Supplemental material

Methods

Recall model

Our recall model is presented in more details in Romani et al. 2013, Katkov et al. 2017. In this contribution we considered a simplified version of the model, where we approximate the matrix of overlaps between random sparse memory representations by a random symmetric L by L similarity matrix (SM) with otherwise independently distributed elements, where L is a number of words in the list. Neglecting the correlations between SM elements is justified in the limit of very sparse encoding of memory items (see Romani et al. 2013). A new matrix is constructed for each recall trial. The sequence $\{k_1, k_2, \ldots, k_r\}$ of recalled items is defined as follows. Item k_1 is chosen randomly among all L presented items with equal probability. When n items are recalled, the next recalled item k_{n+1} is the one that has the maximal similarity with the currently recalled item k_n, excluding the item that was recalled just before the current one, k_{n-1}. After the same transition between two items is experienced for the second time, the recall is terminated since the model enters into a cycle.

Solution of the recall model

The symmetry of SM appears to be a minor difference from the much simpler model of fully random asymmetric SM presented in the main text, but in fact it significantly impacts the statistics of the transitions in the corresponding graphs as we will show below.

If retrieval always proceeds from an item to its most similar, as in the asymmetric case, the dynamics will quickly converge to a two-items loop. The reason is that if item B is most similar to item A, then item A will be most similar to item B with a probability of approximately 0.5. We hence let the system choose the second most similar item if the most similar one has just been retrieved, as explained in the main text. When reaching an already visited item, retrieval can either repeat the original trajectory (resulting in a loop) or continue backward along the already visited items and then open a new sub-trajectory (see Fig. 1b). Here we show how to calculate the probability of returning from a new item to any one of already visited items and the probability that the retrieval proceeds along the previous trajectory in the opposite direction upon the return.

In order to return back from item k to item n, the n^{th} element of the k^{th} row of SM, S_{kn}, has to be the largest of the remaining $L-2$ elements in the k^{th} row (excluding the diagonal and the element corresponding to the item visited just before the k^{th} one). The probability for this would be $\approx \frac{1}{L}$ for an asymmetric matrix. For a symmetric matrix ($S_{nk} = S_{kn}$), we have an additional constraint that the element S_{kn} is *not* the largest in the n^{th} row of S, since we require that the k^{th} item was *not* retrieved after the first retrieval of the n^{th} one. The probability of return is then equal to

$$P\left(S_{kn} = max\left(\vec{S_k}\right) | S_{kn} < max\left(\vec{S_n}\right)\right) \approx \frac{1}{2L} \tag{S1}$$

where $\vec{S_k}$ denotes the vector of relevant elements in the k^{th} row of matrix S. The return probability is therefore reduced by a factor of two due to the symmetric nature of SM but retains the same scaling with L as in the model with asymmetric SM. After the first return to an item n (= 10 in Fig. 1b of the main paper), the trajectory may either begin to cycle, or turn towards previously visited items but in the opposite direction if the original transition from this item ($10 \to 7$ in Fig. 1b) was along the second largest element of $\vec{S_n}$. The marginal probability for this is $\frac{1}{2}$, but we must impose the constraint that the k^{th} item was *not* retrieved after the first retrieval of the n^{th} one. If the item preceding n is j (14 in Fig. 1b), the corresponding probability is given by

$$P\left(max\left(\vec{S_n}\right) < max\left(\vec{S_j}\right) | max\left(\vec{S_n}\right) > max\left(\vec{S_k}\right)\right) \approx \frac{1}{3}\,, \tag{S2}$$

which follows from the observation that any ordering for the maximal elements of three vectors of equal size is equally probable. From this result, we conclude that the average number of sub-trajectories during the retrieval process is $\frac{3}{2}$. All together the chance for the process to enter a cycle after each new item retrieved is $\frac{1}{2L}\frac{2}{3} = \frac{1}{3L}$ and hence the average number of items recalled is estimated by replacing L with $3L$ in the corresponding expression for RC in the model with fully random asymmetric SM, Eq. (2) of the main text:

$$\begin{aligned} RC &= k \cdot \sqrt{L} \\ k &\approx \sqrt{3\pi/2} \approx 2.17 \end{aligned} \tag{S3}$$

