A Neural Dynamic Model Parses Object-Oriented Actions

Mathis Richter (mathis.richter@ini.rub.de)
Jonas Lins (jonas.lins@ini.rub.de)
Gregor Schöner (gregor.schoener@ini.rub.de)

Institut für Neuroinformatik, Ruhr-Universität Bochum, 44870 Bochum, Germany

Abstract

Parsing actions entails that relations between objects are discovered. A pervasively neural account of this process requires that fundamental problems are solved: the neural pointer problem, the binding problem, and the problem of generating discrete processing steps from time-continuous neural processes. We present a prototypical solution to these problems in a neural dynamic model that comprises dynamic neural fields holding representations close to sensorimotor surfaces as well as dynamic nodes holding discrete, language-like representations. Making the connection between these two types of representations enables the model to parse actions as well as ground movement phrases—all based on real visual input. We demonstrate how the dynamic neural processes autonomously generate the processing steps required to parse or ground object-oriented action.

Keywords: relations; neural process model; action parsing; dynamic field theory; grounded cognition; cognitive schemas

Introduction

If you were to describe the arrangement of furniture in your office you would probably make use of the spatial relations between different items. You may recognize without effort that "the bookshelf is to the left of the desk" although this relationship is not directly specified by perception and requires active construal. In fact, relational processing may be foundational to higher cognition (Halford, Wilson, & Phillips, 2010).

Evaluating even simple relations requires several coordinated steps (Logan & Sadler, 1996): (1) binding each object to a role (here, the desk is the *reference object*, the bookshelf is the *target object*); (2) centering the reference frame on the reference object; (3) applying a relational operator (here, "to the left of") to the target object in that frame. We have previously realized these computational steps in a neural process model (Lipinski, Schneegans, Sandamirskaya, Spencer, & Schöner, 2012; Richter, Lins, Schneegans, Sandamirskaya, & Schöner, 2014) that mapped spatial relational phrases supplied in a language-like, discrete form, to a visual scene representation, effectively *grounding* amodal input in perception. Conversely, the model was able to complete partial phrases by referring to the visual scene.

Here, we extend this model to the parsing of object-oriented actions and the grounding of action phrases, such as "the red ball is moving toward (or away from) the yellow ball". This is challenging for a neural process model as the perceptual representations are inherently transient. We add two components that extract motion directions through a neural dynamic model of motion detection (Berger, Faubel, Norman, Hock, & Schöner, 2012) and transform the visual scene to an intrinsic reference frame (van Hengel, Sandamirskaya, Schneegans, & Schöner, 2012) based on the motion direction of a reference object. The new model integrates this form of action parsing with the grounding of spatial relationships of the earlier model.

We use dynamic field theory (DFT; Schöner, Spencer, & the DFT Research Group, 2015) as a theoretical framework. DFT describes neural population activity by activation fields that are defined over metric feature dimensions and evolve continuously in time through a neural dynamics. While the fields capture representations in a modal form close to the sensorimotor surfaces, neural nodes sharing the same dynamics enable modeling discrete, amodal representations. Mutual coupling between fields and nodes allows for interaction between these two kinds of representations. By using only tools from the DFT repertoire we arrive at a seamless process account that is pervasively neural. This requires that we solve the following fundamental problems that arise from restrictions of neural mechanism and are not commonly addressed in accounts of higher cognition.

First, information represented by neural activity cannot be freely moved within and between neural populations, because neural connectivity is fixed. In visual cortex, for instance, visual objects are represented in neural maps. Applying a neural operator to a location or an object in such a map is possible only if it is connected to that location. Connecting operators to every location in a map would require unrealistic neural resources. The alternative is to connect the operator to only one default region, a virtual fovea, and shift the representations of objects to that region. This is analogous to the concept of an attentional neural pointer of Ballard, Hayhoe, Pook, and Rao (1997) and is achieved in our framework by steerable neural mappings (Schneegans & Schöner, 2012).

Second, for similar reasons of limiting the required neural resources, the nervous system represents high-dimensional visual information in multiple low-dimensional neural feature maps, in particular in the early tiers of the cortical hierarchy. To refer to any particular object, corresponding representational pieces must be bound together. In a neural implementation of the classical idea of binding through space (Treisman & Gelade, 1980), we endow every feature map with a spatial dimension shared across maps and process objects sequentially in time (Schneegans, Spencer, & Schöner, 2015).

