BIOLOGICAL SCIENCES, Neuroscience

Computational mechanisms for temporal integration in the anterior claustrum

Kuenbae Sohn, Donghyeon Yoon, Junghwa Lee* & Sukwoo Choi*

School of Biological Sciences, College of Natural Sciences, Seoul National University, 1 Gwanak-ro, Seoul 151-747, Korea,

K.S. and D.Y. contributed equally to this manuscript.

**Correspondence should be addressed to Dr. Junghwa Lee and Dr. Sukwoo Choi, E-mail: antares0715@gmail.com or sukwoo12@snu.ac.kr*

Author contributions: K.S., J.L., and S.C. designed research; and K.S., and D.Y. performed research; K.S., D.Y., J.L., and S.C. analyzed data; and K.S., D.Y., J.L., and S.C. wrote the paper.

Key words: Claustrum, Recurrent Neural Networks, Integration, Trajectory, Dynamic coding

Abstract

The claustrum, with its extensive reciprocal connections to nearly all cortical regions, has long been hypothesized as a key hub for integrating diverse cognitive, sensory and motor information. However, despite its anatomical connectivity, whether and how it functionally integrates different inputs to generate coherent representations has remained unclear. Here, we developed a recurrent neural network (RNN) trained via supervised learning on behavioral metrics of delayed escape—a task that requires inference-based escape behavior and depends critically on anterior claustral activity. A subset of RNN neurons exhibited dynamics similar to those of anterior claustral neurons during this behavior. These neurons formed a recurrent cluster, a structure supported by in vitro stimulation experiments in claustral brain slices. We analyzed the computational properties of this claustrum-like cluster via dimensionality reduction of population activity. The network showed nonlinear integration of temporally distributed inputs and increased synergistic information. Rather than settling into attractors, integrated information was dynamically encoded along continuously evolving neural trajectories. Notably, the evolving trajectories that reflected dynamic integration in the RNN model matched those approximated in claustral recordings, suggesting the model's biological plausibility. We propose that the anterior claustrum dynamically integrates task-relevant input signals over time and broadcasts the evolving representation to downstream brain regions capable of reading and interpreting it in a context-dependent manner.

Introduction

The claustrum is a thin, elongated sheet of gray matter that maintains extensive reciprocal connections with nearly all cortical areas and several subcortical regions. This dense interconnectivity has led to the longstanding hypothesis that the claustrum serves as a hub for integrating internal and external signals into a unified percept, potentially supporting conscious awareness and higher-order cognition (1). However, direct experimental evidence for this hypothesis has remained scarce. While one recent study reported that individual claustral neurons can receive convergent input from multiple cortical areas, which is critical for a behavioral task requiring multimodal sensory inputs (2), most previous studies have either not supported this hypothesis directly or yielded results inconsistent with large-scale multimodal

integration, instead suggesting alternative or more limited functions for the claustrum (3-8). This apparent discrepancy may reflect differences in the specific claustral subregions examined. Although the claustrum is less differentiated in rodents than in primates, important distinctions exist even among rodent species (9). In mice, the anterior boundary of the claustrum typically does not extend beyond the rostral end of the striatum (10). In contrast, in rats, the claustrum extends further rostrally beneath the forceps minor of the corpus callosum (fmi), where claustrum-specific gene expression has been identified (11, 12).

Only a few previous studies have directly investigated this rostral segment of the rat claustrum (rostral-to-striatum, rsCla) (12-16). However, recent findings suggest that neuronal activity in this region plays a critical role in behavioral tasks in which two different kinds of information must be integrated (11, 17), indicating a potentially distinct computational role for the rsCla. Among these tasks, the newly developed delayed escape task provides an inference-based behavioral paradigm in which rats must rapidly escape to a neutral zone following a fear-inducing conditioned stimulus (CS), after a delay that separates the CS from the opening of an outlet. Crucially, the task requires no prior training, thus relying on the flexible integration of two temporally separated events that initially appear unrelated but, through inference, acquire common task relevance. Neural recordings from this task revealed that CS-related information was maintained through persistent activity in rsCla neurons, suggesting that these neurons may integrate the sustained internal threat signal with the subsequent outlet-opening signal to guide future escape behavior.

To investigate how the anterior claustrum integrates information at the population level, we employed a delayed escape task in Long-Evans rats. Because the task could only be performed once per animal, and only a small number of single units were recorded per subject, direct population-level analysis was severely limited. To overcome these constraints, we trained a recurrent neural network (RNN) model solely on behavioral outputs—a strategy previously shown to yield biologically relevant dynamics (18-24). We then compared the network's emergent population activity patterns with in vivo claustral recordings, and used the model to explore circuit-level mechanisms of integrating temporally separated inputs.

Results

RNNs trained on the Delayed Escape Task recapitulate claustrum-like dynamics

The delayed escape task involves inference-based behavior and allows only a single test trial per animal, which means that opportunities to obtain neurophysiological data are extremely limited. To overcome this limitation, we used recurrent neural network (RNN) simulations, a strategy that has been widely adopted in recent neuroscience research (20, 21, 25). In the delayed escape task, when a conditioned stimulus (CS) previously associated with electric shock is presented for the first time in a novel environment, animals infer that a value-neutral alternative space is likely to be safer and therefore exhibit faster escape behavior (Fig. 1A). Importantly, the passage to this neutral space opens after a delay following CS offset, making it essential for CS-related information to be maintained beyond the actual presence of the CS. As a control group, animals were placed in the same environment during the CS period but did not receive the CS. In previous experiments, animals that received the CS escaped to the neutral space significantly faster than controls (11). To model this behavior, we built upon the continuous-time RNN framework developed by Ehrlich et al. (2021) (18) and Ehrlich & Murray (2022) (19) (Fig. 1B).

The trained RNN reproduced the escape latency pattern reported by Han et al. (2024) (11): the CS + door-opening condition escaped faster than the control door-opening only group (Fig. 1C), reflecting the in vivo behavioral results. Furthermore, it closely mimicked the claustral network dynamics observed in that study (Fig. 1D–L). In the previous biological study, the responses of individual claustral units representing each behavioral epoch were subjected to dimensionality reduction and clustering (Fig. 1D). One of these clusters showed persistent firing during and after CS presentation, and the magnitude of this increase was inversely correlated with escape latency (Fig. 1E–G). From this, we concluded that this cluster maintained CS signals until the door to the escape path opened, thereby contributing to delayed escape.

To determine whether this cluster was reproduced in the trained RNN, unit responses in the RNN representing each behavioral epoch were also subjected to dimensionality reduction and clustering. One of the clusters displayed the same persistent activation during and after CS presentation as observed in the in vivo claustrum (Fig. 1H–J). As in the biological in vivo data, stronger activation in this model cluster predicted shorter escape latency (Fig. 1K). Single-unit responses were also nearly identical between real and simulated data: the persistence after the CS did not arise from a single unit firing continuously but rather from many units intermittently increasing their firing, which overlapped to form a persistent signal at the population level (Fig. 1L).

Even when no CS was presented and only the door opening cue was given (the door-opening condition), the population activity pattern generated by the RNN showed a pattern of gradually increasing after the door opened and then decreasing as the crossing approached (Fig. S1A). The other two model clusters exhibited different patterns: Cluster 2 was generally suppressed regardless of CS presence, and Cluster 3 gradually ramped up from the door opening cue to just before the crossing, also regardless of CS presence (Fig. S1).

Next, we quantified the contributions of the three clusters to the RNN output when the CS was presented. The claustrum-like Cluster 1 contributed very little to the output, whereas Cluster 3 was dominant in influencing cross behavior. This suggests that Cluster 1 may not directly drive the output but instead processes the CS and door opening signals and broadcasts this information to other clusters (Fig. S2). In our previous study (11), optogenetic inhibition of the anterior claustrum during the delay period between CS offset and door opening significantly increased escape latency (Fig. 1M, Fig. S3A). Introducing inhibitory input to suppress RNN Cluster 1 during the same period produced a similar increase in latency (Fig. 1M), whereas inhibiting the other clusters had no effect (Fig. S3B). Furthermore, when the interval between CS and door opening was extended from 5 s to 180 s in vivo (11), the latency difference between the CS and neutral groups disappeared. Indeed, the RNN also reproduced this result under the same extended interval (Fig. S3C).

Taken together, the claustrum-like Cluster 1 in the RNN not only reproduced the firing pattern and behavioral correlation of the persistent activity cluster observed in vivo but also successfully mimicked the characteristics of actual claustral neurons in almost every respect, including latency increases following inhibition. This trained RNN model therefore enables deeper analyses under experimental conditions that are otherwise difficult to implement.

Claustral neurons exhibit recurrent connectivity similar to that of the RNN

Because a recurrent neural network (RNN), by definition, contains recurrent connections among its units, finding a claustrum-like cluster whose properties mirror those of the real claustrum suggests that biological claustral neurons may likewise be interconnected through recurrent circuitry. As expected, the RNN exhibited strong recurrent weights within Cluster 1 (Fig. 2A). If a similar architecture exists in vivo, the first prediction is that excitatory connectivity should be detectable between claustral neurons.

To test this, we sparsely expressed ChrimsonR in a subset of claustral cells and prepared sagittal brain slices for patch-clamp recordings to measure synaptic currents (Fig. 2B–C). We activated ChrimsonR-expressing presynaptic neurons with 617 nm light, while patching ChrimsonR-negative neurons to isolate synaptic responses. Patching ChrimsonR-positive cells would have risked contamination by direct photocurrents in addition to synaptic responses. Wide-field illumination induced excitatory postsynaptic currents (EPSCs) that were abolished by NBQX + D-AP5, confirming that they were mediated by glutamatergic transmission (Fig. 2D–E). One remaining concern was that these EPSCs might have originated from ChrimsonR-expressing axons projecting into the claustrum from outside regions. To rule this out, we employed high-resolution optical mapping. The wide field was divided into a 30 × 30 grid (900 pixels), and each pixel was stimulated for 2ms (Fig. 2F). EPSCs could be elicited not only near the recorded soma but also from distant pixels, indicating the presence of local excitatory-to-excitatory connectivity within the claustrum itself (Fig. 2G–H). These findings are consistent with the recent study by Shelton et al. (2025) (2) but contrast with reports that the posterior claustrum exhibits little excitatory-to-excitatory coupling (26).

A more stringent prediction of recurrent wiring is that a brief, strong stimulus should induce sustained neuronal activity. To test this, we expressed the calcium sensor GCaMP6f in the anterior claustrum and delivered a 1 s train of electrical stimulation (20 Hz, 200 μA) in horizontal slices (Fig. 2I–K, Fig. S4C). This stimulation evoked calcium signals that remained elevated for over 10 s. Consistently, patch-clamp recordings from GCaMP6f-positive cells confirmed a prolonged increase in firing rate over a similar period (Fig. S4B). Notably, coronal slices failed to show such persistence, suggesting that horizontal slicing better preserves recurrent excitatory circuits. To determine whether this persistent activity depends on synaptic reverberation, we locally pressure-injected NBQX + D-AP5 at the stimulation site during the sustained phase of the calcium signal. As predicted, the persistent response collapsed rapidly (Fig. 2L–M, Fig. S4D–F), indicating that excitatory synaptic transmission within the claustrum is essential for generating the sustained activity.

The RNN reproduced a similar phenomenon (Fig. 2N, Fig. S4F). Isolating Cluster 1 from the trained network, we reduced all inhibitory weights by 60%. When a 1 s CS-strength input was delivered, the cluster exhibited >10 s of persistent activity, matching the slice data. Reducing inhibitory weights facilitates the persistent activity, which may reflect the reduced inhibitory connectivity often present in horizontal slice preparations. Next, reducing 60% of excitatory weights for only 10% of Cluster 1 units—mimicking the spatially restricted application of NBQX

and D-AP5—largely abolished the persistence (Fig. 2N, Fig. S4D–F). Together, these slice and RNN experiments suggest that the anterior claustrum contains recurrent excitatory connections capable of supporting prolonged activity.

PCA-based trajectory analysis characterizes RNN dynamics

Based on the results so far, the claustrum-like cluster in the RNN resembles the cluster showing persistent activity after CS presentation in the anterior claustrum. To visualize the population activity dynamics of the RNN clusters—particularly the claustrum-like cluster— we applied principal component analysis (PCA) to z-scored firing rates from three conditions: CS + door-opening, door-opening only, and CS only (Fig. 3A–E, Fig. S5–6). At each time point, the ensemble firing rates across all neurons in each cluster were treated as a population activity vector and projected into a three-dimensional space defined by the top 3 principal components. Because cross-latency (the time from door opening to actual crossing) varied across trials, the interval from door opening to crossing was time-normalized within each trial. For the CS only condition, in which cross-latency was undefined, we used the mean cross-latency from the CS + door-opening condition to define a notional crossing time.

In the CS + door-opening condition, the mean trajectory moved away from the start region after CS presentation and progressed along a curved path. Notably, a short-loop pattern appeared immediately after the door opened—5 seconds after CS offset (Fig. 3A). In contrast, in the door-opening only group, the trajectory remained near the start region until the door opened and then followed a long-curved path toward the crossing point (Fig. 3B). As expected, the CS only group exhibited a trajectory similar to that of the CS + door-opening condition until 5 seconds after CS offset, after which it returned toward the start region (Fig. 3C). Interestingly, in two manipulations known to delay crossing latency—(1) inhibition of neural activity during the 5-second interval between CS offset and door opening, and (2) extension of this interval to 180 seconds—the short loop observed in the CS + door-opening group was not observed (Fig. 3D, E).

Sudden changes in trajectory shape, such as the short loop observed in the CS + door-opening condition, have been reported in previous studies of other brain regions to be associated with specific brain functions (27). Therefore, based on the assumption that the persisting CS signal and the door-opening signal combine to drive integration, we examined their interaction during this period. To test this, we assumed that the post-CS segment of the CS only trajectory

(starting 5 seconds after CS offset) and the post-door-opening segment of the door-opening only trajectory each represent the respective inputs to the integration process. We then tested whether their combination could explain the post-door-opening trajectory in the CS + door-opening condition (Fig. 3F-G, Fig S6). A linear combination model failed to reconstruct the early time bins immediately after door opening, likely due to the distinctive short-loop shape. In contrast, a nonlinear model (multilayer perceptron, MLP) achieved a lower residual sum of squares (RSS), outperforming the linear model (Fig. 3G, Fig S6A). The difference in model fit between the nonlinear and linear models was significantly larger in the CS + door-opening condition than in either the inhibition or 180-second interval conditions (Fig. 3G). Notably, the difference in the CS + door-opening condition was specific to the claustrum-like cluster, as the other two clusters showed no comparable difference (Fig. S6B–D). Taken together, these results suggest that nonlinear integration occurs in the claustrum-like cluster specifically during the short-looping period immediately following door opening in the CS + door-opening condition.

Partial information decomposition uncovers neurons carrying synergy between CS and door-opening signals

Trajectory analysis provided evidence for nonlinear integration at the population activity level. To examine the contribution of individual neurons to this nonlinear integration, we measured the synergistic information for each neuron—i.e., information encoded only when both stimuli, CS and door opening, were present. To this end, we performed Partial Information Decomposition (PID) analysis (28, 29). PID quantifies how two input variables (CS and door opening) contribute to the information contained in the output neural activity by decomposing the total mutual information into three components: (1) unique information provided independently by each input, (2) redundant information shared by both inputs, and (3) synergistic information generated only when both inputs are present simultaneously (Fig. 3J).

