

# Diversity and Interaction Structure Shape Performance and Search Dynamics in Joint Cognitive Search: An Agent-Based Simulation

Anonymous CogSci submission

## Appendix

### Generating agent populations

To generate agent populations whose semantic memories differed increasingly from the original semantic space, we proceeded as follows. First, we sampled 15 linearly spaced values from the distribution of Euclidean distances between pairs of vectors in the original semantic space. Let  $D$  be the distribution of pairwise distances  $d(x, y)$  between all pairs of vectors  $x$  and  $y$ , and let  $S := \{s_0, s_1, \dots, s_{14}\}$  be the set of equally spaced distance values computed from the distance distribution  $D$  (henceforth: *shuffling thresholds*).

For each shuffling threshold  $s$ , we generated a population  $A_s$  composed of 100 agents by repeating the following procedure for 100 iterations. At each iteration  $i$ :

- We construct the set  $P_s$  containing all pairs of words  $\langle x, y \rangle$  whose Euclidean distance in the original semantic space is lower than  $s$ ;
- We randomly sample 70 percent of all items in  $P_s$ ;
- We reshuffle the order of the resulting list;
- We iteratively swap the position of all word pairs;

The new semantic model resulting from this process is the semantic memory of a newly instantiated agent  $A_{s_i}$ .

By repeating this 100 times for each shuffling threshold  $s$ , we generate 15 agent populations with 100 distinct agents each. As the shuffling threshold  $s$  increases, agents' semantic memories become increasingly different from the original vector space. Lower values of  $s$  result in agents with only local differences from the original semantic space, as only close-by words are swapped (e.g., “dog” and “cat”). For higher distance thresholds, not only will more word pairs be swapped, but words that are further apart will be swapped (e.g., “dog” and “alpaca”), which induces diversity not only in local neighborhoods, but also in the global structure of the space.

The first population, generated with the lowest shuffling threshold  $s_0$ , was set as the reference population. This is a group whose agents' semantic memories differ very little from the original word2vec model. By pairing these with agents from other populations, we generate pairs whose individual members are increasingly different from each other, or, in other words, pairs of agents that have increasing levels of *cognitive diversity*. This is why we refer to agent populations as “diversity levels”: when paired with a member of the reference group, agents from groups that deviate substantially from

the original matrix will yield more diverse pairs, while members of groups that deviate very little from the original matrix will yield pairs with low internal diversity.

### Defining linearly spaced diversity levels

Given the far from uniform distribution of distances between word pairs (see Figure 1), the procedure outlined in the previous paragraph yields populations whose average distances from reference agents in  $A_0$  are not linearly spaced. Let's define the overall distance between two agents as the average Euclidean distance between vectors of corresponding words. More formally, given any two agents with semantic memories (matrices)  $X$  and  $X'$ , their distance will be defined as:

$$\frac{1}{|X'|} \sum_{x' \in X', x'' \in X''} \sqrt{\sum_{i=0}^d (x''_i - x'_i)^2}$$

where  $x_i$  is the  $i$ -th word vector in the first agent's semantic memory ( $X$ ),  $x'_i$  is the corresponding vector in the second agent's semantic memory ( $X'$ ), and  $d$  is the dimensionality of the vector space. The aggregate distance of a given agent population  $A_i$  from the reference population  $A_0$  will be the average of all distances between pairs of agents  $\langle a_{0j}, a_{ij} \rangle$ , such that  $a_{0j} \in A_0$  and as  $a_{ij} \in A_i$ .

When computing aggregate distances between each population and the reference population  $A_0$ , we observe that, for populations generated with sampling thresholds equal or higher to  $s_8$ , distance values plateau. We therefore formally model and estimate the relation between values of shuffling thresholds and distances to select shuffling thresholds that yield linearly spaced distance values.

