IOP Publishing J. Neural Eng. 18 (2021) 066012 https://doi.org/10.1088/1741-2552/ac3266

Journal of Neural Engineering

PAPER

Towards threshold invariance in defining hippocampal ripples

Yusuke Watanabe[1,2,*], **Mami Okada**[1] **and Yuji Ikegaya**[1,3,*]

[1] Graduate School of Pharmaceutical Sciences, The University of Tokyo, Tokyo 113-0033, Japan
[2] Institute for Advanced Co-Creation Studies, Osaka University, Osaka 565-0871, Japan
[3] Center for Information and Neural Networks, National Institute of Information and Communications Technology, Suita City, Osaka 565-0871, Japan
[*] Authors to whom any correspondence should be addressed.

E-mail: ywatanabe@alumni.u-tokyo.ac.jp **and** yuji@ikegaya.jp

Keywords: sharp wave, hippocampus, convolutional neural network, probabilistic definition, artificial intelligence

RECEIVED
24 September 2021

ACCEPTED FOR PUBLICATION
22 October 2021

PUBLISHED
15 November 2021

Abstract

Objective. Hippocampal ripples are transient neuronal features observed in high-frequency oscillatory bands of local field potentials (LFPs), and they occur primarily during periods of behavioral immobility and slow-wave sleep. Ripples have been defined based on mathematically engineered features, such as magnitudes, durations, and cycles per event. However, the 'ripple' could vary from laboratory to laboratory because their definition is subject to human bias, including the arbitrary choice of parameters and thresholds. In addition, LFPs are often influenced by myoelectric noise arising from animal movement, making it difficult to distinguish ripples from high-frequency noises. These problems have to be overcome. *Approach.* We extracted ripple candidates under few constraints and labeled them as binary or stochastic 'true' or 'false' ripples using Gaussian mixed model clustering and a deep convolutional neural network (CNN) in a weakly supervised fashion. *Main results.* Our automatic method separated ripples and myoelectric noise and was able to detect ripples even when the animals were moving. Moreover, we confirmed that a CNN detected ripples defined by our method. Leave-one-animal-out cross-validation estimated the accuracy, the area under the precision-recall curve, the receiver operating characteristic curve to be 0.88, 0.99 and 0.96, respectively. *Significance.* Our automatic ripple detection method will reduce time spent on performing laborious experiments and analyses.

1. Introduction

Ripples are one form of the local field potential (LFP) oscillations observed in the hippocampus. As the name 'ripple' suggests, hippocampal ripples were first recognized based on their visually distinctive waveforms; they are characterized by 150–250 Hz components and durations of <150 ms (O'Keefe 1976). Ripples mainly occur during periods of awake immobility, consummatory behavior, and slow-wave sleep (Buzsáki *et al* 1992), thereby contributing to memory consolidation (Girardeau *et al* 2009, Ego-Stengel and Wilson 2010) and relating to memory retrieval (Wu *et al* 2017, Norman *et al* 2019). In conventional strategies for the detection of ripples in LFPs, ripples are determined based on criteria such as the ripple band root mean square (RMS) amplitude, duration, cycles per event, and animal head speed (Karlsson and Frank 2009, Ramirez-Villegas *et al* 2015, Kay *et al* 2016, Fernández-Ruiz *et al* 2019, Shin *et al* 2019, Wirtshafter *et al* 2019). In this process, analysts need to define ripples by arbitrarily setting the thresholds of these parameters. Therefore, the thresholds often vary among researchers, and 'ripples' are inconsistent among laboratories, potentially leading to different conclusions.

Another problem is that the thresholds are usually predetermined and fixed in experiments, although the LFP waveforms may differ among animals, recording sites, and electrodes. Thus, in some laboratories, when detected, ripples are manually screened by eye (Ramirez-Villegas *et al* 2015, Norimoto *et al* 2018). Eye inspection requires a great deal of skill and is subject to human bias. This task is also laborious and hinders upscaling to large data sets. Thus, overall, the current method of detecting ripples has problems related to objectivity, consistency, and reproducibility.

Deep convolutional neural networks (CNNs) are, in general, an appropriate tool for capturing the

shapes or local features of objects. CNN is a mathematical model inspired by the visual cortex system (Fukushima 1980, Krizhevsky *et al* 2012). CNNs have been studied especially since the ILSVRC2012 (ImageNet Large Scale Visual Recognition Challenge 2012), and the ability of these models to classify images is beyond that of humans (He *et al* 2016). In the present study, we used a one-dimensional CNN to define ripples with reduced human bias. The one-dimensional CNN extracts local features from LFP signals in the time domain and learns to discriminate true and false ripples at a stochastic scale. Additionally, based on the defined ripples, we evaluated the ability of the CNN to detect a ripple event from a series of 400 ms periods of raw LFPs.

2. Materials and methods

2.1. Animal ethics

All animal experiments were performed following the University of Tokyo Animal Experiments Implementation Manual with the Animal Experiment Committee's approval to minimize pain for experimental animals (approval number: P29-14). All mice were housed on a 12 h dark-light cycle (light from 07:00 to 19:00) at 22 $\pm$ 1 °C with food and water provided ad libitum. Five postnatal 9- to 12 week-old male C57BL/6 J mice (SLC, Shizuoka, Japan) were used.

2.2. Preparation of recording electrodes

Each recording electrode for hippocampal LFP consisted of four tetrodes (diameter 17 μm, size 0007, and polyimide-coated platinum-iridium alloy (90/10%), California Fine Wire Company) with depths from the brain surface that were independently adjustable. The main body for fixing the substrate (EIB-36-PTB, Neuralynx) was three-dimensionally designed with 3D CAD Fusion360 (AUTODESK) and was formed with a photocurable 3D printer (Form2, Formlabs). The platinum coating was applied at the tips of tetrodes so that the impedances ranged from 150 to 300 kΩ.

2.3. Surgery

Mice were anesthetized with 3% isoflurane inhalation. Under a concentration of 1.0%–1.5% isoflurane, electrode implantation was performed as follows. Anesthesia was confirmed by the lack of paw withdrawal, whisker movement, and eye-blink reflexes. The skin was subsequently removed from the head. A craniotomy (2.5 $\times$ 2.0 mm^2) was performed, which was centered at 1.8 mm posterior to the bregma and 1.8 mm ventrolateral to the sagittal suture. Two screws (0.8 $\times$ 3.0 mm^2; Muromachi Kikai Co., Ltd) were embedded in the skull at the bilateral cerebellum until they reached the brain surface. The tips of the tetrodes were placed on the brain surface. One of the four electrodes was used as a reference by placing it in a shallow layer where cortical firings were not recorded. The surface of the brain and the areas around the tetrodes were covered with Kwik-Sil Silicone Elastomer (World Precision Instruments). The skull surface was thinly covered with an adhesive (Super Bond C&B, Sun Medical) and was fixed on the skull using dental cement (Refine Bright, Yamahachi Dental Mfg., Co.). The screw used in this process served as a ground, and this ground and the ground of the board (EIB36-PTB, Neuralynx) were connected with a wire. A wire (A633, COONER WIRE) to record myoelectric potential (MEP) was inserted into the trapezius muscles (muscles at the base of the head and neck). Finally, the ground wire and wire used to record the MEP were covered with dental cement. After the surgeries, electrodes were covered with a cap made with a hot-melt type 3D printer (UP Plus2, Sun Stella Co., Ltd). Then, each mouse was returned to its cage.

2.4. Adjusting the recording electrode positions in the hippocampus

After the operation, the electrode positions in the hippocampus were adjusted while the mice were in their home cages. Every screw for adjustment was tightened one turn every few minutes, which deepened the screw by 250 μm until 1,000 μm from the brain surface was reached. With the waveforms monitored, each screw was tightened from 1/8 to 1/4 turn each time until the recording electrode reached a point just above the hippocampus. Then, each screw was loosened two turns and was left still for one day. Each screw was tightened by 1/4 turn each day until each electrode reached the target depth, and the maximum rotation angle of the screw was limited to 1/8 turn per day. The adjustment was stopped at the point at which large amplitude ripples were observed.

2.5. Electrophysiological recording

The hippocampal LFP and trapezius MEP were simultaneously recorded for five mice. After the recording electrodes in the hippocampus reached the desired depths, which was more than 14 d after surgery, the recording was performed with a data acquisition system (CerePlex Direct, Blackrock) for up to five days in the home cages of mice or a novel environment depending on the trial day, as shown in table 1.

The sampling rate for recording the raw hippocampal LFP and trapezius MEP was 2 kHz. The raw data were recorded after applying a 500 Hz low-pass filter. Data were recorded under free-moving conditions and with food and water provided ad libitum. The novel environment was more spacious than the home cages, and it included three different objects. All recordings were performed under a 12-hr light-dark cycle, with the dark period beginning at 19:00.

Table 1. Recording Environments. Biological signals of Mice #1, #2–4, and #5 were recorded for one day, four days, and five days, respectively. On the first and third days, experiments were performed in the home cages. On the second, fourth, and fifth days, experiments were performed in a novel environment.

Mouse #	Day 1	Day 2	Day 3	Day 4	Day 5
#1	Home	—	—	—	—
#2	Home	Novel	Home	Novel	—
#3	Home	Novel	Home	Novel	—
#4	Home	Novel	Home	Novel	
#5	Home	Novel	Home	Novel	Novel

Table 2. Data Source. The name of each data directory indicates the day of the surgery and the first day of the recording. To record hippocampal LFP, up to three tetrodes, or 12 electrodes, were inserted in the hippocampus.

		Tetrodes	
Mouse #	Data directory	Hippocampus	Trapezius
#1	180709_mouse4 (180 530 ope)	tt2, tt6	tt8
#2	180 909–12_mouse11 (180 802 ope)	tt2, tt6, tt7	tt3
#3	181 001–04_mouse15 (180 907 ope)	tt2, tt6, tt7	tt8
#4	181 007–12_mouse17 (180 919 ope)	tt2, tt6, tt7	tt8
#5	181 016–19_mouse18 (180 928 ope)	tt2, tt6, tt7	tt8

2.6. Histochemical verification of recording sites

After the recording experiments, the mice were anesthetized with urethane. Anesthesia was confirmed by the lack of paw withdrawal, whisker movement, and eye-blink reflexes. Overall, 25 μA currents were applied via electrodes in the hippocampus for 10 s to burn tissues at the recording sites. After the chest was dissected, from the left ventricle, ice-cooled Phosphate buffered saline (PBS; 137 mM NaCl, 2.7 mM KCl, 20 mM Na_2HPO_4, and 1.5 mM KH_2PO_4) and 4% paraformaldehyde (PFA) in PBS (4% PFA in PBS) were perfused in this order. The head was cut and allowed to stand overnight. The brain was then removed and immersed in 4% PFA overnight and then immersed twice in a 30% sucrose/PBS solution overnight. The brain was snap-frozen on dry ice and stored at −80 °C. Coronal sections with a thickness of 40 μm were prepared using a cryostat (CM3050 S, Leica). Each brain section was mounted on a microscope slide, stained with Cresyl Violet, and enclosed with a coverslip. Recording sites in the hippocampus were validated with records of the 3D coordinates of tetrodes, the number of electrodes, and the burn marks in coronal sections.

2.7. Data source

The identifiers used for the data are shown in table 2.

2.8. Tools for data analysis

All code is available on the GitHub repository page (https://github.com/ywatanabe1989/towards-threshold-invariance-in-defining-hippocampal-ripples).

GNU Octave 5.2 (www.gnu.org/software/octave/index) or MATLAB 2017b (MathWorks, Natick, MA, USA; www.mathworks.com/products/matlab.html) was used to read the recorded data from disks. For the remaining analyses, Python 3.6.8 (www.python.org/) was used.

The repository contains a definition file of a singularity container (https://sylabs.io/guides/3.7/user-guide/) to reproduce the Octave and Python environment.

2.9. Definition of ripple candidates

The recorded hippocampal LFP amplitude vector was downsampled from 2 kHz to 1 kHz using the imported function 'scipy.signal.decimate'. This process can be expressed by the pseudocode below.

$$\mathbf{a}_{\text{LFP\_hipp\_1kHz}}(\mathbf{t}) \leftarrow \mathbf{f}_{\text{down\_sample}}\left(\mathbf{a}_{\text{LFP\_hipp\_2kHz}}(\boldsymbol{t}')\right), \tag{1}$$

where $\mathbf{a}_{\text{LFP\_hipp\_1kHz}}(\boldsymbol{t}) \in \mathbb{R}^T$ is the downsampled hippocampal LFP amplitude vector with a virtual sampling rate of 1 kHz, $\mathbf{f}_{\text{down\_sample}}(\cdot)$ is the imported Python function 'scipy.signal.decimate', and $\mathbf{a}_{\text{LFP\_hipp\_2kHz}}(\boldsymbol{t}') \in \mathbb{R}^{2T}$ is the hippocampal LFP amplitude vector recorded at a sampling rate of 2 kHz. Then, the ripple band LFP was extracted by filtering the downsampled LFP series $\mathbf{a}_{\text{LFP\_hipp\_1kHz}}(\boldsymbol{t})$ with a bandpass filter for the ripple band (150–250 Hz) based on the pseudocode below.

$$\mathbf{a}_{\text{ripple\_band}}(\boldsymbol{t}) \leftarrow \mathbf{f}_{\text{ripple\_band\_filter}}\left(\mathbf{a}_{\text{LFP\_hipp\_1kHz}}(\boldsymbol{t})\right), \tag{2}$$

where $\mathbf{a}_{\text{ripple\_band}}(\boldsymbol{t})$ is the extracted ripple band LFP amplitude vector and $\mathrm{f}_{\text{ripple\_band\_filter}}$ is the imported Python function 'ripple_detection.core.filter_band', which uses a 100-degree FIR (finite impulse response) bandpass filter designed using the imported Python function 'scipy.signal.remez' with 25 Hz of the transition band width. By taking the square of the ripple band LFP amplitude (μV) at each time point, the ripple band LFP power (μV^2) was calculated.

$$\mathrm{p}_{\text{ripple\_band}}(t) = \mathrm{a}^2_{\text{ripple\_band}}(t), \tag{3}$$

where $\mathrm{p}_{\text{ripple\_band}}(t)$ and $\mathrm{a}_{\text{ripple\_band}}(t)$ are the ripple band LFP power (μV^2) and ripple band LFP amplitude (μV) at time t, respectively. The ripple

band LFP power was smoothed over time using a 32 ms Gaussian kernel with the parameter $\sigma = 4$ ms. This was conducted with the imported Python function 'ripple_detection.core.gaussian_smooth' as expressed in the following pseudocode:

$$\mathbf{p}_{\text{smoothed\_ripple\_band}}(\boldsymbol{t}) \leftarrow \mathbf{f}_{\text{Gaussian\_smoother}} \times \left(\mathbf{p}_{\text{ripple\_band}}(\boldsymbol{t})\right), \quad (4)$$

where $\mathbf{p}_{\text{smoothed\_ripple\_band}}(\boldsymbol{t})$ is a Gaussian-smoothed ripple band LFP power vector and $\mathrm{f}_{\text{Gaussian\_smoother}}(\cdot)$ is the imported Python function 'ripple_detection.core.gaussian_smooth' with the parameter $\sigma = 4$ ms. Then, the ripple band LFP magnitude (μV) at time t was defined as the root of the smoothed ripple band LFP power (μV^2) at the corresponding time as follows:

$$\mathrm{m}_{\text{ripple\_band}}(t) = \sqrt{\mathrm{p}_{\text{smoothed\_ripple\_band}}(t)}, \quad (5)$$

where $\mathrm{m}_{\text{ripple\_band}}(t)$ is the ripple band LFP magnitude (μV) at time t.