Third, the discrete processing steps this implies and that are critical to all of higher cognition are natural in information processing accounts but hard to achieve in neural process models, in which neural activation evolves continuously in time under the influence of input and recurrent connectivity. In our model, discrete events emerge from continuous neural

dynamics through dynamic instabilities, at which the match between neural representations of *intentional states* and their *conditions of satisfaction* are detected (Sandamirskaya & Schöner, 2010).

Finally, the problem of preserving role-filler binding (Doumas & Hummel, 2012) at the interface between the modal and the amodal representations is also solved by sequential processing.

Methods

Dynamic field theory describes processes that characterize neural activity at the population level. Models in DFT are based on activation patterns defined as dynamic fields, $u(x,t)$, over continuous feature dimensions, x, (e.g., color or space). These activation patterns evolve in time, t, under the influence of lateral interactions and external input based on the following integro-differential equation

$$\tau\dot{u}(x,t) = -u(x,t) + h + s(x,t) + \int g(u(x',t))\, w(x-x')\, dx'.$$

Here, the activation's rate of change, $\dot{u}(x,t)$, depends on $u(x,t)$ itself, on a time constant, τ, a negative resting level, h, and external input, $s(x,t)$, from sensors or other fields. Lateral interaction is determined by convolving the output of the field, $g(u(x,t))$, a sigmoid function with threshold at zero, with an interaction kernel, $w(\Delta x)$. The kernel combines local excitation and surround inhibition along the field's feature dimension.

When presented with localized input above the output threshold, lateral interaction leads to an instability, in which a subthreshold solution becomes unstable and the field moves to a new attractor, a self-stabilized activation peak. From such instabilities, neural events emerge at discrete times from the time-continuous dynamics of the fields. These events are critical for organizing sequential processes in DFT models.

Depending on the tuning of their interaction kernel, dynamic fields may either support multiple peaks or may be selective and only create a single peak that suppresses all others. Fields may also be tuned to hold self-sustained peaks that remain even after input is removed. Fields can be defined over single or multiple dimensions. Dynamic nodes share the fields' dynamic characteristics but do not span a feature dimension. Instead, they represent the 'on' or 'off' state of discrete elements within an architecture.

DFT architectures consist of multiple fields and nodes that are interconnected, where the output of one field is input to another field. Fields of different dimensionalities may be connected along the shared feature dimensions.

Architecture

The DFT architecture shown in Fig. 1 can deal with two types of tasks. First, it can ground a language-like phrase such as "the red object moving toward the yellow object", that is, it can find the objects in the scene that correspond to the phrase. Second, it can generate a phrase such as the one above from observing a video. Solving these tasks within a single neural architecture requires integrating various components, which we describe in more detail now.

Perception

The architecture receives video input from a camera or video file. This input is fed into two three-dimensional perception fields (top right of Fig. 1) that hold a representation of the scene. Both fields share the spatial dimensions of the camera image but the *perception color field* represents the color of objects in the scene and the *perception movement field*, new over the previous model (Richter et al., 2014), represents their movement direction. To create the input to the perception fields, each video frame goes through several preprocessing steps. For the color field, the preprocessing is first based on generic image processing algorithms. After these, activation is generated that scales with the color saturation of objects in the scene. For the movement field, the preprocessing consists of a neural dynamic implementation of the counter-change model of motion perception (Berger et al., 2012). Both perception fields always have stable peaks of activation when there are colored or moving objects in the scene. They project activation into the spatial attention fields along the two spatial dimensions and act as a saliency mechanism. They also project directly into the reference and target field and enable these fields to track moving objects even if spatial attention is currently focused elsewhere.

Attention

The core of the attentional system consists of two three-dimensional *attention fields*. They are defined over the same dimensions as the two perception fields but their activation remains below threshold unless additional input arrives from a feature attention field or a spatial attention field.

A pair of one-dimensional fields spans each feature dimension (color and movement direction): the *intention field* represents feature values for guided search and impacts on the three-dimensional attention fields; the *condition of satisfaction (CoS) field* matches input from the attention fields against what is represented in the intention field.

Two *spatial attention fields* are defined over the two spatial dimensions of the camera image. One field allows for multiple simultaneous peaks and projects into the reference and target fields. The other only allows for a single peak; it can be boosted to induce a selection decision on multiple candidate objects. A peak generated in this spatial attention field suppresses activation at all other locations in the other spatial attention field. It further projects into the three-dimensional attention fields, enabling peaks to form there that represent the feature values at the selected location (which then impact on the CoS fields). This implements a neural mechanism of feature binding across space (Schneegans et al., 2015).

