REPRISE: A Retrospective and Prospective Inference Scheme

Martin V. Butz (martin.butz@uni-tuebingen.de)
Department of Computer Science and Department of Psychology, University of Tübingen
Sand 14, 72076 Tübingen, Germany

David Bilkey (alik@cs.otago.ac.nz)
Department of Psychology, University of Otago
P.O. Box 56, Dunedin, New Zealand

Alistair Knott (alik@cs.otago.ac.nz)
Department of Computer Science, University of Otago
P.O. Box 56, Dunedin, New Zealand

Sebastian Otte (sebastian.otte@uni-tuebingen.de)
Department of Computer Science, University of Tübingen
Sand 14, 72076 Tübingen, Germany

Abstract

Motivated by the close relation of predictive codings and active inference to cognition, we introduce a corresponding artificial neural network-based (ANN) adaptation process, which we term REPRISE: REtrospective and PRospective Inference SchEme. REPRISE first executes a *retrospective* inference process, inferring the unobservable contextual state that best explains its recently encountered sensorimotor experiences. It then executes a *prospective* inference process, inferring upcoming motor activities in the light of the inferred contextual state and a given goal state. First, the ANN – a recurrent neural network – is trained to learn one sensorimotor temporal forward model, that is, the sensorimotor contingencies generated by the behavior of three moving or flying vehicles. During training, additional three bits are provided as input, indicating which mode currently applies. After training, goal-directed control and system state inference are activated: Given a goal state, the system imagines a motor command sequence optimizing it with the prospective objective to minimize the distance to the goal. Meanwhile, the system evaluates the encountered sensorimotor contingencies retrospectively, adapting its vehicle estimation activities and, in order to maintain coherence, the neural hidden states accordingly. This ANN's 'mind' is thus continuously imagining the future and reflecting on the past – showing superior performance on the posed control problems. The architecture effectively demonstrates that neural error signals and neural activities can be projected into the past and into the future, respectively, optimizing both neural context codes that approximately generate the recent past and upcoming behavior in the light of desired goal states.

Keywords: artificial neural networks; forward model learning; inverse sensorimotor control; active inference; dynamics; adaptation; cognitive systems

Introduction

While the predictive brain and active inference principles have strongly influenced cognitive science over the last years (Butz & Kutter, 2017; Clark, 2016; Friston, 2009), it remains unclear how these principles can be implemented in scalable, artificial neural networks. Moreover, it remains unclear how effective abstraction and generalized structures can be realized (Botvinick & Weinstein, 2014; McClelland et al., 2010). Despite the recent remarkable successes in playing challenging computer games and the board game GO (Mnih et al., 2015; Silver et al., 2016), neural systems that are able to infer the current system state, conceptual abstractions, and goal-directed motor control concurrently remain out of reach to large extents (Butz, 2016; Butz & Kutter, 2017).

Perceptual processing reveals a strong tendency towards the segmentation of the continuous sensorimotor stream into meaningful event and event-transition encodings (Zacks & Tversky, 2001; Zacks, Speer, Swallow, Braver, & Reynolds, 2007). On the other hand, the theory of event coding has proposed integrative action-effect codes (Hommel, Müsseler, Aschersleben, & Prinz, 2001). Similarly, forward-inverse control schemes have modeled human behavior, (Wolpert & Kawato, 1998; Wolpert & Flanagan, 2016), where the involved forward-inverse models essentially encode interaction events. Even the memorization of experienced episodes appears event-segmented and event-focused. Moreover, memorized events can be used not only for processing current sensorimotor information, but also for reflecting on the past and for imagining potential futures (Schacter et al., 2012). In sum, our mind seems to cluster sensorimotor contingencies into predictive events. As a result, our state of mind may be viewed as continuously adapting to the current event-respective circumstances, but also to past, future, or even fully hypothetical events (Bar, 2009; Buckner & Carroll, 2007).

In close relation to these aspects, we here present an retrospective and prospective temporal inference scheme (REPRISE), which can be applied in recurrent ANNs. REPRISE infers retrospectively the unobservable current event context, which best explains the recent sensorimotor experiences, while it infers motor control commands prospectively in a goal-directed manner. We thereby build on our previous work, which had accomplished prospective, active motor control inference (Otte, Schmitt, Friston, & Butz, 2017; Otte, Zwiener, & Butz, 2017) but not retrospective system state inference. Our results suggest that REPRISE can yield effective goal-directed control and concurrent system state inference.

