

# Latent Event-Predictive Encodings through Counterfactual Regularization

Dania Humaidan (dania.humaidan@uni-tuebingen.de)<sup>1,2</sup>  
Sebastian Otte (sebastian.otte@uni-tuebingen.de)<sup>1,2</sup>  
Christian Gumbsch (christian.gumbsch@uni-tuebingen.de)<sup>1,2,3</sup>  
Charley Wu (charley.wu@uni-tuebingen.de)<sup>2,4</sup>  
Martin V. Butz (martin.butz@uni-tuebingen.de)<sup>1,2</sup>

<sup>1</sup> Neuro-Cognitive Modeling Group, Dep. of Psychology and Dep. of Computer Science, Sand 14, 72076 Tübingen, Germany

<sup>2</sup> Cluster of Excellence – Machine Learning for Science, Maria-von-Linden-Str. 6, 72076, Tübingen, Germany

<sup>3</sup>: Autonomous Learning Group, MPI for Intelligent Systems, 72076, Tübingen, Germany

<sup>4</sup> Human and Machine Cognition Lab, Maria-von-Linden-Str. 6, 72076, Tübingen, Germany

## Abstract

A critical challenge for any intelligent system is to infer structure from continuous data streams. Theories of event-predictive cognition suggest that the brain segments sensorimotor information into compact event encodings, which are used to anticipate and interpret environmental dynamics. Here, we introduce a SURprise-GAted Recurrent neural network (SUGAR) using a novel form of counterfactual regularization. We test the model on a hierarchical sequence prediction task, where sequences are generated by alternating hidden graph structures. Our model learns to both compress the temporal dynamics of the task into latent event-predictive encodings and anticipate event transitions at the right moments, given noisy hidden signals about them. The addition of the counterfactual regularization term ensures fluid transitions from one latent code to the next, whereby the resulting latent codes exhibit compositional properties. The implemented mechanisms offer a host of useful applications in other domains, including hierarchical reasoning, planning, and decision making.

**Keywords:** event-predictive cognition; event segmentation; artificial neural networks; predictive processing; compositionality; surprise

## Introduction

Several disciplines have proposed that our brains construct generative, event-predictive models to interpret and shape our perception of the world (Baldwin & Kosie, 2021; Butz & Kutter, 2017; Butz, Achimova, Bilkey, & Knott, 2021; Franklin, Norman, Ranganath, Zacks, & Gershman, 2020; Kuperberg, 2021; Zacks, Speer, Swallow, Braver, & Reynolds, 2007). Latent, event-predictive encodings provide top-down contextual guidance, which improves our prediction of local temporal patterns. As we watch scenes from a movie or experience daily life, our learned event models allow us to anticipate switches between events. For instance, we can anticipate when the credits will roll or when the lecture is about to end, enabling us to suitably adjust our behavior in preparation of the transition.

Here we focus on the problem of learning to generate smooth event transitions and to thus enable the fluent processing of continuous streams of sensorimotor information across events. To achieve this, the learning of compact, event-predictive encoding seems to be of particular importance (Baldwin & Kosie, 2021; Zhu et al., 2020). Accordingly, it has been suggested that our brains learn to preempt surprise, that is, to predict unpredictability when a current

event is about to end (Baldwin & Kosie, 2021; Franklin et al., 2020; Kuperberg, 2021). Temporary increases in prediction error have been suggested to offer suitable event segmentation clues (Zacks et al., 2007; Zacks, Kurby, Eisenberg, & Haroutunian, 2011). Others have emphasized the importance of temporary stable latent predictive encodings between transitions (Shin & DuBrow, 2021; Schapiro, Rogers, Cordova, Turk-Browne, & Botvinick, 2013), which is generally compatible with surprise-biased processing. Thus, while surprise marks event transitions, predictability marks ongoing events.

Here we present a SURprise-GAted Recurrent neural network (SUGAR), as a biologically-inspired normative model that offers computational and algorithmic explanations for how compact, compositional event-predictive encodings may develop. Similar to how uncertainty modulates the arbitration between systems in other domains (Lee, Shimojo, & O’Doherty, 2014; Daw, Niv, & Dayan, 2005), we show that surprising sensory information can be used to modulate top-down control, thus fostering better encoding of latent events and transitions between them. However, once surprise is detected, only reactive or retrospective adaptations are possible. Thus, it is critical to preempt surprise when intending to process event transitions fluidly.

