Inferring Event-Predictive Goal-Directed Object Manipulations in REPRISE*

Martin V. Butz[0000−0002−8120−8537], Tobias Menge, Dania Humaidan, and Sebastian Otte[0000−0002−0305−0463]

Cognitive Modeling Group, Tübingen University, Tübingen 72076, Germany
martin.butz@uni-tuebingen.de

Abstract. The recently introduced REtrospective and PRospective Inference SchEme (REPRISE) infers contextual event states in the form of neural parametric biases retrospectively in recurrent neural networks (RNNs), distinguishing, for example, different sensorimotor control dynamics. Moreover, it actively infers motor commands prospectively in a goal-directed manner, minimizing anticipated future loss signals—such as the distance to a goal location. REPRISE struggles, however, when multiple, somewhat competing goals are active in parallel—such as when an object is to-be picked up and carried to a goal location. Moreover, unsuitable statistical correlations in the training data can prevent successful goal-directed motor inference, failing to reach particular goal constellations. We scrutinize this challenge and propose that appropriate gradient separation techniques are missing. First, we show that relative encodings and suitable training schedules can alleviate the problem. Most robust behavior, however, is achieved when the RNN architecture is suitably modularized. In the future, emergent RNN modularizations and more direct gradient separation mechanisms need to be developed.

Keywords: recurrent neural networks · active inference · parametric biases · sensorimotor codes · event-predictive cognition · neurocognitive modeling

1 Introduction

Over the last two decades, our mind has been portrayed as a predictive inference system, which plans and controls highly adaptive, goal-directed behavior by anticipating upcoming states and sensations [1,8,10,11,15]. Despite initial promising results with predictively encoded, generative artificial neural networks in computer vision [27], the development of generative recurrent artificial neural networks (RNNS) that are suitably structured for the invocation of dexterous, adaptive goal-directed control remains highly challenging. Recently, a promising retrospective and prospective active inference scheme (REPRISE) has been proposed, which learns sensorimotor forward models and exploits those models for the invocation of goal-directed behavioral control [7,6].

* Supported by Humboldt Foundation

Various strands of research suggest that our brain develops generative models of the encountered sensorimotor streams of information. These models appear to be loosely hierarchically structured, which is the case when moving from primary to deeper visual processing areas [18,27], revealing structures that can be related to Gestalt perceptions [17]. When focusing on generative models that predict information over time, however, evidence has corroborated suggesting that the involved temporally-predictive generative encodings are structured in an event-predictive manner [3,4,16,33]. *Events*, which can be described as predictive neural attractors that encode distinct sensorimotor dynamics, and *event boundaries*, which encode the conditions under which events typically commence, end, or transition into each other, are considered to constitute fundamental units of thought [26,28]. With respect to anticipatory, goal-directed behavior, such event structures offer themselves well for the development of conceptually abstracted, hierarchical structures, that can be suitably exploited for the generation and modeling of human-like model-based planning. Moreover, close relations can be drawn to hierarchical reinforcement learning [2,8,19]. REPRISE develops hidden stable structures, which can be closely related to event encodings, and which may be exploitable for the invocation of conceptual reasoning, planning, and deeper goal-directed behavioral control [6].

It has been shown that sensorimotor temporally predictive models learned by an RNN can be used to compute inverse, goal-directed motor commands by means of active inference [23], approximating model-predictive control [9]. The active inference process uses prospective, back-propagation through time (BPTT) to infer the motor commands believed to be necessary to reach a desired goal-state. This mechanism was not only used to control flying vehicles towards particular locations in space [23], but also to control a many-joint robot arm towards a particular end-point location and orientation, in which case the forward kinematics is predicted [24,21]. The REPRISE adaptation process [7,6] adds retrospective inference, fostering the emergent distinction of several robotic systems during learning, and thus enabling the control of several systems by one RNN architecture without the provision of information about which system is currently being controlled.

The RNN to be analyzed and improved in this paper uses the REPRISE adaptation process to achieve goal directed behavior for different systems. The scenario consists of several flying and gliding vehicles, which can be controlled by means of thrust-like motors. The general scenario was detailed and investigated elsewhere [6,24,21]. In addition, a transportable object is introduced, which can be attached to and thus transported by the controlled vehicle. Figure 1 shows typical scenes in our simulator. A video about the scenario in which REPRISE controls the vehicles can be found online.[1]

Here, we show that the original architecture struggles when multiple, competing objectives are activated. For example, when the goal is to transport an object through space, the architecture fails to reach and hold on to the object. To scrutinize this challenge, we first evaluate performance in the light of different

[1] https://www.youtube.com/watch?v=KDK94qOaaTE

goal objectives. Moreover, we investigate to which extent a velocity encoding of the sensorimotor dynamics—instead of a location-based encoding—can alleviate the problem. Despite some performance improvements, the main challenge remains and appears to be due to competing gradient signals, which lead to the inference of unsuitable motor control commands. We thus introduce a modularized RNN structure, which solves the gradient interference problem, confirming our interference hypothesis. In conclusion, we discuss alternatives to the introduced hard modularization. In particular, we believe that gradient separation techniques need to be developed to enable the concurrent pursuance of multiple objectives, possibly by a technique introduced concurrently at ICANN 2019 [22].

