Generation of context dependent sequences by multiple-timescale neural network through successive bifurcations

Tomoki Kurikawa
Department of Physics, Kansai Medical University, Shinmachi 2-5-1, Hirakata, Osaka, Japan

Kunihiko Kaneko
Graduate school of arts and sciences, University of Tokyo, Komaba 3-8-1, Meguro-ku, Tokyo, Japan

(Dated: June 9, 2020)

The generation of robust sequential patterns that depend flexibly on the previous history of inputs and outputs is essential to temporal information processing with working memory in our neural system. We propose a neural network with two timescales, in which a sequence of fixed points of the fast dynamics is generated through bifurcations by slow dynamics as a control parameter. By adopting a simple, biologically plausible learning rule, the neural network can recall a complex context-dependent sequence. Considering multiple timescales experimentally observed in cortical areas, this study provides a general scheme to temporal processing in the brain.

Generating sequential neural patterns is a key function to perform a number of cognitive functions. In perception, external stimuli evoke specific sequential patterns in sensory cortices [1, 2]. In spatial working memory task [3, 4], sequences of neural patterns represent previous or future (planned) pathways. Notably, these sequences are changed flexibly dependent on context which includes previous inputs and actions.

A number of theoretical models that generate sequential patterns have been proposed [5–8], which are represented as fixed points in the neural state space. They are destabilized by some tricks, for instance, by adding correlations between the present and next patterns [5–7] into connections, adaptation terms, and winnerless competition [8]. In these studies, however, the pattern to be generated next is determined by the latest pattern and, thus, flexible change of patterns depending on the context, i.e., the history of the previous patterns and inputs, is not possible. How to implement robust and context-dependent sequential patterns is crucial in neuroscience.

To answer the question, here, we develop a neural network composed of slow and fast neurons to memorize the sequence of patterns. The fast neural dynamics respond to an external input and a feedback signal from the slow dynamics. The slow dynamics store history of the inputs from the fast dynamics and feeds stored information back to it (Fig. 1A). Indeed multiple timescales are observed in a neural system [9–12]; neural activities in sensory cortices evolves faster and respond instantaneously to stimuli, while those in association cortices evolve slower and respond to it (Fig. 1A). Indeed multiple timescales are observed in a neural system [9–12]; neural activities in sensory cortices evolves faster and respond instantaneously to stimuli, while those in association cortices evolve slower and respond to it (Fig. 1A).

FIG. 1. A Schematic diagram of our model for $K = 2, M = 3$. B Neural dynamics during learning process of three targets. Upper panel: the time series of a fast variable $x_0$ (solid line) and a slow variable $y_0$ (broken line) during learning process. Bottom panel: $m_0, 1, 2$, overlaps of $x$ with $\xi_0$ (blue), $\xi_1$ (orange), and $\xi_2$ (green). Black line represents overlap between $x$ and $y$. The bars above the panels indicate which of targeted patterns given to the network in this period.

*kurikawt@hirakata.kmu.ac.jp
FIG. 2. Bifurcation of \( y \) with quenched \( y \). A Neural dynamics during recall process of the learned three. Overlaps of neural activities \( m_{\mu}^y \), \( \mu = 0, 1, 2 \) in \( x \) (upper) and \( y \) (bottom) for \( M = 3 \) are plotted in same color as shown in Fig. 1B. \( y \) is sampled from trajectories at \( (200 < t < 500) \) for the bifurcation diagram of \( X \) shown in B. B Bifurcation diagram of \( x \) as quenched \( y \) is updated with the sampling time. Fixed points of \( \dot{X} \) are shown by projecting into PC1 axis. Small circles indicate fixed points with small basins: neural activity beginning only from the vicinity of the target converges to them. Large circles represent fixed points with large basins: neural activities from most initial states converge to them. To identify fixed points, neural states are plotted after transient period. Colored lines indicate locations of the targets \( \xi_{0,1,2} \) in blue, orange, and green, respectively. Vertical arrows show the transition of \( y \) to different target in recall process. C The neural dynamics for given \( y \) at \( t = 225, 285, 335, 375 \) shadowed in B are depicted by projecting into 2-dimensional principle component (PC) space. 15 trajectories (the three from the vicinity of the target, the other from random initial states) are plotted. Large and small circles represent fixed points given in B. X-shaped marks represent locations of the targets.

