Memory structure created through behavioral time scale synaptic plasticity

Yujie Wu and Wolfgang Maass†

Institute of Theoretical Computer Science, Graz University of Technology, Graz, Austria
†Corresponding author. E-mail: yujie.wu@igi.tugraz.at; maass@igi.tugraz.at

ABSTRACT

Recent experimental studies in the awake brain have identified a new rule for synaptic plasticity that appears to be instrumental for the induction of episodic and conjunctive memories in the mammalian brain: behavioral time scale synaptic plasticity (BTSP) Bittner et al. (2015, 2017); Grienberger and Magee (2022). BTSP differs in essential aspects from previously studied rules for synaptic plasticity. But there is so far no theory that enables a principled understanding of the impact of BTSP on the structure of the associative memory that it induces. We extract fundamental mathematical principles from experimental data on BTSP that elucidate the capacity and associative recall capabilities of memory structures that it creates. Predictions of the resulting theory are corroborated by large-scale numerical experiments. In particular, we show that BTSP can create well-separated memory traces for a very large number of memory items, even if these are not orthogonal. Furthermore, BTSP induces the repulsion effect, a well-known fingerprint of memory organization in the human brain, that could not be explained by preceding types of synaptic plasticity.

Introduction

BTSP (behavioral time scale synaptic plasticity) is a plasticity mechanism of pyramidal cells in area CA1 that was discovered and analyzed through experiments in awake and behaving rodents Bittner et al. (2017); Grienberger and Magee (2022); Priestley et al. (2022). Converging experimental data suggest that area CA1 of the hippocampus is the site where episodic memories and conjunctive memories are formed, BTSP is likely to have a strong impact on the structure of associative memory that it induces. There exists already a substantial body of theoretical work on the structure of associative memory in the brain. But these theories rely on plasticity rules such as Hebbian plasticity or Spike Timing Dependent Plasticity (STDP) that strongly differ from BTSP. Furthermore, they provide little guidance for understanding the recall of episodic memories or conjunctive memories from a partial cue with just a small fraction of the information contained in a stored memory item. In addition, they focus primarily on the assumption that the network inputs for different memories are orthogonal, which makes it difficult to understand the structure of episodic or conjunctive memories that contain common components. Therefore we do not constrain our analysis on memory structures induced by BTSP on orthogonal memory items, and we study also memory completion for partial cues where a substantial fraction of the relevant information is missing.

BTSP differs in the following essential aspects from traditionally studied plasticity rules for the induction of memory traces:

- BTSP does not depend on the firing of the postsynaptic neuron.
- BTSP depends on gating input from another area, the entorhinal cortex (EC), that opens in pyramidal cells of area CA1 through the induction of dendritic plateau potential a plasticity window that lasts for several seconds

- This gating input from EC appears to have dominant stochastic features Grienberger and Magee (2022).
- The amplitude of synaptic weight changes depends primarily on the preceding value of this weight Milstein et al. (2021)
- BTSP induces substantial synaptic plasticity already in a single or very few trials.
- BTSP induces plasticity both in synapses from neurons in area CA3 that fire several seconds before and those that fire several seconds after the initiation of the plateau potential in the CA1 neuron.

The latter feature gave rise to the name of this plasticity rule since the window of plasticity that is induced by a plateau potential in CA1 neurons acts on the time scale of behavior (experiences).

Quantitative models for the impact of BTSP and a model for its impact on the induction of place cells have been provided by Milstein et al. (2021). Building on this work we first derive a version of the BTSP rule that is sufficiently simple for supporting theoretical analysis while simultaneously capturing the main features of more complex BTSP rules. We then investigate properties of the memory structure that it creates, both through theory and numerical experiments. We address in particular recall from partial cues and storage of non-orthogonal memory items. The theory that we develop turns out to predict the outcomes of numerical experiments very well at scales that are still accessible for simulations. This provides a justification for also applying the theory for deriving memory capacity estimates for the scale of the human brain, which we cannot address in numerical experiments. Finally, we show that the BTSP model reproduces the repulsion effect discovered by the hippocampus activity of human brain for closely related memory items Chanales et al. (2017); Wanjia et al. (2021); Zheng et al. (2021). This effect could not be explained by previously considered rules for synaptic plasticity.

Hopfield networks, see e.g. Krotov and Hopfield (2020); Sharma et al. (2022) for recent analysis and review, are arguably the most commonly studied neural network models for storing and retrieving memories. In contrast, we investigate memory encoding in a single layer of synapses in a feedforward network: A model for memory encoding in feedforward synaptic connections from area CA3 to area CA1. Nevertheless, because of the prominence of Hopfield networks as neural network models for memory, we compare the capacity and retrieval capabilities of the memory structure that is created by BTSP with that of Hopfield networks, in spite of the completely different architecture of these two types of neural networks.

The stochasticity of plateau potentials of BTSP introduces stochasticity into the imprinting of memories into weights from CA3 to CA1. Stochasticity is not an integral part of common memory models such as Hopfield networks. But the impact of stochasticity on memory encoding had already been analyzed in Amit and Fusi (1994). However, they only analyzed memory encoding in the weights of a recurrent neural network. Their analysis was also based on the assumption that the patterns that are to be stored are uncorrelated, an assumption that may not be satisfied for the formation of conjunctive and episodic memories in the brain, and on which we do not rely in the theory that we are presenting. Nevertheless, the theory of Amit and Fusi (1994) also arrives at a bound for the memory capacity that scales logarithmically with the network size, the same scaling law that we found for the feedforward networks that we are investigating.

A neural algorithm for the creation of neural codes in a feedforward network has already been studied in Dasgupta et al.Dasgupta et al. (2017). They considered neural codes for sensory inputs (odours) in the fly brain, and identified common features between memories for odours that are created by the biological "fly algorithm" and algorithms from computer science such as locality-sensitive hashing (LSH). BTSP shares this relation with locality-sensitive hashing. However, a cornerstone of the fly algorithm, the projection of odour information into a much higher dimensional space, is not met by the architecture of the mammalian hippocampus: The number of neurons in CA1 is both in the rodent and the human brain by at most by a factor of 1.5 larger than the number of neurons in the areas

CA3 and EC3 that provide synaptic inputs to area CA1. Furthermore, the fly algorithm does not involve any form of synaptic plasticity, whereas BTSP is a rule for synaptic plasticity.

Results

A simple mathematical model captures core properties of BTSP. The window for synaptic plasticity of synaptic inputs from area CA3 to a generic pyramidal neuron in area CA1 is opened according to Bittner et al. (2015, 2017); Grienberger and Magee (2022) by synaptic input from layer 3 of the entorhinal cortex (EC3) that causes long-lasting plateau potentials in the apical dendrites of the CA1 neuron, see Fig. 1A. It opens the BTSP plasticity window for several seconds before and after the onset of a plateau potential (= time 0 in Fig. 1B). A mathematical model for the way how BTSP affects the weight of synapses from CA3 was provided in equation (1) of Milstein et al. (2021). Since this rule is still too complicated to support theoretical analysis of its implications in a network context, we formulate a core version of the BTSP rule that contains two simplifications. First, we simplify the role of time: For each onset of a plateau potential (= time 0) BTSP is applied in the same way to all synapses to the CA1 neuron that are activated within the time interval [-3, 2s], but not to synapses that were activated before or after that time window (unless they coincided with another plateau potential, see Fig. 1B). The more complex version of the BTSP rule in Milstein et al. (2021) (see their Fig. 3I) shows in addition differences in weight changes in dependence of the time difference between the activity of the presynaptic neuron and the initiation of a plateau potential in the postsynaptic neuron. Whereas we have a lot of information about these time differences for the formation of place cells in CA1, there are apparently no data on the distribution of these time differences for the case of more general forms of episodic and conjunctive memories, especially not for the human brain. Therefore we consider for each initial weight value a weight update that aims at modelling the average update for a uniform distribution of these time differences in Fig. 3I of Milstein et al. (2021) .

Secondly, one can reduce BTSP to a rule with just two weight values 0 and 1. According to the data and model of Milstein et al. (2021) the direction and amount of the change of a synaptic weight through BTSP depends on the preceding value of the weight Fig. 1C: It goes down for relatively large weights, and by a larger amount if the weight was previously very large. Dually, weights are increased if they were previously small. This continuous rule can also be reproduced by an array of multiple synaptic connections between the pre- and postsynaptic neuron, where the weight of each of them assumes only values 0 and 1, see Fig. 1D.

With these two simplifications, the way how BTSP changes the weight w of a synapse for binary-valued inputs x and a binary-valued plateau signal y can be described by

$$\Delta w_t = \begin{cases} 0 & \text{if } y_t = 0, \\ 1 & \text{if } y_t = 1 \text{ and } w_t = 0, \\ -1 & \text{if } y_t = 1 \text{ and } w_t = 1, \end{cases} \tag{1}$$

where the index t denotes the current time step. Note that plasticity only occurs for weights of synapses where the presynaptic neuron fires, i.e., assumes value 1, during the presentation of the input pattern. The relation of this simplified BTSP rule and the more complex version of the rule in Milstein et al. (2021) is discussed in Methods.

Theoretical analysis of memory structures that result from BTSP. We analyze the impact of BTSP on the memory structure that it creates for the case where M input patterns $\boldsymbol{x}$ of length m are sequentially presented to an array of n neurons that are subject to BTSP, see Fig. 2 A. We refer to these n neurons as CA1 neurons, since BTSP has been demonstrated for pyramidal cells in area CA1. The input patterns $\boldsymbol{x}$ can be viewed as activation patterns of CA3 neurons that project to these CA1 neurons through sparse synaptic connections. Note that we use boldface symbols for vectors and matrices in order to distinguish them from single variables.

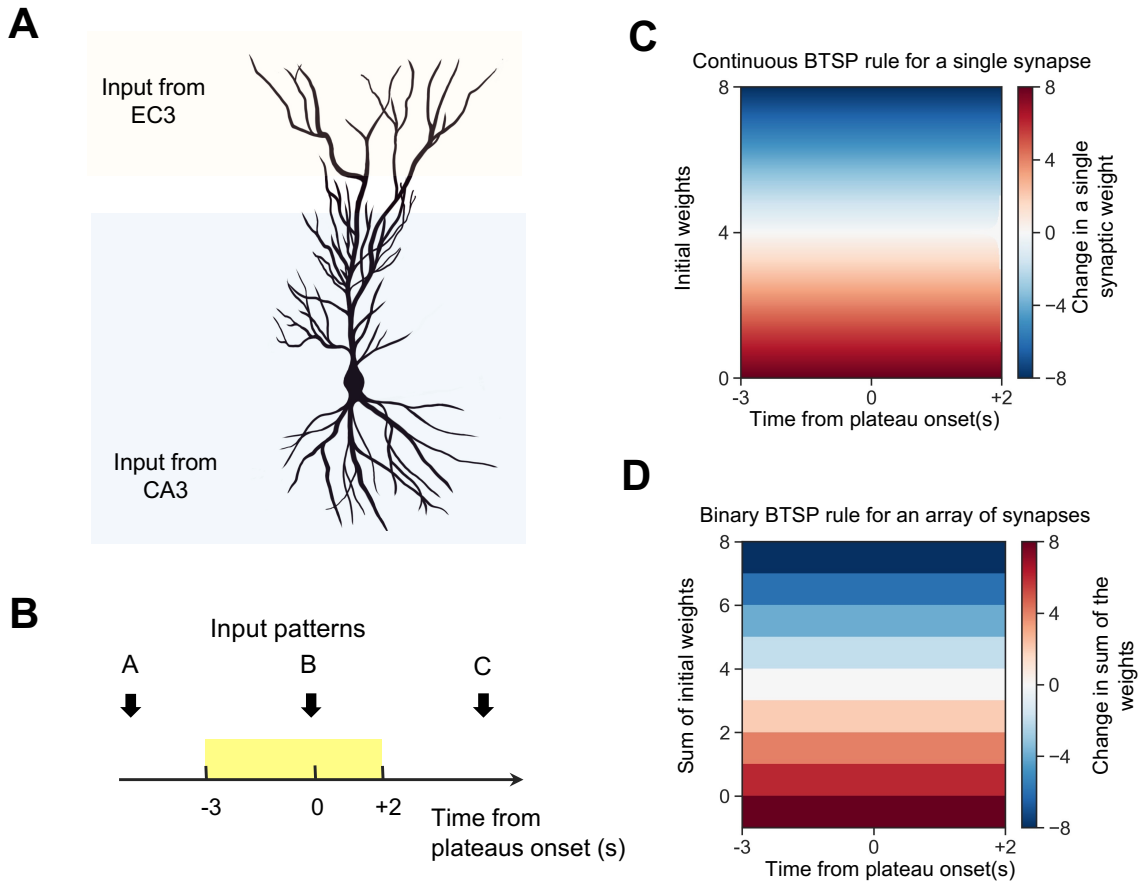

Figure 1. Illustration of BTSP. (**A**)A CA1 pyramidal cell gets synaptic inputs from two different sources: One stream of synaptic inputs comes from layer 3 of the entorhinal cortex (EC3) that arrives at distal synapses in apical dendrites and induces plateau potentials according to Grienberger and Magee (2022). Another stream of synaptic inputs comes from area CA3 and arrives at more proximal synapses, see Fig.5-1Andersen et al. (2007). (**B**) Symbolic drawing of the most important timing feature of BTSP. According to the experimental data reported in Fig. 3 I of Milstein et al. (2021) only input patterns that arrive up to 3s before or at most 2s after the initiation of a plateau potential in the CA1 neuron trigger synaptic plasticity through BTSP. Hence BTSP would apply in this drawing to pattern B, but not to patterns A or C. (**C**) Heatmap of the continuous weight change through BTSP in dependence of the preceding value of the weight, idealized from experimental data of Milstein et al. (2021), for input patterns that arrive within the interval from -3s to 2s relative to the onset of a plateau potential. (**D**) Corresponding experiment for the case that the two neurons are connected by 8 synapses with binary synaptic weights, each subject to the simplified version of the BTSP rule according to Eq. 1 with plateau potentials that gate plasticity for all of these synapses. One sees that the sum of their synaptic weights changes in the same way as the weight of a single synapse with continuous values in panel C. In the simulation, we used random initial binary weights sampled from the Bernoulli distribution with a probability of 0.5. We ran the model for 10,000 trials and plotted the average change of weights according to the sum of initial weight values.