Coordinate transformation

The two-dimensional *reference field* and *target field* each represent the spatial position of their respective objects. The tar-

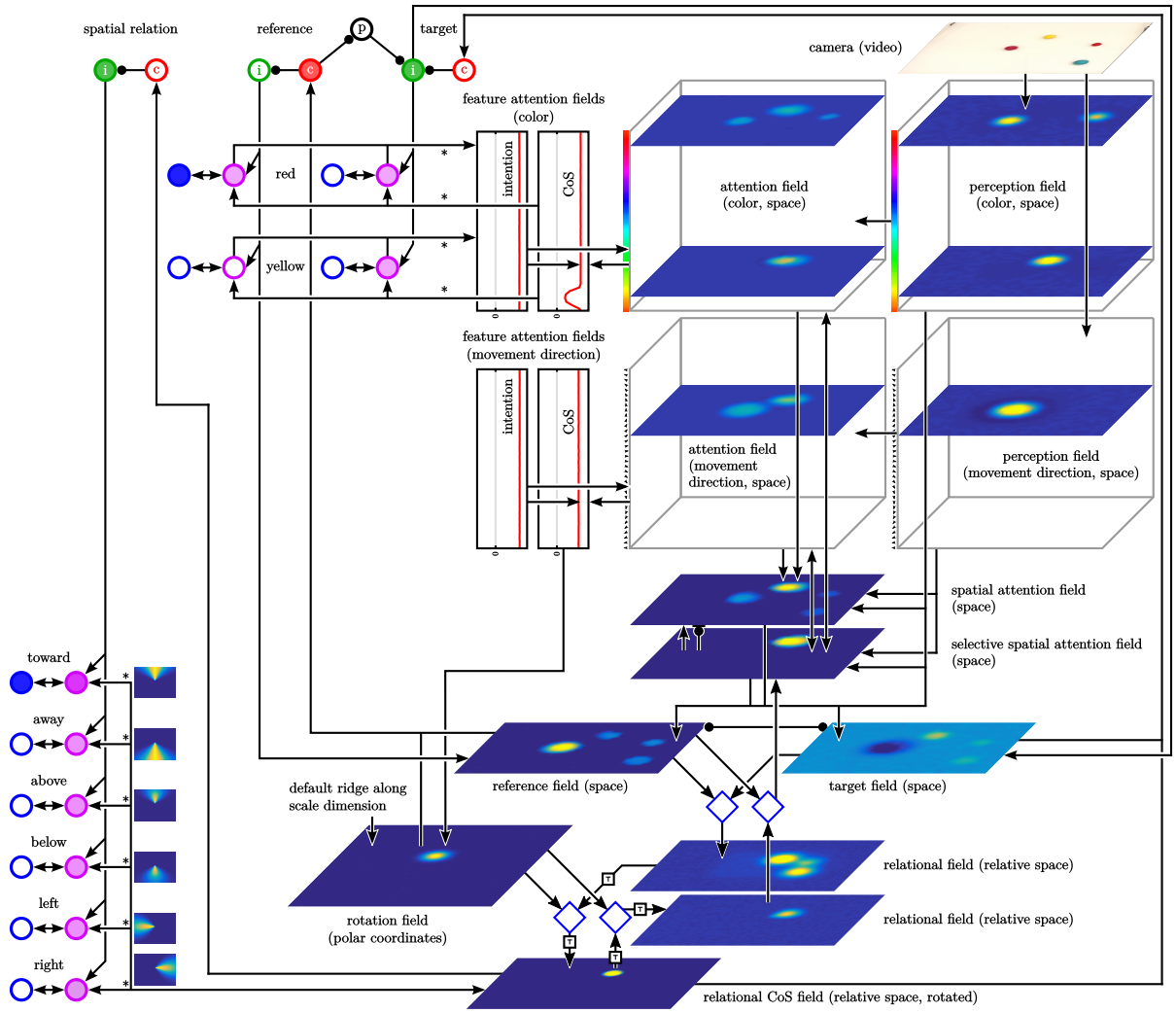

Fig. 1: Architecture with activation snapshots while it is generating a phrase about a video. Fields are shown as color-coded activation patterns; for three-dimensional fields, two-dimensional slices are shown. Node activation is denoted in opacity-coded circles. Spatial templates are illustrated as color-coded weight patterns (bottom left). Excitatory synaptic connections are denoted by lines with arrow heads, inhibitory connections by lines ending in circles. Transformations to and from polar coordinates are marked with a 'T'. Coordinate transformations are denoted as diamonds.

get field projects into the *relational field* via a steerable neural mapping (upper left blue diamond in Fig. 1) that shifts the representation of the target objects so that it is centered on the reference object. This transformation to a new reference frame is implemented as a convolution for performance reasons.