System Architecture and Inference Mechanisms

In order to introduce REPRISE, we distinguish between the actual (not directly observable) dynamical system ϕ and the model Φ of this system, which is encoded by a recurrent artificial neural network (RNN). Focusing on a discrete-time dynamical system, at a certain point in time t, the current state of the dynamical system ϕ may be denoted by $\boldsymbol{\vartheta}^t$, such that the progression through time is determined by

$$\boldsymbol{\vartheta}^t \xmapsto{\phi} \boldsymbol{\vartheta}^{t+1}. \tag{1}$$

Temporal Forward Model

The model Φ is trained to approximate these dynamics, inferring its parameters from sensorimotor experiences during learning. However, seeing that we are dealing with a dynamic, *partially observable Markov decision process* (POMDP) (Sutton & Barto, 1998), the true system state $\boldsymbol{\vartheta}^t$ is typically not directly deducible from current observables $\mathbf{s}^t \in \mathbb{R}^n$. Thus, the dynamical system's internal state $\boldsymbol{\sigma}^t$ must be inferred in each iteration given the current observables $\mathbf{s}^t$ as well as the current motor activities denoted by $\mathbf{x}^t \in \mathbb{R}^k$. With the help of the system's model Φ, the next system state $\boldsymbol{\sigma}^{t+1}$ and the consequent sensory expectations $\tilde{\mathbf{s}}^{t+1}$ are determined by

$$(\mathbf{s}^t, \boldsymbol{\sigma}^t, \mathbf{x}^t) \xmapsto{\Phi} (\tilde{\mathbf{s}}^{t+1}, \boldsymbol{\sigma}^{t+1}), \tag{2}$$

where the mapping Φ essentially models the temporal forward dynamics of the system. Thus, the next system state and sensory expectations depend on the current sensor ($\mathbf{s}^t$) and motor control ($\mathbf{x}^t$) activities as well as, in principle, on the entire state history, which is encoded in the (hidden) state components ($\boldsymbol{\sigma}^t$) in a compressed form.

While learning the model, that is, while pursuing model inference, the system essentially attempts to minimize the quadratic loss between predicted and encountered sensory information over time, that is,

$$\mathcal{L} = \sum_{t=1}^{T} \sum_{i=1}^{n} \frac{1}{2} (\tilde{\mathbf{s}}_i^t - \mathbf{s}_i^t)^2, \tag{3}$$

summing the accumulated losses over the gathered experiences $\{\mathbf{s}_1, \dots, \mathbf{s}_T\}$.

Multiple Dynamical Systems

In this paper, however, we consider not only a single dynamical system, but an ensemble of multiple systems $\phi = \{\phi_1, \dots, \phi_u\}$, reflecting discretely different dynamic contingencies that can occur in the world. These systems differ from each other concerning their behavior, but share the same input, state, and output dimensions. During model inference, the model Φ is trained to approximate all of these dynamical systems within one single RNN architecture. Specifically, the task of the temporal forward model inference process is to approximate that particular dynamical system ϕ_i that is currently active, given observed state $\mathbf{s}^t$ and control commands $\mathbf{x}^t$, as well as the internal system state estimation σ^t.

During model inference, the identity of the currently active dynamical system ϕ_i is represented by means of a context input vector $\mathbf{c} \in \mathbb{R}^u$, which is encoded as a one-hot vector in which the i-th component is set to 1. This encoding is closely related to parametric bias encodings, which can be viewed as an indicator of the current control event the system is situated in (Sugita, Tani, & Butz, 2011; Tani, 2017).

REPRISE

Given an imagined action sequence, an initial state, and the identity of the current dynamical system, the RNN can predict a state progression that is expected when executing the action sequence by means of the learned temporal forward model Φ. To effectively control the system, however, the inverse mapping is required, that is, an action sequence needs to be inferred to approach a desired goal-state (or follow a sequence of goal-states) from an initial state. This becomes even more difficult when the identity of the current actual dynamical system ϕ_i is unknown and has to be inferred as well. In this section we introduce the REPRISE algorithm, a concurrent retrospective and prospective inference scheme, which solves the dual system identification and goal-directed control problem.