We consider a learning phase where each of the M patterns $\boldsymbol{x}$ is presented once to the network (one-shot learning). To model the experimentally found scenario where for each input pattern a plateau potential is generated in a subset of CA1 neurons Grienberger and Magee (2022), we adopt a corresponding convention for our model: A plateau potential is generated independently in each CA1 neuron with some probability f_q, thereby setting the variable y_t in equation (1) to 1 at the time t of the presentation of the input pattern $\boldsymbol{x} = \boldsymbol{x}_t$.

In the recall phase (see Fig. 2 B) we assume that BTSP is no longer active and some cue $\boldsymbol{x}$ is presented to the network in the same way as input patterns during learning, representing firing patterns in area CA3. Each CA1 neuron is modelled as a McCulloch-Pitts neuron (= threshold gates), that compares the weighted sum of its inputs with a threshold, and outputs 1 (i.e., "fires") when the weighted sum is at least as large as the firing threshold. We denote the set of CA1 neurons that fire in response to an input pattern (cue) $\boldsymbol{x}$ by $\boldsymbol{z}(\boldsymbol{x})$. This set of neurons models the memory trace that is activated by the cue $\boldsymbol{x}$ in area CA1.

Note that the analyzing the memory structure for the M input patterns generated in this way is nontrivial since its details depend on the order in which the M patterns are presented, due to the weight-dependence of BTSP Milstein et al. (2021): The direction of a weight change depends on the preceding value of this weight. This leads to correlations between weight updates and memory traces that are created for adjacent input patterns in the sequence. However, we are most interested in properties of the resulting memory structure that do not depend on the order of the sequence of input patterns during learning. This goal complicates the theoretical analysis of implications of BTSP, and renders previously developed analytical tools from Hopfield (1982, 2007); Widrich et al. (2020) for the analysis of memory structures that are created by other plasticity rules ineffective.

An additional difference to previous analyses of memory structures arises from the type of memory that is created through BTSP in area CA1: Episodic memory, or more generally conjunctive memory Bittner et al. (2015), where heterogeneous pieces of information are integrated into a memory trace. An inherent expectation of this type of memory is that not just a few percent, but substantial parts of the original input pattern may be missing in the cue that is used for recall. Furthermore, different input patterns that are stored are not necessarily orthogonal because they may contain common pieces of information, such as a place code. In contrast, most previously developed tools for estimating the capacity of memory models assumed that stored inputs are orthogonal, and all -or almost all- bits of the original input patterns are present in a recall cue. To address these challenges, our theory recruits tools from probability theory and combinatorial mathematics. In particular, it allows us to predict what happens when large parts of input patterns are masked, or when the input patterns are not orthogonal. For the sake of simplicity, the theory uses a few approximations for estimating the values of some very complicated terms. Nevertheless, its predictions turn out to be very precise, as confirmed by numerical experiments.

Memory recall with partial cues. Lateral connections between pyramidal cells in area CA1 are very sparse. Hence we do not address in our analysis any additional error correction that could potentially take place through recurrent activity in CA1, and assume that there are no lateral connections between the CA1 neurons that we model. We focus instead on the structure of memory traces $\boldsymbol{z}(\boldsymbol{x})$ of CA1 neurons that fire during the recall phase in response to cues $\boldsymbol{x}$. We use the Hamming distance (HD) as a metric for evaluating the difference between memory traces from the perspective of downstream brain systems that receive synaptic input from CA1. An important reference value is the average HD $HD(z(\boldsymbol{x}_a), z(\boldsymbol{x}_b))$ between the memory traces in CA1 for any two different patterns $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$ that occurred in the sequence of input patterns during the learning phase. But importantly, we measure this HD always during the recall phase, after all input patterns in the sequence have been processed by BTSP, since this HD is prone to continuous changes during the processing of that sequence.

We analyse the response of CA1 neurons to partial cues from the perspective of downstream systems. For those, it is essential that the set $\boldsymbol{z}(\boldsymbol{x})$ of CA1 neurons that fire during recall in response to a partial cue $\boldsymbol{x}'$ is substantially closer to the memory trace $\boldsymbol{z}(\boldsymbol{x})$ for the original input pattern $\boldsymbol{x}$ (measured by HD $HD(\boldsymbol{z}(\boldsymbol{x}), \boldsymbol{z}(\boldsymbol{x}')))$; and their

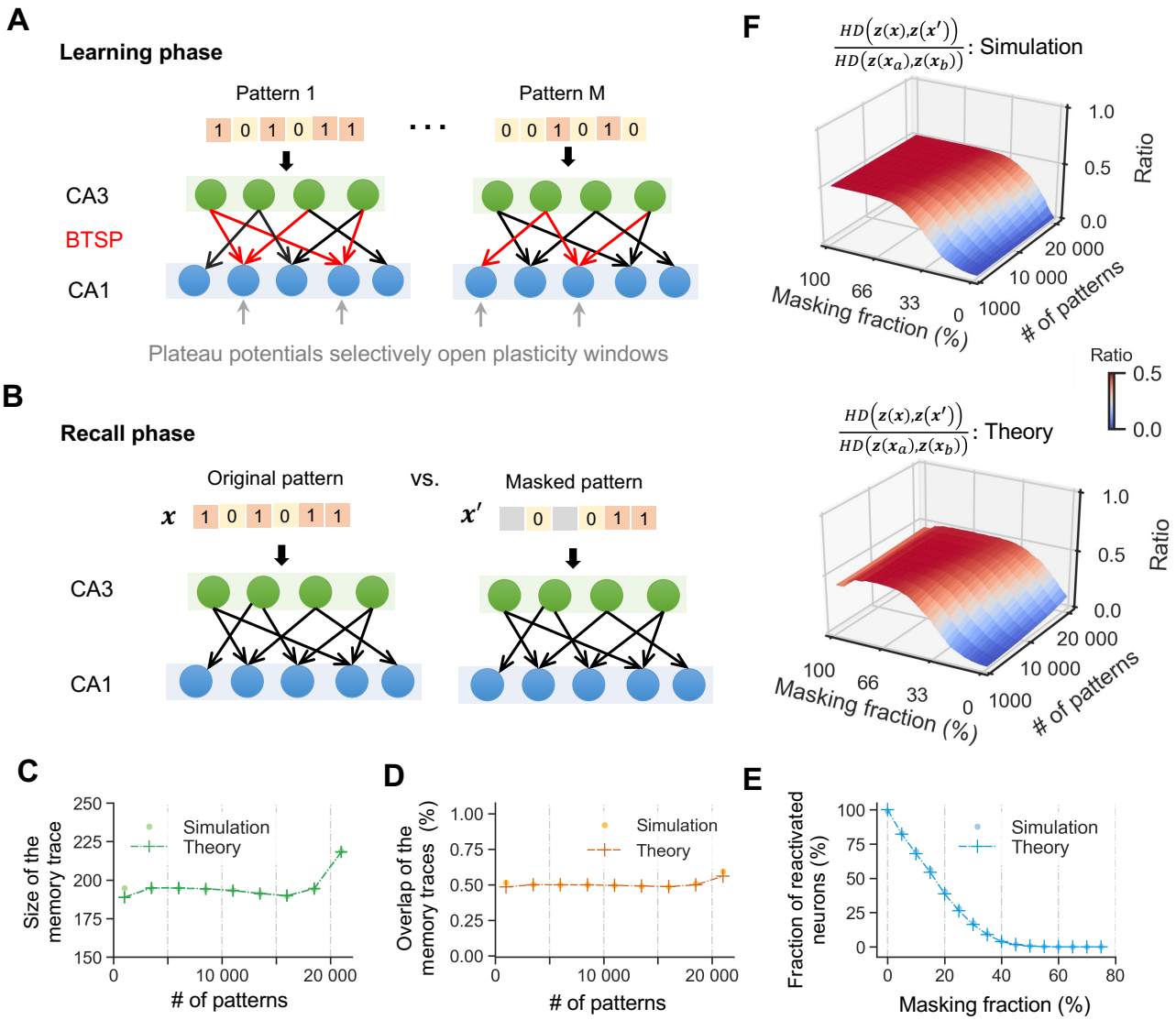

Figure 2. Theoretical analysis and simulation results for recall with partial cues from memory created through BTSP. (**A**) Schematic illustration of the creation of a memory in area CA1 through BTSP. During the learning phase, synaptic connections between neurons in area CA3 and CA1 are subject to BTSP for a sequence of M binary activity patterns in CA3. The activity patterns in area CA3 are usually referred to simply as input patterns in the following. For each input pattern, a random set of CA1 neurons receives a plateau potential, which opens for this input pattern the plasticity window for its synaptic connections from CA3. (**B**) During recall, the activation of CA1 neurons is evaluated for partial cues $\boldsymbol{x}'$ where a random subset of the bits of the original input pattern $\boldsymbol{x}$ that occurred during learning has been masked (i.e., set to 0). (**C**) Average size of a memory trace (= number of CA1 neurons that fire during recall with the full input pattern) as a function of the total number M of input patterns that are learnt. (**D**) Average overlap of these memory traces as a function of the total number M of input patterns that are learnt. (**E**) Fraction of a memory trace that gets reactivated with a partial cue, plotted as a function of the masking fraction. (**F**) Analysis of memory recalls with partial cues after 20,000 input patterns have been learnt through BTSP. The Hamming distance (HD) between the set $\boldsymbol{z}(\boldsymbol{x})$ of CA1 neurons that fire in response to a full input pattern $\boldsymbol{x}$ and the set $\boldsymbol{z}(\boldsymbol{x}')$ of CA1 neurons that fire in response to a partial cue $\boldsymbol{x}'$, divided by the average HD between memory traces for two different input patterns from the set of M patterns. Theoretical results were obtained from Eqs 24, 25 and 26. They agree very well with the results of numerical simulations.

HD should also be smaller than the reference value $HD(z(\boldsymbol{x}_a), z(\boldsymbol{x}_b))$. More precisely, we expect that the ratio $HD(\boldsymbol{z}(\boldsymbol{x}), \boldsymbol{z}(\boldsymbol{x'}))/HD(\boldsymbol{z}(\boldsymbol{x}_a), \boldsymbol{z}(\boldsymbol{x}_b))$ has on average a value well below 1/2 for different inputs patterns $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$ from the learning sequence and partial cues.

In general, this ratio depends not only on the number and type of input patterns that occurred during learning but also on the dimensionality m of input patterns from CA3 and the dimensionality n of memory traces in CA1. Since these structures are too large for computing the ratio on standard computers, we will focus on a model that is downs-scaled by a factor of 1/10 from the size of these structures in the rat brain, and by a factor of 1/100 from the sizes in the human brain, see Table 3.1 of Andersen et al. (2007). We arrive in this way at an input dimension $m = 25,000$ and a dimensionality $n = 39,000$ of memory traces in CA1. We will discuss in a later section the impact of scaling up these values to the actual brain size. CA1 pyramidal cells are covered with about 30,000 dendritic spines according to chapter 5 of Andersen et al. (2007). Hence we assume in our reduced size model that a CA1 neuron receives synaptic inputs from a random set of 15,000 CA3 neurons, yielding a connection probability $f_w = 0.6$. Furthermore, according to Guzman et al. (2016) a fraction $f_p = 0.005$ of CA3 neuron can be estimated to fire during an input pattern, yielding an expected number of $m * f_p = 125$ bits with value 1 in our binary encoding of input patterns that occur during learning. Finally, a plateau potential is generated in a CA1 neuron with a probability around $f_q = 0.005$, suggested by Fig.4g of Grienberger and Magee (2022). We will use these default parameters in the following experiments unless otherwise stated. We used the grid search method to set the firing thresholds of CA1 neurons (see *Methods*).

We show in Fig. 2C the average size of memory traces that are created through BTSP, as a function of the total number M of memory traces that have been induced by BTSP. Fig. 2 D shows the average overlap between two different ones of these memory traces. Fig. 2 E shows the fraction of neurons in a memory trace that gets reactivated by a partial cue, as a function of the masking fraction in the partial cue. The fraction of reactivated CA1 neurons gradually decreases as the masking fraction increases, approaching zero when the masking fraction exceeds 40%. This suggests that additional network structures of the brain, such as CA3, are needed to complement strongly impoverished cues before the query is sent from CA3 to CA1. Fig. 2 F shows that the critical ratio $HD(\boldsymbol{z}(\boldsymbol{x}), \boldsymbol{z}(\boldsymbol{x'}))/HD(\boldsymbol{z}(\boldsymbol{x}_a), \boldsymbol{z}(\boldsymbol{x}_b))$ for evaluating memory recall for partial cues is below 1/2 for any masking fraction, and substantially smaller for masking fractions up to 50 percent, even if the number of patterns that were stored via BTSP in the synaptic weights during the learning phase reaches 20,000. Furthermore, it shows a close match between predictions of the theory for BTSP and numerical experiments.

Comparing the memory structure created through BTSP with that of a classical Hopfield network. Since classical Hopfield networks (HFNs) are arguably the most commonly considered theoretical model for memory organization, we now compare the preceding results from Fig. 2 F with corresponding measurements for a HFN. We constructed the HFN with sparse input patterns by following a recent studySharma et al. (2022). We applied the same connection sparsity $f_w = 0.6$ as for the network where we tested BTSP, and used the same input patterns. For a HFN we write $\boldsymbol{z}(\boldsymbol{x})$ for the set of neurons that are activated by the cue $\boldsymbol{x}$ after 100 synchronous update steps of the HFN after the presentation of the cue.

Fig. 3 C shows that the ratio $HD(\boldsymbol{z}(\boldsymbol{x}), \boldsymbol{z}(\boldsymbol{x'}))/HD(\boldsymbol{z}(\boldsymbol{x}_a), \boldsymbol{z}(\boldsymbol{x}_b))$ for HFNs. For comparison we have replotted in Fig. 3 D the result of Fig. 3 D at the same scale as the one in Fig. 3 C for HFNs. One sees that this ratio has a smaller value for HFNs than for BTSP only for a small number of patterns. But it becomes larger than 1/2, thereby indicating that recall for partial cues is no longer working when the number of patterns becomes larger than 10,000. This result is in line with previous studiesHopfield (1982, 2007); Sharma et al. (2022) of classical HFNs.

