Encoding syntactic positions in working memory: A computational approach

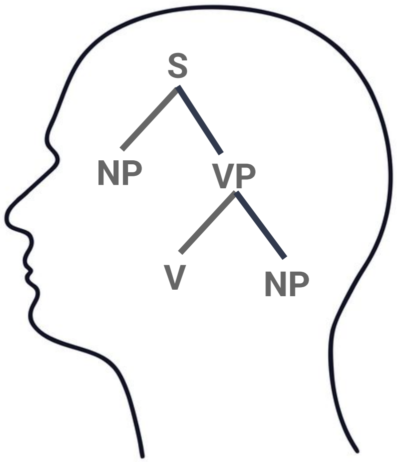
Eva Neu[1], Maayan Keshev[2] and Brian Dillon[1]

[1]UMass Amherst, [2]The Hebrew University of Jerusalem

2025 LSA Annual Meeting, January 9–12

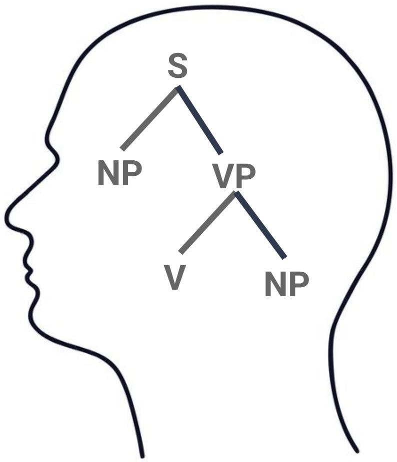
Our talk in a nutshell

How is the syntactic structure of a sentence encoded in the mind of the language user during processing?



Our talk in a nutshell

We propose a computational model of syntax in working memory that is based on processing data from agreement attraction



Agreement attraction

Agreement is susceptible to interference from distractors (e.g., Bock and Miller, 1991; Wagers et al., 2009):

The **key** to the **cabinets** is/*are on the table.

Agreement attraction

Agreement is susceptible to interference from distractors (e.g., Bock and Miller, 1991; Wagers et al., 2009):

> The **key** to the **cabinets** is/*are on the table.

We can learn about working memory from cases where it goes astray!

Agreement attraction

Interference rates between target and distractor are modulated by similarity:

1. Semantically (Fedorenko et al., 2006; Gordon et al., 2011; Smith et al., 2021; Van Dyke, 2007)
2. Morphologically (Badecker & Kuminiak, 2007; Sims, 2012, Slioussar et al., 2022)
3. Syntactically (Arnett & Wagers, 2017; Franck et al., 2002; Van Dye, 2007; Van Dyke & McElree, 2011)

Agreement attraction

Interference rates between target and distractor are modulated by similarity:

1. Semantically (Fedorenko et al., 2006; Gordon et al., 2011; Smith et al., 2021; Van Dyke, 2007)
2. Morphologically (Badecker & Kuminiak, 2007; Sims, 2012, Slioussar et al., 2022)
3. Syntactically (Arnett & Wagers, 2017; Van Dye, 2007; Van Dyke & McElree, 2011)

Agreement attraction

Interference rates between target and distractor are modulated by similarity:

1. Semantically (Fedorenko et al., 2006; Gordon et al., 2011; Smith et al., 2021; Van Dyke, 2007)
2. Morphologically (Badecker & Kuminiak, 2007; Sims, 2012, Slioussar et al., 2022)
3. Syntactically (Arnett & Wagers, 2017; Van Dye, 2007; Van Dyke & McElree, 2011)

Structurally closer distractors interfere more (Franck et al., 2002; Bock and Cutting, 1992):

The **helicopter** for the **flight(s)** over the **canyon(s)** is/*are rusty.

Agreement attraction

To explain why this effect of similarity obtains, we develop a computational model of representing sentences in working memory

Agreement attraction

To explain why this effect of similarity obtains, we develop a computational model of representing sentences in working memory

The model uses vector-symbolic representations, where similarity is represented as cosine similarity (e.g., Cho et al., 2020; Piantadosi et al., 2024; Plate, 1997; Smolensky, 1990; Smolensky et al., 2010)

The architecture

Sentences are encoded in working memory by binding lexical items to syntactic positions

(Keshev et al., 2024a)

The architecture

Sentences are encoded in working memory by binding lexical items to syntactic positions

(Keshev et al., 2024a)

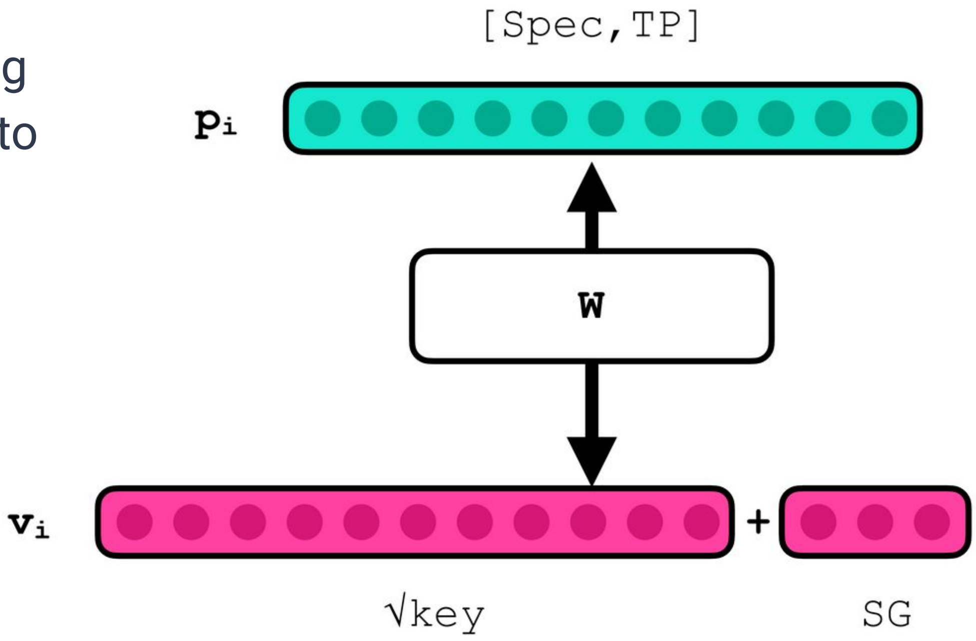
The key to the cabinets is rusty.