Figure 1 shows the dynamic processes REPRISE unfolds for two consecutive time steps. During each step, both a retrospective and a prospective inference phase are executed.

In the retrospective phase, the gradient is propagated R time steps into the past, to reflect on the states that were just experienced. The gradient is fed by the discrepancy between previously predicted system states $\tilde{\mathbf{s}}^{t-i}$, with $i \in 0, \dots, R$, and the actually observed system states $\mathbf{s}^{t-i}$. The discrepancy is then mapped onto the assumed context input $\mathbf{c}^{t-i}$ – essentially a subvector of $\mathbf{s}^{t-i}$ – indicating the dynamical system ϕ_i that is presumably currently active. Additionally, the gradient is used to adapt the RNN's hidden state at time step $t-R-1$, that is, σ^{t-R-1}, such that it better fits the changing context input. As a result, the RNN avoids disadvantageous or even undefined sensory input, motor command, hidden state combinations. Finally, the neural activities are propagated forward again to the present time step, with respect to the inferred hidden state and context input, and the already recorded motor commands and observed system states, yielding an updated σ^t. This retrospective inference cycle is executed r times.

In the prospective phase, neural activities are projected P time steps into the future, starting with the inferred current internal system state σ^t and hypothetically executing a sequence of motor commands $\tilde{\mathbf{x}}^{t+i}$, which was inferred previously. The discrepancies between the predicted future $\tilde{\mathbf{s}}^{t+i}$ and desired goal state sequences $\overset{\star}{\mathbf{s}}^{t+i}$, with $i \in 1, \dots, P$, are then propagated backwards through time from the imagined future back to the present time step, while the gradient is projected onto the individual neurally encoded anticipated motor activity sequence $\tilde{\mathbf{x}}^{t+i}$, effectively optimizing it in the light of the current system state estimates and the desired goal state. This prospective inference cycle is executed p times.

After the retro- and prospective inference phases, the inferred motor activity $\mathbf{x}^t$ is executed by the system ϕ and the forward RNN is updated via (2). This closes the processing loop, repeating REPRISE in the following time step $(t+1)$.

System Evaluations

Our experiments are based on a two dimensional dynamical system simulation of $u = 3$ types of vehicles, constituting three dynamical systems: ϕ_1 is a multi-copter-like vehicle, which we call *rocket*, ϕ_2 is a static omnidirectional vehicle, which we call *stepper*, and ϕ_3 is a dynamical, omnidirectional gliding vehicle, which we call *glider*. The rocket is influ-

