Bayesian inference of spike-time dependent learning rules from single neuron recordings in humans

Ingeborg Gullikstad Hem[1*], Debora Ledergerber[2], Claudia Battistin[3], Benjamin Dunn[1]

1 Department of Mathematical Sciences, NTNU, Norway
2 Swiss Epilepsy Center, Klinik Lengg AG, Switzerland
3 Simula Consulting, Norway

* ingeborg.hem@ntnu.no

Abstract

The spike-timing dependent plasticity (STDP) learning rules are popular in both neuroscience and computer algorithms due to their ability to capture the change in neural connections arising from the correlated activity of neurons. Recent advances in recording technology have made large neural recordings common, but to date, there is no established method for inferring unobserved neural connections and learning rules from them. We use a Bayesian framework and assume neural spike recordings follow a binary data model to infer the connections and their evolution over time from data using STDP rules. We test the resulting method on one simulated and one real case study, where the real case study consists of human electrophysiological recordings. The simulated case study allows validation of the model, and the real case study shows that we are able to infer learning rules from awake human data that align with experimental results in rats and other animals. The real case study also uncovered a trend in maximum synaptic modification and age, however, this will need testing on a larger scale to draw conclusions.

Author summary

In this work, we use single neuron recordings from humans to model plasticity through spike-timing dependent plasticity (STDP) learning. In humans, the learning rules are studied *ex vivo* and *in vitro*, but not *in vivo*, and even though large neural recordings are easily available, to date there is no established method for inferring unobserved learning rules from them. We present a Bayesian framework for modelling functional connectivity and plasticity in awake humans using STDP learning rules. The human spike recordings are assumed to follow a Bernoulli process, and we model how the connection from a pre-synaptic neuron to a post-synaptic neuron evolve in time. We tested the resulting method on one simulated and one real case study consisting of human spike recordings. With the simulated case study, we verified that the Bayesian model can infer the parameters of the learning rule and returns the true (simulated) values. The real case study shows that we are able to infer learning rules from human data, and we obtain results in line with experimental results in rats and other animals.

Introduction

The temporal evolution of neural connectivity, or synaptic plasticity, is considered to be the basis for learning and memory. Uncovering the rules behind the temporal evolution