A network has to keep the history of the previous patterns to generate the sequence correctly.

To achieve the task, we developed a model of two-populations with different timescales, the one with \( N \) fast neurons and the other with \( N \) slow neurons, denoted as \( X \) and \( Y \), respectively. \( X \) receives an external input and \( Y \) receives the output from \( X \) and gives input to \( X \) as shown in Fig 1A. The neural activities of \( x_i \) in \( X \) and \( y_i \) in \( Y \) evolve according to

\[
\begin{align*}
\tau_x \dot{x}_i &= \tanh (\beta_x I_i) - x_i, \\
\tau_y \dot{y}_i &= \tanh (\beta_y x_i) - y_i, \\
I_i &= u_i + \tanh (r_i) + (\eta^\alpha)_i,
\end{align*}
\]

where \( u_i = \sum_{j \neq i} J_{ij}^{X} x_j, \ r_i = \sum_{j \neq i} J_{ij}^{XY} \tanh (y_j) \). \( J_{ij}^{X} \) is a recurrent connection from \( j \) to \( i \)-th neuron in \( X \) and \( J_{ij}^{XY} \) is a connection from \( i \)-th neuron in \( Y \) to \( j \)-th neuron in \( X \). Mean values of \( J^{X} \) and \( J^{XY} \) are set at zero with the variance equal to \( 1/N \). \( X \) is required to generate the pattern \( \xi_0^\alpha \) at the time upon \( \eta^\alpha \), i.e., an attractor that matches \( \xi_0^\alpha \) is generated. The \( i \)-th element of a targeted pattern, denoted as \( (\xi_0^\alpha)_i \), is assigned to the \( i \)-th neuron in \( X \) and randomly sampled according to probability \( P[(\xi_0^\alpha)_i = \pm 1] = 1/2 \). The context signal \( (\eta^\alpha)_i \) is injected to \( i \)-th neuron in \( X \), randomly sampled according to \( P[(\eta^\alpha)_i = \pm 1] = 1/2 \). We set \( N = 100, \beta_x = 2, \beta_y = 20, \tau_x = 1, \) and \( \tau_y = 100 \).

Only \( J^{X} \) changes to generate the target according to

\[
\tau_{syn} J_{ij}^{X} = (1/N) (\xi_i - x_i)(x_j - u_i J_{ij}^{X}),
\]

where \( \tau_{syn} \) is the learning speed (set at 100). This learning rule consists of a combination of a Hebbian term between the target and the presynaptic neuron and an anti-Hebbian term between pre- and post-synaptic neurons with decay term \( u_i J_{ij}^{X} \) for normalization. The form satisfies locality across connection and is biologically plausible [13]. Previously [13, 14], we have applied the learning rule to a single network only of \( X \) and demonstrated that the network learns \( K \) maps between inputs and targets i.e., \( M = 1 \). In this case, however, a sequence \( (M \geq 2) \) is not possible. In the present study, there are two inputs for \( X \), one from a context signal \( \eta \) and the other from \( Y \) which stores previous information. Thus, the network can generate a pattern depending not only on the present input (context) signal, but also on the previous patterns.