We model the relationship between shuffling thresholds  $s$  and population distances  $D_s$  as a logistic function with an added offset parameter:

$$D_s = \frac{a}{1 + e^{-k(s-s_0)}} + \text{offset}$$

We estimate the parameters using SciPy (Virtanen et al., 2020) (see Figure 16). We then use the inverse of the estimated function to sample 20 shuffling thresholds  $s$  which, when applying the same agent generation procedure described in the previous section, yield agent populations whose average distances from agents in the reference population are linearly spaced. **The resulting 20 agent populations, which include 100 agents each, are the populations used throughout all three experiments reported in the present paper.**

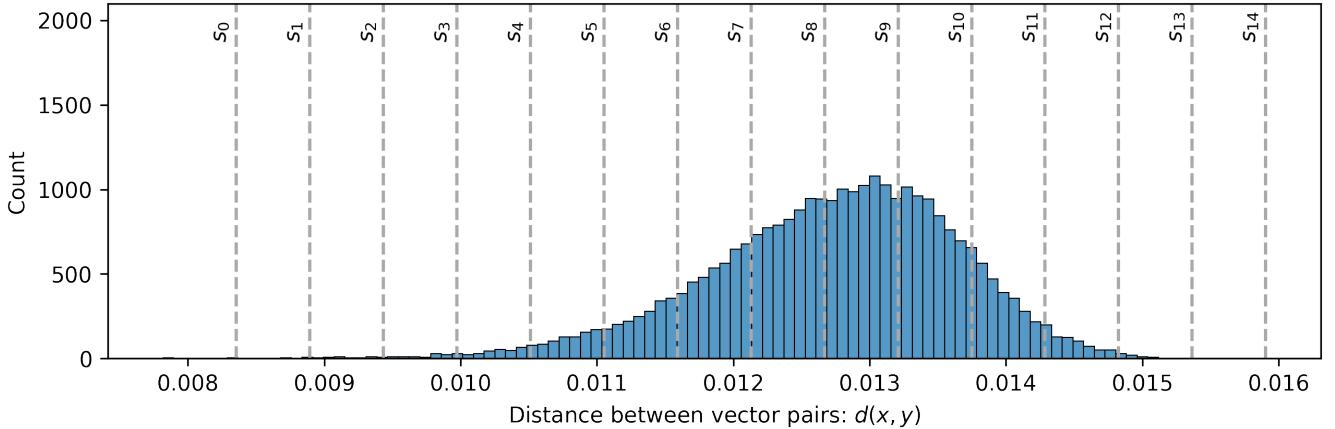


Figure 1: Distribution of distances between all pairs of vectors  $\langle x, y \rangle$  in the word2vec model’s embedding space, and linearly spaced shuffling thresholds sampled to generate agent populations.