In the trained RNN model (as well as in real claustral neurons), the effect of an input stimulus persists in neural activity even after the physical stimulus has ended. Therefore, to define stimulus strength for PID analysis, we used the output of trained decoders—which infer the presence or absence of each stimulus from neural activity—as the effective input variable in the PID computation. We trained two decoders using the z-scored activity of neurons in the RNN claustrum-like cluster: one to distinguish between CS-present and CS-absent trials, and another to classify door-open versus no-door-open trials (Fig. 3H–I, Fig. S7; see Methods). The CS

decoder performed at chance level during the baseline period but sharply increased in accuracy immediately after CS onset, with performance remaining high well beyond the end of CS presentation (Fig. 3H). The door-opening decoder showed high accuracy during the period when the door signal was present (Fig. 3I).

We used the CS decoding accuracy score and door-opening decoding accuracy score as inputs, and the z-scored activity of claustrum-like cluster neurons in each trial as the output. PID was computed for each neuron over time (Fig. 3K–O, see Methods). As expected, CS-unique information rose following CS onset and gradually decayed, whereas door-opening-unique information increased after door opening (Fig. S7C–D). Importantly, synergistic information—representing new information generated only when CS and door-opening representations were jointly present—increased immediately after door opening and then gradually decreased (Fig. 3L). This suggests that neurons in the claustrum-like cluster integrate CS and door-opening signals to produce new, combined information. Moreover, synergistic signals were reduced in two conditions where nonlinear integration was disrupted: when inhibition was applied to the network, and when the CS–door interval was extended to 180 seconds (Fig. 3M–N).

We further divided claustrum-like cluster neurons into the top 25% with the highest synergy (“high-synergy” neurons) and the bottom 25% (“low-synergy” neurons) and analyzed their trajectory features (i.e., the short loop). In high-synergy neurons, the curvature of the neural trajectory immediately after door opening exhibited a more pronounced turning-angle change compared to low-synergy neurons (Fig. S8A–B). Furthermore, when CS + door-opening trials were predicted using trajectories from CS-only and door-opening-only conditions, nonlinear prediction errors were greater for high-synergy neurons than for low-synergy neurons (Fig. 3O). Together, these findings suggest that synergistic information at the single-neuron level is related to the short loop and that, within the claustrum-like cluster, certain neurons play a more dominant role in the nonlinear integration process.

Trajectory-based dynamic coding underlies integration

Through trajectory analysis and PID, we have demonstrated that a subset of neurons within the claustrum-like cluster nonlinearly integrates information about the CS and door-opening. However, the exact mechanism by which this nonlinear integration is implemented remains unclear. To assess the temporal evolution and stability of the integrated representation, we applied cross-temporal decoding. This approach allowed us to examine whether the combined

information from CS and door-opening is encoded in a static manner or in a time-varying fashion.

Using cross-temporal decoding, we examined how the neuronal population encodes information unique to the CS + door-opening condition over time. For each corresponding time bin across the three conditions (CS+Door, CS-only, and Door-only), we trained a decoder to distinguish the CS+Door condition from the other two, and evaluated its classification accuracy across all time bins. In the entire population of claustrum-like cluster, cross-temporal decoding revealed a mixed coding regime: shortly after door opening, decoders trained at specific time bins maintained discrimination power across relatively broad temporal windows (Fig. 4A). However, as time progressed, the discriminative power of decoders became temporally confined, performing well only near the training bin.

When neurons in the claustrum-like cluster were divided into high- and low-synergy groups and decoders were trained and tested separately on each, high-synergy neurons exhibited a dynamic coding pattern. Decoders trained on specific time bins in this group showed strong performance at that time point, with limited generalization to neighboring bins. In contrast, decoders trained on low-synergy neurons maintained more temporally stable performance. These results suggest that high synergy neurons, which contribute critically to nonlinear integration, encode CS and door-opening information using a dynamic coding scheme, whereas low-synergy neurons may support a more stable representation (Fig. 4B–C).

This distinction was further supported by raw firing-rate heatmaps: both high and low synergy neurons showed increased activity following CS onset, but after the door opened, only high-synergy neurons displayed variable firing patterns (Fig. 4D, left). Notably, a similar pattern of transient, variable spiking was observed in our in vivo recordings of claustral single units after door opening in the CS + door-opening group (Fig. 4D, right).

Temporal evolution of the population state can be framed by two hypotheses (Fig. 4E). The attractor hypothesis posits that neural dynamics consist of discrete or continuous attractors, with the integrated state represented by transitions among them in response to varying inputs. The trajectory hypothesis proposes that integration is embodied in the evolving trajectory itself, independent of fixed attractors. To distinguish these views, we performed fixed-point analysis on the RNN, locating states where the velocity vector is nearly zero (30). Fixed points were found

to lie far from the empirical trajectories, indicating that the network's dynamics are trajectory-based rather than attractor-based when integrating CS and door-opening cues (Fig. 4F).

Trajectory features predicted by the RNN are recapitulated in biological recordings

As mentioned earlier, our previous study (11) was limited by the use of single-trial testing, a small number of recorded units per rat, and a limited total number of animals, which restricted both the overall population size and the feasibility of in-depth analyses. To address these limitations, we constructed a trained RNN model that produced reproducible and testable trajectories, particularly those exhibiting short-loop dynamics suggestive of information integration. To qualitatively compare the model-derived trajectories with empirical neural data, we applied Gaussian Process Factor Analysis (GPFA) (31, 32), a dimensionality reduction method well-suited for single-trial time-series data. Although GPFA is typically optimized with multiple trials, it can still recover low-dimensional latent trajectories by modeling shared activity across neurons while explicitly accounting for neuron-specific noise. By applying GPFA to single-trial unit data collected from multiple animals and embedding it into a three-dimensional latent space, we were able to visualize population dynamics in a manner directly comparable to those generated by the RNN, thereby bridging the gap between model prediction and experimental observation (Fig. 4G). Interestingly, the trajectory derived from units recorded in CS rats—the experimental group corresponding to the CS + door-opening condition in the RNN model—resembled the short-loop structure observed in the RNN (Fig. 4G, left). The onset of the short loop was markedly delayed compared to that in the claustrum-like cluster of the RNN. In contrast, trajectories from neutral rats, which received a neutral CS (a light cue not associated with an electrical shock) during testing, exhibited a gentler curvature (Fig. 4G, right). To validate the reliability of this structure, we partitioned the neurons into two subgroups and independently repeated the GPFA analysis. Neurons with comparable basal firing rates were paired and subsequently classified into two groups. To compare trajectories before and after the split, the behavioral reference points on the post-split trajectories were aligned to those on the pre-split trajectories. The resulting latent trajectories remained qualitatively similar in shape and evolution (Fig. 4H), indicating that the short-loop structure was broadly distributed across the neuronal population and represents a robust population-level phenomenon unlikely to arise from random fluctuations or noise.

Discussion

In this study, we trained a vanilla recurrent neural network (RNN) solely on behavioral metrics from the delayed escape task, which had previously been shown to require rsCla neurons for inference-based escape (11). Remarkably, the trained network developed a claustrum-like cluster whose dynamics closely resembled not only the experimentally observed population activity patterns but also the causal relationship between neural activity and behavior. Low-dimensional trajectory analysis revealed a nonlinear integration of the conditioned stimulus (CS) and the temporally delayed door-opening signal, accompanied by an increase in synergistic information. Rather than converging to stable attractor states, the network encoded these signals through continuously evolving trajectories. In the post-integration phase, the trajectories formed a sharply curved “short loop” immediately after door opening—an architecture also observed in real claustrum recordings, thereby supporting the biological plausibility of the model. Such temporal integration through dynamic coding enables flexible responses to inputs that arrive in the claustrum with temporal gaps and offers the advantage of generating richer representations compared with stable coding (21-23, 25, 27, 33). Moreover, contrary to the intuitive expectation that dynamically coded information might be difficult to decode, it can, in fact, be easily read out using simple linear decoders (34), which provides an additional advantage.

Previous studies have shown that the anterior claustrum is largely unresponsive to simple sensory stimuli, becoming active only when such stimuli acquire behavioral relevance possibly through convergent inputs from regions encoding memory, value, context, and motivational states (7, 11, 17, 35-38). It also exhibits stronger responses to cross-modal stimuli when they are semantically congruent (35, 39). These findings suggest that the claustrum integrates only behaviorally meaningful inputs via nonlinear operations to generate unified representations. Within this framework, both the conditioned stimulus (CS) and the door-opening cue in the delayed escape task likely share the common meaning of escape. However, since the meaning of the door-opening cue must be inferred, the integration process requires additional time, which may explain the delayed emergence of the short-loop trajectory in real neural data (see Fig. 4G, left). This interpretation is consistent with Crick and Koch’s proposal that the claustrum functions as a “conductor of consciousness,” binding semantically related information streams into a coherent whole (1). Furthermore, such integrated information is likely broadcast to higher-order cognitive regions such as the PFC, ACC, and OFC, where it can be read out and interpreted (40), a possibility that resonates with theoretical accounts of consciousness mechanisms (1, 41).

Although our conclusions are mainly derived from modeling, the correspondence between model predictions and empirical observations provides a compelling rationale for future experimental validation. Multi-trial behavioral paradigms and large-scale single-neuron recordings will be essential to rigorously test these predictions. Furthermore, elucidating the nature of the preprocessing that shapes inputs to the claustrum, as well as the downstream pathways through which its integrated representations are accessed and utilized, will be critical for understanding its role in brain-wide computation.

Materials and Methods

RNN modeling

The RNN model was implemented in PsychRNN (18, 19) as a continuous-time network with a 1s integration step. Each network consisted of a single hidden layer of 100 ReLU units, of which 80% were excitatory and 20% inhibitory.

Hidden-state $\boldsymbol{h}(t)$ dynamics followed

$$\tau\, \boldsymbol{h}^{\cdot}(t) = -\boldsymbol{h}(t) + f[\,\boldsymbol{W}_{\text{rec}}\boldsymbol{h}(t) + \boldsymbol{W}_{in}x(t) + \boldsymbol{b}_{\text{rec}}] + \boldsymbol{\eta}(t)$$

$$f(x) = \max(x, 0)$$

$$\boldsymbol{y}(t) = W_{\text{out}}\boldsymbol{h}(t) + b_{\text{out}}$$

An internal time constant of τ = 1000ms was adopted, in accordance with electrophysiological measurements (Fig. S4), and ReLU activation f(x)=max(0,x) was used throughout the network. Here, W_{rec}, W_{in}, and W_{out} denote the recurrent, input, and output weight matrices, respectively, while b_{rec} and b_{out} are constant biases applied to the recurrent and output units. Gaussian noise $\boldsymbol{\eta}(t)$ with variance $\sigma^2 = 0.05^2$ was injected into the recurrent layer. Simulations were conducted using a continuous-time recurrent neural network with a discretized time step of 1000ms.
We trained the RNN in a supervised manner using the behavioral data from rat experiments (42). We provided the RNN with sequential inputs that matched the experimental timeline: a 30-second baseline period with no input, followed by a 20-second CS input period, then a 5-second interval with no input, and finally a sustained door-open signal until the end of the trial. In the control group (neutral group), no input was provided during the period corresponding to CS presentation, and trials from the CS and neutral groups were presented in a randomized order. The network's output was treated as the probability of crossing; when this probability exceeded 0.5, the model was considered to have initiated a cross, allowing us to compute latency for each trial.
The network received four input channels, denoted as $\mathbf{x}(t) = [x_{\text{CS}},\ x_{\text{door}},\ x_{\text{inh}},\ x_{\text{ctx}}]^{\text{T}}$ The CS channel ($x_{\text{CS}}(t)$) was set to 2 during the presentation of the CS, specifically from t = 70 s to t = 90 s (20 s duration), and 0 otherwise. The door signal ($x_{\text{door}}(t)$) was set to 1 starting at t = 90 + d s, where d was randomly selected from [0, 1, 2.5, 5] seconds depending on curriculum learning, and remained 0 beforehand. The inhibition signal ($x_{\text{inh}}(t)$) was set to -1 for the 5 seconds immediately following the CS offset (t = 90-95 s) on inhibition trials. Finally, the context input ($x_{\text{ctx}}(t)$) was maintained at 1 throughout the entire trial.

Input weights were first drawn uniformly from the interval [0, 0.15], after which all rows corresponding to inhibitory units were zeroed out to enforce Dale's principle.

A cross was defined as the first time point t^* at which the network output $y(t^*) \geq 0.5$. The desired output trajectory $\tilde{y}(t)$ was specified as a step function transitioning at the target escape

$$\tilde{y}(t) = \begin{cases} 0, & t < t_{tar}, \\ 1, & t \geq t_{tar}. \end{cases}$$

time t_{tar}, which combines the door-opening time and the experimentally measured behavioral latency measured in rats:

$$t_{\text{tar}} = \begin{cases} t_{\text{door}} + \ell_{\text{CS}} = 90 + d + 48.7 \text{ s}, & \text{CS present}, \\ t_{\text{door}} + \ell_{\text{noCS}} = 90 + d + 111.3 \text{ s}, & \text{CS absent}, \end{cases}$$

Using fixed empirical latencies anchored the network to the average rat behavior, such that remaining variability in output timing arose solely from the network's internal dynamics.
The per-trial loss function minimized during training was:

$$\mathcal{L} = -\sum_t [\tilde{y} \ln y + (1 - \tilde{y}) \ln(1 - y)] + \lambda(\|\mathbf{W}_{\text{in}}\| + \|\mathbf{W}_{rec}\|) + \lambda_{FR} \left\langle \|\text{h(t)}\|_2^2 \right\rangle_t$$

Where regularization coefficients were set to $\lambda = 0.01$ and $\lambda_{FR} = 0.95$.
The network was trained using TensorFlow 2.1.0 and the Adam optimizer (learning rate $\alpha = 5 \times 10^{-4}$) for 1.2 million iterations with a batch size of 256. Each training batch comprised trials assigned randomly to CS-present or CS-absent conditions with equal probability (50% each). The delay d between CS offset and door-opening was incrementally increased (0,1, 2.5, and 5 s) once the batch-wise accuracy exceeded thresholds of 0.5, 0.5, 0.5 and 1, respectively. All simulations were executed on a single RTX-2080 Ti GPU using XLA just-in-time compilation. After training was completed, the network was saved for downstream testing under various experiment conditions, including inhibition, extended (180-s) interval, CS-only, no-CS/no-Door-opening, and stimulation of the slice experiments. The resulting cross latencies t^*, neuronal activity $\mathbf{h}(t)$, and weight matrices were exported to MATLAB for further analyses such as clustering, principal component analysis (PCA), and Partial Information Decomposition (PID).

RNN Population Analysis

The recurrent neural network (RNN) simulation was processed with the same population-level pipeline used for the experimental data (42).The first 10 s of each simulation were discarded to eliminate variability caused by initial states. Activity from the next 30 s served as the baseline for Z-score normalization.
For every CS trial (n = 147), Z-scores were averaged neuron-wise and then summarized across five task epochs—cue, interval, early-open, late-open, and the 5 s after the cross. These epoch-wise vectors were embedded into a three-dimensional t-SNE space (perplexity = 24, exaggeration = 48, RNG = mt19937ar). The optimal number of clusters was determined with the gap statistic and applied in a k-means clustering step. Through this procedure, we obtained three clusters consisting of 24, 38, and 38 units (with 24 units in the claustrum-like cluster). Here, the term claustrum-like refers to the fact that the mean response of the units remained persistent not only during the CS presentation but also throughout the subsequent period until crossing. The main difference between the empirical experiment and the RNN results is that, in the actual experiment, each rat was tested with only one latency value, so correlations had to be calculated across animals, whereas in the simulation, multiple trials were generated from a single RNN, allowing correlations to be calculated across trials. Specifically, for each trial, the

mean z-scored activity across the 24 units during the cue-to-cross window was calculated, and this value was paired with the crossing latency from that trial., and these z-scores were paired with the crossing latency from that trial. With 100 trials, this yielded 100 data points. The variability of model latency reflects both the injected noise and differences in initial states. Notably, only a subset of the supervised RNNs we trained produced a claustrum-like cluster; among 100 independent runs, only 5 met this criterion, and all results reported here are based on one representative RNN network from those 5 cases.
As shown in Fig S2, the output contribution of each cluster for each time points was computed as the product of the hidden-state vector $\mathbf{h}(t)$ and the output-weight matrix W_{out}.

