A neural dynamic architecture for the perceptual grounding of simple spatial language

DFT Summer School 2020

Basic Information

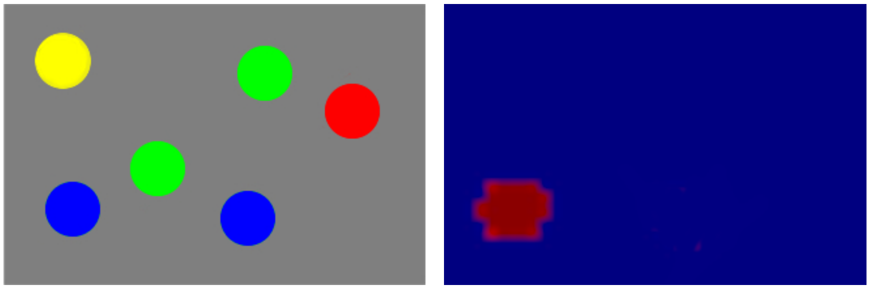
Following [1], the goal of this project is to implement a dynamic field architecture that is able to ground simple spatial language in perception, which means that it is able to find an object in the visual input that matches a linguistic description.

The architecture receives input from a camera or an image, and additional language input. That language input describes an object in terms of a feature and relation to another object, e.g., “a blue object to the left of a green object” (In this tutorial, we limit ourselves to the feature color). The model then grounds that language input by creating a peak in a spatial attention field, which is located on the position of the described object in the perceptual input.

Figure 1a depicts an example image. Grounding “a blue object to the left of a green object” surmounts to creating the peak depicted in Figure 1b.

In the previous tutorial on grounding feature concepts in perception, you already implemented a part of this architecture that is able to find objects with a given feature value. The goal of the present architecture is to extend this architecture by the ability to find a target object, identified by a feature, that stands in a given relation to a reference object identified by a feature. In the example, the blue object is the target object and the green object is the reference object.

Figure 2 shows a depiction of the architecture. In the following, we will go through each of the components in detail, and you will be asked to implement them in cedar.



(a) Example image (b) Spatial attention field

Figure 1: Grounding the relational phrase “a blue object to the left of a green object” in an example image

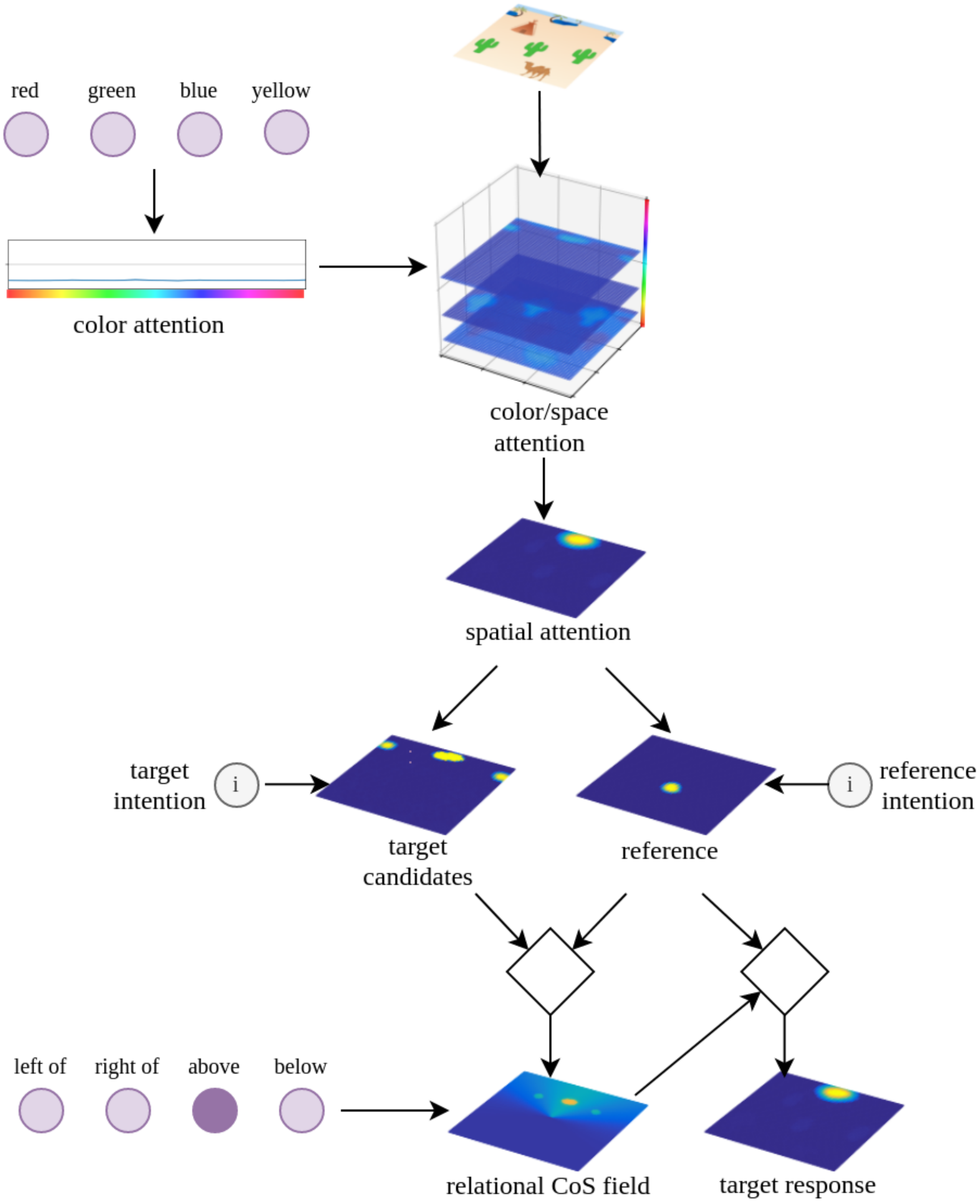


Figure 2: Schematic depiction of the model architecture. The three fields on the top are the ones you built in the visual search project.

