

Spike-frequency adaptation provides a long short-term memory to networks of spiking neurons

Short title: Short-term memory using spike-frequency adaptation

Darjan Salaj^{1,†}, Anand Subramoney^{1,†}, Ceca Krašniković^{1,†}, Guillaume Bellec^{1,2}, Robert Legenstein¹, Wolfgang Maass^{1,*}

May 7, 2020

¹*Institute of Theoretical Computer Science, Graz University of Technology, Inffeldgasse 16b, Graz, Austria*

²*Laboratory of Computational Neuroscience, Ecole Polytechnique Fédérale de Lausanne (EPFL), Bâtiment AAB, offices 135-141, CH-1015 Lausanne*

[†] *These authors contributed equally to this work.*

^{*} *To whom correspondence should be addressed; E-mail: maass@igi.tugraz.at.*

Abstract

Brains are able to integrate memory from the recent past into their current computations, seemingly without effort. This ability is critical for cognitive tasks such as speech understanding or working with sequences of symbols according to dynamically changing rules. But it has remained unknown how networks of spiking neurons in the brain can achieve that. We show that the presence of neurons with spike frequency adaptation makes a significant difference: Their inclusion in a network moves its performance for such computing tasks from a very low level close to the level of human performance. While artificial neural networks with special long short-term memory (LSTM) units had already reached such high performance levels, they lack biological plausibility. We find that neurons with spike-frequency adaptation, which occur especially frequently in higher cortical areas of the human brain, provide to brains a functional equivalent to LSTM units.

Introduction

Our brains are able to constantly process new information in the light of recent experiences and dynamically changing rules, seemingly without any effort. But we do not know how networks of spiking neurons (SNNs) in the brain accomplish that. The performance of both spike-based and rate-based models for recurrent neural networks in the brain have stayed on a rather low performance level for such tasks, far below the performance level of the human brain and artificial neural network models. Artificial neural network models that perform well on such tasks use, instead of neuron models, a special type of unit called Long Short-Term Memory (LSTM) unit. LSTM units store information in registers — like a digital computer — where it remains without perturbation by network activity for an indefinite amount of time, until it is actively updated or recalled. Hence these LSTM units are not biologically plausible, and it has remained an open problem how neural networks in the brain achieve so high performance on cognitively demanding tasks that require integration of information from the recent past into current computational processing. We propose that the brain achieves this — at least for some tasks — without separating computation and short-term memory in different network modules: Rather it intertwines computing and memory with the help of inherent slow dynamic processes in neurons and synapses.

Arguably the most prominent internal dynamics of neurons on the time scale of seconds — which is particularly relevant for many cognitive tasks — is spike-frequency adaptation (SFA). It

42 is expressed by a substantial fraction of neocortical neurons (Allen Institute, 2018). SFA reduces
43 the excitability of a neuron in response to its firing, see Fig. 1. Neurons with SFA have often
44 been included in SNN models that aim at modelling the dynamics of brain networks (Gutkin and
45 Zeldenrust, 2014), but not in computational studies. We show that neurons with SFA do in fact
46 significantly enhance the computational power of SNNs. This is somewhat surprising, because on
47 first sight their history dependence, which even varies strongly from neuron to neuron, tends to
48 obstruct — rather than enhance — network computations. We propose that this may hold for hand-
49 constructed circuits, whereas evolutionary and learning processes are able to exploit advantages
50 of such diverse forms of SFA. We demonstrate this in SNN models for a series of demanding
51 benchmark tasks for network computations that all require integration of information over time:
52 Recalling features of fleeting sensory inputs, speech recognition, time series classification, and
53 operations on sequences of symbols.

54 We also compare the performance of SNNs with SFA to the performance of SNNs that have a
55 different type of slow hidden dynamics, although on a smaller time scale — short-term plasticity
56 (STP) of synapses. But the contribution of synaptic short-term plasticity — especially synaptic
57 facilitation — to computational performance turns out to be lower. Interestingly, the most common
58 form of STP in synapses between pyramidal cells, synaptic depression, tends to provide better
59 support for such computations than synaptic facilitation. References to the related literature can
60 be found in the Discussion.

61 Altogether our results suggest that neurons with SFA provide to SNNs a similar performance
62 boost for computations that require a long short-term memory as LSTM units do for artificial
63 neural networks. Hence we refer to SNNs that contain neurons with SFA as Long short-term
64 memory SNNs (LSNNs). Since the term short-term memory is more common in the literature on
65 LSTM networks, but the term working memory is more common in the neuroscience literature,
66 and both appear to refer to the same phenomena, we treat these two notions as synonyms and let
67 their use depend on the context.

68 SNNs are currently of high interest not only for modelling neural networks of the brain, but
69 also as a computing paradigm for drastically more energy-efficient computer hardware. Hence it is
70 of interest to see that the performance of LSTM networks, and thereby many recent achievements
71 in Artificial Intelligence, can be ported to spike-based computing hardware.

72 Results

73 Experimental data and a simple model for SFA

74 The SFA of a neuron is usually measured in terms of the gradual increase of interspike intervals in
75 its spike response to a constant input drive. An example for such measurement is the Adaptation
76 Index (AI) that is employed by the Allen Institute (Allen Institute, 2018), see Fig. 1A for samples
77 of neurons with different AI, Fig. 1B for the distribution of AI values, and the Methods for the
78 definition of the AI. These data suggest that the human neocortex has a larger fraction of neurons
79 with SFA than the mouse neocortex, and that within the human brain the frontal lobe has a
80 larger fraction than the temporal gyrus. The analysis of experimental data in (Pozzorini et al.,
81 2013, 2015) lead to the conclusion that SFA takes place on multiple time scales, with a history
82 dependence that lasts up to 20 s in neocortical neurons. Various models for adapting neurons
83 have been proposed in (Gerstner et al., 2014; Teeter et al., 2018). We employ a very simple model
84 for SFA, the generalized leaky integrate-and-fire model $GLIF_2$ from (Teeter et al., 2018; Allen
85 Institute, 2017), which we will refer to as the ALIF (adaptive LIF) model. A practical advantage
86 of this simple model is that it can be very efficiently simulated and is amenable to gradient descent
87 training methods. It assumes that the firing threshold $A(t)$ contains a variable component $a(t)$ that
88 increases by a fixed amount after each of its spikes $z(t)$ (Fig. 1C), and then decays exponentially
89 back to 0. This variable threshold models the inactivation of voltage-dependent sodium channels
90 in a qualitative manner. We write $z_j(t)$ for the spike output of neuron j , that switches from 0 to
91 1 at time t when the neuron fires at time t , and otherwise has value 0. With this notation one can

92 define the ALIF model by the equations:

$$\begin{aligned} A_j(t) &= v_{\text{th}} + \beta a_j(t), \\ a_j(t + \delta t) &= \rho_j a_j(t) + (1 - \rho_j) z_j(t) \delta t, \end{aligned} \tag{1}$$

93 where v_{th} is the constant baseline of the firing threshold $A_j(t)$, and $\beta > 0$ scales the amplitude of
94 the activity-dependent component. The parameter $\rho_j = \exp\left(\frac{-\delta t}{\tau_{a,j}}\right)$ controls the speed by which
95 $a_j(t)$ decays back to 0, where $\tau_{a,j}$ is the adaptation time constant and δt is the duration of a
96 discrete time step (which we chose to be 1 ms). An LSNN is a network of spiking neurons that
97 contains some ALIF neurons (see Methods for details on neuron and synapse models). It typically
98 suffices to use ALIF neurons with some spread of time constants $\tau_{a,j}$ around the required duration
99 of working memory for solving a task (see Table S1 in Supplement for details on how the choice
100 of adaptation time constant impacts performance).

101 Methods for training recurrent SNNs

102 We focused on the best performing training method for recurrent SNNs that is currently known:
103 Backpropagation through time (*BPTT*) with the pseudo-derivative for spiking neurons from (Bellec
104 et al., 2018b). While *BPTT* is not assumed to be biologically plausible as an online learning
105 method, results from training with *BPTT* inform us about computational capabilities of different
106 types of SNNs. They also inform us about performance levels that could in principle be attained
107 through evolution. In order to test whether complex cognitive computations, such as the 12AX
108 task, can in principle also be learnt by brain networks, we also trained the same LSNN with a
109 biologically plausible learning method: *e-prop*. For LSNNs, *e-prop* tends to achieve in general an
110 almost as good computational performance level as *BPTT* (Bellec et al., 2019).

111 SFA provides a functionally powerful working memory for spike-based 112 computing

113 Our brains are able to recall an image, even after having seen many other images in between. We
114 wondered whether LSNNs would be able to model such fundamental working memory task. Note
115 that remembering an image requires retaining substantially more than a single bit, even if it is
116 encoded in a highly compressed form in a higher cortical area. In contrast, most models for working
117 memory have focused on retaining just a single bit, and this memory content occurred during
118 training and testing. We formulated our more demanding computational task as the STORE-
119 RECALL task illustrated in Fig. 2. The network received a sequence of frames, each consisting of
120 a vector of 20 binary features — arranged in a 4×5 grid (top of Fig. 2) — which were presented
121 for 200 ms. Each frame can be seen as corresponding to the compressed representation of an
122 image in a higher visual area such as IT. In addition, the network received occasional STORE and
123 RECALL commands, marked in yellow and green in Fig. 2. The STORE command corresponds
124 to directing attention to a particular frame of the input stream. The computational task was
125 to reproduce, during a RECALL command, the feature vector that had been presented during
126 the preceding STORE command. The delay between the STORE and RECALL commands was
127 randomly chosen from the interval between 200 and 1600 ms.

128 We trained an LSNN that consisted of 500 ALIF neurons, whose firing thresholds had time
129 constants of 800 ms, to solve this task. Sigmoidal readout neurons, one for each of the 20 binary
130 input features, were trained to reproduce the feature value that had been present during STORE.
131 Binary feature values were extracted by rounding the activity of readout neurons at the half-way
132 (100 ms) mark of each 200 ms time window. A sample segment of a test trial is shown in Fig. 2,
133 with the activity of input neurons at the top and the activation of readout neurons at the bottom.
134 In order to probe the generalization capability of the LSNN we made sure that none of the patterns
135 shown during testing had occurred during training, and in fact had a Hamming distance of at least
136 5 bits to all training patterns. Note that previous models for working memory only aimed at
137 storing a single bit, and the model could only be tested for the same content for which it was
138 trained. Here we require that the working memory can be used for content other than what was
139 used during training. The resulting recall performance of the LSNN was 99.09%, i.e., 99.09% of

140 the stored feature vectors were perfectly reproduced during recall. This demonstrates that LSNNs
141 have inherent high-dimensional working memory capabilities. SFA was essential for this, because
142 the recall performance of a recurrent network of LIF neurons without SFA, trained in exactly
143 the same way, stayed at chance level (see Methods). A closer inspection of the time course of
144 firing thresholds of a sample subset of neurons in the LSNN provides insight into how LSNNs are
145 able to solve this task: A pattern-specific subset S of neurons is highly activated during STORE,
146 which raises their firing thresholds (shown as blue curves in Fig. 2). Many neurons are activated
147 again during RECALL, but the firing activity of neurons in the subset S remains lower this time,
148 thereby providing a negative imprint of their activation pattern during STORE. Readout neurons
149 can easily be trained to decode these negative imprints, and to reproduce the originally stored
150 pattern.

151 Interestingly, the firing activity of the network was rather low during the delay between STORE
152 and RECALL. Furthermore we found that a Support Vector Machine (SVM) was not able to
153 decode the stored feature vector from the firing activity of the LSNN neurons during the delay
154 (the decoding accuracy during the delay was 4.38%, as opposed to 100% decoding accuracy during
155 RECALL; see Methods). Hence the type of working memory that an LSNN exhibits corresponds
156 to the activity-silent form of working memory in the human brain that had been examined in
157 the experiments of (Wolff et al., 2017). It had also been shown there that the representation of
158 working memory content changes drastically between memory encoding and subsequent network
159 reactivation during the delay by an “impulse stimulus”: A classifier trained on the network activity
160 during encoding was not able to classify the memory content during a network reactivation, and vice
161 versa. The same holds for our LSNN model (see Methods), since the reactivation of the network
162 during RECALL provides a negative, rather than a positive imprint, of the high-dimensional
163 memory content.