Analysis of the capability of BTSP to store non-orthogonal memory items. Episodic and conjunctive memory requires storing patterns that have common parts and are therefore not orthogonal. In contrast, most studies of the

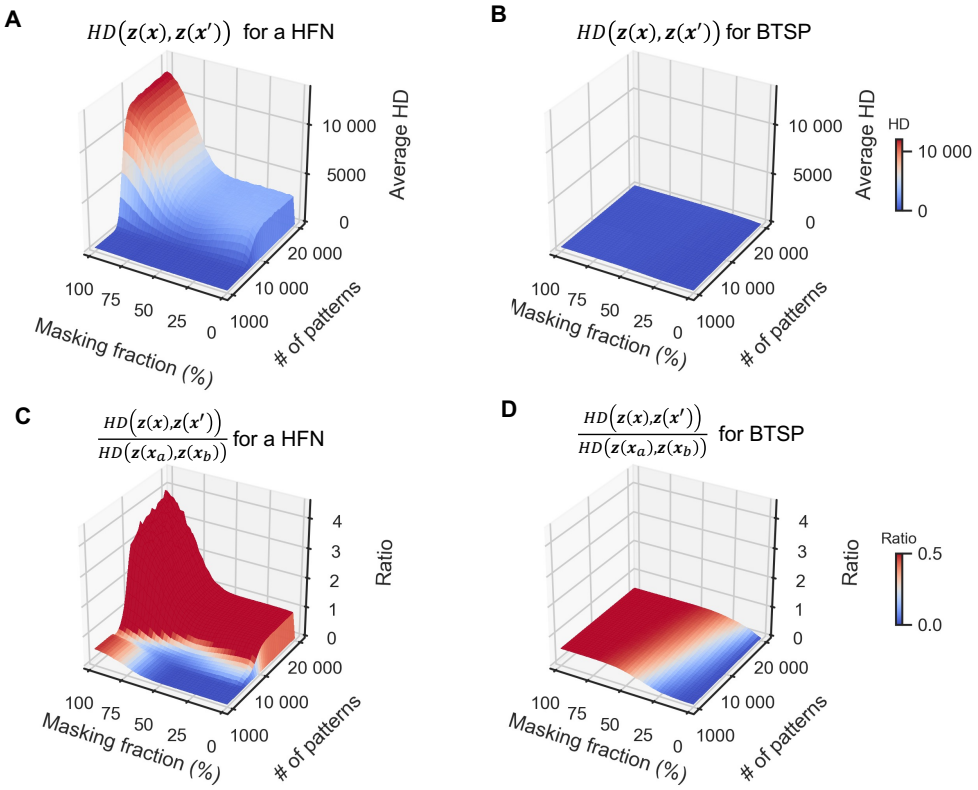

Figure 3. Comparison of recall with partial cues from the memory structure of a classical Hopfield network(HFN) and the memory structure created through BTSP. Panels (**A**) and (**B**) show that the HD between the memory trace that is recalled with a full and partial cue is substantially smaller in the memory structure that is created by BTSP. Panels **C** and **D** also show that the ratio $HD(\boldsymbol{z}(\boldsymbol{x}), \boldsymbol{z}(\boldsymbol{x'}))/HD(\boldsymbol{z}(\boldsymbol{x}_a), \boldsymbol{z}(\boldsymbol{x}_b))$ is in general smaller in that memory structure. Note that panels B and D are adapted from Fig. 2 F at the scale that is needed to show the corresponding results for the HFN. We used for HFNs the same connection probability f_w as for BTSP. All simulation results are averaged over 10 runs. Grid search was used to set the threshold of HFN units and the firing threshold v_{th} for the BTSP models to optimize their performance (see *Methods*).

memory capacity of HFNs only address the storage of orthogonal input patterns. It is shown in the right column of Fig. 4 that our theory predicts that BTSP creates useful memory structures also for non-orthogonal input patterns. This is confirmed by the results of numerical simulations, shown on the left. Note that in these and all preceding analyses an input pattern had on average 125 bits that were on, i.e., had value 1. Hence input patterns with up to 30 common 1's, as considered in Fig. 4, are highly non-orthogonal. Nevertheless, increasing the number of common bits causes only a moderate increase in the ratio $HD(\boldsymbol{z}(\boldsymbol{x}),\boldsymbol{z}(\boldsymbol{x'}))/HD(\boldsymbol{z}(\boldsymbol{x}_a),\boldsymbol{z}(\boldsymbol{x}_b))$ that is essential for evaluating its capability to handle partial cues for recall. We assumed here that always 33% of the pattern was masked during recall.

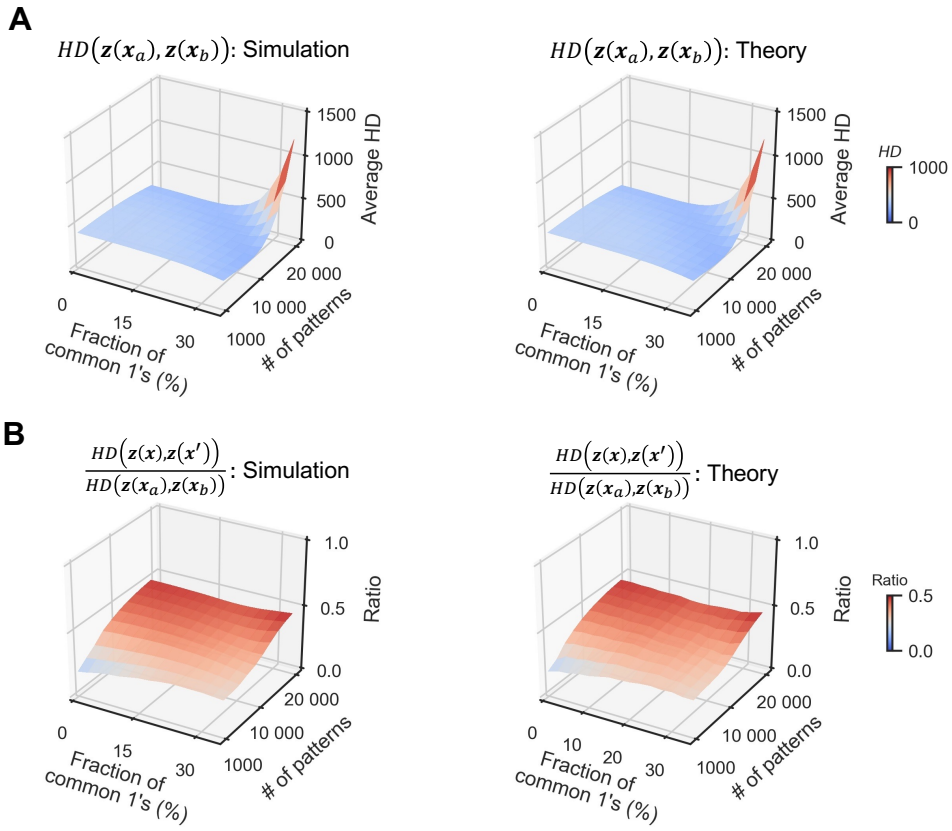

Figure 4. Analysis of the memory structure created by BTSP for non-orthogonal input patterns. (**A**) Evaluation of the effect of common bits on the average distance $HD(\boldsymbol{z}(\boldsymbol{x}_a),\boldsymbol{z}(\boldsymbol{x}_b))$ in the recall for patterns with 33% masking fraction. The fraction of common 1's refers to the average number of common 1's in any two input patterns divided by the average number of 1's in an input pattern. (**B**) Evaluation of the effects of the common bits on the ratio $HD(\boldsymbol{z}(\boldsymbol{x}),\boldsymbol{z}(\boldsymbol{x'}))/HD(\boldsymbol{z}(\boldsymbol{x}_a),\boldsymbol{z}(\boldsymbol{x}_b))$ in the recall for patterns with 33% masking fraction. The ratio stays well below 0.5 if the number of common bits and the total number of patterns are not too large. Average results over 10 trials are reported in the simulations. Theoretical results were obtained from Eqs.35 and 34. They predict the simulation results very well(compare left and right columns).

The capacity of the memory structure induced by BTSP at the scale of the brain. We have so far considered an input dimension of $m = 25,000$, which can be viewed as roughly corresponding to the size of area CA3, and a dimension of $n = 39,000$ for the memory traces in area CA1. According to Andersen et al. (2007) the real size of these structures in the rat brain is 10 times larger, and 100 times larger in the human brain. Hence the question arises of how the previously discussed results on the capacity of memory structures created through BTSP scale up to these sizes. Unfortunately, this scale is difficult to probe with numerical experiments. But the theory that we developed

can be applied to any scale. We show in Fig.5 A - C that the theory fits the results of numerical experiments very well for scaling factors between 0.05 and 1.5 of the previously considered sizes, where numerical experiments are still feasible. We then show in Fig. 5 D-G the prediction of several key Hamming distances as well as the capacity predictions of this theory for actual brain sizes. We always assume here a masking fraction of 33% for partial cues. Those values of the total number of input patterns where the ratio $HD(\boldsymbol{z}(\boldsymbol{x}), \boldsymbol{z}(\boldsymbol{x'}))/HD(\boldsymbol{z}(\boldsymbol{x}_a), \boldsymbol{z}(\boldsymbol{x}_b))$ is close to 0, say has a value below 0.1, can be viewed as regions where recall from partial cues works well. Our theoretical analysis predicts that the maximal number of input patterns for which this is the case grows logarithmically with the size of area CA1. It yields for the scale of the human brain a value of $25,000$. Hence it covers the number of memories that the brain creates during a limited amount of time, say during a few years. But it suggests that other and larger brain structures are needed for storing memories in the long term.

BTSP induces a repulsion effect for closely related memory items. It has been shown in several experimental studies Chanales et al. (2017); Wammes et al. (2022); Wanjia et al. (2021); Zheng et al. (2021) that hippocampal activity in the human brain, especially in CA1, exhibits for closely related memory items less similarity than for more dissimilar memory items. In other words, representation similarity is related to the similarity of the memory content by a U-shaped function. This repulsion effect is thought to be important for the encoding of episodic memories, as it enables the separation of events that differ just in a few key details, such as two scenarios that look alike but occurred on different days or lead to different outcomes. We wondered whether BTSP, in particular its weight dependence, would suffice to explain the experimentally observed repulsion effect in human memory.

A closer analysis of the impact of BTSP confirms this conjecture. For simplicity, we first analyze the most transparent case where just two input patterns are learnt through BTSP, and all except 95% of weights to CA1 neurons have a value of 0. If a CA1 neuron z_j gets a plateau potential for only one of the two input patterns, most weights from neurons that are active in that pattern will be set to 1 (since most of them were initially 0). These weights remain unchanged when the other pattern is presented since z_j does not receive a plateau potential for this other input pattern in the considered case. Therefore, if the two input patterns have a high overlap ratio (e.g., 80%), z_j is likely to fire for both patterns after learning. However, if the overlap ratio decreases, z_j is more likely to fire for only one of the two patterns, which causes the decrease of the dark curve in Fig. 6 B and also the decrease of the left part of the U-shape of the curve in Fig. 6 A.

On the other hand, if z_j receives plateau potentials for both input patterns during learning, most of its weights from neurons that are active in both patterns will have value 0 after learning, since they are first increased and then decreased by BTSP. However, if the two patterns have more disjoint sets of 1's, the long-term depression(LTD) part of BTSP will only affect weights from those input bits that have value 1 in both input patterns, and z_j is, therefore, more likely to fire after learning for both input patterns. According to the light curve in Fig. 6B this effect occurs for input patterns with at most 25% common 1's. More precisely, it is responsible for the increase in the orange curve for small overlap ratios and the right part of the U-shape in 6 A. Combining these two aspects, our results suggest that the change in the U-shaped curve is caused by different sets of CA1 neurons driven by the synaptic depression induced by the BTSP. This theoretical analysis is corroborated by the results of numerical simulations, see Fig. 6 A and B.

We examined in simulations also a more complex scenario that better reflects the experimental design in the studies of Chanales et al. (2017); Wammes et al. (2022); Wanjia et al. (2021); Zheng et al. (2021): Not just two input patterns are presented, but a larger set of M patterns. We focused then on the similarity of the memory traces for two of these M patterns, that appeared at random positions within the batch of M input patterns, in dependence on the similarity of these two input patterns. Fig. 6 C shows that the overlap of the memory traces for these two input patterns is smaller if they are more similar, as shown for the human brain in Fig. 3C of Chanales et al. (2017).