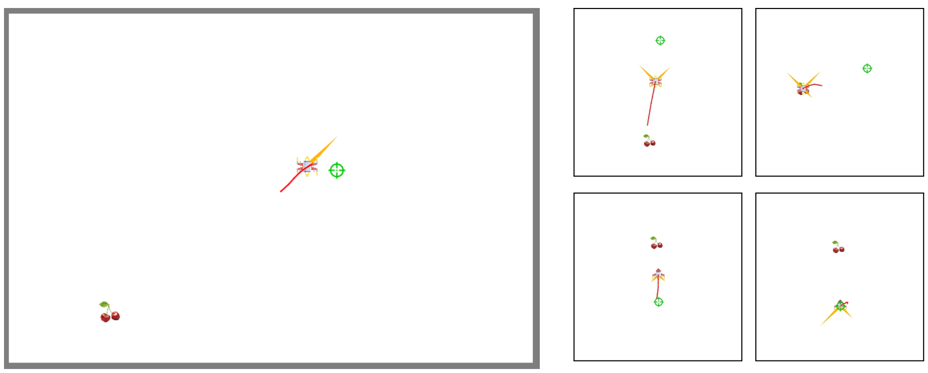

Fig. 1. Exemplar scene with whole training / testing area (an areas of 3×2 units, left image) and, furthermore, four zoomed-in exemplar scenes (right) with green circular goal locations and the red cherry target object. The vehicles are controlled by REPRISE. Left: REPRISE currently attempts to reach the cherry object controlling the "glider" vehicle, which undergoes simulated inertia and can be controlled by four thrust-like motors (yellow lines emanating from the vehicle). Right: REPRISE controlling the "glider" picking up and transporting the cherry (top); REPRISE controlling the "rocket"heading towards the goal location (bottom). The rocket undergoes inertia and gravity and can be controlled by two thrust motors pushing the vehicle obliquely upwards. The red line extending from the vehicles depicts the flight trajectory REPRISE currently anticipates.

2 REPRISE Model and Setup

The used RNN model mostly corresponds to the REPRISE architecture introduced elsewhere [5]. For simplicity purposes, we first introduce the mechanism that is applied to train the sensorimotor temporal forward models. Afterwards, we detail the mechanisms that unfold while actively inferring vehicle properties retrospectively and goal directed motor control commands prospectively.

2.1 Forward Model Process

The forward model assumes a discrete-time dynamical system, with the states being dependent on the time step t. The model furthermore assumes a *partially observable Markov decision process* (POMDP), such that the states have to be separated into perceivable states $s^t \in \mathbb{R}^n$ and hidden states $\sigma^t \in \mathbb{R}^m$. The system controls are represented by the component $x^t \in \mathbb{R}^k$. The next system state is determined by the previous state and the mapping Φ (1). Throughout this paper, we use an LSTM-like RNN architecture [14] for training the mapping Φ:

$$(s^t, \sigma^t, x^t) \xrightarrow{\Phi} (s^{t+1}, \sigma^{t+1}). \tag{1}$$

Note that in accordance to [5], the hidden state $\sigma^t \in \mathbb{R}^m$ is separated into more stable contextual state estimations $c^t \in \mathbb{R}^u$ and dynamic LSTM hidden cell states $\sigma_c^t \in \mathbb{R}^h$ with $h + u = m$ (we use $h = 16$, $u = 5$ in this work).

2.2 Inference of the Action Sequence

A trained forward model can be used to predict a series of T system-states $\{s^{t+1}, s^{t+2}, ..., s^{t+T}\}$ into the future (1), depending on the sequence of controls $\{x^{t+1}, x^{t+2}, ..., x^{t+T}\}$. Following [23], the inference of the control sequence needed to reach the desired system states $\{g^{t+1}, g^{t+2}, ..., g^{t+T}\}$ is achieved by using *back propagation through time* (BPTT) in combination with sensorimotor anticipations of control-dependent future system states. The control-sequence is thereby updated each time step t with the motor command-respective propagated errors $\frac{\partial \mathcal{L}}{\partial x_i^{t'}}$:

$$\frac{\partial \mathcal{L}}{\partial x_i^{t'}} = \sum_{h=1}^{H} \left[\frac{\partial net_h^{t'}}{\partial x_i^{t'}} \frac{\partial \mathcal{L}}{\partial net_h^{t'}} \right] = \sum_{h=1}^{H} w_{ih} \frac{\partial \mathcal{L}}{\partial net_h^{t'}}, \tag{2}$$

where the loss $\mathcal{L}$ is defined as the squared discrepancy between future desired system states and predicted states $\{s^{t+1}, s^{t+2}, ..., s^{t+T}\}$ (e.g., vehicle and goal locations) under a temporal future horizon T, that is, $\mathcal{L} = \frac{1}{2}\sum_{t'=t+1}^{T} \left(g^{t'} - s^{t'}\right)^2$, $net_h^{t'}$ denotes the value of the weighted summed activity reaching hidden neuron h at time t', and H the number of hidden neurons in the RNN. Each time step t, the process of prediction, inference, and update is iterated a small number of times to achieve a convergence towards the desired states.