164 We also examined how the time constants of the thresholds of ALIF neurons should be chosen
165 to achieve good performance for a task that requires a particular time span of working memory.
166 We studied this for a variant of the STORE-RECALL task where the time-varying input vector of
167 features was just 1D instead of 20D, but where the expected delays between STORE and RECALL
168 varied between 0.2 to 16 s for different versions of the task. It turned out that good performance
169 did not require a tight coupling between the required length of working memory and the adaptation
170 time constants of ALIF neurons in the LSNN, see Table S1. In particular, good working memory
171 performance was also achieved when the required time span for working memory was substantially
172 larger than these time-constants, suggesting that the network had learned to automatically refresh
173 or stagger the implicit memory in firing thresholds of different neurons. We also verified that good
174 performance for many memory retention time spans could be achieved by a single network with
175 a mixture of time constants of firing thresholds drawn from a uniform or power-law distribution.
176 This suggests that brains can solve working memory tasks for many different retention spans using
177 SFA neurons with a generic spread of time constants.

178 Finally, we wondered whether the adaptive firing threshold of ALIF neurons affects the auto-
179 correlation function of their firing activity — termed intrinsic timescale in (Wasmuht et al., 2018).
180 We tested this for an LSNN consisting of 200 LIF and 200 ALIF neurons that was trained to solve
181 a 1D version of the STORE-RECALL task. It turned out that during the delay between STORE
182 and RECALL these intrinsic time constants were in the same range as those measured in monkey
183 cortex, see Fig. 1C in (Wasmuht et al., 2018). Furthermore LIF and ALIF neurons exhibited very
184 similar distributions of these time constants (see Fig. S1), suggesting that these intrinsic time
185 constants are determined largely by their network inputs, and less by the neuron type.

186 **Working memory performance of variants of SNNs with other slow pro-** 187 **cesses in neurons or synapses**

188 There exist numerous other slow processes on the time scale of seconds in neurons and synapses,
189 that can potentially also enhance network computations on this time scale. We examined the
190 performance of three other candidates besides SFA on a simple version of the STORE-RECALL
191 task:

- 192 • LIF neurons whose excitability gets increased through their firing: ELIF neurons

- 193 • Depressing short-term plasticity of synapses (STP-D)
- 194 • Facilitating short-term plasticity of synapses (STP-F).

195 The ELIF neuron is a dual version of an ALIF neuron whose excitability is increased through
196 preceding firing, rather than decreased (see Methods). ELIF neurons appear to be particularly
197 suitable for creating a working memory through persistent firing. Facilitating short-term plasticity
198 of synapses also supports that, and was conjectured by (Mongillo et al., 2008) to produce a working
199 memory. We also evaluated the performance of depressing short-term plasticity, because this is
200 the standard dynamics of synapses between pyramidal cells (Markram et al., 2015). The dynamics
201 of the salient hidden variables in these three models is illustrated in Fig. S2. The performance of
202 corresponding variants of the SNN is shown in Fig. 3A for a 1D variant of the STORE-RECALL
203 task from Fig. 2, with a delay between STORE and RECALL commands varying between 200 ms
204 and 3600 ms. It turns out that SNNs with ALIF neurons, i.e., LSNNs, learn to solve this task
205 much faster than the other variants of the SNN model, and also reach the highest performance
206 level. Furthermore only the networks with STP-D or ELIF neurons eventually approach reasonably
207 good — although lower — performance levels. We were surprised to see that facilitating short-term
208 plasticity of synapses (STP-F) did not provide the working memory capability needed to solve this
209 task, although we used here a really long time constant for facilitation with a mean of 2000 ms —
210 much larger than the mean of 507 ms that had been found experimentally in (Wang et al., 2006)
211 for synaptic connections between pyramidal cells in the PFC. Similar results hold for the time
212 series classification task sMNIST, see Fig. 3C and the subsection on sMNIST below.

213 We also found that replacing ALIF by ELIF neurons reduced the working memory capability
214 of the network for both tasks, see Fig. 3A and C. One possible reason is that information that
215 is stored in the firing threshold of a neuron is better protected in the case of an ALIF neuron,
216 since an increased firing threshold suppresses subsequent accidental firing, and hence accidental
217 modifications of the memory that is stored in the firing threshold. In contrast, for an ELIF neuron
218 the information that is stored in the firing threshold is quite vulnerable, since a decreased firing
219 threshold invites accidental firing.

220 Performance of LSNNs for speech recognition and other benchmark tasks 221 that require substantial integration of information over time

222 **Google Speech Commands dataset.** We trained LSNNs and networks of LIF neurons on
223 the keyword spotting task with Google Speech Commands Dataset (Warden, 2018) (v0.02). The
224 dataset consists of 105,000 audio recordings of people saying thirty different words. Fully connected
225 networks were trained to classify audio recordings, that are clipped to one second length, into one
226 of 12 classes (10 keywords, as well as two special classes for silence and unknown words; the
227 remaining 20 words had to be classified as “unknown”). Comparison of maximum performance of
228 trained spiking networks against state-of-the-art artificial recurrent networks is shown in Table 1.
229 Averaging over 5 runs, the LSNN reached $90.88 \pm 0.22\%$, and the LIF network reached $88.79 \pm 0.16\%$
230 accuracy. Thus an SNN without ALIF neurons can already solve this task quite well, but the LSNN
231 halves the performance gap to the published state-of-the-art in machine learning. The only other
232 report on a solution of this task with spiking networks is (Cramer et al., 2019). There the authors
233 encode the audio features to spike trains using cochlea model and train a network of LIF neurons
234 using surrogate gradients with *BPTT* and achieve $50.9 \pm 1.1\%$ accuracy.

235
236 **Delayed-memory XOR task.** We also tested the performance of LSNNs on a previously con-
237 sidered benchmark task for SNNs, where two items in the working memory have to be combined
238 non-linearly: The Delayed-memory XOR task (Huh and Sejnowski, 2018). The network is required
239 to compute the exclusive-or operation on the history of input pulses when prompted by a go-cue
240 signal, see Fig. 3B.

241 The network receives on one input channel two types of pulses (up or down), and a go-cue on
242 another channel. If the network received two input pulses since the last go-cue signal, it should
243 generate the output “1” during the next go-cue if the input pulses were different or “0” if the input
244 pulses were the same. Otherwise, if the network only received one input pulse since the last go-cue
245 signal, it should generate a null output (no output pulse). Variable time delays are introduced

246 between the input and go-cue pulses. Time scale of the task was 600 ms which limited the delay
247 between input pulses to 200 ms.

248 This task was solved in (Huh and Sejnowski, 2018), without providing a performance statistics,
249 by using a type of neuron that has not been documented in biology — a non-leaky quadratic
250 integrate and fire neuron. We are not aware of previous solutions by networks of LIF neurons. To
251 compare and investigate the impact of SFA on the performance of delayed-memory XOR task, we
252 trained networks of ALIF and LIF neurons of the same size as in (Huh and Sejnowski, 2018) — 80
253 neurons. Across 10 runs, networks of ALIF neurons solved the task with $95.19 \pm 0.014\%$ accuracy,
254 whereas the networks of LIF neurons converged at lower $61.30 \pm 0.029\%$ accuracy.

255

256 **Sequential MNIST (sMNIST).**

257 Finally, we compared the performance of LSNNs and other variants of SNNs with that of LSTM
258 networks on a more demanding benchmark task for time series classification: The classification of
259 pixel-wise sequentially presented handwritten digits (sMNIST dataset), see Fig. 3C,D and Fig. S3.
260 This task requires integration of information over a longer time span than for recognizing speech
261 commands. It also requires very good generalization capability, since the pixel sequences for
262 different handwriting styles of the same digit may vary widely. LSNNs achieved here about the
263 same performance level as LSTM networks, whereas networks that contain only LIF and not
264 ALIF neurons performed poorly, see Fig. 3D. Besides a fully connected LSNN, we also tested the
265 performance of a variant of the model, called SC-LSNN, that integrates additional constraints of
266 SNNs in the brain: It is sparsely connected (12% of possible connections are present) and consists
267 of 75% excitatory and 25% inhibitory neurons that adhere to Dale’s law. By adapting the sparse
268 connections with the rewiring method in (Bellec et al., 2018a) during *BPTT* training, the SC-LSNN
269 was enabled to perform even better than the fully-connected LSNN. The resulting architecture of
270 the SC-LSNN is shown in Fig. S3C. Its activity of excitatory and inhibitory neurons, as well as
271 the time courses of adaptive thresholds for (excitatory) ALIF neurons of the SC-LSNN are shown
272 in Fig. S3B.

273 Fig. 3C shows that, apart from LSNNs, SNNs with experimentally reported parameters for
274 short-term synaptic plasticity (STP-D) also achieve very high performance. Furthermore, SNNs
275 with STP-D perform substantially better for this task than networks with data-based synaptic
276 facilitation (STP-F), similar as for STORE-RECALL (see Fig. 3A).

277 **SFA supports demanding cognitive computations with dynamically chang-** 278 **ing rules**

279 The 12AX task — which can be viewed as a simplified version of the Wisconsin Card Sorting
280 task — tests the capability of subjects to apply dynamically changing rules for pattern recognition
281 and to ignore currently irrelevant inputs (O’Reilly and Frank, 2006; MacDonald III, 2008). It also
282 probes — at least in the more demanding version that we consider — the capability to maintain and
283 update a hierarchical working memory. The task consists of a sequence of trials where the subject
284 is first shown a context cue to indicate which one of two possible sequences is the “correct” symbol
285 sequence in the current trial. These sequences consist of two relevant symbols, interspersed with
286 distractor symbols, including those belonging to the “wrong” sequence in the current context. At
287 every step, the subject has to press one of two buttons depending on whether the correct sequence
288 has been completed or not. The context of the trial switches randomly after a few presentation of
289 symbols.

290 To model this, we gave as network inputs sequences of 90 symbols from the set $\{1, 2, A, B, C,$
291 $X, Y, Z\}$, with repetitions as described in Methods. See the top of Fig. 4 for an example. After
292 each symbol, the network should output “R” if this symbol terminates a context dependent target
293 sequence and “L” otherwise. Specifically, given a context where the most recently received digit
294 was a “1”, the subject should output “R” only after presentation of a symbol X that terminates
295 a subsequence A...X. This occurs, for example, for the 7th symbol in the trial shown in Fig. 4. In
296 case that the most recent input digit was a “2”, the subject should respond “R” only after the
297 symbol Y in a subsequent subsequence B...Y (see the 20th symbol in Fig. 4). The letters C and Z
298 are irrelevant and serve as distractors. This task requires a hierarchical working memory, because
299 the most recently occurring digit determines whether subsequent occurrences of “A” or “B” should

300 be placed into working memory. Note also that neither the content of the higher-level working
301 memory — the digit — nor the content of the lower level working memory — the letter A or B —
302 are simply recalled. Instead, they both affect processing rules, where the higher-level processing
303 rule affects what is placed into the lower level working memory. A simpler version of this task,
304 where X and Y were relevant only if they directly followed A or B respectively, and where fewer
305 irrelevant letters occurred in the input, was solved in (O’Reilly and Frank, 2006) through artificial
306 neural network models that were endowed with special working memory modules. Note that for
307 this simpler version no lower order working memory is needed, because one just has to wait for an
308 immediate transition from A to X in the input sequence, or for an immediate transition from B
309 to Y. But neither the simpler nor the more complex version of the 12AX-task has previously been
310 solved by a network of spiking neurons.

311 We show in Fig. 4 that a generic LSNN can solve this quite demanding version of the 12AX
312 task. The LSNN received spike trains from the input population of spiking neurons, producing
313 Poisson spike trains. Possible input symbols {1, 2, A, B, C, X, Y, Z} were encoded using one-hot
314 coding; each input symbol was signaled through a high firing rate of a separate subset of 5 input
315 neurons for 500 ms. The LSNN consisted of 200 recurrently connected spiking neurons (100 ALIF
316 and 100 LIF neurons), with all-to-all connections between them. The output consisted of two
317 readouts, one for L, one for the R response. During each 500 ms time window the input to these
318 readouts was the average activity of neurons in the LSNN during that time window. The final
319 output symbol was based on which of the two readouts had the maximum value. After training
320 with *BPTT* the LSNN produced an output string with all correct symbols in 97.79% of episodes,
321 where 90 symbols had to be processed during each episode. But also after training with *e-prop*, a
322 biologically realistic learning method (Bellec et al., 2019), the LSNN produced fully correct output
323 sequences in 92.89% of the episodes. In contrast, a recurrent SNN with the same architecture
324 but no neurons with SFA could achieve only 0.39% fully correct output strings after training with
325 *BPTT* (not shown). Note that it was not necessary to create a special network architecture for the
326 two levels of working memory that our more complex version of the 12AX task requires: A near
327 perfectly performing network emerged from training a generic LSNN. This shows that neurons
328 with SFA enable generic recurrent networks of spiking neurons to solve demanding cognitive tasks
329 involving dynamically changing rules and two levels of working memory.