The architecture

Sentences are encoded in working memory by binding lexical items to syntactic positions

(Keshev et al., 2024a)

The <u>key</u> to the cabinets is rusty.

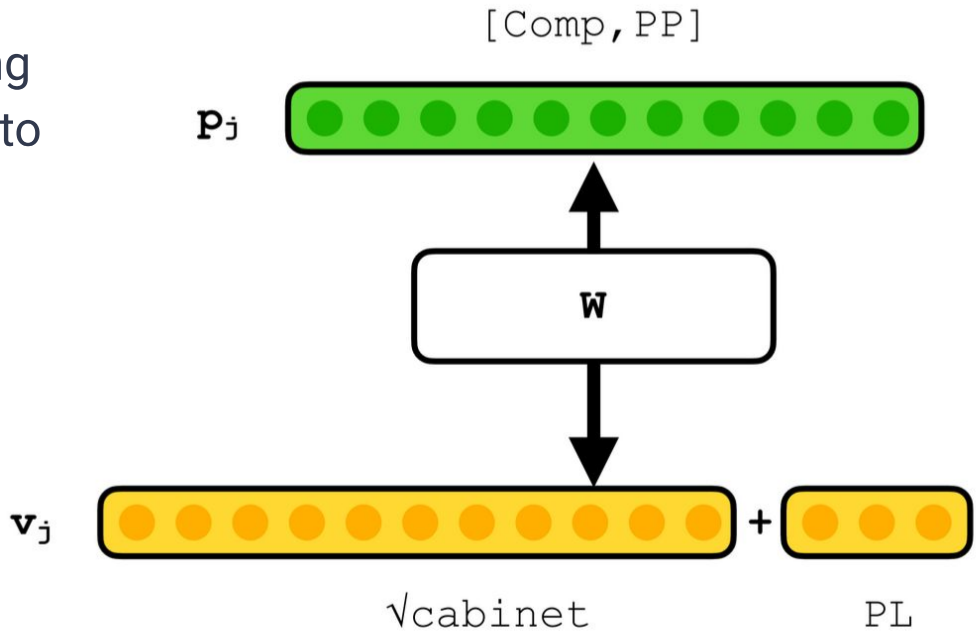


The architecture

Sentences are encoded in working memory by binding lexical items to syntactic positions

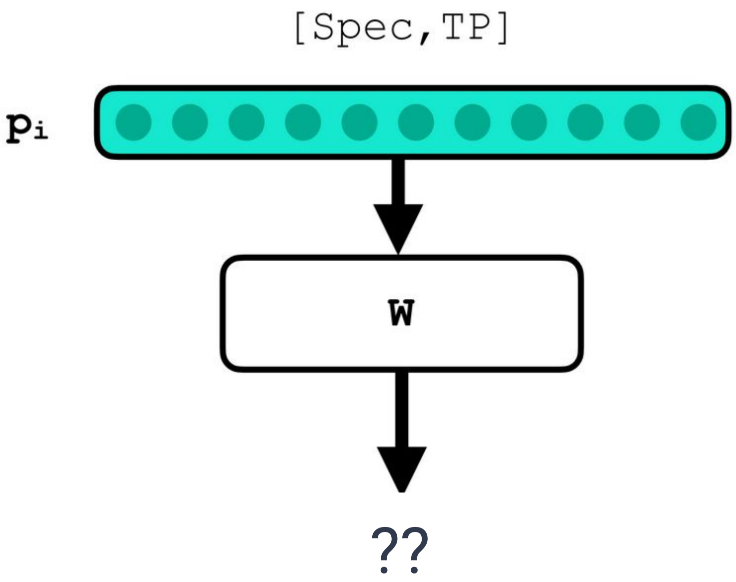
(Keshev et al., 2024a)

The key to the <u>cabinets</u> is rusty.



The architecture

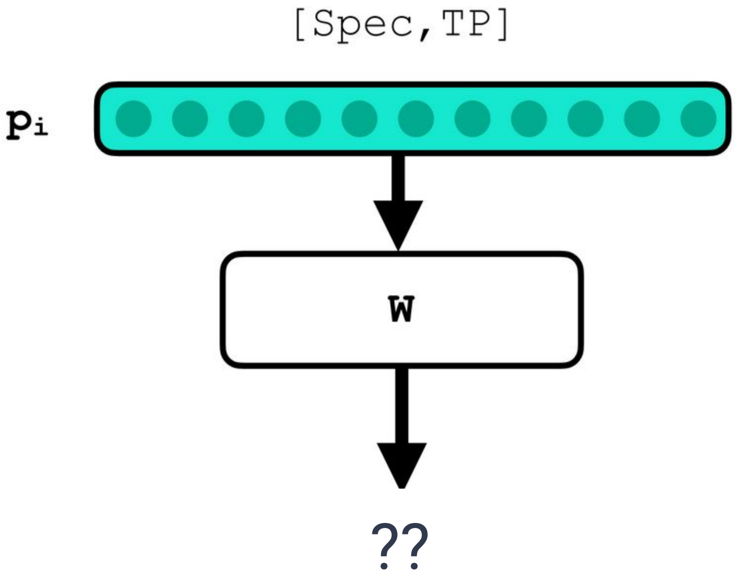
When processing agreement, speakers need to retrieve the item vector bound to subject position