Participants, Stimuli and Procedure

Experiments and data analysis were performed at the Weizmann Institute of Science. Ethics approval was obtained by the IRB (Institutional Review Board) of the Weizmann Institute of Science. Each participant accepted an informed consent form before participation and was receiving from 50 to 85 cents for approximately $5 - 25$ min, depending on the task. In total 1039 participants, were recruited to perform memory experiments on the Amazon Mechanical Turk® (https://www.mturk.com). All participants were previously selected in the lab Prof. Mike Kahana from the University of Pennsylvania (private communication). Presented lists were composed of non-repeating words randomly selected from a pool of 751 words produced by selecting English words (Healey et al. 2014) and then maintaining only the words with a frequency per million greater than 10 (Medler and Binder 2005). The stimuli were presented on the standard Amazon Mechanical Turk® web page for Human Intelligent Task. Each trial was initiated by the participant by pressing “Start Experiment” button on computer screen. List presentation followed 300 ms of white frame. Each word was shown within a frame with black font for 500 or 1000 ms (depending on presentation rate) followed by empty frame for 500 ms. After the last word in the list, there was a 1000 ms delay before participant performed the task. The set of list lengths was: 8, 16, 32, 64, 128, 256 and 512 words. We also performed the same experiments using a set of 325 short sentences expressing well-know facts, such as ‘Earth is round’ or ‘Italians eat pizza’, etc. We repeated our experiments with random lists of $8, 16, 32, 64$, and 128 such sentences, each presented for 2500 ms followed by empty frame for 1000 ms. Each participant performed experiment A (free recall) and Experiment B (recognition) with lists of the same length. In more details

- 348 participants performed the two experiments with presentation rate of 1.5 sec/word: 265 participants did both experiments for only one list length, 54 for two list lengths, 18, 9 and 2 for 3, 4 and 5 list lengths respectively.
- 375 participants performed the two experiments with presentation rate of 1 sec/word: 373 participants did both experiments for only one list length, 2 for two list lengths.
- 331 participants performed the two experiments with presentation rate of 3.5 sec/fact: 328 participants did both experiments for only one list length, 3 for two list lengths. 15 participants performed also experiments with the words.

Experiment A - Free recall. Participants were instructed to attend closely to the stimuli in preparation for the recalling memory test. After presentation and after clicking a “Start Recall” button, participants were requested to type in as many items (words/sentences) as they could in any order. After the finishing the typing (following non-character input) the information was erased from the screen, such that participants were seeing only the currently typed item. Only one trial was performed by each participant. The time for recalling depended on the length of the learning set, from 1 minute and 30 seconds up to 10 minute and 30 seconds, with a 1 minute and 30 seconds increase for every length doubling. The obvious misspelling errors were corrected. Repetitions and the intrusions (items that were not in the presented list) were ignored during analysis.

Experiment B - Recognition task. In recognition trial, after presentation and after clicking a “Start Recognition” button, participants were shown 2 items, one on top of another. One item was randomly selected among just presented in the list (target), and another one was selected from the rest of the pool of words or sentences. The vertical placement of the target was random. Participants were requested to click on the items they think was presented to them during the trial. Each list was followed with 5 recognition trials per participant, but only the first trial was considered in the analysis. Time for all trials was limited to 45 min, but in practice each response usually took less than two seconds.

Analysis of the results

The average number of recalled items (RC) for each list length and its standard error were obtained from the distribution of the number of recalled items across participants.

In the case of sentences, additional problem that arose concerned the different possible phrasing of the same facts. For example, if a fact presented was ‘Italians like pizza’ and a participant reported ‘Pizza is loved by Italians’, we had to find a way to identify it as correctly recalled. To this end, we used word2vec software developed by Google (Mikolov et al. 2013). Word2vec is a group of related models that are used to produce multidimensional word embeddings. These models are shallow, two-layer neural networks that are trained to reconstruct linguistic contexts of words. Word2vec takes as its input a large corpus of text and produces a vector space, typically of several hundred dimensions, with each unique word in the corpus being assigned a corresponding vector in the space. Word vectors are positioned in the vector space such that words that share common contexts in the corpus are located in close proximity to one another in the space. We used word2vec to compare the sentences reported by participants to the ones presented. A sentence vector was computed as the average of all the word vectors in the sentence, and the similarity between any two sentences with vectors S_1 and S_2 was defined via the cosine of the angle between them, as $1 - \cos(S_1, S_2)$. If the similarity was greater than 0.9 the recall was considered to be correct (this threshold was confirmed by manual inspections of multiple cases). The cases in which the similarity was between 0.8 and 0.9 were checked manually.