Reference field and target candidates field

At the beginning of the grounding process, the location with the mentioned reference feature (green in the example) is attended in the *spatial attention field* and stored in a *reference field*. That field receives input from the *spatial attention field*. It is selective, such that a selection decision is made in case multiple objects with the reference feature exist. Furthermore, it is self-sustained, which means that the peak remains in the absence of input. Finally, it only forms a peak when it receives global excitatory input from a *reference intention node*. Create the field and the node (Hint: Selective fields require strong global inhibition. Self-sustained fields require strong local excitation.). Try to store an object location there by controlling a color node and the *reference intention node* through Boosts, and make sure that the peak is sustained when the Boosts are deactivated.

After a reference object has been selected, a set of target candidates, i.e., the locations of all objects with the target feature, have to be stored in a *target candidates field*. That field receives input from the *spatial attention field* and is self-sustained, such that peaks remain in the absence of input. It only forms peaks when it receives global excitatory input from a *target intention node*. Create the field and the node (Hint: Self-sustained fields require strong local excitation. Since this field is not selective, mid-range inhibition is necessary to prevent unlimited growth of peaks.). Again, try to store object locations there by controlling a color node and the *reference intention node* through Boosts. Make sure that the peaks are sustained when the Boosts are deactivated.

Grounding relations

Once a reference object and a set of target candidate objects have been stored, we want to find all target objects that stand in a given spatial relation to the reference object (e.g., all blue objects to the left of a certain green object). The first step to achieving this is to transform the spatial locations of the target candidates into a different coordinate system that is centered on the reference object. A proposal for a neural basis of coordinate transforms is explained in [3]. For performance reasons, we implement it approximatively as a cross-correlation between the outputs of the *target candidates field* and the *reference field*. For this purpose, create a convolution component and a flip component. Connect the *target candidates field* to the first input of the convolution component. Further, connect the *reference field* to the flip component, and connect the output of that component to the second input of the convolution component. Set the "mode" property of the convolution to "full".

The next step is to create a *relational CoS field*. That field receives the target candidates in the relative coordinate system as input, creating subthreshold bumps of activation there. Moreover, it receives an activated spatial relation pattern (left of, right of, above, or below) as input. When these two inputs overlap, peaks form on the relative positions of the target candidates that bear the activated spatial relation to the reference object. Create the field and the four spatial relation patterns. The cedar component for a spatial relation pattern is called SpatialTemplate; its "mu th" property specifies the angle of the spatial pattern in radians. Further, create four spatial relation concept nodes. Each of these nodes should feed its respective spatial relation pattern into the *relational CoS field* when it is active. Biologically, this means that the synaptic weights between the node and the field correspond to the spatial pattern. In cedar, you can achieve this using a ComponentMultiply that receives input from the node and the pattern, and feeds its output to the field. Test your architecture by selecting a reference object and a set of target candidates, and then activating a spatial relation concept node. Make sure that the *relational CoS field* forms peaks on the spatial locations of all and only the target candidates that bear the activated spatial relation to the reference object.

Now create another coordinate transform that converts the peak locations back to the coordinate system of the visual input, and feed its output into a *target response field*.

The model you implemented so far has been proposed by [1]. It is self-contained and, therefore, can already serve as a basis for your project presentation. So make a backup, have a cup of tea, and relax. If you still have time and energy, proceed with the optional upcoming section.

Optional: Hypothesis testing and process organization

Recall that the *reference field* is selective. This means that, when there are multiple reference objects with a given feature, an arbitrary selection decision is made. It might turn out later that this selection decision is wrong. If the *relational CoS field* does not form any peaks, this is an indicator that a wrong reference object has been selected. For instance, if the input phrase is “a blue object to the left of a green object”, and the wrong green object has been selected, the *relational CoS field* does not form any peaks, even though there is a blue object to the left of the other green object. In such cases, the original selection decision has to be undone, and a new reference object has to be selected.

An extension of the architecture that adds the ability of autonomous hypothesis testing is described in [2]. That paper describes the steps that are necessary to supplement the architecture you built so far by this ability. It further adds a process organization system that guides the unfolding of the architecture, taking away the necessity to guide its unfolding by providing manual boost inputs at appropriate moments in time. Follow that paper to extend your architecture, and feel free to ask questions if it is unclear to you how to implement the components described there in cedar.

References

[1] J Lipinski, S Schneegans, Y Sandamirskaya, J P Spencer, and G Schöner. A neuro-behavioral model of flexible spatial language behaviors. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 38(6):1490–1511, 2012.

[2] Mathis Richter, Jonas Lins, Sebastian Schneegans, Yulia Sandamirskaya, and Gregor Schöner. Autonomous neural dynamics to test hypotheses in a model of spatial language. In *Proceedings of the Annual Meeting of the Cognitive Science Society, 36(36)*. Cognitive Science Society, 2014.

[3] Sebastian Schneegans and Gregor Schöner. A neural mechanism for coordinate transformation predicts pre-saccadic remapping. *Biological cybernetics*, 106(2):89–109, 2012.