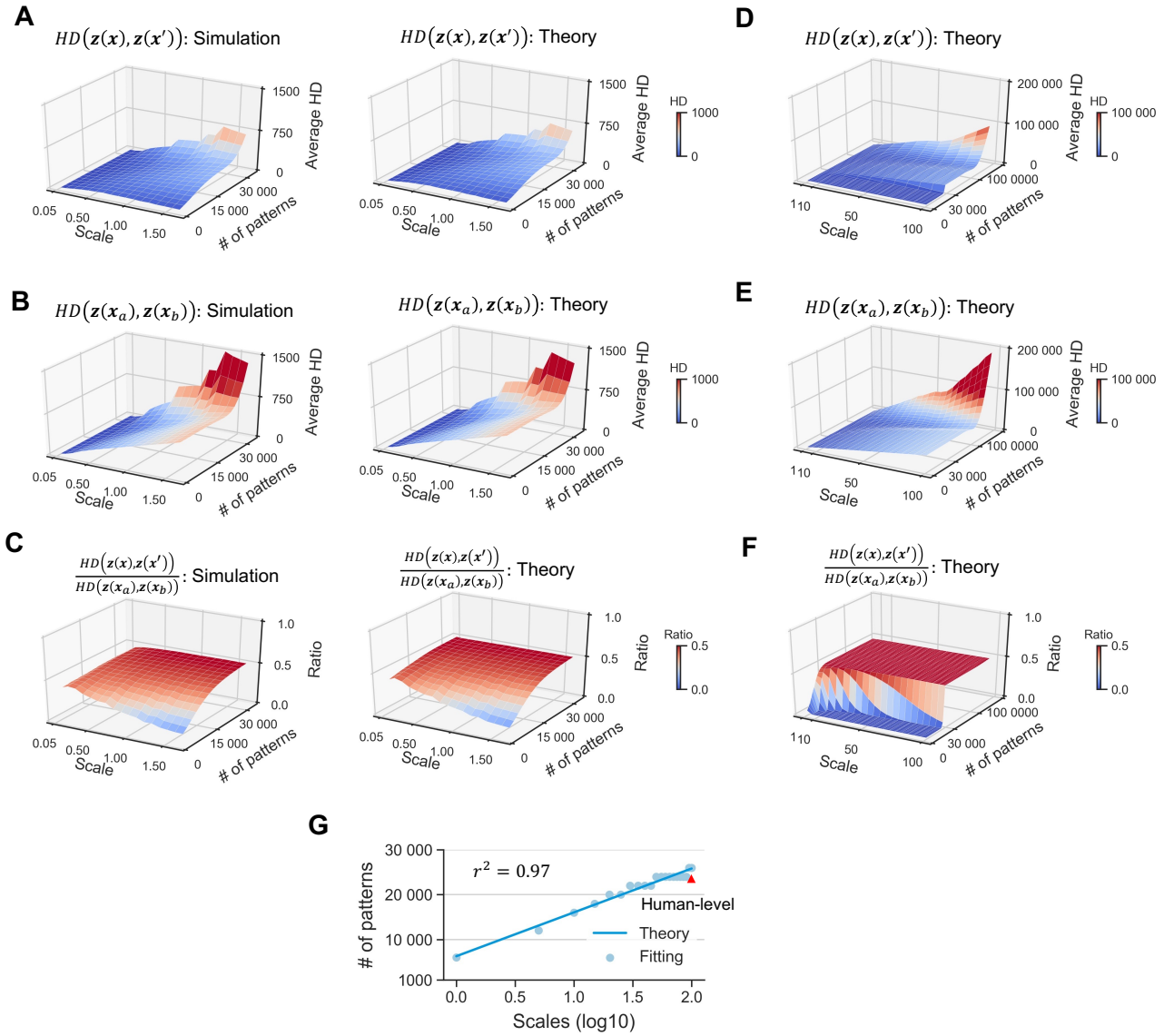

Figure 5. Memory capacity of the memory structure created through BTSP at the scale of the brain. $HD(\boldsymbol{z}(\boldsymbol{x}), \boldsymbol{z}(\boldsymbol{x'}))$ refers to the average HD between the memory traces recalled by any complete pattern $\boldsymbol{x}$ or the corresponding masked pattern $\boldsymbol{x'}$ with 33% masking fraction. We scaled the input size m and output size n according to the scaling factor defined on the x-axis. (**A**) Validation of our theory on predicting $HD(\boldsymbol{z}(\boldsymbol{x}), \boldsymbol{z}(\boldsymbol{x'}))$ at various scales that are still accessible for numerical simulations. (**B**) The same for the estimate of $HD(\boldsymbol{z}(\boldsymbol{x}_a), \boldsymbol{z}(\boldsymbol{x}_b))$ at different scales. (**C**) The ratio $HD(\boldsymbol{z}(\boldsymbol{x}), \boldsymbol{z}(\boldsymbol{x'}))/HD(\boldsymbol{z}(\boldsymbol{x}_a), \boldsymbol{z}(\boldsymbol{x}_b))$at difference scales. The theoretical predictions were obtained by Eq.25 and Eq.26. Note that these predictions in all three cases fit the results of numerical simulations very well. (**D**) - (**F**): Predictions of the same measurements as in panels A-C at larger scales, up to the scale of the human brain (scaling factor 100). Of particular interest in panel F is that region where the ratio stays well below 0.5, say below 0.1. (**G**) Evaluation of the maximal number of patterns where the ratio stays below 0.1 in panel F, for many different scales. We used here a log scale for the x-axis in order to indicate that this maximal number increases logarithmically with the number of neurons in CA1, reaching a value around 25,000 for the scale of the human brain. The straight line resulted from a linear fit.

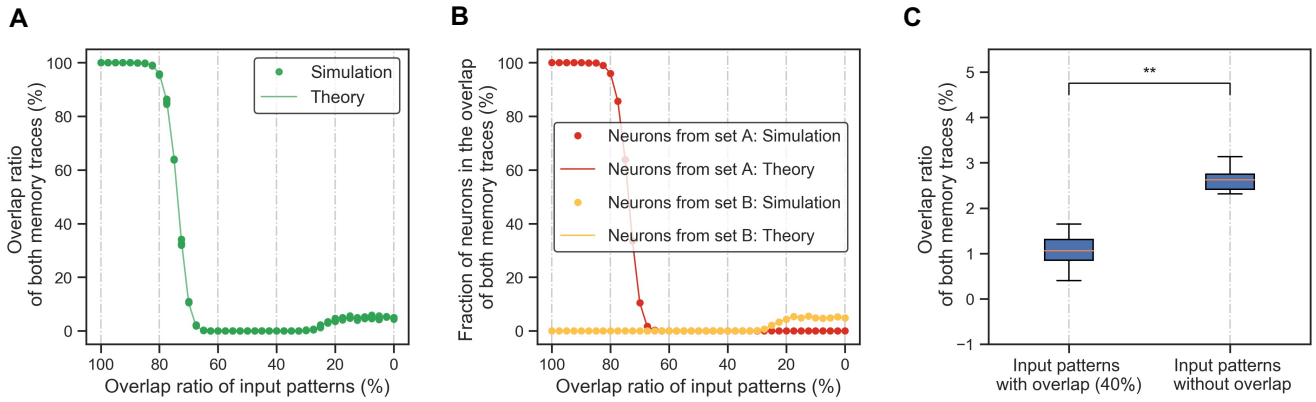

Figure 6. Reproduction of the repulsion effect through BTSP. (**A**) The overlap ratio of memory traces for a pair of similar patterns as a function of the overlap ratio (= ratio of common 1's) of the corresponding input patterns exhibits a U-shape if these memory traces are induced by BTSP. Only the theoretically tractable case where just two patterns are learnt is considered, and theory (based on Eq.37) and numerical experiments both produce the same U-shaped curve. (**B**) Elucidating the cause for this U-shaped curve. We consider two sets of CA1 neurons: The set A of neurons that get exactly one plateau potential when the two input patterns are processed, and the set B of neurons that get a plateau potential for both of them. Theoretical predictions were obtained by Eqs. 40 and 41, respectively. These predictions agree very well with the results of numerical experiments. (**C**) Results of numerical experiments where two more-or-less similar patterns belong to a set of 1000 input patterns for which memory traces are created through BTSP. Also in this case the similarity (= overlap ratio) of input patterns with 40% overlap ratio was lower than that of non-overlapping pairs of input patterns (p=0.005, two-tailed one-sample t-test).The error bar represented the standard deviation over 20 trials. $^{**}p < 0.01$.

Discussion

BTSP is one of the very few rules for synaptic plasticity in the brain that has been examined not only in-vitro, but in awake and behaving animals. It differs in several essential aspects from the previously found plasticity rule. But except for Cone and Shouval (2021) it has not yet been examined analytically. Their analysis focused on rodent navigation, and was based on the assumption that the same CA1 neuron is subject to plateaus in repetitions of the same path through a maze. In contrast, we present each input pattern only once, in order to model generic induction of episodic memories or conjunctive memories through one-shot learning. Our assumption about the probability distribution of plateau potentials is based on the data of Grienberger and Magee (2022). These data suggest that repeated plateau potentials in a CA1 neuron during trial repetitions occur with very low probability.

We have presented a simplified model for BTSP that supports a comprehensive theoretical analysis of the memory structure which BTSP produces. We found that the results differ strongly from previously analyzed memory models and plasticity rules. Predictions of this theory agree very well with the results of large-scale simulations.

Since BTSP has so far only been documented in area CA1 of the hippocampus, we focused our theoretical analysis on those types of memory that are especially supported by CA1: Instantaneously created traces for episodic and conjunctive memories. A characteristic feature of both types of memory is that it can be recalled from partial cues, i.e., inputs to area CA1 where substantial parts of the original input pattern that gave rise to the memory trace are missing, e.g., parts of an experience that gave rise to an episodic memory. Another characteristic feature is that input patterns that give rise to separate episodic or conjunctive memories may share substantial parts of their content. Hence the common assumption in the memory models, that different input patterns are orthogonal appears to be problematic for these types of memories. Therefore we have addressed with our theory especially the case of recall

with partial cues where many bits of the original input patterns are masked, and the case of non-orthogonal input patterns. We found that the memory structure that is created through BTSP can handle both of these challenges for a fairly large number of input patterns.

The theory that we have developed also applies to large network sizes, in particular to the actual size of area CA1 in rats and humans. The theory is needed to tackle this scale because it is beyond standard numerical simulation capabilities. We found that the memory capacity of CA1, for a type of memory that also supports recall from partial cues and memory for non-orthogonal patterns, scales only logarithmically with its size, reaching a value around 25,000 for the scale of the human brain. This capacity can be argued to suffice for the initial storage of a very large number of episodic and conjunctive memories, but it also suggests that additional brain areas are required for long-term storage.

We have also shown that BTSP induces the well-known repulsion effect, whereby very similar input patterns lead to more dissimilar memory traces than less similar input patterns. This repulsion effect has been reported in numerous studies of the human brain Chanales et al. (2017); Wammes et al. (2022); Wanjia et al. (2021); Zheng et al. (2021). But to the best of our knowledge, an explanation of this effect in terms of underlying mechanisms has been missing. We have demonstrated that BTSP induces this repulsion effect. More precisely, the fact that according to Milstein et al. (2021) the decision between potentiation and depression of a synaptic weight depends for BTSP on the current value of the synaptic weight, is responsible for the repulsion effect.

Altogether, our theoretical results on functional properties of memory structures that are created through BTSP provide a number of predictions for future experimental work on BTSP and its impact on memory structures that it creates in the hippocampus, especially for more quantitative analyses of recall of memories that are induced through BTSP. It would also be desirable to examine in models the functional impact of further features of the anatomy of CA1 and its connections from and to other areas. Another open question is whether the theoretical analysis of BTSP can be extended to network models that reflect at least some of these additional features.

Methods

A simplified model for BTSP For ease of analysis, we denote vectors and matrices in bold in the following to distinguish them from scalar values. The experimental BTSP Milstein et al. (2021) describes the change in synaptic weight as a function of the product of the eligibility traces of pre-synaptic spikes and plateau signals,

$$\frac{d\boldsymbol{W}}{dt} = \kappa^{+}(\boldsymbol{W}^{max} - \boldsymbol{W}) \circ q^{+}(\boldsymbol{IS} \times \boldsymbol{ET}^{T}) - \kappa^{-}\boldsymbol{W} \circ q^{-}(\boldsymbol{IS} \times \boldsymbol{ET}^{T}), \tag{2}$$

where $\circ$ denotes the element-wise multiplication, $\times$ denotes the matrix multiplication, $\boldsymbol{W}^{max}$ denotes a constant weight matrix with all elements equal to a weight saturation value w^{max}, κ^{+} and κ^{-} are the learning rates that control the magnitudes of synaptic potentiation and depression, q^{+} and q^{-} are the generalized sigmoid functions. $\boldsymbol{ET}$ and $\boldsymbol{IS}$ are the eligibility traces that model the time effects of presynaptic spikes and plateau signals, respectively. Each synapse from an input neuron to a CA1 neuron has an eligibility trace $\boldsymbol{ET}$ of the presynaptic inputs $\boldsymbol{x}_t$, which yields

$$\tau_E \frac{d\boldsymbol{ET}}{dt} = -\boldsymbol{ET} + \lambda_E \boldsymbol{x}_t, \tag{3}$$

where the decay factor τ_E and the learning rate factor λ_E are used for controlling the time effects. Similarly, the trace $\boldsymbol{IS}$ models the time effects of plateau signals $\boldsymbol{y}_t$,

$$\tau_I \frac{d\boldsymbol{IS}}{dt} = -\boldsymbol{IS} + \lambda_I \boldsymbol{y}_t, \tag{4}$$

where the decay factor τ_I and the learning rate factor λ_I are used for controlling the time effects. The function of q^{+} and q^{-} control the amplitude of plasticity signal overlap which is defined by the generalized sigmoid function

$s(x,\alpha,\beta)$ with a scale and offset to meet the following edge condition: $s=0$ when $x=0$,$s=1$ when $x=1$. The expression can be formalized by

$$\begin{cases} \hat{s}(x,\alpha,\beta) = \frac{1}{1+e^{-(\beta(x-\alpha))}}, \\ s(x,\alpha,\beta) = \frac{\hat{s}(x,\alpha,\beta)-\hat{s}(0,\alpha,\beta)}{\hat{s}(1,\alpha,\beta)-\hat{s}(0,\alpha,\beta)}, \\ q^{+}(x) = s(x,\alpha^{+},\beta^{+}), \\ q^{-}(x) = s(x,\alpha^{-},\beta^{-}), \end{cases} \tag{5}$$

where α^{+}, α^{-}, β^{+} and β^{-} are the scaling factors to control the shape of the generalized sigmoid function.

We discretize the experimental rule of Eq.2 by using the Euler method:

$$\begin{cases} \Delta \boldsymbol{W}_t & = dt\,\kappa^{+}(\boldsymbol{W}^{max}-\boldsymbol{W}_t)\circ q^{+}(\boldsymbol{IS}_t \times \boldsymbol{ET}_t^T) - dt\,\kappa^{-}\boldsymbol{W}_t \circ q^{-}(\boldsymbol{IS}_t \times \boldsymbol{ET}_t^T), \\ \boldsymbol{W}_{t+1} & = \boldsymbol{W}_t + \Delta \boldsymbol{W}_t. \end{cases} \tag{6}$$

In the first simplification, we replaced the modelling of eligibility traces with the binary variables $\boldsymbol{x}$ and $\boldsymbol{y}$. It yields

$$\Delta \boldsymbol{W}_t = dt\,\kappa^{+}(\boldsymbol{W}^{max}-\boldsymbol{W}_t)\circ q^{+}(\boldsymbol{y}_t \times \boldsymbol{x}_t^T) - dt\,\kappa^{-}\boldsymbol{W}_t \circ q^{-}(\boldsymbol{y}_t \times \boldsymbol{x}_t^T). \tag{7}$$

According to the definition of $s(x,\alpha,\beta)$, q^{+} and q^{-} satisfy the following edge conditions

$$q^{+}(1) = q^{-}(1) = 1, q^{+}(0) = q^{-}(0) = 0. \tag{8}$$

By setting $\kappa^{+} = \kappa^{-} = 1/dt$ and $w^{max} = 1$, we can obtain the BTSP rule as Eq.1. Given any initial binary weight value, the edge conditions guarantee that the weight update produces the binary flipping behaviours.