In the real world, orienting reflexes help us to investigate the causes of a surprising event, such as when we trip over some overlooked item. As a result, we learn new causal relationships and refine the involved event-predictive encodings (Zacks et al., 2007). Moreover, we learn to preempt the surprise by preparing for an upcoming transition and thus switching fluently to the new events—as when we switch from walking to climbing up some stairs or when we open the fridge to get milk for our tea (Baldwin & Kosie, 2021; Kuperberg, 2021). Accordingly, we augment SUGAR with an event boundary anticipation module, which is designed to learn to control event switches. We show that as long as sufficient information is provided to anticipate event switches, the module learns to switch in time. However, fluctuations and unnecessary openings still occur. We show that the addition of a novel counterfactual regularization (CFR) loss can improve learning reliability as well as the precision and compactness of the developing latent event-characterizing encodings. CFR evaluates whether an event switch was indeed

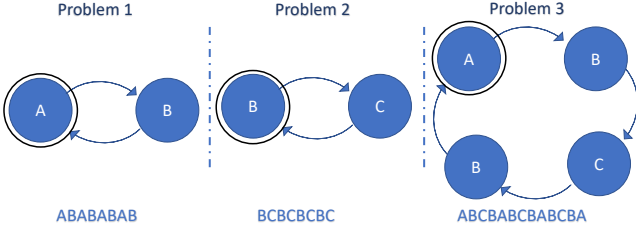


Figure 1: Sequence of letters are generated from one of the three problems. Switching between the problems occurs at uniformly sampled time points within the interval of  $[10,30]$ . Each problem starts from the highlighted letter.

helpful for avoiding surprise, effectively balancing costs of event switches with rewards from preventing surprise.

We show that SUGAR with CFR is able to both anticipate event transitions and transition between latent event codes fluently, fully avoiding surprising outcomes. Moreover, we show that the emergent latent codes exhibit compositional properties of the hierarchical task dynamics. Our model architecture and learning techniques may be useful for both modeling cognition and improving the performance of artificial systems in a variety of reasoning, planning, and decision-making tasks.

### Hierarchical Sequence Prediction Task

In this paper we focus on the prediction of simple, hidden grammatical regularities within a sequence prediction task. The general structure of the task is shown in Figure 1. Three events, or problems, are encoded as a systematic progression through a sequence of symbols:  $[A,B]^*$ ,  $[B,C]^*$ , and  $[A,B,C,B]^*$ . Transitions between the problems occur randomly, where the number of steps before the next transition  $\tau$  is sampled uniformly randomly from  $\tau \in \{10, \dots, 30\}$ . The sequences are fed sequentially into the SUGAR model as one-hot encoded sensory information  $x^t$ , which is provided to the event processing layer (cf. Figure 2). In the experiments we combine either problems 1 and 2, denoted by Problem 1+2, or all three, which we denote by Problem 1-3.

In addition to the symbol sequence, SUGAR’s event anticipation layer (cf. Figure 2) receives additional information about which event is going to come next, denoted as contextual information (CI) input. Again by means of a one-hot encoding, CI indicates from a randomly chosen point in time  $\tau^* < \tau$  onward which problem will be sampled once the next event switch occurred. Moreover, masked information is provided about the number of iterations  $\tau$  before an event switch will occur, called the event boundary (EB) input. This EB input mimics information in the real world that helps us to anticipate that a new event is about to start. For example, we get ready to grasp a glass while reaching for it, anticipating the grasp initiation by tracking the distance between our hand and the glass. In the experiments presented below, we first provide simple one-hot encoded switching information with additional randomly fluctuating inputs, challenging

the learner to identify the correct switching signal. Later, we provide multiple relevant inputs in the form of an input vector whose activities tend to increase while the switch comes close. The actual switch occurs when all input vector values have reached full activity.

### The SUGAR model

The SUGAR model includes four main event-predictive layers (see Fig. 2). An *event processing* layer is designed to process the currently unfolding event dynamics by predicting the next letter in the sequence. An *event anticipation* layer processes event-predictive embeddings to provide contextual guidance about which of the three problems (Fig. 1) is currently active. This guidance is provided in the form of latent event encodings, which are sent to the event processing layer. Third, an *event switching* layer acts as a gate controlling whether top-down contextual guidance is fed into the event processing layer. Finally, an *event boundary* layer is designed to learn to predict when the next event transition will take place.