330 SFA supports brain-like operations on sequences

331 A generic difficulty for neural networks is learning to carry out operations on sequences of symbols
332 in such a way that they generalize to new sequences, a fundamental capability of the human brain
333 (Marcus, 2003). Actually, not only humans, but also non-human primates are able to carry out
334 operations on sequences of items, and numerous neural recordings — starting with (Barone and
335 Joseph, 1989) up to recent results such as (Carpenter et al., 2018; Liu et al., 2019) — provide
336 information about the neural codes for sequences that accompany such operations in the brain.
337 One fundamental question is how serial order of items is encoded in working memory. Behind this
338 is the even more basic question of how transient structural information — the serial position of an
339 item — is combined in the brain with content information about the identity of the item (Lashley,
340 1951). Obviously, this question also lies at the heart of open questions about the interplay between
341 neural codes for syntax and semantics that enable language understanding in the human brain.
342 The experimental data both of (Barone and Joseph, 1989) and (Liu et al., 2019) suggest that
343 the brain uses a factorial code where position and identity of an item in a sequence are encoded
344 separately by some neurons, thereby facilitating flexible generalization of learned experience to new
345 sequences. But so far we had been lacking spiking neural network models that were able to carry
346 out such tasks, and whose emergent neural codes could then be compared with neural recordings
347 from the brain. We show here that LSNNs can be trained to carry out complex operations on
348 sequences, are able to generalize such capabilities to new sequences, and produce spiking activity
349 and neural codes that offer interesting links to recorded data.

350 One basic operation on sequences of symbols is remembering and reproducing a given sequence
351 (Liu et al., 2019), a task which non-human primates can also learn, and for which neural codes for
352 sequences have been investigated (Barone and Joseph, 1989; Liu et al., 2019). A more complex
353 operation that can also be carried out by the brain is the reversal of a sequence (Marcus, 2003;

354 Liu et al., 2019). We show that an LSNN learns to carry out both of these operations, and is able
355 to apply them to new sequences of symbols that did not occur during training.

356 We trained an LSNN consisting of 128 LIF and 192 ALIF neurons to carry out these two
357 operations on sequences of 5 symbols from a repertoire of 31 symbols, which we labeled by the
358 letters a, b, c, ..., x, y, z, A, B, C, D, E from the English alphabet. Four additional symbols were
359 used: “*” denoted the end of the input sequence (EOS), “?” a prompt for an output symbol,
360 and one symbol each for the DUPLICATE and REVERSE commands (see Fig. 5A). Each of the
361 altogether 35 input symbols were given to the network in the form of higher firing activity of a
362 dedicated population of 5 input neurons outside of the LSNN (“one hot encoding”). It was not
363 necessary to assign particular values to adaptation time constants of firing thresholds of neurons
364 with SFA; we simply chose them uniformly randomly to be between 1 ms and 6000 ms, mimicking
365 the diversity of SFA effects found in the neocortex (Allen Institute, 2018) in a qualitative manner.
366 The network output was produced by linear readouts (one per potential output symbol, each with
367 a low pass filter with a time constant of 250 ms) that received spikes from neurons in the LSNN,
368 see the row “Output” in Fig. 5A). The final output symbol was selected using the readout which
369 had the maximum value at the end of each 500 ms time window (a softmax instead of the hard
370 argmax was used during training), mimicking winner-take-all computations in neural circuits of
371 the brain (Chettih and Harvey, 2019) in a qualitative manner.

372 After training, an LSNN was able to apply duplication and reversal also to new sequences,
373 achieving a success rate of 0.9588 (average over 5 random seeds) for unseen sequences. The
374 “success rate” was defined as the fraction of test episodes/trials where the full output sequence was
375 generated correctly. Sample episodes of the trained LSNN are shown in Fig. 5A. For comparison,
376 we also trained a LIF network in exactly the same way with the same number of neurons. It
377 achieved a performance of 0.0 (zero).

378 Emergent coding properties of neurons in the LSNN are analyzed in Fig. 5B-F. Neurons are
379 sorted in Fig. 5B,C according to the time of their peak activity (averaged over 1000 episodes), like
380 in (Harvey et al., 2012). A number of network neurons (about one-third) participate in sequential
381 firing activity independent of the type of task and the symbols involved (Fig. 5B). Instead, these
382 neurons have learned to abstract the overall timing of the tasks. This kind of activity is reminiscent
383 of the neural activity relative to the start of a trial that was recorded in rodents after they had
384 learned to solve tasks that had a similar duration (Tsao et al., 2018).

385 The time of peak activity of other neurons depended on the task and the concrete content,
386 see Fig. 5C. Interestingly enough, these neurons change their activation order already during the
387 loading of the input sequence in dependence of the task (duplication or reversal). Using 3-way
388 ANOVA, we were able to categorize each neuron as selective to a specific condition or a non-linear
389 combination of conditions based on the effect size ω^2 . Each neuron could belong to more than one
390 category if the effect size was above the threshold of 0.14 (as suggested by (Field, 2013)). Similar
391 to recordings from the brain (Carpenter et al., 2018), a diversity of neural codes emerged that
392 encode one or several of the variables symbol identity, serial position in the sequence, and type of
393 task. In other words, a large fraction of neurons are mixed-selective, i.e. selective to non-linear
394 combinations of all three variables. Peri-condition time histogram (PCTH) plots of two sample
395 neurons are shown in Fig. 5E,F: One neuron is selective to symbol “g” but at different positions
396 depending on task context. The other neuron is selective to symbol “C” occurring at position 5 in
397 the input, independent of task context. Thus one sees that a realization of this task by an SNN,
398 which was previously not available, provides rich opportunities for a comparison of emergent spike
399 codes in the model and neuronal recordings from the brain.

400 Discussion

401 An important open problem in computational neuroscience is to understand how brains carry out
402 computations that involve not just current cues, but information from the recent past. In fact,
403 brains are able to store not just single bits for subsequent computational use over a time scale of
404 many seconds, but previously experienced images, movie scenes, and dialogues that require a fairly
405 large storage capacity. This problem is usually formulated as a question about the implementation
406 of working memory — or short-term memory — in the brain. But this formulation is somewhat

407 biased against the possibility that computing and short-term memory are so intertwined in neural
408 networks of the brain that it becomes really difficult to separate mechanisms and network modules
409 that hold short-term memory from those that constitute the computational machinery of the
410 network.

411 There already exists fairly wide agreement that different forms of working or short-term memory
412 can be distinguished in the brain, see e.g. (Olivers et al., 2011; Kamiński and Rutishauser, 2019;
413 Masse et al., 2019). Recent experimental data show clearly that, for a highly trained task, discrete
414 attractors of the network dynamics, implemented by persistent firing, hold an intended movement
415 direction in the anterior lateral motor cortex (Inagaki et al., 2019). But the question remains
416 whether the brain uses the same mechanism — especially without extensive training — for holding
417 quickly changing high-dimensional memory content, such as a movie scene, previously read text, or
418 a sequence of images. Neural codes for storing information from a sequences of two images — after
419 extensive training — had been examined in (Warden and Miller, 2007). A complex interaction was
420 found between the memory traces of two sequentially presented images, thereby speaking against
421 an assumption that each is held by a separate discrete attractor in working memory.

422 Several publications argue that the brain uses a variety of mechanisms for working memory,
423 each with its specific advantages and disadvantages, whose engagement depends on the specific
424 task (Olivers et al., 2011; Trübtschek et al., 2017; Kamiński and Rutishauser, 2019; Barbosa
425 et al., 2019; Hu et al., 2020). In particular, (Wolff et al., 2017; Trübtschek et al., 2017; Kamiński
426 and Rutishauser, 2019; Masse et al., 2019; Barbosa et al., 2019) point to an activity-silent form
427 of working memory that is used by the brain for maintaining working memory while it is not in
428 the focus of attention. A model for such activity-silent memory had been proposed already in
429 (Mongillo et al., 2008), based on facilitating short-term plasticity of synapses. This mechanism
430 requires a facilitating short-term plasticity of synapses between excitatory neurons (pyramidal
431 cells), which had previously been discovered in the medial prefrontal cortex of ferret (Wang et al.,
432 2006). However the model of (Mongillo et al., 2008) used a time constant of 1500 ms for the time
433 constant F of facilitation, whereas this parameter for the facilitation-dominant synapse type E1
434 of (Wang et al., 2006) has a reported average value of 507 ms with a standard deviation of 37
435 ms. An experimentally testable prediction of this form of activity-silent working memory is that
436 an unspecific network reactivation between storage and recall would make the content of working
437 memory decodable from the resulting network activity. However, the experimental data of (Wolff
438 et al., 2017) do not support this prediction.

439 We examined in this paper whether the arguably most prominent dynamic feature of neurons
440 on the time-scale of seconds, SFA, supports computations that require a working memory. Exper-
441 imental data show that SFA does in fact produce history-dependence of neural firing on a time scale
442 of several seconds up to 20 seconds (Pozzorini et al., 2013, 2015). We found that this prominent
443 feature of a fairly large fraction of neurons in the neocortex provides an inherent working memory
444 capability to neural networks. Our results suggest that this working memory capability is func-
445 tionally quite powerful, and enables networks of spiking neurons to solve a variety of cognitively
446 demanding tasks that were previously beyond the reach of SNN models. In particular, SFA enables
447 flexible operations on sequences of symbols (Fig. 4, 5). This allows us, for the first time, to study
448 emergent neural codes for symbols and their position in a sequence in a model network of spiking
449 neurons, and to compare them with recordings from neurons in the neocortex for corresponding
450 tasks (Fig. 5). When we compared the contribution of SFA with the contribution of the other two
451 most prominent slow processes in neurons or synapses, synaptic facilitation and synaptic depres-
452 sion, we found that the contribution of SFA is substantially more powerful for a basic working
453 memory task (Fig. 3A). A comparison for a demanding time series classification task with a lower
454 demand on the retention time span (Fig. 3C) suggests that synaptic depression works for such tasks
455 about equally well, but not synaptic facilitation. The good performance of synaptic depression
456 for tasks that require shorter retention time of working memory is consistent with the modelling
457 results of (Masse et al., 2019) and (Hu et al., 2020). However, as already pointed out in (Masse
458 et al., 2019), synaptic depression tends to work best for tasks that require rather short working
459 memory maintenance. Our results are also consistent with the finding of (Masse et al., 2019) that
460 persistent activity is more prominent if the working memory content has to be manipulated, rather
461 than just maintained. Compare the higher firing activity in Fig. S3B for the sMNIST task that
462 requires continuous manipulation of working memory content with the low firing activity in Fig. 2,

463 where the working memory content just has to be maintained. One sees this difference also in the
464 sequence manipulation task of Fig. 5. There the working memory just has to be maintained during
465 the first half of a trial, yielding an average firing rate of 16.6 Hz over all neurons. But this average
466 firing rate increased to 26.7 Hz during the second halves of the trials, where the stored information
467 had to be manipulated (averages taken over 50,000 trials during testing).

468 On first sight one might think that working memory can only be held in neural activity through
469 increased firing. Our results show that it can just as well be attained through decreased firing,
470 which is the way how neurons with SFA provide evidence of preceding strong activation. Whereas
471 this mechanism may be intuitively less plausible, it looks equally viable from the perspective of
472 downstream networks in the brain. Whether preceding firing activity leaves a positive or nega-
473 tive imprint in subsequent firing appears to be of secondary relevance for readout neurons if the
474 downstream integration of evidence involves a weighted sum, since weights can have positive or
475 negative signs. One may argue that there are actually, from the systems-perspective, two benefits
476 in maintaining working memory in the form of a negative imprint, i.e., through decreased excitabil-
477 ity of neurons. One is that encoding working memory through non-firing consumes less energy.
478 Another is that this form of working memory is less vulnerable to disturbances through interven-
479 ing network activity, since a decreased excitability protects a neuron from accidental activation —
480 and hence potential overwriting of its memory content. One first piece of experimental evidence
481 for the negative imprinting hypothesis was provided by the previously mentioned result of (Wolff
482 et al., 2017). It was examined there whether a classifier that had been trained to decode from the
483 network activity the stored memory content during encoding would be able to decode the memory
484 content also during a subsequent network reactivation through an unspecific impulse. The answer
485 was negative, which is consistent with the negative imprinting hypothesis. We confirmed for the
486 task of Fig. 2 that the same holds true for our model with SFA (see subsection “Decoding memory
487 from the network activity” in Methods). It is actually well-known that negative imprinting is used
488 by the brain for a particular type of long-term memory called recognition memory: Familiarity of
489 an object is encoded through reduced firing of a large fraction of neurons in the perirhinal cortex
490 and adjacent areas, see (Winters et al., 2008) for a review.