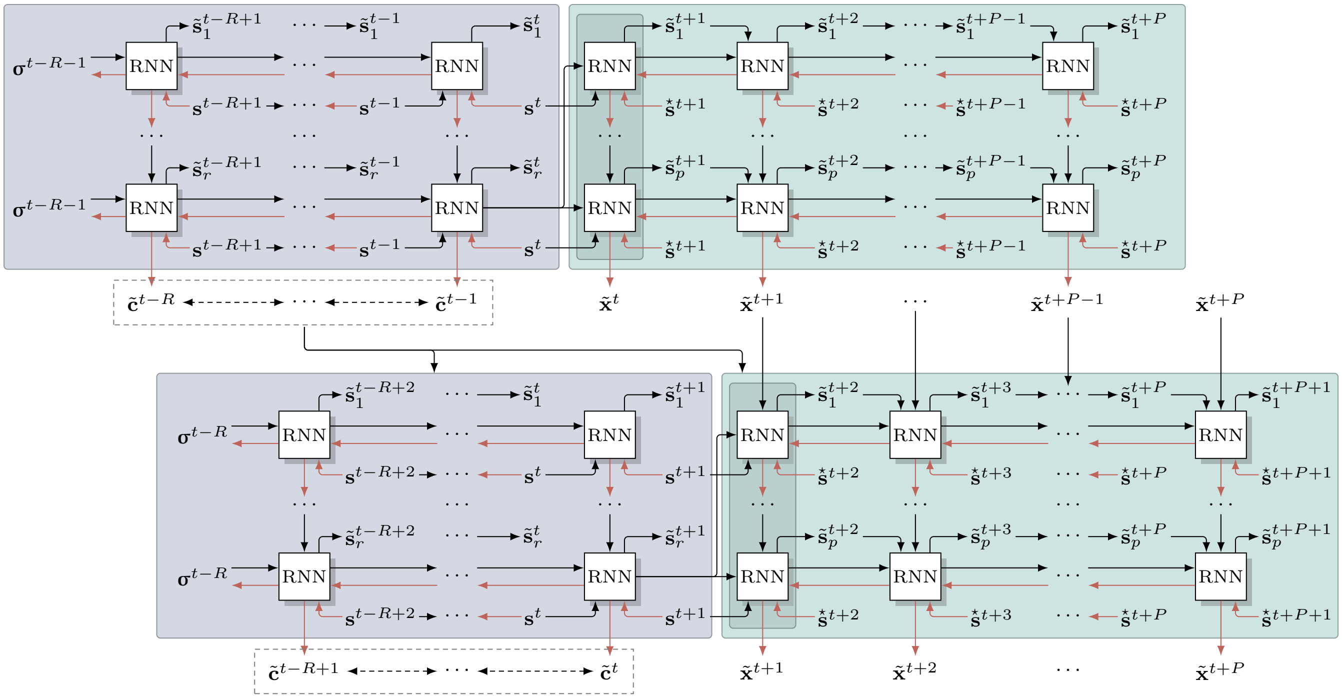

Figure 1: Illustration of REPRISE for two consecutive time steps t (top part) and $t+1$ (bottom part). The right boxes illustrate future imaginations actively inferring prospective motor activities, while the left boxes show retrospective reflections of the recent past for system state inference, including the system identities $\mathbf{c}^{t'}$ as well as the corresponding neural hidden states $\boldsymbol{\sigma}^{t}$. Black lines indicate context and information forward flow, while the red lines indicate gradient flow. $\tilde{\mathbf{x}}^{t'}$ and $\tilde{\mathbf{c}}^{t'}$ refer to the action and context input vectors, respectively, for a particular time step t'. $\tilde{\mathbf{s}}_{\tau}^{t'}$ refers to a particular sensory prediction in the τ-th optimization cycle, whereas $\overset{\star}{\mathbf{s}}^{t'}$ refers to a desired sensory goal state.

enced by simulated gravity and undergoes inertia. It has two propulsion motors that are spread at a 45° angle from the vertical axis to both sides, inducing thrust forces in the respective direction. The two other motor inputs are irrelevant for the rocket. The stepper has four thrust motors that are spread at 45° and 135° angle from the vertical axis to both sides, inducing steps in the opposite direction. Finally, the glider has the same four thrust motors as the stepper. However, in contrast to the stepper, the glider undergoes inertia without any friction. Each motor unit can be throttled within the interval $[0,1]$. Upon invocation, each vehicle is positioned in a rectangular free space of size 3 by 2 units. It is surrounded by borders, which block the vehicle.

Model Prediction Performance

During training, we use stochastic backpropagation through time to optimize the model parameters, that is, the weights of the considered RNN architectures based on simulated sensorimotor experiences. The sensorimotor experiences are encountered by executing somewhat random motor commands in $[0,1]^4$, where motor command generation is structured such that sufficient upwards thrust is generated and a reasonable exploration of the complete rectangular free space is loosely ensured.

At each time step the network is fed with the current position of the vehicle, the current motor commands (activities of the four thrust motors, where in the case of the rocket the second two motors have no effect), and a three bit one-hot vector, which indicates the vehicle that is currently controlled, i.e., which ϕ_i applies. The network output is the prediction of the resulting change of the vehicle position.

We trained the considered RNNs in 3000 epochs, consisting of 2000 control steps each. We applied backpropagation through time every 50 iterations and Adam as the weight adaptation mechanism (Kingma & Ba, 2014). The learning rate was annealed, such that $\eta = 0.01$, $\eta = 0.001$, $\eta = 0.0001$ during the first, second, and third 1000 sequences (first and second moment smoothing factors were set to the standard values $\beta_1 = 0.9$, $\beta_2 = 0.999$). Each vehicle was simulated for 2000 time steps (i.e. on epoch), after which the hidden state of the RNN was reset to zero and a new vehicle was initialized.