The average number of items that remain in memory after presentation of a list of length L was computed from the results of recognition experiments as in (Standing 1973). Suppose that M out of L items are remembered on average after an exposure to the list, the rest are lost. The chance that one of the remembered items is presented during a recognition trial is then M/L, while the chance that a lost word is presented is $1 - M/L$. We assume that in the first case, a participant correctly points to a target item, while in the second case, she/he is guessing. The fraction of correct responses c can then be computed as

$$c = \frac{M}{L} + \frac{1}{2} \cdot \left(1 - \frac{M}{L}\right) . \tag{S4}$$

Hence the average number of remembered items can be computed as

$$M = L \cdot (2c - 1) . \tag{S5}$$

The above analysis is based on a simplified assumption that after the list is presented, each word either remains in memory or completely erased. Our experiments cannot shed light on what really happens to words that are not recognized as familiar after the presentation of the list – they could have never been encoded in memory in the first place, encoded but then erased completely or alternatively degraded to the degree that prevents recognition but not fully erased. Different psychological studies accept different assumptions, e.g. Standing 1973 paper talks about ‘words that remain in memory’, indicating all-or-none processes, while SAM model assumes that all presented words remain in memory but at different degrees. In order to test our conjecture that dependence of recall on presentation speed can be explained away by the different number of words that are candidates for recall, we assumed that the latter can be reliably estimated by our recognition experiments, i.e. that words that could not be recognized could also not be recalled, which may or may not necessarily imply all-or-none encoding/forgetting. We believe that our experimental results a posteriori support this conjecture. In particular, if our estimates for the number of words that remain recall candidates after the whole list is presented (M) was not good, or meaningless, there would be no reason for our RC vs M data to collapse to practically the same curve for two presentation rates, given that both RC and M are dramatically different for these two conditions.

In order to estimate a standard error of the mean for the number of remembered items across participants, for each list length, we performed a bootstrap procedure (Efron and Tibshirani 1994). We generated multiple bootstrap samples by randomly sampling a list of N participants with replacement N times. Each bootstrap sample differs from the original list in that some participants are included several times while others are missing. For each bootstrap sample b out of total number B, with $B = 500$, we compute the estimate for the average number of remembered items, $M(b)$, according to Eq. (S5). The standard error of M is then calculated as a sample standard deviation of B values of $M(b)$:

$$se_B = \sqrt{\sum_{b=1}^{B} \frac{\left(M(b) - \bar{M}\right)^2}{B - 1}} , \tag{S6}$$

where $\bar{M} = \sum_{b=1}^{B} \frac{M(b)}{B}$.

Effects of short-term memory on free recall

It is well known from the previous literature that short-term memory (STM) plays a certain role in free recall, as in particular reflected in pronounced recency effects, i.e. increased probability to recall one of the words in the end of a list that are observed if recall begins immediately after presentation but not if it is delayed (see e.g. Glanzer [1966]). To evaluate the influence of STM in our data, we computed the serial position curves (Murdock Jr [1962]) and observed increased probability of recall at the end of the list, especially for longer lists, mostly for the last one to three serial positions (Fig. S2). The overall effect of STM on recall was quite small for all conditions, less then one extra word recalled.