Ripple candidates were defined as LFP events with ripple band LFP magnitudes that continuously exceeded 1 SD for at least 15 ms. The defined onsets and offsets were redefined to the time points at which the ripple band LFP magnitude reached the mean value based on the imported Python function 'ripple_detection.core.extend_threshold_to_mean', as shown below.

$$\overline{\mathrm{m}_{\text{ripple\_band}}} = \frac{\sum_{t=1}^{T} \mathrm{m}_{\text{ripple\_band}}(t)}{T}, \quad (6)$$

where T is the total sampling time for the electrode, which is calculated by multiplying the recording time (s) by the virtual sampling rate of 1000 (Hz). Consequently, every ripple candidate was defined with a set of onset and offset times.

Also, as shown in figure 1, the normalized ripple band LFP magnitude at time t was calculated as follows:

$$\mathrm{m}_{\text{normalized\_ripple\_band}}(t) = \frac{\mathrm{m}_{\text{ripple\_band}}(t)}{\mathrm{s}_{\mathrm{m}_{\text{ripple\_band}}}}, \quad (7)$$

where $\mathrm{m}_{\text{normalized\_ripple\_band}}(t)$ is the normalized ripple band LFP magnitude at time t and $\mathrm{s}_{\mathrm{m}_{\text{ripple\_band}}}$ is the standard deviation of the ripple band LFP magnitude (μV) over time t as

$$\mathrm{s}_{\mathrm{m}_{\text{ripple\_band}}} = \sum_{\tau=1}^{T} \sqrt{\frac{\left(\mathrm{m}_{\text{ripple\_band}}(\tau) - \overline{\mathrm{m}_{\text{ripple\_band}}}\right)^2}{T}}. \quad (8)$$

2.10. Definition of the 'three variables' of ripple candidates

From every ripple candidate, the following 'three variables' were calculated: the duration (ms), the normalized peak magnitude (a.u.), and the mean normalized magnitude of the MEP of the trapezius for each ripple candidate (a.u.) (figures 2 and 4).

The duration (ms) of the $\mathrm{i_c}$ th ripple candidate $\mathrm{d}(\mathrm{i_c})$ was defined as follows:

$$\mathrm{d}(\mathrm{i_c}) = t_{\text{offset}}(\mathrm{i_c}) - t_{\text{onset}}(\mathrm{i_c}), \quad (9)$$

where $t_{\text{offset}}(\mathrm{i_c})$ and $t_{\text{onset}}(\mathrm{i_c})$ are the offset (ms) and onset (ms) of the $\mathrm{i_c}$th ripple candidate, respectively. Note that from the definition of ripple candidates (see the previous section), the ripple band LFP magnitude at both onsets and offsets was the mean of the ripple band LFP magnitude for each recording electrode.

$$\begin{aligned}&\mathrm{m}_{\text{ripple\_band}}(t_{\text{offset}}(\mathrm{i_c})) \\ &= \mathrm{m}_{\text{ripple\_band}}(t_{\text{onset}}(\mathrm{i_c})) \\ &= \overline{\mathrm{m}_{\text{ripple\_band}}}. \end{aligned} \quad (10)$$

The normalized ripple peak magnitude (a.u.) for the $\mathrm{i_c}$th ripple candidate $\mathrm{m}_{\text{normalized\_ripple\_peak}}(\mathrm{i_c})$ was defined as follows:

$$\mathrm{m}_{\text{normalized\_ripple\_peak}}(\mathrm{i_c}) = \frac{\mathrm{m}_{\text{ripple\_peak}}(\mathrm{i_c})}{\mathrm{s}_{\mathrm{m}\,\text{ripple\_peak}}} \quad (11)$$

$$\begin{aligned}\mathrm{m}_{\text{ripple\_peak}}(\mathrm{i_c}) \leftarrow \max(\{&\mathrm{m}_{\text{ripple\_band}}(\tau) \,|\, \tau \\ &= t_{\text{onset}}(\mathrm{i_c}),\, t_{\text{onset}}(\mathrm{i_c}) + 1,\, \ldots,\, t_{\text{offset}}(\mathrm{i_c})\}) \end{aligned} \quad (12)$$

$$\overline{\mathrm{m}_{\text{ripple\_peak}}} = \frac{\sum_{\mathrm{i_c}=1}^{\mathrm{N_c}} \mathrm{m}_{\text{ripple\_peak}}(\mathrm{i_c})}{\mathrm{N_c}}. \quad (13)$$

$$\mathrm{s}_{\mathrm{m}_{\text{ripple\_peak}}} = \sum_{\mathrm{i_c}=1}^{\mathrm{N_c}} \sqrt{\frac{\left(\mathrm{m}_{\text{ripple\_peak}}(\mathrm{i_c}) - \overline{\mathrm{m}_{\text{ripple\_peak}}}\right)^2}{\mathrm{N_c}}}, \quad (14)$$

where $\mathrm{N_c}$ is the sample size of ripple candidates defined from the corresponding electrode.

The mean normalized magnitude of the MEP of the trapezius for ripple candidates (a.u.) was defined as follows. Fst, the magnitude of the MEP was obtained as the same as the pseudocode and equations (1) and (3)–(5).

$$\mathbf{a}_{\text{MEP\_1kHz}}(\boldsymbol{t}) \leftarrow \mathbf{f}_{\text{down\_sample}}(\mathbf{a}_{\text{MEP\_2kHz}}(\boldsymbol{t}')) \quad (15)$$

$$\mathrm{p}_{\text{MEP\_1kHz}}(t) = \mathrm{a}^2_{\text{MEP\_1kHz}}(t) \quad (16)$$

$$\mathbf{p}_{\text{smoothed\_MEP\_1kHz}}(\boldsymbol{t}) \leftarrow \mathbf{f}_{\text{Gaussian\_smoother}} \times (\mathbf{p}_{\text{MEP\_1kHz}}(\boldsymbol{t})) \quad (17)$$

$$\mathrm{m}_{\text{MEP\_1kHz}}(t) = \sqrt{\mathrm{p}_{\text{smoothed\_MEP\_1kHz}}(t)}, \quad (18)$$

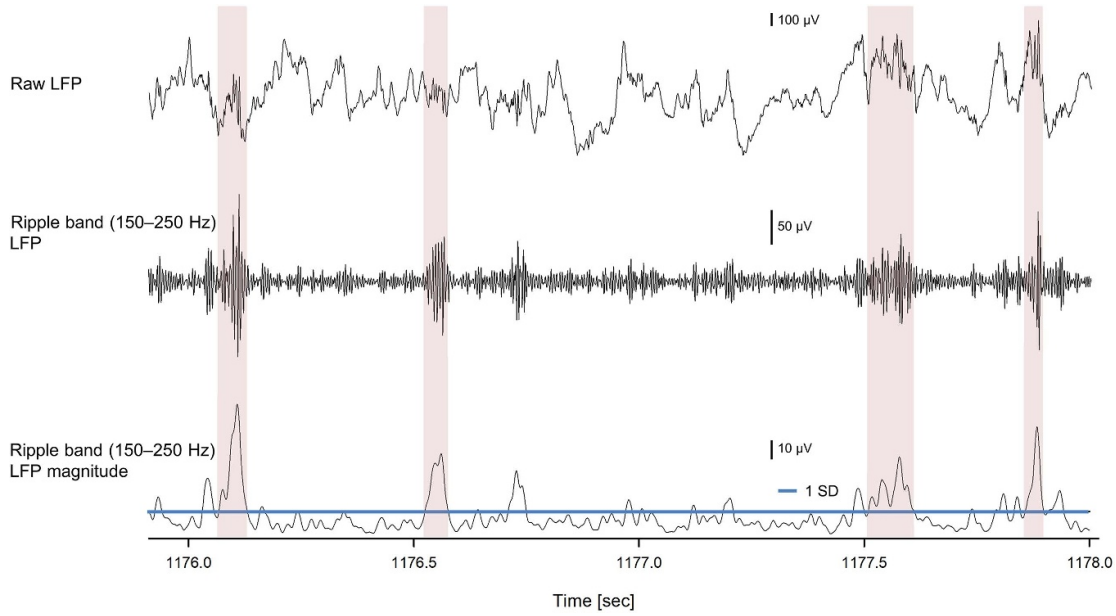

Figure 1. Selection of ripple candidates. *Top*: an LFP trace recorded from the CA1 pyramidal layer of the right hippocampus from a freely moving mouse. *Middle*: the corresponding 150–250 Hz-bandpassed LFP trace (see equations (1) and (2) in section 2). *Bottom*: the corresponding normalized ripple band (150–250 Hz) LFP magnitude (see equations (1)–(8) in section 2). Red areas show the periods determined as ripple candidates.

where $\mathbf{a}_{\mathrm{MEP\_1kHz}}(\boldsymbol{t}) \in \mathbb{R}^T$ is the downsampled MEP amplitude vector with a virtual sampling rate of 1 kHz, $\mathbf{a}_{\mathrm{MEP\_2kHz}}(t') \in \mathbb{R}^{2T}$ is the MEP amplitude vector recorded at 2 kHz, and $\mathrm{m}_{\mathrm{MEP\_1kHz}}(t)$ is the magnitude of the MEP at the time t. Finally, by averaging the magnitude of the MEP during a ripple and dividing the mean by the SD, we defined the normalized magnitude of the MEP for the $\mathrm{i_c}$ th ripple candidate $\mathrm{m_{MEP}}(\mathrm{i_c})$ (a.u.) as follows:

$$\mathrm{m_{MEP}}(\mathrm{i_c}) = \frac{\sum_{\tau=t_{\mathrm{onset}}(\mathrm{i_c})}^{t_{\mathrm{offset}}(\mathrm{i_c})} \mathrm{m_{MEP\_1kHz}}(\tau)}{\mathrm{d}(\mathrm{i_c})} \tag{19}$$

$$\mathrm{m_{normalized\_MEP}}(\mathrm{i_c}) = \frac{\mathrm{m_{MEP}}(\mathrm{i_c})}{\mathrm{s_{m_{MEP}}}} \tag{20}$$

$$\overline{\mathrm{m_{MEP}}} = \frac{\sum_{\mathrm{i_c}=1}^{\mathrm{N_c}} \mathrm{m_{MEP}}(\mathrm{i_c})}{\mathrm{N_c}} \tag{21}$$

$$\mathrm{s_{m_{MEP}}} = \sqrt{\frac{\sum_{\mathrm{i_c}=1}^{\mathrm{N_c}} \left(\mathrm{m_{MEP}}(\mathrm{i_c}) - \overline{\mathrm{m_{MEP}}}\right)^2}{\mathrm{N_c}}}. \tag{22}$$

2.11. Relationship between hippocampal LFPs and myoelectricity of the trapezius

To evaluate the correlations between FFT powers of hippocampal LFP and the MEP of trapeziuses, we set bin size as 1024 ms. After being given a random initial sampling time ranging from 0 to 1023 ms, hippocampal LFP data from each recording electrode were binned to consecutive 1024 ms samples without overlaps. Additionally, the MEP of the trapezius during the corresponding periods was sampled. Note that the samples here were different from those in the previous section, 'Definition of the 'Three Variables' of Ripple Candidates'. Here, not only ripple-candidate periods but also non-ripple-candidate periods were sampled. In each 1024 ms bin, the mean magnitude of the MEP of $\mathrm{i_{bin}}$th bin sample was determined as shown below.

$$\mathrm{m_{MEP\_bin}}(\mathrm{i_{bin}}) = \frac{\sum_{\tau=t_{\mathrm{bin\_onset}}(\mathrm{i_{bin}})}^{t_{\mathrm{bin\_onset}}(\mathrm{i_{bin}})+1{,}023} \mathrm{m_{MEP\_1kHz}}(\tau)}{1024}, \tag{23}$$

where $t_{\mathrm{bin\_onset}}(\mathrm{i_{bin}})$ is the onset (ms) of the $\mathrm{i_{bin}}$th time bin, and $\mathrm{m_{MEP\_1kHz}}(\tau)$ is the magnitude of the MEP (μV) at time τ, as calculated from the pseudocode and equations (15)–(18). Then, the mean magnitude of MEP $\mathrm{m_{normalized\_MEP\_bin}}(\mathrm{i_{bin}})$ was normalized as follows:

$$\mathrm{m_{normalized\_MEP\_bin}}(\mathrm{i_{bin}}) = \frac{\mathrm{m_{MEP\_bin}}(\mathrm{i_{bin}})}{\mathrm{s_{m_{MEP\_bin}}}} \tag{24}$$

$$\overline{\mathrm{m_{MEP\_bin}}} = \frac{\sum_{\mathrm{i_{bin}}=1}^{\mathrm{N_{bin}}} \mathrm{m_{MEP\_bin}}(\mathrm{i_{bin}})}{\mathrm{N_{bin}}} \tag{25}$$

$$\mathrm{s_{m_{MEP\_bin}}} = \sqrt{\frac{\sum_{\mathrm{i_{bin}}=1}^{\mathrm{N_{bin}}} \left(\mathrm{m_{MEP\_bin}}(\mathrm{i_{bin}}) - \overline{\mathrm{m_{MEP\_bin}}}\right)^2}{\mathrm{N_{bin}}}}. \tag{26}$$

From the hippocampal LFP in each time bin, the FFT power (a.u.) of f Hz component (f = 0, 1, …, 499) was obtained using the imported Python function 'scipy.fftpack.fft'.