Before the context-dependent sequence, we analyzed if our learning rule generates simple sequences well, namely the pattern to be generated next is determined only by the present pattern. Fig 1B shows a sample learning process for \( K = 1 \). We applied \( \eta^0 \) to a network and presented \( \xi_0^0 \) as the first pattern of a target sequence. After the transient time, \( x \) converges to \( \xi_0^0 \) due to synaptic change. \( y \) follows \( x \) according to Eq. 2 and, consequently, moves to the target. A learning step of a single pattern is accomplished, when the neural dynamics sat-
isfy the following two criteria: \( x \) sufficiently approached the target pattern, i.e., \( m^x = \Sigma_i x_i (\xi^0_j) / N > 0.85 \), and \( y \) is sufficiently close to the fast one, i.e., \( \Sigma_i x_i y_i / N < 0.5 \). After the completion of one learning step, a new pattern \( \xi^1_0 \) is presented instead of \( \xi^0_0 \) with a perturbation of fast variables \( z_i \) by multiplying random number \( s_i \), uniformly sampled from zero to one. We sequentially execute these steps from \( \mu = 0 \) to \( M - 1 \), to learn a sequence. Fast and slow variables learn the next sequence until this procedure is repeated 20 times for each sequence.

We present an example of recall process after the learning process for \( (K, M) = (1, 3) \) in Fig. 2A. In recall, connectivity is not changed. Initial states of the fast variables are set at random values sampled from a uniform distribution from 0 to 1, while slow variables are set at values of the final state of the learning process. The targets appear sequentially in \( X \) in a correct order. Note that, in the recall process, the transition occurs spontaneously without any external application. The present model is able to memorize multiple sequences, say \( M = 11 \) for \( K = 1 \), and \( M = 3 \) for \( K = 2 \).

To examine the robustness of the recall, we explored trajectories from different initial conditions under a gaussian white noise (See Supplemental Materials for details). All of these trajectories converge correctly to a target sequence after some transient periods for weak noise. By increasing the noise strength, recall performance of noisy dynamics is the same as that of noiseless dynamics up to noise strength \( s = 0.3 \). Even by applying strong and instantaneous perturbation to both of \( x \) and \( y \), the trajectory recovers the correct sequence. The sequence is represented as a limit cycle composed of \( X \) and \( y \) and thus recalled robustly.

To elucidate how such robust recall is possible, we analyzed the phase space of \( x \) by freezing \( y \) for the moment. In other words, \( y \) is regarded as a bifurcation parameter for fast dynamics in the following analysis. Specifically, we focused on neural dynamics for \( 200 \leq t \leq 500 \) shown in Fig. 2A. In this period, fast dynamics show transitions from \( \xi^0_0 \) to \( \xi^1_0 \) at \( t = 290 \), from \( \xi^1_0 \) to \( \xi^1_0 \) at \( t = 375 \) and from \( \xi^1_0 \) to \( \xi^1_0 \) at \( t = 220, 460 \). We sampled the slow variables every 5 unit time from \( t = 200 \) to \( 500 \), \( \{y_t = 200, y_t = 205, \ldots, y_t = 500\} \), along the trajectory and analyzed dynamics of \( x \) with slow variables quenched at each of the sampled \( y_t = 200, 205, \ldots, 500 \). Fig. 2B shows the bifurcation diagram of \( x \) against change of \( y \), while Fig. 2C(i) shows the trajectories of \( x \) for specific \( y \).

Now consider the neural dynamics for \( y_r = 225 \) just after the transition from \( \xi^1_0 \) to \( \xi^1_0 \) (Fig. 2C(ii)). For this \( y \), a single fixed point corresponding to the present pattern \( (\xi^0_0) \) exists, leading to its stability against noise. As \( y \) is changed, the basin of \( \xi^0_0 \) shrinks, while a fixed point corresponding to the next target \( (\xi^1_0) \) appears and its basin expands as shown in Fig. 2C(ii). At \( y_r = 290 \), the fixed point of \( \xi^1_0 \) turns to be unstable. Thus, the neural state \( x \) staying at \( \xi^0_0 \) goes out of \( \xi^0_0 \) and falls onto \( \xi^1_0 \). The transition occurs. If the stronger noise is applied, the state will be kicked out of \( \xi^0_0 \) earlier than the noiseless case, resulting in a decrease in the duration of stay at the target (see Supplemental Materials for details).