491 A major structural difference between standard models for neural networks in the brain and
492 artificial neural networks (ANNs) that are used in artificial intelligence and deep learning for solving
493 computational tasks that involve memory from the recent past lies in the type of neurons (units)
494 that are used. Well-performing ANNs usually employ LSTM units or similar units that allow to
495 store a bit or analog variable in a memory register — like in a digital computer — where it is
496 protected from perturbation by ongoing network activity. Our results show — rather surprisingly
497 — that such drastic protection of working memory content is not needed: We showed that almost
498 the same performance can be achieved by LSNNs, i.e., SNNs that contain neurons with SFA. This
499 holds in spite of the fact that memory content that is stored in an adaptive firing threshold of a
500 neuron with SFA is not fully protected from the disturbance through network activity. But it is at
501 least somewhat shielded, because a neuron that holds memory in the form of an increased firing
502 threshold has an inherent tendency not to respond to smaller membrane depolarizations.

503 Our results show that biologically rather realistic models for spiking neural networks in the
504 brain that also contain neurons with SFA reach for demanding cognitive tasks for the first time
505 the performance level of humans, which could previously only be reached with ANNs that employ
506 biologically unrealistic LSTM units. This paves the way for reaching a key-goal of brain modelling
507 — to combine detailed experimentally data from neurophysiology on the level of neurons and
508 synapses with brain-like functionality of the network.

509 Recurrent networks of spiking neurons are also of interest from the perspective of novel comput-
510 ing technology. Spike-based computing hardware has the potential to provide substantially more
511 energy-efficient implementations of artificial intelligence and deep learning results than standard
512 digital hardware. But its performance has so far been significantly inferior to that of non-spiking
513 neural networks. Our results show that this performance gap is becoming quite small for the case
514 of recurrent neural networks if one integrates neurons with SFA into the spike-based network.

515 Altogether, we have shown that a well-known feature of a substantial fraction of neurons in the
516 neocortex — SFA — provides an important new facet to our understanding of computations in
517 SNNs: It enables SNNs to integrate working memory from the recent past seamlessly into ongoing
518 network computations.

519 Materials and Methods

520 Table of Contents:

- 521 • Adaptation Index
- 522 • Network models
- 523 • Training methods
- 524 • Tasks

525 Adaptation index

The adaptation index (AI) is a quantitative measure of firing rate adaptation that has been recorded for a wide variety of cells in the Allen institute database (Allen Institute, 2018). It measures the rate at which firing of a spiking neuron speeds up or slows down when the neuron is fed with a step current of 1 second. Given the induced spike times, it is defined as:

$$\frac{1}{N-1} \sum_{n=1}^{N-1} \frac{\text{ISI}_{n+1} - \text{ISI}_n}{\text{ISI}_{n+1} + \text{ISI}_n},$$

526 where ISI_n is n-th inter spike interval (ISI) and N is the number of ISIs induced during the stimulus
527 duration. Hence regular doubling of the ISI produces for example $\text{AI} = 0.33$.

528 Network models

529 **Leaky integrate and fire (LIF) neurons.** A LIF neuron j spikes as soon at its membrane
530 potential $V_j(t)$ is above its threshold v_{th} . At each spike time t , the membrane potential $V_j(t)$ is
531 reset by subtracting the threshold value v_{th} and the neuron enters a strict refractory period for 2
532 to 5 ms (depending on the experiment) where it cannot spike again. Between spikes the membrane
533 voltage $V_j(t)$ is following the dynamic:

$$\tau_m \dot{V}_j(t) = -V_j(t) + R_m I_j(t).$$

534 Our simulations were performed in discrete time with a time step $\delta t = 1$ ms. In discrete
535 time, the input and output spike trains are modeled as binary sequences $x_i(t), z_j(t) \in \{0, \frac{1}{\delta t}\}$
536 respectively. Neuron j emits a spike at time t if it is currently not in a refractory period, and its
537 membrane potential $V_j(t)$ is above its threshold. During the refractory period following a spike,
538 $z_j(t)$ is fixed to 0. The neural dynamics in discrete time reads as follows:

$$V_j(t + \delta t) = \alpha V_j(t) + (1 - \alpha) R_m I_j(t) - v_{\text{th}} z_j(t) \delta t, \quad (2)$$

539 where $\alpha = \exp(-\frac{\delta t}{\tau_m})$, with τ_m being the membrane constant of the neuron j . The spike of neuron
540 j is defined by $z_j(t) = H\left(\frac{V_j(t) - v_{\text{th}}}{v_{\text{th}}}\right) \frac{1}{\delta t}$, with $H(x) = 0$ if $x < 0$ and 1 otherwise. The term
541 $-v_{\text{th}} z_j(t) \delta t$ implements the reset of the membrane voltage after each spike.

542 In all simulations the R_m was set to $1 \text{ G}\Omega$. The input current $I_j(t)$ is defined as the weighted
543 sum of spikes from external inputs and other neurons in the network:

$$I_j(t) = \sum_i W_{ji}^{\text{in}} x_i(t - d_{ji}^{\text{in}}) + \sum_i W_{ji}^{\text{rec}} z_i(t - d_{ji}^{\text{rec}}), \quad (3)$$

544 where W_{ji}^{in} and W_{ji}^{rec} denote respectively the input and the recurrent synaptic weights and d_{ji}^{in}
545 and d_{ji}^{rec} the corresponding synaptic delays.

546 **Adaptive leaky integrate and fire (ALIF) neurons.** An ALIF neuron extends the LIF
547 neuron with an SFA mechanism. The SFA is realized by replacing the fixed threshold v_{th} with the
548 adaptive threshold $A_j(t)$ which follows the dynamic described in equation (1). The spiking output
549 of ALIF neuron j is then defined by $z_j(t) = H\left(\frac{V_j(t) - A_j(t)}{A_j(t)}\right) \frac{1}{\delta t}$.
550

551 Adaptation time constants of ALIF neurons were chosen to match the task requirements while
 552 still conforming to the experimental data from rodents (Allen Institute, 2018; Pozzorini et al.,
 553 2013, 2015; Mensi et al., 2012). For an analysis of the impact of the adaptation time constants on
 554 the performance see Table S1 in Supplement.

555
 556 **LIF neurons whose excitability gets increased through their firing: ELIF neurons.**
 557 There exists experimental evidence that some neurons fire for the same stimulus more for a rep-
 558 etition of the same sensory stimulus. We refer to such neurons as ELIF neurons, since they are
 559 becoming more excitable. Such repetition enhancement was discussed for example in (Tartaglia
 560 et al., 2015). But to the best of our knowledge, it has remained open whether repetition enhance-
 561 ment is a network effect, resulting for example from a transient depression of inhibitory synapses
 562 onto the cell that is caused by postsynaptic firing (Kullmann et al., 2012), or a result of an intrinsic
 563 firing property of some neurons. We used a simple model for ELIF neurons that is dual to the
 564 ALIF neuron model: The threshold is lowered through each spike of the neuron, and then decays
 565 exponentially back to its resting value. This can be achieved by using a negative value for β in
 566 equation (1).

567
 568 **Models for Short-Term Plasticity (STP) of synapses.** We modelled the STP dynamic
 569 according to the classical model of STP in (Mongillo et al., 2008). The STP dynamics in discrete
 570 time, derived from the equations in (Mongillo et al., 2008), are as follows:

$$u'_{ji}(t + \delta t) = \exp\left(\frac{-\delta t}{F}\right) u'_{ji}(t) + U_{ji}(1 - u_{ji}(t))z_i(t)\delta t, \quad (4)$$

$$u_{ji}(t + \delta t) = U_{ji} + u'_{ji}(t), \quad (5)$$

$$r'_{ji}(t + \delta t) = \exp\left(\frac{-\delta t}{D}\right) r'_{ji}(t) + u_{ji}(t)(1 - r'_{ji}(t))z_i(t)\delta t, \quad (6)$$

$$r_{ji}(t + \delta t) = 1 - r'_{ji}(t), \quad (7)$$

$$W_{ji}^{STP}(t + \delta t) = W_{ji}^{rec} u_{ji}(t) r_{ji}(t), \quad (8)$$

571 where $z_i(t)$ is the spike train of the pre-synaptic neuron and W_{ji}^{rec} scales the synaptic efficacy of
 572 synapses from neuron i to neuron j . Networks with STP were constructed from LIF neurons with
 573 the weight W_{ji}^{rec} in equation (3) replaced by the time dependent weight $W_{ji}^{STP}(t)$.

574 STP time constants of facilitation-dominant and depression-dominant network models were
 575 based on values of experimental recordings in (Wang et al., 2006) of PFC-E1 and PFC-E2 synapse
 576 types respectively. Recordings in (Wang et al., 2006) were performed in medial prefrontal cortex of
 577 young adult ferrets. For the STORE-RECALL task, both facilitation and depression time constants
 578 were equally scaled up until the larger time constant matched the requirement of the task (see
 579 section on ‘‘Comparing networks with different slow processes’’ below). In the sMNIST task for
 580 the depression-dominant network model (STP-D) we used values based on PFC-E2: $F = 20$ ms,
 581 $D = 700$ ms and $U = 0.2$, and for facilitation-dominant network model (STP-F) we used values
 582 based on PFC-E1: $F = 500$ ms, $D = 200$ ms and $U = 0.2$.

583
 584 **Weight initialization.** Initial input and recurrent weights were drawn from a Gaussian distribu-
 585 tion $W_{ji} \sim \frac{w_0}{\sqrt{n_{in}}} \mathcal{N}(0, 1)$, where n_{in} is the number of afferent neurons and $\mathcal{N}(0, 1)$ is the zero-mean
 586 unit-variance Gaussian distribution and $w_0 = \frac{1\text{Volt}}{R_m} \delta t$ is a normalization constant (Bellec et al.,
 587 2018b).

588 Training methods

589 **BPTT.** In artificial recurrent neural networks such as LSTMs, gradients can be computed with
 590 backpropagation through time (BPTT). In spiking neural networks, complications arise from the
 591 non-differentiability of the output of spiking neurons. In our discrete time simulation, this is
 592 formalized by the discontinuous step function H arising in the definition of the spike variable $z_j(t)$.
 593 All other operations can be differentiated exactly with BPTT. For feedforward artificial neural
 594 networks using step functions, a solution was to use a pseudo derivative $H'(x) := \max\{0, 1 - |x|\}$,

595 but we observed that this convention is unstable with recurrently connected neurons. We found
596 that dampening this pseudo-derivative with a factor $\gamma < 1$ (typically $\gamma = 0.3$) solves that issue.
597 Hence we use the pseudo-derivative:

$$\frac{dz_j(t)}{dv_j(t)} := \gamma \max\{0, 1 - |v_j(t)|\}, \quad (9)$$

598 where $v_j(t)$ denotes the normalized membrane potential $v_j(t) = \frac{V_j(t) - A_j(t)}{A_j(t)}$. Importantly, gradients
599 can propagate in adaptive neurons through many time steps in the dynamic threshold without
600 being affected by the dampening.

601
602 ***e-prop***. In the 12AX task the networks were trained using biologically plausible learning method
603 *random e-prop* (Bellec et al., 2019) in addition to *BPTT*.

604 Tasks

605 **The STORE-RECALL task of Fig. 2** The input to the network consisted of STORE, RE-
606 CALL, and 20 bits which were represented by sub-populations of spiking input neurons. STORE
607 and RECALL commands were represented by 4 neurons each. The 20 bits were represented by
608 population coding where each bit was assigned 4 input neurons (2 for value zero, and 2 for value
609 one). When a sub-population is active, it would exhibit a Poisson firing with frequency of 400 Hz.
610 To measure the generalization capability of a trained network, we first generate a test set dictio-
611 nary of 20 unique feature vectors (random bit strings of length 20) that have at least a Hamming
612 distance of 5 bits among each other. For every training batch a new dictionary of 40 random bit
613 strings (of length 20) would be generated where each string has a Hamming distance of at least 5
614 bits from any of the bit string in the test set dictionary. This way we ensure that, during training,
615 a network never encounters any bit string similar to one from the test set. Each input sequence
616 consisted of 10 steps (200 ms each) where a different population encoded bit string is shown during
617 every step. Only during the RECALL period, the 20 bit input populations are silent. At every
618 step, the STORE or the RECALL populations were activated interchangeably with probability
619 0.2 which resulted in distribution of delays between the STORE-RECALL pairs in the range [200,
620 1600] ms.