The Pearson correlation coefficients between the mean magnitude of the MEP (a.u.) and the FFT

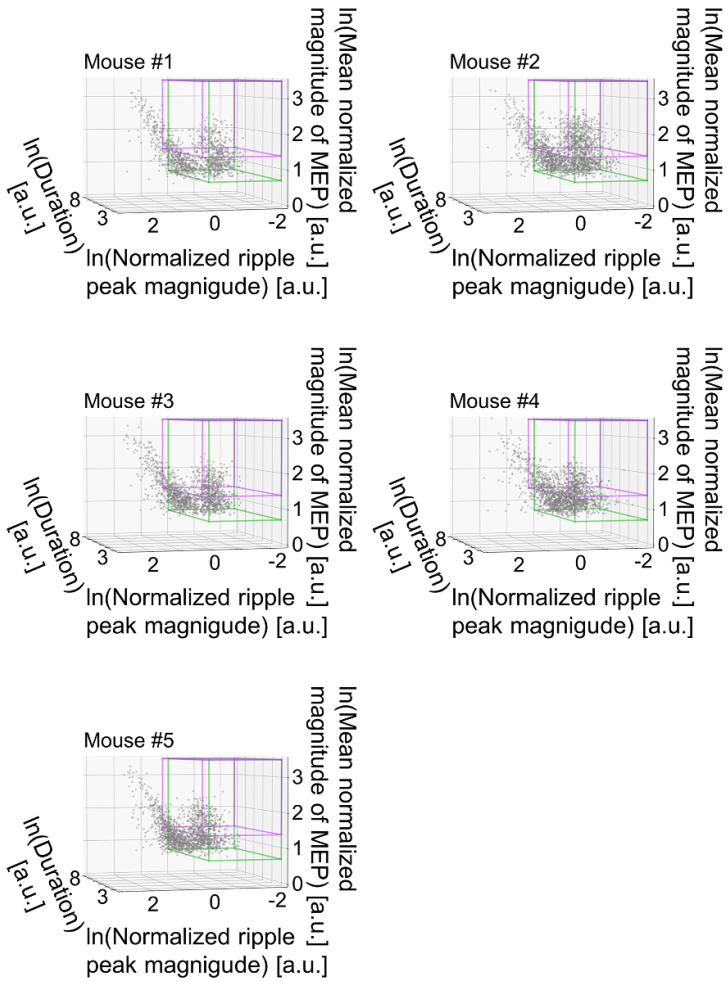

Figure 2. Three-dimensional visualization of ripple candidate. Each scatter-plotted sample shows ripple candidate. The axes of the three-dimensional space are the natural logarithm of (i) the duration (ms), (ii) mean normalized magnitude of the myoelectricity potential (MEP) of the trapezius (μV), and (iii) normalized ripple peak magnitude (ripple band: 150–250 Hz) (μV) for each ripple (see equations (9)–(22) in section 2). Each panel shows data for each mouse. For visualization purposes, randomly chosen samples were plotted (0.20% and 0.05% of all data for Mice #1 and #2–#5, respectively). The green and purple cubes show the regions where the ripple candidates are defined as 'ripples' with the criteria of Fernández-Ruiz *et al* (2019) and Kay *et al* (2016); these cubes do not overlap.

$$r_{m_{normalized\_MEP\_bin},\,p(f)} = \frac{\sum_{i_{bin}=1}^{N_{bin}} \left(m_{normalized\_MEP\_bin}\left(i_{bin}\right) - \overline{m_{normalized\_MEP\_bin}}\right)\left(p\left(f,\ i_{bin}\right) - \overline{p\left(f\right)}\right)/N_{bin}}{s_{m_{normalized\_MEP\_bin}} \cdot s_{p(f)}} \quad (27)$$

$$\overline{m_{MEP\_bin}} = \frac{\sum_{i_{bin}=1}^{N_{bin}} m_{normalized\_MEP\_bin}\left(i_{bin}\right)}{N_{bin}} \quad (28)$$

$$s_{m_{normalized\_MEP\_bin}} = \sqrt{\frac{\sum_{i_{bin}=1}^{N_{bin}} \left(m_{normalized\_MEP\_bin}\left(i_{bin}\right) - \overline{m_{normalized\_MEP\_bin}}\right)^2}{N_{bin}}} \quad (29)$$

$$\overline{p(f)} = \frac{\sum_{i_{bin}=1}^{N_{bin}} p(f,\ i_{bin})}{N_{bin}} \tag{30}$$

$$s_{p(f)} = \sqrt{\frac{\sum_{i_{bin}=1}^{N_{bin}} \left(p(f,\ i_{bin}) - \overline{p(f)}\right)^2}{N_{bin}}}, \tag{31}$$

where i_{bin} is the index for time bins, N_{bin} is the total number of time bins, and $p(f,\ i_{bin})$ is the FFT power (a.u.) of the f Hz component of the i_{bin}th 1024 ms hippocampal LFP sample (figure 3).

2.12. Partial correlation

The partial correlation between variables X and Y, controlling for the influence of variable Z, can be expressed as r_{XY_Z}, as shown below,

$$r_{XY\_Z} = \frac{r_{XY} - r_{XZ} \cdot r_{YZ}}{\sqrt{1 - r_{XZ}^2} \cdot \sqrt{1 - r_{YZ}^2}}, \tag{32}$$

where r_{XY}, r_{XZ}, and r_{YZ} are the Pearson correlation coefficients between the index variables.

2.13. Construction of a deep CNN

Based on Fawaz *et al* (2019), as one of the best models for time series classification, we chose ResNet 1D (He *et al* 2016), a deep CNN. We implemented ResNet with six minor revisions. We modified the ResNet 1D model and call it 'our CNN' in this paper (figure 7). Specifically, there were six modifications as follows. (a) The first filter length in each 'block' was shortened from 8 to 7. (b) The activation function was changed from the ReLU function (Hahnloser *et al* 2000) to the LeakyReLU function (Maas *et al* 2013), and the differential coefficient in the negative region was 0.1. (c) The number of filters in every convolutional layer was quadrupled (e.g. 64–256). (d) Two fully connected layers and a softmax layer were added at the end in the same order as listed. These two layers functioned as a stochastic classifier for two-class classification tasks. Dropout (Srivastava *et al* 2014) was applied in the first fully connected layer to prevent overfitting. Note that in our model, dropout was only applied on the downstream side of the batch normalization (Ioffe and Szegedy) layers because it was reported that the coapplication of dropout and batch normalization functions might lead to incongruity in results (Li *et al* 2019). (e) The model performed mixed-precision training based on the imported Python module 'apex'; the model got able to process data in a 32-bit floating format and a 16-bit floating format. (f) The model was replicated on four GPUs (ASUS GeForce GTX 1080 TI 11GB Turbo × 4) with the imported Python module 'torch.nn.parallel', and a multi-GPU parallel computing model was formed.

2.14. Isolating and trimming each ripple candidate as the CNN inputs

Our CNN requires an input vector to be a fixed length. We fixed the input length at 400 ms for a sampling rate of 1 kHz. Thus, the ith input vector was expressed as $\mathbf{x_i} \in \mathbb{R}^{400}$. If an input vector $\mathbf{x_i} \in \mathbb{R}^{400}$ includes multiple ripple candidates, the output may be unclear. Therefore, the following two steps were performed for each ripple candidate: (a) isolating the candidate and (b) either zero-padding or trimming the isolated candidate for the time length to be 400 ms (figure 8).

First, the i_cth ripple candidate $\mathbf{c}_{i_c} \in \mathbb{R}^{d(i_c)}$ was isolated from the onset to the offset as follows:

$$\mathbf{c}_{i_c} = \left(c_{i_c\_1},\ c_{i_c\_2},\ \ldots,\ c_{i_c\_d(i_c)}\right)^T \in \mathbb{R}^{d(i_c)}. \tag{33}$$

When the duration of a candidate was less than 400 ms, the candidate was padded with zeros to be extended symmetrically from the middle time (zero-padding). If the duration was 400 ms or longer, the LFP signal around the middle time was extracted (trimming). By zero-padding or trimming, 400 ms data sequence derived from the i_cth ripple candidatare $\mathbf{c}_{400\_i_c}$ was determined as follows:

$$\mathbf{c}_{400\_i_c} = \begin{cases} \left(0_{\mathbf{pre}}(i_c)^T, \mathbf{c_{i_c}}^T, 0_{\mathbf{post}}(i_c)^T\right)^T & when\ d(i_c) \le 400\,[ms] \\ \left(c_{i_c\_(t_{mid}(i_c)-199)}, c_{i_c\_(t_{mid}(i_c)-198)}, \cdots, c_{i_c\_(t\,mid(i_c)+200),}\right) & otherwise \end{cases} \tag{34}$$

$$t_{mid}(i_c) = \frac{t_{onset}(i_c) + t_{offset}(i_c)}{2}, \tag{35}$$

where $0_{\mathbf{pre}}(i_c) \in \mathbb{R}^{D_{pre}(i_c)}$ and $0_{\mathbf{post}}(i_c) \in \mathbb{R}^{(400-d(i_c)-D_{pre}(i_c))}$ are zero vectors to pad; $D_{pre}(i_c)$ is defined as $D_{pre}(i_c) = \left\lfloor \frac{400-d(i_c)}{2} \right\rfloor$; $t_{offset}(i_c)$ and $t_{onset}(i_c)$ are the offset and onset (ms) of the i_cth ripple candidate, respectively; and $c_{i_c\_t}$ is the hippocampal LFP amplitude (μV) at time t, which is also a component of the i_cth ripple candidate $\mathbf{c}_i$.

2.15. Dataset splitting

Dataset splitting was performed according to the mouse ID numbers (#1, #2, #3, #4, and #5). The following ten sub-datasets were defined. Sub-datasets $\mathcal{D}_{-1}$, $\mathcal{D}_{-2}$, $\mathcal{D}_{-3}$, $\mathcal{D}_{-4}$, and $\mathcal{D}_{-5}$ were defined as datasets that 'excluded' the data from mice #1, #2, #3, #4, and #5, respectively. Sub-datasets $\mathcal{D}_1$, $\mathcal{D}_2$, $\mathcal{D}_3$, $\mathcal{D}_4$,

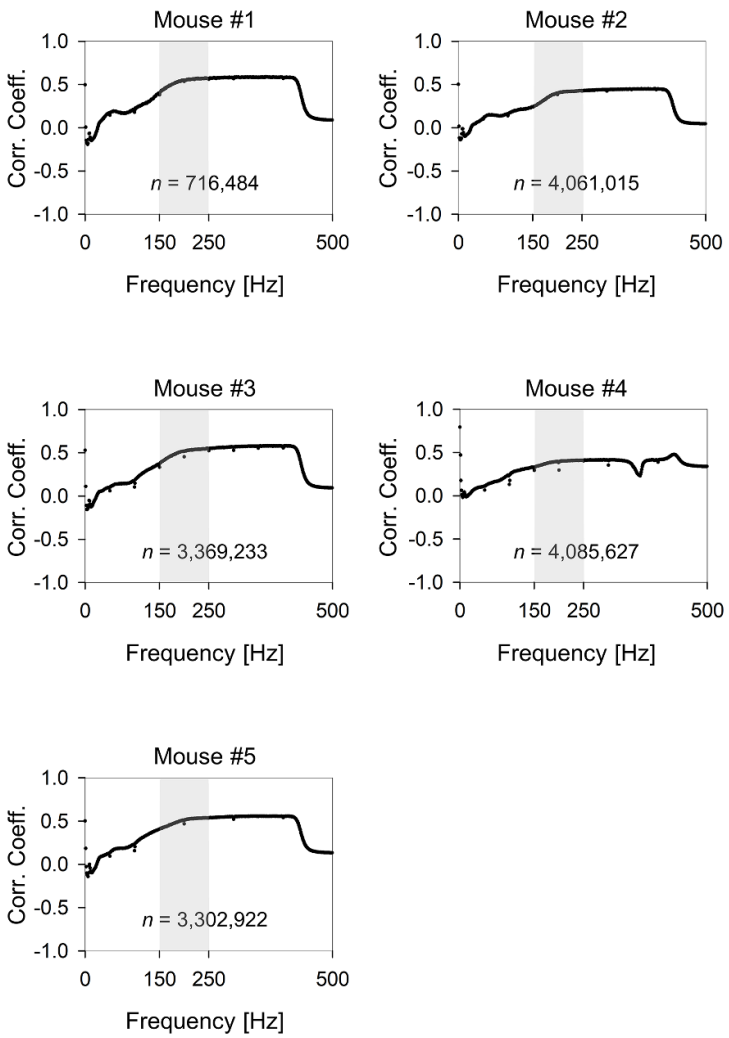

Figure 3. Relationship between hippocampal LFPs and trapezius myoelectric potentials (MEPs). Each panel shows the Pearson correlation coefficients between the normalized mean magnitude of the trapezius MEP and FFT powers of the hippocampal LFP for each mouse (*x*-axis: the corresponding frequency (see equations (23)–(31) in section 2). All pairs of correlations were statistically significant in all mice (*ps < .05; the test for no correlation; *n*s for 1,024 ms samples are indicated in panels). Gray area shows the ripple band (150–250 Hz).

and $\mathcal{D}_5$ were defined as datasets that 'only included' the data from mice #1, #2, #3, #4, and #5, respectively (figure 9). Here, $\mathcal{D}_1$, $\mathcal{D}_2$, $\mathcal{D}_3$, $\mathcal{D}_4$, and $\mathcal{D}_5$ were prepared as the test dataset in Result section titled 'Detecting Ripples by Using Our CNN.'

2.16. Confident learning to define ripples

To define ripples from ripple candidates, we conducted Confident Learning with the imported Python module 'cleanlab' (Northcutt *et al* 2019; https://github.com/cgnorthcutt/cleanlab). 'cleanlab.pruning.get_noise_indices' was applied on the ten sub-datasets defined in the previous section. The number of folds was set as five. Thus, each ripple candidate was labeled with five tags, depending on the datasets. For example, a ripple candidate acquired from mouse #1 was included in the sub-datasets $\mathcal{D}_1$, $\mathcal{D}_{-2}$, $\mathcal{D}_{-3}$, $\mathcal{D}_{-4}$, and $\mathcal{D}_{-5}$. Each ripple candidate was assigned five labels in total depending on sub-datasets.