With further shift of \( y \), \( y_r = 295, 300, \ldots \), there appears a regime of coexistence of \( \xi^0_1 \) and \( \xi^0_1 \) with large basins (Fig. 2C(iii)). The basin of the attractor \( \xi^0_1 \) shrinks and finally banishes (Fig. 2C(iv)) and the transition from \( \xi^1_0 \) to \( \xi^0_0 \) occurs at \( t = 375 \). The next transition from \( \xi^0_0 \) to \( \xi^0_0 \) occurs in the same manner at \( t = 460 \). These processes provide the mechanism for the robust sequential recall: fixed points of \( x \) of the current and next targets coexist and then the former target is unstable by the change in the slow variables.

We explored the success rate of learning. Increasing \( M \) and \( K \) generally leads to a decrease in the success rate of recalls. For \( N = 100 \) and \( K = 1 \), up to \( M = 11 \), the success rate keeps over 80% and decrease rapidly beyond \( K = 12 \). For \( K = 2 \), the success rate is around 80% for \( M = 3 \) and decreases gradually as \( M \) increases (Fig. 3A, see the detailed results in Supplemental Materials). Further, we investigated how the balance between timescales of slow variables \( \tau_y \) and learning \( \tau_{syn} \) affects the success rate. We calculated the success rate as a function of \( \tau_{syn} \) for different \( \tau_y \) by fixing \( \tau_x \) at 1, which are plotted after rescaling \( \tau_{syn} \) by \( \tau_y \) in Fig. 3B. The ratios take a common curve which shows an optimal value ~ 1 at \( \tau_{syn} \) almost same as \( \tau_y \) [17]. The balance between \( \tau_{syn} \) and \( \tau_y \) regulates the success rate, when these are sufficiently smaller than \( \tau_x \). Detailed analysis is given in Supplemental Materials.

We examined if this model learns more complex sequence \( (M = 6) \) in which the same patterns exist in the sequence such as \( \{\xi_0^0, \xi_1^0, \ldots, \xi_5^0\} = (A, B, C, D, B, E) \). The next patterns to \( B \) are \( C \) or \( E \), depending on if the previous pattern is \( A \) or \( D \). Then, neural dynam-
ics have to keep information of the target A or D to recall the target C or E correctly. Our model succeeded in recall of this sequence as plotted in Fig. 4A. When the target B is recalled after the target either A or D, there is no clear difference in the fast variables $x$ indicated by circles in Fig. 4B. However, the slow variables $y$ are different depending on the previous targets in Fig. 4C, which stabilize different patterns of $x$. Further, we demonstrated our model succeeded in recalling more complex sequences ($M = 8$) such as $(\xi_0^0, \xi_1^1, \cdots, \xi_7^7) = (A, B, C, D, E, B, C, F)$ in Supplementary Materials.

As the final example, we explored context-dependent sequences, namely, $(\xi_0^0, \xi_1^1, \xi_2^2) = (A, B, C)$ upon $\eta^0$ and $(\xi_0^1, \xi_1^1, \xi_2^2) = (C, B, A)$ upon $\eta^1$. In this case, the flow $A \rightarrow B \rightarrow C$ on the state space under $\eta^0$ should be reversed under $\eta^1$. The learned network succeeds in generating these sequences as shown in Fig. 4D. Though orbits of $x$ under different signals are almost overlapped in the two-dimensional space, those of $y$ are not. This difference in $y$, in addition to different context signals, allows orbits of $x$ to rotate reversely.