Additional figures

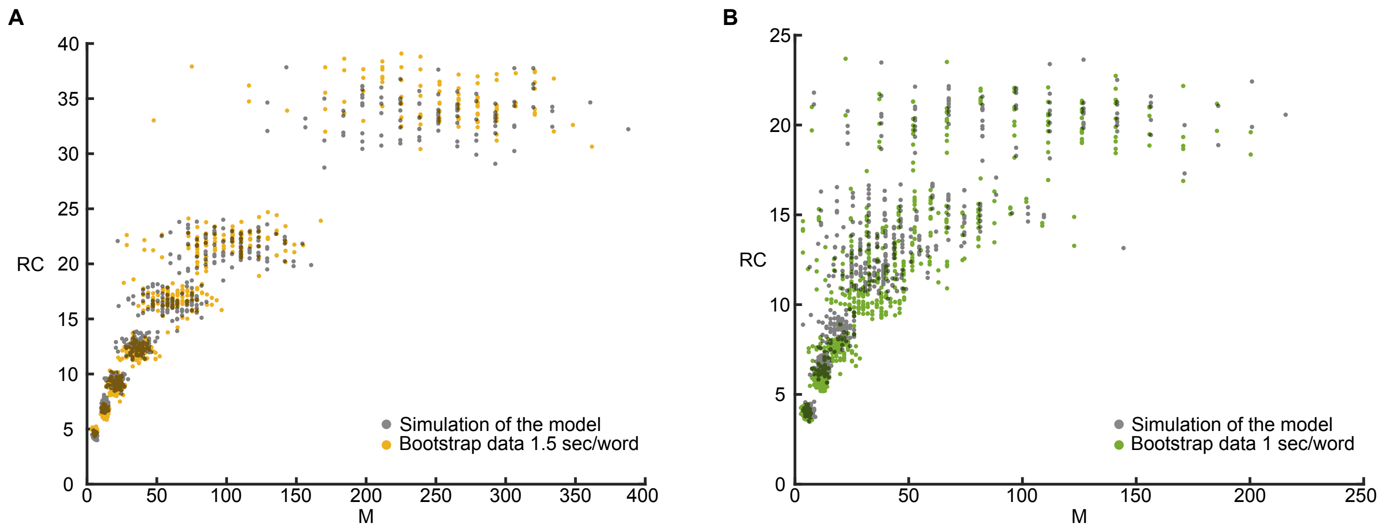

Figure S1. Bootstrap analysis and comparison to model simulations.
(**A**) 1.5 seconds per word presentation rate; (**B**) 1 second per word presentation rate.
100 bootstrap samples for each list length are shown with colored dots with coordinates $M(b)$ and $RC(b)$, where $RC(b)$ is an average number of recalled words computed for each bootstrap sample b. Black dots show corresponding simulation results, obtained as follows. From the results of recognition experiment, we calculate, for each list length L, the fraction of correct recognitions across the participants, c, and therefore the probability $p = (2c - 1)$ that a presented word is remembered. With these two numbers, we simulate multiple recognition and recall experiments. For recognition experiment, we draw a binomial random variable with probability c for each participant independently, simulating their recognition answers, from which we compute the number of remembered words averaged for all participants as explained in the Methods. We then drew L binomial variables with probability p for each participant, simulating the number of remembered of words by this participant during the recall experiment. With the number of remembered words known for each participant, we run the recall model (see Methods) to obtain the average recall performance over participants. Every simulation described above produced 7 pairs of results (M, RC), one per list length. We repeated the whole procedure 100 times, same as the number of bootstrap samples.