The event processing, anticipation, and boundary layers are implemented by long short-term RNNs (LSTMs; Hochreiter & Schmidhuber, 1997), although other architectures may also be suitable these types of problem. In our current problem, the event processing layer receives as input the sequence value  $\mathbf{x}_{t-1}$  from the previous time step  $t-1$ , along with current top-down information  $\mathbf{x}_o^t$  from the event switching layer.

We investigate three progressively enhanced SUGAR variants in this paper, which control the gating mechanism in the event switching layer in distinct manners (Fig. 2). We first describe the exact processing dynamics of the switching layer.

#### Event Switching Layer

The event switching layer is implemented using a state switching gate, which are also used in gated recurrent unit RNNs (GRU; Cho et al., 2014). The switching gate thus controls when the top-down guidance signal from the event anticipation layer will be passed down to the event processing layer. Input to the event switching layer comes from the event anticipation layer ( $\mathbf{x}_a$ ) and the event boundary layer ( $\mathbf{x}_b$ ). We denote its input layer with  $\eta$  and its update gate with  $\zeta$ .

**Forward pass** The scalar activity  $x_\zeta$  of the update gate controls the switch between maintaining the previous latent event code or updating it with novel contextual information: It is defined by:

$$\begin{aligned} net_\zeta^t &= x_s^t \\ x_\zeta^t &= \phi_\zeta(net_\zeta^t), \end{aligned} \quad (1)$$

where  $x_s^t$  denotes the surprise signal input at the current point in time  $t$ , and  $\phi_\zeta$  denotes the cell’s sigmoid activation function.

The activity  $\mathbf{x}_\eta$  of the input layer is determined by combining and integrating the output of the event anticipation layer  $\mathbf{x}_a$  as well as the previous hidden state of the event switching

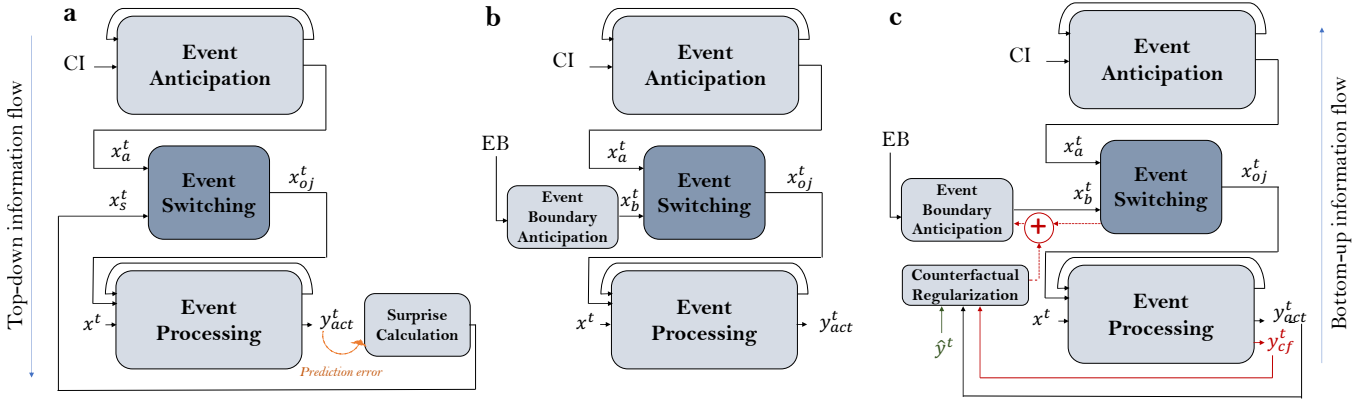


Figure 2: **a)** SUGAR<sub>a</sub> calculates a surprise signal online and uses this signal to control the update gate in the event switching module. **b)** SUGAR<sub>b</sub> replaces the surprise signal with a dedicated event boundary anticipation module, which receives fuzzy event boundary information. **c)** SUGAR<sub>c</sub> adds CFR, to foster more precise gate opening and thus the emergence of more compact latent event encodings. CI: contextual information input, EB: event boundary information input.

layer  $\mathbf{x}_h$ :

$$\begin{aligned} net_{\eta j}^t &= \sum_i w_{ij}^a x_{ai}^t + \sum_{j'} w_{jj'}^n x_{hj'}^{t-1} \\ x_{\eta j}^t &= \phi_{\eta}(net_{\eta j}^t), \end{aligned} \quad (2)$$

where the activation function  $\phi_{\eta}$  is linear,  $i$  is the index of the input layer unit,  $j$  and  $j'$  are the indices of the hidden layer unit.