The shifted representation of the target objects is then rotated around the reference object. This transforms the target representation into an intrinsic reference frame defined by the reference object's movement direction. This rotatory transformation, new over the previous model, is realized by a steerable neural mapping that shifts activation patterns along the azimuth of the polar coordinate representation of the relational field (lower left blue diamond in Fig. 1). The extent of the shift is determined by the movement direction of the reference object, which is held by the *rotation field*.

The rotated target representation is projected into the *relational CoS field*. A second input to this field from spatial concept nodes encodes the associated spatial templates through weight patterns (illustrated in the lower left of Fig. 1). Overlap of the two inputs leads to a peak that represents the selected target. The steerable neural maps thus make it possible to apply the relational operator encoded in the fixed weight patterns to objects at any visual location in any orientation, implementing neural pointers.

The relational CoS field projects into the selective spatial attention field via reverse transformations for rotation and shift (upper and lower right diamonds in Fig. 1). Selective spatial attention projects into the three-dimensional attentional fields, forming peaks there that in turn project to the

feature fields, which may activate production nodes.

Concepts

Concepts like 'red' or 'toward' are represented by discrete nodes (denoted by circles in Fig. 1) that project with patterned synaptic weights into their respective feature fields. The nodes come in pairs: *memory nodes* (blue circles) act as an interface to a user who may activate them as input or observe them as output; *production nodes* (pink circles) gate the impact of their respective memory nodes onto the architecture. Note that there are copies of such pairs of nodes for each role that a concept may appear in (e.g., two pairs for 'red', as reference and as target), enabling role-filler binding. The synaptic weight patterns between nodes and fields could be learned by Hebbian learning rules but are hand-tuned here.

Process organization

The processes within the architecture are organized by instabilities of neural nodes that switch components 'on' or 'off'. These discrete events thus emerge from the time-continuous neural dynamics. Process organization is based on a structural principle borrowed from behavioral organization (Richter, Sandamirskaya, & Schöner, 2012). The core structure is the *elementary behavior*, which consists of two dynamic substrates. The *intention node* (green circle in Fig. 1) determines whether a process is active and has impact on connected structures. The *condition of satisfaction node* (CoS, red circle) is activated once a process has terminated and inhibits the intention node, turning the process off. Here, we employ elementary behaviors that control the grounding of the reference object (reference behavior), the target object (target behavior), and the spatial relation term (spatial relation behavior) (top left in Fig. 1). Role-filler binding is preserved during grounding by processing reference and target objects sequentially, organized by the *precondition* node (black circle) that inhibits the intention node of the target behavior until the reference behavior has terminated.

Results

In the following, we describe the dynamic processes that unfold within the architecture as it executes tasks. The results come from numerical solutions of the architecture's differential equations.[1] To simplify visual object recognition, we use a scene with uniformly colored objects on a white background.

Parsing an action

Fig. 2 illustrates the processes within the architecture as it generates a phrase about a video in which a red ball rolls toward a yellow ball (see top right of Fig. 1).

At $t = 0$ we give a boost into the architecture, which impacts the intention nodes of all behaviors. After this boost, the architecture runs autonomously in continuous time, without any further intervention from user or program. First, the reference object is grounded; the target behavior is inhibited by the precondition constraint until the reference behavior is finished. Without information about which objects to describe, the architecture decides based on their saliency. At t_1, the selective spatial attention field shows a saliency advantage for the moving red object in the lower left corner.

At t_2, the spatial attention field has made a selection decision and formed a peak. This creates a self-sustained peak in the reference field, selecting the moving object as reference. It also activates the production node 'reference: red' (top of Fig. 2) by projecting activation into the color CoS field via the attention color-space field (both not shown in Fig. 2; see Fig. 1). At the same time, the rotation angle field (not shown in Fig. 2) forms a representation of the object's movement direction, which it receives from the attentional movement-space field. It will later be used as a parameter to rotate the target objects. At this point, the architecture has grounded the reference object. That is, it has formed a connection between the continuous representations in the fields and the discrete representations in the nodes.