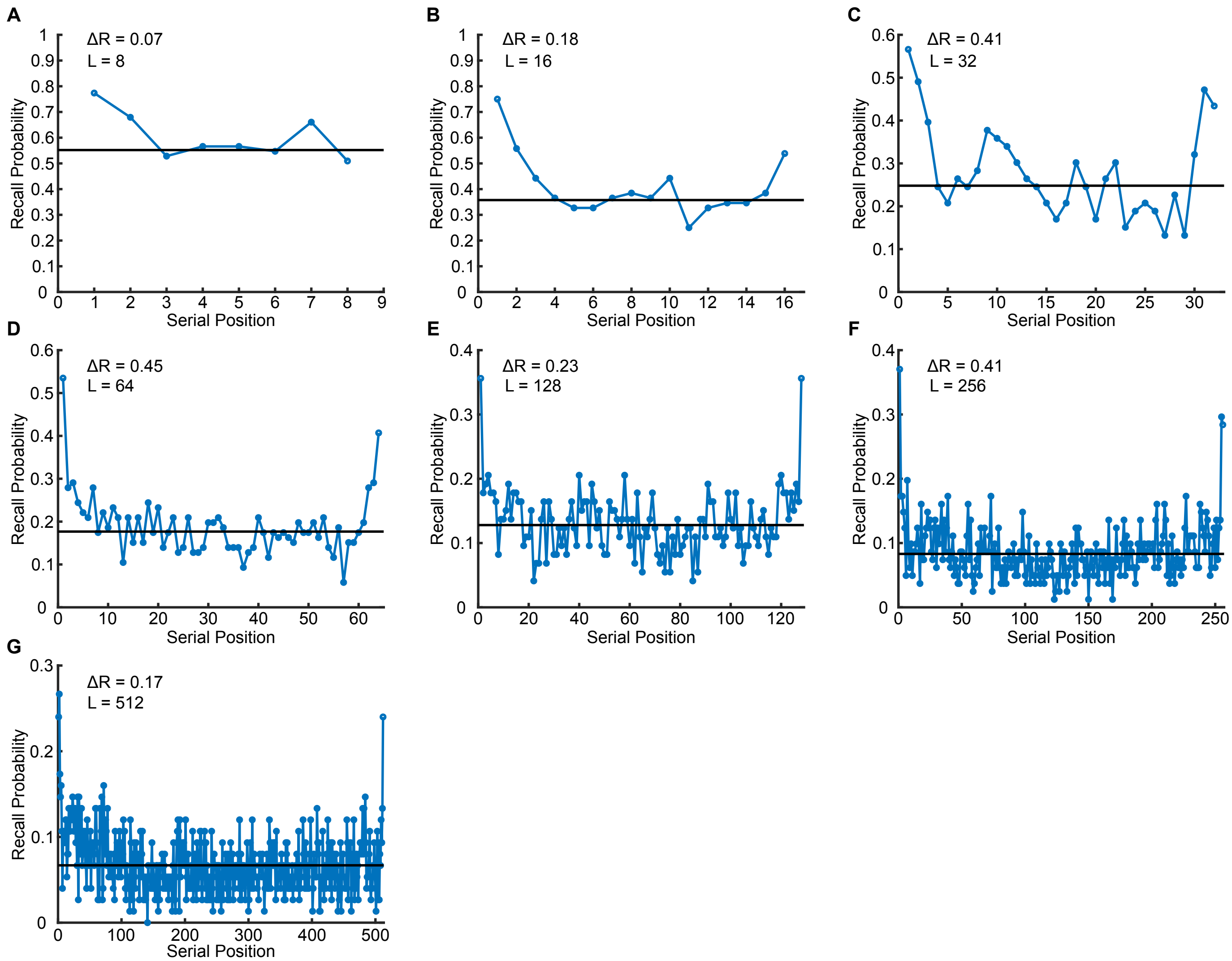

Figure S2. Effects of short-term memory on free recall.
Probability to recall a word as a function for its serial position in the presented list, for presentation rate of 1.5 sec per word. Black horizontal curve illustrates the average recall probability that is computing by excluding first 3 and last 2 words in the list. The additional number of words recalled (ΔRC) is computed by summing the excess recall probability for 1 to 3 words in the end of the list that are significantly better recalled than the rest.

References

Bradley Efron and Robert J Tibshirani. *An introduction to the bootstrap.* CRC press, 1994.

A. R Glanzer, M andCunitz. Two storage mechanisms in free recall. *Journal of Verbal Learning & Verbal Behavior*, 5(4): 351–360, 1966.

M Karl Healey, Patrick Crutchley, and Michael J Kahana. Individual differences in memory search and their relation to intelligence. *Journal of Experimental Psychology: General*, 143(4):1553, 2014.

M Katkov, S Romani, and M Tsodyks. Memory retrieval from first principles. *Neuron*, 94(5):1027–1032, 2017.

DA Medler and JR Binder. Mcword: An on-line orthographic database of the english language, 2005.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

Bennet B Murdock Jr. The serial position effect of free recall. *Journal of experimental psychology*, 64(5):482, 1962.

Sandro Romani, Itai Pinkoviezky, Alon Rubin, and Misha Tsodyks. Scaling laws of associative memory retrieval. *Neural computation*, 25(10):2523–2544, 2013.

Lionel Standing. Learning 10000 pictures. *The Quarterly journal of experimental psychology*, 25(2):207–222, 1973.