Slice electrophysiology
ChrimsonR virus(pAAV-CamKIIa-ChrimsonR-mScarlet-KV2.1, titer ≥ 1×10^{13} virus molecules/mL, addgene) was injected unilaterally 200nL and GCaMP6f virus (pENN.AAV.CamKII.GCaMP6f.WPRE.SV40, titer ≥ 1×10^{13} virus molecules/mL, addgene) was injected bilaterally 400nL into the rsCla of 3–4 weeks old Long-Evans rats. After 4 weeks from injection, the rats were anesthetized with isoflurane and then decapitated. The isolated whole brains were placed in the slicing chamber filled with a warm (34~36 °C) artificial cerebrospinal fluid (aCSF) solution containing 120 mM NaCl, 3.5 mM KCl, 1.25 mM NaH_2PO_4, 26 mM $NaHCO_3$, 1.3 mM $MgCl_2$, 2 mM $CaCl_2$, and 11 mM D-(+)-glucose, and continuously bubbled at room temperature with 95% O_2/5% CO_2. Acute sagittal or horizontal slices containing the rsCla (300 µm thickness) were prepared using a vibratome (VT1200, Leica). Slices were then stored under submerged conditions at 34 °C for 1.5 hour before recordings. For recording, slices were transferred to the recording chamber in which aCSF was continuously perfused. The perfused aCSF was continuously aerated with 95% O_2/5% CO_2 and maintained at 32 °C.
Whole-cell patch clamp recordings were performed using micropipettes (4~6 MΩ) which were pulled from borosilicate glass capillaries (1.2 mm OD, 0.94 mm ID, Warner Instruments; Massachusette, USA) with a micropipette puller (Pc-10, Narishige). The pipettes were filled with the following solution: 120 mM potassium D-gluconate, 0.2 mM EGTA, 10 mM HEPES, 5 mM NaCl, 2 mM Mg-ATP, 0.3 mM Na-GTP, and 1 mM $MgCl_2$, with the pH adjusted to 7.2 with KOH and osmolality adjusted to approximately 297 mmol/kg with sucrose. Recordings were made under infrared differential interference contrast (IR-DIC)–enhanced visual guidance from neurons located in the rsCla slices. Neurons were first current-clamped neurons and then used with membrane potentials lower than -50 mV. The neurons were voltage-clamped at -70 mV, and solutions were delivered to slices via superfusion driven by gravity at a flow rate of 1.3 ml/min. The pipette series resistance was monitored throughout the experiments, and if it changed by >20%, the data were discarded. To avoid direct light stimulation, whole cell recordings were conducted using neurons in which the expression of ChrimsonR was absent. Wide-field photostimulation was executed via a microscope's objective lens (40×/0.80NA) using a power of 6.4 mW (measured at the lens level) and a wavelength of 617 nm. This stimulation was achieved through a digital mirror device (Polygon 400, mightex) for a duration of 2 ms. To stimulate each of the divided parts of the entire optical field, the same digital mirror device (DMD) was used. The entire optical field was partitioned into a 30 by 30 grid, yielding a total of 900 equivalent sections. Each section received 2 ms of stimulation, and the heatmap was generated using the Matlab interpolation algorithm with the EPSC data. The resultant EPSC amplitudes were used to generate an EPSC heatmap encompassing the entire field. Whole-cell currents were filtered at 2 kHz, digitized at up to 10 kHz, and stored on a microcomputer (Clampex 10.7 software, Molecular Devices). We used 10 µM of NBQX (Tocris), and 50 µM of D-AP5 (Tocris). One or two neurons were recorded per animal (a single neuron per slice). All recordings were completed within 5 hours after slice preparation.
In brain slices where GCaMP6f was expressed, electrical stimulation was applied using a Concentric Bipolar Electrode (FHC) via an isolator (WPI, A360). Stimulation was delivered at

150- 200 μA, 20 Hz, 1sec duration. The puffer pipette was positioned 75 μm below the slice surface and 50 μm away from the electrode. At the 10-second mark post electrical stimulation, pressure injection was carried out under a pressure of 20 psi. We used 100 μM of NBQX (Sigma) and 500 μM of D-AP5 (Tocris) for this experiment. The analysis of fluorescence changes in the recorded images follows these steps: Initially, cell boundaries were manually marked for the 35 recorded GCaMP6f slice samples. Subsequently, a U-Net deep learning model was trained on these images to develop a cell segmentation model. Using this model, the attached cells were segmented and separated utilizing the watershed algorithm. Finally, all cells were manually reviewed once more through video inspection.

RNN simulation adjusted for slice-stimulation conditions

To mimic the effect of electrical stimulation and applied NBQX+D-AP5 in the acute slice, we constructed a reduced circuit composed exclusively of Cluster 1 (C1) units. All recurrent weights outside C1 were zeroed, leaving only C1 → C1 connections in $\boldsymbol{W}_{rec}$. Because inhibitory neurons may lose efficacy in acute slices, every inhibitory weight in $\boldsymbol{W}_{rec}$ was scaled to 40 % of its original value.

Stimulation was modeled by injecting an external drive $x_{\text{stim}}(t) = 2$ for 1 s. The corresponding input weights were drawn uniformly from 0 to 0.15 for excitatory neurons only. Input weights to the inhibitory neurons were set to zero.

In our model, we simulated the effects of applying NBQX + D-AP5 by introducing recurrent inhibition, which began 10 seconds after the initial stimulation. The inhibition was targeted, affecting only 10% of the neurons in the network. For these selected neurons, the strength of the excitatory inputs they received from other neurons ($\boldsymbol{W}_{rec}$) was reduced by 60%. This simulation was designed to replicate key findings from our slice experiments (Fig. S4): First, the effect of NBQX + D-AP5 is localized and does not spread widely; and second, drug application resulted in an approximately 60% reduction in the amplitude of EPSCs.

Visualization of PCA trajectories

To visualize neural dynamics, trial-wise Z-scored firing rates from a specific neuronal cluster were extracted for each simulation group (CS + Open, Open-only, and CS-only). Trials were concatenated across groups and reshaped into a 2D matrix of size (trial x time) × neuron for principal component analysis (PCA). PCA was performed on the combined matrix, and the first three PCs were used for visualization.

Since each trial had a different latency to the cross event, trajectories were time-normalized. Each trial was divided into two segments: a fixed-length pre-open segment (Segment A, from the start to the open event; 55s) and a variable-length post-open segment (Segment B, from the open event to the cross event). For trials without a defined Cross event (e.g., CS-only group), the average cross latency from the CS + Open group was used. Segment B was interpolated to a fixed length of 30 bins using shape-preserving piecewise cubic interpolation. The two segments were concatenated and smoothed using a moving average. The smoothing window size was determined heuristically to attenuate approximately 25% of the energy of the PCA scores for each trial.

Time-normalized trajectories were averaged within each group to produce group-level mean trajectories and corresponding standard errors of the mean (SEM). These trajectories, uniform in the number of time bins, were visualized either separately for each group or overlaid for direct comparison.

For both the inhibition simulation and the 180s interval simulation, corresponding trial data were combined with CS + Open trials. These combined datasets were reshaped and processed using the same PCA and time-normalization procedures described above.

Prediction of CS+Open Trajectories from Component Conditions

To evaluate how well CS+Open trajectories could be reconstructed from their component conditions, we extracted time-normalized PCA trajectories from the CS+Open, Open-only, and CS-only groups. For the CS+Open condition, trajectories were aligned to the door-opening event, which occurred 5 seconds after CS offset; for the CS-only condition, trajectories were taken beginning at the corresponding time point—5 s after CS offset. All trajectories were preprocessed using the same procedures described in the trajectory visualization section. To standardize across conditions, trajectories were trimmed to include only the period from the open event to the cross event (30 bins), which served as the input for modeling.

We trained two models to predict individual CS+Open trial trajectories from the group-averaged Open-only and CS-only trajectories:

Linear regression model: At each time bin, the trial trajectory was regressed onto the corresponding bins of the mean Open-only and CS-only trajectories, yielding a bin-wise linear estimate.

Multilayer perceptron (MLP): To test for nonlinear integration, we implemented a feedforward neural network with two hidden layers ([6, 3] units) and L2 regularization ($\lambda \in [0, 0.25, 0.5, 0.75, 1]$), with the optimal λ chosen to minimize the RSS for each condition. For each condition (CS+Open, inhibition, 180 s interval), the λ yielding the lowest residual sum of squares (RSS) was selected. The MLP received concatenated mean trajectories from the Open-only and CS-only conditions as inputs and was trained to predict the trial trajectory of the CS+Open condition. Training was performed independently for each trial using a 70/15/15% split of bins for training, validation, and test sets, with min–max normalization applied to the inputs.

Prediction performance was quantified as the RSS at each time bin. For the MLP, training was repeated 10 times per trial, and RSS values were averaged across repetitions to yield a trial-wise mean RSS time series. To assess the benefit of nonlinear integration, improvement by the MLP over the linear model was calculated as the bin-wise difference between their RSS values, normalized by the linear model RSS. These normalized scores were averaged across bins to obtain a single summary value for each trial.

Decoding analysis (for Fig 3H-I)

CS Decoding:

We first gathered the time-normalized Z-scores of all cluster neurons from two trial types: CS + Open (cue present) and Open-only (cue absent). Time normalization from open to crossing was identical to the procedure described above.

For each time bin we formed a trial-by-neuron matrix and trained a linear discriminant model with fitcdiscr('linear') in MATLAB R2022b. Trials in which a cue was delivered were labelled 1 from cue onset onward, whereas cue-absent trials were labelled 0 throughout. Model performance was assessed by five-fold cross-validation, and accuracy was reported as 1 – kfoldLoss. Repeating this procedure over all bins produced a temporal profile of CS-decoding accuracy.

Door Open Decoding:

A Door-open decoder was generated in the same way, but the training data comprised CS + Open versus CS-only trials. Here, bins occurring after the outlet opened were labelled 1, and all earlier bins (as well as every bin in CS-only trials) were labelled 0. Five-fold cross-validated

linear discriminant analysis again yielded a time-resolved accuracy curve that reflected when RNN activity best predicted door opening.

Partial Information Decomposition (PID) Analysis

To quantify how CS and door-open information were distributed across RNN neurons, we applied Partial Information Decomposition (PID) analysis to decoding accuracies as X variables and neural activity as a Y in cluster 1–3. For each cluster, z-scored activity was extracted and time-normalized using a fixed window of 85 bins, which included a 55s pre-open segment and a 30-bin post-open segment interpolated to a fixed length. Trials from four conditions were included: CS + Open, Open-only, CS-only, and None.

For each neuron and each time bin, we computed PID terms (redundancy, CS-unique information, Open-unique information, and synergy) using MATLAB code(the Neuroscience Information Theory Toolbox (28)).

At each time bin, we used the neuron's z-scored activity rounded to one decimal place across all trials as the Y (time × trial), and the CS and door-open decoding scores as X_1 and X_2, respectively (time × trial) (see Fig. 3L).

Specifically, the construction of X_1 (cue-related events) was as follows.

For the CS+Open and CS-only conditions, decoding accuracy were computed as described above. Trials in which a cue was delivered were labeled 1 from cue onset onward, whereas cue-absent trials were labeled 0 throughout. A decoder was trained to classify the z-scored firing rates in each time bin of CS+Open and Open-only trials according to these labels. To express decoding accuracy as a score ranging from 0 to 1, we subtracted 0.5 from the decoding accuracy, multiplied the result by 2, and set any negative values to 0. The resulting scores were smoothed with a Gaussian kernel (window size = 15). To compute the probability of coincident events, smoothed scores were rounded to the first decimal place. For Open-only and None conditions, all time bins were assigned a value of 0.

The construction of X_2 (door-open-related events) was as follows.

For CS+Open trials, as described above, time bins after the outlet opening were labeled 1, and all earlier bins as well as bins in CS-only trials were labeled 0. A decoder was trained to classify each time bin's z-scored activity in CS+Open and CS-only trials based on these labels. Decoding scores were then computed using the same method as described for X_1.

For Open-only trials, time bins following outlet opening were labeled 1, and all earlier bins as well as bins in None trials were labeled 0. A decoder was trained to classify z-score in Open-only and None trials, and decoding scores were computed identically.

For CS-only and None conditions, all bins were assigned a value of 0.

For each discrete state j of Y and each combination of states k from a source A (either X_1 alone, X_2 alone, or their joint distribution [X_1, X_2]), we computed the corresponding conditional probabilities required for PID analysis.

$$P(A = k|Y = j) = P(Y = j, A = k)/P(Y = j),\ P(Y = j|A = k) = P(Y = j, A = k)/P(A = k)$$

The information that source A conveys about Y when Y is in state j is then

$$I(Y = j; A) = \sum_k P(Y = j, A = k) \log_2 \frac{P(Y = j|A = k)}{P(Y = j)}$$

We defined its redundancy by, for each j, taking the minimum of $I(Y = j; A)$ over the X1 and X2, and then computing the dot product of that vector of minima with the state-probabilities $P(Y = j)$. X_1-unique information is the mutual information between Y and X1 minus the redundancy.

X2-unique information is the mutual information between Y and X2 minus the redundancy. Synergy is obtained by taking the joint mutual information, subtracting both unique-information terms, and then adding back the redundancy.
PID terms were averaged across neurons within each cluster. For visualization, time-resolved PID terms were shown as mean ± SEM across neurons.

Cross-temporal Decoding Analysis
Cross-temporal decoding of CS+Open information was performed as follows: For each time bin, a decoder was trained to classify z-scored neural activity from CS+Open trials (all time bins labelled as 1) versus CS-only and Open-only trials (all time bins labelled as 0). The trained decoder was then tested across all time bins to predict trial type. Decoding accuracy was computed as (correct trials / total trials × 100) and visualized as a two-dimensional heatmap indexed by training and testing time bins.