Figure 2 contrasts the sensory prediction error development during learning for several RNN architectures, showing averaged mean errors and standard deviations across 20 independently weight-initialized (normally distributed values with standard deviation 0.1) networks. Standard RNNs with one hidden layer of 27 (1026), 36 (1692), and 54 neurons (3510 weights) perform consistently worse than long short-term memory (LSTM) RNNs with forget gates and peep-

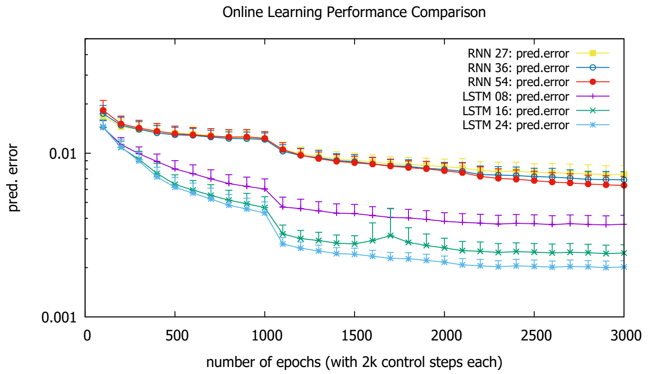

Figure 2: Learning progress comparing standard RNNs and LSTMs with different numbers of hidden units.

hole connections (Gers, Schraudolph, & Schmidhuber, 2002). While 16 hidden memory cells (1680 weights) clearly outperform 8 hidden memory cells (584 weights), the advantage of yet another 8 hidden cells, that is, 24 cells in total (3288 weights) is less pronounced.

REPRISE Performance

To evaluate the robustness and abilities of REPRISE, including all relevant settings, we contrast the resulting control performance of the RNN with 36 hidden neurons with the LSTM with 16 hidden units, which have approximately an equal number of weights (1692 versus 1680, respectively). Each network was tested to reach a sequence of 50 uniformly randomly positioned targets within the centered inner area of size 1.5 time 1.5 units of the rectangular space. Thereby, the simulation is divided into a sequence of discrete events, where the agent becomes one of the vehicles ϕ_i for a continuous series of 150 timesteps. One of the agents tasks is to infer which of these events is under way at any given time. The values in the tables below are averages over the 20 independently trained networks and 50 considered targets, whereby the target positions and vehicle succession were the same for all runs.

We applied Adam in all inference processes. Prospective inference was always looking $P = 7$ steps into the future, executing the inference cycle $p = 20$ times. Detailed evaluations were run contrasting different learning rates η_c and η_σ for the retrospective context $\mathbf{c}$ and system state σ inference. In our standard setting, retrospective inference covered $R = 20$ time steps into the past, while $r = 20$ inference cycles were performed. Note that during optimization the motor commands and the context inputs were clamped to their value range $[0,1]$, and the neural hidden states σ were clamped in accordance to the range of the respective neurons' activation function.

Figure 3 shows a typical flight sequence generated by an LSTM controlled by the REPRISE algorithm, in ten iteration steps. Although glider and rocket initially slightly overshoot the target, they quickly zoom in. For the stepper, the projected path is less direct, which is probably partially the case because the goal is simply not directly reachable in seven steps. It should be noted that although the images suggest that the motor effort while staying at the goal is minimized, this is not always the case, as there is currently no incentive in the system that stresses motor effort minimization.