2.3 System Architecture and Scenario

The RNN evaluated in this paper is trained to learn a forward mapping for several types of systems $\phi = \{\phi_1, \ldots, \phi_u\}$ (see Table 1). All systems are controlled by five motor commands, that is, $x^t \in \mathbb{R}^5$, which are composed of four thrust-like motor commands [0..1], which are causing forces in four oblique directions, and one signal to attach the target [0..1]. In the case of system ϕ_1, that is, the "rocket", two of the motor commands are ignored, while the other two induce oblique upwards impulses. All systems generate five state information signals, that is, $s^t \in \mathbb{R}^5$, which includes the current $(x, y) \in \Re^2$ position of the system, the current relative distance to an object in the scene $(d_x, d_y) \in \Re^2$, and a binary attachment signal $a \in \{0, 1\}$.

Following [5], REPRISE maintains a neural system context $c^t \in \mathbb{R}^u$ – elsewhere referred to as parametric bias neurons [29,30,32]. Previous work with parametric bias neurons has typically focused on imitation learning and particularly the progressively compact encoding of particular motion sequences, such as approaching an object from a certain direction or grasping an object [29,31,25], or moving the own body along trajectories generated by another robot [20]. In contract, REPRISE infers neural context activities such that the sensorimotor dynamics of the currently active system ϕ_i can be predicted accurately. The system's context c^t is initialized randomly and not preset but updated each time step during the training and inference procedure by using BPTT retrospectively over recent time steps with a retrospective temporal horizon R, similar to the motor control inference (cf. Section 2.5) but projecting and accumulating the gradient in the context neurons, enforcing a constant activity over R.

Table 1. Systems used for training and control

System	Type	Gravity	Inertia	Thrusts	Attach	Sensors
ϕ_1	Rocket	yes	yes	2	1	5
ϕ_2	Glider	no	yes	4	1	5
ϕ_3	Stepper	no	no	4	1	5

2.4 Training the Forward Model

With the perceivable state s^t, the context neurons $c^t \in \mathbb{R}^u$, and the motor control commands m^t as input, and the changes in s^t as target output, the architecture is trained sequentially with a learning rate of $1e-3$ and $1e-4$ with 150 blocks each. The blocks are composed of 2.001 time steps, after which the hidden states of the neurons are reset to zero activities. We trained thirty RNNs independently, where we initialized the weight values by means of randomly, normally-distributed values with a standard deviation of 0.1. Training behavior is similar to [6], ensuring that the motor repertoire and motions are explored in a sufficiently distributed manner. Additionally, a heuristic script was implemented to induce goal object reaching trajectories, such that the vehicle moves to the object in a typically suboptimal manner, often circling the object first, but eventually reaching it and thus attaching and transporting the object.

As a result, three training modes are used during a block to change the controls of the vehicle: free flight, seek, and transport. During the *free flight* training mode, the vehicle moves randomly through the environment. It thereby uses either random controls (50%), keeps the last ones (30%), or sets them to zero (20%). During the *seek* training mode, the controls are set to move the vehicle towards a target object. When the object's distance is below a certain threshold, the system attaches the object. During the *transport* training mode, the controls are set randomly in the same way as in the free flight mode while the object is kept attached. The mode switches either when an object is attached / detached, after 50 time steps, or by chance with a probability of 1%.

2.5 Evaluating the Active Control Inference

The trained RNN is evaluated during active, goal-directed inference. Throughout the evaluations, we successively probe the ability to reach a random goal location $(x_g, y_g, a = 0)$, to reach an object location and holding onto the object $(x_o, y_o, a = 1)$, and to transport an object to a goal location, that is, to reach a random goal location with the object being attached $(x_g = x_o, y_g = y_o, a = 1)$, in which case the object needs to be transported to the current goal location. Each goal constellation is activated for 150 time steps. Moreover, vehicles switch during training and goal-directed behavior every 150 time steps. Please note, though, that a robustness for random vehicle switches has been confirmed elsewhere, as long as the switches do not occur overly frequently [6]. After each goal location reaching trial, the goal location is reset to a random position. The performance is determined by calculating the mean and standard deviation over all RNNs and runs for each goal inference-type respectively.