621 The training and the test performance were computed as average over 256 and 512 random
622 input sequences respectively. Networks were trained for 4000 iterations and stopped if the error
623 on the training batch was below 1%. We used the Adam optimizer with default parameters and
624 initial learning rate of 0.01 which is decayed every 200 iterations by a factor of 0.8. We also used
625 learning rate ramping, which, for the first 200 iterations, monotonically increased the learning
626 rate from 0.00001 to 0.01. To avoid unrealistically high firing rates, the loss function contained
627 a regularization term (scaled with coefficient 0.001) that minimizes the squared difference of the
628 average firing rate of individual neurons from a target firing rate of 10 Hz. To improve convergence,
629 we also included an entropy component to the loss (scaled with coefficient 0.3) which was computed
630 as the mean of the entropies of the sigmoid neurons outputs.

631 We trained an ALIF network and a LIF network, both consisting of 500 recurrently connected
632 neurons. The membrane time constant was $\tau_m = 20$ ms. For adaptation parameters we used
633 $\beta_{ALIF} = 4$ mV and $\tau_a = 800$ ms with baseline threshold voltage 10 mV. Synaptic delay was 1 ms.
634 The input to the sigmoidal readout neurons were the neuron traces that were calculated by passing
635 all the network spikes through a low-pass filter with a time constant of 20 ms.

636 We ran 5 training runs with different random seeds for both ALIF and LIF network models.
637 All runs of the ALIF network converged after ~ 3600 iterations to a training error below 1%. At
638 that point we measured the accuracy on 512 test sequences generated using the previously unseen
639 test bit strings which resulted in test accuracy of 99.09% with standard deviation of 0.17%. The
640 LIF network was not able to solve the task in any of the runs (all runs resulted in 0% training
641 and test accuracy with zero standard deviation). On the level of individual feature recall accuracy,
642 the best out of 5 training runs of the LIF network was able to achieve 49% accuracy which is
643 the chance level since individual features are binary bits. In contrast, all ALIF network runs had
644 individual feature level accuracy of above 99.99%.

645

646 **Decoding memory from the network activity.** We trained a Support Vector Machine (SVM)
647 to classify the stored memory content from the network spiking activity in the step before the
648 RECALL (200 ms before the start of RECALL command). We performed a cross-validated grid-
649 search to find the best hyperparameters for the SVM which included kernel type [linear, polynomial,
650 RBF] and penalty parameter C of the error term [0.1, 1, 10, 100, 1000]. We trained SVMs on
651 test batches of the 5 different training runs (see above). SVMs trained on the period preceding
652 the RECALL command of a test batch achieved an average of 4.38% accuracy with standard
653 deviation of 1.29%. In contrast SVMs trained on a period during the RECALL command achieved
654 an accuracy of 100%. This demonstrates that the memory stored in the network is not decodable
655 from the network firing activity before the RECALL input command.

656 Additionally, analogous to the experiments of (Wolff et al., 2017), we trained SVMs on network
657 activity during the encoding (STORE) period and evaluated them on the network activity during
658 reactivation (RECALL), and vice versa. In both scenarios, the classifiers were not able to classify
659 the memory content of the evaluation period (0.0% accuracy).

660

661 **Comparing networks with different slow processes on a simplified version of the**
662 **STORE-RECALL task.** For the comprehensive comparison of networks endowed with dif-
663 ferent slow processes in neuron and synapse dynamics we used a single dimensional version of the
664 STORE-RECALL task where only a single feature needs to be stored and recalled from memory.
665 The input to the network consisted of 40 input neurons: 10 for STORE, 10 for RECALL, and
666 20 for population coding of a binary feature. Each sequence consisted of 20 steps (200 ms each)
667 where the STORE or the RECALL populations were activated with probability 0.09 interchange-
668 ably which resulted in delays between the STORE-RECALL pairs to be in the range [200, 3600]
669 ms.

670 The training batch and the test performance were computed as average over 128 and 2048
671 random input sequences respectively. All networks were trained for 400 iterations. We used the
672 Adam optimizer with default parameters and initial learning rate of 0.01 which was decayed every
673 100 iterations by a factor of 0.3. The same firing rate regularization term was added to the loss
674 as in the original STORE-RECALL setup (see above).

675 All networks consisted of 60 recurrently connected neurons. The membrane time constant was
676 $\tau_m = 20$ ms. For ALIF and ELIF networks, we used $\beta_{ALIF} = 1$ mV and $\beta_{ELIF} = -0.5$ mV with
677 $\tau_a = 2000$ ms. Synapse parameters of STP-D network were $F = 51 \pm 15$ ms, $D = 2000 \pm 51$ ms
678 and $U = 0.25$, and of STP-F network $F = 2000 \pm 146$ ms, $D = 765 \pm 71$ ms and $U = 0.28$. The
679 baseline threshold voltage was 10 mV for all models except ELIF for which it was 20 mV. Synaptic
680 delay was 1 ms across all network models.

681

682 **Google Speech Commands task.** Features were extracted from the raw audio using the Mel
683 Frequency Cepstral Coefficient (MFCC) method with 30 ms window size, 1 ms stride and 40 output
684 features. The network models were trained to classify the input features to one of the 10 keywords
685 (yes, no, up, down, left, right, on, off, stop, go) or to two special classes for silence or unknown
686 word (where the remainder of 20 recorded keywords are grouped). The training, validation and
687 test set were assigned 80, 10, and 10 percent of data respectively while making sure that audio
688 clips from the same person stay in the same set.

689 All networks were trained for 18,000 iterations using the Adam optimizer with batch size 100.
690 The output spikes of the networks were averaged over time, and the linear readout layer was applied
691 to those values. During the first 15,000 we used a learning rate of 0.001 and for the last 3000
692 we used a learning rate of 0.0001. The loss function contained a regularization term (scaled with
693 coefficient 0.001) that minimizes the squared difference of average firing rate between individual
694 neurons and a target firing rate of 10 Hz.

695 Both ALIF and LIF networks consisted of 2048 fully connected neurons in a single recurrent
696 layer. The neurons had a membrane time constant of $\tau_m = 20$ ms, the adaptation time constant
697 of ALIF neurons was $\tau_a = 100$ ms, adaptation strength was $\beta = 2$ mV. The baseline threshold was
698 $v_{th} = 10$ mV, and the refractory period was 2 ms. Synaptic delay was 1 ms.

699

700 **Delayed-memory XOR task.** The pulses on the two input channels were generated with 30

701 ms duration and the shape of a normal probability density function normalized in the range $[0, 1]$.
702 The pulses were added or subtracted from the baseline zero input current at appropriate delays.
703 The go-cue was always a positive current pulse. The 6 possible configurations of the input pulses
704 $(+, -, ++, --, +-, -+)$ were sampled with equal probability during training and testing.

705 Networks were trained for 2000 iterations using the Adam optimizer with batch size 256. The
706 initial learning rate was 0.01 and every 200 iterations the learning rate was decayed by a factor of
707 0.8. The loss function contained a regularization term (scaled with coefficient 50) that minimizes
708 the squared difference of the average firing rate of individual neurons from a target firing rate of
709 10 Hz. This regularization resulted in networks with mean firing rate of 10 Hz where firing rates
710 of individual neurons were spread in the range $[1, 16]$ Hz.

711 Both ALIF and LIF networks consisted of 80 fully connected neurons in a single recurrent layer.
712 The neurons had a membrane time constant of $\tau_m = 20$ ms, a baseline threshold $v_{th} = 10$ mV, and
713 a refractory period of 3 ms. ALIF neurons had an adaptation time constant of $\tau_a = 500$ ms and an
714 adaptation strength of $\beta = 1$ mV. Synaptic delay was 1 ms. For training the network to classify
715 the input into one of the three classes, we used the cross-entropy loss between the labels and the
716 softmax of three linear readout neurons. The input to the linear readout neurons were the neuron
717 traces that were calculated by passing all the network spikes through a low-pass filter with a time
718 constant of 20 ms.

719
720 **The sequential MNIST (sMNIST) task.** The input consisted of sequences of 784 pixel values
721 created by unrolling the handwritten digits of the MNIST dataset, one pixel after the other in a
722 scanline manner as indicated in Fig. S3A. For comparing different spiking network models, we used
723 1 ms presentation time for each pixel (Fig. 3C). LSTM networks also work well for tasks on larger
724 time-scales. Hence for comparing LSNNs with LSTM networks, we used a version of the task with
725 2 ms presentation time per pixel, thereby doubling the length of sequences to be classified to 1568
726 ms (Fig. 3D). A trial of a trained LSNN (with an input sequence that encodes a handwritten digit
727 “3” using population rate coding) is shown in Fig. S3B. The top row of Fig. S3B shows a version
728 where the grey value of the currently presented pixel is encoded by population coding, through the
729 firing probability of 80 input neurons. Somewhat better performance was achieved when each of
730 the 80 input neurons was associated with a particular threshold for the grey value, and this input
731 neuron fired whenever the grey value crossed its threshold in the transition from the previous to
732 the current pixel (this input convention was used for the results of Fig. 3C,D). Grey values of
733 pixels were presented to the LSTM network simply as analog numbers.

734 Networks were trained for 36,000 iterations using the Adam optimizer with batch size 256. The
735 initial learning rate was 0.01 and every 2500 iterations the learning rate was decayed by a factor of
736 0.8. The loss function contained a regularization term (scaled with coefficient 0.1) that minimizes
737 the squared difference of average firing rate between individual neurons and a target firing rate of
738 10 Hz.

739 The neurons had a membrane time constant of $\tau_m = 20$ ms, a baseline threshold of $v_{th} = 10$ mV,
740 and a refractory period of 5 ms. The adaptation time constants of ALIF and ELIF neurons were
741 $\tau_a = 700$ ms in Fig. 3C. ALIF neurons had $\tau_a = 1400$ ms in Fig. 3D. The adaptation strength of
742 ALIF neurons was $\beta = 1.8$ mV, and of ELIF neurons $\beta = -0.9$ mV. Synaptic delay was 1 ms. The
743 output of the LSNN is produced by the softmax of 10 linear output neurons that receive spikes
744 from all neurons in the network, as shown in the bottom row of Fig. S3B. For training the network
745 to classify to one of the ten classes we used cross-entropy loss computed between the labels and the
746 softmax of output neurons. The input to the linear readout neurons were the neuron traces that
747 were calculated by passing all the network spikes through a low-pass filter with a time constant of
748 20 ms.

749
750 **The 12AX task.** The input for each training and testing episode consisted of a sequence of 90
751 symbols from the set $\{1, 2, A, B, C, X, Y, Z\}$. A single episode could contain multiple occurrences of
752 digits 1 or 2 (up to 23), each time changing the target sequence (A...X or B...Y) after which the
753 network was supposed to output R. Each digit could be followed by up to 26 letters before the next
754 digit appeared. More precisely, the following regular expression describes the string that is pro-
755 duced: $[12][ABCXYZ]\{1, 10\}((A[CZ]\{0, 6\}X|B[CZ]\{0, 6\}Y)|([ABC][XYZ]))\{1, 2\}$. Each choice
756 in this regular expression is made randomly.

757 The neurons had a membrane time constant of $\tau_m = 20$ ms, a baseline threshold $v_{th} = 30$ mV,
758 a refractory period of 5 ms, and synaptic delays of 1 ms. ALIF neurons had an adaptation strength
759 of $\beta = 1.7$ mV, and adaptation time constants were chosen uniformly from [1, 13500] ms.

760 A cross-entropy loss function was used along with a regularization term (scaled with coefficient
761 15) that minimizes the squared difference of average firing rate between individual neurons and a
762 target firing rate of 10 Hz. The LSNN was trained for 10,000 iterations with a batch size of 20
763 episodes and a fixed learning rate of 0.001. An episode consisted of 90 steps, with between 4 to
764 23 tasks generated according to the task generation procedure described previously. We trained
765 the network with *BPTT* using 5 different seeds, which resulted in average test success rate 97.79%
766 with standard deviation 0.42%. The network trained with *random e-prop* using 5 different seeds
767 resulted in average test success rate 92.89% with standard deviation 0.75%.

