. The prediction error of the system greatly varies for the different surprise thresholds. After only a few training epochs, the prediction accuracy of the active forward models for the smallest tested surprise threshold is close to the level of sensory noise. In contrast, the mean prediction error of the system for the largest threshold is only slightly below 1. For  $\theta = 50$  the prediction error varies among the different events.

To further analyze this difference in prediction accuracy we examined the number of forward models generated for the different surprise thresholds and determined which forward



(a) Mean prediction error for different forward models (b) Mean prediction error for different surprise thresholds

Figure 4: Mean prediction error of all active forward models for different events for (a) different forward models without sensory noise; (b) different surprise thresholds  $\theta$  using RLS-based forward models and Gaussian distributed sensory noise ( $\sigma = 0.001$ ). Three events are shown: object lying ( $\diamond$ ), light object dragging ( $\square$ ), and heavy object dragging ( $\circ$ ).

models were most active for which event. In the following, we first refer to the case with the lowest level of sensory noise ( $\sigma = 0.001$ ). Later, we analyze how a variation in sensory noise influences the number of event models. All identified event models are shown taxonomically structured in Figure 5.

For  $\theta = 100$  two forward models developed for predicting object positions. One of these models was active when the object was lying, the other one was active for the rest of the time. Thus the system differentiated between a moving and a still object in terms of object position. For  $\theta = 50$  three models were predicting  $x_o$  and  $z_o$ -position. Here the system further differentiated between slowly carrying a heavy object or transporting a light object at normal speed. For the  $y_o$ -position one additional model was generated, which was active when an object was falling. For  $\theta = 10$  every type of transportation was represented by a different forward model. Hence, for the smallest surprise threshold, the system even differentiated between slow and normal hand movements, while for larger values of  $\theta$  this was not the case.

Sensory noise additionally influences the granularity of the performed event segmentation. While for low noise levels ( $\sigma = 0.01$  and  $\sigma = 0.001$ ) the identified events did not differ, an increase in noise results in a coarser event segmentation. For  $\sigma = 0.05$  and  $\theta = 10$  the system used one forward model for every sensory dimension describing hand position and three forward models to predict changes in object position: one for a lying object, one for a transported object, and one for a falling object. The segmentation further coarsened for  $\sigma = 0.1$ , where the system only distinguished between lying and moving objects. If both noise level and surprise threshold were too large ( $\sigma \geq 0.05$ ,  $\theta \geq 50$ ) the system did not detect event transitions at all, such that only one forward model was generated per sensory dimension.

$\theta \geq 50$	object exists				
$\theta \geq 50$	object is lying	object moves			object is falling
$\theta = 100$		object moves alongside hand			
$\theta = 10$		object moves with hand normally		object is dragged slowly	
$\theta = 50$		object is dragged	object is carried		
$\theta = 10$					
Real Events					

Sensory noise levels:  $\sigma = 0.1$ ,  $\sigma = 0.05$ ,  $\sigma \leq 0.01$

Figure 5: The event models regarding object position identified by the system for different surprise thresholds and sensory noise levels. The column on the left states the surprise threshold  $\theta$ . The associated sensory noise  $\in [-\sigma, \sigma]$  is illustrated by color, as explained by the color legend below the table. Each row shows the identified event model for this surprise threshold and noise level. The bottom row illustrates the real, underlying interactions found in our scenario.

In a third test we evaluated the planning capabilities of our system, using its representation of events and event boundaries to generate goal-directed behavior. Furthermore we analyzed how the granularity of the underlying event representations influences planning. Thus, we varied granularity by using three values of the surprise threshold ( $\theta \in \{10, 50, 100\}$ ) under the low noise condition  $\sigma = .001$ . Every simulation consisted of 30 training and 30 testing epochs. During each training epoch one sticky object and one marble were generated. The goal of the system was to consume the objects. The system was given a fixed time interval (500 simulation steps) to interact with each object. If the system failed to remove an object in the given time period, an external algorithm performed the required movements. During testing, we introduced a new object that closely resembled the marble in its visual characteristics (small, similar color, no spikes). Once grasped, however, the new object behaved like a sticky object and attached to the hand. The challenge of successfully interacting with the new object is thus to grasp the object like a marble (by means of the appropriate event boundary models) but to then transport the object like a sticky object (with the help of the appropriate event models).

Already after the first training epoch, the system consumed 50% of the novel objects correctly when a fine-grained event segmentation was used ( $\theta = 10$ ). From 25 training epochs onward, the system managed to successfully interact with every new object. For  $\theta = 50$ , the system on average only interacted with 70% of the new objects correctly after all 30 training epochs. For  $\theta = 100$ , the system did not manage

to interact with any test object successfully. The difference in performance can be explained by examining the identified events (see Figure 5). For  $\theta = 100$  the system does not distinguish between transporting and dropping an object, such that the ‘detachment’ event boundary was not learned and thus could not be used as a subgoal. Similarly, for  $\theta = 50$  the system does not distinguish between transporting a marble or a sticky object, such that the ‘grasp’ and ‘attach’ event boundaries were mixed – often leading to unsuccessful grasps.