3 Evaluations and System Modifications

With the task and system in hand, we now first evaluate the original system from [5]. We show that it is able to reach not only absolute goal locations as already shown in [5], but also relatively encoded object locations as long as no additional objectives are applied. However, we also show that the system struggles to issue and maintain the attachment command to transport the object and it cannot pursue two somewhat competing goals in parallel. We then evaluate the reason for this failure, exploring the effect of a velocity-based encoding, instead of a location-based one. The corroborated results, which improve but do not come near to optimality, suggests that the active inference mechanism suffers from a *gradient interference problem*, which applies when multiple somewhat disjunctive objectives are being pursued. Thus, we modularize the RNN architecture to prevent the gradient interference and in fact reach near optimal, goal-directed transport behavior.

3.1 Gradient Interference and Simple Alleviations

We first evalute the REPRISE's active inference performance with the training schedule detailed above. In the first experiment, the standard training schedule and the successive objectives are applied as detailed above. We report results of the consequent goal-directed behavioral control performance.

Figure 2 shows the obtained results. During the goal location approach, a distance of .042 units on average is maintained during the last 50 of the 150 time steps. Thus, the goal is reached and proximity is being maintained rather well. In contrast, the target object is reached with a mean of only 0.535 units during the last 50 time steps while attempting to reach it. As a partial consequence of this failure, also the transport behavior fails. Although the desired behavior and the error signal magnitude is similar for inferring the goal location and the target object location, the results show strongly differing performance values.

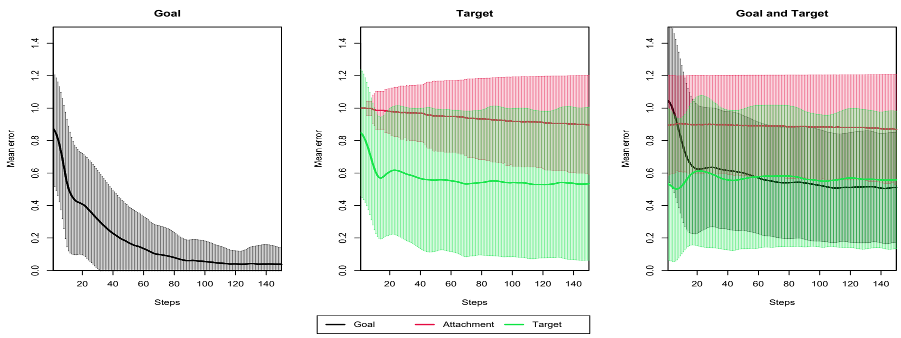

Fig. 2. The red colored graphs show the mean error of the attachment control over 150 time steps. The black colored graphs depict the Euclidean distance to the goal location, the green ones show the Euclidean distance to the target object, while red ones report the error in the currently encountered attachment signal. Each of the graphs additionally shows respective standard deviations.

Thus, the question is where this discrepancy comes from. One addition to the error signal in the object reaching task is that an additional error is induced, which signals that an object should be attached. Might it be due to the error signal from the attachment objective?

We thus re-evaluated the trained networks' active inference abilities without attachment objective during target object reaching trials. Figure 3 shows that the obtained results during target object reaching clearly improve (center figure). The goal location / target object are now reached with an average Euclidean distance of .080 / 0.145 units during the last 50 of 150 reaching time steps of each respective trial. The only variable in contrast to the default case above is the absence of the back-propagated error signal from the attachment objective (i.e. $a = 1$ versus no loss via a).

The results suggest that the attachment objective-based error signal (during prospective, active inference) might be back-projected onto particular relative target object distances, due to biased statistical correlations encountered during training. In fact, these statistics must be biased simply because an actual attachment can only occur when sufficiently close to the object, because attachments typically occur when flying slowly (due to the seek algorithm implementation), and attachment maintenance implies that the target object stays close to the vehicle. As a result, the skewed statistics between successful target attachments and flying behavior interrupt actual target approaching behavior, preventing actual successful target object attachments. This interpretation is further sup-

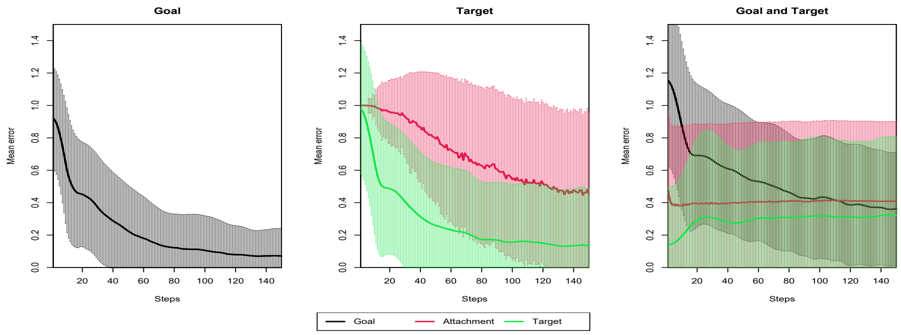

Fig. 3. Default mode without attachment control inference during *target inference*.

ported by results when changing the training schedule to *free flight* only: similar performance is reached in the target object reaching trials even when the attachment objective (i.e. $a = 1$) is applied (results not shown). This latter result is most likely the case because there is no statistical bias in the velocity and relative location encodings when correlated with the attachment signal.