Fixed Point Analysis
We implemented the FixedPointFinder toolbox to identify fixed points in the trained RNN model(30).
Fixed point analysis was performed separately for the following input epochs, defined by specific timestep intervals and corresponding input vectors:

Baseline (40–69 s): [0, 0, 0, 1, 0, 0, 0, 0]
CS (70–89 s): [2, 0, 0, 1, 0, 0, 0, 0]
Interval (90–94 s): [0, 0, 0, 1, 0, 0, 0, 0]
Open (95–239 s): [0, 1, 0, 1, 0, 0, 0, 0]

The recurrent and input weights, bias vector, and the alpha parameter ($\alpha = \Delta t / \tau$) were used to represent the trained dynamics throughout the epochs.
For each epoch, the neural state trajectories x and corresponding input vector u were constructed. Initial conditions for fixed point searches were formed by combining all neural states within the epoch with a randomly sampled subset of global network states (n_trial= (number of epoch states)+ 512 randomly sampled states; x: n_trial × n_neurons matrix,u:n_trial x 8 matrix). The input vector u was tiled across trials to match the dimensionality.
Fixed points were then identified jointly via optimization To find fixed point candidates, x and their one-step forward states F were computed, and the difference between x and F was minimized in an iterative gradient descent loop, where x was updated using the Adam optimizer. If change in loss became smaller than a convergence tolerance($1x10^{-4}$) times the learning rate or if the loss is smaller than tolerance, then the iteration was stopped. The maximum number of iterations was set to 10,000. We discarded the candidate as an outlier if the distance between a fixed point candidate x* and the centroid of the initial states was larger than 10.0 times the average distance between the centroid and each initial state. Redundant candidates were further merged if the Euclidean distance between any pair of x* was less than $1x10^{-2}$.The remaining fixed points x* and their Jacobian eigenvalues were collected. Fixed points were classified based on their dynamical stability: points were labeled as stable (attractors) if all eigenvalues had absolute magnitudes less than 1.0, and as saddle points otherwise.

Trajectory Analysis of Biological Data Using GPFA
For the biological data, z-scored activity pooled from in vivo recordings of nonexploratory cluster 1 neurons was reduced to a three-dimensional space using Gaussian-Process Factor Analysis (GPFA), with Gaussian smoothing kernel size of 20 bins (implemented using MATLAB code from Lakshmanan et al. 2015 (43)). Each pseudopopulation trajectory from CS or Neutral rats was subsequently smoothed with a Gaussian kernel (10-bin window) and visualized. We

employed GPFA because it estimates latent states shared across neurons while explicitly modeling the noise characteristics of individual neurons, making it well-suited for single-trial trajectory visualization. Unlike PCA, which identifies orthogonal axes that explain maximal variance without incorporating temporal structure, GPFA accounts for temporal dynamics, providing a key advantage in analyzing population activity over time.
Validation was performed by independently applying GPFA to two neuronal subsets derived from in vivo CS group recordings. Neurons were ranked in descending order based on their mean z-scored firing rates during the baseline to 5-second post-crossing window. The ranked list was then divided into two subsets comprising odd- and even-indexed neurons, respectively. GPFA was applied to each subset using identical parameter settings as described above. For visualization, neural trajectories were smoothed with a 10-bin Gaussian window. The trajectory of one subset (subset 1) was chosen as the reference, and the others were aligned to it using the Procrustes method (MATLAB function). Specifically, anchor points from each trajectory were mapped to the corresponding anchors in the reference trajectory by minimizing the sum of squared errors under a linear transformation. The procedure included centering and normalization, followed by singular value decomposition (SVD) to estimate the optimal rotation, and finally applying translation and scaling. Anchor points consisted of the trial start, CS onset, CS offset, door opening, and crossing time.

Acknowledgements

This work was supported by the NRF (https://www.nrf.re.kr) of Korea grant funded by the Korean Government Ministry of Education (https://english.moe.go.kr), Science and Technology (NRF-2021R1A2B5B03002345), awarded to SC.

Reference

1. F. C. Crick, C. Koch, What is the function of the claustrum? *Philos Trans R Soc Lond B Biol Sci* **360**, 1271-1279 (2005).
2. A. M. Shelton *et al.*, Single neurons and networks in the mouse claustrum integrate input from widespread cortical sources. *Elife* **13** (2025).
3. M. G. White *et al.*, Anterior Cingulate Cortex Input to the Claustrum Is Required for Top-Down Action Control. *Cell Rep* **22**, 84-95 (2018).
4. G. Atlan *et al.*, Claustrum neurons projecting to the anterior cingulate restrict engagement during sleep and behavior. *Nat Commun* **15**, 5415 (2024).
5. C. A. Faig *et al.*, Claustrum projections to the anterior cingulate modulate nociceptive and pain-associated behavior. *Curr Biol* **34**, 1987-1995 e1984 (2024).
6. M. Niu *et al.*, Claustrum mediates bidirectional and reversible control of stress-induced anxiety responses. *Sci Adv* **8**, eabi6375 (2022).

7. M. Chevee, E. A. Finkel, S. J. Kim, D. H. O'Connor, S. P. Brown, Neural activity in the mouse claustrum in a cross-modal sensory selection task. *Neuron* **110**, 486-501 e487 (2022).
8. K. Narikiyo *et al.*, The claustrum coordinates cortical slow-wave activity. *Nat Neurosci* **23**, 741-753 (2020).
9. M. H. Chong, R. Gămănuţ, Anatomical and physiological characteristics of claustrum neurons in primates and rodents. *Frontiers in Mammal Science* **3**, 1309665 (2024).
10. Q. Wang *et al.*, Organization of the connections between claustrum and cortex in the mouse. *J Comp Neurol* **525**, 1317-1346 (2017).
11. Y. Han *et al.*, Delayed escape behavior requires claustral activity. *Cell Rep* **43**, 113748 (2024).
12. C. M. Dillingham *et al.*, The Anatomical Boundary of the Rat Claustrum. *Front Neuroanat* **13**, 53 (2019).
13. X. Zhang *et al.*, Susceptibility to kindling and neuronal connections of the anterior claustrum. *J Neurosci* **21**, 3674-3687 (2001).
14. M. M. Jankowski, S. M. O'Mara, Dynamics of place, boundary and object encoding in rat anterior claustrum. *Front Behav Neurosci* **9**, 250 (2015).
15. M. M. Jankowski, M. N. Islam, S. M. O'Mara, Dynamics of spontaneous local field potentials in the anterior claustrum of freely moving rats. *Brain Res* **1677**, 101-117 (2017).
16. K. Grasby, A. Talk, The anterior claustrum and spatial reversal learning in rats. *Brain Res* **1499**, 43-52 (2013).
17. S. Park, K. Sohn, D. Yoon, J. Lee, S. Choi, Single-unit activity in the anterior claustrum during memory retrieval after trace fear conditioning. *PLoS One* **20**, e0318307 (2025).
18. D. B. Ehrlich, J. T. Stone, D. Brandfonbrener, A. Atanasov, J. D. Murray, PsychRNN: An Accessible and Flexible Python Package for Training Recurrent Neural Network Models on Cognitive Tasks. *eNeuro* **8** (2021).
19. D. B. Ehrlich, J. D. Murray, Geometry of neural computation unifies working memory and planning. *Proc Natl Acad Sci U S A* **119**, e2115610119 (2022).
20. R. Kim, T. J. Sejnowski, Strong inhibitory signaling underlies stable temporal dynamics and working memory in spiking neural networks. *Nat Neurosci* **24**, 129-139 (2021).
21. J. P. Stroud, K. Watanabe, T. Suzuki, M. G. Stokes, M. Lengyel, Optimal information loading into working memory explains dynamic coding in the prefrontal cortex. *Proceedings of the National Academy of Sciences* **120**, e2307991120 (2023).
22. C. D. Brody, R. Romo, A. Kepecs, Basic mechanisms for graded persistent activity: discrete attractors, continuous attractors, and dynamic representations. *Current opinion in neurobiology* **13**, 204-211 (2003).
23. W. Chaisangmongkon, S. K. Swaminathan, D. J. Freedman, X.-J. Wang, Computing by robust transience: how the fronto-parietal network performs sequential, category-based decisions. *Neuron* **93**, 1504-1517. e1504 (2017).
24. K. Mohan, O. Zhu, D. J. Freedman, Interaction between neuronal encoding and population dynamics during categorization task switching in parietal cortex. *Neuron* **109**, 700-712. e704 (2021).
25. J. P. Stroud, J. Duncan, M. Lengyel, The computational foundations of dynamic coding in working memory. *Trends Cogn Sci* **28**, 614-627 (2024).
26. J. Kim, C. J. Matney, R. H. Roth, S. P. Brown, Synaptic Organization of the Neuronal Circuits of the Claustrum. *J Neurosci* **36**, 773-784 (2016).

27. J. Voigts *et al.*, Spatial reasoning via recurrent neural dynamics in mouse retrosplenial cortex. *Nat Neurosci* **28**, 1293-1299 (2025).
28. N. M. Timme, C. Lapish, A Tutorial for Information Theory in Neuroscience. *eNeuro* **5** (2018).
29. N. Timme, W. Alford, B. Flecker, J. M. Beggs, Synergy, redundancy, and multivariate information measures: an experimentalist's perspective. *J Comput Neurosci* **36**, 119-140 (2014).
30. M. D. Golub, D. Sussillo, FixedPointFinder: A Tensorflow toolbox for identifying and characterizing fixed points in recurrent neural networks. *Journal of open source software* **3**, 1003 (2018).
31. B. M. Yu *et al.*, Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. *J Neurophysiol* **102**, 614-635 (2009).
32. K. C. Lakshmanan, P. T. Sadtler, E. C. Tyler-Kabara, A. P. Batista, B. M. Yu, Extracting Low-Dimensional Latent Structure from Time Series in the Presence of Delays. *Neural Comput* **27**, 1825-1856 (2015).
33. C. D. Brody, A. Hernández, A. Zainos, R. Romo, Timing and neural encoding of somatosensory parametric working memory in macaque prefrontal cortex. *Cerebral cortex* **13**, 1196-1207 (2003).
34. E. M. Meyers, Dynamic population coding and its relationship to working memory. *J Neurophysiol* **120**, 2260-2268 (2018).
35. H. R. Naghavi, J. Eriksson, A. Larsson, L. Nyberg, The claustrum/insula region integrates conceptually related sounds and pictures. *Neurosci Lett* **422**, 77-80 (2007).
36. M. M. Reus-Garcia *et al.*, The Claustrum is Involved in Cognitive Processes Related to the Classical Conditioning of Eyelid Responses in Behaving Rabbits. *Cereb Cortex* **31**, 281-300 (2021).
37. D. R. Ollerenshaw *et al.*, Anterior claustrum cells are responsive during behavior but not passive sensory stimulation. *BioRxiv*, 2021.2003. 2023.436687 (2021).
38. B. W. Stewart *et al.*, Pathological claustrum activity drives aberrant cognitive network processing in human chronic pain. *Curr Biol* **34**, 1953-1966 e1956 (2024).
39. M. Froesel *et al.*, Macaque claustrum, pulvinar and putative dorsolateral amygdala support the cross-modal association of social audio-visual stimuli based on meaning. *Eur J Neurosci* **59**, 3203-3223 (2024).
40. C. Seguin, O. Sporns, A. Zalesky, Brain network communication: concepts, models and applications. *Nat Rev Neurosci* **24**, 557-574 (2023).
41. S. Dehaene, M. Kerszberg, J. P. Changeux, A neuronal model of a global workspace in effortful cognitive tasks. *Proc Natl Acad Sci U S A* **95**, 14529-14534 (1998).
42. Y. Han *et al.*, Delayed escape behavior requires claustral activity. *Cell Reports* **43** (2024).
43. K. C. Lakshmanan, P. T. Sadtler, E. C. Tyler-Kabara, A. P. Batista, B. M. Yu, Extracting low-dimensional latent structure from time series in the presence of delays. *Neural computation* **27**, 1825-1856 (2015).

Fig. 1

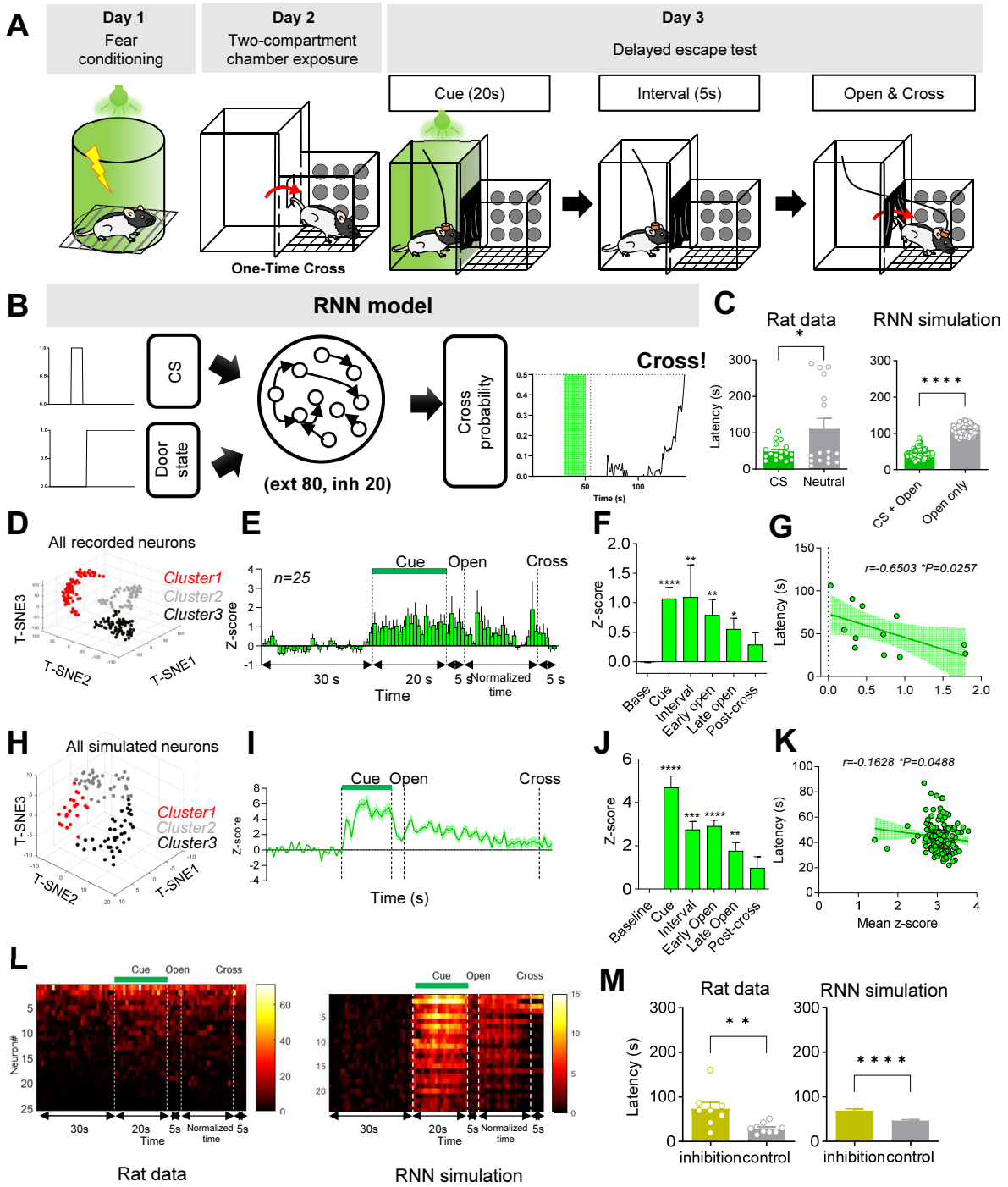
A
Day 1
Fear conditioning
Day 2
Two-compartment chamber exposure
Day 3
Delayed escape test
Cue (20s)
Interval (5s)
Open & Cross
One-Time Cross
B
RNN model
CS
Door state
(ext 80, inh 20)
Cross probability
Cross!
Time (s)
C
Rat data
RNN simulation
Latency (s)
CS
Neutral
CS + Open
Open only
D
All recorded neurons
Cluster1
Cluster2
Cluster3
T-SNE3
T-SNE2
T-SNE1
E
n=25
Cue
Open
Cross
Z-score
30 s
20 s
5 s
Normalized time
5 s
Time
F
Z-score
Base
Cue
Interval
Early open
Late open
Post-cross
G
r=-0.6503 *P=0.0257
Latency (s)
H
All simulated neurons
Cluster1
Cluster2
Cluster3
T-SNE3
T-SNE2
T-SNE1
I
Cue
Open
Cross
Z-score
Time (s)
J
Z-score
Baseline
Cue
Interval
Early Open
Late Open
Post-cross
K
r=-0.1628 *P=0.0488
Latency (s)
Mean z-score
L
Cue
Open
Cross
Neuron#
30s
20s
5s
Normalized time
5s
Time
Rat data
RNN simulation
M
Rat data
RNN simulation
Latency (s)
inhibition
control

Figure 1 | RNN simulation capable of performing the Delayed Escape Test

(A) Schematic diagram of the Delayed Escape Task. On Day 1, animals were conditioned with a light cue paired with an electric footshock. On Day 2, they were adapted to the test context with only a single crossing permitted. On Day 3, animals were placed in one compartment of the test context and presented with a 20-s CS, followed by a 5-s delay, after which the outlet door was opened to allow escape. The test was conducted only once.