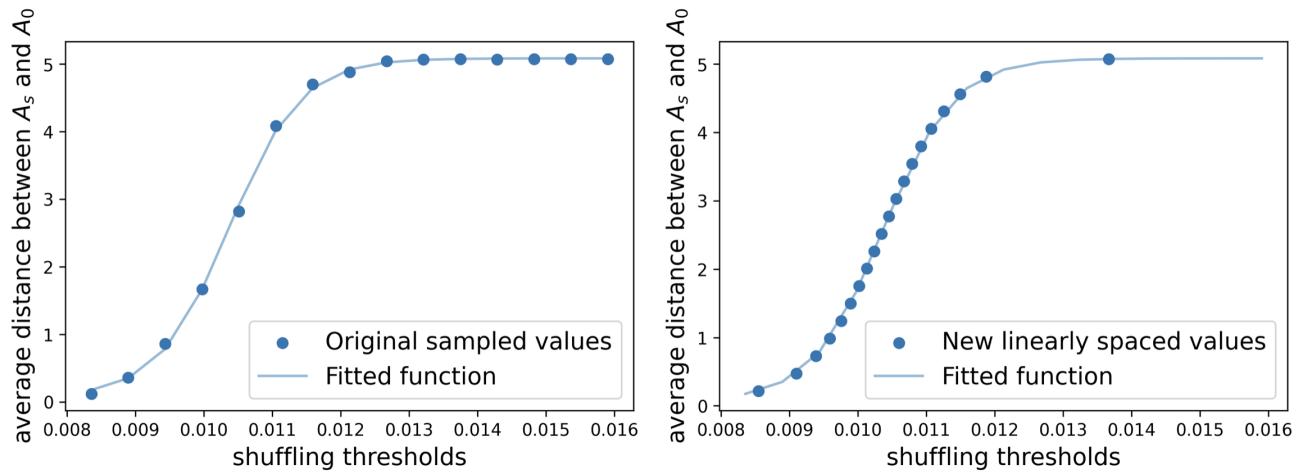


Figure 2: Generalized logistic curve linking *shuffling thresholds* to cognitive distance between the population resulting from a given shuffling threshold  $s$  ( $A_s$ ) and the reference population  $A_0$ .

### Instantiating the final populations

- **Step 1:** Starting from the original word2vec model, we generate a number of agent populations whose semantic memories incrementally differ from the original embedding space. To achieve this, we define 15 linearly spaced threshold values (*shuffling thresholds*), and, for each threshold, we create a set of “noised” copies of the original embedding space by iteratively swapping the position of animal pairs whose distance in the original space is lower than the shuffling threshold. This yields 15 populations ( $A_{i \in \mathbb{Z} \cap [0, 14]}$ ) of 100 agents each;
- **Step 2:** We set the first of these populations (the one corresponding to the lowest shuffling threshold, and therefore the most similar to the original embedding space) as the reference population ( $A_0$ );
- **Step 3:** We compute the average cognitive diversity between the reference population  $A_0$  and each of the remain-

ing populations  $A_{i \in \mathbb{Z} \cap [1, 14]}$ ). Cognitive diversity between  $A_0$  and a given population  $A_i$  (which we denote as  $D_i$ ) is computed by calculating the average Euclidean distance between vector embeddings of corresponding animals for all possible pair of agents  $A_{0j}, A_{ij}$ , and averaging the resulting pair-level scores;

- **Step 4:** As the resulting diversity values are not linearly spaced, we estimate the mathematical relation between shuffling thresholds used in Step 1 and cognitive diversity ( $D_{i \in \mathbb{Z} \cap [1, 14]}$ ) for each group, and using the resulting equation, sample 20 new shuffling threshold, which we used (as in Step 1) to generate populations of agents with linearly spaced distances from  $A_0$  (see Figure 2);
- **Step 5:** Finally, we randomly pair agents from  $A_0$  with agents from each of the 20 resulting populations. This yields 20 groups of agent *pairs* with increasing within-pair diversity. As high shuffling thresholds result both in more

pairs of animals being swapped and in more distant pairs of animals being swapped, pair diversity is the result of both local and global differences in agents' semantic space.

The resulting 20 groups of agent pairs, each containing 100 pairs of agents, correspond to 20 diversity scores which operationalize our parametric manipulation of cognitive diversity. We will refer to these 20 populations as our *diversity levels*. These populations were used throughout all three experiments reported in the present paper. Note that the advantage of inducing diversity by swapping words whose distance is below a set threshold, rather than simply adding noise to the vector space, is that it makes it possible to systematically manipulate diversity while preserving the topology of the original space. This results in agents having the same performance in the individual condition, and performance differences observed in joint search are thus uniquely dependent on our manipulations of cognitive diversity (Experiment 1), turn-taking structure (Experiment 1-2), and agent complexity (Experiment 1-3).

### Stopping threshold

We selected the stopping threshold for simulations by: a) running individual simulations with parametrically varying thresholds – specifically, quantiles in the distribution of pairwise distances between words in the original vector space; b) visualizing the distribution of agent performances, that is, how many words are named before one of the two stopping conditions is met; c) selecting the threshold that yielded the performance distribution whose mean is closest to half the number of words defined in the vector space (see Figure 3). This leaves ample room for variation induced by the experimental manipulations. The resulting distance was the 0.15 quantile in the distribution of pairwise distances between animals in the original vector space.