How sequential patterns are generated in neural systems is investigated over several decades [5-8]. In standard methods, Hebbian learning between a target $\mu$ and the next $\mu + 1$, i.e., $J_{ij} = \xi_i^\mu \xi_j^\mu + 1$, allows for sequential patterns [5, 6] [19, 21]. Alternatively, Rabinovich [8] proposed saddle-node connections leading to hetero-clinic orbits for robust sequential patterns. In these studies, however, a pattern and the next are tightly connected with a given connectivity, and context-dependent, flexible recalls are not possible (but see [22]). In our model, in contrast, information about previous patterns is stored in the slow dynamics without a change in the connectivity. The fast dynamics accordingly generate context-dependent sequences. As an alternative method, a reservoir network [23-26] can generate complex sequential pattern, which, however, requires elaborate learning and complex reservoir, so that shaped trajectories are vulnerable to noise [27] in contrast to our model.

Network models with multiple timescales have been studied for sequential patterns. To this end, in [19, 20, 24, 25], a slow term, e.g., an adaptation term, that destabilizes the current pattern is introduced, where an additional mechanism is required for the transition to the next. The feedback from the slow population in our model, in contrast, not only destabilizes the current pattern, but also simultaneously stabilizes the next targeted pattern dependent on the context. Some studies [29, 31] investigated slow dynamics concatenate elements of fast dynamics by using manually designed or elaborate learning algorithms. In our model, contrarily, connections regarding to slow dynamics are not required to be tuned to a specific task, which can allow for the flexibility against different tasks.

Multiple timescale dynamics are widely observed across several cortical areas [10, 12, 32]. In particular, context-dependent modulation in flow structure in neural state space is broadly observed in Hippocampus (HPC) and prefrontal cortex (PFC). In contrast, HPC responds quickly to the location of animals [33] with fast timescale than PFC, one of the slowest cortical areas [11]. In context-dependent decision making [34], PFC represents the context and HPC is modulated by PFC. Our model is
in line with these experimental observations and presents a basic, general framework of how a multiple timescale system provides context-dependent sequential recall, essential to cognitive functions.

I. ACKNOWLEDGEMENTS

We thank Omri Barak for fruitful discussion. This work was partly supported by JSPS KAKENHI (nos. 18K15343 and 20H00123).