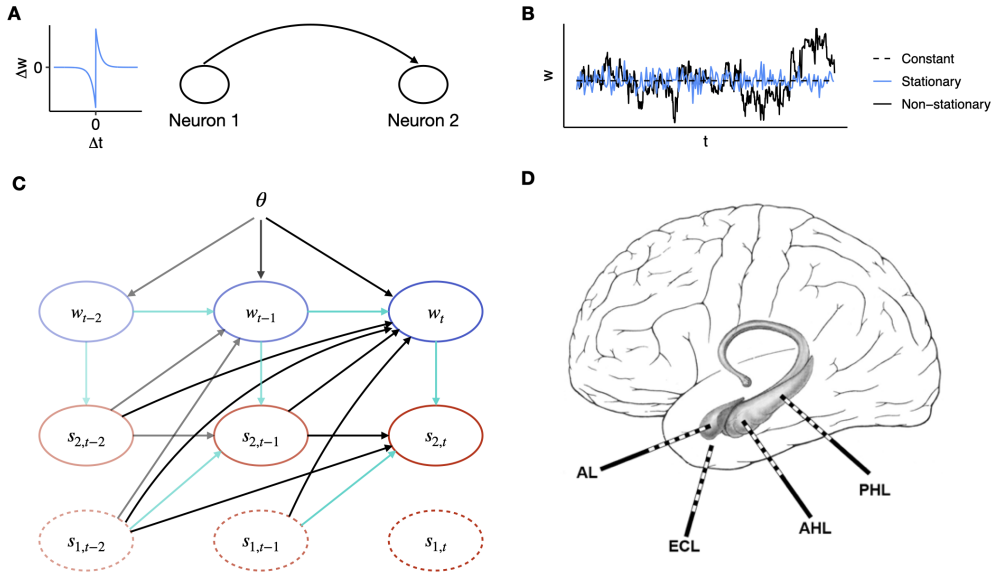


Fig 1. Graphical description of learning rules and model used for inference. **A)** An example of the STDP learning rule (left) for a connection from neuron 1 to neuron 2 (right). The STDP graph shows the effect on the synaptic weight Δw is the change in the synaptic weight when the time difference between pre- and postsynaptic spikes is Δt. **B)** The connection w_t in time from neuron 1 to neuron 2 can be modelled as constant, stationary or non-stationary. **C)** We assume the value of the connection w_t at time t is depending on the previous connection value (w_{t-1}), the previous spiking from both neurons ($s_{1,t'}$ and $s_{2,t'}$ for $t' < t$) and unknown model parameters $\boldsymbol{\theta}$ we want to infer. Black lines represent dependencies through the learning rule, blue lines represent dependencies through the likelihood model. **D)** We apply the model to human data recorded from the hippocampal head (AH) and body (PH), entorhinal cortex (EH) and amygdala (A). Figure shows the left (L) side of the brain.

of neural connectivity can provide valuable insights into the normal functioning of memory processes. Furthermore, this understanding can be used to draw inferences about the disruptions in these processes that occur in neurological disorders such as Alzheimer's disease [1–4] and epilepsy [5–7], which are often characterized by memory loss [8].

The spike-timing dependent plasticity (STDP) learning rule [9–11] can be used to describe the evolution of neural connectivity in time. It has been widely studied *in vitro* and *ex vivo* in a range of brain areas and species, including humans [12], and the relevance of them has been assessed *in vivo* [13–15]. In the STDP rule, both the sign and amount of change in connectivity depend on the relative timing of spiking neurons [11], and yields non-stationary connections. The STDP learning rules drive the evolution of the connection w_t from one neuron to another (Fig 1A). Δw is the change in the synaptic weight when the time difference between pre- and postsynaptic spikes is Δt. Synaptic depression will result if the postsynaptic spike occurs before the presynaptic spike ($\Delta t < 0$) and vice versa for synaptic facilitation. The connection w_t can be constant (no change in time), stationary (random and time independent changes) or non-stationary (time dependent changes) (Fig 1B).

In awake behaving human or animal subjects it is not possible to asses absolute or real connectivity and plasticity between neurons. By taking into account the spiking of multiple, individually recorded neurons, however, we can estimate their functional

connectivity with modelling of their putative pre- and post-synaptic spikes [16]. This can be done using generalized linear models [17–20], and prior knowledge can be incorporated by applying a Bayesian approach [21–24]. The generalized linear models can model connectivity, but to infer the learning rules driving the connectivity, more complex models are needed.

Connectivity rules have so far not been used for modelling disease mechanisms of memory impairments, and how these learning rules look like in the human brain has not yet been investigated in detail. Statistical models have been used to model and infer learning rules in simulated data [25, 26] and in experiments where memristors mimic synapses [27], but even though the technology to record the required spike data is available [28], to date not much work exists on how to use such recordings to infer learning rules in awake humans. We present a framework that paves the way for the study of synaptic plasticity in the awake human brain, by assuming that the connections between pairs of neurons are driven by an STDP learning rule, and through that inferring parameters $\boldsymbol{\theta}$ describing this rule. Our model for neural connectivity and activity dynamics can be represented in a network graph describing the dependencies in the neural spike data $\boldsymbol{s_i}$ and unknown synaptic weights (parameterizing the connections) $\boldsymbol{w} = (w_1, w_2, \ldots, w_T)$ for two neurons (Fig 1C). To uncover the unknown connections we use a Bayesian framework and assume a binary data model. We test the resulting method on both simulated and human spike recording data from the hippocampus, entorhinal cortex and amygdala [29] (Fig 1D).

Through the simulated case study we show that the Bayesian model we present is able to infer the true (simulated) parameters of the STDP rule. In spiking neuron data from human intracranial recordings, we show that the model is also able to infer the parameters from real data. Our results align with experimental results previously found in animals. Inspection of results from the human spike recordings showed a trend in increasing age and increasing maximum synaptic modification. To identify trends in learning rules across properties of the subjects was not the goal of our study, however, the data contains subjects of a wide range of ages (18–56), and the trend caught our attention in the process of visualizing the results. It has long been observed that the learning process and memory function changes with age [30], and this trend in learning rules supports an continued change but not in the way one might expect.