There is a small technical issue with the time-less version of BTSP that we consider here for simplicity: For certain values of the initial weights the weight may arrive at an equilibrium out of which no further weight update through BTSP is possible. If one takes the temporal aspect of BTSP into account, as in the more complex model of Milstein et al. (2021), then this problem does not arise because inputs that arrive at the edges of the plasticity interval are subject to slightly different weight updates by BTSP. In our simplified setup where these temporal aspects are not considered, one could enable the weight to get out of an equilibrium by adding stochastic components to the weight updates. The same comment applies to our binary version of this simplified version of BTSP.

Notations used in the subsequent theoretical analysis. Suppose $\boldsymbol{x}$ consists of a bit vector of length m, $\boldsymbol{y}$ consists of a bit vector of length n, the support set of bit positions in $\boldsymbol{x}$ that is on (i.e., have value 1) as $supp(\boldsymbol{x})$ and the size of the support set of $\boldsymbol{x}$ be $\iota(\boldsymbol{x})$. During the learning phase, BTSP runs once for all M patterns. We denote the set of the positions in $\boldsymbol{y}$ that induces the plateau potential for $\boldsymbol{x}$ as $Q(\boldsymbol{x})$. We use $\boldsymbol{z}(\boldsymbol{x})$ to represent the pattern of CA1 neurons that are produced for input pattern $\boldsymbol{x}$ after learning of all M patterns is finished. A summary of the list of key parameters in the following derivations is provided in table 1.

Estimation of the Hamming distance of memory traces created by BTSP on orthogonal memory pattern. Suppose that $\boldsymbol{x}$ is a random binary input pattern with the input density of f_d and $\boldsymbol{x'}$ is the masked pattern of $\boldsymbol{x}$ with the fraction f_d of 1's bits being masked to zeros. For simplicity of notations, we overload $\boldsymbol{W}$ which in the following analysis refers to the weights after learning of M samples.

We formulate the expected Hamming distance in terms of the probabilities that individual neurons fire differently for

Table 1. Definitions and explanations of main parameters used for the derivation.

Variables	Category	Explanations	Definitions
$f(k_x, N_x, p_x)$	General notations	Probability of binomial distribution	
$F(k_x, N_x, p_x)$		Cumulative binomial distribution	
$Q(\boldsymbol{x})$		The set collects positions in $\boldsymbol{y}$ that induce plateau potential for $\boldsymbol{x}$	
$supp(\boldsymbol{x})$		The set collects the positions of 1's bits in $\boldsymbol{x}$	
$\iota(\boldsymbol{x})$		Size of the support set of $\boldsymbol{x}$	
$S(j)$		The set collects the index of patterns that induced plateau potentials at position j in $\boldsymbol{y}$ during learning	
$\boldsymbol{y}(j)$		Unit vector with only j-th bit position being 1	
c		Number of common bits in different non-orthogonal input patterns	
f_q		Probability of plateau potential	
f_p		Input density	
f_w		Probability of connections	
f_d		Fraction of 1's bits being masked to zeros	
p_e, p_o	Estimate for orthogonal patterns	Conditional probability of an input neuron in $supp(\boldsymbol{x})$ transmitting a spike to a CA1 neuron, given whether or not the CA1 neuron gets plateau potential for $\boldsymbol{x}$ during learning	Eq.16 and 17
p_{f1}, p_{f2}		Conditional probability that a CA1 neuron can fire for $\boldsymbol{x}$, given whether or not the CA1 neuron got plateau for $\boldsymbol{x}$ during learning	Eqs.18 and 19
$p_{f_{\text{avg}}}$		Probability that the any given CA1 neuron can fire for $\boldsymbol{x}$ after learning	Eq.20
$\hat{f}_p$		Probability of the non-common-place bits being 1	
c_w	Estimate for non-orthogonal patterns	Number of connections from an input neuron which receives common 1's bits to a CA1 neuron	
$\hat{p}_e, \hat{p}_o$		Conditional probability of an input neuron in $supp(\boldsymbol{x})$ transmitting a spike to a CA1 neuron, given whether or not the CA1 neuron got plateau for $\boldsymbol{x}$ during learning	Eqs.27 and 28
$\hat{p}_{f1}, \hat{p}_{f2}$		Conditional probability that a CA1 neuron can fire for $\boldsymbol{x}$, given whether or not the CA1 neuron got plateau for $\boldsymbol{x}$ during learning	Eqs.29 and 30
$\hat{p}_{f_{\text{avg}}}$		Probability that any given CA1 neuron can fire for $\boldsymbol{x}$ after learning	Eq.31
$\hat{p}_{c1}, \hat{p}_{c2}$		Conditional probability that CA1 neurons fire differently for $\boldsymbol{x}$ and $\boldsymbol{x}'$, given whether or not the CA1 neuron got plateau for $\boldsymbol{x}$ during learning	Eqs.32 and 33
ω	Estimate for repulsion effects	Probability of non-zeros entries of binary random weight	

any complete pattern and the masked pattern

$$
\begin{aligned}
&E(HD(\boldsymbol{z}(\boldsymbol{x}),\boldsymbol{z}(\boldsymbol{x'})))\\
&=\sum_{j=1}^{n} P(z_j(\boldsymbol{x}) \neq z_j(\boldsymbol{x'}))\\
&=\sum_{j=1}^{n} P(j \in Q(\boldsymbol{x}))P(z_j(\boldsymbol{x}) \neq z_j(\boldsymbol{x'})|j \in Q(\boldsymbol{x})) + P(j \notin Q(\boldsymbol{x}))P(z_j(\boldsymbol{x}) \neq z_j(\boldsymbol{x'})|j \notin Q(\boldsymbol{x}))\\
&=nf_q P(z_j(\boldsymbol{x}) \neq z_j(\boldsymbol{x'})|j \in Q(\boldsymbol{x})) + n(1-f_q)P(z_j(\boldsymbol{x}) \neq z_j(\boldsymbol{x'})|j \notin Q(\boldsymbol{x}))
\end{aligned}
\tag{9}
$$

where $Q(\boldsymbol{x})$ refers to the set of indices of the CA1 neurons that got plateau potentials for $\boldsymbol{x}$ during the learning process. Eq.9 indicates that to estimate this expression $E(HD(\boldsymbol{z}(\boldsymbol{x}),\boldsymbol{z}(\boldsymbol{x'})))$, we only need to estimate the probability that a CA1 neuron fires differently for $\boldsymbol{x}$ and $\boldsymbol{x'}$ for both the set $Q(\boldsymbol{x})$ and its complement.

To do this, we divide the derivations into two parts. In the first part, we estimate the firing probability of any CA1 neuron in response to $\boldsymbol{x}$ given size I of $supp(\boldsymbol{x})$. In the second part, we estimate the average Hamming distance and relax the assumptions on the size of the support set.

Preparation. We will use the properties of the probability of binomial distribution $f(k_x, N_x, p_x)$ and its cumulative binomial distribution $F(k_x, N_x, p_x)$ for estimating the average Hamming distance. To this end, we first introduce the following Lemma which is fundamental for the following derivations.

Lemma 1. *Let $f(k_x, N_x, p_x)$ be the probability of binomial distribution that a random variable X gets k_x success in N_x Bernoulli trials. Then, the probability that X is even can be given by*

$$P(X \text{ is even}) = \frac{1}{2}(1+(1-2p_x)^{N_x}). \tag{10}$$

Similarly, the probability that X is odd can be given by

$$P(X \text{ is odd}) = \frac{1}{2}(1-(1-2p_x)^{N_x}). \tag{11}$$

The proof is provided in the *Appendix. A*.

Part 1: Estimate of the firing probability of CA1 neurons after learning M patterns by BTSP.

The firing behaviour of any given CA1 neuron is determined by whether the summation of afferent spikes is greater than a given firing threshold v_{th}. Thus, a basic estimate is the spiking transmission probability that any input neuron i can transmit a spike to a specific CA1 neuron at the end of learning.

Since this transmission probability depends on the open status of the weight w_{ji} (i.e., $w_{ji}=1$), we start with the modelling of the effect of the weight flipping on w_{ji}. We introduce the unit vector $\boldsymbol{y}(j) \in \mathbb{R}^{n\times 1}$ with only the j bit position being 1. By using the fact that the weight $\boldsymbol{W}$ can be written as the summation of multiple weight updates, we write the expression of weights in the order of the bit position j,

$$\boldsymbol{W} = \sum_{j=1}^{n} \boldsymbol{y}(j) \times \hat{1}(\sum_{k\in S(j)} \boldsymbol{x}_k^T), \tag{12}$$

where the set $S(j)$ collects the index of patterns that induced the plateau potentials in position j during the learning process. $\hat{1}(x)$ is an indicator function which models the impact of the parity of weight update:

$$\hat{1}(x) = \begin{cases} 1, & \text{if } x \text{ is odd,}\\ 0, & \text{if } x \text{ is even.}\end{cases} \tag{13}$$

We then calculate the weighted sum of inputs by

$$\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{W} \times \boldsymbol{x} = \sum_{j=1}^{n} \left(\boldsymbol{y}(j) \times \hat{1}(\sum_{k \in S(j)} \boldsymbol{x}_k^T) \right) \times \boldsymbol{x} = \sum_{j=1}^{n} \boldsymbol{y}(j) \times \left(\hat{1}(\sum_{k \in S(j)} \boldsymbol{x}_k^T) \times \boldsymbol{x} \right). \tag{14}$$

where $\boldsymbol{x}_k$ represents the input patterns that induce the plateau potential at position j in the learning phase.

Let u_j be the j_{th} element of $\boldsymbol{u}$. According to the definition of $\boldsymbol{y}(j)$, we have

$$u_j(\boldsymbol{x}) = \hat{1}(\sum_{k \in S(j)} \boldsymbol{x}_k^T) \times \boldsymbol{x} = \sum_{i \in supp(\boldsymbol{x})} \hat{1}(\sum_{k \in S(j)} x_{ki}), \tag{15}$$

where x_{ki} denote the element i in the vector $\boldsymbol{x}_k$.

Note that $\hat{1}(\sum_{k \in S(j)} x_{ki})$ is binary. Owning to the independent assumption on f_p and f_q, the variable $\hat{1}(\sum_{k \in S(j)} x_{ki})$ can be modelled as Bernoulli variable. We next analyze the probability of $P(\hat{1}(\sum_{k \in S(j)} x_{ki}) = 1)$ in the following two cases.

- For any $j_1 \in Q(\boldsymbol{x})$, we estimate the probability p_e that any input neuron with its position i belongs to $supp(\boldsymbol{x})$ can transmit a spike to one CA1 neuron that receives plateau signals for $\boldsymbol{x}$ during learning:

$$p_e := P(\hat{1}(\sum_{k \in S(j_1)} x_{ki}) = 1 | \boldsymbol{x}) = P(\sum_{k \in S(j_1)} x_{ki} \text{ is odd} | \boldsymbol{x}) = \frac{1}{2}(1 + (1 - 2 f_p f_q)^{M-1}), \tag{16}$$

  where the estimate of the parity in the last equality uses Lemma 1.

- For any $j_2 \notin Q(\boldsymbol{x})$, we estimate the probability p_o that any input neuron with its position i belongs to $supp(\boldsymbol{x})$ can transmit a spike to a CA1 neuron that does not receive plateau signals for $\boldsymbol{x}$ during learning

$$p_o := P(\hat{1}(\sum_{k \in S(j_2)} x_{ki}) = 1 | \boldsymbol{x}) = P(\sum_{k \in S(j_2)} x_{ki} \text{ is odd} | \boldsymbol{x}) = \frac{1}{2}(1 - (1 - 2 f_p f_q)^{M-1}). \tag{17}$$

The above equations have given the estimate of probability that any connection can keep active after the learning of M patterns. On the other hand, the transmission probability is also affected by the inherited connection probability f_w. By incorporating the influence of f_w, we use the cumulative binomial distribution F to estimate the firing probability as follows,

- For any $j_1 \in Q(\boldsymbol{x})$, we have:

$$p_{f1}(j_1, \boldsymbol{x}, I) := P(u_{j_1} > v_{th} | \boldsymbol{x}, j_1 \in Q(\boldsymbol{x})) = 1 - F(v_{th}, I, f_w p_e). \tag{18}$$

- For any $j_2 \notin Q(\boldsymbol{x})$, we have:

$$p_{f2}(j_2, \boldsymbol{x}, I) := P(u_{j_2} > v_{th} | \boldsymbol{x}, j_2 \notin Q(\boldsymbol{x})) = 1 - F(v_{th}, I, f_w p_o). \tag{19}$$

The above equations 18 and 19 show that the formulas of firing probability do not depend on the specific positions in y. For brevity, we will omit the indices j_1 and j_2 of the brackets in the following derivations.

Given the estimate of p_{f1} and p_{f2}, we can obtain the average firing probability for any j

$$\begin{aligned} & P(u_j > v_{th} | \boldsymbol{x}) \\ & \quad = P(j \in Q(\boldsymbol{x})) P(u_j > v_{th} | j \in Q(\boldsymbol{x})) + P(j \notin Q(\boldsymbol{x})) P(u_j > v_{th} | j \notin Q(\boldsymbol{x})) \\ & \quad = f_q p_{f1}(\boldsymbol{x}, I) + (1 - f_q) p_{f2}(\boldsymbol{x}, I) \end{aligned} \tag{20}$$

where F is the cumulative binomial distribution can be calculated by incomplete beta function.

Part 2: Estimate of the average Hamming distance of memory traces recalled by complete or masked patterns.

Given the firing probability of any CA1 neuron, we estimate $P(z_j(\boldsymbol{x}) \neq z_j(\boldsymbol{x'}))$ in two separate cases accordingly.

- For any $j_1 \in Q(\boldsymbol{x})$, using the estimate of the average firing probability then gives

$$\begin{aligned}
&P(z_{j_1}(\boldsymbol{x}) \neq z_{j_1}(\boldsymbol{x'}) | j_1 \in Q(\boldsymbol{x}), \iota(\boldsymbol{x}) = I) \\
&= P(z_{j_1}(\boldsymbol{x}) = 1, z_{j_1}(\boldsymbol{x'}) = 0 | j_1 \in Q(\boldsymbol{x}), \iota(\boldsymbol{x}) = I) \quad \text{(Note that } P(z_{j_1}(\boldsymbol{x}) = 0, z_{j_1}(\boldsymbol{x'}) = 1) = 0) \\
&\approx P(z_{j_1}(\boldsymbol{x}) = 1 | j_1 \in Q(\boldsymbol{x}), \iota(\boldsymbol{x}) = I) P(z_{j_1}(\boldsymbol{x'}) = 0 | j_1 \in Q(\boldsymbol{x}), \iota(\boldsymbol{x}) = I) \\
&= F(v_{th}, I(1-f_d), f_w p_e)(1 - F(v_{th}, I, f_w p_e)).
\end{aligned} \tag{21}$$

  Remark: to keep the estimation simple, we approximate $P(z_{j_1}(\boldsymbol{x}) = 1, z_{j_1}(\boldsymbol{x'}) = 0)$ as two independent terms, $P(z_{j_1}(\boldsymbol{x}) = 1)$ and $P(z_{j_1}(\boldsymbol{x'}) = 1)$ in the above estimate. It ignores the dependence on the same parity of the weight updates indicated by Eqs. 18 and 19. Nevertheless, our numerical estimates show good consistency using the simple strategy. A more precise but complicated estimate is provided in the following derivation for the non-orthogonal memory items (Eqs.32 and 33), where we use the total probability rule to deal with this dependency owning to common 1's bit.