However, the solution to the first and second objective should still be improved. Moreover, the third objective, that is, transporting the approached object to the goal location is still far from being solved. In fact, in typical trials when both objectives are set, but the vehicle starts not close enough to the target object, the vehicle usually moves to a point exactly between object and goal locations, indicating the dynamics of two competing gradients. Thus, these results and observations point to a *multiobjective interference problem*, partially combined with the problem of *biased training statistics*. In the next section, we further focus on these issues and consider additional solution options.

3.2 Prediction of Velocities

One statistical bias that must be reflected in the training statistics is the location of the vehicle: besides being often close to the transportable object, also the distribution of absolute locations will be non-uniformly distributed. Apart from the non-appealing option to artificially optimize motor training, we explored the option to encode velocities instead of locations as the sensory information, while still applying the same training schedule. The sensory state $s^t \in \mathbb{R}^5$ is changed from location and relative distance to the target object encodings, to velocity encodings of the vehicle and the target.

The inference modes for the velocity based RNN are the same as for the location based architecture above. However, there is the additional need to transform the desired goal reaching objectives into a desired velocity objective during active inference. This need can also be viewed as being advantageous, since it offers better control over the converging speed of the system towards the goal and target position. We apply a simple constant target velocity encoding of unit per second pointing towards the target. Please note that the stepper system cannot be used with the velocity based architecture since it has no inertia and therefore

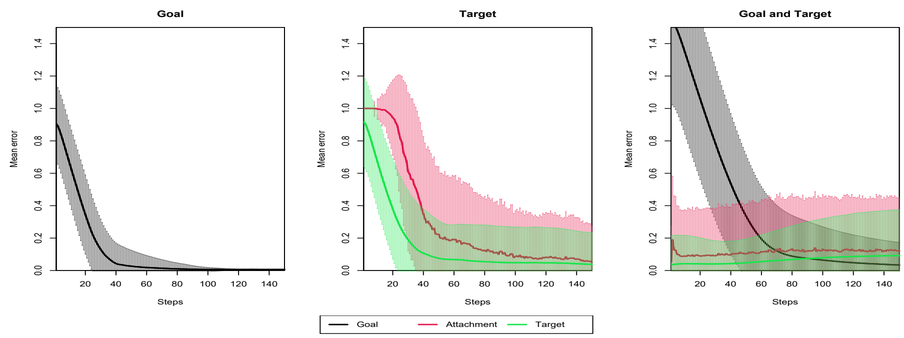

Fig. 4. Performance of a velocity based RNN with a constant velocity error signal towards the goal or target object.

no velocity. Thus, the reported results in the velocity encoding below are about glider and rocket only.

The results shown in Figure 4 imply a huge performance improvement in the goal location and target object reaching trials. Goal locations are now reached with a mean of 0.006 units during the last 50 time steps, while the target objects are reached with with a mean of 0.045 units, while the target is attached with a mean of 93% during the last 50 time steps. During the *transport to goal objective* , however, this value decreases to 88%, implying accidental disruptive detachment events. Interestingly, when changing the target velocity from the applied constant to a distance-linear or even -exponential mapping, the object and target location approach performance stays comparable, while the object attachment during transport strongly decreases to values below 20% (not shown).

These results show that with a velocity based forward model, the *goal location* and *target location* reaching objectives are achieved more accurately and reliably when compared to the location-based runs. However, attached targets are still rather frequently lost during the transport trials, that is, the *goal and target* inference trials – particularly when the velocity gradient signal is linearly or exponentially enhanced. This suggests that stronger velocity objectives can negatively affect attachment behavior. Once again, this is likely due to the training statistics, seeing that the linear or exponential error signal focuses back-propagation more strongly on velocity control than the constant setting, such that this error signal affects the attachment motor command inadequately.

3.3 Modularized Network

All results so far have pointed towards interacting, interdependent gradient information, which affects motor inference in unsuitable manners. To avoid the interaction of the object attachment with the object and goal location reaching objectives, we now modularize the RNN architecture. The main difference between the modularized architecture and the one used above is that the position/velocity of the system as well as the distance to the target object are predicted by one of the modularized sub-networks, while the attachment-status is predicted independently by the other modularized sub-network (see Figure 5). The active inference of the motor control commands is done in these two sub-networks, thereby separating the back-propagated errors of motor thrust commands from the attachment control command. The sub-networks have eight memory cells each, that is, combined the same number of LSTM-cells as the monolithic network. The inferred context, motor controls, and attachment control, and furthermore the whole system state is fed into both sub-networks, in the same way as was done in the default RNN.