The hidden cell state  $\mathbf{x}_h$  is then determined by fusing the input layer activities with the previous hidden layer activities, dependent on the update gate's current activity:

$$x_{hj}^t = x_{\zeta}^t x_{hj}^{t-1} + (1 - x_{\zeta}^t) x_{\eta j}^t \quad (3)$$

Finally, the output of the event switching layer, fed to the event processing layer, is defined using a regular feed-forward pass:

$$\begin{aligned} net_{oj}^t &= \sum_i w_{ij}^o x_{hi}^t \\ x_{oj}^t &= \phi_o(net_{oj}^t), \end{aligned} \quad (4)$$

where the respective activation function  $\phi_o$  is linear.

**Backwards pass** To update the parameters of the network, standard back-propagation through time is used throughout the model, which adapts the weights using ADAM. The error signal defined by the squared reconstruction loss, which is determined at the prediction output of the event processing layer. Thus, the full model is trained end-to-end.

### Event Boundary Processing

In this work, we consider three different variants of SUGAR, concerning how surprise is processed and thus how event boundaries are anticipated. First, we calculate a surprise signal by monitoring the prediction errors in the event processing module (Fig. 2a). As expected, the resulting system responds

to event boundaries after the fact, instead of anticipating future changes. Second, we provide fuzzy, hidden ground truth information about event boundaries, which is processed by the event boundary anticipation module and then passed onto the event switching module via  $x_s^t$  (Fig. 2b). Lastly, we add counterfactual regularization (CFR) to improve event boundary predictions further (Fig. 2c).

**Explicit Surprise Processing** We first implemented SUGAR<sub>a</sub>, which calculates surprise explicitly (Fig. 2a) and invokes event transitions upon its detection. Surprise is calculated by maintaining a moving average of the recent prediction error and the standard deviation of this error with a low pass filter rate of 0.1. The raw surprise value is determined by calculating the current mean absolute difference between the event processing module's output  $y_{act}^t$  and the target value  $\hat{y}^t$ . Thus, the larger the difference, the greater the surprise. We then pass this value through a suitably parameterized sigmoidal function, which yields zero for hardly any surprise and values close to one for values with large surprise. The resulting surprise estimation value is then directly used as the input  $x_s^t$  to the update gate.

**Event Boundary Anticipation** Seeing that calculated surprise can only be processed after the fact, we next add an event boundary anticipation module, yielding SUGAR<sub>b</sub>. This module aids the event processing module to preempt surprise and thus to generate fluid event transitions. It processes event boundary information as input and generates  $x_s^t$  as output (Fig. 2b). While this module can indeed be trained to preempt surprise, our evaluations have shown that the learning progress is not as robust as desired and the developing latent event-predictive encodings are not fully stable. We thus continue with introducing counterfactual regularization to the event boundary anticipation module.

**Counterfactual Regularization (CFR)** The event boundary anticipation module receives its gradient signal  $\delta_{\zeta}^t$  from

the event processing layer through the update gate. We incorporate the counterfactual regularization as an additional term directly into the gradient equation of the update gate:

$$\delta_{\zeta,reg}^t = \delta_{\zeta}^t + \beta \left( \sqrt{(y_{act}^t - \hat{y}^t)^2} - \sqrt{(y_{cf}^t - \hat{y}^t)^2} \right), \quad (5)$$

where  $\beta$  indicates the gate status, taking on a value of 1 if the gate was open and 0 otherwise, although more nuanced and scaled activities of  $\beta$  may be useful in the future for more complex systems.  $y_{act}^t$  is the actual prediction of the event processing network, while  $y_{cf}^t$  is the alternative prediction had the gate been on the counterfactual state.  $\hat{y}^t$  is the true label. During back-propagation  $\delta_{\zeta,reg}^t$  thus regularizes the utility of gate openings.

The comparison of the two possible prediction outcomes indicates the amount of error that was avoided or created, depending on whether the difference in Eq. 5 yields a negative or positive value, respectively. Note that a closed gate is unaffected by this additional loss component, since  $\beta = 0$ . CFR thus supports the opening of gates when the gate activity decreases prediction error (preventing surprise) by actively comparing it with what would have happened had the gate remained closed (cf. SUGAR<sub>c</sub>, Fig. 2c). As a result of more selective gate openings, the latent event-predictive encodings  $x_{oj}^t$  can be expected to develop more static latent codes, thus providing better guidance for generating the appropriate sequences via the event processing layer. In the future, this mechanism may be useful for gate control in more challenging applications, where overlapping event structures may need to be segmented by means of controlling an array of gates.