Results

Bayesian framework for inferring learning rules

We present a Bayesian framework for inferring STDP learning rules in spike data, that takes inspiration from [25], where spike counts are treated as observed variables, whereas dynamic, directed synaptic pairwise connections are regarded as hidden ones. For the spike count we use a time bin of 2 ms, which yields binary data $s \in \{0, 1\}$, for both putative pre-synaptic ($\boldsymbol{s}_1$) and post-synaptic neurons ($\boldsymbol{s}_2$). We assume that the spike data for the post-synaptic neuron, $s^2_{2,t}$, follows a Bernoulli process, and the mean of the Bernoulli process (likelihood) is a linear combination of background noise, covariates (if any), and pre-synaptic spikes from neuron 1 multiplied with a weight trajectory modelling the connection between the two neurons. This connection, denoted w_t, is assumed to change in time and to be driven by a spike-timing dependent plasticity (STDP) learning rule. We use a parametric representation of the learning rule, involving two parameters (A_+ and A_-) for maximum synaptic modification, and two parameters (τ_+ and τ_-) for time intervals where synaptic strengthening and weakening occur. A_+ and τ_+ describe the learning rule when the time difference between pre- and post-synaptic spikes is positive, and A_- and τ_- when the time difference is negative. To

simplify the computational expense of the model, we assume that $A = A_+ = A_-/1.05$ and $\tau = \tau_+ = \tau_-$. With a Bayesian approach, we can include prior knowledge about the parameters through the prior distributions. Based on the findings of [11] we choose the following priors for A and τ: $A \sim \text{Gamma}(2, 200)$ and $\tau \sim \text{Gamma}(4, 100)$. The goal of the modelling is to infer the parameters A and τ, which again describe the shape of the learning rule for a neuron pair. We study two types of learning rules: An additive version where the weight trajectory evolution is additive in the learning rule, and a multiplicative one where the evolution of the weight trajectory also has a multiplicative dependence on the weight in the previous time step. We use a Metropolis-Hastings algorithm to perform the inference (S1 Appendix). See Methods for additional details on the model.

Model performance evaluation with simulated data

To verify that the algorithm performs as expected, we applied it to a simulated case study. We simulated data with 2 ms time bins from the true model (i.e., the model we fit to the data). Unless otherwise stated, we used the following values: $A = 0.005$ and $\tau = 0.02$, $w_0 = 0.6$ (initial value of weight trajectory), $b_1 = b_2 = -3.476$ (corresponding to a spike rate for neuron 1 and 2 of 15 spikes/second), $\sigma_w = 0.001$ (weight trajectory noise), 180 seconds of data, and an additive learning rule. We did not include covariates in these datasets, as the goal of the analysis is the learning rule parameters and not covariate parameters, but did include covariates when investigating the performance of the Bayesian model in different scenarios (see Methods). We simulated 100 datasets, and divided them into three groups based on the evolution of the weight trajectory (Fig 2A): One where the true weight trajectory approaches 0 and the connection contains little information (difference between last and first value of the weight trajectory is less than -0.35). One where the weight trajectory only grows and the connection (with time) becomes saturated (difference between last and first value of the weight trajectory is greater than 0.35). And one in between (difference between last and first value of the weight trajectory between -0.35 and 0.35). We denote these three groups Δw_{low}, Δw_{mid} and Δw_{high}, respectively. Group Δw_{mid} resembles real data the most, as the two other groups are either non-informative (Δw_{low}) or becomes saturated with time (Δw_{high}). We applied the temporal correlation between spike trains method to simulated data in order to confirm that we identify the correct correlations and lags S1 Fig. We used 2000 samples for the inference, where the first 500 are burn-in samples (sampling based algorithms need a warm-up period) and were thus discarded. We have only included results where the effective sample size (ESS) is at least 15 (1% of 1500) for all model parameters (A, τ, w_0 and b_2). Early investigation of trace plots showed that the sampler has been able to explore the parameter space in a satisfactory way when ESS > 15. This gives 100 model fits with the additive learning rule (i.e., all converged), and 68 model fits with multiplicative learning rule. We display results from one dataset from each of the three groups, where both the additive and multiplicative model yield sufficiently large numbers of effective samples.