- For any $j_2 \notin Q(\boldsymbol{x})$, we have

$$\begin{aligned}
&P(z_{j_2}(\boldsymbol{x}) \neq z_{j_2}(\boldsymbol{x'}) | j_2 \notin Q(\boldsymbol{x}), \iota(\boldsymbol{x}) = I) \\
&\quad = P(z_{j_2}(\boldsymbol{x}) = 1, z_{j_2}(\boldsymbol{x'}) = 0 | j_2 \notin Q(\boldsymbol{x}), \iota(\boldsymbol{x}) = I) \\
&\quad \approx F(v_{th}, I(1-f_d), f_w p_o)(1 - F(v_{th}, I, f_w p_o)).
\end{aligned} \tag{22}$$

Combining Eqs.21 and 22 gives

$$\begin{aligned}
&E(HD(\boldsymbol{z}(\boldsymbol{x}), \boldsymbol{z}(\boldsymbol{x'})) | \iota(\boldsymbol{x}) = I) \\
&= n\left(f_q P(z_{j_1}(\boldsymbol{x}) \neq z_{j_1}(\boldsymbol{x'}) | j_1 \in Q(\boldsymbol{x}), \iota(\boldsymbol{x}) = I) + (1-f_q) P(z_{j_2}(\boldsymbol{x})) \neq z_{j_2}(\boldsymbol{x'}) | j_2 \notin Q(\boldsymbol{x}), \iota(\boldsymbol{x}) = I)\right) \\
&\approx n\left(f_q F(v_{th}, I(1-f_d), f_w p_e)(1 - F(v_{th}, I, f_w p_e)) + (1-f_q) F(v_{th}, I(1-f_d), f_w p_o)(1 - F(v_{th}, I, f_w p_o))\right).
\end{aligned} \tag{23}$$

We use the total probability rule to relax the assumption of the size of the support set on the average firing rate $p_{f_{avg}}(\boldsymbol{x})$

$$p_{f_{avg}}(\boldsymbol{x}) := \sum_{I=0}^{m} C_m^I f_p^I (1-f_p)^{m-I} (f_q p_{f1}(\boldsymbol{x}, I) + (1-f_q) p_{f2}(\boldsymbol{x}, I)), \tag{24}$$

and relax the assumption on the average Hamming distance $E(HD(\boldsymbol{z}(\boldsymbol{x}), \boldsymbol{z}(\boldsymbol{x'})))$,

$$\begin{aligned}
&E(HD(\boldsymbol{z}(\boldsymbol{x}), \boldsymbol{z}(\boldsymbol{x'}))) \\
&= \sum_{I=0}^{m} Pr(\iota(\boldsymbol{x}) = I) E(HD(\boldsymbol{z}(\boldsymbol{x}), \boldsymbol{z}(\boldsymbol{x'})) | \iota(\boldsymbol{x}) = I) \quad \text{(by substituting Eq.23)} \\
&\approx n \sum_{I=0}^{m} C_m^I f_p^I (1-f_p)^{m-I} (f_q F(v_{th}, I(1-f_d), f_w p_e)(1 - F(v_{th}, I, f_w p_e)) + (1-f_q) F(v_{th}, I(1-f_d), f_w p_o)(1 - F(v_{th}, I, f_w p_o))).
\end{aligned} \tag{25}$$

Finally, we estimate the average Hamming distance between spiking patterns recalled by any two orthogonal input patterns. To this end, we introduce the notations $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$ to represent any two different orthogonal patterns. Owing

to the assumption of orthogonal memory items, we can simply utilize the estimate of the average firing rate $p_{f_{avg}}$ and estimate $HD(\boldsymbol{z}(\boldsymbol{x}_a), \boldsymbol{z}(\boldsymbol{x}_b)$ as below,

$$
\begin{aligned}
&E(HD(\boldsymbol{z}(\boldsymbol{x}_a), \boldsymbol{z}(\boldsymbol{x}_b))) \\
&= \sum_{j=1}^{n} P(z_j(\boldsymbol{x}_a) \neq z_j(\boldsymbol{x}_b)) \\
&= \sum_{j=1}^{n} P(z_j(\boldsymbol{x}_a) = 1, z_j(\boldsymbol{x}_b) = 0) + P(z_j(\boldsymbol{x}_a) = 0, z_j(\boldsymbol{x}_b) = 1) \\
&= \sum_{j=1}^{n} P(z_j(\boldsymbol{x}_a) = 1) P(z_j(\boldsymbol{x}_b) = 0) + P(z_j(\boldsymbol{x}_a) = 0) P(z_j(\boldsymbol{x}_b) = 1) \\
&= 2np_{f_{avg}}(1 - p_{f_{avg}}).
\end{aligned}
\tag{26}
$$

Estimation of the Hamming distance of memory traces for non-orthogonal memory items. Suppose $\boldsymbol{x}$ is a random binary input pattern with c common 1's bits which share with other all non-orthogonal input patterns. Let the input density of the remaining 1's bits in $\boldsymbol{x}$ be $\hat{f}_p$. Following the estimation of orthogonal memory items, we first estimate the firing rate of CA1, then estimate $E(HD(\boldsymbol{z}(\boldsymbol{x}), \boldsymbol{z}(\boldsymbol{x'})))$ and finally estimate $E(HD(\boldsymbol{z}(\boldsymbol{x}_a), \boldsymbol{z}(\boldsymbol{x}_b)))$.

The estimate of non-orthogonal memory items requires modelling the effects of common bits on the firing probability of CA1 neurons. Note that whether an input neuron i that receives the common 1's bit information can transmit a spike signal to a CA1 neuron j depends on the parity of weight updates w_{ji} during the learning process. Namely, if there is an odd number of plateau potentials evoking a CA1 neuron during learning, then the weights connected to the input neuron must be in an open state (i.e., $w_{ji} = 1$) after learning; otherwise they will be closed.

To model this parity, we analyze the conditional spiking transmission probability $P(\hat{1}(\sum_{k \in S(j)} x_{ki}) = 1 | S_j)$ given the size $s_j$of $S(j)$ in the following two separate cases.

- For any $j_1 \in Q(\boldsymbol{x})$, given the size s_{j_1} of $S(j_1)$,we estimate the probability $\hat{p}_e(s_{j_1})$ that any input neuron with its position $i \in supp(\boldsymbol{x})$ can transmit a spike to the CA1 neuron that receives plateau signals for $\boldsymbol{x}$ during learning:

$$
\hat{p}_e(s_{j_1}) := P(\hat{1}(\sum_{k \in S(j_1)} x_{ki}) = 1 | \boldsymbol{x}, s_{j_1}) = P(\sum_{k \in S(j_1)} x_{ki} \text{ is odd} | \boldsymbol{x}) = \frac{1}{2}(1 + (1 - 2\hat{f}_p)^{s_{j_1} - 1}). \tag{27}
$$

- For any $j_2 \notin Q(\boldsymbol{x})$, given the size s_{j_2} of $S(j_2)$, we introduce $\hat{p}_o(s_{j_2})$ that any neuron with its position $i \in supp(\boldsymbol{x})$ can transmit a spike to the CA1 neuron that does not receive plateau signals for $\boldsymbol{x}$:

$$
\hat{p}_o(s_{j_2}) := P(\hat{1}(\sum_{k \in S(j_2)} x_{ki}) = 1 | \boldsymbol{x}, s_{j_2}) = P(\sum_{k \in S(j_2)} x_{ki} \text{ is odd} | \boldsymbol{x}) = \frac{1}{2}(1 - (1 - 2\hat{f}_p)^{s_{j_2}}). \tag{28}
$$

Given the basic spike transmission probability, we will use the cumulative binomial distribution to estimate the firing probability by following the derivation of Eqs.27 and 28. The main difference from the estimation of orthogonal memory terms is that we will expand the estimation along the number of patterns projected to $S(j)$ to model the effect of common bits on weight updates. To further incorporate the impact of connection probability f_w on the common bits, we introduce c_w to represent the number of connections from the input neurons that receives common 1's bits to a CA1 neuron. We next evaluate the firing probability of CA1 neurons given the value of c_w.

- For any $j_1 \in Q(\boldsymbol{x})$, we have

$$
\begin{aligned}
&\hat{p}_{f1}(\boldsymbol{x}, I, c_w, c) \\
&= P(z_{j_1}(\boldsymbol{x}) = 1 | j_1 \in Q(\boldsymbol{x}), \iota(\boldsymbol{x}) = I) \\
&= \sum_{l'=0}^{M} P(z_{j_1}(\boldsymbol{x}) = 1, s_{j_1} = l' | j_1 \in Q(\boldsymbol{x}), \iota(\boldsymbol{x}) = I) \quad \text{(by the total probability rule)} \\
&= \sum_{l'=0}^{M} P(s_{j_1} = l') P(z_{j_1}(\boldsymbol{x}) = 1 | j_1 \in Q(\boldsymbol{x}), \iota(\boldsymbol{x}) = I, s_{j_1} = l') \quad \text{(splitting the summation into two parts)} \\
&= \sum_{l=0}^{\lfloor M/2 \rfloor} P(s_{j_1} = 2l) P(z_{j_1}(\boldsymbol{x}) = 1 | j_1 \in Q(\boldsymbol{x}), \iota(\boldsymbol{x}) = I, s_{j_1} = 2l) + P(s_{j_1} = 2l+1) P(z_{j_1}(\boldsymbol{x}) = 1 | j_1 \in Q(\boldsymbol{x}), \iota(\boldsymbol{x}) = I, s_{j_1} = 2l+1) \\
&= \sum_{l=0}^{\lfloor M/2 \rfloor} P(s_{j_1} = 2l)(1 - F(v_{th} - c_w, I - c, f_w \hat{p}_e(2l))) + P(s_{j_1} = 2l+1)(1 - F(v_{th}, I - c, f_w \hat{p}_e(2l+1))) \\
&= \sum_{l=0}^{\lfloor M/2 \rfloor} C_M^{2l} f_q^{2l} (1 - f_q)^{M-2l} (1 - F(v_{th} - c_w, I - c, f_w \hat{p}_e(2l))) + C_M^{2l+1} f_q^{2l+1} (1 - f_q)^{M-2l-1} (1 - F(v_{th}, I - c, f_w \hat{p}_e(2l+1))) \\
&\approx \sum_{l=0}^{\lfloor M/2 \rfloor} \frac{(M f_q)^{2l}}{e^{M f_q} (2l)!} (1 - F(v_{th} - c_w, I - c, f_w \hat{p}_e(2l))) + \frac{(M f_q)^{2l+1}}{e^{M f_q} (2l+1)!} (1 - F(v_{th}, I - c, f_w \hat{p}_e(2l+1)))
\end{aligned}
\tag{29}
$$

The last equation uses the Poisson approximation to the binomial probability $f(l, M, f_q)$ for simplifying analysis.

- Similarly, for any $j_2 \notin Q(\boldsymbol{x})$, we have

$$
\begin{aligned}
&\hat{p}_{f2}(\boldsymbol{x}, I, c_w, c) \\
&= P(z_{j_2}(\boldsymbol{x}) = 1 | j_2 \notin Q(\boldsymbol{x}), \iota(\boldsymbol{x}) = I) \\
&= \sum_{l'=0}^{M} P(z_{j_2}(\boldsymbol{x}) = 1, s_{j_2} = l', j_2 \notin Q(\boldsymbol{x}), \iota(\boldsymbol{x}) = I) \\
&= \sum_{l'=0}^{M} P(s_{j_2} = l') P(z_{j_2}(\boldsymbol{x}) = 1 | j_2 \notin Q(\boldsymbol{x}), \iota(\boldsymbol{x}) = I, s_{j_2} = l') \\
&= \sum_{l=0}^{\lfloor M/2 \rfloor} P(s_{j_2} = 2l) P(z_{j_2}(\boldsymbol{x}) = 1 | j_2 \notin Q(\boldsymbol{x}), \iota(\boldsymbol{x}) = I, s_{j_2} = 2l) \\
&+ P(s_{j_2} = 2l+1) P(z_{j_2}(\boldsymbol{x}) = 1 | j_2 \notin Q(\boldsymbol{x}), \iota(\boldsymbol{x}) = I, s_{j_2} = 2l+1) \\
&\approx \sum_{l=0}^{\lfloor M/2 \rfloor} \frac{(M f_q)^{2l}}{e^{M f_q} (2l)!} (1 - F(v_{th}, I - c, f_w \hat{p}_o(2l))) + \frac{(M f_q)^{2l+1}}{e^{M f_q} (2l+1)!} (1 - F(v_{th} - c_w, I - c, f_w \hat{p}_o(2l+1)))
\end{aligned}
\tag{30}
$$

Combining Eqs.29 and 30 gives the estimate of the average firing rates of CA1 neurons

$$
\begin{aligned}
&\hat{p}_{f_{avg}}(\boldsymbol{x}, c) \\
&= \sum_{I'=0}^{m-c} \sum_{c_w=0}^{c} P(\iota(\boldsymbol{x}) = I' + c, c_w) P(u_j > v_{th} | \iota(\boldsymbol{x}) = I' + c, c_w) \\
&= \sum_{I'=0}^{m-c} \sum_{c_w=0}^{c} P(\iota(\boldsymbol{x}) = I' + c) P(c_w) \left(f_q \hat{p}_{f1}(\boldsymbol{x}, I' + c, c_w, c) + (1 - f_q) \hat{p}_{f2}(\boldsymbol{x}, I' + c, c_w, c) \right) \\
&= \sum_{I'=0}^{m-c} \sum_{c_w=0}^{c} C_{m-c}^{I'} C_c^{c_w} \hat{f}_p^{I'} (1 - \hat{f}_p)^{m-c-I'} f_w^{c_w} (1 - f_w)^{c-c_w} (f_q \hat{p}_{f1}(\boldsymbol{x}, I' + c, c_w, c) + (1 - f_q) \hat{p}_{f2}(\boldsymbol{x}, I' + c, c_w, c)),
\end{aligned}
\tag{31}
$$

The estimate of $E(HD(\boldsymbol{z}(\boldsymbol{x}),\boldsymbol{z}(\boldsymbol{x'}))$ follows the derivation for orthogonal memory items. As suggested by Eq.9 , to estimate $E(HD(\boldsymbol{z}(\boldsymbol{x}),\boldsymbol{z}(\boldsymbol{x'})))$, we only need to estimate $P(z_j(\boldsymbol{x})=1,z_j(\boldsymbol{x'})=0|\iota(\boldsymbol{x})=I)$.