## Results

We now evaluate the three SUGAR implementations, revealing its capabilities, justifying the addition of both the event boundary anticipation module and the CFR.

### Calculated surprise

SUGAR<sub>a</sub> uses prediction error to calculate surprise. This resulted in gate openings that occurred immediately after the switch between problems, in addition to several openings during the events. The resulting prediction error was larger when compared with the one obtained when the perfect surprise signal was provided, but smaller when the gate was kept closed (cf. Figure 3). Thus, using calculated surprise signals to modulate top-down control improves performance.

The error dynamics in Figure 3 (middle) show error spikes at the event transition boundaries, indicating that the event processing module indeed learns about the intra-event regularities in the data. However, even though surprise is registered, the reactive processing of surprise cannot prevent error spikes because surprise only offers post-hoc guidance, without any anticipatory prediction. As shown in Figure 3 (bottom), providing ground-truth knowledge about upcoming event boundaries can clearly improve performance, effectively avoiding the error spikes. Thus, the performance of

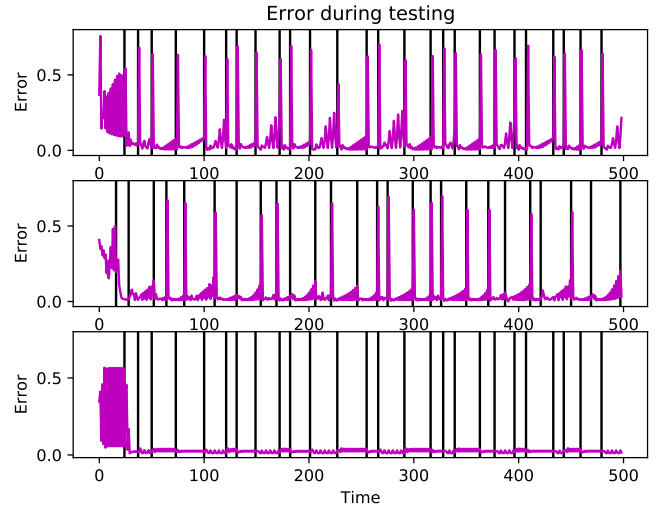


Figure 3: Prediction error during testing when the update gate is always closed (top), controlled by SUGAR<sub>a</sub> (middle), or precisely opened only at event switches (fully informed SUGAR<sub>a</sub>, bottom). Black lines indicate event transition boundaries.

SUGAR can be significantly improved when the gate is able to open in anticipation of, rather than in reaction to surprising outcomes.

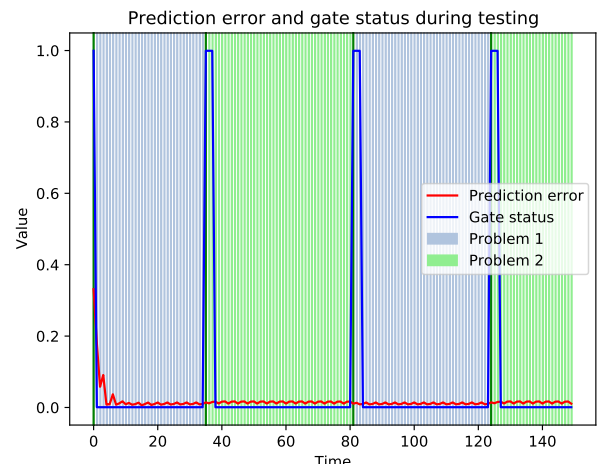


Figure 4: Prediction error and surprise signals during testing on Problem1+2 using SUGAR<sub>b</sub>, one hot plus event boundary indicator plus random input.

### Event Boundary Anticipation Module

In order to achieve more fluid transitions between events, we extended the structure by adding the event boundary anticipation module as detailed above (SUGAR<sub>b</sub>). This module receives event boundary information in the form of an increased value when a transition is about to happen. It utilizes this signal to control the gate, ideally opening it right before an event transition occurs, rather than right after.