(B) Schematic diagram of the task structure that the RNN model was designed to simulate.

(C) Graphs of crossing latency, measured as the interval from door opening to crossing in rats and as derived from RNN simulations. (*p = 0.0335, unpaired *t*-test; RNN, ****p < 0.0001, Mann–Whitney).

(D) Clustering of all recorded claustral neurons in rats (n = 203).

(E) Mean Z-scored firing rates of non-exploratory Cluster 1 neurons in the CS group (n = 25, rat data). Non-exploratory Cluster 1 denotes the cluster showing increased persistent activity both during CS presentation and thereafter. The CS group corresponds to the CS + door-opening condition in the RNN model.

(F) Average Z-score for each behavioral period shown in E. Bars: Friedman test, ****p < 0.0001; Dunn's multiple comparisons—Base vs Cue, ****p < 0.0001; Base vs Interval, **p = 0.0085; Base vs Early Open, **p = 0.0075; Base vs Late Open, *p = 0.0455; Base vs Post-cross, p > 0.9999.

(G) Correlation between crossing latency and Cluster 1 activity in the CS group (n = 12; Spearman r = 0.6503, *p = 0.0257).

(H) Clustering for simulated neurons (n = 100 neurons).

(I) Mean Z-scored firing rates of Cluster 1 neurons in the CS + door-opening condition (n = 24 neurons, RNN) across all simulated trials.

(J) Average Z-score for each behavioral period shown in I. Bars: Friedman ****p < 0.0001; Dunn's—Base vs Cue, ****p < 0.0001; Base vs Interval, ***p = 0.0001; Base vs Early Open, ****p < 0.0001; Base vs Late Open, **p = 0.0034; Base vs Post-cross, p = 0.7134.

(K) Correlation between crossing latency and Cluster 1 activity in the CS + door-opening condition (147 trials; Spearman r = –0.1628, *p = 0.0488).

(L) Heatmaps of single-neuron activity corresponding to panels E and I. Left: rat data (firing rate, Hz); Right: RNN data (Z-score). Neurons are ordered by overall activity. (M) Crossing latency in rats with anterior claustrum inhibition during the 5-s period between CS offset and door opening, compared with control virus–expressing animals (left, **p = 0.0079, Mann–Whitney). Corresponding results in the RNN are shown for selective inhibition of Cluster 1 neurons during the same 5-s window simulation (right, ****p < 0.0001, Mann–Whitney).

Panels A, D, E, F, and L(left) are adapted from Figs. 1A, 4B, 4D, 4E, and S4E of Han et al. (12), *Cell Reports*, © Elsevier.

Fig. 2

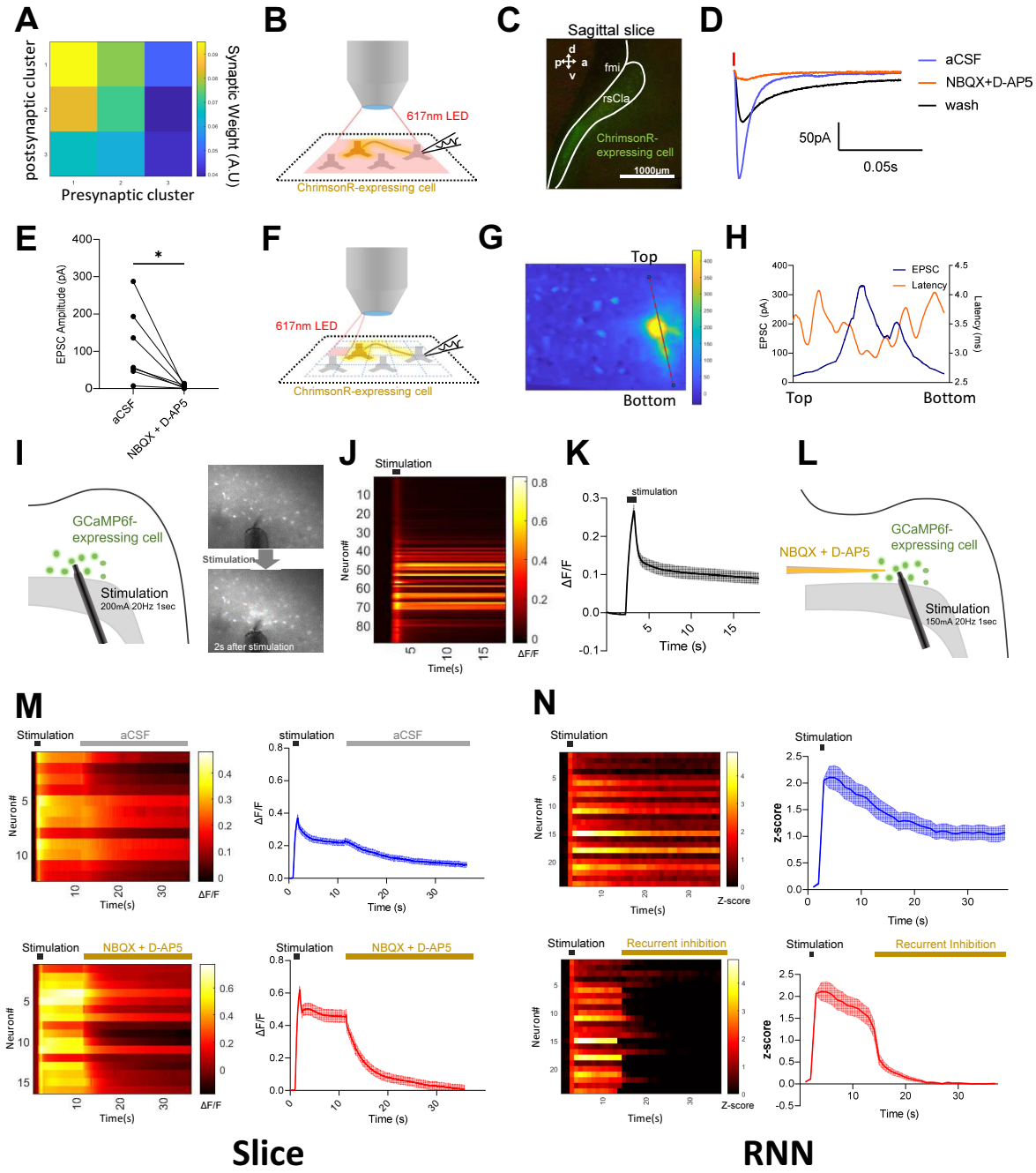
A
postsynaptic cluster
Presynaptic cluster
Synaptic Weight (A.U)
B
617nm LED
ChrimsonR-expressing cell
C
Sagittal slice
fmi
rsCla
ChrimsonR-expressing cell
1000μm
D
aCSF
NBQX+D-AP5
wash
50pA
0.05s
E
EPSC Amplitude (pA)
aCSF
NBQX + D-AP5
F
617nm LED
ChrimsonR-expressing cell
G
Top
Bottom
H
EPSC
Latency
EPSC (pA)
Latency (ms)
Top
Bottom
I
GCaMP6f-expressing cell
Stimulation
200mA 20Hz 1sec
Stimulation
2s after stimulation
J
Stimulation
Neuron#
Time(s)
ΔF/F
K
stimulation
ΔF/F
Time (s)
L
NBQX + D-AP5
GCaMP6f-expressing cell
Stimulation
150mA 20Hz 1sec
M
Stimulation
aCSF
Neuron#
Time(s)
ΔF/F
stimulation
aCSF
Time (s)
Stimulation
NBQX + D-AP5
Neuron#
Time(s)
ΔF/F
Stimulation
NBQX + D-AP5
Time (s)
N
Stimulation
Neuron#
Time(s)
Z-score
Stimulation
z-score
Time (s)
Stimulation
Recurrent inhibition
Neuron#
Time(s)
Z-score
Stimulation
Recurrent Inhibition
z-score
Time (s)
Slice
RNN

Figure 2 | Comparison of recurrent connectivity in the claustrum and in the RNN simulation
(A) Mean absolute inter-cluster synaptic weight in the RNN.
(B) Schematic representation of whole-cell recordings of ChrimsonR-non-expressing neurons during optogenetic stimulation using LEDs to stimulate ChrimsonR-expressing neurons across the entire optical field.
(C) Confocal image showing the expression pattern of the ChrimsonR virus in sagittal claustrum slices.
(D) Representative EPSCs evoked by 2ms LED pulses before and after pharmacological treatments (red bar indicates stimulation).
(E) Pooled data of EPSC amplitudes (n=7) with statistical analysis using the Wilcoxon matched-pairs signed rank test (*p=0.0156).
(F) Schematic depiction of local stimulation using a digital mirror device (DMD).
(G) Optical stimulation of each divided part produces variable amplitudes of EPSCs in whole-cell patched neurons expressing no ChrimsonR. Left: Heatmap showing EPSC amplitudes upon stimulation of the designated part of optical field. Please note that when stimulating closer to the recorded neuron, larger amplitude EPSCs were induced.
(H) The graph displays EPSC amplitudes and latencies in the region marked with dashed lines on the heatmap shown in G.
(I) Schematic representation showing an experimental configuration in which brief electrical stimulation produces a persistently enhanced activity in rsCla slices (left). Representative images of GCaMP6f fluorescence changes immediately before and 2 seconds after stimulation (right). (J) Heatmap of fluorescence changes for individual puncta in the slice shown in I.
(K) Population-averaged fluorescence trace of all puncta in J (mean ± SEM).
(L) Schematic representation showing an experimental configuration in which effects of the blockers for AMPA/NMDA receptors on the persistent response were examined.
(M) Calcium imaging before and after an aCSF puff (top panels) and an NBQX + D-AP5 puff (bottom panels): heatmaps (left) and population-averaged traces (right).
(N) RNN analogue: heatmaps (left) and averaged Z-scores (right) for Cluster 1 neurons following brief excitation (top panels) and after the addition of recurrent inhibition (bottom panels).

Fig. 3

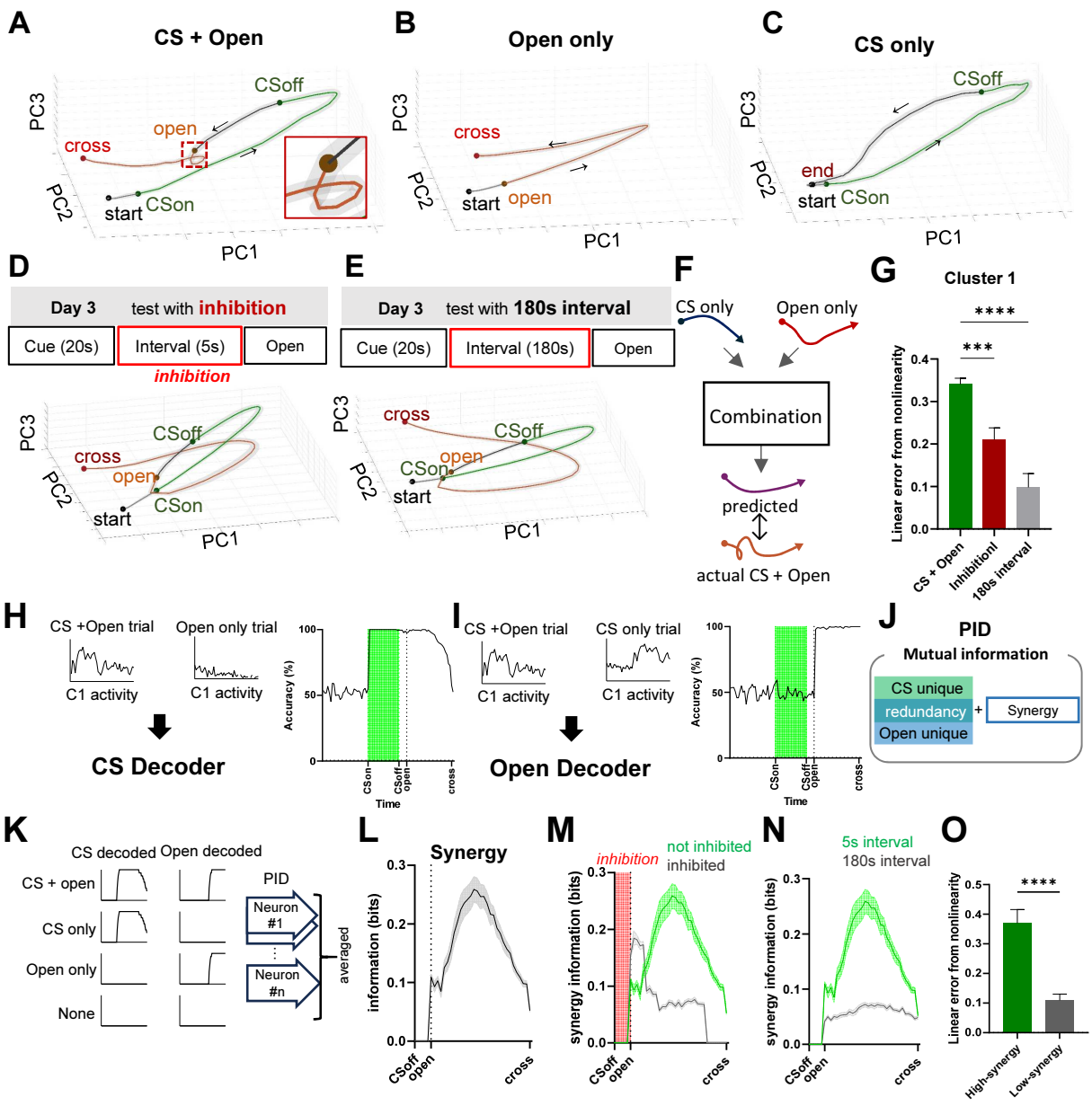
A
CS + Open
PC3
PC2
PC1
CSoff
open
cross
start
CSon
B
Open only
cross
start
open
C
CS only
CSoff
end
start
CSon
D
Day 3
test with inhibition
Cue (20s)
Interval (5s)
Open
inhibition
E
Day 3
test with 180s interval
Cue (20s)
Interval (180s)
Open
F
CS only
Open only
Combination
predicted
actual CS + Open
G
Cluster 1
****
***
Linear error from nonlinearity
CS + Open
Inhibition
180s interval
H
CS +Open trial
Open only trial
C1 activity
CS Decoder
Accuracy (%)
Time
I
CS +Open trial
CS only trial
Open Decoder
J
PID
Mutual information
CS unique
redundancy
Open unique
Synergy
K
CS decoded
Open decoded
CS + open
CS only
Open only
None
PID
Neuron #1
Neuron #n
averaged
L
Synergy
information (bits)
M
inhibition
not inhibited
inhibited
synergy information (bits)
N
5s interval
180s interval
O
Linear error from nonlinearity
High-synergy
Low-synergy

Figure 3 | PCA Trajectories, Decoder Accuracy, and Partial Information Decomposition of Cluster 1 RNN Neurons under different simulation conditions.