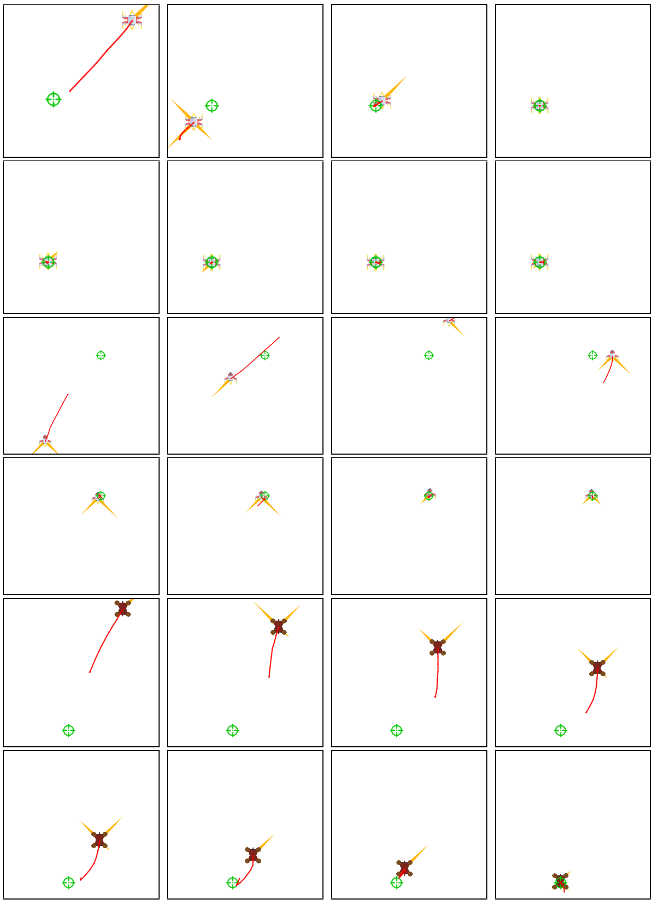

Figure 3: Typical subsequence flight sequences for the three different vehicle types controlled by REPRISE: rocket, glider, and stepper.

Tables 1 and 2 show the average distance to the goal location that remained after 150 time steps, that is, control iterations. The first row of results shows the performance when the context bits are set to the correct values. Given sufficiently small learning rates η_σ, additional retrospective system state inference yields similar performance. The next four rows show the performance of REPRISE with different learning rate combinations, without providing the context bit values. Clearly, overly large or small values yield mediocre performance, however, it appears that quite a large value range yields robust target reaching behavior. Consistently the best setting is with $\eta_c = .01$ and $\eta_\sigma = .001$, adapting the context bits ten times faster than the hidden states, which is most likely the case because without proper context inputs, overly fast hidden state estimations will lead to unstable adaptations. Thus, most robust performance is reached when both, context and hidden state activities are adapted, yielding performance

Table 1: LSTM: Distance to target after 150 control steps

Average	η_σ=0	η_σ=1e-4	η_σ=.001	η_σ=.01	η_σ=.1
c set	0.006	0.006	0.006	0.007	0.051
η_c=1e-4	0.082	0.057	0.034	-	-
η_c=.001	0.039	0.024	0.011	0.011	-
η_c=.01	0.024	-	0.006	0.007	0.055
η_c=.1	0.025	-	-	0.008	0.038
Rocket	η_σ=0	η_σ=1e-4	η_σ=.001	η_σ=.01	η_σ=.1
c set	0.014	0.014	0.013	0.015	0.095
η_c=1e-4	0.069	0.052	0.022	-	-
η_c=.001	0.041	0.037	0.013	0.011	-
η_c=.01	0.026	-	0.006	0.011	0.069
η_c=.1	0.028	-	-	0.010	0.051
Stepper	η_σ=0	η_σ=1e-4	η_σ=.001	η_σ=.01	η_σ=.1
c set	0.002	0.001	0.001	0.001	0.006
η_c=1e-4	0.063	0.043	0.022	-	-
η_c=.001	0.037	0.017	0.008	0.014	-
η_c=.01	0.024	-	0.007	0.005	0.035
η_c=.1	0.025	-	-	0.004	0.024
Glider	η_σ=0	η_σ=1e-4	η_σ=.001	η_σ=.01	η_σ=.1
c set	0.004	0.004	0.004	0.004	0.053
η_c=1e-4	0.116	0.078	0.060	-	-
η_c=.001	0.040	0.017	0.012	0.009	-
η_c=.01	0.021	-	0.005	0.006	0.061
η_c=.1	0.021	-	-	0.010	0.039

Table 2: RNN: Distance to target after 150 control steps

Average	η_σ=0	η_σ=1e-4	η_σ=.001	η_σ=.01	η_σ=.1
c set	0.037	0.039	0.037	0.035	0.036
η_c=1e-4	0.162	0.019	0.019	-	-
η_c=.001	0.038	0.052	0.020	0.018	-
η_c=.01	0.040	-	0.017	0.019	0.029
η_c=.1	0.169	-	-	0.044	0.034