At t_3, the behavior to ground the reference object has been inhibited by its CoS node and the behavior to ground the target object has become active. However, even though the reference behavior is inactive, the peak in the reference field is still tracking the position of the moving object, because it receives input from the perception fields. Contrary to the reference behavior, the selective spatial attention field is not boosted during the target behavior, allowing multiple target candidates to be projected to downstream fields. The target field has formed three peaks at the positions of the remaining objects. The field's output is transformed and projected into the relational field, where the target positions are now represented relative to that of the reference object. This representation is rotated around the reference object and projected into the relational CoS field.

At t_4, the relational CoS field has formed a peak at the target position that overlaps most with the spatial template for the relation 'toward'. This activates the corresponding production node 'spatial: toward'.

At t_5, the activation from the relational CoS field is transformed and projected back into the selective spatial attention field, from there into the attentional color-space field, and from there into the target field as well as the color CoS field. The peak in the color CoS field activates the production node 'target: yellow'.

At this point, the architecture has produced the relational phrase 'red toward yellow' and has created a grounding of this phrase in sensorimotor representations.

Grounding a phrase

The architecture can also ground a phrase that it is given by user input. Due to space constraints, we cannot describe the process at the same level of detail. The process is very similar to that of grounding spatial relations reported earlier (Richter et al., 2014). The user supplies the phrase by activating mem-

[1] The architecture is implemented and simulated using the C++ framework *cedar* (Lomp, Zibner, Richter, Rano, & Schöner, 2013).