However, unlike the estimate of orthogonal memory items, the interaction of common bits and masking bits complicates the estimates. For simplification, we use the average number of masking common 1's bits, cf_d, to model the effects of the masking fraction on common 1's bits approximately. On this basis, we evaluate $P(z_j(\boldsymbol{x})=1,z_j(\boldsymbol{x'})=0|\iota(\boldsymbol{x})=I)$ in two different cases, $j_1\in Q(\boldsymbol{x})$ and $j_2\notin Q(\boldsymbol{x})$.

- For any $j_1\in Q(\boldsymbol{x})$, we have

$$
\begin{aligned}
&\hat{p}_{c1}(j_1,\boldsymbol{x},\boldsymbol{x'},I,c_w)\\
&=P(z_{j_1}(\boldsymbol{x})=1,z_{j_1}(\boldsymbol{x'})=0|j_1\in Q(\boldsymbol{x}),\iota(\boldsymbol{x})=I)\\
&=\sum_{l'=0}^{M}P(z_{j_1}(\boldsymbol{x})=1,z_{j_1}(\boldsymbol{x'})=0,s_{j_1}=l'|j_1\in Q(\boldsymbol{x}),\iota(\boldsymbol{x})=I)\\
&=\sum_{l'=0}^{M}P(s_{j_1}=l')P(z_{j_1}(\boldsymbol{x})=1,z_{j_1}(\boldsymbol{x'})=0|j_1\in Q(\boldsymbol{x}),\iota(\boldsymbol{x})=I,s_{j_1}=l')\\
&=\sum_{l=0}^{\lfloor M/2\rfloor}P(s_{j_1}=2l)P(z_{j_1}(\boldsymbol{x})=1,z_{j_1}(\boldsymbol{x'})=0|j_1\in Q(\boldsymbol{x}),\iota(\boldsymbol{x})=I,s_{j_1}=2l)\\
&+P(s_{j_1}=2l+1)P(z_{j_1}(\boldsymbol{x})=1,z_{j_1}(\boldsymbol{x'})=0|j_1\in Q(\boldsymbol{x}),\iota(\boldsymbol{x})=I,s_{j_1}=2l+1)\\
&\approx\sum_{l=0}^{\lfloor M/2\rfloor}\frac{(Mf_q)^{2l}}{e^{Mf_q}(2l)!}(1-F(v_{th}-c_w,I-c,f_w\hat{p}_e(2l))F(v_{th}-c_w(1-f_d),(I-c)(1-f_d),f_w\hat{p}_e(2l))\\
&+\frac{(Mf_q)^{2l+1}}{e^{Mf_q}(2l+1)!}(1-F(v_{th},I-c,f_w\hat{p}_e(2l+1)))F(v_{th},(I-c)(1-f_d),f_w\hat{p}_e(2l+1)),
\end{aligned}
\tag{32}
$$

- For any $j_2\notin Q(\boldsymbol{x})$, we have

$$
\begin{aligned}
&\hat{p}_{c2}(j_2,\boldsymbol{x},\boldsymbol{x'},I,c_w)\\
&=P(z_{j_2}(\boldsymbol{x})=1,z_{j_2}(\boldsymbol{x'})=0|j_2\notin Q(\boldsymbol{x}),\iota(\boldsymbol{x})=I)\\
&=\sum_{l'=0}^{M}P(z_{j_2}(\boldsymbol{x})=1,z_{j_2}(\boldsymbol{x'})=0,s_{j_2}=l'|z_{j_2}(\boldsymbol{x'})=0,j_2\notin Q(\boldsymbol{x}),\iota(\boldsymbol{x})=I)\\
&=\sum_{l'=0}^{M}P(s_{j_2}=l')P(z_{j_2}(\boldsymbol{x})=1,z_{j_2}(\boldsymbol{x'})=0|j_2\notin Q(\boldsymbol{x}),\iota(\boldsymbol{x})=I,s_{j_2}=l')\\
&=\sum_{l=0}^{\lfloor M/2\rfloor}P(s_{j_2}=2l)P(z_{j_2}(\boldsymbol{x})=1|j_2\notin Q(\boldsymbol{x}),\iota(\boldsymbol{x})=I,s_{j_2}=2l)\\
&\approx\sum_{l=0}^{\lfloor M/2\rfloor}\frac{(Mf_q)^{2l}}{e^{Mf_q}(2l)!}(1-F(v_{th},I-c,f_w\hat{p}_o(2l)))F(v_{th},(I-c)(1-f_d),f_w\hat{p}_o(2l))\\
&+\frac{(Mf_q)^{2l+1}}{e^{Mf_q}(2l+1)!}(1-F(v_{th}-c_w,I-c,f_w\hat{p}_o(2l+1)))F(v_{th}-c_w(1-f_d),(I-c)(1-f_d),f_w\hat{p}_o(2l+1))
\end{aligned}
\tag{33}
$$

As the above Eqs.32 and 33 do not depend on the specific positions in y, we omit the indices j_1 and j_2 in the following derivations for brevity.

Combining Eqs.32 and 33 gives the estimate of $E(HD(\boldsymbol{z}(\boldsymbol{x}),\boldsymbol{z}(\boldsymbol{x'})))$

$$
\begin{aligned}
&E(HD(\boldsymbol{z}(\boldsymbol{x}),\boldsymbol{z}(\boldsymbol{x'})))\\
&=n\sum_{I=c}^{m}Pr(\iota(\boldsymbol{x})=I)P(z_j(\boldsymbol{x})=1,z_j(\boldsymbol{x'})=0,\forall j|\iota(\boldsymbol{x})=I)\\
&=n\sum_{I=c}^{m}Pr(\iota(\boldsymbol{x})=I)(f_qP(z_{j_1}(\boldsymbol{x})=1,z_{j_1}(\boldsymbol{x'})=0|\iota(\boldsymbol{x})=I,j_1\in Q(\boldsymbol{x}))\\
&+(1-f_q)(P(z_{j_2}(\boldsymbol{x})=1,z_{j_2}(\boldsymbol{x'})=0|\iota(\boldsymbol{x})=I),j_2\notin Q(\boldsymbol{x})))\\
&\approx n\sum_{I'=0}^{m-c}\sum_{c_w=0}^{c}C_{m-c}^{I'}C_c^{c_w}\hat{f}_p^{I'}(1-\hat{f}_p)^{m-c-I'}f_w^{c_w}(1-f_w)^{c-c_w}(f_q\hat{p}_{c1}(\boldsymbol{x},\boldsymbol{x'},I'+c,c_w)+(1-f_q)\hat{p}_{c2}(\boldsymbol{x},\boldsymbol{x'},I'+c,c_w)).
\end{aligned}
\tag{34}
$$

To estimate the Hamming distance of spiking patterns between non-orthogonal memory items, we introduce $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$ to represent any two different non-orthogonal memory items. We have

$$
\begin{aligned}
&E(HD(\boldsymbol{z}(\boldsymbol{x}_a),\boldsymbol{z}(\boldsymbol{x}_b)))\\
&=\sum_{j=1}^{n}P(z_j(\boldsymbol{x}_a)=1,z_j(\boldsymbol{x}_b)=0)+P(z_j(\boldsymbol{x}_a)=0,z_j(\boldsymbol{x}_b)=1)\\
&=\sum_{j=1}^{n}P(z_j(\boldsymbol{x}_a)=1)P(z_j(\boldsymbol{x}_b)=0)+P(z_j(\boldsymbol{x}_a)=0)P(z_j(\boldsymbol{x}_b)=1)\\
&\approx 2n\hat{p}_{f_{avg}}(\boldsymbol{x}_a,c)(1-\hat{p}_{f_{avg}}(\boldsymbol{x}_a,c)).
\end{aligned}
\tag{35}
$$

Note that because of the common bits, the last equation can only provide an approximate estimate. Our numerical estimates show good consistency using the simple strategy. A more precise estimate can be obtained by further modelling the impact of common bits which has been shown in Eqs.29 and 30.

Details of the analysis of the masking fraction and common bits in Figs 2 and 4. We used $m=25,000, n=39,000, f_p=0.005, f_q=0.005, f_w=0.6$ as the default parameter setting in our simulations. For each specific masking fraction f_d, we generate $\boldsymbol{x'}$ by randomly masking the number of $\lfloor mf_pf_d\rfloor$ 1's bit in $\boldsymbol{x}$.

We run each experiment 10 trials and reported the average results in this experiment. In Fig.2, the size of the memory trace shown in C can be estimated by np_{avg} , the overlap of memory traces shown in D can be estimated by p_{avg} and the fraction of reactivated neuron can be estimated by $1-HD(\boldsymbol{z}(\boldsymbol{x}),\boldsymbol{z}(\boldsymbol{x'}))/p_{avg}$, where p_{avg} is defined by Eq.24 and $HD(\boldsymbol{z}(\boldsymbol{x}),\boldsymbol{z}(\boldsymbol{x'}))$ is defined by Eq.25. The ratio $HD(\boldsymbol{z}(\boldsymbol{x}),\boldsymbol{z}(\boldsymbol{x'}))/HD(\boldsymbol{z}(\boldsymbol{x}_a),\boldsymbol{z}(\boldsymbol{x}_b))$ can be obtained by the quotient of Eq.25 to Eq.26.

In Fig.4, we generated the specific number of non-orthogonal memory items for each specific fraction f_c. In particular, we first calculate the corresponding common 1s bits $c:=\lfloor mf_cf_p\rfloor$ for all patterns and generated the remaining bits with probability $\hat{f}_p=(mf_p-c)/(m-c)$ such that the average 1s element of each pattern remains the same as mf_p used for orthogonal memory items. We generated the corresponding data sets for each specific value of c and randomly masked 33% elements of each non-orthogonal memory item for testing. The predicted ratio was obtained from Eqs. 35 and 34.

In Figs 2 and 4, We selected the optimal v_{th} by minimizing the predicted ratio with the masking fraction 33% and the constrain where the mean population activity of CA1 neurons for the complete patterns falls within a reasonable range [0.05%,10%]. In the simulation, we sampled M at intervals of 2000 and sampled the masking fraction and the fraction of common 1's bits at intervals of 0.02.

Details of the setup of the scaling experiment in Fig.5. For each specific scaling factor γ defined in the x-axis, we simulated the model with $m = \gamma * 25,000, n = \gamma * 39,000$ and used other default hyper-parameters in this experiment(i.e., $f_p = 0.005, f_q = 0.005, f_w = 0.6$). For each scale, we searched for two different optimal thresholds v_{th} in the range $M \in [1,000, 40,000]$ and $M \in [40,000, 100,000]$, respectively. The optimisation criterion follows the experimental setup described above, where the predicted ratio was minimised with the maximum number of patterns within this range and the constraint of the mean population activity of CA1 neurons. For each specific masking fraction f_d, we first calculate the masked bits d for each pattern and randomly masked d bits in the support set accordingly. To obtain the data points for simulation, we sampled M at intervals of 2000 and sampled the simulated scale at intervals of 0.05. We reported the average simulation results over 10 runs.

Prediction of $HD(\boldsymbol{z}(\boldsymbol{x}), \boldsymbol{z}(\boldsymbol{x'})))$ in Fig. 5A is obtained from Eq.25. Predictions of $HD(\boldsymbol{z}(\boldsymbol{x}_a), \boldsymbol{z}(\boldsymbol{x}_b)))$ in Fig. 5B are obtained from Eqs.25 and 26. Prediction in Fig. 5C can be simply derived by using the above two results.

Details of the comparison with classical Hopfield network in Fig.3. We built the BTSP model with the default parameter configuration and build the HFN with the same input size and connection probability. To establish the HFN with sparse binary inputs, we followed the previous studiesHopfield (1982); Sharma et al. (2022) and formalized the weight matrix $\boldsymbol{W}^H$ of the HFN by

$$w_{ji}^H = \begin{cases} 0, & i = j, \\ \frac{1}{m}\sum_{h=1}^{M}(x_{hi} - f_p)(x_{hj} - f_p), & i \neq j. \end{cases} \tag{36}$$

To generate a HFN with the sparsely-connected structure, we first generated a fixed random binary matrix based on the connection probability f_w and multiply it with w_{ji}^H after each weight update. We searched for the optimal threshold of the state update function from $[0, 0.01]$ with the minimum step 0.001. By referring to the memory capacity defined in Gerstner et al. (2014), we adopted an optimal criterion that maximizes the number of patterns that HFN can recall correctly within the error tolerance $m * 0.001$. We used the resulting optimal threshold 0.001 in Fig. 3. We used the synchronous update rule to iterate the state of all neurons 100 times for retrieval. To obtain the data points for numerical simulations we sampled M at intervals of 1000 and sampled the masking fraction at intervals of 0.02.

Details of the analysis of the repulsion effects. For the similarity curve of overlapping patterns, we first initialised a binary weight $\boldsymbol{W}$ with the probability of non-zeros entries ω by Bernoulli sampling. We used $m = 25,000, n = 39,000, f_p = 0.005, f_q = 0.02, f_w = 0.6, v_{th} = 60, \omega = 0.05$ in this experiment.

In Fig. 6C, we considered the case that a BTSP model learns a group of data sequentially and recorded the change of overlapping similarity. Specifically, we first generated 1000 orthogonal memory patterns with the probability f_p. We then randomly chose one pattern and accordingly generated an overlapping pattern with the 40% overlap ratio, and one pattern without overlapping to the chosen pattern. The generated patterns, as well as orthogonal memory items, are randomly fed into the network and BTSP is applied to each pattern. After learning, we calculate the overlapping similarity between the selected patterns and the two specific patterns separately. All trials were averaged over 20 runs with different random initialised parameters.