Table 3: LSTM: Average accumulated distance to target

Average	η_σ=0	η_σ=1e-4	η_σ=.001	η_σ=.01	η_σ=.1
c set	0.061	0.060	0.060	0.061	0.109
η_c=1e-4	0.157	0.139	0.109	-	-
η_c=.001	0.106	0.100	0.079	0.082	-
η_c=.01	0.090	-	0.072	0.077	0.127
η_c=.1	0.092	-	-	0.077	0.115

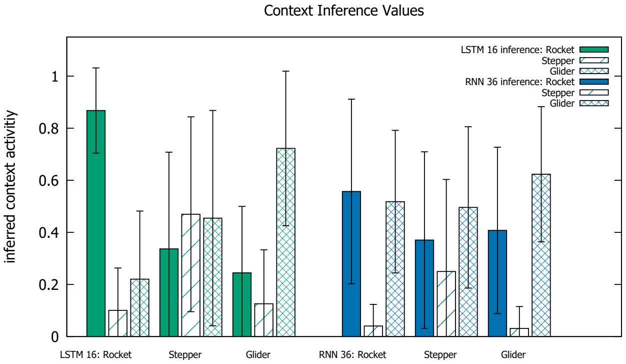

Figure 4: Inferred values of the context values **c**.

that is actually competitive – in the RNN case even superior – to the one when the context information is provided. We believe that superiority is achieved due to the additional gradient dynamics, which appear to prevent premature motor command convergence, hovering close to but not directly on the target.

Table 1 additionally shows the performance differences when focusing in on the three different vehicle types in the LSTM case. While the parameter dependencies are very similar, it appears that the rocket was hardest to control, most likely due to the fact that gravity needs to be continuously counteracted.

Table 3 shows for the LSTM case that the average distance to the target object over the 150 steps averaged over all vehicles is smallest when the context information is provided. This was indeed the case for all three vehicles (not shown). This is expectable as context inference inevitably yields erroneous behavior during the first control steps, confirming that the switch in the vehicle identity causes initial disruptions, which are quickly stabilized.

As a final evaluation result, Figure 4 shows the inferred context input activations for the three vehicles, contrasting again LSTM with RNN performance. Clearly, the LSTM architecture is better-suited to infer the underlying control system, although the estimates are still far from optimal. Using eye inspection, it was observed that once the goal has been reached, the estimates sometimes drifted off towards more incorrect estimates – probably because the sensorimotor information was not sufficiently informative. This observation in particular suggests that both context estimation stability could be enforced, switching only when error signals suggest to do so, and active motor inference may be further optimized to maintain high context estimation certainty (Friston et al., 2015), thus generating motor commands that minimize uncertainties in the model states estimates σ.

Summary, Conclusions & Future Perspectives

We have shown that REPRISE maintains an ANN neural activity that reflects on the past and projects its own state into the future, thus continuously optimizing its internal (generative) state estimates about its own body and the environment as well as imagined upcoming environmental interactions. We have developed this system as a first step towards sensorimotor-grounded, event-oriented abstractions. Essentially, the context vector **c**, which was provided as an input vector during training, signals the contextual "event" the system is currently in. Here we have shown that the event was inferable after model learning. An important next challenge is to foster the learning of contextual encodings during training – akin to parametric bias neural activities (Sugita et al., 2011; Tani, 2017) – probably by using unexpected changes in prediction errors as an indicator signal for an event change (Butz, 2016; Gumbsch, Otte, & Butz, 2017).

Once the automatic learning of event encodings is achieved, event-predictive cognition on the compact event-encoding level will become possible, potentially offering a patch towards conceptual and compositionally recombinable event schema abstractions. As an additional challenge, it should be kept in mind that the implemented processes currently fully rely on error backpropagation through time. Implementations of Bayesian inference or even more general free-energy-based inference processes along similar lines are well-imaginable.

Acknowledgments

Funding from the Feodor-Lynen Grant of the Humboldt Foundation is gratefully acknowledged.