[1] L. M. Jones, A. Fontanini, B. F. Sadacca, P. Miller, and D. B. Katz, Proceedings of the National Academy of Sciences 104, 18772 (2007).
[2] A. Ponce-Alvarez, V. Nacher, R. Luna, A. Riehle, and R. Romo, Journal of Neuroscience 32, 11956 (2012).
[3] C. D. Harvey, P. Coen, and D. W. Tank, Nature 484, 62 (2012).
[4] A. M. Wikenheiser and A. D. Redish, Current Opinion in Neurobiology 32, 8 (2015).
[5] D. Kleinfeld, Proceedings of the National Academy of Sciences 83, 9469 (1986).
[6] H. Sompolinsky and I. Kanter, Physical Review Letters 57, 2861 (1986).
[7] H. Nishimori, T. Nakamura, and M. Shiino, Physical Review A 41, 3346 (1990).
[8] P. Seliger, L. S. Tsimring, and M. I. Rabinovich, Physical Review E 67, 011905 (2003), arXiv:0205026 [nlin].
[9] R. Chaudhuri, K. Knoblauch, M. A. Gariel, H. Kennedy, and X. J. Wang, Neuron 88, 419 (2015).
[10] U. Hasson, J. Chen, and C. J. Honey, Trends in Cognitive Sciences 19, 304 (2015).
[11] J. D. Murray, A. Bernacchia, D. J. Freedman, R. Romo, J. D. Wallis, X. Cai, C. Padoa-Schioppa, T. Pasternak, H. Seo, D. Lee, and X.-J. Wang, Nature neuroscience 17, 1661 (2014).
[12] C. J. Honey, T. Thesen, T. H. Donner, L. J. Silbert, C. E. Carlson, O. Devinsky, W. K. Doyle, N. Rubin, D. J. Heeger, and U. Hasson, Neuron 76, 423 (2012).
[13] T. Kurikawa and K. Kaneko, PLoS computational biology 9, e1002943 (2013).
[14] T. Kurikawa, O. Barak, and K. Kaneko, arXiv preprint (2019), arXiv:1906.11770.
[15] For the input from Y to X, we considered two nonlinear filters by the hyperbolic tangent function under the following biological assumptions. First, for \(\tanh(y_i)\), the activity of \(y_i\) is assumed to be amplified in a nonlinear way at a synapse onto \(x_i\). Second, for \(\tanh(r_i)\), we considered a big branch of dendrite of \(x_i\) to which all inputs from \(Y\) are injected and assumed that activity of the branch (i.e., summation of total inputs from \(Y\)) surges beyond the threshold.
[16] The other fixed point corresponding to \(\xi^0\) also appears, but its basin is quite small. Thus, we can neglect this fixed point.
[17] For \(\tau_y = 10\), which are close to \(\tau_x\), the success rate cannot reach the optimal value.
[18] We have to tune the strength of external input \(\eta\) in Eq. 3 at 1.3. Still, success rate is small, just over 10%.
[19] S. Recanatesi, M. Katkov, S. Romani, and M. Tsodyks, Frontiers in Computational Neuroscience 9, 1 (2015).
[20] E. Russo and A. Treves, Physical Review E 85, 1 (2012), arXiv:1203.1067.
[21] T. Haga and T. Fukai, Physical Review Letters 123, 78101 (2019).
[22] S. O. Verduzco-Flores, M. Bodner, and B. Ermentrout, Journal of computational neuroscience 32, 403 (2012).
[23] W. Maass, T. Natschl"ager, and H. Markram, Neural Computation 14, 2531 (2002).
[24] H. Jaeger, Science 304, 78 (2004).
[25] D. Sussillo and L. F. Abbott, Neuron 63, 544 (2009).
[26] D. V. Buonomano and W. Maass, Nature Reviews Neuroscience 10, 113 (2009).
[27] R. Laje and D. V. Buonomano, Nature neuroscience 16, 925 (2013).
[28] C. Gros, New Journal of Physics 9 (2007), 10.1088/1367-2630/9/4/109, arXiv:0705.0078.
[29] D. Perdikis, R. Huys, and V. K. Jirsa, PLoS computational biology 7, e1002198 (2011).
[30] Y. Yamashita and J. Tani, PLoS computational biology 4, e1000220 (2008).
[31] J. Fonollosa, E. Netci, and M. Rabinovich, PLOS Computational Biology 11, e1004592 (2015).
[32] C. A. Runyan, E. Piasini, S. Panzeri, and C. D. Harvey, Nature 548, 92 (2017).
[33] D. Kumaran, D. Hassabis, and J. L. McClelland, Trends Cogn Sci 20, 512 (2016).
[34] K. G. Guise and M. L. Shapiro, Neuron 94, 183 (2017).
II. SUPPLEMENTAL MATERIALS

A. recall process

Our model memorizes several sequences for different context signals. We exemplify procedure of memorization by focusing on the learning and recall process for \( K = 2 \). Learning two sequences are accomplished in similar way to learning a single sequence analyzed in the main text. The model learns two sequences alternatively: the first sequence, \( \xi_0^1, \xi_1^1, \ldots, \xi_{M}^1 - 1, \xi_0^1 \), is learned once with the same criteria for \( K = 1 \). After resetting the fast and slow variables, the second sequence, \( \xi_0^1, \xi_1^1, \ldots, \xi_{M}^1 - 1, \xi_0^1 \), is learned in the same way. We repeated 20 times of these processes before finishing the learning.

Fig. 6A shows a recall process for \( K = 2, M = 4 \) after learning. In the presence of \( \eta^0 \), the sequence \( \xi_0^0, \xi_1^0, \xi_2^0, \xi_3^0 \) is recalled as shown. Then, after switching the input from \( \eta^0 \) to \( \eta^1 \) at \( t = 1000 \), the required sequence \( \xi_0^1, \xi_1^1, \xi_2^1, \xi_3^1 \) is recalled successfully.