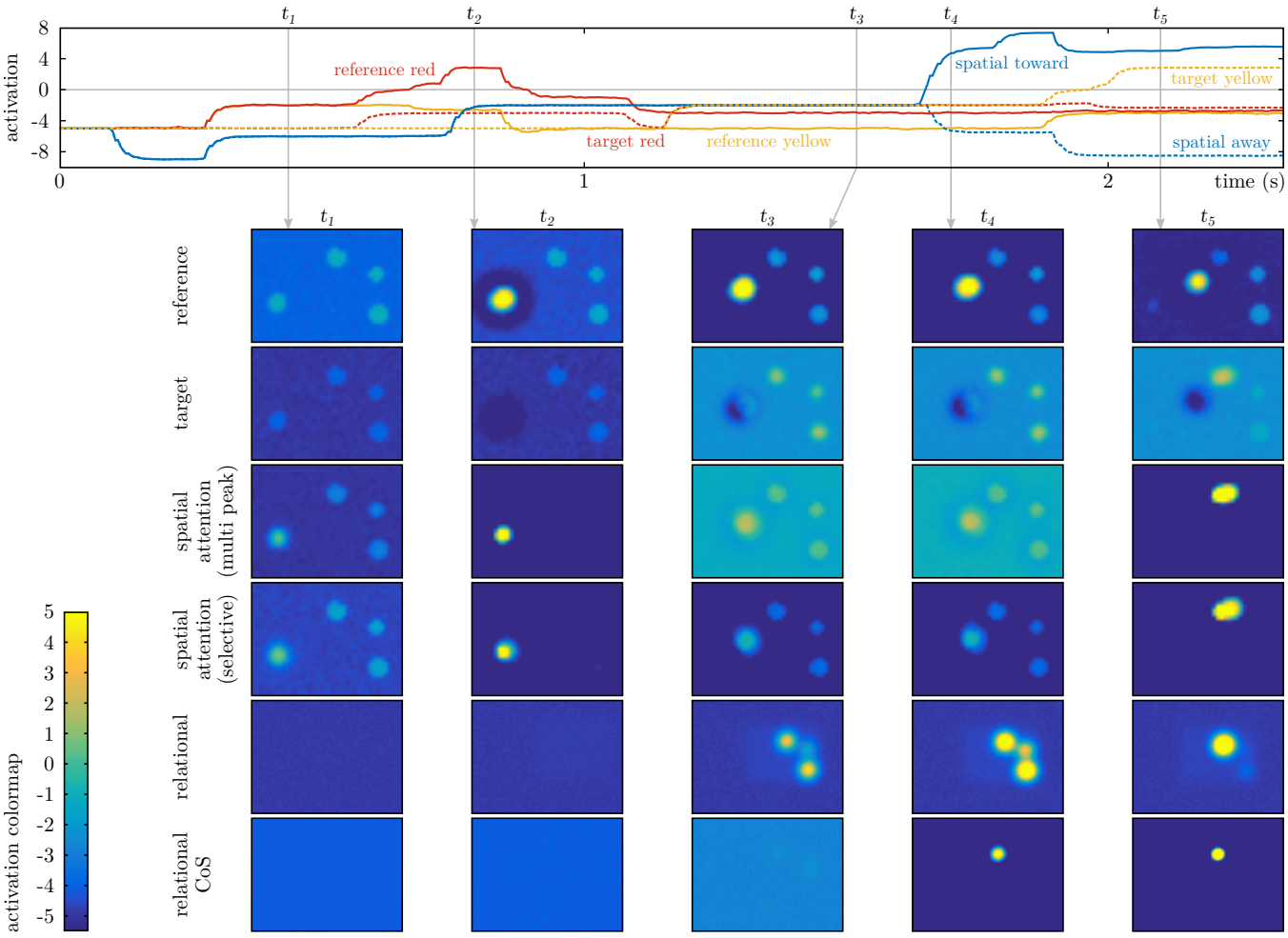

Fig. 2: Activation time courses of relevant production nodes (top) and activation snapshots of relevant fields at five points in time (bottom). Fields are color-coded using the color map on the bottom left.

ory nodes through manual boosts. Visual search for objects is then guided, as opposed to bottom-up saliency-driven. For instance, to ground the reference object, its red color is represented in the color intention field, bringing up peaks of red objects in the attentional color-space field—analogously with yellow objects for the target. Similarly, the template for spatial relations preshapes the relational CoS field and only allows peaks that overlap with the template. The grounding is established once a representation in the fields has been established for each element of the supplied phrase.

Discussion

We have extended a neural process model of spatial relations to include parsing of object-oriented actions and grounding of movement phrases. In the model, space-time continuous activation patterns are both coupled to sensory input and linked to neural representations of cognitive schemas like *move toward* or *move away from*. This provides a neural processing account of the interaction between sensorimotor activation, conceptual processing, and language, that theories of perceptual symbols (Barsalou, 2008) and embodied construction-grammar (Bergen & Chang, 2013) postulate. The integrative nature of the model leads us to confront fundamental issues such as the neural pointer problem, the binding problem, and how discrete processing steps emerge from time-continuous neural dynamics. Our solutions derive from the conceptual commitments of the theoretical framework of dynamic field theory.

We build on existing modeling approaches to the grounding of language that are neurally inspired but do not typically adhere to neural principles as consistently. For instance, the Neural Theory of Language (Feldman, 2006) is a hybrid framework that combines neural network concepts with ideas that are not compatible with neural process thinking. Similarly, Madden, Hoen, and Dominey's (2010) model for embodied language complements neural networks with algorithms that are not neurally based. Some models invoke neural concepts to account for psychophysical data. For instance, Regier and Carlson (2001) use the notion of an attentional vector sum to capture spatial terms. Such models are

not typically embedded into architectures that autonomously generate the complete sequence of processing steps required to ground and generate language.

The ambition of a neural process account for higher cognition is shared with the group of Eliasmith (2013). Their Neural Engineering Framework (NEF) enables spiking neural networks to realize vector symbolic architectures (Gayler, 2004). In this substantially different approach, concepts and objects are represented by high-dimensional vectors through an encoding and decoding stage and transient neural patterns computed by superposition and projection. DFT, in contrast, is based on self-stabilized activation patterns defined over a few feature dimensions. Whether DFT and NEF can span the same range of cognitive phenomena—which approach is more consistent with neural reality is open for now.

The current model is open to extension in various directions, such as incorporating learning, scaling the number of concepts, and building more complex sequences of processing steps. Higher-order schemata (e.g., source-path-goal; Lakoff & Johnson, 1999) may be realized similar to what we demonstrated here. Exploiting the working memory implicit in our representations may enable us to link to relational mental models (Knauff, 2013).

References

Ballard, D. H., Hayhoe, M. M., Pook, P. K., & Rao, R. P. (1997). Deictic codes for the embodiment of cognition. *Behav Brain Sci*, *20*(4), 723–42; discussion 743–67.

Barsalou, L. W. (2008). Grounded cognition. *Annu Rev Psychol*, *59*, 617–45.

