Visual working memory (VWM) is a complex process of encoding and retrieving information. Current models provide a theoretical framework for understanding how information is stored\(^1\), but few models are explicit about the underlying neural mechanisms of VWM. A neural simulation can generate more explicit predictions about the structure of memory and the time course of encoding.

**Model architecture:**
- Tokens index stored representations
- Types represent stimulus features
- The Binding pool is a shared resource pool of distributed representations in which multiple stimuli are stored

**Tokens** = tokens project to overlapping subsets of the binding pool

**Binding Pool** = stores distributed Type representations, sustains activity during retention interval

**Types** = points within a color dimension (or any other)

**Encoding and retrieval of a single Type:**
- One token is activated per item
- Encoding is serial
- Types and tokens both project to the binding pool
- Binding pool nodes receiving convergent input are activated

**Type nodes are treated as vectors in a color wheel.** The population mean is computed by adding the vectors.
- Location = retrieved color value
- Vector length = ‘confidence’ of retrieval

In order to better understand the different types of responses, the model output was analyzed with a mixture model.1

The distributed nature of the Binding Pool results in a noisy retrieval of the original Type representation. Typically, the original Type input is activated to a higher degree than other Type nodes.

**Retrieval:** Information stored in a given Token can be reconstructed at the Type layer.

The same connections which had been used to encode the type representation into the binding pool are now activated in the reverse direction.

**Simultaneously and sequentially presented colors exhibit similar within-task interference for working memory representations**

Garrett Swan and Brad Wyble
The Pennsylvania State University

### Experiment: Sequential vs. simultaneous presentation

A property of this model is that simultaneously presented stimuli are encoded serially. Thus, a prediction is that encoding of simultaneous and sequential stimuli should produce similar patterns of errors.

#### Results

- **2x3 ANOVA:** No significant main effects for Block or interactions after Bonferroni corrections.

The null pattern of results from the behavioral data lends credence to the serial encoding inherent in the model.

The model can also simulate errors in change detection paradigms and can reproduce ensemble statistic effects (see supplemental).

Future work will generate predictions to be tested empirically.


Model and behavioral data were analyzed using code from Paul Bays at www.bayslab.com
How can the model be applied to change detection tasks?

Contrary to the continuous report task, each Type needs to be reconstructed from memory and compared to the originally input Type in a change detection task. For 50% of the simulations, a reconstructed Type will be compared to a different color, representing a changed item. In order to detect a change, we utilize the properties of the mean vector to create a two-dimensional threshold:

Old item comparison

New item comparison
What about ensemble statistic effects?

For the model to be able to simulate ensemble statistic (specifically, Brady and Alvarez 2011) an additional Type layer needs to be added.

The process of encoding is the same as before except for the addition of a new Type layer so that there are two features.

Binding color and size together into a single representation.
During the first retrieval step, Type layer 2 is reconstructed to determine what colored circles to retrieve.

The distributed representations of the red items cause a systematic bias when reconstructing Type layer 1. This bias simulates the ensemble effect without requiring a separate ensemble representation.

Retrieval of a single Type now requires two steps:

1. [Diagram showing the first retrieval step]
2. [Diagram showing the second retrieval step]