768
769 **Symbolic computation on strings of symbols.** The input to the network consisted of 35
770 symbols - 31 symbols represented symbols from the English alphabet {a, b, c, d, ... x, y, z, A,
771 B, C, D, E}, one symbol was for “end-of-string” (EOS) ‘*’, one for cue for the output prompt
772 ‘?’, and two symbols to denote whether the task instruction was duplication or reversal. The task
773 and the rest of the symbols were encoded using separate one-hot vectors of dimension 2 and 33
774 respectively. Inputs to the network were transformed into spike trains using a population of 5
775 spiking neurons for each input component for a total of 175 input neurons. This population fired
776 at a “high” rate (200 Hz) to encode 1, and at a “low” rate (2 Hz) otherwise. The output consisted
777 of 32 linear readouts: 31 for symbols from the English alphabet and one additional readout for
778 the “end-of-string” symbol. The input to these linear readouts was the value of neuron traces at
779 the end of each step of 500 ms during the output period, i.e, the second half of each episode. The
780 neuron traces were calculated by passing all the network spikes through a low-pass filter with a
781 time constant of 250 ms. The final output symbol was produced using the argmax over the value
782 of all the readouts (a softmax instead of the hard argmax was used during training). The network
783 was trained to minimize the cross entropy error between the softmax applied on the output layer
784 and targets. The loss function contained a regularization term (scaled with coefficient 5) that
785 minimizes the squared difference of average firing rate between individual neurons and a target
786 firing rate of 20 Hz.

787 The training was performed for 50,000 iterations, with a batch size of 50 episodes. We used
788 the Adam optimizer with default parameters and fixed learning rate of 0.001. Each symbol was
789 presented to the network for a duration of 500 ms. The primary metric we used for measuring
790 the performance of the network was success rate, which was defined as the percentage of episodes
791 where the network produced the full correct output for a given string i.e. all the output symbols
792 in the episode had to be correct. The network was tested on 50,000 previously unseen strings.

793 The network consisted of 128 LIF and 192 ALIF neurons. All the neurons had a membrane
794 time constant of $\tau_m = 20$ ms, a baseline threshold $v_{th} = 30$ mV, a refractory period of 5 ms, and a
795 synaptic delay of 1 ms. ALIF neurons in the network had an adaptation strength of $\beta = 1.7$ mV,
796 and an adaptation time constant randomly uniformly chosen from the range [1, 6000] ms. All
797 other parameters were the same as in the other experiments. We trained the network using 5
798 different seeds and tested it on previously unseen strings. Average test success rate was 95.88%
799 with standard deviation 1.39%.

800
801 **Analysis of spiking data.** We used 3-way ANOVA to analyze if a neuron’s firing rate is signif-
802 icantly affected by task, serial position in the sequence, symbol identity, or combination of these
803 (similar to (Lindsay et al., 2017)). We refer to these factors as “conditions”. The analysis was
804 performed on the activity of the neurons of the trained LSNN during 50,000 test episodes. For
805 the analysis, neurons whose average firing rate over all episodes was lower than 2Hz or greater
806 than 60Hz were discarded from the analysis to remove large outliers. This left 279 out of the 320
807 neurons. From each episode, a serial position from the input period was chosen randomly, and
808 hence each episode could be used only once, i.e., as one data point. This was to make sure that
809 each entry in the 3-way ANOVA was completely independent of other entries, since the neuron
810 activity within an episode is highly correlated. Each data point was labeled with the corresponding
811 triple of (task type, serial position, symbol identity). To ensure that the dataset was balanced,
812 the same number of data points per particular combination of conditions was used, discarding

813 all the excess data points, resulting in a total of 41,850 data points. To categorize a neuron as
814 selective to one or more conditions, or combination of conditions, we observed p-values obtained
815 from 3-way ANOVA and calculated the effect size ω^2 for each combination of conditions. If the
816 p-value was smaller than 0.001 and ω^2 greater than 0.14 for a particular combination of conditions,
817 the neuron was categorized as selective to that combination of conditions. The ω^2 threshold of
818 0.14 was suggested by (Field, 2013) to select large effect sizes. Each neuron can have large effect
819 size for more than one combination of conditions. Thus the values shown in Fig. 5D sum to > 1 .
820 The neuron shown in Fig. 5E had the most prominent selectivity for the combination of Task \times
821 Position \times Symbol, with $\omega^2 = 0.394$ and $p < 0.001$. The neuron shown in Fig. 5F was categorized
822 as selective to a combination of Position \times Symbol category, with $\omega^2 = 0.467$ and $p < 0.001$.
823 While the 3-way ANOVA tells us if a neuron is selective to a particular combination of conditions,
824 it does not give us the exact task/symbol/position that the neuron is selective to. To find the
825 specific task/symbol/position that the neuron was selective to, Welch's t-test was performed, and
826 a particular combination with maximum t-statistic and $p < 0.001$ was chosen to be shown in
827 Fig. 5E,F.

828 References

- 829 Allen Institute. Allen Cell Types Database Technical white paper: GLIF models
830 <http://help.brain-map.org/download/attachments/8323525/glifmodels.pdf>. Technical
831 report, October 2017. v4.
- 832 Allen Institute. © 2018 Allen Institute for Brain Science. Allen Cell Types Database, cell feature
833 search. Available from: celltypes.brain-map.org/data. 2018.
- 834 J. Barbosa, H. Stein, R. Martinez, A. Galan, K. Adam, S. Li, J. Valls-Solé, C. Constantinidis,
835 and A. Compte. Interplay between persistent activity and activity-silent dynamics in prefrontal
836 cortex during working memory. *bioRxiv*, page 763938, 2019.
- 837 P. Barone and J.-P. Joseph. Prefrontal cortex and spatial sequencing in macaque monkey. *Exper-*
838 *imental brain research*, 78(3):447–464, 1989.
- 839 G. Bellec, D. Kappel, W. Maass, and R. Legenstein. Deep rewiring: Training very sparse deep
840 networks. In *International Conference on Learning Representations*, Feb. 2018a. URL https://openreview.net/forum?id=BJ_wN01C-.
- 842 G. Bellec, D. Salaj, A. Subramoney, R. Legenstein, and W. Maass. Long short-term memory and
843 learning-to-learn in networks of spiking neurons. In *Advances in Neural Information Processing*
844 *Systems*, pages 787–797, 2018b.
- 845 G. Bellec, F. Scherr, A. Subramoney, E. Hajek, D. Salaj, R. Legenstein, and W. Maass. A solution
846 to the learning dilemma for recurrent networks of spiking neurons. *bioRxiv*, page 738385, 2019.
- 847 A. F. Carpenter, G. Baud-Bovy, A. P. Georgopoulos, and G. Pellizzer. Encoding of serial order in
848 working memory: neuronal activity in motor, premotor, and prefrontal cortex during a memory
849 scanning task. *Journal of Neuroscience*, 38(21):4912–4933, 2018.
- 850 S. N. Chettih and C. D. Harvey. Single-neuron perturbations reveal feature-specific competition
851 in V1. *Nature*, 567(7748):334–340, 2019.
- 852 B. Cramer, Y. Stradmann, J. Schemmel, and F. Zenke. The Heidelberg spiking datasets for the
853 systematic evaluation of spiking neural networks. *arXiv preprint arXiv:1910.07407*, 2019.
- 854 A. Field. *Discovering statistics using IBM SPSS statistics*. Sage, 2013.
- 855 W. Gerstner, W. M. Kistler, R. Naud, and L. Paninski. *Neuronal dynamics: From single neurons*
856 *to networks and models of cognition*. Cambridge University Press, 2014.
- 857 B. Gutkin and F. Zeldenrust. Spike frequency adaptation. *Scholarpedia*, 9(2):30643, 2014. doi:
858 10.4249/scholarpedia.30643. revision #143322.

- 859 C. D. Harvey, P. Coen, and D. W. Tank. Choice-specific sequences in parietal cortex during a
860 virtual-navigation decision task. *Nature*, 484(7392):62–68, 2012.
- 861 B. Hu, M. E. Garrett, P. A. Groblewski, D. R. Ollerenshaw, J. Shang, K. Roll, S. Manavi, C. Koch,
862 S. R. Olsen, and S. Mihalas. Adaptation supports short-term memory in a visual change detec-
863 tion task. *bioRxiv*, 2020.
- 864 D. Huh and T. J. Sejnowski. Gradient descent for spiking neural networks. In *Advances in Neural*
865 *Information Processing Systems*, pages 1433–1443, 2018.
- 866 H. K. Inagaki, L. Fontolan, S. Romani, and K. Svoboda. Discrete attractor dynamics underlies
867 persistent activity in the frontal cortex. *Nature*, 566(7743):212–217, 2019.
- 868 J. Kamiński and U. Rutishauser. Between persistently active and activity-silent frameworks: novel
869 vistas on the cellular basis of working memory. *Annals of the New York Academy of Sciences*,
870 2019.
- 871 D. M. Kullmann, A. W. Moreau, Y. Bakiri, and E. Nicholson. Plasticity of inhibition. *Neuron*, 75
872 (6):951–962, 2012.
- 873 A. Kusupati, M. Singh, K. Bhatia, A. Kumar, P. Jain, and M. Varma. Fastgrnn: A fast, accurate,
874 stable and tiny kilobyte sized gated recurrent neural network. In *Advances in Neural Information*
875 *Processing Systems*, pages 9017–9028, 2018.
- 876 K. S. Lashley. *The problem of serial order in behavior*, volume 21. Bobbs-Merrill Oxford, United
877 Kingdom, 1951.
- 878 G. W. Lindsay, M. Rigotti, M. R. Warden, E. K. Miller, and S. Fusi. Hebbian learning in a random
879 network captures selectivity properties of the prefrontal cortex. *Journal of Neuroscience*, 37(45):
880 11021–11036, 2017.
- 881 Y. Liu, R. J. Dolan, Z. Kurth-Nelson, and T. E. Behrens. Human replay spontaneously reorganizes
882 experience. *Cell*, 178(3):640–652, 2019.
- 883 A. W. MacDonald III. Building a clinically relevant cognitive task: case study of the ax paradigm.
884 *Schizophrenia bulletin*, 34(4):619–628, 2008.
- 885 G. F. Marcus. *The Algebraic Mind: Integrating Connectionism and Cognitive Science*. MIT Press,
886 2003.
- 887 H. Markram, E. Muller, S. Ramaswamy, M. W. Reimann, M. Abdellah, C. A. . Sanchez, and
888 G. A. A. Kahou. Reconstruction and simulation of neocortical microcircuitry. *Cell*, 163(2):
889 456–492, 2015.
- 890 N. Y. Masse, G. R. Yang, H. F. Song, X.-J. Wang, and D. J. Freedman. Circuit mechanisms for
891 the maintenance and manipulation of information in working memory. *Nature Neuroscience*,
892 page 1, 2019.
- 893 S. Mensi, R. Naud, C. Pozzorini, M. Avermann, C. C. Petersen, and W. Gerstner. Parame-
894 ter extraction and classification of three cortical neuron types reveals two distinct adaptation
895 mechanisms. *Journal of neurophysiology*, 107(6):1756–1775, 2012.
- 896 G. Mongillo, O. Barak, and M. Tsodyks. Synaptic theory of working memory. *Science*, 319(5869):
897 1543–1546, 2008.
- 898 C. N. Olivers, J. Peters, R. Houtkamp, and P. R. Roelfsema. Different states in visual working
899 memory: When it guides attention and when it does not. *Trends in cognitive sciences*, 15(7):
900 327–334, 2011.
- 901 R. C. O’Reilly and M. J. Frank. Making working memory work: a computational model of learning
902 in the prefrontal cortex and basal ganglia. *Neural computation*, 18(2):283–328, 2006.