In the most realistic dataset group, Δw_{mid}, we see that the model is able to recover the true (simulated) value of A, and the posterior is more informative in terms of mode and uncertainty of the density, than the prior. (Fig 2B). This is true for both learning rules. For τ we see the same trend but not as prominent. For group Δw_{low}, the data does not contain enough information for the model to learn about the model parameters, and the shape of the posterior does not differ much from the prior. For group Δw_{high}, the model with additive learning rule is able to recover the true value of A, while the model with multiplicative rule is not. The weight trajectory w_t is computed at all time steps t, which we visualize to see how the posterior connection varies in time (Fig 2C). The models with additive and multiplicative learning rules

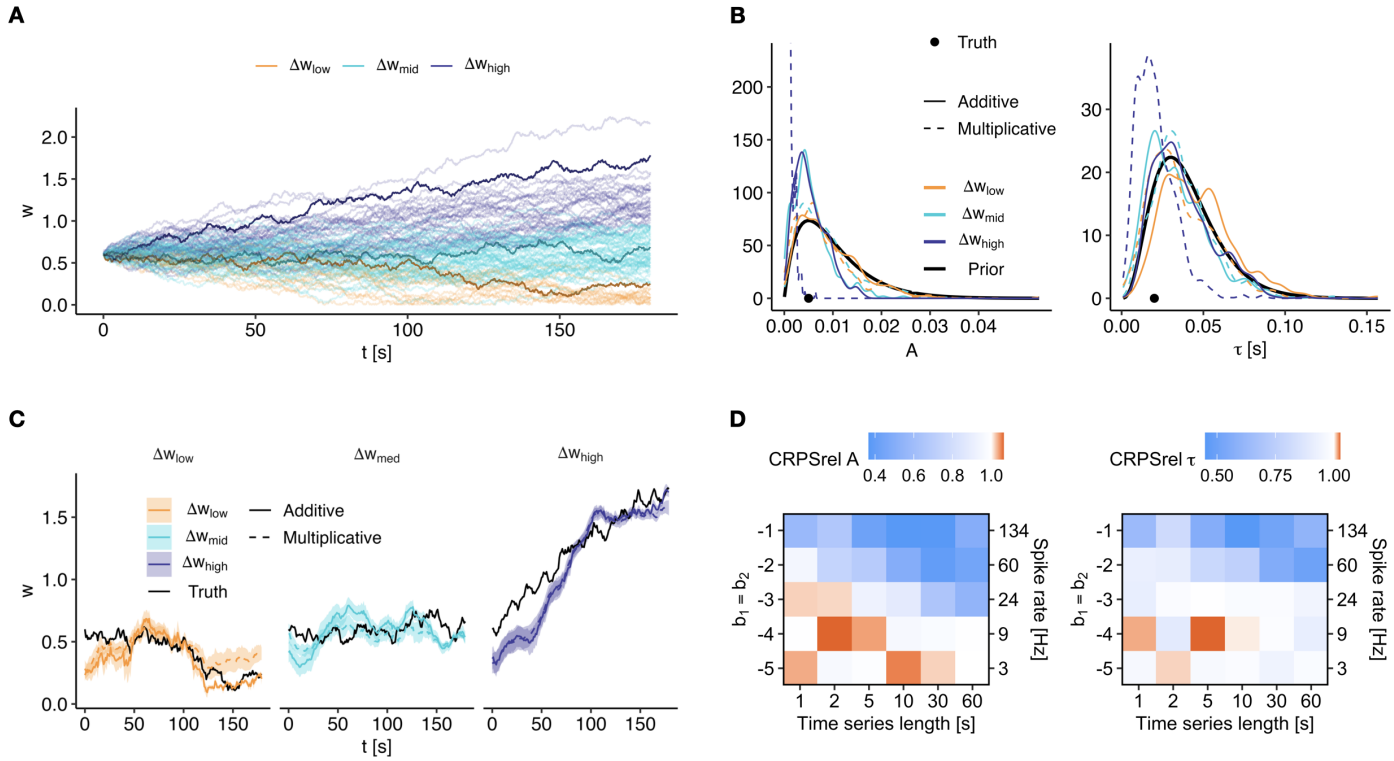


Fig 2. The Bayesian model uncovers the true (simulated) learning rule parameters. A) Weight trajectories of the 100 datasets used in the simulation study, divided into three classes Δw_{low}, Δw_{mid} and Δw_{high} (difference between last and first weight trajectory values below -0.35, between -0.35 and 0.35, and above 0.35). The results from three of the 100 datasets are displayed here, with highlighted weight trajectories in the graph. **B)** Prior and posterior distributions of A (left) and τ (right) for the three datasets. Note that the y-axis of the left graph is cut at 300. For A, the posterior is more informative than the prior (not for dataset group Δw_{low}). **C)** The posterior weight trajectories $w_t \pm 1$ standard deviation. The models to a large extent uncover the true (simulated) weight trajectory. **D)** The CRPSrel of A and τ for different true values of $b_1 = b_2$ and T, as the median over 10 simulated datasets. Only the additive learning rule is used in the model fit. CRPSrel refers to the CRPS relative to the CRPS for the prior distribution, so the prior has CRPSrel $= 1$. Longer spike trains with more spiking gives better values of the CRPSrel for both A and τ.