(A-C) Time-normalized, trial-averaged PCA trajectories for each simulation condition. All trajectories begin at the Start event (black circle), proceed through CS onset and offset (CSon & CSoff; green circle) in the CS-containing and inhibition simulations, then through the Open event (orange circle) in the Open-containing and inhibition simulations, and terminate either at the Cross event (red circle) for crossing conditions or at the End event (black circle) for the CS-only condition.

(A) PCA trajectory for the CS+door-opening condition. The mean latency from door opening to crossing was 44.16 ± 0.9583 s.

(B) PCA trajectory for the door-opening only condition.

(C) PCA trajectory for the CS only condition.

(D) Simulated inhibition applied during a 5-s interval between CSoff and door-opening. Top: schematic of the simulated task. Bottom: time-normalized, trial-averaged PCA trajectory.

(E) Simulation with a 180-s interval between CSoff and door-opening (no inhibition applied). Top: schematic of the simulated task. Bottom: time-normalized, trial-averaged PCA trajectory.

(F) Schematic of the trajectory-combination model: the predicted CS + door-opening trajectory (purple line)—obtained by model of which input are the CS only and door-opening only trajectories—is plotted against the actual CS + door-opening trajectory (orange line).

(G) Model-fit comparison for cluster 1: difference in residual sum of squares (ΔRSS) between the linear regression and MLP models, normalized to the mean RSS of the linear model. Bar colors denote condition (CS + door-opening = green, inhibition = red, 180-s interval = gray). One-way ANOVA with Holm–Sidak's multiple comparisons test: CS + door-opening vs. inhibition, **$p = 0.0063$; CS + Open vs. 180-s interval, ****$p < 0.0001$. Bars show mean ± SEM. Solid lines represent mean PCA trajectories; shaded areas denote SEM.

(H) Left: Schematic of the CS decoder. A classifier was trained to distinguish CS + door-opening trials from door-openin g only trials using cluster 1 neuron activity. Right: CS decoding accuracy (%) across time bins.

(I) Left: Schematic of the door-opening decoder. A classifier was trained to distinguish CS + door-opening trials from C S only trials using cluster 1 neuron activity. Right: Door-opening decoding accuracy (%) across time bins.

(J) Conceptual diagram of Partial Information Decomposition. Mutual information about CS and door-opening is decom posed into CS-unique information, door-opening-unique information, redundancy, and synergy.

(K) Schematic of the PID analysis. CS decoding accuracy and door-opening decoding accuracy were computed for eac h trial type (CS + door-opening, CS only, door-opening only, and None), and used as input variables for neuron-wise PI D. PID terms were then averaged across neurons.

(L) Synergy information of cluster 1 neurons.

(M) Comparison of synergy between inhibition and no-inhibition simulations.

(N) Comparison of synergy between the 180-s interval and the 5-s interval simulations. Green line: 5-s interval; green shaded box indicates CS presentation window. Gray line: 180-s interval. Solid lines represent means; shaded areas indicate SEM.

(O) Model fit comparison between cluster 1 RNN neurons with high synergy and those with low synergy. The difference in residual sum of squares (ΔRSS) between linear regression and MLP models is shown, normalized to the mean RSS of the linear model. Bar colors indicate condition (high synergy = green, low synergy = gray). (High synergy: $n = 147$ trials; Low synergy: $n = 147$.) Mann-Whitney test: high vs. low synergy, ****$p < 0.0001$. Bars show mean ± SEM.

In this figure, Open denotes door-opening.

Fig. 4

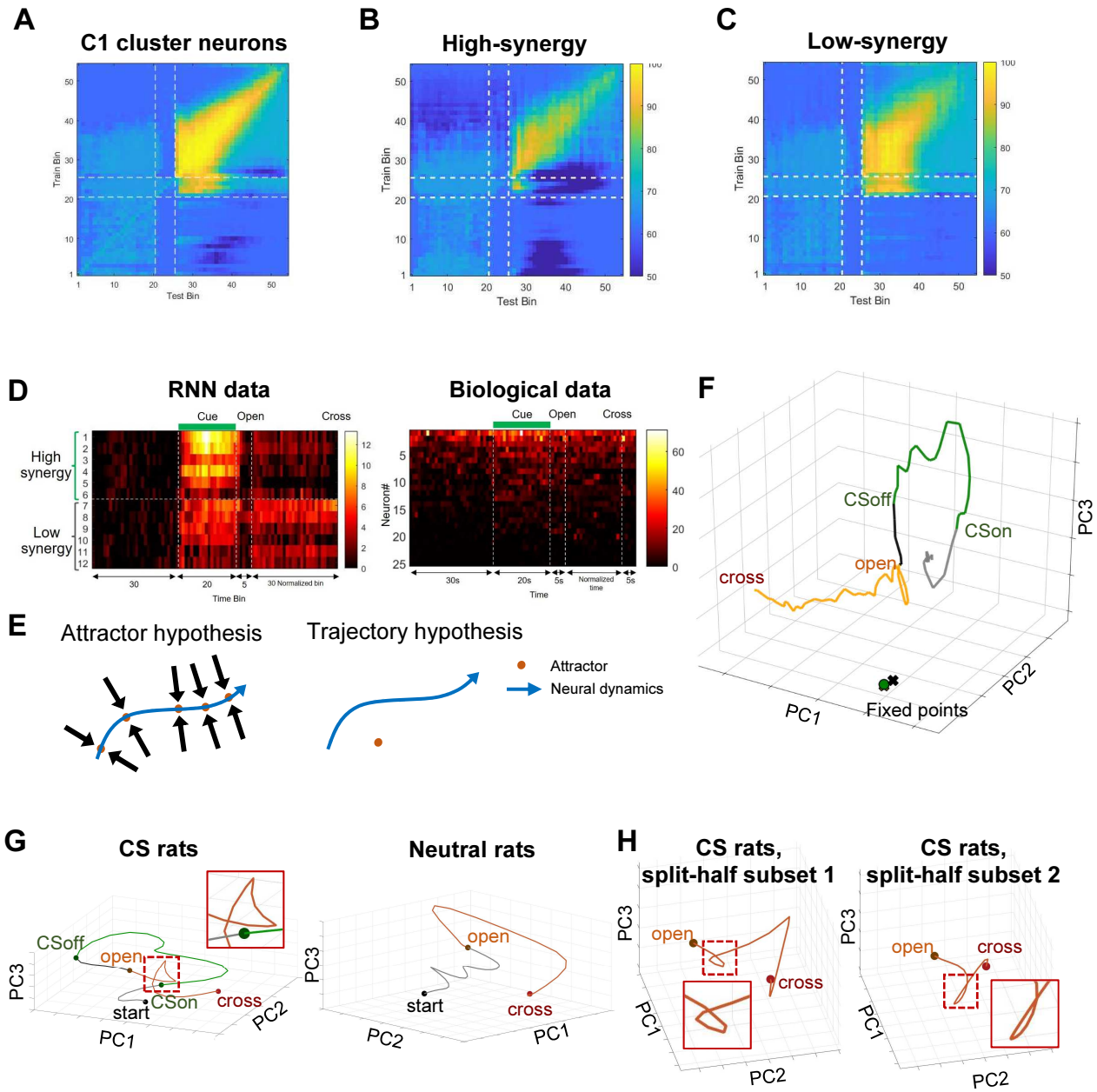
A
C1 cluster neurons
Train Bin
Test Bin
B
High-synergy
Train Bin
Test Bin
C
Low-synergy
Train Bin
Test Bin
D
RNN data
Cue
Open
Cross
High synergy
Low synergy
Time Bin
Biological data
Cue
Open
Cross
Neuron#
Time
F
CSoff
CSon
open
cross
PC3
PC2
PC1
Fixed points
E
Attractor hypothesis
Trajectory hypothesis
Attractor
Neural dynamics
G
CS rats
CSoff
open
CSon
start
cross
PC3
PC1
PC2
Neutral rats
open
start
cross
PC3
PC2
PC1
H
CS rats, split-half subset 1
open
cross
PC3
PC1
PC2
CS rats, split-half subset 2
open
cross
PC3
PC1
PC2

Figure 4 | Trajectory Coding Hypothesis and Biological Neural Trajectories.
(A) Cross-temporal decoding of integration-specific information from cluster 1 RNN neurons.
(B) Cross-temporal decoding of integration-specific information from cluster 1 RNN neurons with high synergy.
(C) Cross-temporal decoding of integration-specific information from cluster 1 RNN neurons with low synergy.
(D) Heatmaps of raw firing rates for RNN neurons from a representative CS + door-opening trial (left) and for biological cluster 1 non-exploratory neurons in the CS group during the delayed escape task (right). Neurons in the right heatmap are ordered by overall activity.
(E) Hypotheses on dynamic coding. (Left) Attractor hypothesis: Multiple or continuous stable converging points exist, and changes in input cause neural dynamics to move from one point to another, thereby encoding varying states. (Right) Trajectory hypothesis: Neural dynamics are not, or are only minimally, related to attractors (which may not exist); instead, changes in input cause changes in trajectory, which encode varying states.
(F) Fixed-point analysis of a representative trial in an RNN. Green dots represent stable fixed points (CS period), while black and yellow X-marks denote unstable fixed points (interval and open periods).
(G) GPFA trajectories of in vivo single-unit recording data: (left) CS rats, (right) neutral rats (n = 25 neurons, 16 rats). The mean latency from door opening to crossing was 56.363 ± 8.22 s.
(H) Validation of GPFA trajectories in CS group recordings. Neurons were ranked in descending order of mean z-scored firing rates (baseline to 5-s post-crossing) and split into odd- and even-indexed subsets. The trajectory from subset 1 (left; n = 13 neurons, 6 rats) served as the reference, while the trajectory from subset 2 (right; n = 12 neurons, 10 rats) was aligned accordingly. For clarity, only the segment from door opening to crossing is displayed. The mean latency from door opening to crossing for subset 1 was 53.306 ± 14.857 s. The mean latency from door opening to crossing for subset 2 was 50.969 ± 7.417 s.
Raw data from Fig. 4 of Han et al. (12), *Cell Reports*, were reanalyzed using GPFA. © Elsevier. Panel D (Right) are adapted from Fig. S4E of Han et al. (12), *Cell Reports*, © Elsevier.

Fig. S1

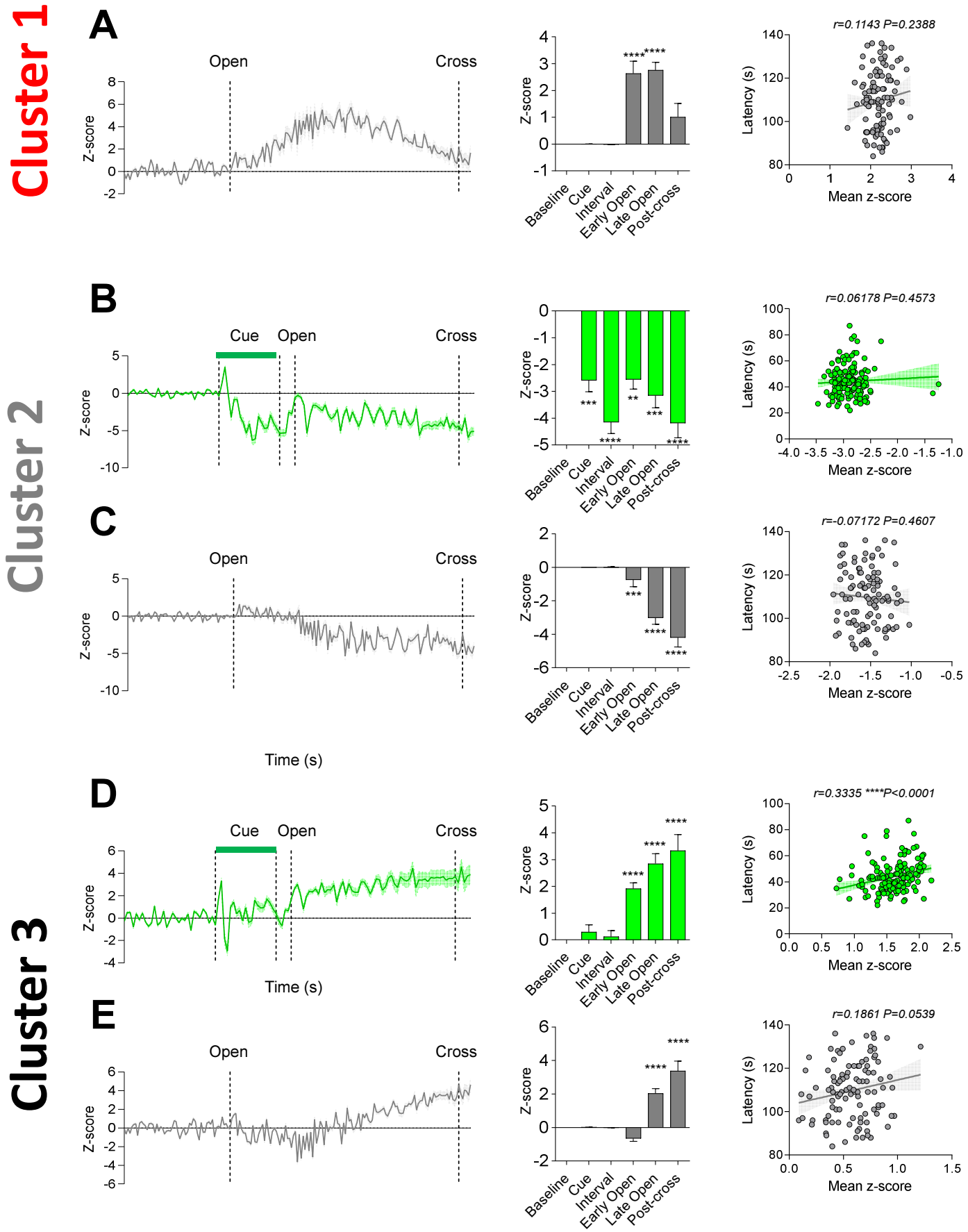
Cluster 1
Cluster 2
Cluster 3
A
B
C
D
E
Open
Cross
Cue
Z-score
Time (s)
Baseline
Cue
Interval
Early Open
Late Open
Post-cross
Latency (s)
Mean z-score
r=0.1143 P=0.2388
r=0.06178 P=0.4573
r=-0.07172 P=0.4607
r=0.3335 ****P<0.0001
r=0.1861 P=0.0539

Figure S1 | RNN activity of Clusters 1–3 under CS and door-opening only conditions

(A) **Cluster 1, door-opening only.** Left, Z-scored firing rate of Cluster 1 neurons (n = 24) in a representative trial. Middle, average Z-score for each behavioral period across all trials; Friedman test, ****p < 0.0001; Dunn's post hoc—Base vs Cue, ****p < 0.0001; Base vs Interval, ***p = 0.0001; Base vs Early Open, ****p < 0.0001; Base vs Late Open, **p = 0.0034; Base vs Post-escape, p = 0.7134. Right, Spearman correlation between crossing latency and Cluster 1 activity (n = 147; r = 0.1143, p = 0.2388).