2.17. Optimizing our CNN for defining ripples in confident learning

Our CNN requires the input length to be fixed. We set the input length to 400 (ms) (or (points) in our case since the virtual sampling rate was 1 kHz). The mini-batch size was set to 1024. The cross-entropy loss function was adopted to solve a classification problem. The learning rate was fixed to 1.0×10^{-3}. The training was performed in a supervised manner for one epoch. At the beginning of each epoch, the feeding order of samples $\{\mathbf{c}_{400\_i_c}\}$ ($i_c = 1, 2, \ldots$) into our CNN was shuffled. As the optimizer, the imported Python class 'ranger. Ranger' was used based on combining two mechanisms: RAdam (Tong *et al* 2019) and Lookahead (Zhang *et al* 2019). The dropout ratio of the first fully connected layer from the input side was set to 0.5 during the training stage.

During inference stages, the dropout value was set to zero, and batch normalization was performed in inference mode. That is, the learned parameters

of the batch normalization layer (gamma and beta) were fixed. The forward path of our CNN to define ripples ($f_{CNN\_define}(\cdot)$) can be expressed by the following pseudocode and equation set:

$$\hat{\mathbf{y}}_{\mathbf{i_c}} \leftarrow f_{CNN\_define}(\mathbf{c}_{400\_\mathbf{i_c}}) \tag{36}$$

$$\hat{\mathbf{y}}_{\mathbf{i_c}} = \begin{pmatrix} \hat{y}_{i_c\_0} \\ \hat{y}_{i_c\_1} \end{pmatrix} = \begin{pmatrix} P(\mathbf{c}_{\mathbf{i_c}} \in F_{Ripple\_CNN}) \\ P(\mathbf{c}_{\mathbf{i_c}} \in T_{Ripple\_CNN}) \end{pmatrix} \Leftrightarrow \begin{pmatrix} 1 - P(\mathbf{c}_{\mathbf{i_c}} \in T_{Ripple\_CNN}) \\ P(\mathbf{c}_{\mathbf{i_c}} \in T_{Ripple\_CNN}) \end{pmatrix}, \tag{37}$$

where F_{Ripple_CNN} and T_{Ripple_CNN} are the labels indicating false and true ripples estimated by our CNN, respectively, and $P(\mathbf{c}_{\mathbf{i_c}} \in T_{Ripple\_CNN})$ is the probability that the i_cth ripple candidate is defined as a true ripple by our CNN.

Based on the output $\hat{\mathbf{y}}_{\mathbf{i_c}}$, estimated labels ($= T_{Ripple\_CNN}$ or F_{Ripple_CNN}) were allocated for each ripple candidate $\mathbf{c}_{\mathbf{i_c}}$ by setting a decision threshold ($= d_{threshold\_define}$) as follows:

$$\begin{aligned} \mathbf{c}_{\mathbf{i_c}} \in T_{Ripple\_CNN} \textit{ when } \hat{y}_{i_c\_1} &= P(\mathbf{c}_{\mathbf{i_c}} \in T_{Ripple\_CNN}) \geqslant d_{threshold\_define} \\ \mathbf{c}_{\mathbf{i_c}} \in F_{Ripple\_CNN} \textit{ when } \hat{y}_{i_c\_1} &= P(\mathbf{c}_{\mathbf{i_c}} \in T_{Ripple\_CNN}) < d_{threshold\_define} \end{aligned} \tag{38}$$

2.18. Optimizing our CNN for detecting ripples

Optimization was performed on our CNN for ripple detection almost in the same way as the previous section. The differences are as follows. The mini-batch size was set to 1024. The training was performed for one epoch on labels acquired by the Gaussian mixture model clustering and Confident Learning method (F_{Ripple_GMM}/T_{Ripple_GMM}; see the previous section). To reflect the loss from the minority group (F_{Ripple_GMM} group), under-sampling was performed. Input for our CNN to detect ripples was 400 ms LFP raw signals $\{\mathbf{s}_{i_r}\}$ ($i_r = 1, 2, \ldots$). For simplicity, 400 ms raw LFP signals that included more than any part of two ripple candidates were excluded for this analysis. The forward path of our CNN to detect ripples ($f_{CNN\_detect}(\cdot)$) can be expressed by the following pseudocode and equation.

$$\hat{\mathbf{y}}_{i_r} \leftarrow f_{CNN\_detect}(\mathbf{s}_{i_r}) \tag{39}$$

$$\hat{\mathbf{y}}_{i_r} = \begin{pmatrix} \hat{y}_{i_r\_0} \\ \hat{y}_{i_r\_1} \end{pmatrix} = \begin{pmatrix} P(\mathbf{s}_{i_r} \in F) \\ P(\mathbf{s}_{i_r} \in T) \end{pmatrix} \Leftrightarrow \begin{pmatrix} 1 - P(\mathbf{s}_{i_r} \in T) \\ P(\mathbf{s}_{i_r} \in T) \end{pmatrix}, \tag{40}$$

where $\mathbf{s}_{i_r} \in \mathbb{R}^{400}$ is the i_rth 400 ms raw LFP signal, $\hat{\mathbf{y}}_{i_r}$ is the i_rth output vector, and F and T are the labels indicating signal s_{i_r} includes a false ripple or a true one, respectively.

Based on the output $\hat{\mathbf{y}}_{i_r}$, predicted labels ($= \hat{F}$ or $\hat{T}$) were allocated for each 400 ms raw LFP signal $\mathbf{s}_{i_r}$ ($i_r = 1, 2, \ldots$) by setting a decision threshold ($= d_{threshold\_detect}$) as follows:

$$\begin{aligned} \mathbf{s}_{i_r} \in \hat{F} \textit{ when } \hat{y}_{i_{r1}} = P(\mathbf{s}_{i_r} \in T) &< d_{threshold\_detect} \\ \mathbf{s}_{i_r} \in \hat{T} \textit{ when } \hat{y}_{i_{r1}} = P(\mathbf{s}_{i_r} \in T) &\geqslant d_{threshold\_detect}. \end{aligned} \tag{41}$$

3. Results

3.1. Thresholds in existing ripple detection methods and the defined 'ripples'

We examined the effects of differences in thresholds on defined 'ripples'. We used hippocampal LFP data and MEP data from the trapezius. These data were recorded simultaneously over five consecutive days from five mice (see Materials and Method).

First, ripple candidates were defined based on the definition of Kay *et al* (2016) with the following two modifications (*ns*: 621 589 for Mouse #1; 3855 213 for Mouse #2; 2979 288 for Mouse #3; 3471 373 for Mouse #4; and 3176 687 for Mouse #5). First, the threshold of the peak magnitude of a ripple band was reduced from 2 SD to 1 SD, and second, the process of sorting ripple candidates using animal head speed was removed (figure 1; see section 2). These two modifications aimed to include more true latent ripples in ripple candidate sets.

To define ripples from the candidates, we used the MEP of the trapezius and not the animal head speed for the following reasons. Existing methods define ripples only when the animal head speed is less than a threshold (e.g. 4 cm s^{-1}). Generally, vibrations or impacts on recording electrodes generate noise in recorded signals. Thus, setting a threshold for animal head speed helps avoid noisy periods when detecting ripples is difficult due to physical activities. However, animal head speed is not appropriate to use for detecting ripples (e.g. <50 ms), mainly because of the video-capturing limitations in time and space (e.g. $\leqslant$ 60 fps and ~50 MPs). On the other hands, MEP has higher resolutions in time and voltage scale (e.g. the sampling rate of 20 kHz and 16-bit signed integer data). Thus, we chose not animal head speed but MEP to indicate noisy LFP periods associated with animal movements.

First, from each ripple candidate, the following 'three variables' were calculated: the duration (ms), normalized peak ripple magnitude (a.u.), and mean normalized magnitude of the MEP of the trapezius for a ripple candidate (a.u.) (see section 2). Additionally, 'three ln-variables' were calculated by taking the natural logarithm of each of the 'three variables'.

Next, each ripple candidate was plotted in a 3D space with the 'three ln-variables' as the axes (figure 2). In figure 2, detecting ripples with one of the

existing methods (Karlsson and Frank 2009, Ramirez-Villegas *et al* 2015, Kay *et al* 2016, Fernández-Ruiz *et al* 2019, Shin *et al* 2019, Wirtshafter and Wilson 2019) is equal to establishing a specific cube in the 3D space in parallel with the axes and selecting the ripple candidates in the cube as ripples. For example, the green and purple cubes in figure 2 are constructed based on the thresholds used by Kay *et al* (2016) and Fernández-Ruiz *et al* (2019), respectively. Here, we assumed that the mean normalized magnitude of the MEP of the trapezius for a ripple candidate (a.u.) is directly proportional to the animal head speed in the same time resolution with the corresponding research.

$$\mathrm{m}_{\mathrm{normalized\_MEP}}(\mathrm{i_c}) = \frac{\mathrm{e}}{4}\,|v_{\mathrm{head}}(\mathrm{i_c})|, \qquad (42)$$

where $\mathrm{m}_{\mathrm{normalized\_MEP}}(\mathrm{i_c})$ is the normalized mean magnitude of the MEP (μV) of the $\mathrm{i_c}$th ripple candidate, as defined in equation (20), e is the Napier's constant, and v_{head} is the mean animal head speed (cm s^{-1}) during the $\mathrm{i_c}$th ripple candidate. As a result, the ripple candidates included in the cubes corresponding to the methods of Kay *et al* (2016) and Fernández-Ruiz *et al* (2019) were not consistent. As graphically shown in figure 2, it was revealed that 'ripples' are sensitive to the thresholds set in the existing detection methods.

3.2. Relationship between hippocampal LFPs and animal movements

As mentioned in the previous section, screening ripple candidates using animal head speed leads to inconsistent ripple detection. Thus, we use the mean normalized magnitude of the MEP for screening. If these two assumptions are valid, inversely, the larger the mean normalized magnitude of the MEP is, the larger the amplitude of the ripple band (150–250 Hz) LFP should be. We checked this hypothesis with our data.

All LFP and MEP data were binned to 1024 ms. The Pearson correlation coefficients between the FFT power of f Hz component (a.u.) (f = 0, 1, …, 499) and the mean magnitude of the MEP (a.u.) were calculated for the bins (figure 3; see section 2, especially equations (23)–(31)). The correlations were statistically significant in all five mice data (*p*s < .05, the test for no correlation; *n*s for 1024 ms bins are shown in figure 3). This result supported the hypotheses discussed earlier; noise caused by animal movements contaminates the recorded hippocampal LFP, and the mean normalized magnitude of the MEP (a.u.) is a useful barometer for noisy periods.

3.3. Defining ripples based on gaussian mixed clustering

As the previous section considers, the existing detection methods are designed to exclude false ripples from ripple candidate sets. The sorting process can be optimized. In existing approaches, although each ripple candidate is associated with the 'three variables' simultaneously, the thresholding processes for each variable are applied one by one. That is, the multivariate nature of these variables is not considered. In general, when trying to separate multivariate samples into two classes, an appropriate multivariate model yields better outcomes than univariate models applied one by one. Here, the higher the correlations among variables are, the more dominant the positive effect of a multivariate model is (Ichihara 1990). We assumed that the multivariate nature of the 'three ln-variables' should be utilized in defining hippocampal ripples.

Table 3. Partial correlations among the 'three ln-variables' of ripple candidates. In general, the partial correlation between variables *X* and *Y*, controlling for the influence of variable *Z*, or $\mathbf{r}_{XY\_Z}$, is defined in equation (32). The table shows the partial correlations among the 'three ln-variables' of ripple candidates: the natural logarithm of (i) the duration (ms), (ii) normalized ripple peak magnitude (ripple band: 150–250 Hz) (μV) for each ripple, and (iii) mean normalized magnitude of the myoelectricity potential (MEP) of the trapezius (μV) (see equations (9)–(22) in section 2). Each pair of the the 'three ln-variables' was correlated (*p*s < .001, *n*s = 621 589; 3855 213; 2979 288; 3471 373; and 3176 687 ripple candidates for Mouse #1, #2, #3, #4, and #5, multiple comparison using the test for no correlation with the Bonferroni correction).

ln(duration)	*X*	*Z*	*Y*
ln(normalized peak magnitude)	*Y*	*X*	*Z*
ln(mean normalized magnitude of MEP)	*Z*	*Y*	*X*
$\mathbf{r}_{XY\_Z}$ (Mean ± Std.)	0.53 ± 0.03	0.15 ± 0.07	0.23 ± 0.05

We first calculated the correlations among the 'three ln-variables'. The partial correlation coefficients among them were calculated to exclude a possible spurious correlation (table 3). For all three patterns, the partial correlations were statistically significant (*p*s < .001, *n* = 14 104 250 ripple candidates from five mice, multiple comparison using the test for no correlation with the Bonferroni correction). These results suggest that the process of distinguishing false ripples from ripple candidates can be optimized using a multivariate approach.