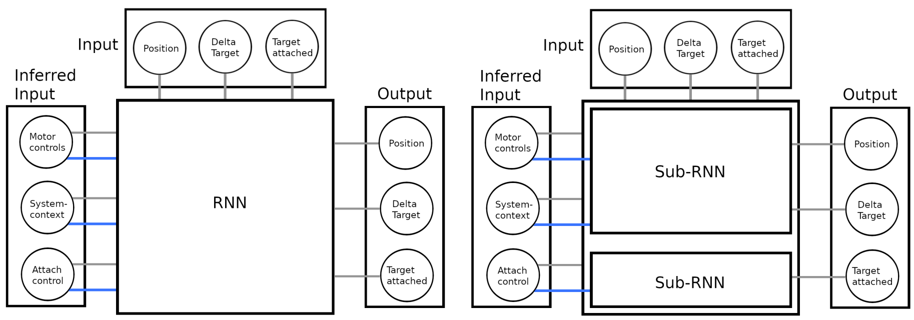

Fig. 5. Model of the default RNN (left) and the modularized RNN (right): Grey connections are used for forward-passes and blue connections for inference. The system-state input is connected to the whole RNN in both cases. For the modularized RNN, the output and the inferred input (motor controls, system state, and attachment control) are connected to either one (but not both) of the two sub-RNNs, effectively modularizing the back-propagated error signals.

In the first experiment, the modularized architecture is used to infer the goal-directed behavior for velocity based system states using the constant target velocity mapping used above. Figure 6 shows that the goal and target object locations are reached effectively with a mean of 0.018 and 0.028 units during the last 50 trials. Furthermore, the goal location is reached with a mean of 0.036 units during the transport trial with the target still being attached in 98% of the cases. Thus, the gradient interference problem is solved to a large extent.

Besides the velocity-based encoding, we were also interested in the performance gain from the modularization with the spatial encoding used above. The results in Figure 7 show that an error of 0.014 and 0.037 units is reached during the last 50 steps of each trial while pursuing a goal location and a target object location, respectively. During the transport trials, the goal location is reached with a mean of 0.086 units, the deterioration of which is mostly due to the approx. 6% of trials during which the object is accidentally detached. The performance with the location-based modularization of the RNN model is thus largely comparable to the velocity-based one, albeit more detachments occur. Note that the slower goal and target approach behavior is due to the trials with the stepper.

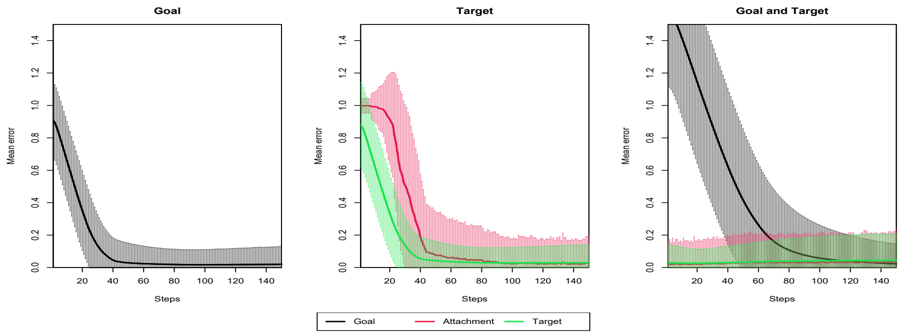

Fig. 6. Performance with velocity state signals in the modularized RNN architecture.

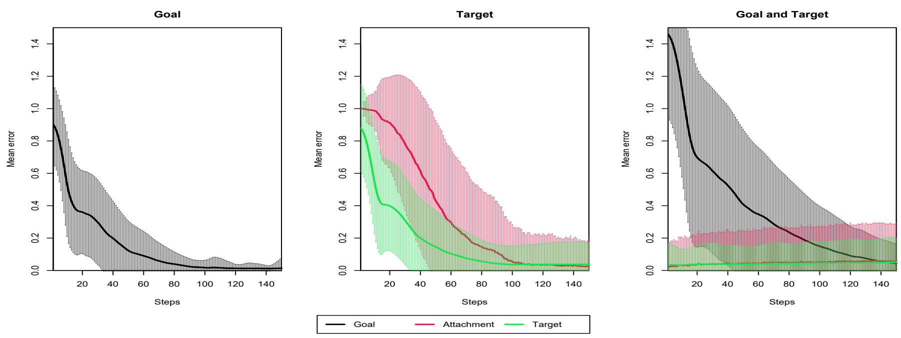

Fig. 7. Performance with location state signals in the modularized RNN architecture.

Overall, the modularized RNN yields improved performance particularly in reaching, attaching, and transporting the target object. Thus, it appears that gradient interference was prevented even more successfully due to the modularized architecture.