The architecture

Distributed vector representations
→ Same units are activated to
code different positions/items

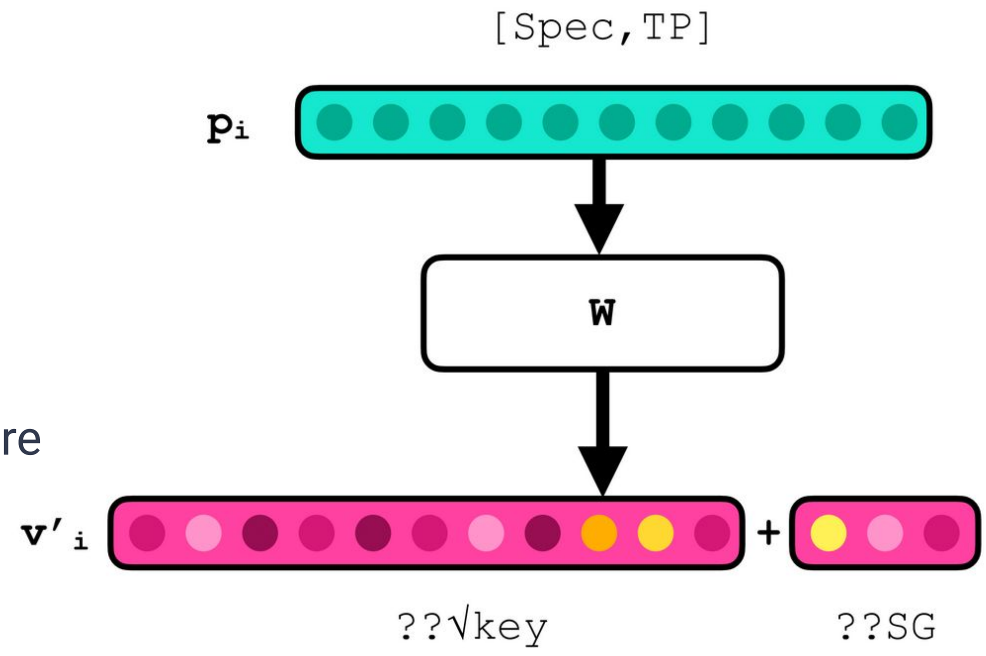
All associations are superimposed
on the same connection matrix



The architecture

Unless position vectors are orthogonal, item vector is not perfectly recovered

Items in similar positions are more likely to be misretrieved



Our task for today

Develop a method to systematically compute position vectors such that higher cosine similarity corresponds to higher rates of interference

Computing position vectors

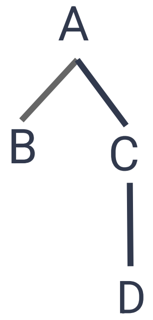
Step 1 Assign a constituency parse to the sentence

Step 2 Assign each node a base vector depending on its category

Step 3 For each node, compute its position vector as the weighted sum of its own base vector and its mother's position vector (TCM; Howard and Kahana, 2002)

position vector (x) =
α × base vector (x) + (1 − α) × position vector (x's mother)

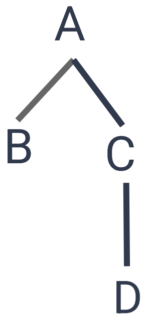
Computing position vectors



position vector (x) =

$\alpha \times$ base vector (x) + $(1 - \alpha) \times$ position vector (x's mother)

Computing position vectors

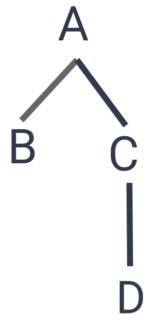


position vec (A) = base vec (A)

position vector (x) =

α × base vector (x) + (1 − α) × position vector (x’s mother)

Computing position vectors

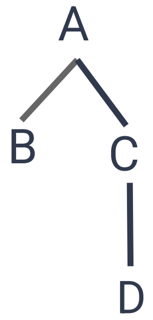


position vec (A) = base vec (A)

position vec (B) = α × base vec (B) + (1 − α) × position vec (A)

position vector (x) =
α × base vector (x) + (1 − α) × position vector (x’s mother)

Computing position vectors



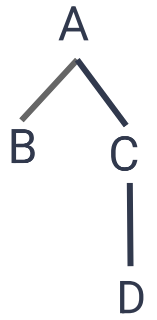
position vec (A) = base vec (A)

position vec (B) = $\alpha \times$ base vec (B) + $(1 - \alpha) \times$ position vec (A)

position vec (C) = $\alpha \times$ base vec (C) + $(1 - \alpha) \times$ position vec (A)

position vector (x) =
$\alpha \times$ base vector (x) + $(1 - \alpha) \times$ position vector (x’s mother)

Computing position vectors



position vec (A) = base vec (A)

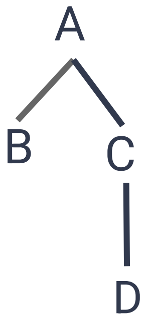
position vec (B) = α × base vec (B) + (1 − α) × position vec (A)

position vec (C) = α × base vec (C) + (1 − α) × position vec (A)

position vec (D) = α × base vec (D) + (1 − α) × position vec (C)

position vector (x) =
α × base vector (x) + (1 − α) × position vector (x’s mother)

Computing position vectors



position vec (A) = base vec (A)

position vec (B) = $\alpha \times$ base vec (B) + $(1 - \alpha) \times$ position vec (A)

position vec (C) = $\alpha \times$ base vec (C) + $(1 - \alpha) \times$ position vec (A)

position vec (D) = $\alpha \times$ base vec (D) + $(1 - \alpha) \times$ position vec (C)

position vector (x) =
$\alpha \times$ base vector (x) + $(1 - \alpha) \times$ position vector (x's mother)

Computing position vectors

1. Position vectors contain the category information of the current node and all dominating nodes

2. More distant nodes make up a smaller part of the representation

Model results: Cosine similarities

The **helicopter** for the **flight(s)** over the **canyon(s)** is rusty.