Next, we defined ripples using the multivariate nature. First, the histograms of the 'ln-variables' were plotted (figure 4). From the histograms, we hypothesized each distribution could be approximated as a sum of Gaussian functions. Thus, in the 3D space, Gaussian mixed model clustering is expected to separate ripple candidates well. The optimal numbers for clustering were estimated using the elbow method: 2, 2, 3, 2, and 3 for Mouse #1, #2, #3, #4, and #5, respectively (figure 5). Then, GMM clustering was performed on each mouse data (GMM; figure 6). As mentioned earlier, true ripples are thought to be included in the cluster whose centroid of which is smallest in the dimension of the MEP magnitude: 'Cluster T'. The other clusters, 'Cluster F_1

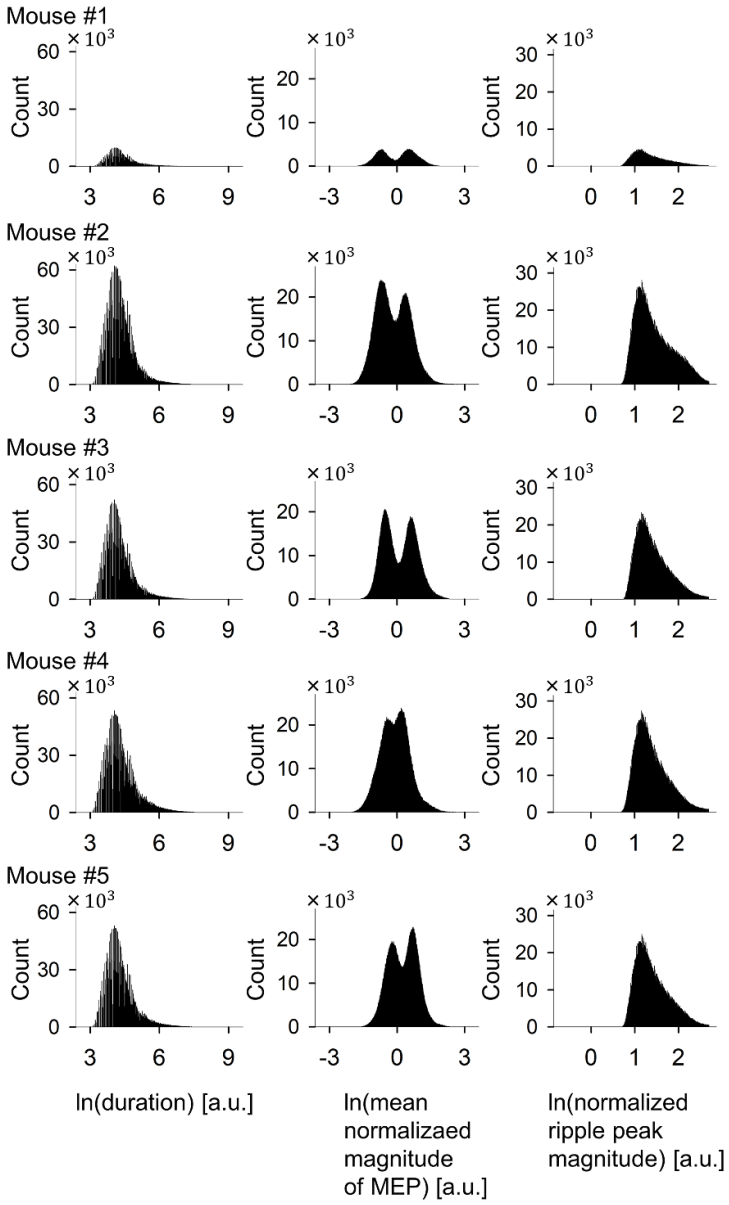

Figure 4. Distributions of the three parameters of the ripple candidates. Histograms of the ln(duration), ln(mean normalized magnitude of the MEP of the trapezius), and ln(normalized ripple peak magnitude) for each ripple candidate. Each row shows data for each mouse. We postulated that each distribution could be approximated as the sum of two Gaussian functions from these observations.

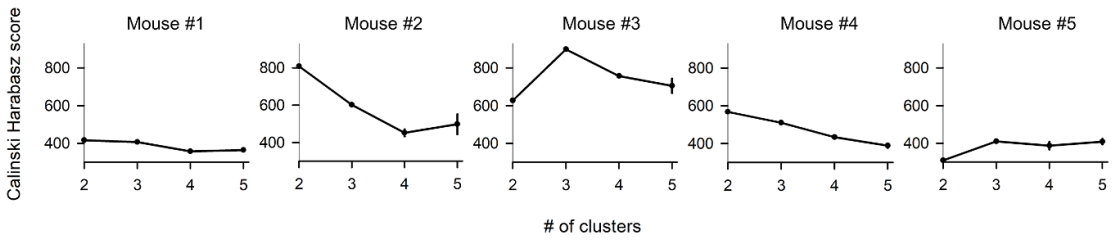

Figure 5. The elbow method to check the performance of Gaussian mixture model clustering. Each panel shows the Calinski Harabasz score for each mouse, as a measure of clustering performance. The higher the score, the better the performance. On each condition, the Calinski Harabasz score was calculated five times with different random seeds. The error bars shows Mean ± Std. The elbow method estimated the optimal number of clusters as 2, 2, 3, 2, and 3 for Mouse #1, #2, #3, #4, and #5, respectively.

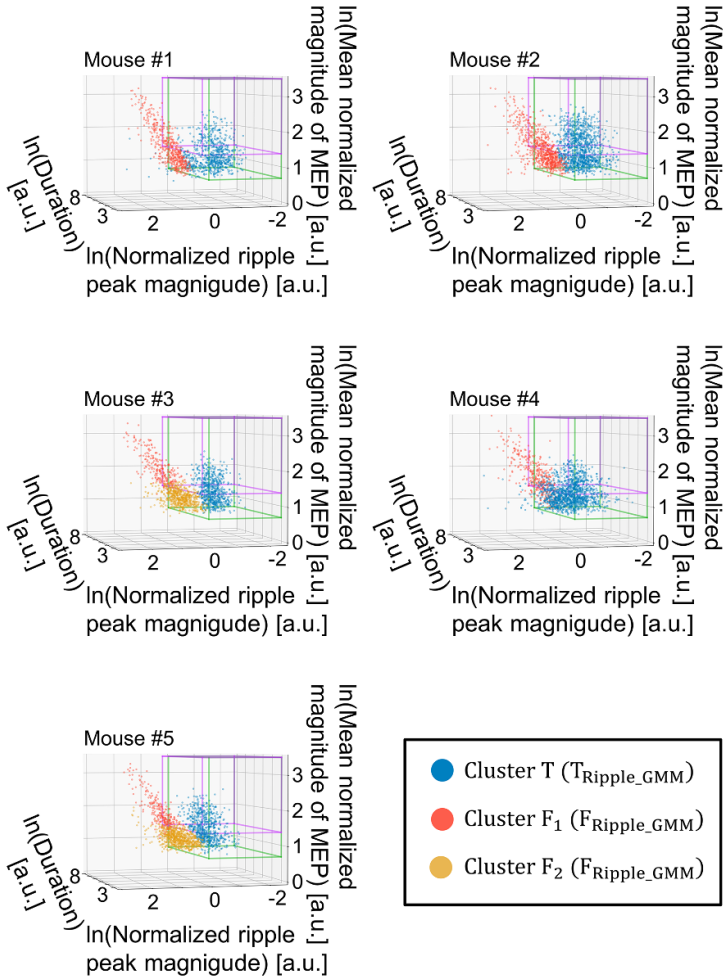

Figure 6. Ripples labeled using Gaussian mixed model clustering. Each plot in the three-dimensional space shows true-ripple-like events (*Blue*; tagged as T_{Ripple_GMM} in Cluster T) and false-ripple-like events (*Red*; tagged as F_{Ripple_GMM} in Cluster F_1/ Yellow; tagged as F_{Ripple_GMM} in Cluster F_2) defined with Gaussian mixed model clustering (GMM). The number of clusters for the GMM clustering was determined using the elbow method for each mouse data: 2, 2, 3, 2, 3 for Mouse #1, #2, #3, #4, and #5, respectively (figure 4). The sample points and green/purple cubes are the same as those in figure 2. Note that the axes of the 3D space coincided with the variables used in the GMM clustering: the natural logarithm of (i) the duration (ms), (ii) mean normalized magnitude of the myoelectricity potential (MEP) of the trapezius (μV), and (iii) normalized ripple peak magnitude (ripple band: 150–250 Hz) (μV) for each ripple (see equations (9)–(22) in section 2). The determined decision boundaries were hyperplane in this three-dimensional space.

or Cluster F_2' are expected to include false ripples. We labeled the ripple candidates assigned to Cluster T as T_{Ripple_GMM} and to Cluster F_1 or Cluster F_2 as F_{Ripple_GMM}, respectively.

3.4. Defining ripples by using our CNN in a weakly supervised manner

Each ripple candidate was labeled as T_{Ripple_GMM} and F_{Ripple_GMM}. However, in general, the labels contained errors, or noise was present. GMM clustering uses only pre-defined and limited information from ripple candidates.

Label errors can be estimated and mitigated by Confident Learning. Specifically, the imported Python module 'cleanlab' (Northcutt *et al* 2019) acts with any classifier to estimate the latent true labels without hyperparameters. We conjectured that 'noisy' labels, T_{Ripple_GMM} and F_{Ripple_GMM}, could be 'cleaned' with the cleanlab module and a CNN because, in such a system, CNN is known to search for optimal timely-local features in original signals, which is difficult for humans to pre-define.

First, we constructed a ResNet-based deep CNN (figure 7; also see section 2). We call this model 'our CNN' in this paper. Because only a fixed-length input is allowed for this CNN (e.g. 400 ms at a sampling rate of 1 kHz in our case), we isolated each ripple candidate and converted it to 400 ms data (figure 8; see section 2).

Next, dataset splitting was performed for defining ripples, leading to ten sub-datasets (figure 9). This splitting aimed to (a) examine whether the difference of datasets influences labels defined with our GMM and Confident Learning system and (b) avoid data leakage in the ripple detection experiment in the next paragraph. Each ripple candidate was tagged with five cleaned labels according to the corresponding five sub-datasets (e.g. ripple candidates acquired from

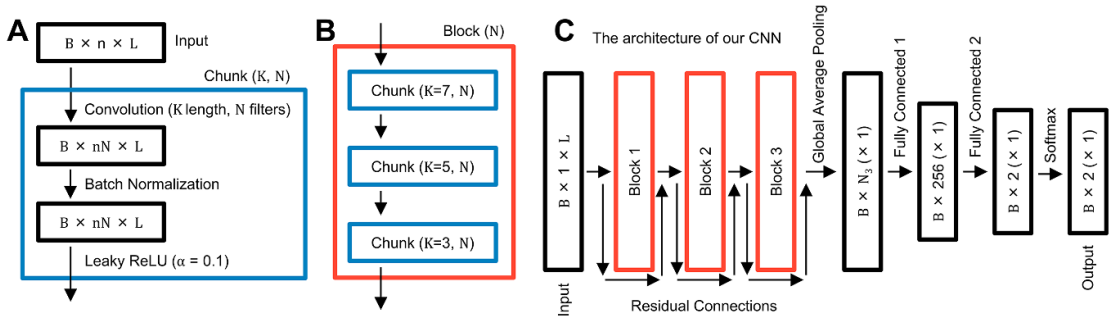

Figure 7. The structure of our CNN. (A) A chunk (Blue) is the smallest unit in our CNN. A chunk has two parameters: K and N. N × convolution, K ((ms) in our use case) length filters were installed in the first layer. B is the mini-batch size, or the sample size in each training loop. In the Batch Normalization layer (Ioffe and Szegedy), the filtered traces are normalized, and perturbations are injected. The output is activated by the Leaky ReLU activation function (Maas et al 2013) with a differential coefficient of 0.1 in the negative region ($\alpha = 0.1$). (B) A block (Red) is the second smallest unit, and it includes three chunks with the parameters K = 7, 5, and 3 in this order. (C) The whole structure of our CNN. To extract local features from input signals blocks 1–3 were connected with the convolution pathway and residual connections. Global average pooling was performed to squash the time dimension of the extracted features. Fully connected layers were adopted as a classification module.

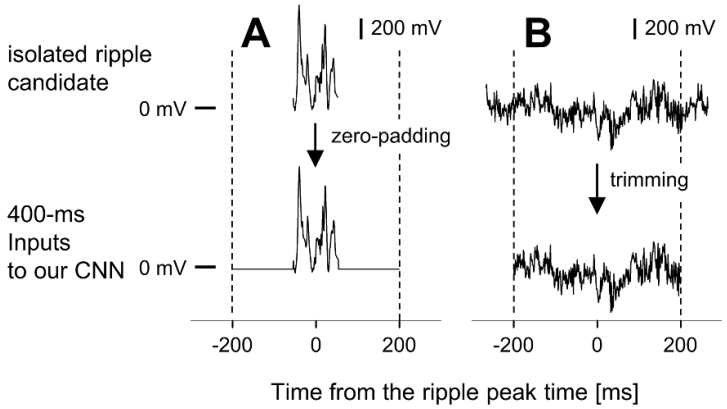

Figure 8. Isolating and trimming each ripple candidate as a preprocessing. (A) When a ripple candidate was 400 ms or less, it was extended with zero-padding to 400 ms. (B) Otherwise, the LFP in the range of (−200 ms, +200 ms) from the peak time of the ripple band filtered signal was extracted (see equation (34) in section 2). These synthesized 400 ms time series were used as inputs to our CNN to define ripples (see equations (33)–(35) in section 2).

mouse #1 was included in sub-datasets $\mathcal{D}_1$, $\mathcal{D}_{-2}$, $\mathcal{D}_{-3}$, $\mathcal{D}_{-4}$, and $\mathcal{D}_{-5}$). Between any pair of sub-datasets, the cleaned labels' concordance rate was more than 0.85 (figure 10). This result suggests that cleaned labels were relatively robust regarding the datasets. Also, the correlations of the predicted probabilities for the T_{Ripple_CNN} group were more than 0.66 (figure 11; $ps < .001$, $n = 14\,104\,250$ ripple candidates, multiple comparisons using the test for no correlation with Bonferroni correction). These results support the robustness of the cleaned labels about the datasets. The determined labels were 3D-scatter-plotted in figure 12.

Next, we checked the properties of the ripple candidates based on predicted subgroups (T_{Ripple_GMM}, F_{Ripple_GMM}, T_{Ripple_CNN} or F_{Ripple_CNN}). For the ripple candidates defined from mouse #1, #2, #3, #4, and #5, the cleaned labels determined using the sub-dataset $\mathcal{D}_{-2}$, $\mathcal{D}_{-3}$, $\mathcal{D}_{-4}$, $\mathcal{D}_{-5}$, and $\mathcal{D}_{-1}$ were considered, respectively (figure 11). We assessed the following four ripple candidate subgroups: ripple candidates tagged as T_{Ripple_GMM} to T_{Ripple_CNN} (T2T), T_{Ripple_GMM} to F_{Ripple_CNN} (T2F), F_{Ripple_GMM} to T_{Ripple_CNN} (F2T), and F_{Ripple_GMM} to F_{Ripple_CNN} (F2F) (figure 13, table 4).

The medians of the 'three variables' were different between any two groups for each mouse (figures 14–16 and table 4; $ps < .001$; ns are indicated in the figures; multiple comparisons using the Brunner-Munzel test with the Benjamini–Hochberg correction after the Kruskal–Wallis test). We also calculated the effect sizes for all the pairs of the above comparisons (table 5) to quantify how substantial the supported differences were, independently of the sample sizes. Here, we adopted Cliff's delta statistic (Cliff 1996) as an effect size index because it does not require any assumption regarding the data distribution nor homoscedasticity between two compared groups.