B. Robustness of sequence

We investigated the robustness in the sequence recall. First, we applied strong one-shot perturbation into neural dynamics, where multiplicative noise is added into neural activities of all neurons \( x_i \) and \( y_i \), as \( x_i \rightarrow (1 - r_i^x)x_i, y_i \rightarrow (1 - r_i^y)y_i \) \( (i = 0, 1, \ldots, N \) and \( r^x,y \) is randomly chosen from the uniform distribution from 0 to 1). The trajectory with the one-shot perturbation is plotted in Fig. 6A. After the perturbation, the neural dynamics rapidly recover to a limit cycle in which the neural activity exhibits the transition from one target to another in the order.

Next, we examined the robustness against the change in the initial states and noise. Here, Gaussian white noise \( \zeta(t) \) is added satisfying \( <\zeta(t)\zeta(t')> = s \delta_{ij} \delta(t - t') \) for \( i = j \), otherwise 0 into neural dynamics \( x \) and \( y \) in Eqs. (1), where \( \delta_{ij} \) is the Kronecker’s and Dirac’s delta, respectively, and \( s \) is noise strength. Fig. 6B shows a trajectory from a random initial state under the noise by using the overlaps of the slow and fast variables for \( K = 1, M = 5 \). After transient period, the trajectory converges to the limit cycle that generates the correct sequence recall. We tested other 9 trajectories under noise from 9 random conditions and found that all trajectories converge to the limit cycle.

Further, the robustness against the noise strength is examined. Dynamics of \( X \) for increasing the noise strength are plotted in Fig. 6C. Below \( s = 0.3 \), the sequence is recalled with the correct order. For much stronger noise \( (s = 0.5) \), a few patterns appears intermittently and the others do not appear. Fig. 6D(i) shows the success rate of recalls as a function of the noise strength. The success rate keeps around 0.8 (same as the ratio in the case without noise) up to \( s = 0.1 \) and decreases rapidly. All of these results demonstrate that sequential patterns in our model are quite robust against the change in the initial states and noise.

Finally we measured the duration time in which the fast dynamics stay on each target. The duration time is normalized by that measured in the neural dynamics without noise. The normalized duration time is plotted as a function of the noise strength in Fig. 6D(ii). We found that the normalized duration time decreases as the noise strength increases. After the fast dynamics converged to the target attractor, its basin volume reduces over time as analyzed in Fig. 2. Thus, stronger noise is likely to kick out the neural state from the target earlier, resulting in the decrease in duration time as the noise strength increases.

C. Adequate balance between timescales of slow variables and learning shapes a fine structure of bifurcation

How does the balance of timescales affect memory capacity? First, we present a typical failed recall for \( \tau_y >> \tau_{syn} \), \( (\tau_y = 100, \tau_{syn} = 10) \) in Fig. 7B. Some of targets are recalled sequentially, but in a wrong order, while other targets do not appear in the recall process. To clarify the underlying mechanism of the failed recall, we analyze neural dynamics of fast variables with slow variables quenched in a similar manner to Fig. 2 whereas, in contrast to the analysis in Fig. 2 for \( \tau_y = 100, \tau_{syn} = 100 \), slow variables are sampled from the trajectory in the final learning of the sequence (after learning the sequence 19 times) because all of targets does not appear in the recall process. Fig. 7D depicts a bifurcation diagram of the fast variables for the same network shown in Fig. 7B. We found all the targets are stable for certain \( y \), although \( \xi_0^1 \) does not appear in the recall process. We also found that the coexistence of fixed points corresponding to \( \xi_0^0 \) and \( \xi_0^1 \) does not appear: the fixed point corresponding to \( \xi_0^2 \) has a large basin across all of \( y \). This leads to transition from \( \xi_0^0 \) to \( \xi_0^2 \) by skipping \( \xi_0^1 \) and then generates the failed recall.