References

Bar, M. (2009). Predictions: A universal principle in the operation of the human brain. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *364*(1521), 1181-1182. doi: 10.1098/rstb.2008.0321

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). doi: 10.1098/rstb.2013.0480

Buckner, R. L., & Carroll, D. C. (2007). Self-projection and the brain. *Trends in Cognitive Sciences*, *11*, 49-57.

Butz, M. V. (2016). Towards a unified sub-symbolic computational theory of cognition. *Frontiers in Psychology*, *7*(925). doi: 10.3389/fpsyg.2016.00925

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.

Clark, A. (2016). *Surfing uncertainty: Prediction, action and the embodied mind*. Oxford, UK: Oxford University Press.

Friston, K. (2009). The free-energy principle: a rough guide to the brain? *Trends in Cognitive Sciences*, *13*(7), 293 - 301. doi: 10.1016/j.tics.2009.04.005

Friston, K., Rigoli, F., Ognibene, D., Mathys, C., FitzGerald, T., & Pezzulo, G. (2015). Active inference and epistemic value. *Cognitive Neuroscience*, *6*, 187-214. doi: 10.1080/17588928.2015.1020053

Gers, F. A., Schraudolph, N., & Schmidhuber, J. (2002). Learning precise timing with LSTM recurrent networks. *Journal of Machine Learning Research*, *3*, 115-143.

Gumbsch, C., Otte, S., & Butz, M. V. (2017). A computational model for the dynamical learning of event taxonomies. In *Proceedings of the 39th annual meeting of the cognitive science society* (p. 452-457). Cognitive Science Society.

Hommel, B., Müsseler, J., Aschersleben, G., & Prinz, W. (2001). The theory of event coding (TEC): A framework for perception and action planning. *Behavioral and Brain Sciences*, *24*, 849-878.

Kingma, D. P., & Ba, J. L. (2014). Adam: A method for stochastic optimization. *ArXiv e-prints*, *abs/1412.6980*.

McClelland, J. L., Botvinick, M. M., Noelle, D. C., Plaut, D. C., Rogers, T. T., Seidenberg, M. S., & Smith, L. B. (2010). Letting structure emerge: connectionist and dynamical systems approaches to cognition. *Trends in Cognitive Sciences*, *14*(8), 348–356. doi: 10.1016/j.tics.2010.06.002

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., … Hassabis, D. (2015, February). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529–533. doi: 10.1038/nature14236

Otte, S., Schmitt, T., Friston, K., & Butz, M. V. (2017). Inferring adaptive goal-directed behavior within recurrent neural networks. *26th International Conference on Artificial Neural Networks (ICANN17)*, 227-235.

Otte, S., Zwiener, A., & Butz, M. V. (2017). Inherently constraint-aware control of many-joint robot arms with inverse recurrent models. *26th International Conference on Artificial Neural Networks (ICANN17)*, 262-270.

Schacter, D., Addis, D., Hassabis, D., Martin, V., Spreng, R., & Szpunar, K. (2012, November). The future of memory: Remembering, imagining, and the brain. *Neuron*, *76*(4), 677–694. doi: 10.1016/j.neuron.2012.11.001

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., … Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, *529*(7587), 484–489. doi: 10.1038/nature16961

Sugita, Y., Tani, J., & Butz, M. V. (2011). Simultaneously emerging braitenberg codes and compositionality. *Adaptive Behavior*, *19*, 295-316. doi: 10.1177/1059712311416871

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: An introduction*. Cambridge, MA: MIT Press.

Tani, J. (2017). *Exploring robotic minds*. Oxford, UK: Oxford University Press.

Wolpert, D. M., & Flanagan, J. R. (2016). Computations underlying sensorimotor learning. *Current Opinion in Neurobiology*, *37*, 7 - 11. (Neurobiology of cognitive behavior) doi: 10.1016/j.conb.2015.12.003

Wolpert, D. M., & Kawato, M. (1998). Multiple paired forward and inverse models for motor control. *Neural Networks*, *11*, 1317–1329. doi: 10.1016/S0893-6080(98)00066-5

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

Zacks, J. M., & Tversky, B. (2001). Event structure in perception and conception. *Psychological Bulletin*, *127*(1), 3–21. doi: 10.1037/0033-2909.127.1.3