First, the F2F group differed in terms of duration (tables 4 and 5, annotated as *1). Second, F2T was close to T2T regarding duration and the ripple peak magnitude but the MEP magnitude (tables 4 and 5, annotated as *2). Third, F2T and F2F groups were not

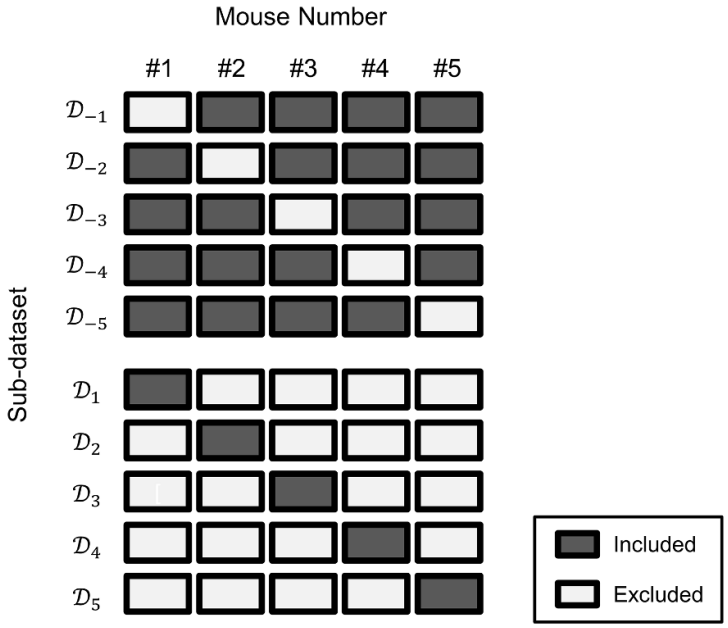

Figure 9. Data splitting for defining and detecting ripple experiments. Data splitting was conducted before Confident Learning to perform the cross-validation method on ripple detection experiments in the later section (figure 16). Sub-dataset $\mathcal{D}_{-1}$, $\mathcal{D}_{-2}$, $\mathcal{D}_{-3}$, $\mathcal{D}_{-4}$, and $\mathcal{D}_{-5}$ did not include the corresponding mouse's data. Sub-dataset $\mathcal{D}_1$, $\mathcal{D}_2$, $\mathcal{D}_3$, $\mathcal{D}_4$, and $\mathcal{D}_5$ consisted of only the corresponding mouse's data. Sub-dataset $\mathcal{D}_1$, $\mathcal{D}_2$, $\mathcal{D}_3$, $\mathcal{D}_4$, and $\mathcal{D}_5$ were used as the test dataset in ripple detection experiments.

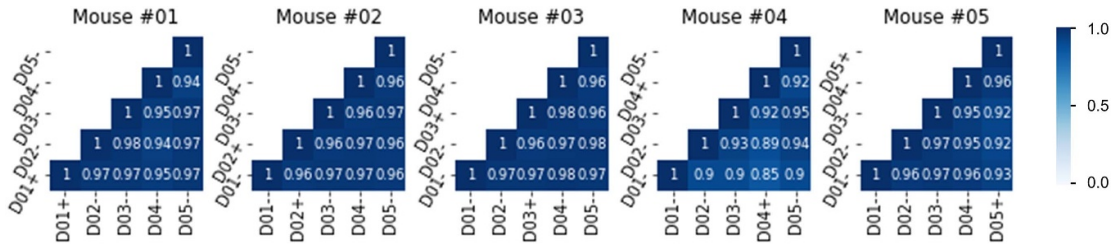

Figure 10. Concordance rates of ripple labels defined with CNNs trained on different datasets. The matrices show the concordance rates of the labels (F_{Ripple_CNN} or T_{Ripple_CNN}) estimated with Confident Learning after GMM clustering. Every ripple candidate was included in five datasets (e.g. a ripple candidate recorded for mouse #1 was included in the sub-dataset $\mathcal{D}_1$, $\mathcal{D}_{-2}$, $\mathcal{D}_{-3}$, $\mathcal{D}_{-4}$, and $\mathcal{D}_{-5}$ as shown in figure 8). Each ripple candidate was classified five times in total as true- ($T_{Ripple\_CNN}|\mathcal{D}$) or false-ripple-like event ($T_{Ripple\_CNN}|\mathcal{D}$) according to the sub-dataset $\mathcal{D}$.

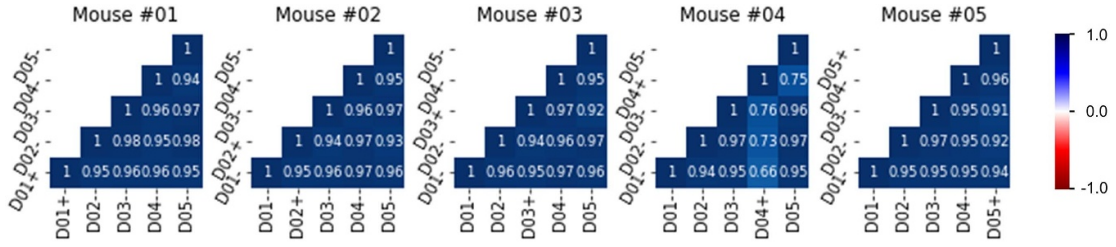

Figure 11. Correlations among estimated probabilities as true ripples based on our CNN and different datasets. The matrices show the Pearson correlation values of predicted probabilities as ripples ($=P\left(\mathbf{c}_i \in T_{Ripple\_CNN}\right)$; see equation (37) between sub-datasets. $\mathbf{c}_i$: the i_cth ripple candidate (see equation (33)), T_{Ripple_CNN}: the label for true ripple estimated with Confident Learning.

close in duration but the ripple peak and MEP magnitude (tables 4 and 5, annotated as *3). Fourth, the ripple peak magnitude was not a dominant factor; the effect sizes were judged as small or medium (tables 4 and 5, annotated as *4).

In summary, the T2T and F2T group reflects the phenomenon of hippocampal ripples, and the proposed method succeeded in defining ripples even during movement at a stochastic scale.

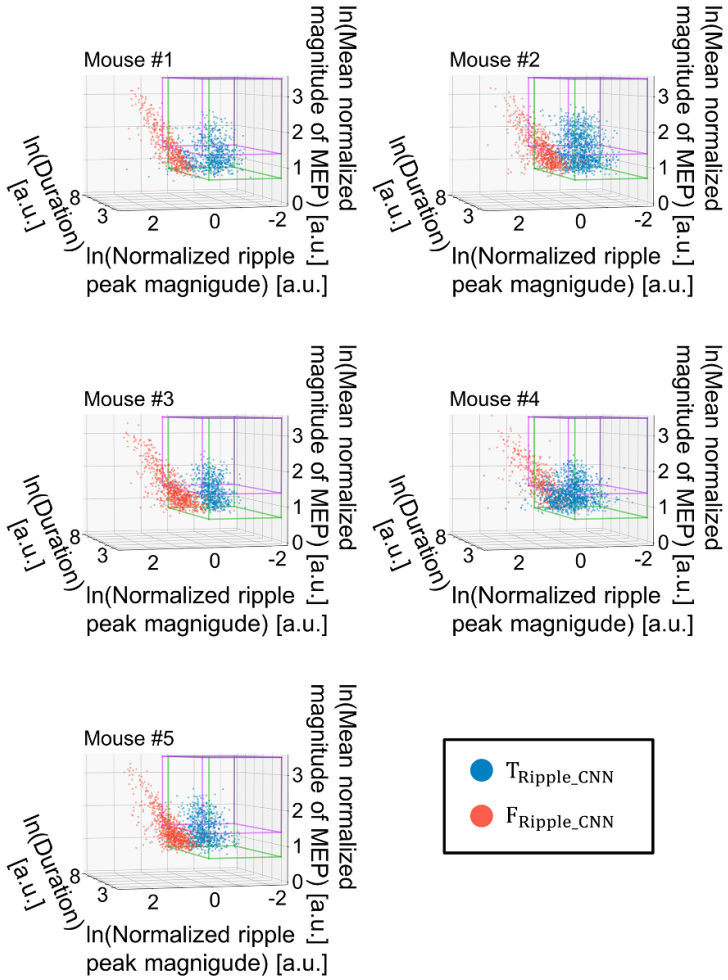

Figure 12. Ripples defined using our CNN. Each plot in the 3D space shows a true-ripple-like event (*Blue*; labeled T_{Ripple_CNN}) or false-ripple-like event (*Red*; labeled F_{Ripple_CNN}) defined with Condient Learning after using our CNN trained on T_{Ripple_GMM}/F_{Ripple_GMM} labels (tags meaning true/false ripples defined with Gaussian mixture clustering). The scatter-plotted samples, and green/purple cubes are the same as those in figures 2 and 5. Note that the input to our CNN was not the three 'ln-variables', as in the case of the Gaussian mixed model clustering (figure 5), but a 400 ms LFP raw trace. Also, the decision boundaries in this 3D space were not hyperplanes, unlike those with the Gaussian clustering.

3.5. Detecting ripples by using our CNN

Next, we determined whether our CNN detects 'ripples' from unseen LFP signals. Here, 'ripples' means ripples defined using our method, GMM clustering plus Confident Learning using our CNN. Specifically, we quantified the ability of our CNN to classify 400 ms raw LFP signals into two classes: one-false-ripple-including group (F) and one-true-ripple-including group (T) (figure 17(A)). To evaluate the results, we used the leave-one-animal-out cross-validation method (figure 17(B)).

First, we defined the two groups: F and T. F/*T* group consists of 400 ms raw LFP signals, each of which included just one event tagged as $F_{Ripple\_CNN}/T_{Ripple\_CNN}$. For simplicity, we excluded 400 ms raw LFP signals that included any part of more than two ripples from this experiment.

The classification results are shown in table 6 and figure 18. The area under the precision-recall curve and receiver operating curve were 0.986 ± 0.078 and 0.955 ± 0.010, respectively (n = 5 mice, Mean ± Std.). When the decision threshold ($d_{threshold\_detect}$; figure 17(A); equation (38)) was fixed as 0.5, confusion matrices were obtained (figure 18). The class-balanced accuracy was 0.882 ± 0.043. F1-score was 0.869 ± 0.070. Because these scores exceeded their chance level, our trained CNN was validated.

4. Discussion

Our method first defined 'noisy ripples' using GMM clustering, based on our large dataset and a low-human-biased version of an existing ripple detection method. It does not need pre-defined thresholds like the ripple band peak magnitude. After that, we used CNN to capture ripples' temporally local features by combining the latent label finding system (Confident Learning).

By using our method, the ripples during resting states and the ones during active conditions were defined.

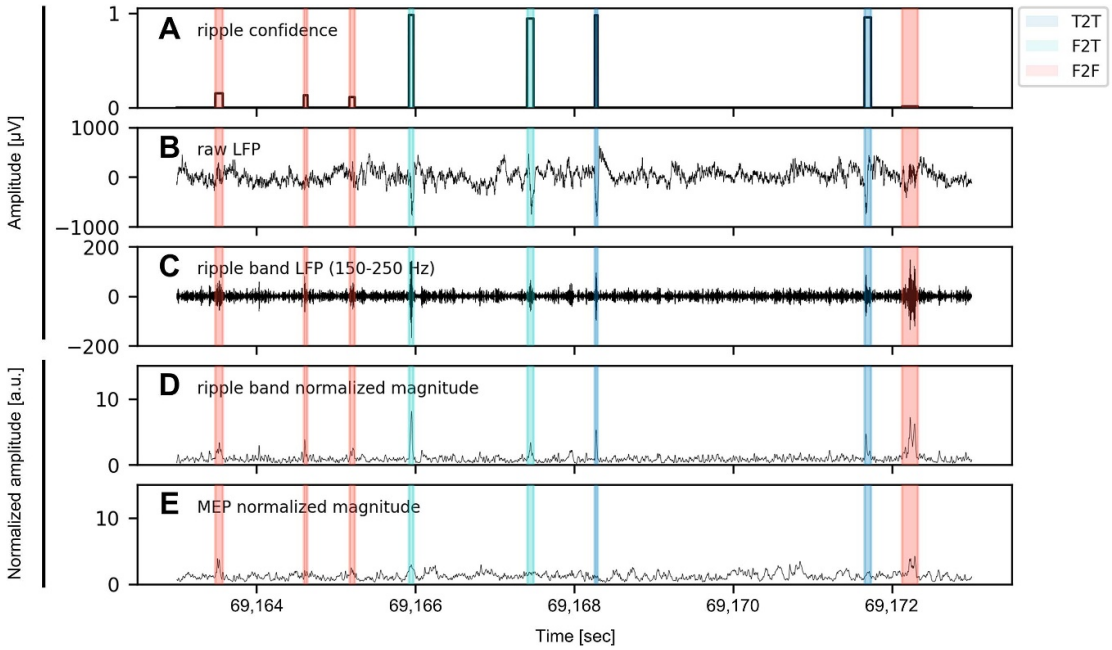

Figure 13. Representative traces of ripples defined by our CNN. (A) Confidence (predicted probabilities) for ripples determined by CNN in Confident Leaning. (B) Hippocampal raw LFP trace. (C) The corresponding trace of the hippocampal ripple band LFP (150–250 Hz), normalized ripple band magnitude (D) and normalized MEP of the trapezius (E) Colored rectangles show the three groups of ripple candidates: Blue, T2T group (T_{Ripple_GMM} to T_{Ripple_CNN}); Light Blue, F2T group (F_{Ripple_GMM} to T_{Ripple_CNN}); Red, F2F group (F_{Ripple_GMM} to F_{Ripple_CNN}). T_{Ripple_GMM}/F_{Ripple_GMM}: true or false ripples defined with GMM clustering. T_{Ripple_CNN}/F_{Ripple_CNN}: true or false ripples defined with CNN in Confident Learning stage. Note that T2T and F2T group were determined as high ripple confidence and their raw traces were supercomposed with downward sharp waves. On the other hand, F2F group was accompanied with transient increase of the MEP magnitude.

Table 4. Medians and median absolute deviation of the 'three variables' for determined ripple groups. The table shows the median and median absolute deviation for four defined groups regarding the 'three variables: duration (ms), normalized ripple peak magnitude, and mean normalized magnitude of MEP for each ripple candidate.