(B) **Cluster 2, CS.** Left, Z-scored firing rate of Cluster 2 neurons (n = 38) in a representative trial. Middle, average Z-score for each behavioral period across all trials; Friedman ****p < 0.0001; Dunn's—Base vs Cue, ***p < 0.003; Base vs Interval, ****p < 0.0001; Base vs Early Open, **p = 0.0072; Base vs Late Open, ***p = 0.0002; Base vs Post-escape, ****p < 0.0001. Right, correlation between crossing latency and Cluster 2 activity (n = 147; r = 0.0618, p = 0.4573).

(C) **Cluster 2, door-opening only.** Left, Z-scored firing rate of Cluster 2 neurons (n = 38) in a representative trial. Middle, average Z-score for each behavioral period across all trials; Friedman ****p < 0.0001; Dunn's—Base vs Cue, p > 0.9999; Base vs Interval, p > 0.9999; Base vs Early Open, ***p = 0.0009; Base vs Late Open, ****p < 0.0001; Base vs Post-escape, ****p < 0.0001. Right, correlation between crossing latency and Cluster 2 activity (n = 147; r = –0.0717, p = 0.4607).

(D) **Cluster 3, CS.** Left, Z-scored firing rate of Cluster 3 neurons (n = 38) in a representative trial. Middle, average Z-score for each behavioral period across all trials; Friedman ****p < 0.0001; Dunn's—Base vs Cue, p > 0.9999; Base vs Interval, p > 0.9999; Base vs Early Open, ****p < 0.0001; Base vs Late Open, ****p < 0.0001; Base vs Post-escape, ****p < 0.0001. Right, correlation between crossing latency and Cluster 3 activity (n = 147; r = 0.3335, ****p < 0.0001).

(E) **Cluster 3, door-opening only.** Left, Z-scored firing rate of Cluster 3 neurons (n = 38) in a representative trial. Middle, average Z-score for each behavioral period across all trials; Friedman ****p < 0.0001; Dunn's—Base vs Cue, p = 0.2867; Base vs Interval, p > 0.9999; Base vs Early Open, p > 0.9999; Base vs Late Open, ****p < 0.0001; Base vs Post-escape, ****p < 0.0001. Right, , correlation between crossing latency and Cluster 3 activity (n = 147; r = 0.1861, p = 0.0539).

Fig. S2

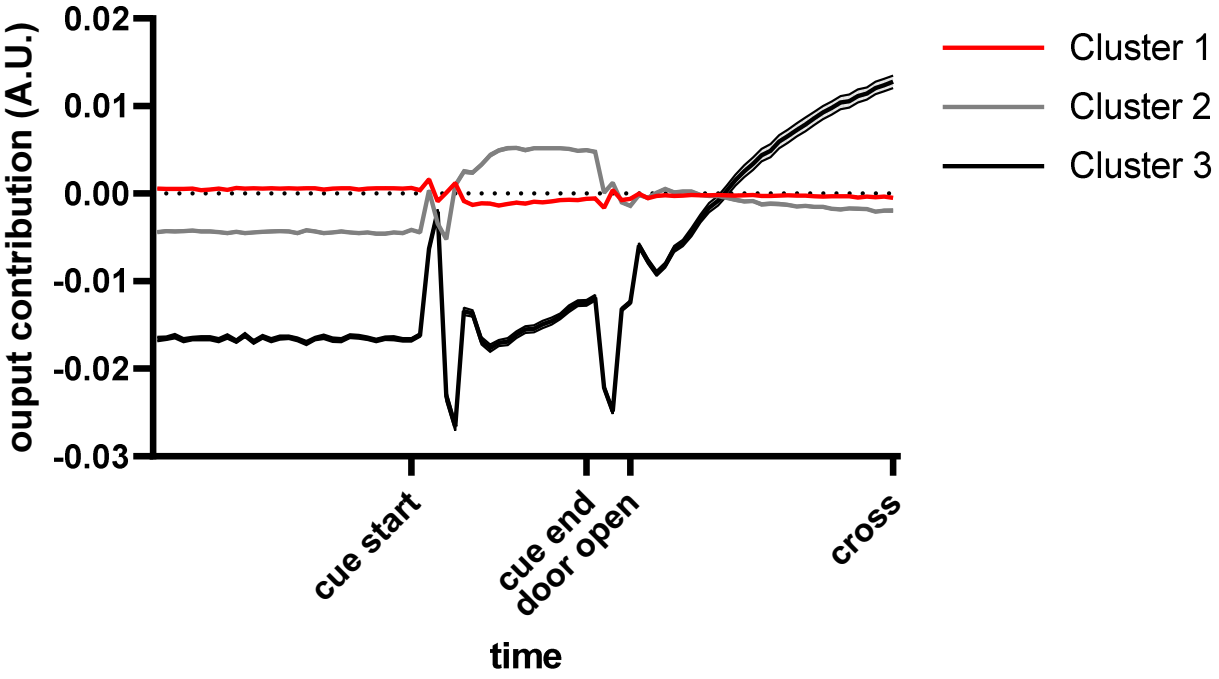

Figure S2 | Contribution of each cluster to the output during the delayed escape test
Relative contributions of Clusters 1–3 to the network output across behavioral periods in the delayed escape test.

Fig. S3

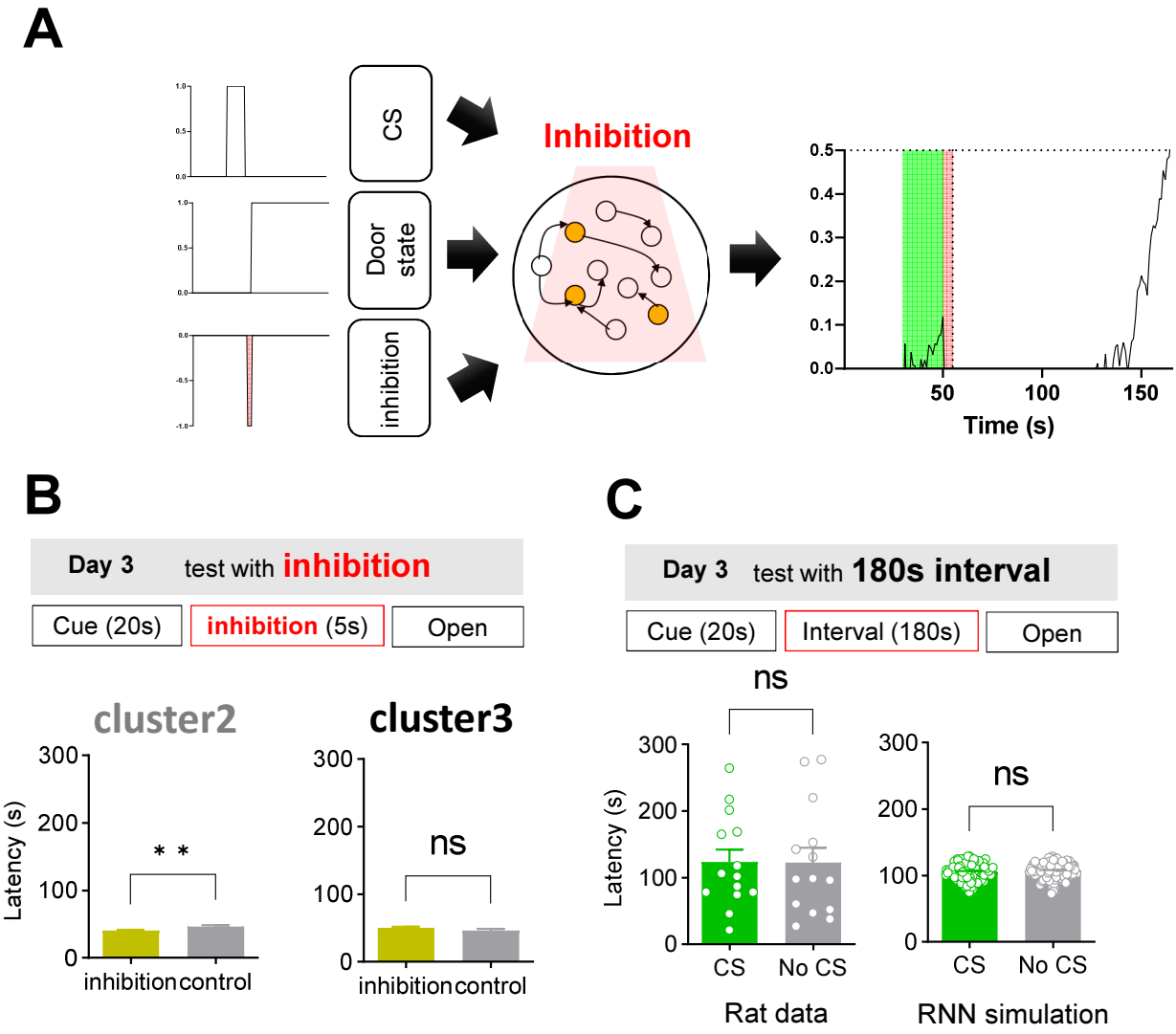

Figure S3 | RNN activity of Clusters 2 and 3 under inhibition (left) and escape latency under 180-s delay condition

(A) Schematic of the inhibition-simulation protocol. In the RNN model, inhibition was applied for 5 s during the delay interval after CS offset and before the door-opening signal.

(B) Bar graphs showing crossing latency under inhibition of Cluster 2 or Cluster 3 neurons.: Cluster 2, **p = 0.0023; Cluster 3, p = 0.1511 (both Mann–Whitney).

(C) Cross-latency under a prolonged 180-s CS–door-opening interval. Bar graphs show crossing latencies in rats (left) and in the RNN model (right): rats, p = 0.9720 (unpaired *t*-test); RNN, p = 0.3574 (unpaired *t*-test).

Fig. S4

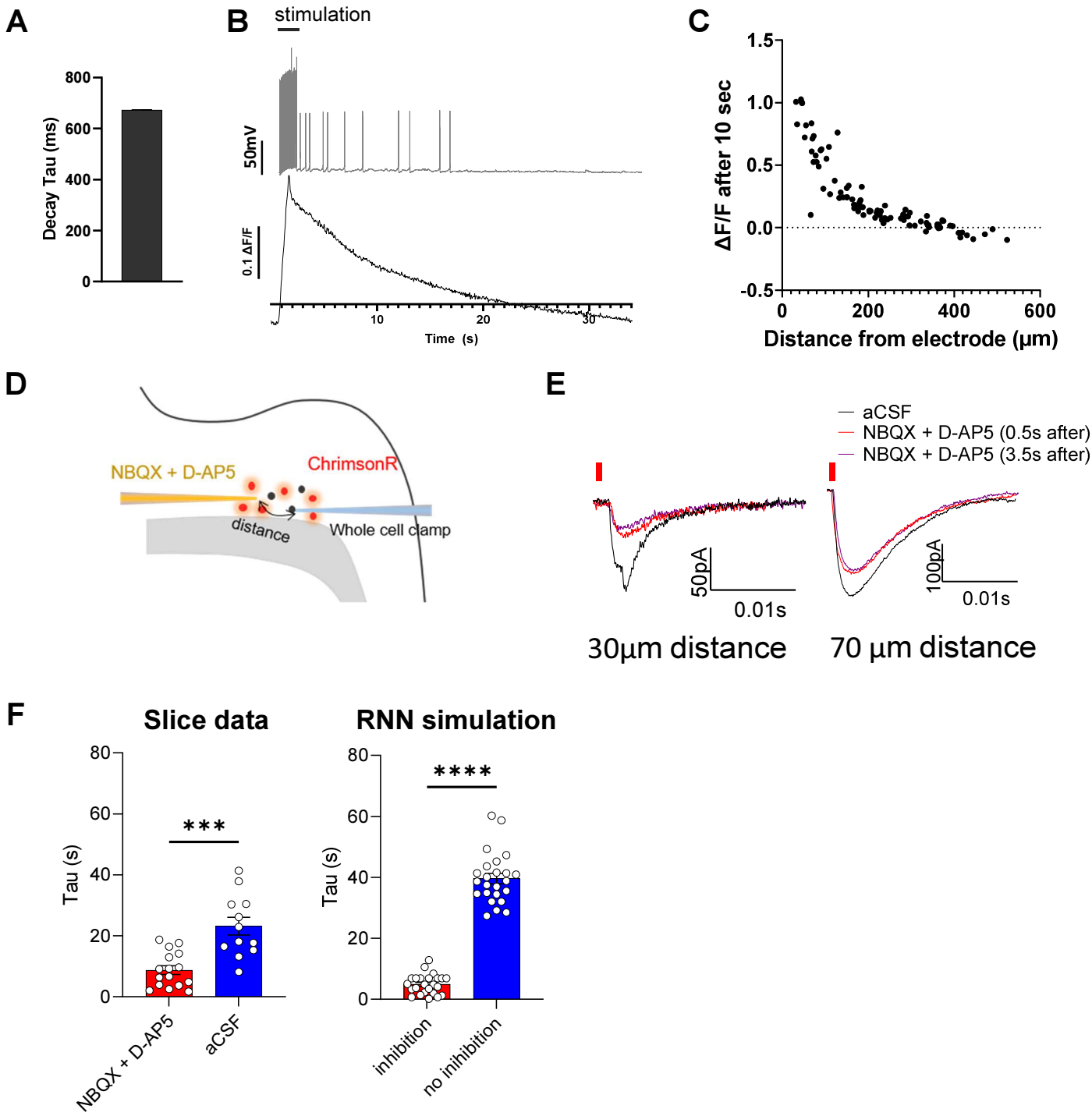

Figure S4 | Slice physiology and pharmacological manipulation of claustral persistent activity

(A) Membrane-potential decay τ during whole-cell recordings.
(B) Top: voltage trace showing persistent spiking after 20 Hz, 200 µA, 1-s stimulation; Bottom: corresponding GCaMP6f fluorescence changes recorded from the same cell.
(C) Relationship between the distance from the stimulating electrode to the recorded cell and the persistence ratio (peak fluorescence / fluorescence at 10 s), calculated from the dataset shown in Fig. 2J.
(D) Schematic of pressure injection of drugs onto whole-cell patch-clamped neurons in horizontal claustral slices.
(E) Representative EPSCs recorded under control (aCSF, no pressure injection) and after NBQX + D-AP5 pressure injection with the drug pipette positioned 30 µm or 70 µm from the recorded cell. EPSCs were evoked by electrical stimulation at 0.5 s and 3.5 s after injection. (F) Decay kinetics measured from the graphs in Fig. 2M and N under NBQX + D-AP5 or aCSF treatment. Left, decay τ after application of NBQX + D-AP5 (n = 16) or aCSF (n = 12) 10 s after electrical stimulation; **p = 0.0012, Mann–Whitney. Right, decay τ with versus without recurrent inhibition in the RNN (n = 24 each); ****p < 0.0001, Mann–Whitney.