We tested SUGAR<sub>b</sub> using Problem 1+2. Using the event boundary anticipation module to modulate top-down control

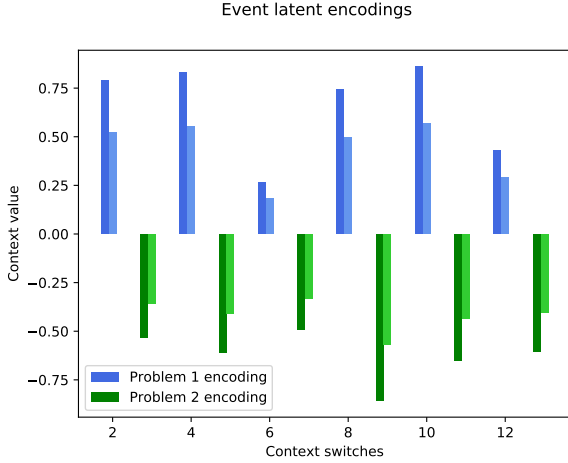


Figure 5: Latent event encodings ( $\mathbf{x}_o$ ) for Problem 1+2 emerging in SUGAR<sub>b</sub>.

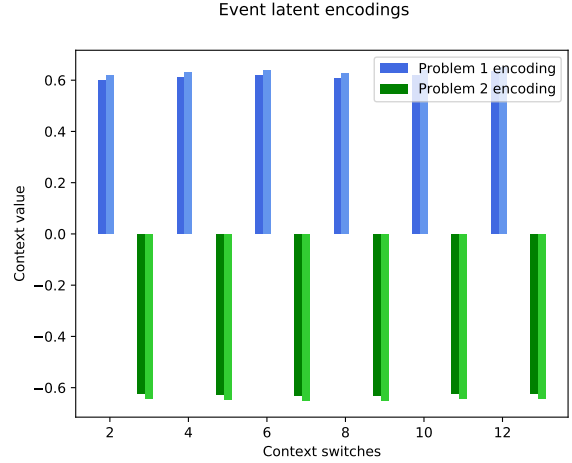


Figure 7: Latent event encodings ( $\mathbf{x}_o$ ) for Problem 1+2 emerging in SUGAR<sub>c</sub>.

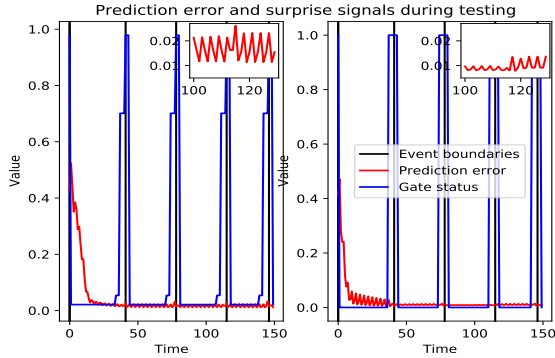


Figure 6: Gate status and prediction error within events and at events boundaries for SUGAR<sub>b</sub> (left) and SUGAR<sub>c</sub> (right).

leads to predictions that are as accurate as the ones generated when the perfect surprise signal is simply provided just in time. Even when random binary values  $[0, 1]$  are added to the event boundary signal EB, the network still learns to recognize the ones that indicate the context switch and learns to open the gate at the right moment. The prediction error dynamics shown in Figure 4 indicate fluent transitions between events. The gate status additionally shows how gate spikes anticipate contextual switches. This allows for signalling anticipatory, contextual encoding of events just before they switch, thus enabling more accurate predictions on the lower-level event processing layer.

### Counterfactual Regularization

Surprise signals offer guidance about when the current event has ended and when to switch to the new context of the next event. In the real world, we often receive signals from complementary cognitive systems that foreshadow when the current event is about to end (e.g. from the visual system as the perceived distance between one’s hand and a glass of water approaches zero). We mimicked this idea by gradually changing the input to the event boundary module be-

fore the switch, reaching a particular signal value just before the actual event switch. The results in Figure 6 confirm that this leads to gate openings in anticipation of the switch. However, closer inspections of the developing latent event codes revealed that these codes tended to dynamically change across subsequent event switches (Fig. 5), which still results in slightly increased surprise signals between events (Fig. 6 left).

In order to increase stability during events, we thus added the CFR term (Fig. 2c; Eq. 5). SUGAR<sub>c</sub> re-runs the event processing layer at all times steps  $t$  when the gate was open with the counterfactual state (i.e. gate closed) of the event switching gate. Then, we compute the difference between the two prediction errors, thus inducing a penalty for non-beneficial gate openings (Equation 5). As a result, the network avoids unnecessary gate openings.