	T2T[*2]	F2T[*2,3]	T2F	F2F[*3]
Duration (ms)	66 ± 1	67 ± 2	67 ± 9	112 ± 17[*1]
MEP magnitude (SD) [*4]	0.58 ± 1.13	1.51 ± 1.24	0.84 ± 1.13	1.96 ± 1.06
Ripple peak magnitude (SD)	3.99 ± 1.06	3.46 ± 1.06	3.02 ± 1.06	3.73 ± 1.03

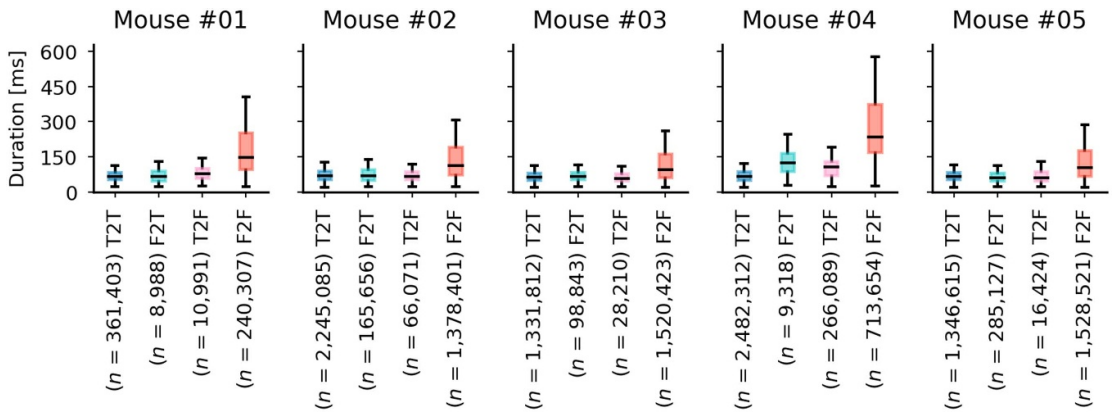

Figure 14. The distributions of the durations among the ripple candidate subgroups. Each panel shows the boxplot (without outliers) of the duration of ripple candidates for each mouse. Ripple candidates were classified into four groups based on GMM clustering and Confident Learning using CNN: T2T (T_{Ripple_GMM} to T_{Ripple_CNN}; *Blue*), F2T (F_{Ripple_GMM} to T_{Ripple_CNN}; *Light Blue*), T2F (T_{Ripple_GMM} to F_{Ripple_CNN}; *Pink*), F2F (F_{Ripple_GMM} to F_{Ripple_CNN}; *Red*). In all four groups for all mouse, the differences in medians were statistically significant ($ps < .01$; Brunner-Munzel test with the Bonferroni correction after Kruskal–Wallis test). Cliff's delta statistic, a effect size, for all pairs of the four groups are shown in table 4.

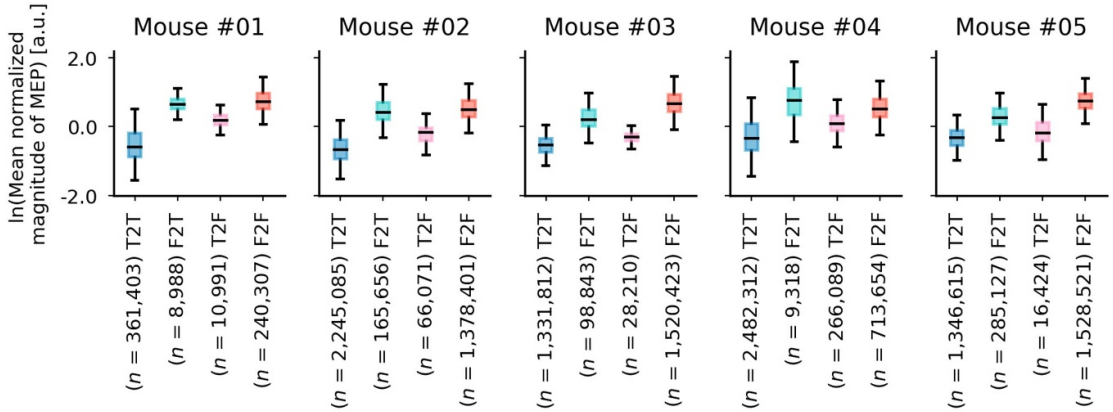

Figure 15. The distributions of the mean normalized magnitude of the MEP among ripple candidate subgroups. Each panel shows the boxplot (without outliers) of the mean normalized magnitude of the MEP of ripple candidates for each mouse. Ripple candidates were classified into four groups based on GMM clustering and Confident Learning using CNN: T2T (T_{Ripple_GMM} to T_{Ripple_CNN}; *Blue*), F2T (F_{Ripple_GMM} to T_{Ripple_CNN}; *Light Blue*), T2F (T_{Ripple_GMM} to F_{Ripple_CNN}; *Pink*), F2F (F_{Ripple_GMM} to F_{Ripple_CNN}; *Red*). In all four groups for all mouse, the differences in medians were statistically significant ($ps < .01$; Brunner-Munzel test with the Bonferroni correction after Kruskal–Wallis test). Cliff's delta statistic, a effect size, for all pairs of the four groups are shown in table 4.

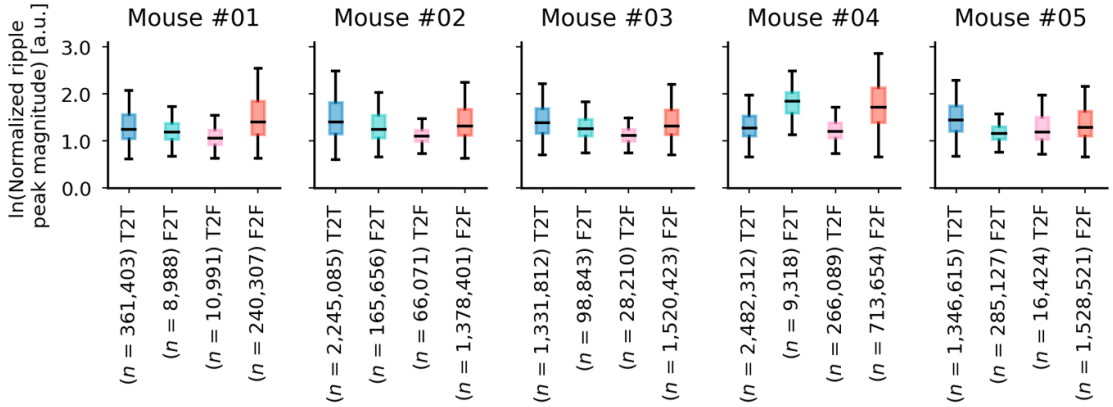

Figure 16. The distributions of the normalized ripple peak magnitudes among ripple candidate subgroups. Each panel shows the boxplot (without outliers) of the normalized ripple peak magnitudes of ripple candidates for each mouse. Ripple candidates were classified into four groups based on GMM clustering and Confident Learning using CNN: T2T (T_{Ripple_GMM} to T_{Ripple_CNN}; *Blue*), F2T (F_{Ripple_GMM} to T_{Ripple_CNN}; *Light Blue*), T2F (T_{Ripple_GMM} to F_{Ripple_CNN}; *Pink*), F2F (F_{Ripple_GMM} to F_{Ripple_CNN}; *Red*). In all four groups for all mouse, the differences in medians were statistically significant ($ps < .01$; Brunner-Munzel test with the Bonferroni correction after Kruskal–Wallis test). Cliff's delta statistic, a effect size, for all pairs of the four groups are shown in table 4.

In the ripple detection stage, only one-electrode-derived LFP data is used. The output of our CNN is the predicted probability for including a ripple. If an experiment needs to tune the precision-recall tradeoff for ripple detection, just changing the threshold for the predicted probabilities is enough after once calculating the probabilities.

The method in this work mitigates the issue of inconsistent definitions of hippocampal ripples.

First, we showed that setting thresholds for animal speed, ripple peak magnitude, or ripple duration one by one did not fit the candidates' distribution. GMM clustering removes the implicit assumption of independence between the variables used in previous ripple detectors and alleviates the problem by optimizing the boundary.

Then, we adopted a CNN to capture local features from raw LFP, which are characteristic of ripples. A CNN has multiple learnable filters and optimizes them to solve a given task. It is expected that the more variety and quality the dataset gets, the more generalized definition of ripples will be determined.

The proposed method captured ripples even during moving. Concerning duration, the ripples defined by our method (T_{Ripple_CNN} or T2T and F2T group) were closer to those defined by a multielectrode method in previous studies (Sullivan *et al* 2011, Buzsáki 2015). Additionally, concerning the MEP magnitude of the trapezius, the F2T group was larger than the T2T group. Thus, the proposed method detects ripples during both stable and moving states, the latter of which is avoided with existing methods.

Table 5. Absolute Cliff's delta statistics for the 'three variables.' The table shows the absolute Cliff's delta values among T2T, F2T, T2F and F2F groups regarding the 'three variables: duration (ms), normalized ripple peak magnitude (μV), and mean normalized magnitude of MEP (μV) for each ripple candidate (Mean $\pm$ Std.; $ns = 621\,589$; 3855 213; 2979 288; 3471 373; and 3176 687 ripple candidates for Mouse #1, #2, #3, #4, and #5). Note that Cliff's delta statistic ranges from -1 to 1, and the absolute value indicates the difference; specifically, < 0.147 is negligible, < 0.330 is small, < 0.474 is medium, and $\geqslant 0.474$ is large (Romano 2006). *1; F2T and T2T groups were close in duration and the ripple peak magnitude but the MEP magnitude. *2; F2T and F2F groups were not close in duration but the ripple peak magnitude and the MEP magnitude. *3; The effect size of the ripple peak magnitude was determined as small or medium; it was not a dominant factor.

Variable					
Duration		T2T	F2T	T2F	F2F
	T2T	—	0.168 ± 0.254 (small) *2	0.180 ± 0.152 (small)	0.610 ± 0.212 (large) *1
	F2T	—	—	0.147 ± 0.099 (small)	0.549 ± 0.126 (large) *1,3
	T2F	—	—	—	0.593 ± 0.160 (large) *1
	F2F	—	—	—	—
MEP magnitude		T2T	F2T	T2F	F2F
	T2T	—	0.901 ± 0.071 (large) *2	0.497 ± 0.162 (large)	0.939 ± 0.074 (large)
	F2T	—	—	0.757 ± 0.138 (large)	0.329 ± 0.214 (small) *3
	T2F	—	—	—	0.836 ± 0.149 (large)
	F2F	—	—	—	—
Ripple peak magnitude *4		T2T	F2T	T2F	F2F
	T2T	—	0.327 ± 0.180 (small) *2	0.385 ± 0.138 (medium)	0.206 ± 0.171 (small)
	F2T	—	—	0.364 ± 0.189 (medium)	0.193 ± 0.113 (small) *3
	T2F	—	—	—	0.457 ± 0.157 (medium)
	F2F	—	—	—	—

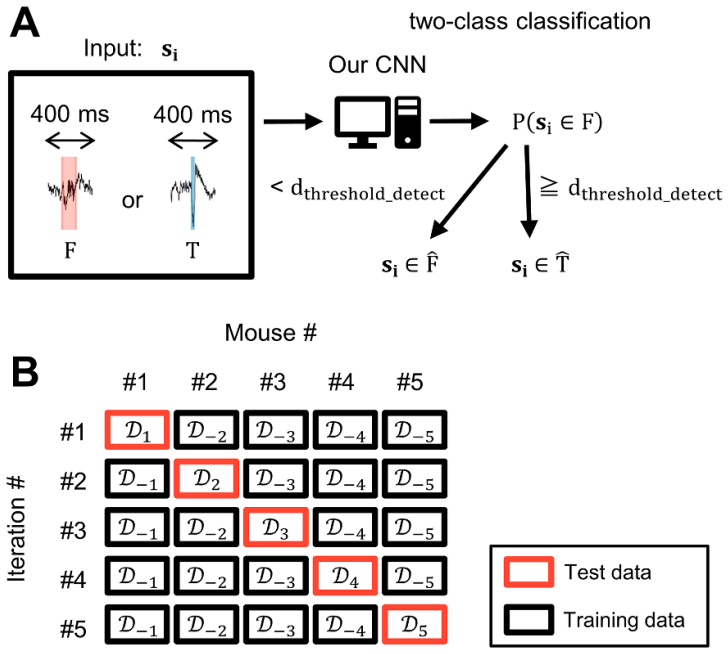

Figure 17. Experimental scheme for detecting ripples. *A*. The experimental design for ripple detection using our CNN. Our CNN requires the length of each input signal vector $\mathbf{s}_i$ to be fixed (e.g. 400 points, or 400 ms for a 1 kHz sampling rate in this study). Input signals $\mathbf{s}_i$ ($i = 1, 2, \ldots$) were separated into the false-ripple- including group (F) or the true-ripple-including group (T): Each signal in F group included a false ripple candidate and that in T group included a true ripple. The output of our CNN, $P(\mathbf{s}_i \in T)$, was interpreted as the probability that the ith input signal $\mathbf{s}_i$ to belonged to T group, which was binarized based on a decision threshold ($d_{threshold\_detect}$) when necessary. *B*. Data splitting was performed based on the leave-one-animal-out cross-validation method. Note that, as shown in the panel, to detect ripples from Mouse#k's signals, test and training labels were derived from the sub-dataset $\mathcal{D}_k$ and $\mathcal{D}_{-k}$, respectively (see also figure 9).

Some studies report that ripples occur even when animals feed or walk (Buzsáki 2015). Our approach offers a way to define/detect ripples during active states with a fully automated way from just a single electrode in the hippocampus.

Our approach can handle 'small' ripples. Ideally, if ripples are sufficiently observed and the volume conductance theory is true, the ripple peak magnitude will be continuous from 0 μV. The proposed method defines ripples probabilistically, without

Table 6. Scalar metrics for detecting ripples in the leave-one-animal-out cross-validation. Metrics dependent on decision thresholds ($\mathbf{d}_{\mathbf{threshold\_detect}}$; figure 17(A); equation (38)) were calculated when they were fixed as 0.5. F/T group consists of 400 ms raw LFP signals each of which included just one event tagged as $F_{\mathrm{Ripple\_CNN}}/\mathrm{T}_{\mathrm{Ripple\_CNN}}$, bACC: class-balanced accuracy, PRE-REC AUC: the area under the precision-recall curve, PRE-ROC AUC: the area under the receiver operating characteristic curve.