### Search Metrics

Fluency hinges on agents not ending up in overly *sparse* neighborhoods (where no sub-threshold associations are available), which can happen:

- as a result of overexploitation (e.g., being very effective in mentioning farm animals can be dangerous, because it leaves a gap in semantic space from where it is potentially difficult to connect to a new patch of animals).
- as a result of ending up in a neighborhood which is per se very sparse (e.g., mentioning the animal *kiwi* can be detrimental, as this word has very few semantic neighbors).

As a consequence, agents' performance depends on agents' ability to avoid overexploiting local neighborhoods, and on agents' ability to avoid venturing into extremely sparse neighborhoods.

We quantify these two indices of agents' behavior using the following metrics:

- **Exploration index:** this is computed as the average distance between two consecutively named animals in a simulation, and it is a proxy for agents' ability to engage in exploratory behavior. In individual search simulations, this is

computed based on distances between vectors for consecutively named animals in the agent's semantic space (henceforth: *individual exploration index*). Note that, when averaging across all simulations, this metric will be identical for all individuals. In joint search simulations, agents' semantic memories differ, and the distance between two given animals can be different for the two agents. We therefore compute two different exploration indices: i) the first is obtained by averaging distances between consecutively named words *relative to the speaker's semantic space*, which we name *active exploration index*, ii) the other *relative to the listener's semantic space*, which we name *passive exploration index*. The former captures the extent to which agents engage in more exploratory search, while the quantifies the amount of exploration caused by interaction with another agent. Formally, exploration for a given simulation can be expressed as:

$$e_i = \frac{1}{T-1} \sum_{t=2}^T \|s_{w_t} - s_{w_{t-1}}\|$$

where  $T$  is the fluency of the agents in that simulation,  $w_t$  and  $w_{t-1}$  are the animals named at time  $t$  and  $t-1$  respectively,  $s_{w_t}$  and  $s_{w_{t-1}}$  are the positions of animals  $w_t$  and  $w_{t-1}$  in the agent's semantic memory (i.e., the speaker at trial  $t$  for active exploration, the listener at trial  $t$  for passive exploration). The *exploration index* for an agent pair  $p$  is computed as:

$$E_p = \frac{1}{240} \sum_{i=1}^{240} e_i$$

- **Neighborhood density index:** this metric describes agents' tendency to visit denser areas of the semantic space, where more sub-threshold associations are available. For each word  $w$  named in the course of a simulation, we quantify its neighborhood density in the agent's original semantic space by computing the number of words whose distance from  $w$  is lower than the stopping threshold  $t$ . For each simulation, we compute neighborhood density indices relative to the speaker's semantic space (*speaker density index*) and the listener's semantic space (*listener's density index*). These metrics quantify, respectively, the extent to which agents remain within dense regions of its semantic space, and whether they tend to do so driven by their own behavior, or by the partner's behavior. Formally, the density index for a given simulation can be defined as:

$$d_i = \frac{1}{T} \sum_{t=1}^T |W_t|,$$

$$\text{where } W_t := \{w : \|v_{w_t} - v_w\| \leq k\}$$

Here,  $T$  is the fluency on the simulation,  $W_t$  is the set of words whose Euclidean distance from the animal named at time  $t$  is lower than the set distance threshold  $k$ , and  $|W_t|$  is its cardinality. Note that  $v_{w_t}$  and  $v_w$  are drawn from the speaker's semantic memory for the *speaker density index*, and from the listener's semantic memory for the *listener*