perform similar in recovering the true (simulated) trajectory w_t. Note that any value of the parameters will increase the likelihood value when the connection w_t becomes saturated and keeps increasing towards infinity. However, this is not an issue for the spike rates and time series lengths we have used in the simulated datasets displayed here and we expect additional mechanisms keep this from happening in real neuron.

When using statistical models, we must have enough information in the data to obtain reliable results. In our case, longer time series and higher spike rates of the two neurons gives more informative data. We therefore study the accuracy of estimating A and τ for different values of the spike rates for the two neurons and different time series lengths (Fig 2D). The accuracy is measured using the continuous ranked probability score (CRPS) [31]. The CRPS is a proper scoring rule which measures both accuracy and the uncertainty of the estimate (a low CRPS is desired). Assessing the uncertainty in addition to accuracy is suitable in the Bayesian setting where the results are posterior distributions, and ensures that accurate but uncertain estimates are ranked worse than accurate and certain estimates. The CRPS has the mean absolute error as a special case for point estimates.

To see how the posterior distributions of A and τ compare to the prior, we report the CRPS of the parameters from the inference divided by the CRPS of the prior, which we denote CRPSrel. The prior then has CRPSrel = 1, and values less than 1 mean the posterior is more certain about a more accurate point estimate than the prior, that is, the model has learned from the data. The median over 10 simulated datasets is used for each spike rate and time series length. We see that for higher spike rates and long time series our model returns a posterior that is more accurate with higher certainty than the prior for both A and τ, but for low spike rates and short time series the posterior is not performing better than the prior (Fig 2D). Longer time series and more frequent spiking (higher values of $b_1 = b_2$) results in more accurate parameter estimates of b_2, w_0 and w_t (S2 Fig, left). The times series needs to be sufficiently long for the model to lead satisfactory mixing (ESS > 15 for all parameters). For w_0, we get a median ESS smaller than 15 for nearly half of the scenarios (combination of low T and low $b_1 = b_2$), indicating poor mixing of the sampler (S2 Fig, right) In longer time series with spike rates above 24 Hz, we get ESS > 15, meaning the sampler is able to adequately explore the parameter space.

With the simulated data, we confirmed that the Bayesian model performs as intended, and that it is able to recover the true (simulated) learning rule parameters.

Inferring learning rules in humans

To test the usability of the model on real neural data, we used data from [29]. This dataset contains single neuron spike trains from 13 epilepsy patients, and in our work we have investigated the recordings from ten of them. To ensure that we studied connections that are more than random noise, we pre-selected 105 neuron pairs using temporal correlations of the spike trains [32] from seven of the subjects prior to the inference. From this procedure we also get information on which connections are excitatory, and which are inhibitory (see Methods). For each of the 105 neuron pairs we have three different sets of covariates: No covariates (cov0), covariates indicating trial difficulty (covD, four levels of difficulty based on work load in the trial), and covariates indicating trial stage (covS, four trial stages: Warning and fixation, encoding, maintenance and test). See [29] for details. We fit models with both additive and multiplicative learning rules and all covariate sets, yielding 6 model fits for each dataset.

To infer the learning rule parameters from the 105 neuron pairs, we ran the Bayesian inference algorithm for 4000 iterations, where the first 500 are used as burn-in (warm-up) and discarded. This gives 3500 samples of each parameter and the weight trajectory at each time step. We discard all model fits where one or more of the model parameters (b_2, β_k, A, τ, w_0) have effective sample size (ESS) less than 35 (as for the simulated data we used 1% of the total number of samples), as the Markov chains of those model fits did not converge. For some model fits, the sampler got stuck in parts of the parameter space and stopped due to numerical errors. These model fits were also deemed as not having converged and thus discarded. We report the number of converged models for each covariate group and learning rule type (Fig 3A).