- 903 C. Pozzorini, R. Naud, S. Mensi, and W. Gerstner. Temporal whitening by power-law adaptation
904 in neocortical neurons. *Nature neuroscience*, 16(7):942, 2013.
- 905 C. Pozzorini, S. Mensi, O. Hagens, R. Naud, C. Koch, and W. Gerstner. Automated high-
906 throughput characterization of single neurons by means of simplified spiking models. *PLoS*
907 *computational biology*, 11(6), 2015.
- 908 E. M. Tartaglia, G. Mongillo, and N. Brunel. On the relationship between persistent delay activity,
909 repetition enhancement and priming. *Frontiers in psychology*, 5:1590, 2015.
- 910 C. Teeter, R. Iyer, V. Menon, N. Gouwens, D. Feng, J. Berg, A. Szafer, N. Cain, H. Zeng, M. Hawry-
911 lycz, et al. Generalized leaky integrate-and-fire models classify multiple neuron types. *Nature*
912 *communications*, 9(1):1–15, 2018.
- 913 D. Trübutschek, S. Marti, A. Ojeda, J.-R. King, Y. Mi, M. Tsodyks, and S. Dehaene. A theory of
914 working memory without consciousness or sustained activity. *Elife*, 6:e23871, 2017.
- 915 A. Tsao, J. Sugar, L. Lu, C. Wang, J. J. Knierim, M. B. Moser, and E. I. Moser. Integrating time
916 from experience in the lateral entorhinal cortex. *Nature*, 561(7721):57–52, 2018.
- 917 Y. Wang, H. Markram, P. H. Goodman, T. K. Berger, J. Ma, and P. S. Goldman-Rakic. Hetero-
918 geneity in the pyramidal network of the medial prefrontal cortex. *Nature neuroscience*, 9(4):534,
919 2006.
- 920 M. R. Warden and E. K. Miller. The representation of multiple objects in prefrontal neuronal
921 delay activity. *Cerebral Cortex*, 17(suppl_1):i41–i50, 2007.
- 922 P. Warden. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint*
923 *arXiv:1804.03209*, 2018.
- 924 D. F. Wasmuht, E. Spaak, T. J. Buschman, E. K. Miller, and M. G. Stokes. Intrinsic neuronal
925 dynamics predict distinct functional roles during working memory. *Nature communications*, 9
926 (1):3499, 2018.
- 927 B. D. Winters, L. M. Saksida, and T. J. Bussey. Object recognition memory: neurobiological
928 mechanisms of encoding, consolidation and retrieval. *Neuroscience & Biobehavioral Reviews*, 32
929 (5):1055–1070, 2008.
- 930 M. J. Wolff, J. Jochim, E. G. Akyürek, and M. G. Stokes. Dynamic hidden states underlying
931 working-memory-guided behavior. *Nature Neuroscience*, 20(6):864, 2017.

932 Acknowledgements

933 **General:** We would like to thank Pieter Roelfsema and Chris Summerfield for detailed comments
934 on an earlier version of the manuscript.

935 **Funding:** This research was partially supported by the Human Brain Project (Grant Agreement
936 number 785907) and the SYNCH project (Grant Agreement number 824162) of the European
937 Union. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the
938 Quadro P6000 GPU used for this research. Computations were carried out on the Human Brain
939 Project PCP Pilot Systems at the Juelich Supercomputing Centre, which received co-funding from
940 the European Union (Grand Agreement number 604102) and on the Vienna Scientific Cluster
941 (VSC).

942 **Author contributions:** The study was conceived by DS, AS, GB, WM. The experiments were
943 designed by DS, AS, GB, RL, WM and were conducted by DS, AS, CK. The manuscript was
944 written by DS, AS, CK, RL, GB, WM.

945 **Competing interests:** The authors declare no competing interests.

946 **Data and materials availability:** An implementation of the network model in Tensorflow/Python
947 is available at <https://github.com/IGITUGraz/LSNN-official/>. The sMNIST dataset is avail-
948 able at <https://www.tensorflow.org/datasets/catalog/mnist>. The google speech commands
949 dataset is available at [https://storage.cloud.google.com/download.tensorflow.org/data/](https://storage.cloud.google.com/download.tensorflow.org/data/speech_commands_v0.02.tar.gz)
950 [speech_commands_v0.02.tar.gz](https://storage.cloud.google.com/download.tensorflow.org/data/speech_commands_v0.02.tar.gz).

951 **Figures and Tables**

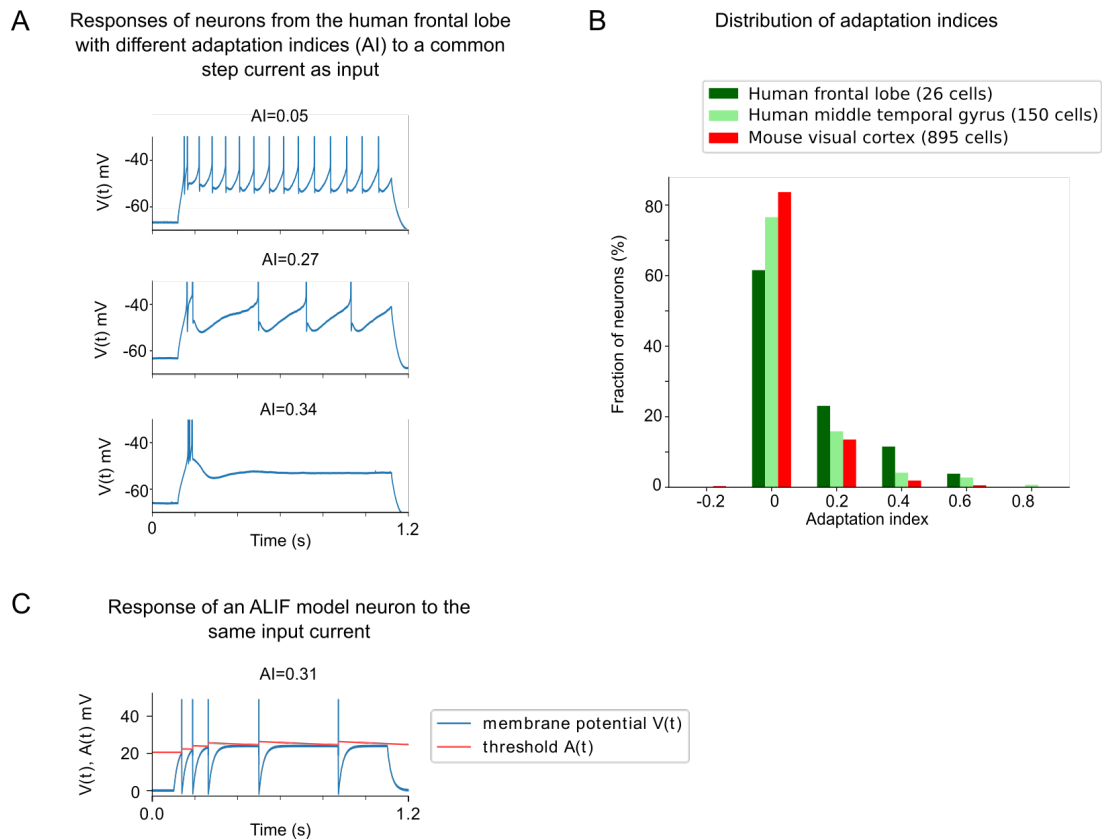


Figure 1: Experimental data on neurons with SFA, and a simple model for SFA. (A) The response to a 1-second long step current is displayed for three sample neurons in the neocortex. The adaptation index AI measures the rate of increase of interspike intervals. $AI > 0$ means that a neuron exhibits SFA. **(B)** Distribution of adaptation indices in neurons from human and rodent neocortex. Source of data for A and B: (Allen Institute, 2018). **(C)** Response of a simple model for a neuron with SFA — the adaptive LIF (ALIF) model — to the same input current as in A.

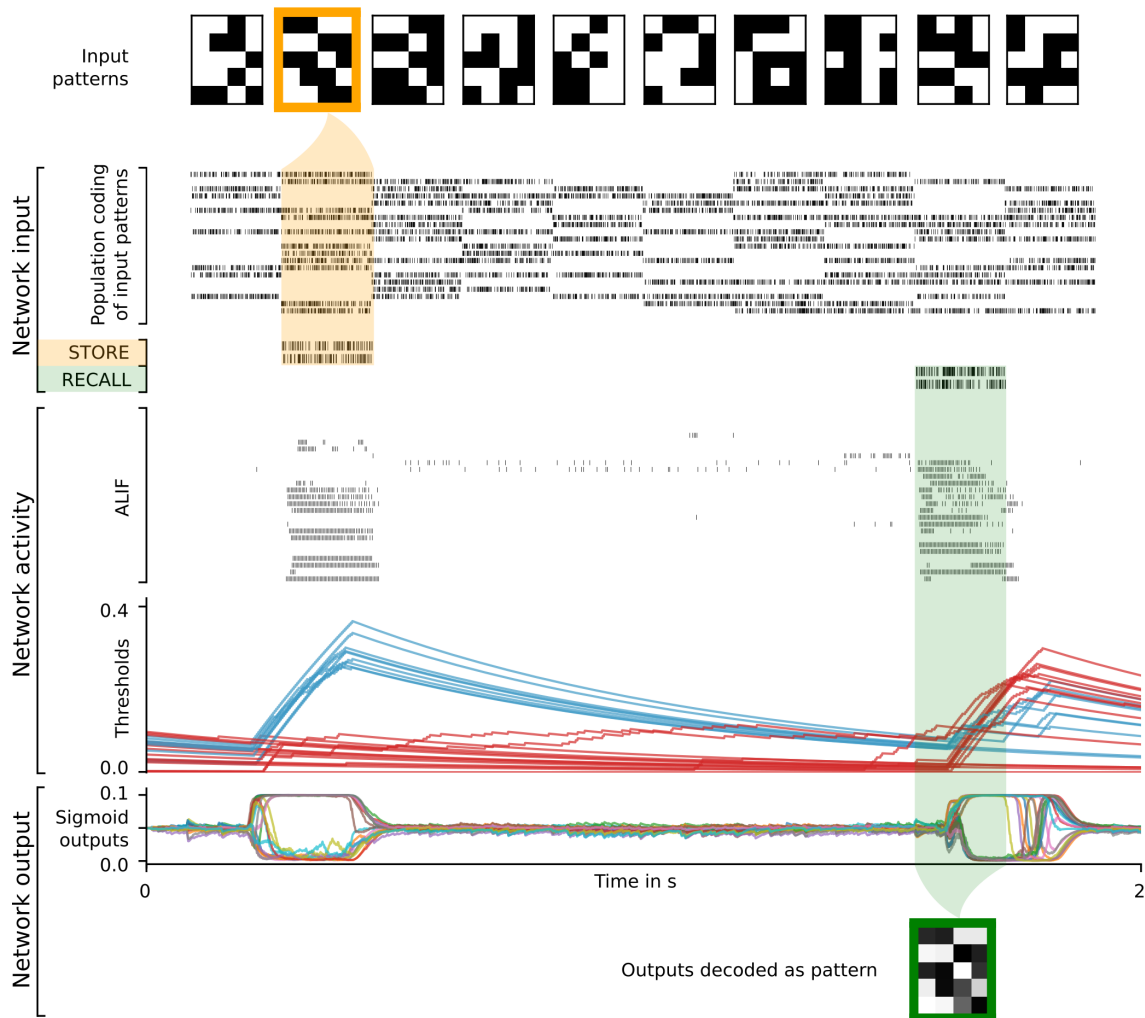


Figure 2: High-dimensional working memory capability of an LSNN. Rows top to bottom: Stream of randomly drawn 20 dimensional input patterns, represented by the firing activity of 20 populations of input neurons (subsamped), firing activity of two additional populations of input neurons for the STORE and RECALL commands, firing activity of 25 sample ALIF neurons in the LSNN (we first ordered all ALIF neurons with regard to the variance of their dynamic firing thresholds, and then picked every 20th), temporal evolution of the firing thresholds of these 25 neurons, traces of the activation of 20 sigmoidal readout neurons, and their average value during the 200 ms time window of the RECALL command represented by grey values. During the RECALL command (green shading) the network successfully reproduced the pattern that had been given as input during the preceding STORE command (yellow shading). Coloring of the threshold traces in blue or red was done after visual inspection to highlight the emergent two disjoint populations of ALIF neurons. The activity of one of them peaks during the STORE command, and provides a negative imprint of the stored pattern during RECALL through a reduced firing response. The other one peaks during RECALL.