4 Conclusion

This paper has evaluated, analyzed, and modified the neuro-cognitive architecture REPRISE [6], focusing on its goal-directed active inference mechanism. Control performance decreases when attempting to pursue competing objectives concurrently. This competition in the back-propagated goal-error-based gradient may be either due to subtle and unexpected biases resulting from the sensorimotor statistics encountered during training or due to inherently competing gradients. The latter is, for example, the case when the objective is to transport the object to the goal, which is only possible once the object is held by the vehicle. When all three objectives, that is, vehicle at goal location, target object close to the vehicle, and target object attached to the vehicle are activated but not satisfied yet, the vehicle typically flies towards a position between goal location and target object location, hovering between the two locations. The observed successful performance in the modularized architecture was only possible because the system's objective was first to move to the target object and then to transport it to the goal location. We thus essentially informed the system about the hierarchical nature of the problem, inducing hierarchical planning somewhat artificially.

These results and interpretations imply two fundamental and critical future work directions, addressing (i) the identified gradient interference problem and (ii) the necessary hierarchical planning mechanism. The gradient interference problem was solved by modularizing the RNN architecture, while the hierarchical planning challenge was solved by inducing successive goals in an appropriate sequence. Both of these solutions essentially induce hand-crafted learning and inference biases. The successful solution confirms that we identified the problem correctly. However, the hand-craftedness is obviously not satisfactory or even applicable in other scenarios and system architectures. Clearly, a suitable network modularization is not always possible and often not known in advance. We believe that the further exploration of emergent modularizations based on predictability signals and the inference of not only temporal but causal correlations between sensory and motor structures will be essential. Moreover, such modularizations may be fostered further by contextual neurons and their selective activation dependent on the goal-respective event context. REPRISE currently only utilizes contextual neurons to modify the active sensorimotor forward model. It may well be the case that enhancements of this approach, with a further focus on respective goals, will enable the necessary modularization. Thereby, surprise signals may need to be processed in a dedicated manner as suggested in [3] and successfully done in more explicit event-predictive architectures in [13] and [12]. Complementary to this, however, also modifications of back-propagation that directly prevent gradient interferences may be possible. A first approach in this direction that separates error gradient-based backpropagation over a modularized RNN architecture was able to extract the individual sine waves from time series data that consists of multiple superimposed sine-waves without informing the system about the number of contained waves [22]. We believe that the suggested modularization mechanisms and gradient separation mechanisms may be well-suited to identify individual events and possible event transitions, while simultaneously forming conceptual, event-specific neural modules.

In [13,12] this was done for simple object transportation tasks and for complex robot behavioral control tasks. Our current goal is to develop such modularizations without clear-cut separations of event- and event-boundary-predictive encodings in suitably structured RNN architectures. Once we have accomplished this, we expect that the system will be able to naturally plan on deeper, conceptual levels, which would enable it to plan and execute sequential object manipulations and to use tools in highly versatile, goal-directed manners.

References

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

2. Botvinick, M., Weinstein, A.: Model-based hierarchical reinforcement learning and human action control. Philosophical Transactions of the Royal Society of London B: Biological Sciences **369**(1655) (2014). https://doi.org/10.1098/rstb.2013.0480
3. Butz, M.V.: Towards a unified sub-symbolic computational theory of cognition. Frontiers in Psychology **7**(925) (2016). https://doi.org/10.3389/fpsyg.2016.00925
4. Butz, M.V.: Which structures are out there? learning predictive compositional concepts based on social sensorimotor explorations. MIND Group, Frankfurt am Main (2017). https://doi.org/10.15502/9783958573093, http://predictive-mind.net/papers/which-structures-are-out-there
5. Butz, M.V., Bilkey, D., Humaidan, D., Knott, A., Otte, S.: Learning, Planning, and Control in a Monolithic Neural Event Inference Architecture. ArXiv e-prints arXiv:1809.07412 (Sep 2018)
6. Butz, M.V., Bilkey, D., Humaidan, D., Knott, A., Otte, S.: Learning, planning, and control in a monolithic neural event inference architecture. Neural Networks (2019). https://doi.org/https://doi.org/10.1016/j.neunet.2019.05.001, http://www.sciencedirect.com/science/article/pii/S0893608019301339
7. Butz, M.V., Bilkey, D., Knott, A., Otte, S.: Reprise: A retrospective and prospective inference scheme. Proceedings of the 40th Annual Meeting of the Cognitive Science Society pp. 1427–1432 (2018)
8. Butz, M.V., Kutter, E.F.: How the Mind Comes Into Being: Introducing Cognitive Science from a Functional and Computational Perspective. Oxford University Press, Oxford, UK (2017)
9. Camacho, E.F., Bordons, C.: Model Predictive Control. Springer-Verlag (1999)
10. Clark, A.: Surfing Uncertainty: Prediction, action and the embodied mind. Oxford University Press, Oxford, UK (2016)
11. Friston, K.: The free-energy principle: a rough guide to the brain? Trends in Cognitive Sciences **13**(7), 293 – 301 (2009). https://doi.org/10.1016/j.tics.2009.04.005
12. Gumbsch, C., Butz, M.V., Martius, G.: Autonomous identification and goal-directed invocation of event-predictive behavioral primitives (2019)
13. Gumbsch, C., Otte, S., Butz, M.V.: A computational model for the dynamical learning of event taxonomies. In: Proceedings of the 39th Annual Meeting of the Cognitive Science Society. pp. 452–457. Cognitive Science Society (2017)
14. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation **9**, 1735–1780 (1997)
15. Hohwy, J.: The Predictive Mind. Oxford University Press, Oxford, UK (2013)
16. Hommel, B., Müsseler, J., Aschersleben, G., Prinz, W.: The theory of event coding (TEC): A framework for perception and action planning. Behavioral and Brain Sciences **24**, 849–878 (2001)
17. Jäkel, F., Singh, M., Wichmann, F.A., Herzog, M.H.: An overview of quantitative approaches in gestalt perception. Quantitative Approaches in Gestalt Perception **126**, 3–8 (Sep 2016). https://doi.org/10.1016/j.visres.2016.06.004
18. Khaligh-Razavi, S.M., Kriegeskorte, N.: Deep supervised, but not unsupervised, models may explain it cortical representation. PLoS Comput Biol **10**(11), e1003915– (Nov 2014). https://doi.org/10.1371/journal.pcbi.1003915, http://dx.doi.org/10.1371%2Fjournal.pcbi.1003915
19. McClelland, J.L., Botvinick, M.M., Noelle, D.C., Plaut, D.C., Rogers, T.T., Seidenberg, M.S., Smith, L.B.: Letting structure emerge: connectionist and dynamical systems approaches to cognition. Trends in Cognitive Sciences **14**(8), 348–356 (2010). https://doi.org/10.1016/j.tics.2010.06.002