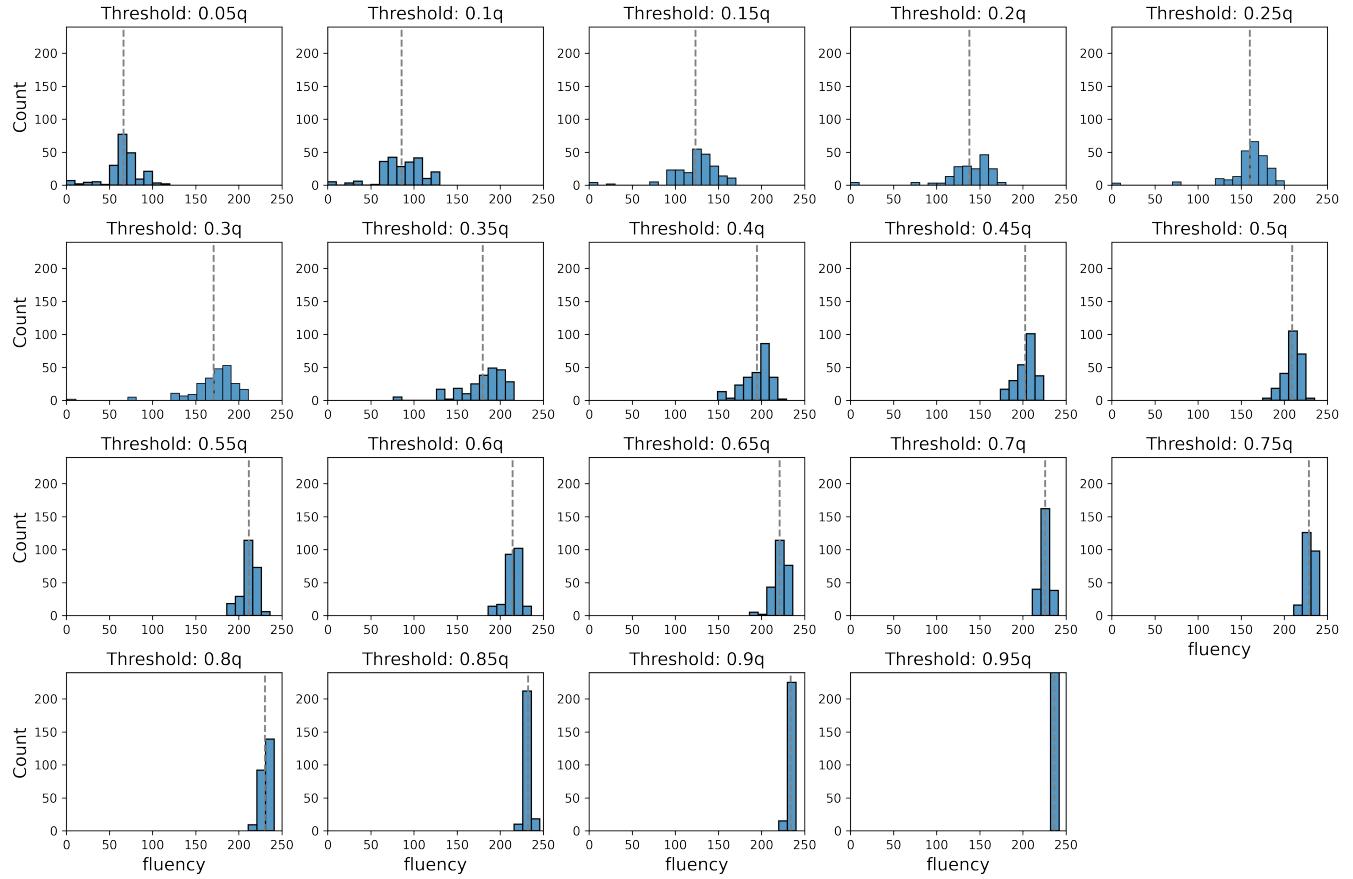


Figure 3: Performance (number of words named before a stopping condition is met) for a baseline agent with the original word2vec space as semantic memory for different stopping thresholds. Each panel represents the distribution of performance over 240 simulations (each with a different initial prompt) for multiple thresholds, defined as incremental quantiles (.5 increment) in the distribution of pairwise distances between words in the semantic space. We selected the .15 quantile to leave sufficient room for variation induced by experimental manipulations and eliminate the potential for both floor and ceiling effects.