Model results: Cosine similarities

The **helicopter** for the **flight(s)** over the **canyon(s)** is rusty.

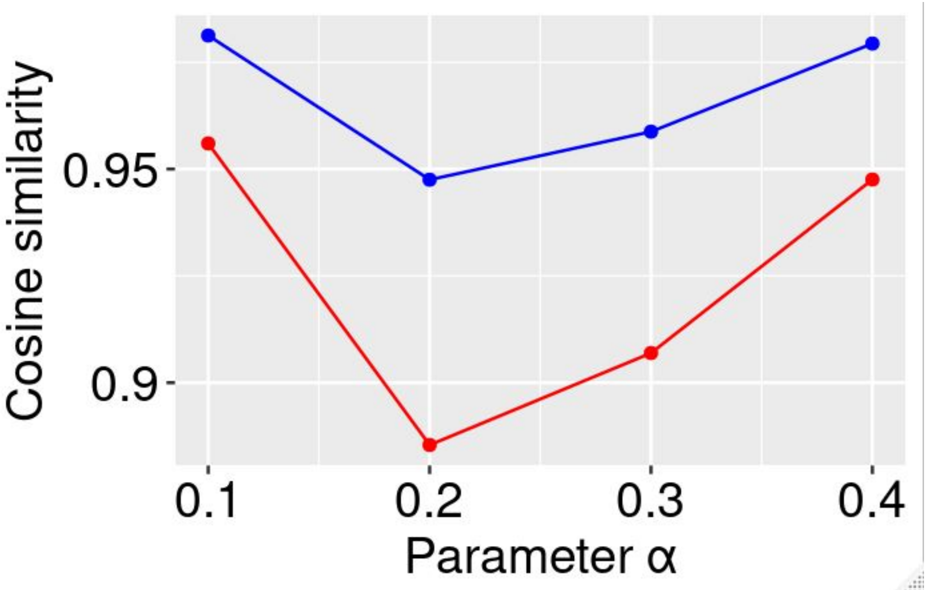
‘Flight’ is more likely to cause interference than ‘canyon’

Cosine similarities

The **helicopter** for the **flight(s)** over the **canyon(s)** is rusty.

'Flight' is more likely to cause interference than 'canyon'

Model correctly predicts 'flight' to be more similar to target than 'canyon'



From cosine similarities to predicting error rates

To test whether the model captures the effect of syntactic similarity on interference rates…

1. We fit the model to empirical data for a single distractor using maximum likelihood estimation,

2. We then test the predictions of this fitted model against empirical data for two distractors in different syntactic positions

Single distractor

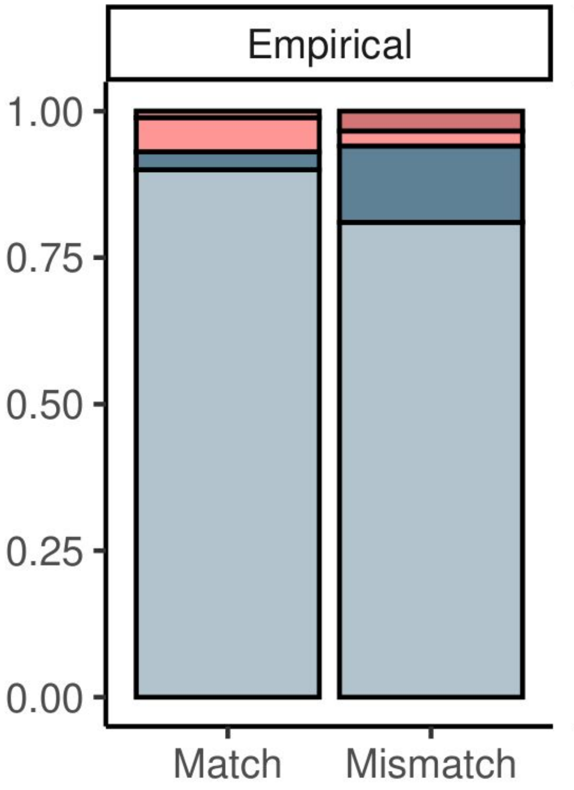
Empirical data: 4-AFC task for sentences with a single distractor, which is either singular or plural (Keshev et al., 2024b)

The apprentice of the chef/chefs worked diligently.
Who worked diligently?
The apprentice / **the apprentices** / **the chef** / **the chefs**

Distractor – Nonveridical
Distractor – Veridical
Target – Nonveridical
Target – Veridical