Dependence on decision threhsolds		Dependent (when $\mathbf{d}_{\mathbf{threshold\_detect}} = 0.5$)			Independent		
Test mouse #		False Ripple (F)	True Ripple (T)	Weighted Avg.	bACC	PRE-REC AUC (Macro Avg.)	ROC AUC (Macro Avg.)
#1	precision	0.594	0.998	0.917	0.912	0.896	0.953
	recall	0.992	0.832	0.863			
	f1-score	0.743	0.907	0.874			
	sample size	43 823	176 337	220 160			
#2	precision	0.771	0.970	0.924	0.912	0.918	0.958
	recall	0.906	0.918	0.915			
	f1-score	0.833	0.943	0.918			
	sample size	307 187	1010 701	1317 888			
#3	precision	0.933	0.934	0.934	0.920	0.969	0.973
	recall	0.874	0.966	0.934			
	f1-score	0.903	0.950	0.933			
	sample size	373 834	687 030	1060 864			
#4	precision	0.188	0.998	0.933	0.806	0.747	0.944
	recall	0.982	0.630	0.658			
	f1-score	0.315	0.773	0.736			
	sample size	98 251	1129 525	1227 776			
#5	precision	0.980	0.846	0.900	0.861	0.950	0.949
	recall	0.731	0.990	0.886			
	f1-score	0.838	0.912	0.882			
	sample size	428 914	640 142	1069 056			
Mean ± Std.	precision	0.693 ± 0.287	0.949 ± 0.057	0.922 ± 0.012	0.882 ± 0.043	0.986 ± 0.078	0.955 ± 0.010
	recall	0.897 ± 0.094	0.867 ± 0.130	0.851 ± 0.100			
	f1-score	0.726 ± 0.212	0.897 ± 0.064	0.869 ± 0.070			
	sample size	250 402 ± 152 414	728 747 ± 333 127	979 149 ± 391 761			

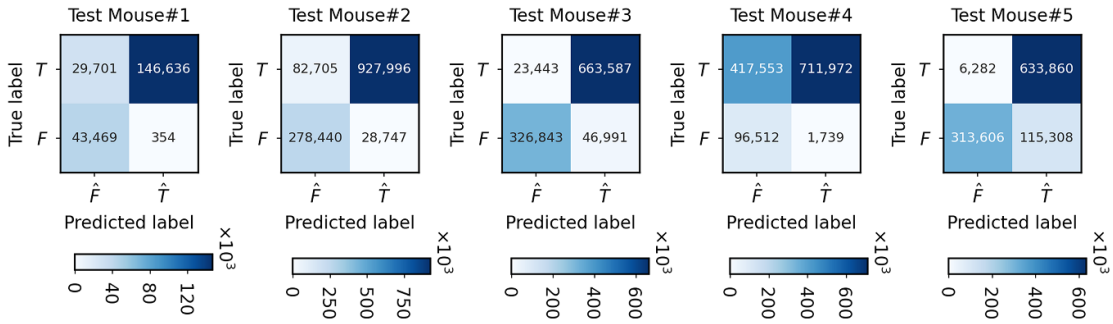

Figure 18. Confusion matrices for ripple detection. The panels show confusion matrices for binary classification of False ripple group (F) vs True ripple group (T), when the binarization threshold was fixed to 0.5 (see figure 17 and equation (41) in section 2). Each panel shows the confusion matric for each fold in the leave-one-animal-out cross-validation method.

sorting out small ripples. Predicted binary labels are obtained by adjusting the cut-off threshold on the ripple probability after all calculation has finished.

The capacity of handling small ripples leads to the robustness of detecting ripples at the recording site in the hippocampus and extending the time limitation for continuously detecting ripples. LFP amplitude is inversely proportional to the distance between the virtual origin and the recording sites (Buzsáki *et al* 2012). Also, it is known that recording electrodes tend to 'slip', especially in the electrode's longitudinal direction under the free moving condition, at least because of the elasticity of the brain.

Under these settings, our method can contribute to detect ripples longer, especially in experiments at day-to-week scales. This quality enhancement of ripple detection in stabilization and prolongation will be ideal for revealing the more detailed relationships between hippocampal ripples and memory consolidation or long-term memory.

In this study, we did not distinguish awake/sleep states. It is known that awake ripples are faster than quiet ripples. For example, in rats, the modal frequency of SPW-Rs has been reported as 167 Hz during slow-wave non-REM sleep, 177 Hz during sleep in the home cage, 187 Hz during immobility, and drinking in the maze (Buzsáki 2015). We defined ripple candidates using a ripple-bandpass filter (150–250 Hz). Also, CNN filters could learn such frequency components, as discussed in the next paragraph.

From the viewpoint of time-frequency analysis, the lengths of the convolution filters (Ks), hyperparameters of our CNN, were related to the frequency bands' limits to extract features.

Our CNN extracted temporally local features over consecutive 36 ms. This is determined by the following settings. In our settings, 7-, 5-, and 3-point convolution filters were included in each of three blocks of our CNN (Ks = 7, 5, 3; figure 7), and the effective sampling rate of the hippocampal LFP was 1 kHz. Thus, it is regarded that our CNN includes at least 36-degree filters as a whole.

It is possible that low-frequency components, which are difficult to express in 36 ms, may not be treated by our CNN. Ripples (150–250 Hz) often overlap with sharp waves (5–50 Hz; O'Keefe 1976, Buzsaki 2015, Watanabe 2017) and gamma waves (25–75 Hz; Ramirez-Villegas *et al* 2015).

It may be helpful to test whether low-frequency sine waves are learned. If these low-frequency components are not utilized, the following three modifications might be helpful: (a) to enlarge each filter length, (b) to use the Inception module (Szegedy *et al* 2015), or (c) like the SincNet (Ravanelli *et al* 2019) to add a relatively long bandpass filter to the first layer of the neural network.

Our method has a limitation; for simplicity, it is assumed that there is not more than one ripple event within a 400 ms LFP input sequence for our CNN in the detection step. However, this limitation is not practical. Ripples often appear in clusters (i.e. multiple events with less than 100 ms intervals between them). The proportion of clustered ripples is approximately 50% during brief pauses and approximately 20% during prolonged periods of immobility while awake or during periods of sleep in home cages (Buzsáki 2015).

Our CNN to detect ripples should be modified to detect when and how many ripples are in a time series. This refinement might be realized by mimicking object detection models in 2D or 3D. Specifically, the faster RCNN model (Ren *et al* 2015) or the CenterNet model (Zhou *et al* 2019) may be the candidates.

When using the 1D modified versions of these models, it might be needed to consider the balance between the convolutional filters' length (Ks) and the frequency band limits to extract local features (i.e. 150–250 Hz), because of the time-frequency tradeoff (the Gabor uncertainty).

Lastly, this work suggests that ripples are identified with local features using CNN, or 'shape.' Ripples are known to have different propagation pathways regarding their shape (Ramirez-Villegas *et al* 2015). Also, ripple shape is associated with spiking sequence

patterns (Taxidis *et al* 2015). Moreover, ripples seem to convey content-specific information (Norman *et al* 2019). As our method does not need pre-defined features and can learn features from LFP, the proposed method might distinguish ripple subtypes regarding the shape or even function. All we need may be the labels with a proper time resolution and appropriate fine-tuning scheduling.

Data availability statement

The data generated and/or analysed during the current study are not publicly available for legal/ethical reasons but are available from the corresponding author on reasonable request.

Acknowledgments

This work was supported by JST Exploratory Research for Advanced Technology (ERATO; JPMJER1801) and JSPS Grants-in-Aid for Scientific Research (18H05525). Y W and Y I conceptualized the study; M O performed the experiments; Y W performed the data analysis; Y W wrote the original draft; and all authors reviewed and edited the final manuscript.

Conflict of interest

The authors declare no conflict of interest.

ORCID iDs

Yusuke Watanabe https://orcid.org/0000-0001-9541-6073
Yuji Ikegaya https://orcid.org/0000-0003-2260-8191

References

Buzsáki G 2015 Hippocampal sharp wave-ripple: a cognitive biomarker for episodic memory and planning *Hippocampus* **25** 1073–188
Buzsáki G, Anastassiou C A and Koch C 2012 The origin of extracellular fields and currents—EEG, ECoG, LFP and spikes *Nat. Rev. Neurosci.* **13** 407–20
Buzsáki G, Horvath Z, Urioste R, Hetke J and Wise K 1992 High-frequency network oscillation in the hippocampus *Science* **256** 1025
Cliff N 1996 Answering ordinal questions with ordinal data using ordinal statistics *Multivariate Behav. Res.* **31** 331–50
Ego-Stengel V and Wilson M A 2010 Disruption of ripple-associated hippocampal activity during rest impairs spatial learning in the rat *Hippocampus* **20** 1–10
Fawaz H I, Forestier G, Weber J, Idoumghar L and Muller P-A 2019 Deep learning for time series classification: a review *Data Min. Knowl. Discovery* **33** 917–63
Fernández-Ruiz A, Oliva A, De Oliveira E F, Rocha-Almeida F, Tingley D and Buzsáki G 2019 Long-duration hippocampal sharp wave ripples improve memory *Science* **364** 1082–6
Fukushima K 1980 Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position *Biol. Cybern.* **36** 193–202
Girardeau G, Benchenane K, Wiener S I, Buzsáki G and Zugaro M B 2009 Selective suppression of hippocampal ripples impairs spatial memory *Nat. Neurosci.* **12** 1222–3
Hahnloser R H R, Sarpeshkar R, Mahowald M A, Douglas R J and Seung H S 2000 Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit *Nature* **405** 947–51
He K, Zhang X, Ren S and Sun J 2016 Deep residual learning for image recognition *2016 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (Las Vegas, NV: IEEE) pp 770–8 (available at: http://ieeexplore.ieee.org/document/7780459/) (Accessed 24 January 2020)
Ichihara K 1990 *Statistics for Bioscience: Practical Technique and Theory* (Tokyo: Nankodo Co., Ltd)
Ioffe S and Szegedy C 2015 Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift *Proc. of the 32nd International Conf. on Machine Learning, PMLR* **37** 448–56
Karlsson M P and Frank L M 2009 Awake replay of remote experiences in the hippocampus *Nat. Neurosci.* **12** 913–8
Kay K, Sosa M, Chung J E, Karlsson M P, Larkin M C and Frank L M 2016 A hippocampal network for spatial coding during immobility and sleep *Nature* **531** 185–90
Krizhevsky A, Sutskever I and Hinton G E 2012 ImageNet classification with deep convolutional neural networks *Advances in Neural Information Processing Systems* vol 25, ed F Pereira, C J C Burges, L Bottou and K Q Weinberger (Curran Associates, Inc.) pp 1097–105 (available at: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf) (Accessed 24 January 2020)
Li X, Chen S, Hu X and Yang J 2019 Understanding the disharmony between dropout and batch normalization by variance shift *2019 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)* pp 2677–85
Maas A L, Hannun A Y and Ng A Y 2013 Rectifier nonlinearities improve neural network acoustic models *Proc. icml* p 3
Norimoto H, Makino K, Gao M, Shikano Y, Okamoto K, Ishikawa T, Sasaki T, Hioki H, Fujisawa S and Ikegaya Y 2018 Hippocampal ripples down-regulate synapses *Science* **359** 1524–7
Norman Y, Yeagle E M, Khuvis S, Harel M, Mehta A D and Malach R 2019 Hippocampal sharp-wave ripples linked to visual episodic recollection in humans *Science* **365** eaax1030
Northcutt C G, Jiang L and Chuang I L 2019 Confident learning: estimating uncertainty in dataset labels (arXiv:191100068) [cs, stat] (Accessed 24 January 2020)
O'Keefe J 1976 Place units in the hippocampus of the freely moving rat *Exp. Neurol.* **51** 78–109
Ramirez-Villegas J F, Logothetis N K and Besserve M 2015 Diversity of sharp-wave-ripple LFP signatures reveals differentiated brain-wide dynamical events *Proc. Natl Acad. Sci. USA* **112** E6379–87
Ravanelli M and Bengio Y 2019 Speaker recognition from raw waveform with SincNet (arXiv:180800158) [cs, eess] (Accessed 24 January 2020)
Ren S, He K, Girshick R and Sun J 2015 Faster R-CNN: towards real-time object detection with region proposal networks *Advances in Neural Information Processing Systems* vol 28, ed C Cortes, N D Lawrence, D D Lee, M Sugiyama and R Garnett (Curran Associates, Inc.) pp 91–99 (available at: http://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networks.pdf) (Accessed 24 January 2020)
Romano J, Kromrey J D, Coraggio J and Skowronek J 2006 Appropriate statistics for ordinal level data: should we really be using t-test and Cohen'sd for evaluating group differences on the NSSE and other surveys *Annual Meeting of the Florida Association of Institutional Research* pp 1–33
Shin J D, Tang W and Jadhav S P 2019 Dynamics of awake hippocampal-prefrontal replay for spatial learning and memory-guided decision making *Neuron* **104** 1110–25.e7

Srivastava N, Hinton G, Krizhevsky A, Sutskever I and Salakhutdinov R 2014 Dropout: a simple way to prevent neural networks from overfitting *J. Mach. Learn. Res.* **15** 1929–58

Sullivan D, Csicsvari J, Mizuseki K, Montgomery S, Diba K and Buzsáki G 2011 Relationships between hippocampal sharp waves, ripples, and fast gamma oscillation: influence of dentate and entorhinal cortical activity *J. Neurosci.* **31** 8605–16

Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V and Rabinovich A 2015 Going deeper with convolutions *2015 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* 1–9

Taxidis J, Anastassiou C A, Diba K and Koch C 2015 Local field potentials encode place cell ensemble activation during hippocampal sharp wave ripples *Neuron* **87** 590–604

Tong Q, Liang G and Bi J 2019 Calibrating the adaptive learning rate to improve convergence of ADAM (arXiv:190800700) [cs, math, stat] (Accessed 24 January 2020)

Watanabe Y and Ikegaya Y 2017 Caffeine increases hippocampal sharp waves *in vitro Biol. Pharm. Bull.* **40** 1111–5

Wirtshafter H S and Wilson M A 2019 Locomotor and hippocampal processing converge in the lateral septum *Curr. Biol.* **29** 3177–3192.e3

Wu C-T, Haggerty D, Kemere C and Ji D 2017 Hippocampal awake replay in fear memory retrieval *Nat. Neurosci.* **20** 571–80

Zhang M R, Lucas J, Hinton G and Ba J 2019 Lookahead optimizer: k steps forward, 1 step back *NIPS'19: Proc. of the 33rd International Conf. on Neural Information Processing Systems* 9597–608

Zhou X, Wang D and Krähenbühl P 2019 Objects as points (arXiv:190407850) [cs] (Accessed 24 January 2020)