Fig. S5

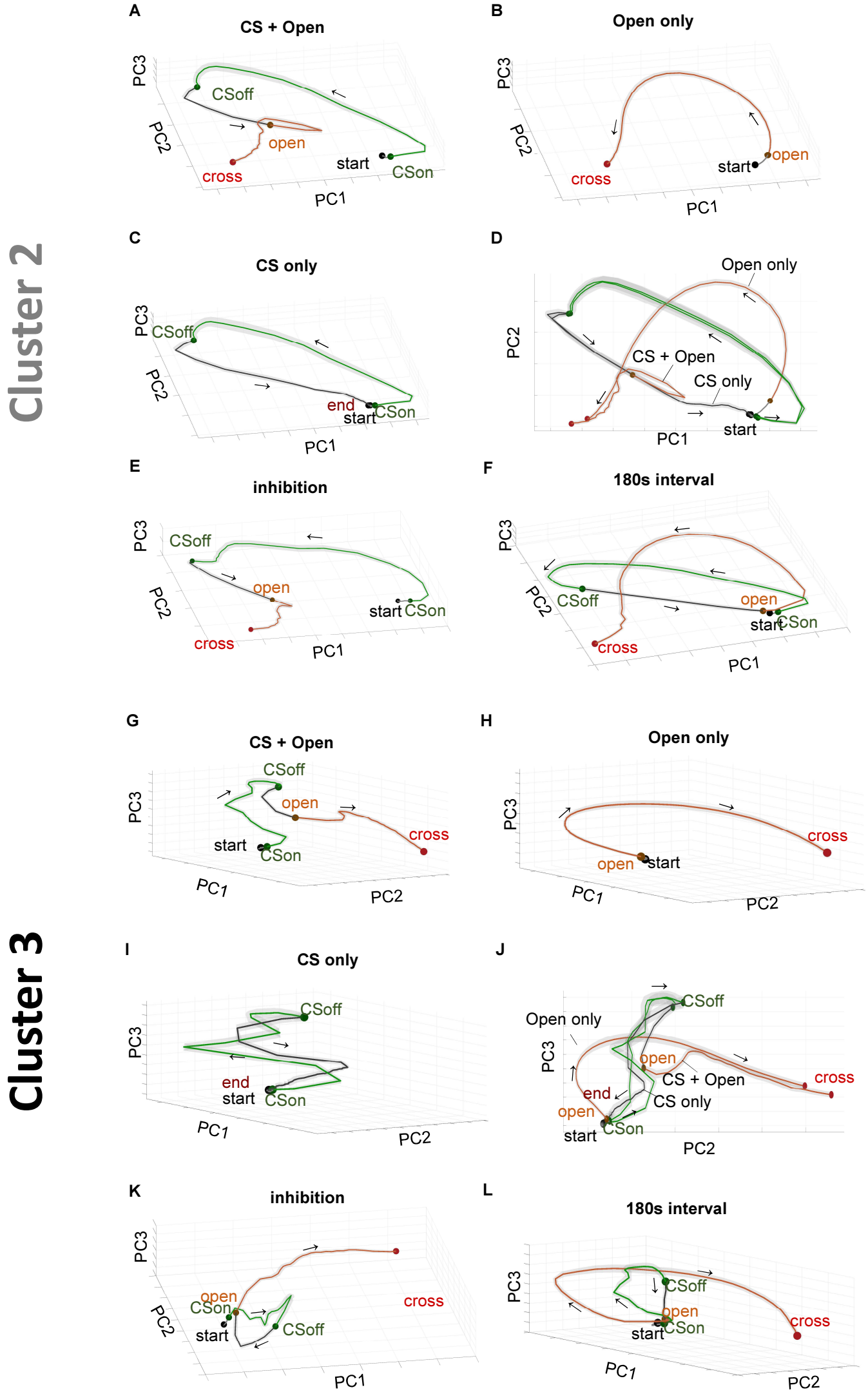
Cluster 2
Cluster 3
A
CS + Open
PC3
PC2
PC1
CSoff
open
start
CSon
cross
B
Open only
PC3
PC2
PC1
cross
start
open
C
CS only
PC3
PC2
PC1
CSoff
end
start
CSon
D
Open only
PC2
PC1
CS + Open
CS only
start
E
inhibition
PC3
PC2
PC1
CSoff
open
start
CSon
cross
F
180s interval
PC3
PC2
PC1
CSoff
open
start
CSon
cross
G
CS + Open
PC3
PC1
PC2
CSoff
open
start
CSon
cross
H
Open only
PC3
PC1
PC2
cross
open
start
I
CS only
PC3
PC1
PC2
CSoff
end
start
CSon
J
Open only
PC3
PC2
CSoff
open
CS + Open
cross
end
CS only
open
start
CSon
K
inhibition
PC3
PC2
PC1
open
CSon
start
CSoff
cross
L
180s interval
PC3
PC1
PC2
CSoff
open
start
CSon
cross

Figure S5 | PCA trajectories of Clusters 2 and 3 under various simulation conditions

(A–F) Cluster 2: time-normalized, trial-averaged PCA trajectories for (A) CS + door-opening, (B) door opening only, (C) CS only, (D) overlay of A–C, (E) inhibition simulation, and (F) 180 s delay between CS offset and door-opening.
(G–L) Cluster 3 trajectories arranged as in panels A–F.

Fig. S6

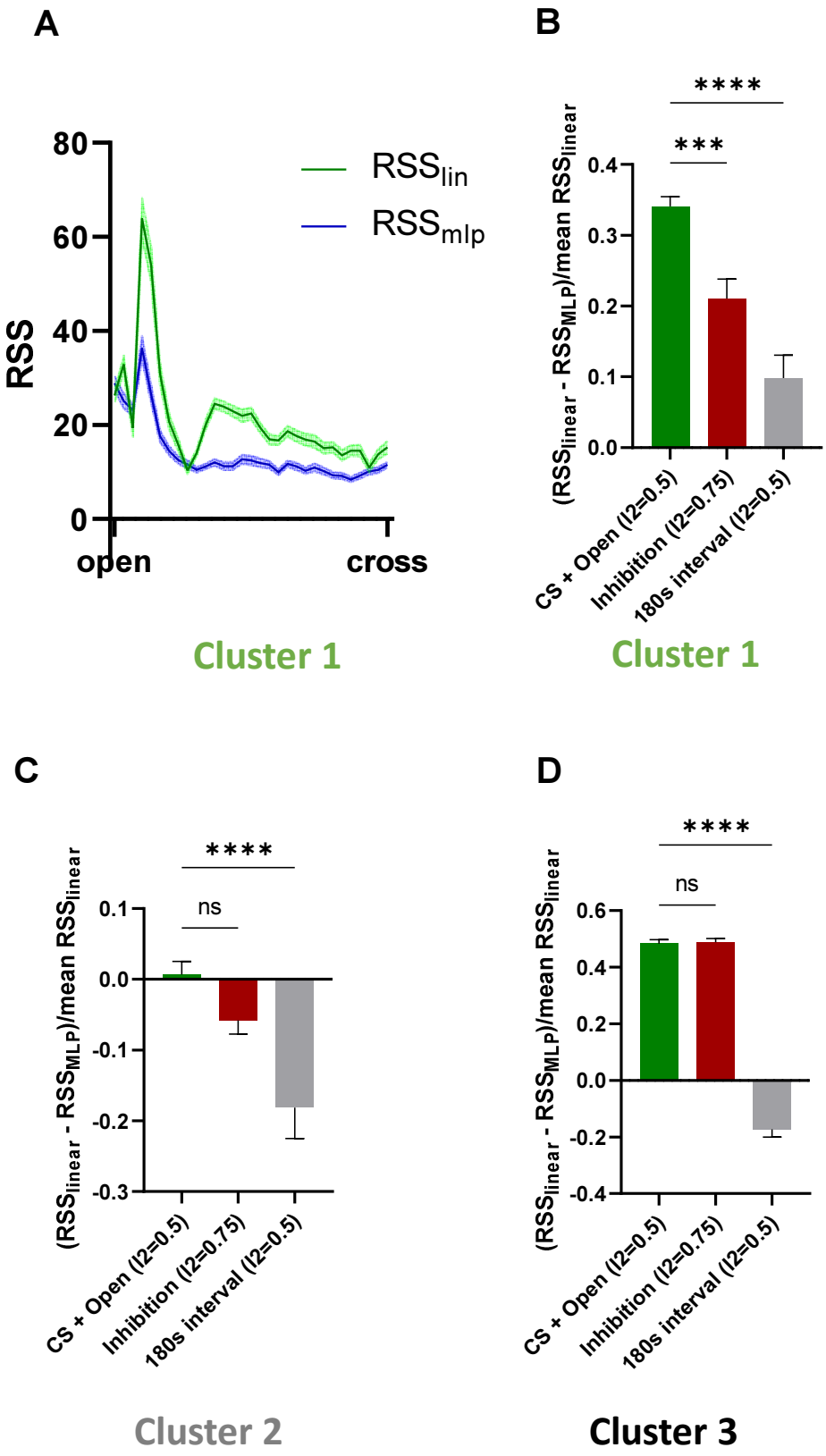

Figure S6 | Model-fit comparison across clusters

(A) Residual sum of squares (RSS) from a linear regression model and a multilayer perceptron (MLP) across normalized time bins. Data are taken from the Open event to the Cross event (for the CS only case, from the open event to the mean cross latency of the CS + door-opening condition). Green line: RSS of the linear regression model; blue line: RSS of the MLP.

(B–D) Difference in residual sum of squares (ΔRSS) between linear regression and MLP fits, normalized by the mean RSS of the linear model, across conditions.; bars, mean ± SEM.

(B) Cluster 1: CS + door-opening vs inhibition, **p = 0.0063; CS + door-opening vs 180 s delay, ****p < 0.0001.

(C) Cluster 2: CS + door-opening vs inhibition, p = 0.6735; CS + door-opening vs 180 s delay, **p = 0.0021.

(D) Cluster 3: CS + door-opening vs inhibition, p = 0.4067; CS + door-opening vs 180 s delay, ****p < 0.0001 (one-way ANOVA with Holm–Šidák).

Fig. S7

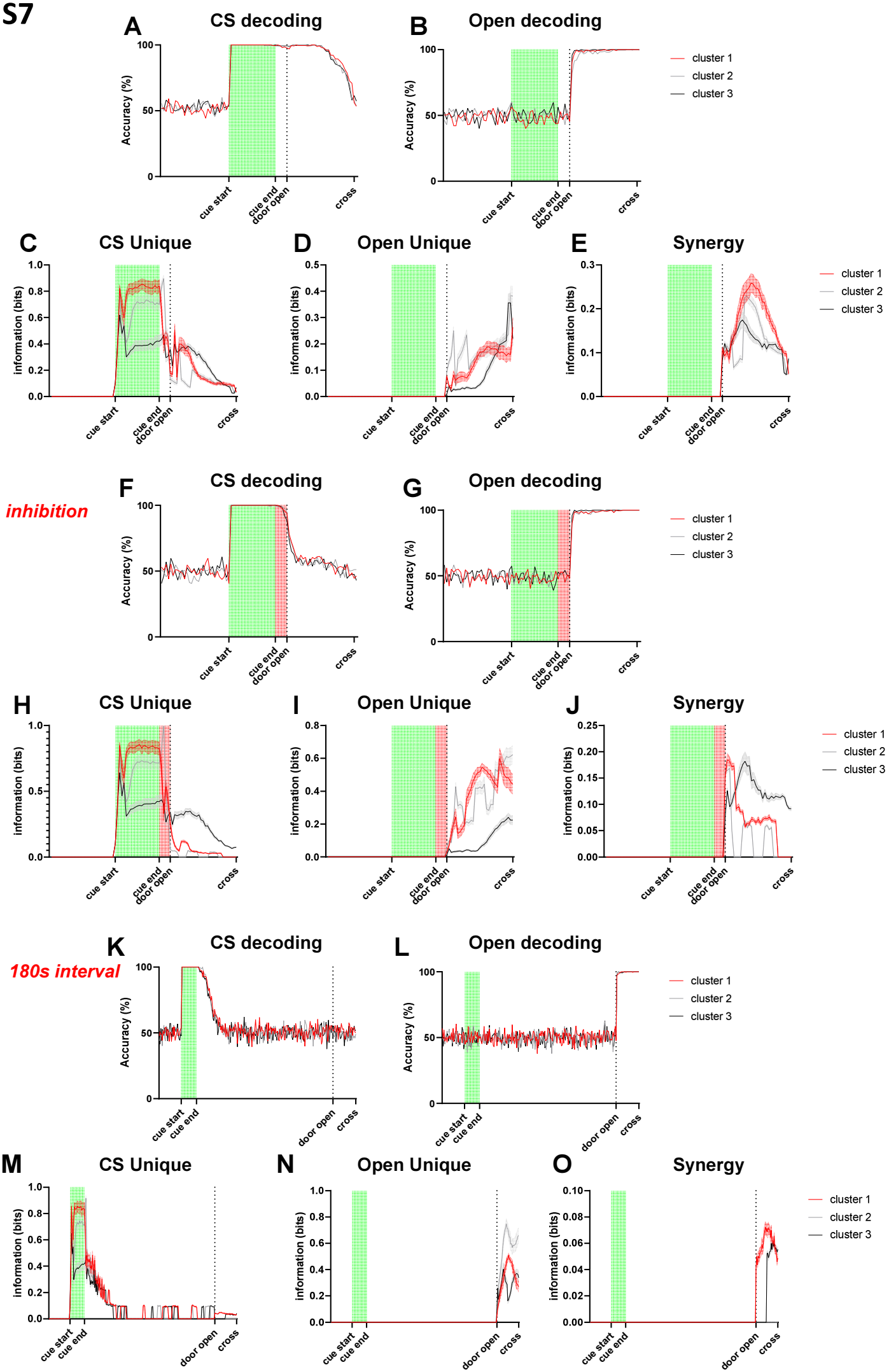
A
CS decoding
B
Open decoding
Accuracy (%)
cue start
cue end
door open
cross
cluster 1
cluster 2
cluster 3
C
CS Unique
D
Open Unique
E
Synergy
information (bits)
inhibition
F
CS decoding
G
Open decoding
H
CS Unique
I
Open Unique
J
Synergy
180s interval
K
CS decoding
L
Open decoding
M
CS Unique
N
Open Unique
O
Synergy

Figure S7 | Decoding performance and PID analysis across clusters

(A) Decoding accuracy for classifying CS-present versus CS-absent trials using cluster-specific decoders.
(B) Decoding accuracy for classifying Open-present versus Open-absent trials using cluster-specific decoders.
(C–E) PID metrics (CS-unique, Open-unique, synergy) for neurons in each cluster.
(F–J) Same analyses as in panels A–E under the inhibition condition.
(K–O) Same analyses as in panels A–E under a 180 s CS–Open interval.

Fig. S8

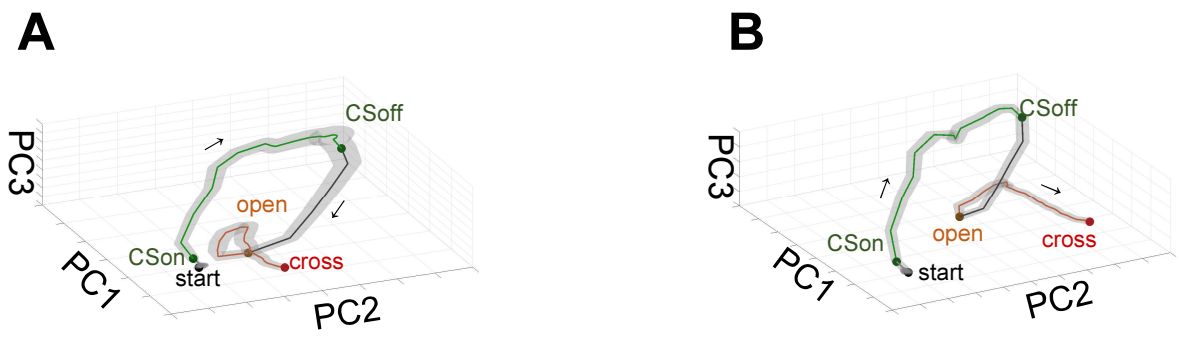

Figure S8 | PCA trajectories of high- and low-synergy neurons

(A–B) Time-normalized, trial-averaged PCA trajectories for (A) high-synergy and (B) low-synergy Cluster 1 RNN neurons.