Single distractor

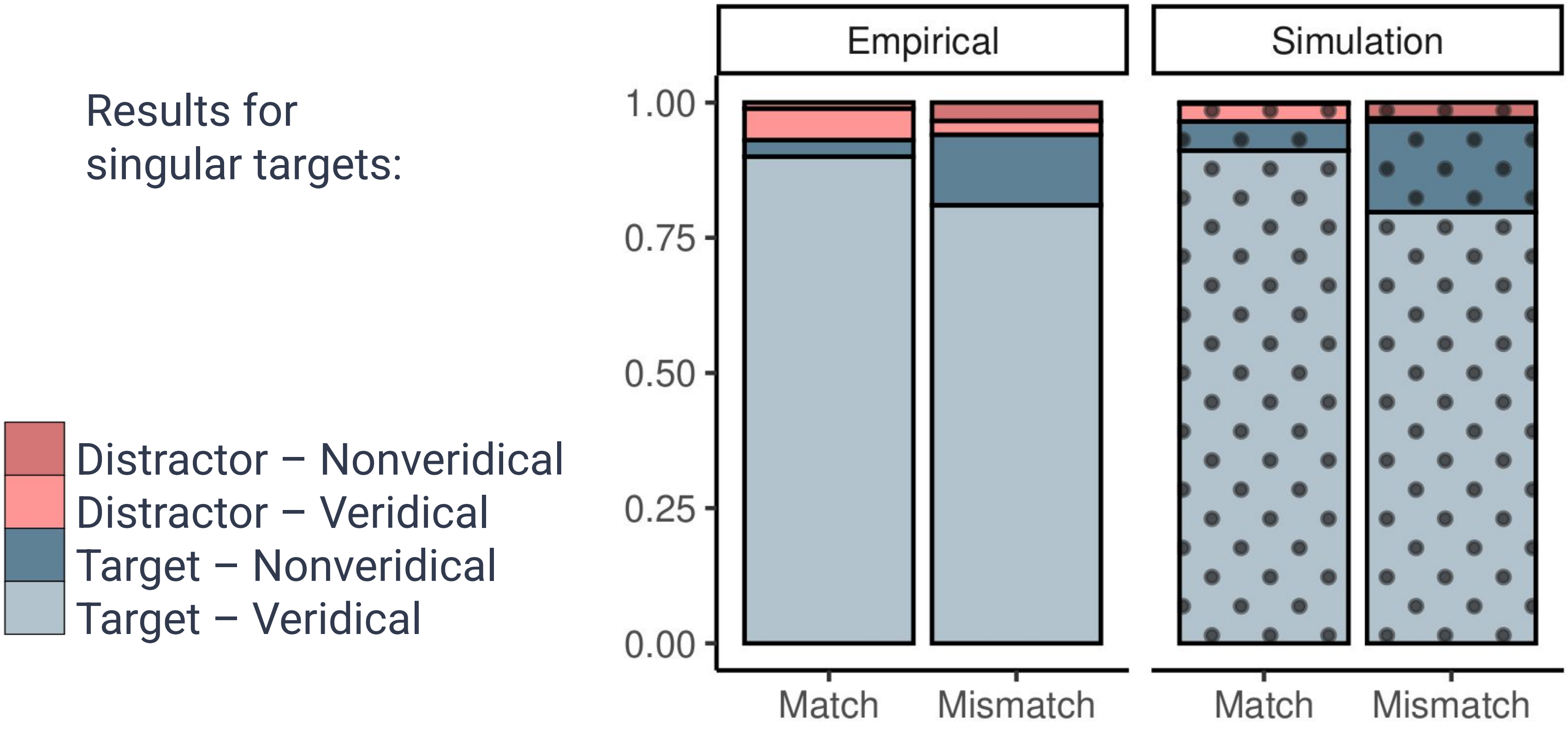
Results for
singular targets:



Distractor – Nonveridical
Distractor – Veridical
Target – Nonveridical
Target – Veridical

Single distractor

Results for singular targets:



Two distractors

We use the fitted model to make predictions for interference rates from distractors in different syntactic positions

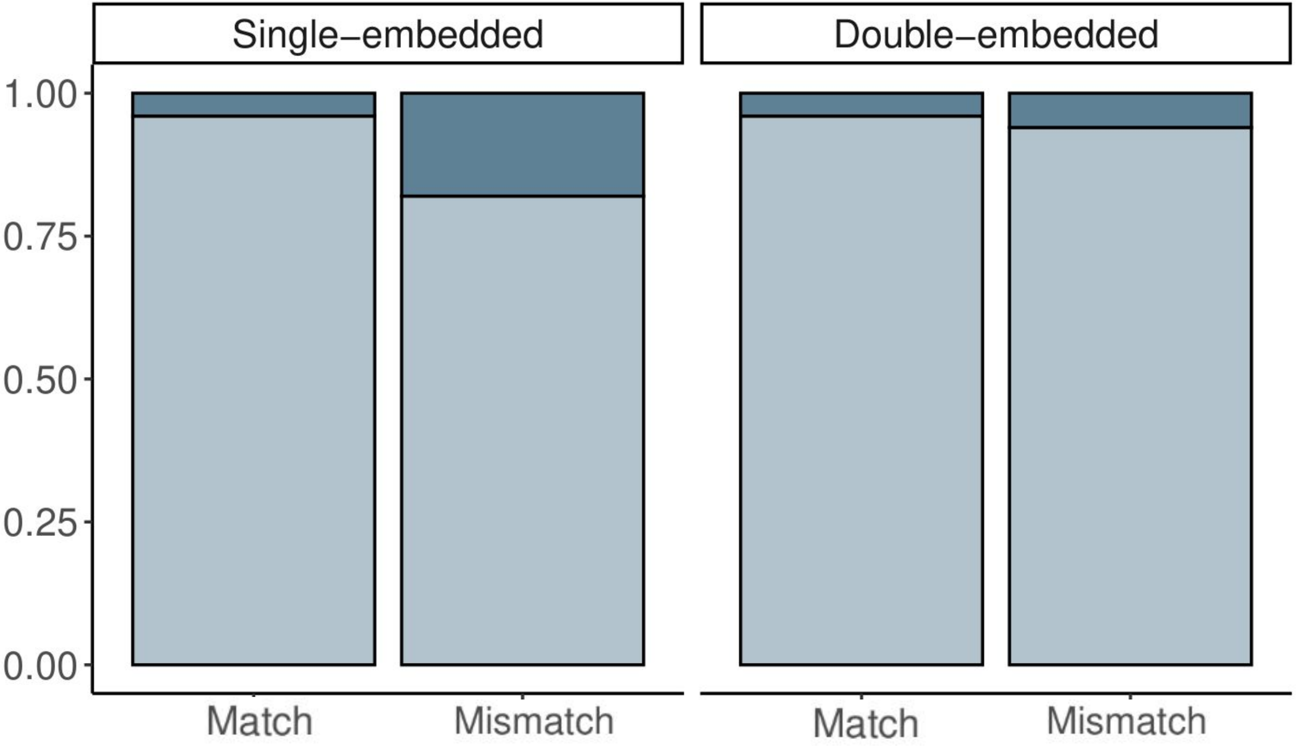
Empirical data: 2-AFC task for subjects with two distractors, one of which is plural