Our results show that CFR indeed produces more stable latent event encodings, which do not change over time problem-respectively (cf. Figure 7). Moreover, Figure 6 shows that the consequent more focused gate openings in anticipation of an event switch fully avoid both error spikes and unnecessary gate activities during events (Fig. 6, right-hand side).

**Latent event encodings** Finally, we examine the latent event encodings  $\mathbf{x}_o$  learned by the event anticipation layer. Since CFR enforces maximally static, event-specific encodings, we expected that the top-down signals from the event switching layer  $\mathbf{x}_o$  may capture similarities between the event-respective dynamics in the event processing layer. Using Problem 1-3 (Fig. 1), we investigated the latent event encodings developed in SUGAR<sub>c</sub> (Fig. 2c) for the three problems.

Figure 8 shows the respective encodings for different runs with differently initialized networks. Note that the encodings for Problem 3 are always more similar to that of Problem 1. In fact, they somewhat lie between the encodings of Problem

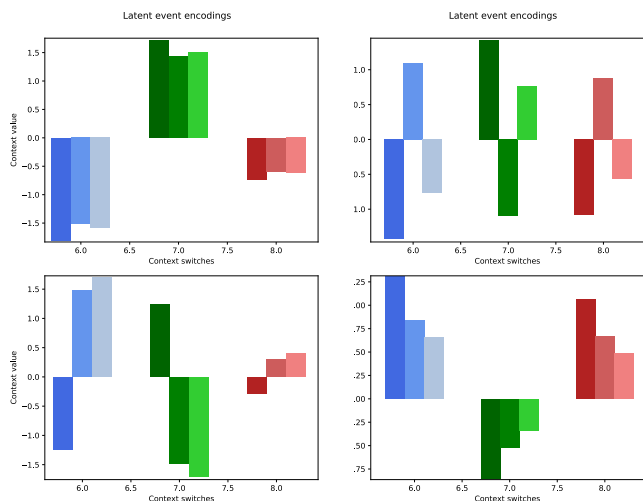


Figure 8: Latent event encodings ( $\mathbf{x}_o$ ) for four randomly initialized networks trained on Problem 1-3. In all cases, the code for Problem 3 (red bars) lies in between the code for Problem 1 and 2 (blue and green bars, respectively).

1 and Problem 3, but closer to Problem 1. This is because, apart from the fact that the symbols in Problem 1 are used in Problem 3 as well, Problem 3 starts with the same symbol sequence as does Problem 1. Still, there is a clear difference, since the symbol subsequence of Problem 2 is also contained in Problem 3. Thus, the latent encodings tend to reflect properties of semantics and compositionality—in this case with respect to sequential structures—a claim that should certainly be investigated further.

## Discussion

In this work, we introduced SUGAR—a normative neural network model that yields event-predictive encodings, which exhibit properties suggestive of compositionality. These encodings provide top-down contextual guidance for predicting hierarchically generated sequences, where a gating layer modulates this control. We compared both prediction error trained surprise as well as using noisy signals about event boundaries. For the latter to be effective, we introduce a novel counterfactual regularization term, which yields more accurate predictions, particularly at event boundaries, and more stable latent event encodings.

An important limitation in our current work is the need to provide an external signal that foreshadows event switches. This was inspired by the fact that orienting reflexes or other sensory signals often provide good indications for impending event boundaries (e.g., when our hand touches a glass or when we reach the beginning of a flight of stairs; Baldwin & Kosie, 2021; Kuperberg, 2021). Future work could improve on these methods by using anticipatory signals from the event-predictive processing module to prepare for event transition processing, possibly in an attention-based manner. Additionally, the scalability of the network as well as other types of event-like sequence processing tasks (e.g., Humaidan, Otte, & Butz, 2020) should be investigated. We ex-

pect to thus enable the networks to guide event processing even more effectively by means of the developing, predictive process-focusing, top-down event encodings.

## Conclusion

Event cognition plays a crucial role in providing contextual guidance for solving problems and interpreting the world around us (Radvansky & Zacks, 2014). An understanding of the correct event context is important to foster the effective compression of the sensorimotor processing stream into compositional latent encodings for both artificial and biological agents (Gershman & Niv, 2010). Yet in order to effectively learn hidden structures, inductive learning biases are necessary (Battaglia et al., 2018; Franklin et al., 2020). Here we studied how the gating of top-down control via surprise and the use of counterfactual regularization can help for both avoiding unnecessary gate openings and learning more stable, latent event-predictive encodings with compositional properties. Seeing that event-predictive structures are closely interlinked with language structures and can suitably simulate explanatory thought processes (Butz et al., 2021), we believe that SUGAR-based architectures with counterfactual regularization mechanisms may be useful for developing useful hierarchical and compositional encodings in other domains and tasks—and thus particularly for developing more advanced machine learning systems (Butz, 2021). Our results provide preliminary evidence that gating mechanisms combined with counterfactual error regularization are an important inductive learning bias, which may help bridge the gap between artificial and human intelligence.