Interestingly, failed recalls for \( \tau_y << \tau_{syn} \) are distinct from those for \( \tau_y >> \tau_{syn} \). For \( \tau_y = 100, \tau_{syn} = 1000 \), only the most recently learned target is stable for almost of \( y \) and thus only this target is recalled in Fig. 7C. We sampled slow variables from the final learning of the sequence and analyzed bifurcation structure of the fast variables against the sampled slow variables in the same way as above. Fig. 7E shows the bifurcation diagram of \( x \). For all of \( y \), only the latest target (here, \( \xi_2^2 \)) is a fixed point and the others are not. Thus, transitions between targets are missed except that to the latest target.

D. Recall of context-dependent sequence

We demonstrate that our model succeeded in recalling context-dependent sequences \( (M = 8) \) such as
FIG. 5. Recall dynamics for $K = 2, M = 4$, when the context is switched at $t = 1000$. The fast variables (the upper panel) and slow ones (the lower panel) are plotted by using the overlap $m^\mu_\alpha$ ($\alpha = 0, 1$ and $\mu = 0, 1, 2$). Index of the overlap is indicated below the panels.

$$(\xi_0^0, \xi_1^0, \xi_2^0, \xi_3^0, \xi_4^0, \xi_5^0, \xi_6^0, \xi_7^0) = (A, B, C, D, E, B, C, F)$$

in Fig. 8. In this case, neural dynamics have to keep previous three targets in memory to recall the target $D$ or $F$ after $BC$ correctly. Although the difference in the activities of $Y$ after recalling $ABC$ or $EBC$ is quite small, it is sufficient to discriminate which of sequences should be recalled. By using this difference, the fast variables change to the next different pattern correctly.
FIG. 6. **A and B** Trajectory of the overlaps of $x$ (upper panel) and $y$ (lower panel) is plotted. The trajectory with the targets for noise strength $s = 0.1$ is shown in A and that with one-shot perturbation (at $t = 150$) in B. Each color indicates the target used in calculation of the overlap (the same color code is used in the following panels). **C** The time series of $x$ are plotted by using the overlaps. The realization of the network and the target and the context signal patterns are identical across panels, whereas the noise strength $s$ is increased from upper to lower panels. **D** The success rate and normalized residence time at each pattern is plotted against the noise strength $s$ in (i) and (ii), respectively. The success rate is defined in the same manner as in Fig. 5A and calculated across 25 realizations of networks and target and context patterns The normalized residence time is defined in the text and obtained from only success recalls out of 25 realizations. Dots in (ii) indicate the duration time of different realizations.
FIG. 7.  A The success rate of recalls as functions of $\tau_{syn}$ for given $\tau_y$. Different color represents different $\tau_y$ indicated by bars above the panels. The success rate is calculated across 50 realizations. The curve for $\tau_y = 1000$ is not shown in Fig. 3B for visibility. B and C The neural activities of $X$ (upper) and $Y$ (lower) in the recall process are shown by using the overlaps with the targets. Index of the used targets are indicated by bars above panels. The neural dynamics for $\tau_y = 100$, $\tau_{syn} = 10$ are shown in B, while those for $\tau_y = 100$, $\tau_{syn} = 1000$ in C. D and E The bifurcation diagrams of the fast variables with quenched $y$ are shown in the same manner as shown in Fig. 2B. The bifurcation diagrams corresponding to B and C are plotted in D and E, respectively. The fixed points are plotted as circles and colored lines represent locations of the target 0 (blue), 1 (orange), and 2 (green) by projection onto 1st principle components in the upper panels. In the lower panels, sampled $y$ from the learning process are plotted by using the overlaps with the same color code with B and C.

FIG. 8. Recall processes for context-dependent sequences for $K = 1, M = 8$. A The neural activities of $X$ upon $\eta^0$ are plotted by using their overlaps with targets. Colors and alphabets indicate which of targets is overlapped. In B and C, neural dynamics plotted in A are shown by projecting the fast dynamics (B) and the slow (C) onto 2-dimensional PC space. X-shaped marks represent the locations of the targets.