(Keung & Staub, 2018)

The **helicopter** for the **flight(s)** over the **canyon(s)**...
…**is** / …**are**

Two distractors

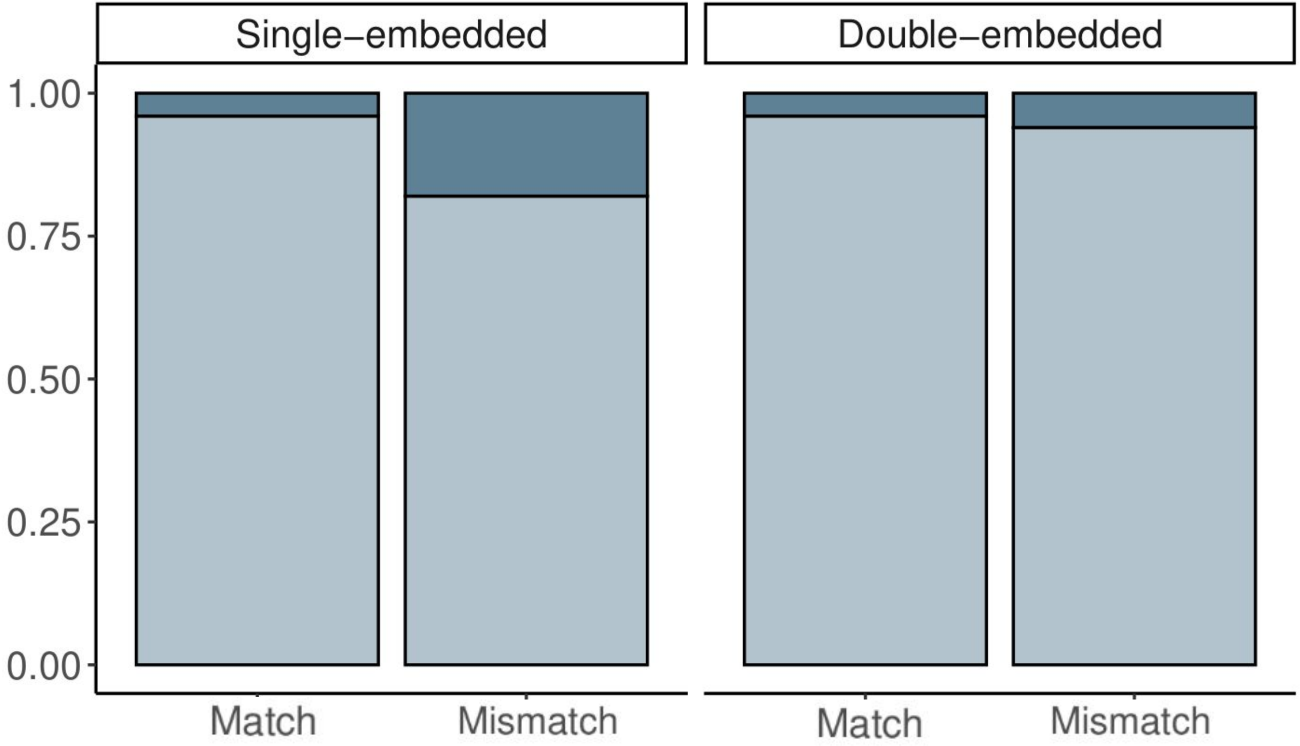
Empirical data:



Two distractors

Empirical data:

Single-embedded distractors: plural distractors trigger more plural agreement

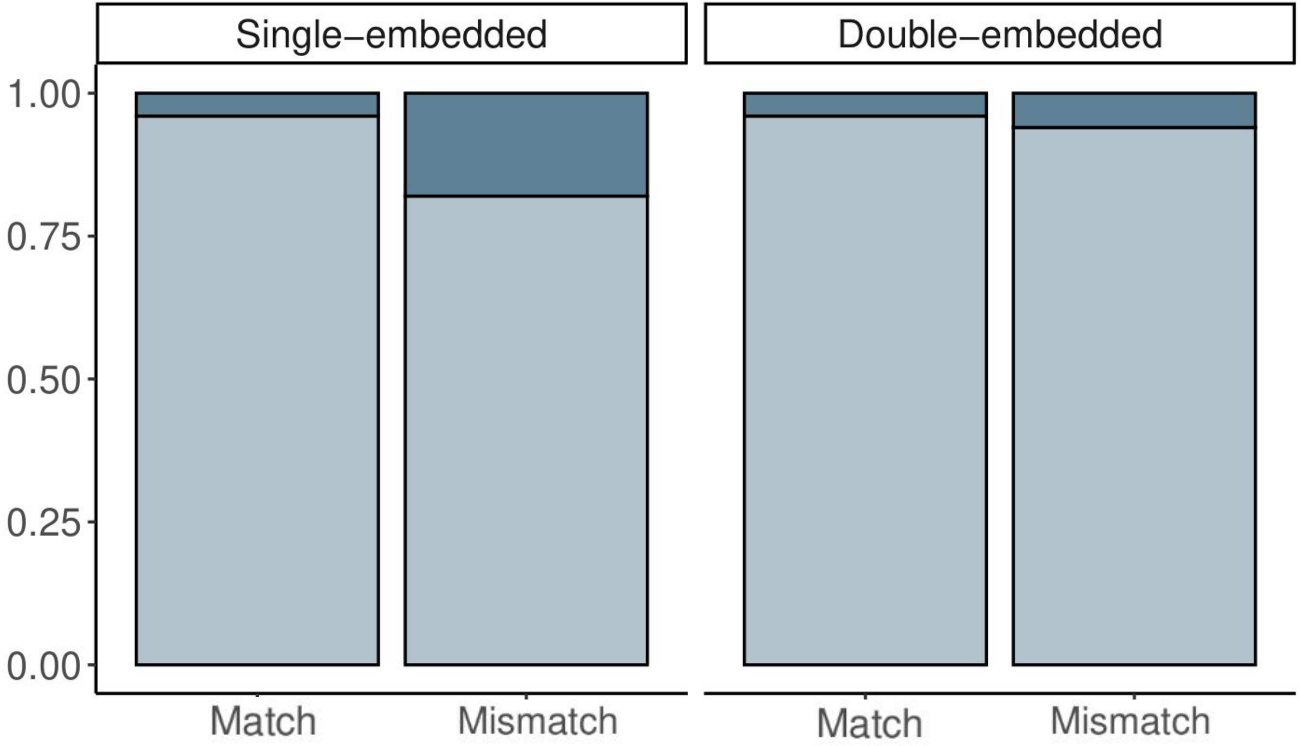


Plural
Singular

Two distractors

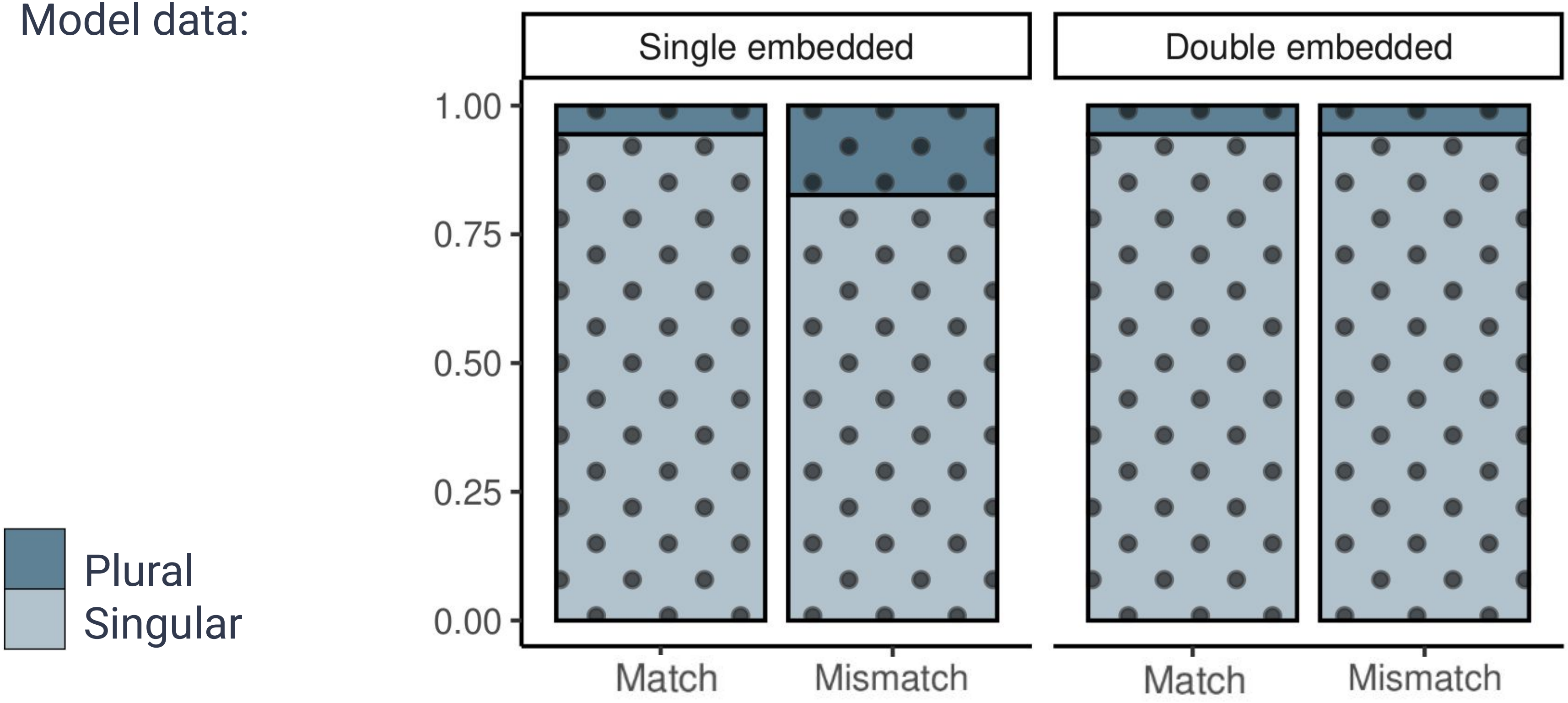
Empirical data:

Double-embedded distractors: close to no interference



Two distractors

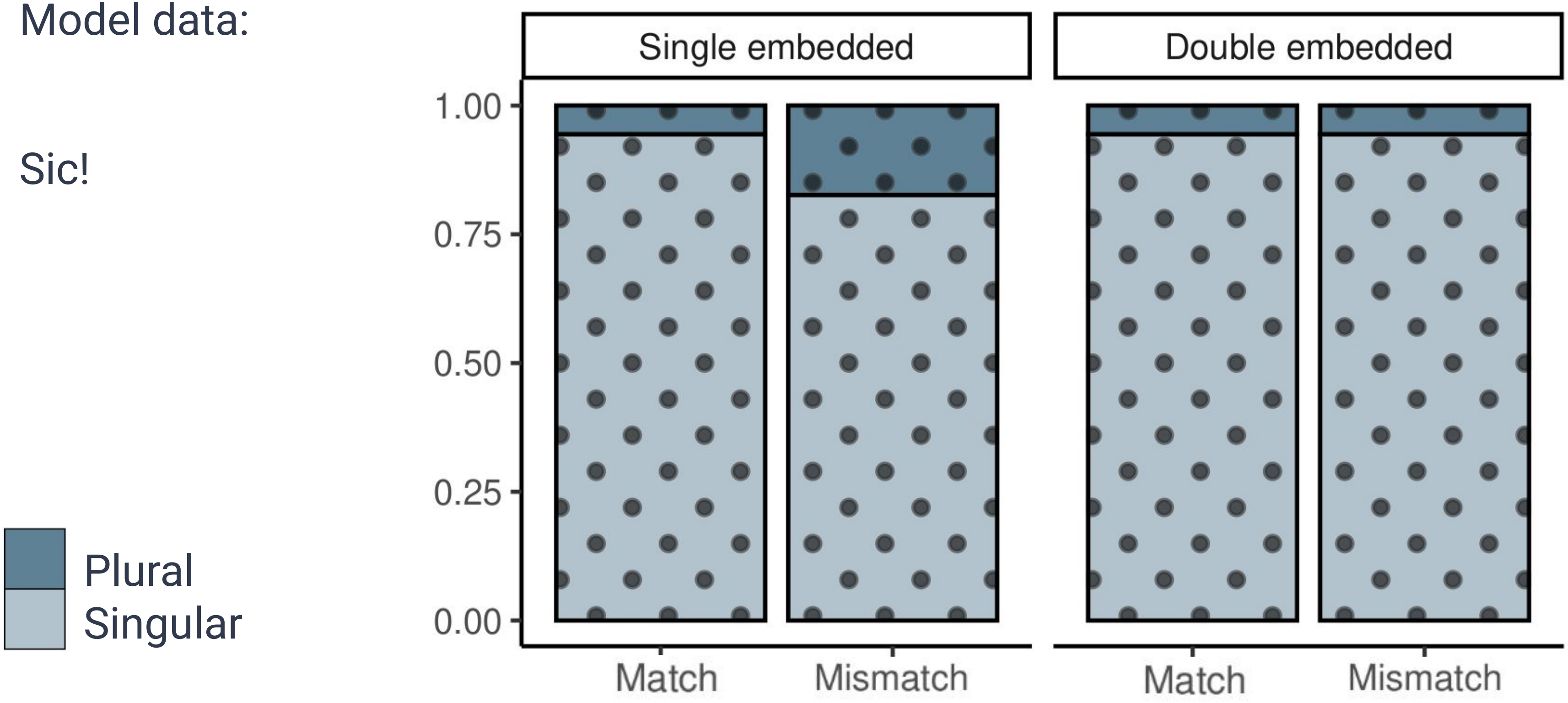
Model data:



Two distractors

Model data:

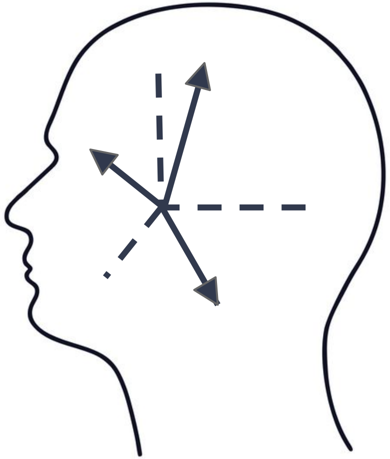
Sic!



Conclusion

We have developed an algorithm to compute vector representations of syntactic positions

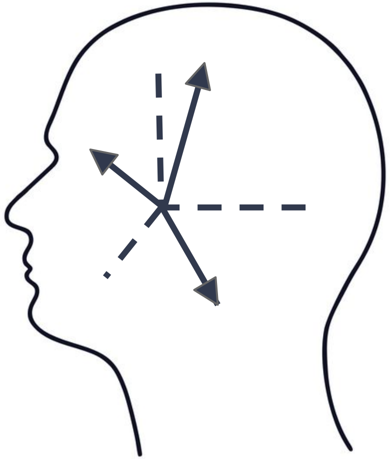
Cosine similarity between these vectors captures which positions are more likely to cause interference in agreement attraction



Conclusion

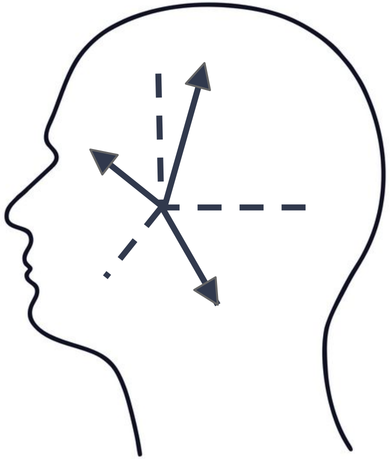
Possible future directions:

1. Predict and empirically test other syntactic configurations,
2. Add boundaries (e.g., clausal domains),
3. Use non-orthogonal base vectors,
4. Apply algorithm to a dependency structure instead,
5. What about movement?,
6. …



Conclusion

Possible future directions:



Many thanks to Samuel Amouyal, Mandy Cartner, Niki Koesterich, Aya Meltzer-Asscher and Stephanie Rich

This research was supported by the National Science Foundation (NSF BCS-1941485 to BD) and the Binational Science Foundation (NSF-BSF 2146798 to BD)