With simulated data, we could evaluate the model performance by measuring how close to the truth the posterior distributions were. For real data, we cannot do this, and instead we compare the Bayesian approach to the straight-forward maximum likelihood approach, which assumes a constant weight trajectory ($w_t = w$) and thus no learning rule. More specifically, for each of the 105 neuron pairs and a given covariate set, we compared the log-likelihood value of the Bayesian model (with both additive and multiplicative learning rules) to the log-likelihood of the maximum likelihood approach. For all three covariate sets, the Bayesian model (both learning rules) almost always outperformed the maximum likelihood model. Furthermore, the Bayesian model with additive learning rule outperformed the maximum likelihood approach more often than

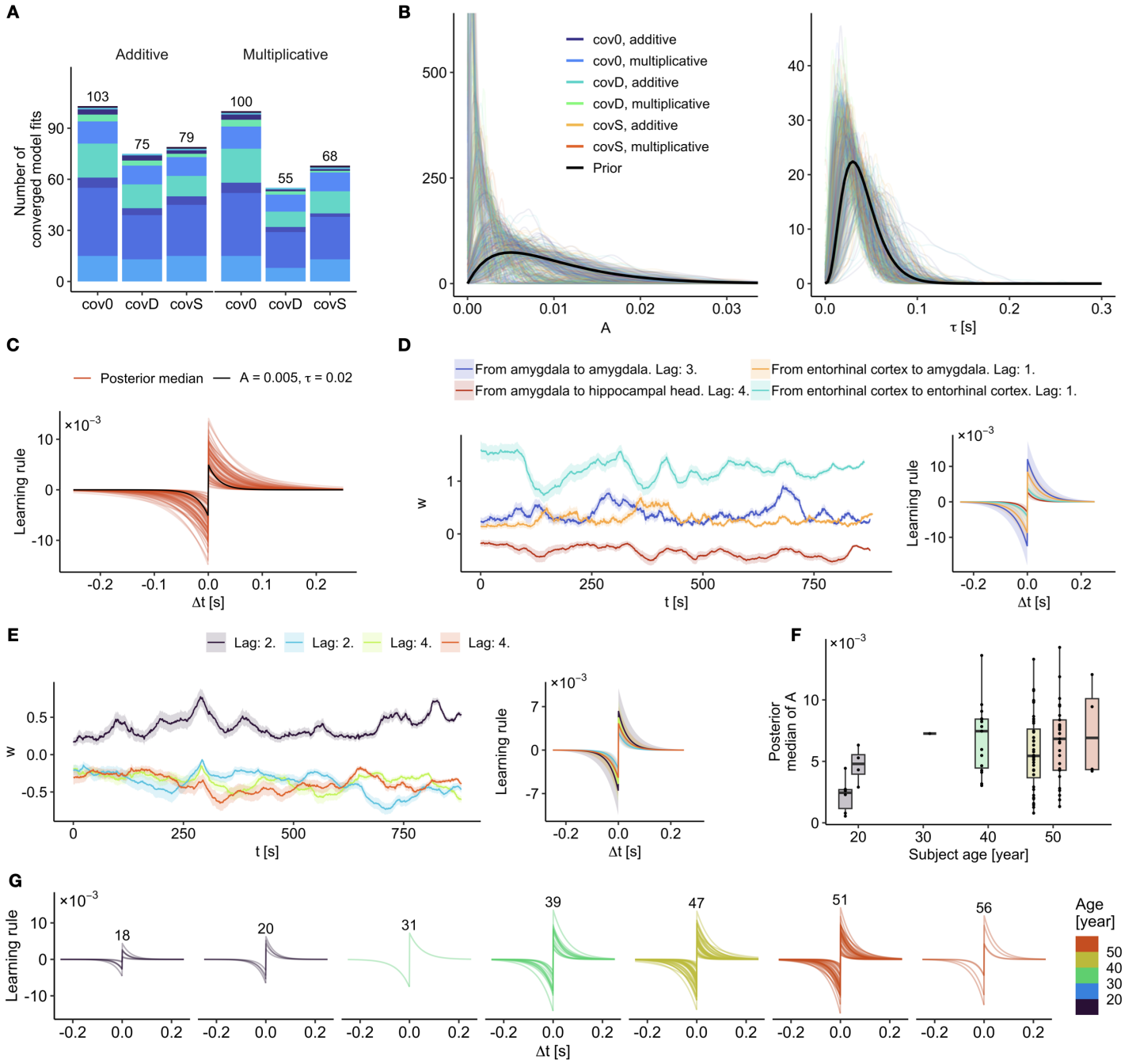


Fig 3. Results from case study with neural spike trains recorded in human temporal lobes. Only results from the model with additive learning rule without covariates (cov0) are displayed in C-G. **A)** The number of converged model fits for each learning rule and covariate set, out of a total of 105 neuron pairs. Colors indicate different subject and sessions. **B)** All posterior distributions of A (left) and τ (right), and the corresponding prior distributions. Note that the y-axis in the left graph is truncated at 600. **C)** Posterior median of A and τ are used to visualize the (additive) learning rule, together with the learning rule with parameters $A = 0.005$ and $\tau = 0.02$. The learning rule for all 103 converged model fits are displayed in the graph. **D-E)** Posterior weight trajectories (left) and corresponding learning rules (right) from four neuron pairs. The weight trajectories are displayed with posterior mean ± 1 standard deviation. The learning rules are based on posterior medians and corresponding uncertainties on lower and upper quartiles of A and τ. **D)** Recorded in different subjects. **E)** Recorded in the same subject and session (subject C). All neurons are located in the hippocampal head. **F)** Posterior median of the maximum synaptic modification A, sorted by age of subject. None of the subjects in the study have the same age. The box plots consist of median, upper and lower quartiles, and the lowest and highest value (there was no outliers). The individual points are included as well. **G)** Posterior learning rules based on posterior median of A (shown specifically in F) and τ, sorted by age of subject.