Bergen, B., & Chang, N. (2013). Embodied construction grammar. In T. Hoffmann & G. Trousdale (Eds.), *Oxford Handbook of Construction Grammar.* Oxford University Press.

Berger, M., Faubel, C., Norman, J., Hock, H. S., & Schöner, G. (2012). The counter-change model of motion perception: an account based on dynamic field theory. In *ICANN* (pp. 579–586).

Doumas, L. A. A., & Hummel, J. E. (2012). Computational models of higher cognition. In K. J. Holyoak & R. G. Morrison (Eds.), *Oxford Handbook of Thinking and Reasoning.* Oxford University Press.

Eliasmith, C. (2013). *How to build a brain: A neural architecture for biological cognition*. New York, NY: Oxford University Press.

Feldman, J. A. (2006). *From molecule to metaphor: A neural theory of language*. Cambridge, MA: MIT Press.

Gayler, R. W. (2004). Vector symbolic architectures answer Jackendoff's challenges for cognitive neuroscience. *CoRR*, *abs/cs/0412059*.

Halford, G. S., Wilson, W. H., & Phillips, S. (2010). Relational knowledge: the foundation of higher cognition. *Trends Cogn Sci*, *14*(11), 497–505.

Knauff, M. (2013). *Space to reason: A spatial theory of human thought*. Cambridge, MA: MIT Press.

Lakoff, G., & Johnson, M. (1999). *Philosophy in the flesh: The embodied mind and its challenge to western thought*. New York: Basic Books.

Lipinski, J., Schneegans, S., Sandamirskaya, Y., Spencer, J. P., & Schöner, G. (2012). A neuro-behavioral model of flexible spatial language behaviors. *J Exp Psychol Learn*, *38*(6), 1490–1511.

Logan, G. D., & Sadler, D. D. (1996). A computational analysis of the apprehension of spatial relations. In P. Bloom, M. Peterson, L. Nadel, & M. Garrett (Eds.), *Language and Space* (pp. 493–529). Cambridge, USA: MIT Press.

Lomp, O., Zibner, S. K. U., Richter, M., Rano, I., & Schöner, G. (2013). A software framework for cognition, embodiment, dynamics, and autonomy in robotics: cedar. In V. Mladenov (Ed.), *ICANN* (pp. 475–482). Heidelberg: Springer.

Madden, C., Hoen, M., & Dominey, P. F. (2010). A cognitive neuroscience perspective on embodied language for human-robot cooperation. *Brain Lang*, *112*(3), 180–8.

Regier, T., & Carlson, L. A. (2001). Grounding spatial language in perception: an empirical and computational investigation. *J Exp Psychol Gen*, *130*(2), 273–98.

Richter, M., Lins, J., Schneegans, S., Sandamirskaya, Y., & Schöner, G. (2014). Autonomous neural dynamics to test hypotheses in a model of spatial language. In P. Bello & et al. (Eds.), *36th CogSci* (pp. 2847–2852). Austin, TX: Cognitive Science Society.

Richter, M., Sandamirskaya, Y., & Schöner, G. (2012). A robotic architecture for action selection and behavioral organization inspired by human cognition. In *IEEE/RSJ IROS* (pp. 2457–2464).

Sandamirskaya, Y., & Schöner, G. (2010). An embodied account of serial order: how instabilities drive sequence generation. *Neural Networks*, *23*(10), 1164–79.

Schneegans, S., & Schöner, G. (2012). A neural mechanism for coordinate transformation predicts pre-saccadic remapping. *Biol Cybern*, *106*(2), 89–109.

Schneegans, S., Spencer, J. P., & Schöner, G. (2015). Integrating what and where: Visual working memory for objects in a scene. In G. Schöner, J. P. Spencer, & the DFT Research Group (Eds.), *Dynamic thinking: A primer on dynamic field theory.* Oxford University Press.

Schöner, G., Spencer, J. P., & the DFT Research Group. (2015). *Dynamic thinking: A primer on dynamic field theory*. Oxford University Press.

Treisman, A., & Gelade, G. (1980). A feature-integration theory of attention. *Cognitive Psychol*, *12*(1), 97–136.

van Hengel, U., Sandamirskaya, Y., Schneegans, S., & Schöner, G. (2012). A neural-dynamic architecture for flexible spatial language: intrinsic frames, the term "between", and autonomy. In *IEEE RO-MAN* (pp. 150–157).