*density index*. The neighborhood density index for a given agent pair  $p$  is computed as:

$$d_p = \frac{1}{240} \sum_{i=1}^{240} d_i$$

We hypothesize that diversity will induce changes in the agents' tendency to engage in exploratory search, thus avoiding overexploitation, and in agents' tendency to end up in sparse regions of their semantic memories (as quantified by the neighborhood density index). By inducing variation in these two factors, diversity may result in differences in overall fluency.

## References

- Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., et al. (2020). Scipy 1.0: Fundamental algorithms for scientific computing in python. *Nature methods*, 17(3), 261–272.

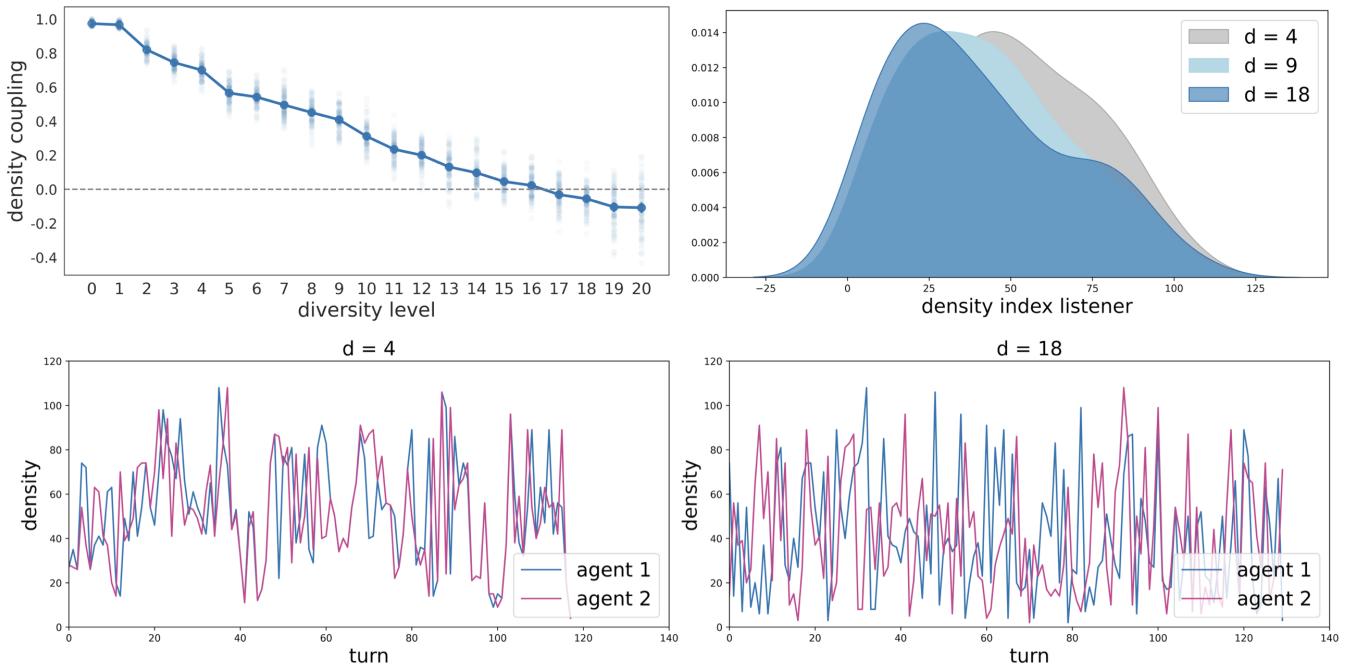


Figure 4: Top left: average correlation between response neighborhood density for listener and speaker for each pair and diversity level in Experiment 1. Response neighborhood densities become anticorrelated after diversity level 16. Top right: distribution of listener neighborhood density values for a sample trial with the same initial seed word across three diversity