20. Murata, S., Yamashita, Y., Arie, H., Ogata, T., Sugano, S., Tani, J.: Learning to perceive the world as probabilistic or deterministic via interaction with others: A neuro-robotics experiment. IEEE Transactions on Neural Networks and Learning Systems **28**(4), 830–848 (April 2017). https://doi.org/10.1109/TNNLS.2015.2492140
21. Otte, S., Hofmaier, L., Butz, M.V.: Integrative collision avoidance within rnn-driven many-joint robot arms. Artificial Neural Networks and Machine Learning – ICANN 2018 (11141), 748–758 (Oct 2018)
22. Otte, S., Rubisch, P., Butz, M.V.: Gradient-based learning of compositional dynamics with modular rnns. Artificial Neural Networks and Machine Learning – ICANN 2019 (2019), accepted for publication
23. Otte, S., Schmitt, T., Friston, K., Butz, M.V.: Inferring adaptive goal-directed behavior within recurrent neural networks. 26th International Conference on Artificial Neural Networks (ICANN17) pp. 227–235 (2017)
24. Otte, S., Zwiener, A., Butz, M.V.: Inherently constraint-aware control of many-joint robot arms with inverse recurrent models. 26th International Conference on Artificial Neural Networks (ICANN17) pp. 262–270 (2017)
25. Park, J., Kim, D., Nagai, Y.: Learning for goal-directed actions using rnnpb: Developmental change of “what to imitate”. IEEE Transactions on Cognitive and Developmental Systems **10**(3), 545–556 (Sep 2018). https://doi.org/10.1109/TCDS.2017.2679765
26. Radvansky, G.A., Zacks, J.M.: Event cognition. Oxford University Press, Oxford, UK (2014)
27. Rao, R.P., Ballard, D.H.: Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. Nature Neuroscience **2**(1), 79–87 (January 1999). https://doi.org/10.1038/4580
28. Richmond, L.L., Zacks, J.M.: Constructing experience: Event models from perception to action. Trends in Cognitive Sciences **21**(12), 962–980 (2017). https://doi.org/10.1016/j.tics.2017.08.005
29. Sugita, Y., Tani, J., Butz, M.V.: Simultaneously emerging braitenberg codes and compositionality. Adaptive Behavior **19**, 295–316 (2011). https://doi.org/10.1177/1059712311416871
30. Tani, J.: Model-based learning for mobile robot navigation from the dynamical systems perspective. IEEE Transactions. System, Man and Cybernetics (Part B), Special Issue on Learning Autonomous Systems **26**(3), 421–436 (1996)
31. Tani, J.: Learning to generate articulated behavior through the bottom-up and the top-down interaction processes. Neural Networks **16**(1), 11 – 23 (2003). https://doi.org/https://doi.org/10.1016/S0893-6080(02)00214-9
32. Tani, J.: Exploring Robotic Minds. Oxford University Press, Oxford, UK (2017)
33. Zacks, J.M., Speer, N.K., Swallow, K.M., Braver, T.S., Reynolds, J.R.: Event perception: A mind-brain perspective. Psychological Bulletin **133**(2), 273–293 (2007). https://doi.org/10.1037/0033-2909.133.2.273