## Acknowledgments

This research was funded by the German Research Foundation (DFG) within Priority-Program SPP 2134 - project “Development of the agentive self” (BU 1335/11-1, EL 253/8-1) and under Germany’s Excellence Strategy – EXC 2064/1 – 390727645. The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting DH and CG and the German Federal Ministry of Education and Research (BMBF): Tübingen AI Center, FKZ: 01IS18039A for supporting CMW.

## References

- Baldwin, D. A., & Kosie, J. E. (2021). How does the mind render streaming experience as events? *Topics in Cognitive Science, 13*, 79–105.
- Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., ... Pascanu, R. (2018). Relational inductive biases, deep learning, and graph networks. *ArXiv e-prints, 1806.01261*.
- Butz, M. V. (2021). Towards strong AI. *Künstliche Intelligenz, 35*, 91-101. doi: 10.1007/s13218-021-00705-x
- Butz, M. V., Achimova, A., Bilkey, D., & Knott, A. (2021). Event-predictive cognition: A root for conceptual human thought. *Topics in Cognitive Science, 13*, 10-24. doi: 10.1111/tops.12522

- Butz, M. V., & Kutter, E. F. (2017). *How the mind comes into being: Introducing cognitive science from a functional and computational perspective*. Oxford, UK: Oxford University Press.
- Cho, K., van Merriënboer, B., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (emnlp)* (pp. 1724–1734). Doha, Qatar.
- Daw, N. D., Niv, Y., & Dayan, P. (2005). Uncertainty-based competition between prefrontal and dorsolateral striatal systems for behavioral control. *Nature neuroscience*, 8(12), 1704–1711.
- Franklin, N. T., Norman, K. A., Ranganath, C., Zacks, J. M., & Gershman, S. J. (2020). Structured event memory: A neuro-symbolic model of event cognition. *Psychological Review*, 127(3), 327.
- Gershman, S. J., & Niv, Y. (2010). Learning latent structure: carving nature at its joints. *Current opinion in neurobiology*, 20(2), 251–256.
- Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8), 1735–1780.
- Humaidan, D., Otte, S., & Butz, M. V. (2020). Fostering event compression using gated surprise. In I. Farkas, P. Masulli, & S. Wermter (Eds.), *International conference on artificial neural networks – icann 2020* (pp. 155–167). Springer.
- Kuperberg, G. R. (2021). Tea with milk? a hierarchical generative framework of sequential event comprehension. *Topics in Cognitive Science*, 13, 256–298.
- Lee, S. W., Shimojo, S., & O’Doherty, J. P. (2014). Neural computations underlying arbitration between model-based and model-free learning. *Neuron*, 81(3), 687–699.
- Radvansky, G. A., & Zacks, J. M. (2014). *Event cognition*. Oxford University Press.
- Schapiro, A. C., Rogers, T. T., Cordova, N. I., Turk-Browne, N. B., & Botvinick, M. M. (2013). Neural representations of events arise from temporal community structure. *Nature neuroscience*, 16(4), 486–492.
- Shin, Y. S., & DuBrow, S. (2021). Structuring memory through inference-based event segmentation. *Topics in Cognitive Science*, 13, 106–127. doi: 10.1111/tops.12505
- Zacks, J. M., Kurby, C. A., Eisenberg, M. L., & Haroutunian, N. (2011). Prediction error associated with the perceptual segmentation of naturalistic events. *Journal of Cognitive Neuroscience*, 23, 4057–4066.
- Zacks, J. M., Speer, N. K., Swallow, K. M., Braver, T. S., & Reynolds, J. R. (2007). Event perception: A mind-brain perspective. *Psychological Bulletin*, 133(2), 273–293. doi: 10.1037/0033-2909.133.2.273
- Zhu, Y., Gao, T., Fan, L., Huang, S., Edmonds, M., Liu, H., ... Zhu, S.-C. (2020). Dark, beyond deep: A paradigm shift to cognitive AI with humanlike common sense. *Engineering*, 6(3), 310–345. doi: 10.1016/j.eng.2020.01.011