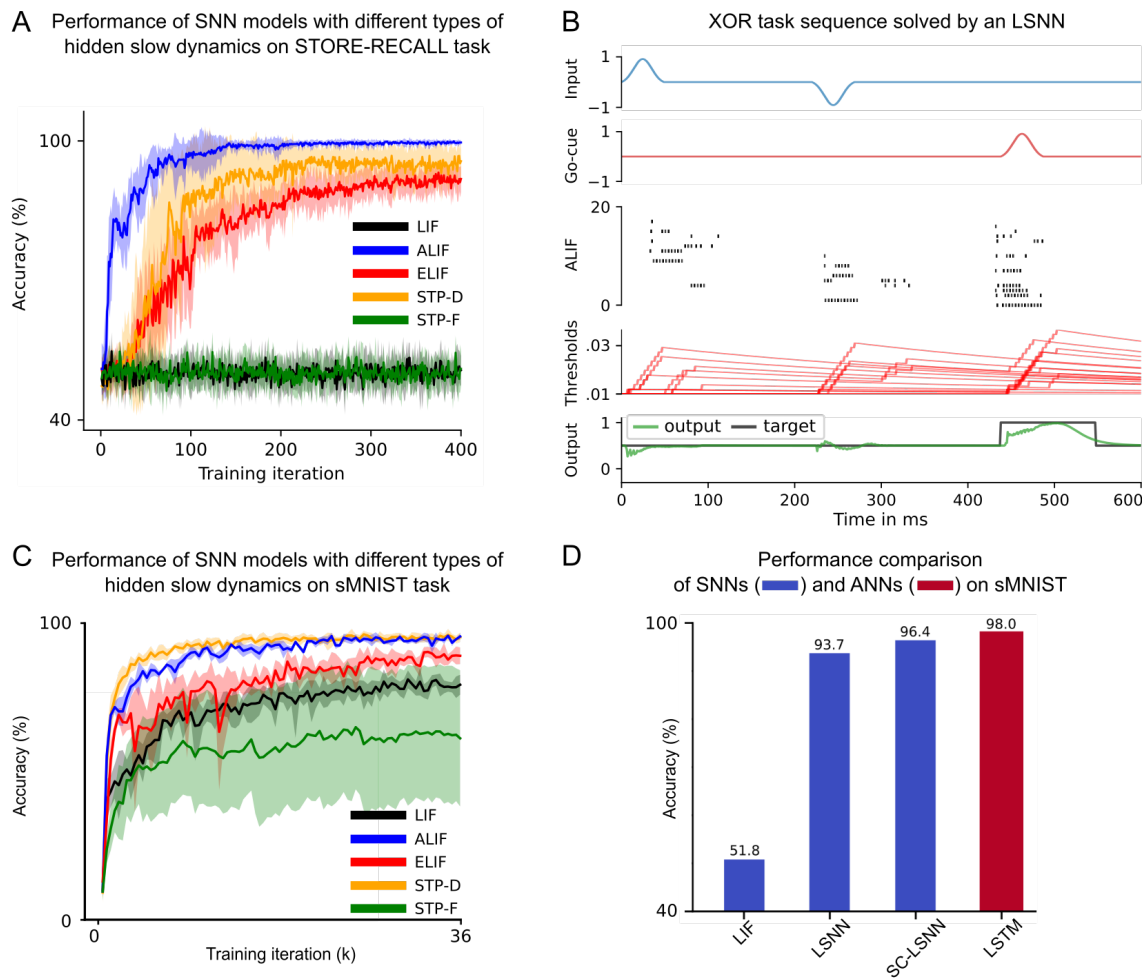


Figure 3: Performance comparisons for common benchmark tasks that require substantial integration of information over time. (A) Learning curves of networks with different slow processes, for a 1D version of the STORE-RECALL task from Fig. 2. The standard SNN (“LIF”) as well as SNNs with STP-F cannot learn the task. SNNs with STP-D come closest to the performance level of the LSNN, but require substantially longer training. Mean accuracy and standard deviation are shown for 7 runs with different network initializations for all 5 network types. (B) Trained LSNN solving the delayed-memory XOR task (Huh and Sejnowski, 2018). Plot of a trial with input consisting of two different types of pulses is shown. From top to bottom: Input pulses, go cue, neuron spike raster, threshold traces, network output. (C) Learning curves of five variants of the SNN model (same as in A) for the sMNIST time series classification task. Mean accuracy and standard deviation are shown for a minimum of 4 runs with different network initializations for all 5 network types. (D) sMNIST performance of two versions of LSNNs are compared with that of an equally large network of LIF neurons, and an LSTM network. SC-LSNN is a sparsely connected LSNN consisting of excitatory and inhibitory neurons.

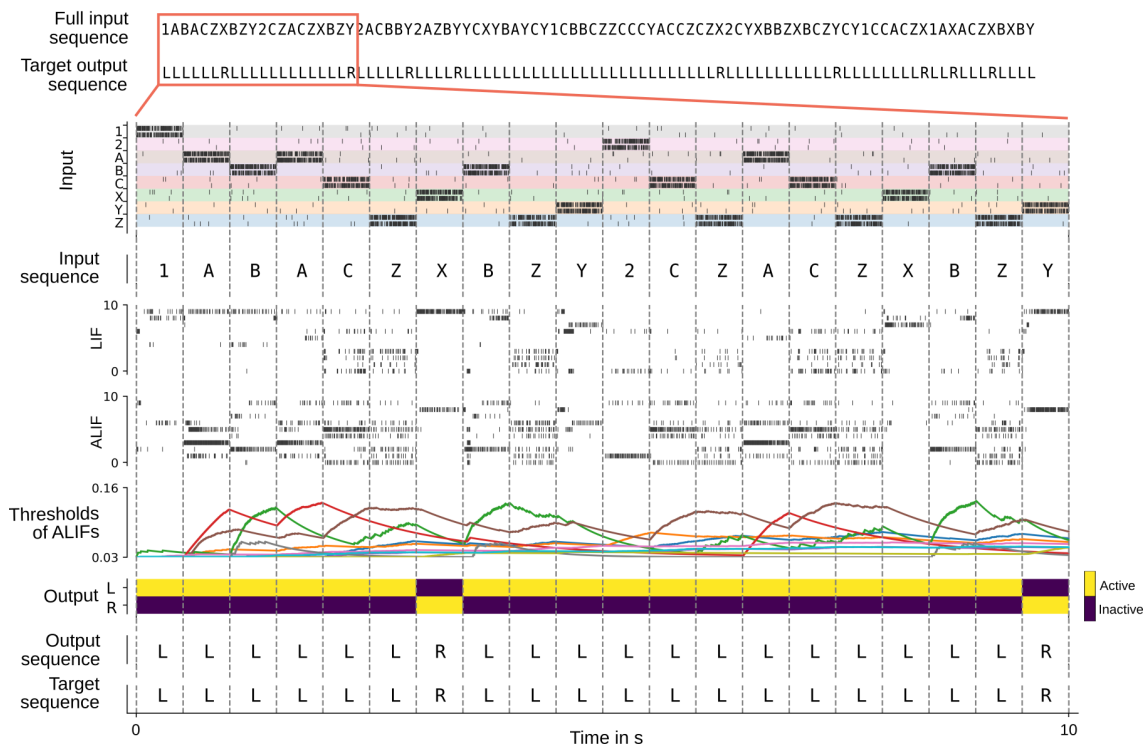


Figure 4: Solving the 12AX task by a network of spiking neurons (LSNN), trained with *random e-prop*. A sample trial of the trained network is shown. From top to bottom: Full input and target output sequence for a trial, consisting of 90 symbols each, blow-up for a subsequence of the input symbols, firing activity of 10 sample LIF neurons and 10 sample ALIF neurons from the LSNN, time course of the firing thresholds of these 10 ALIF neurons, activation of the two readout neurons, the resulting sequence of output symbols which the network produced, and the target output sequence.

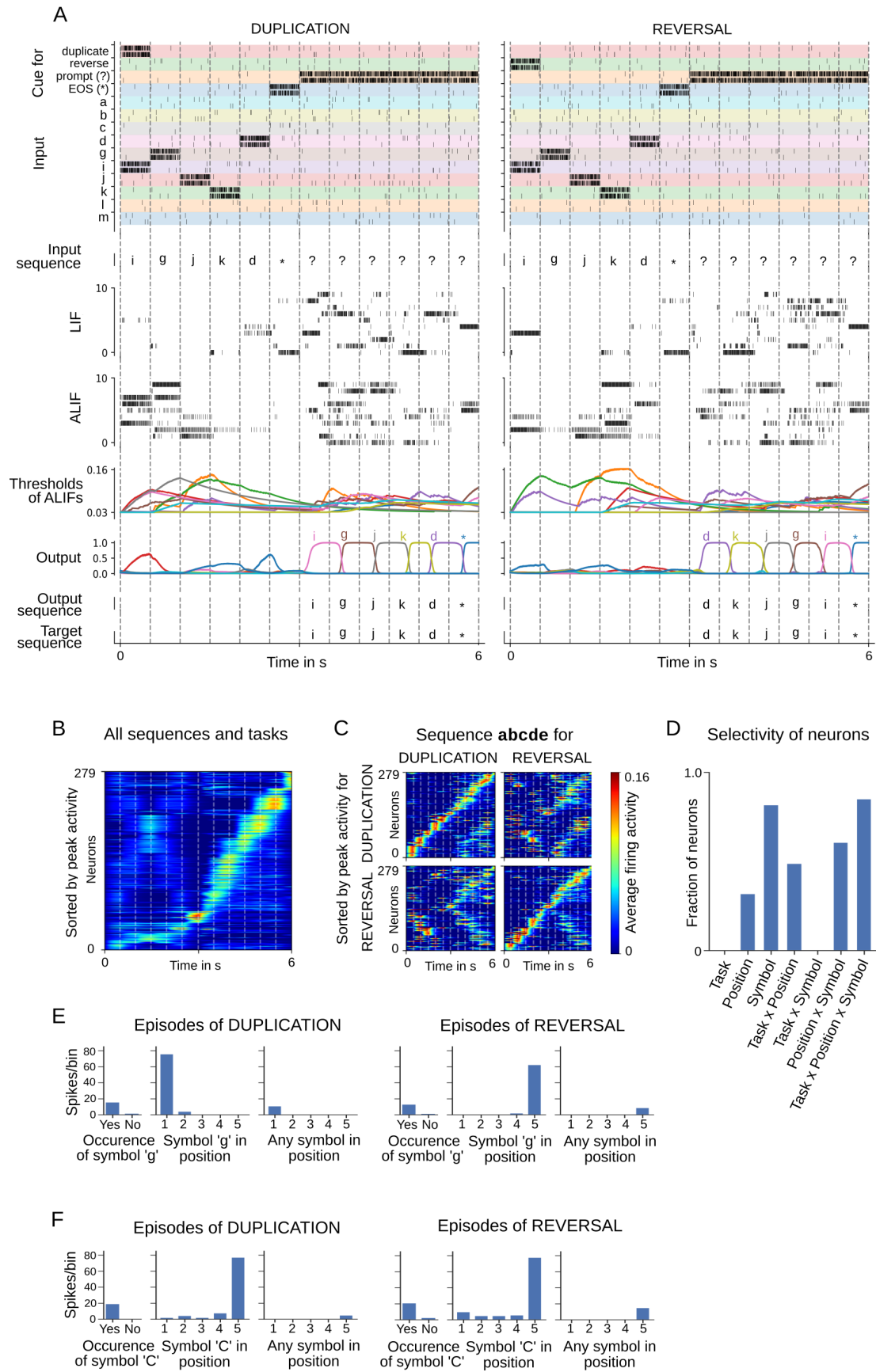


Figure 5: (Caption on the next page.)

Figure 5: Analysis of an LSNN trained to carry out operations on sequences. (A) Two sample episodes where the LSNN carried out sequence duplication (left) and reversal (right). Top to bottom: Spike inputs to the network (subset), sequence of symbols they encode, spike activity of 10 sample LIF, and ALIF neurons in the LSNN, firing threshold dynamics for these 10 ALIF neurons, activation of linear readout neurons, output sequence produced by applying argmax to them, target output sequence. (B-F) Emergent neural coding of 279 neurons in the LSNN. Neurons sorted by time of peak activity. (B) A substantial number of neurons are sensitive to the generic timing of the tasks, especially for the second half of trials when the output sequence is produced. (C) Neurons separately sorted for duplication episodes (left column) and reversal episodes (right column). Many neurons respond to input symbols according to their serial position, but differentially for different tasks. (D) Histogram of neurons categorized according to conditions with statistically significant effect (3-way ANOVA). Firing activity of a sample neuron that fires primarily when: (E) the symbol “g” is to be written at the beginning of the output sequence. The activity of this neuron depends on the task context during the input period; (F) the symbol “C” occurs in position 5 in the input, irrespective of the task context.

Model	test accuracy (%)
FastGRNN-LSQ (Kusupati et al., 2018)	93.18
LSNN	91.21
LIF network	89.04

Table 1: Google Speech Commands. Accuracy of the spiking network models on the test set compared to the state-of-the-art artificial recurrent model reported in (Kusupati et al., 2018). Accuracy of the best out of 5 simulations for LSNNs and LIF networks is reported.