## Conclusion

Based on event segmentation theory (Zacks et al., 2007) and the principle of free energy minimization (Friston, 2009), we have developed a computational model of hierarchical, behavior-grounded event segmentation. Our system uses a strictly statistical measure of “surprise” to segment the sensorimotor stream, which an agent experiences while interacting with its environment, into events encoded by temporal forward models. In a continuous, noisy simulation, our system was able to identify event models characterizing particular object interactions, e.g. ‘carrying an object’ or ‘dropping an object’. Furthermore, the environment was structured into conceptual event and event boundary encodings, which discriminate the critical features that are crucial for the occurrence of an event, e.g., ‘hand contact’ is required to manipulate an object. Due to the event-based architecture, the system accomplished to plan hierarchical behavior consisting of multiple subroutines to reach desired goal states.

We showed that a change in the confidence-threshold, which determines when a transient free energy signal is considered “surprising”, affects the granularity of the event segmentation. Depending on this threshold, the system accomplished to identify events for concrete object interactions, e.g. ‘carrying a marble’, or abstract representations of interactions, which subsume several more concrete events, e.g. ‘moving an object’. Similarly, an increase in sensory noise entailed a coarser segmentation. Thus, based on these simple statistical principles, a hierarchy can emerge, similar to the event taxonomy proposed by Zacks and Tversky (2001), in which abstract events comprise several more concrete events.

The developing hierarchical organization of event models and consequent event-oriented behavior is closely related to hierarchical reinforcement learning (Botvinick & Weinstein, 2014; Sutton, Precup, & Singh, 1999). Similar to our event models, options predict changes in the system’s state when performing a sequence of behavior without considering low-level steps. While the composition of options has been shown to enable the learning of complex behavior when using a pre-defined set of goals (Kulkarni, Narasimhan, Saeedi, & Tenenbaum, 2016), our system determines subgoals by itself by the principle of surprise-detection.

In sum the proposed model offers a general algorithm for online, hierarchical event segmentation learning given continuous sensorimotor experiences. Two main learning biases lead to successful segmentations: first, the partition of the de-

veloping generative model into probabilistic models of events and transitions between events; second, the focused learning of event transitions based on transient free energy signals. Besides the emergence of event taxonomies, we also showed that the developing conceptual structures can be learned to invoke hierarchical, goal-directed planning and behavioral control.

Our current research aims at integrating boundedly complex, non-linear forwards models and recurrent context information. Such enhancements will be necessary to handle non-uniform motion patterns and partially only indirectly observable causes of event transitions robustly. As a result, we hope to be able to show the more general and more scalable applicability of the principles we have introduced herein.

## References

- Botvinick, M., & Weinstein, A. (2014). Model-based hierarchical reinforcement learning and human action control. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 369(1655).
- Butz, M. V. (2016). Towards a unified sub-symbolic computational theory of cognition. *Frontiers in Psychology*, 7(925).
- Butz, M. V., Shirinov, E., & Reif, K. L. (2010). Self-organizing sensorimotor maps plus internal motivations yield animal-like behavior. *Adaptive Behavior*, 18(3-4), 315–337.
- Friston, K. (2009). The free-energy principle: a rough guide to the brain? *Trends in Cognitive Sciences*, 13, 293 - 301.
- Friston, K., Rigoli, F., Ognibene, D., Mathys, C., FitzGerald, T., & Pezzulo, G. (2015). Active inference and epistemic value. *Cognitive Neuroscience*, 6, 187-214.
- Gumbsch, C., Kneissler, J., & Butz, M. V. (2016). Learning behavior-grounded event segmentations. In *Proceedings of the 38th annual conference of the cognitive science society* (pp. 1787–1792).
- Kneissler, J., Drugowitsch, J., Friston, K., & Butz, M. V. (2015). Simultaneous learning and filtering without delusions: a bayes-optimal combination of predictive inference and adaptive filtering. *Frontiers in computational neuroscience*, 9.
- Kulkarni, T. D., Narasimhan, K., Saeedi, A., & Tenenbaum, J. (2016). Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. In *Advances in neural information processing systems* (pp. 3675–3683).
- Sutton, R. S., Precup, D., & Singh, S. (1999). Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112, 181-211.
- 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, 273–293.
- Zacks, J. M., & Tversky, B. (2001). Event structure in perception and conception. *Psychological Bulletin*, 127, 3–21.