Details of the derivations of the overlap ratio of memory traces for a pair of similar patterns. Suppose $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$ are two random input patterns with c common bits and other $I - c$ bits of different 1s. $\boldsymbol{W}^R$ is a random binary weight matrix with a probability of ω for non-zeros entries.

The overlap ratio of memory traces for the two input patterns can be formalized by

$$\frac{E(\boldsymbol{z}^T(\boldsymbol{x}_a) \times \boldsymbol{z}(\boldsymbol{x}_b))}{\frac{1}{2}(E(\|\boldsymbol{z}(\boldsymbol{x}_a)\|) + E(\|\boldsymbol{z}(\boldsymbol{x}_b)\|))} = \frac{\sum_{j=1}^{n} E(z_j(\boldsymbol{x}_a) z_j(\boldsymbol{x}_b))}{E(\|\boldsymbol{z}(\boldsymbol{x}_a)\|)} = \frac{\sum_{j=1}^{n} Pr(z_j(\boldsymbol{x}_a) z_j(\boldsymbol{x}_b))}{\sum_{j=1}^{n} Pr(z_j(\boldsymbol{x}_a))}. \tag{37}$$

where $\|\boldsymbol{z}(\boldsymbol{x}_a)\|$ denotes L_1 norm of $\boldsymbol{z}(\boldsymbol{x}_a)$.

We next estimate the probability $P(z_j(\boldsymbol{x}_a)z_j(\boldsymbol{x}_b)=1)$. To decompose $P(z_j(\boldsymbol{x}_a)z_j(\boldsymbol{x}_b)=1)$ into tractable and basic probabilistic factors, we rewrite the expression of $u_j(\boldsymbol{x}_a)$ by dividing the weighted sum $u_j(\boldsymbol{x}_a)$ into two parts, $o_j(\boldsymbol{x}_a,\boldsymbol{x}_b)$ and $r_j(\boldsymbol{x}_a,\boldsymbol{x}_b)$

$$o_j(\boldsymbol{x}_a,\boldsymbol{x}_b) := \sum_{i\in supp(\boldsymbol{x}_a)\cap supp(\boldsymbol{x}_b)} w^R_{ji}x_i, \quad r_j(\boldsymbol{x}_a,\boldsymbol{x}_b) := \sum_{i\in supp(\boldsymbol{x}_a)-supp(\boldsymbol{x}_b)} w^R_{ji}x_i = u_j(\boldsymbol{x}_a,\boldsymbol{x}_b) - o_j(\boldsymbol{x}_a,\boldsymbol{x}_b). \tag{38}$$

On this basis, we decompose the $P(z_j(\boldsymbol{x}_a)z_j(\boldsymbol{x}_b)=1)$ into three probabilistic factors

$$\begin{aligned}
&P(z_j(\boldsymbol{x}_a)z_j(\boldsymbol{x}_b)=1)\\
&=P(u_j(\boldsymbol{x}_a)>v_{th},u_j(\boldsymbol{x}_b)>v_{th})\\
&=\sum_{v=0}^{c}P(u_j(\boldsymbol{x}_a)>v_{th},u_j(\boldsymbol{x}_b)>v_{th},o_j(\boldsymbol{x}_a,\boldsymbol{x}_b)=v), \quad \text{(by the total probability rule)}\\
&=\sum_{v=0}^{c}P(o_j(\boldsymbol{x}_a,\boldsymbol{x}_b)=v)P(r_j(\boldsymbol{x}_a,\boldsymbol{x}_b)>v_{th}-v,r_j(\boldsymbol{x}_b,\boldsymbol{x}_a)>v_{th}-v|o_j(\boldsymbol{x}_a,\boldsymbol{x}_b)=v),\\
&=\sum_{v=0}^{c}P(o_j(\boldsymbol{x}_a,\boldsymbol{x}_b)=v)P(r_j(\boldsymbol{x}_a,\boldsymbol{x}_b)>v_{th}-v)P(r_j(\boldsymbol{x}_b,\boldsymbol{x}_a)>v_{th}-v).
\end{aligned} \tag{39}$$

The above equation indicates that these three factors depend on the number of patterns projected to the position j. We therefore estimate $P(z_j(\boldsymbol{x}_a)z_j(\boldsymbol{x}_b)=1)$ according to the projection of j in three separate cases.

- For any $j\in Q(\boldsymbol{x}_a)\cap Q(\boldsymbol{x}_b)$, we have

$$\begin{aligned}
&P(z_j(\boldsymbol{x}_a)z_j(\boldsymbol{x}_b)=1|j\in Q(\boldsymbol{x}_a)\cap Q(\boldsymbol{x}_b))\\
&=\sum_{v=0}^{c}P(o_j(\boldsymbol{x}_a,\boldsymbol{x}_b)=v)P(r_j(\boldsymbol{x}_a,\boldsymbol{x}_b)>v_{th}-v)P(r_j(\boldsymbol{x}_b,\boldsymbol{x}_a)>v_{th}-v),\\
&=\sum_{v=0}^{c}f(v,c,\omega)(1-F(v_{th}-v,I-c,1-\omega))^2\\
&=\sum_{v=0}^{c}f(v,c,\omega)F^2(v_{th}-v,I-c,\omega) \quad (\text{by } F(v_{th}-v,I-c,1-\omega)+F(v_{th}-v,I-c,\omega)=1).
\end{aligned} \tag{40}$$

- For any $j\in Q(\boldsymbol{x}_a)\cup Q(\boldsymbol{x}_b)-Q(\boldsymbol{x}_a)\cap Q(\boldsymbol{x}_b)$, similarly, we have

$$\begin{aligned}
&P(z_j(\boldsymbol{x}_a)z_j(\boldsymbol{x}_b)=1|j\in Q(\boldsymbol{x}_a)\cup Q(\boldsymbol{x}_b)-Q(\boldsymbol{x}_a)\cap Q(\boldsymbol{x}_b))\\
&=\sum_{v=0}^{c}f(v,c,1-\omega)(1-F(v_{th}-v,I-c,\omega))(1-F(v_{th}-v,I-c,1-\omega)),\\
&=\sum_{v=0}^{c}f(v,c,1-\omega)F(v_{th}-v,I-c,1-\omega)F(v_{th}-v,I-c,\omega).
\end{aligned} \tag{41}$$

- For any $j\notin Q(\boldsymbol{x}_a)\cup Q(\boldsymbol{x}_b)$, we have

$$P(z_j(\boldsymbol{x}_a)z_j(\boldsymbol{x}_b)=1|j\notin Q(\boldsymbol{x}_a)\cup Q(\boldsymbol{x}_b))=\sum_{v=0}^{c}f(v,c,\omega)F^2(v_{th}-v,I-c,1-\omega). \tag{42}$$

Combining the above three cases then gives

$$
\begin{aligned}
&E(z_j(\boldsymbol{x}_a)z_j(\boldsymbol{x}_b)) \\
&= P(z_j(\boldsymbol{x}_a)z_j(\boldsymbol{x}_b) = 1) \\
&= f_q^2 P(z_j(\boldsymbol{x}_a)z_j(\boldsymbol{x}_b) = 1 | j \in Q(\boldsymbol{x}_a) \cap Q(\boldsymbol{x}_b)) + (1-f_q)^2 P(z_j(\boldsymbol{x}_a)z_j(\boldsymbol{x}_b) = 1 | j \notin Q(\boldsymbol{x}_a) \cup Q(\boldsymbol{x}_b)) \\
&+ 2f_q(1-f_q)P(z_j(\boldsymbol{x}_a)z_j(\boldsymbol{x}_b) = 1 | j \in Q(\boldsymbol{x}_a) \cup Q(\boldsymbol{x}_b) - Q(\boldsymbol{x}_a) \cap Q(\boldsymbol{x}_b)) \\
&= \sum_{v=0}^{c} f(v,c,\boldsymbol{\omega}) \left(f_q^2 F^2(v_{th}-v, I-c, \boldsymbol{\omega}) + (1-f_q)^2 (F^2(v_{th}-v, I-c, 1-\boldsymbol{\omega})) \right) \\
&+ 2\sum_{v=0}^{c} f(v,c,1-\boldsymbol{\omega}) f_q(1-f_q) F(v_{th}-v, I-c, 1-\boldsymbol{\omega}) F(v_{th}-v, I-c, \boldsymbol{\omega}).
\end{aligned}
\tag{43}
$$

The estimate of $Pr(z_j(\boldsymbol{x}_a))$ can be easily derived by following the derivation of Eqs.40-43

$$
\begin{aligned}
&E(z_j(\boldsymbol{x}_a)) \\
&= P(z_j(\boldsymbol{x}_a) = 1) \\
&= \sum_{v=0}^{c} f(v,c,\boldsymbol{\omega}) \left(f_q^2 F(v_{th}-v, I-c, \boldsymbol{\omega}) + (1-f_q)^2 (F(v_{th}-v, I-c, 1-\boldsymbol{\omega})) \right) \\
&+ \sum_{v=0}^{c} f(v,c,1-\boldsymbol{\omega}) f_q(1-f_q) \left(F(v_{th}-v, I-c, 1-\boldsymbol{\omega}) + F(v_{th}-v, I-c, \boldsymbol{\omega}) \right).
\end{aligned}
\tag{44}
$$

Substituting Eq.43 and Eq.44 into Eq.37 then gives the estimate of the overlap ratio of memory traces.

Acknowledgments. We would like to thank Jeffrey C. Magee, Aaron D. Milstein, and Sandro Romani for their helpful comments on our manuscript. This research was partially supported by the Human Brain Project (Grant Agreement number 785907) of the European Union and a grant from Intel.

References

Amit, D. J. and Fusi, S. (1994). Learning in neural networks with material synapses. *Neural Computation*, 6(5):957–982.

Andersen, P., Morris, R., Amaral, D., Bliss, T., and O'Keefe, J. (2007). *The hippocampus book*. Oxford university press.

Bittner, K. C., Grienberger, C., Vaidya, S. P., Milstein, A. D., Macklin, J. J., Suh, J., Tonegawa, S., and Magee, J. C. (2015). Conjunctive input processing drives feature selectivity in hippocampal ca1 neurons. *Nature neuroscience*, 18(8):1133–1142.

Bittner, K. C., Milstein, A. D., Grienberger, C., Romani, S., and Magee, J. C. (2017). Behavioral time scale synaptic plasticity underlies ca1 place fields. *Science*, 357(6355):1033–1036.

Chanales, A. J., Oza, A., Favila, S. E., and Kuhl, B. A. (2017). Overlap among spatial memories triggers repulsion of hippocampal representations. *Current Biology*, 27(15):2307–2317.

Cone, I. and Shouval, H. Z. (2021). Behavioral time scale plasticity of place fields: Mathematical analysis. *Frontiers in computational neuroscience*, 15:640235.

Dasgupta, S., Stevens, C. F., and Navlakha, S. (2017). A neural algorithm for a fundamental computing problem. *Science*, 358(6364):793–796.

Gerstner, W., Kistler, W. M., Naud, R., and Paninski, L. (2014). *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge University Press.

Grienberger, C. and Magee, J. C. (2022). Entorhinal cortex directs learning-related changes in ca1 representations. *Nature*, pages 1–9.

Guzman, S. J., Schlögl, A., Frotscher, M., and Jonas, P. (2016). Synaptic mechanisms of pattern completion in the hippocampal ca3 network. *Science*, 353(6304):1117–1123.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558.

Hopfield, J. J. (2007). Hopfield network. *Scholarpedia*, 2(5):1977.

Krotov, D. and Hopfield, J. J. (2020). Large associative memory problem in neurobiology and machine learning. In *International Conference on Learning Representations*.

Milstein, A. D., Li, Y., Bittner, K. C., Grienberger, C., Soltesz, I., Magee, J. C., and Romani, S. (2021). Bidirectional synaptic plasticity rapidly modifies hippocampal representations. *Elife*, 10:e73046.

Priestley, J. B., Bowler, J. C., Rolotti, S. V., Fusi, S., and Losonczy, A. (2022). Signatures of rapid plasticity in hippocampal ca1 representations during novel experiences. *Neuron*.

Sharma, S., Chandra, S., and Fiete, I. (2022). Content addressable memory without catastrophic forgetting by heteroassociation with a fixed scaffold. In *International Conference on Machine Learning*, pages 19658–19682. PMLR.

Wammes, J., Norman, K. A., and Turk-Browne, N. (2022). Increasing stimulus similarity drives nonmonotonic representational change in hippocampus. *ELife*, 11:e68344.

Wanjia, G., Favila, S. E., Kim, G., Molitor, R. J., and Kuhl, B. A. (2021). Abrupt hippocampal remapping signals resolution of memory interference. *Nature communications*, 12(1):1–11.

Widrich, M., Schäfl, B., Pavlović, M., Ramsauer, H., Gruber, L., Holzleitner, M., Brandstetter, J., Sandve, G. K., Greiff, V., Hochreiter, S., et al. (2020). Modern hopfield networks and attention for immune repertoire classification. *Advances in Neural Information Processing Systems*, 33:18832–18845.

Zheng, L., Gao, Z., McAvan, A. S., Isham, E. A., and Ekstrom, A. D. (2021). Partially overlapping spatial environments trigger reinstatement in hippocampus and schema representations in prefrontal cortex. *Nature Communications*, 12(1):6231.

Appendix

A. Proof of Lemma 1.

Proof. Recall that $f(k_x, N_x, p_x)$ be the probability of a binomial distribution that a random variable X gets k_x successes in N_x Bernoulli trials. Note that

$$((1-p_x)+p_x)^{N_x} = \sum_{k_b=0}^{N_x} C_{N_x}^{k_b} p_x^{k_b} (1-p_x)^{N_x-k_b} = P(X=0)+P(X=1)+\cdots+P(X=N_x), \tag{45}$$

$$((1-p_x)-p_x)^{N_x} = \sum_{k_b=0}^{N_x} C_{N_x}^{k_b} (-p_x)^{k_b} (1-p_x)^{N_x-k_b} = P(X=0)-P(X=1)+\cdots+(-1)^{N_x}P(X=N_x). \tag{46}$$

Combining the two equations then gives

$$P(X \text{ is even}) = \frac{1}{2}(1+(1-2p_x)^{N_x}), \tag{47}$$

$$P(X \text{ is odd}) = \frac{1}{2}(1-(1-2p_x)^{N_x}). \tag{48}$$

□