levels in Experiment 1. The distribution shifts towards lower values, indicating increased tendency to visit sparse neighborhoods. Bottom left: response neighborhood density over time for a sample trial at diversity level 4 in Experiment 1. Bottom right: response density over time for a randomly sampled pair at diversity level 18 in Experiment 1. The differences illustrate the decoupling between density values for listener and speaker emerging at high levels of diversity.

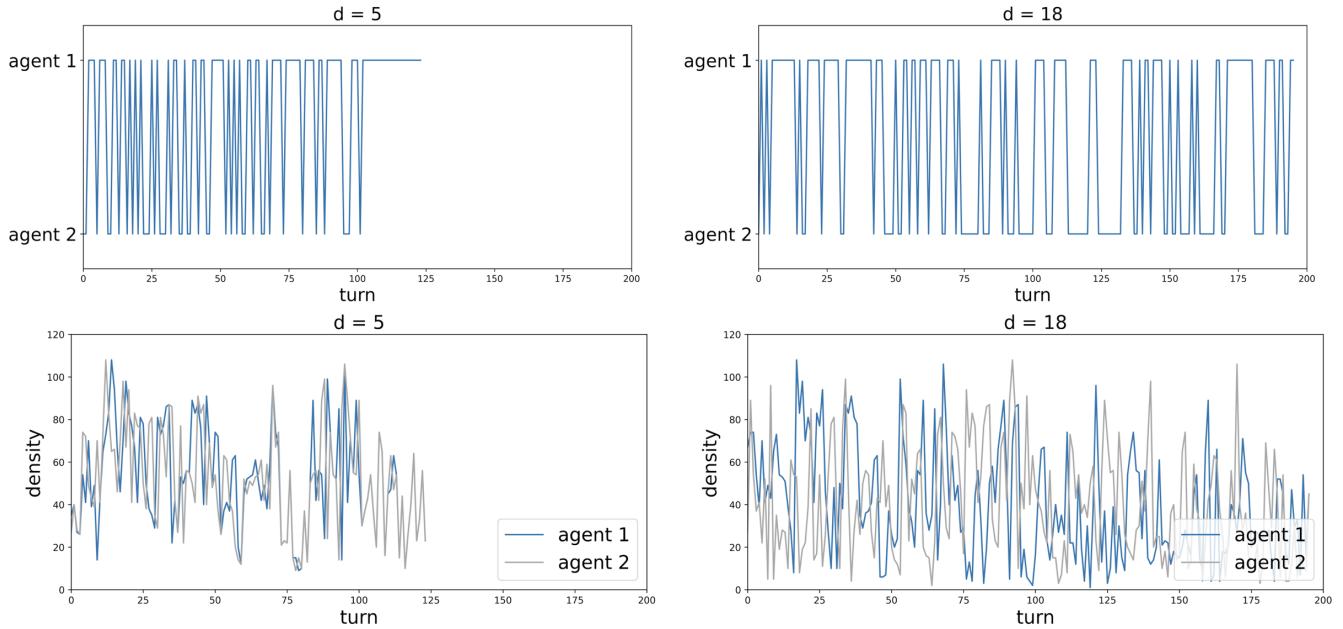


Figure 5: Turn holder (top) and neighborhood density for the last named animal (bottom) for a sample simulation in the competitive turn-taking condition from Experiment 2. Figures display data for a low-diversity pair (left) and a high-diversity pair (right). In the competitive condition, at high levels of diversity, agents tend to hold the floor for several turns (top) and exploit local neighborhoods that are dense for them, but sparse for the partner (bottom). This behavior allows agents to optimally exploit differences in their semantic memories without encountering cognitive fixation.