the Bayesian model with a multiplicative rule. We obtained information on whether each connection is excitatory or inhibitory from the temporal correlation of spike trains procedure. For 9 of the neuron pairs, the maximum likelihood approach estimates an inhibitory connection (determined by the temporal correlation procedure) as excitatory. We only compare the log-likelihood for neuron pairs where the maximum likelihood estimates the correct (based on temporal correlation procedure) connection type (excitatory/inhibitory). To check that the Bayesian model behaves as we would expect in one of the cases where the maximum likelihood estimates disagree with the temporal correlation procedure, we show how the Bayesian model performs in one of those cases (S3 Fig, third panel). We do not see anything indicating that the connection is wrongly classified using the temporal correlation: The multiplicative learning rule is small, indicating that the learning is close to non-existing, while the additive also is small compared to the other panels. However, the learning rules for one of the excitatory connection shows the same trend (S3 Fig, fourth panel), indicating that the connection is correctly identified with the temporal correlation. However, that 9 out of 105 neuron pairs are wrongly classified, both either the temporal correlation procedure and the maximum likelihood approach, is possible as we do not know the truth and instead use tail probabilities and estimates.

The posterior distributions of A are (in general) more narrow than the prior, and the mean has shifted compared to the prior (Fig 3B, left). This suggestes that we have learned about the parameter from the data. The same is true for τ (Fig 3B, right), but to a smaller extent. This is expected due to its role in the hierarchical model [33]. The posterior of the learning rule parameters does not seem to change between the different sets of covariates (Fig 3B and S4 Fig). In comparing the two learning rules, we see that out of the 105 neuron pairs the additive and multiplicative learning rule led to convergence in 103 and 100 neuron pairs, respectively, for the model without covariates (cov0, Fig 3A). In the shared group of 100 neuron pairs where models with both learning rules converged, the additive learning rule led to the highest (best) log-likelihood value in 93 pairs, and the multiplicative led to the highest log-likelihood value in only 7. The learning rule parameters are largely not changed by the addition of covariates (S4 Fig). Both the posterior weight trajectories w_t and learning rules are highly similar, even though different covariate sets are used S3 Fig. The multiplicative rule tends to yield a slightly smoother weight trajectory than the additive. More smoothing occurs when the corresponding learning rule has a small maximum synaptic modification A, as this results in smaller changes Δw. Based on these two findings, and since out goal is to infer the learning rule parameters, we only consider the model with additive learning rule and no covariates (cov0) in the remainder of this section.

In order to compare the results from our model to previous experimental animal results, we compare the learning rules from the 103 model fits with additive learning rule and no covariates to the learning rules with parameters proposed by [11] ($A = 0.005$, $\tau = 0.02$), which are found to match experimental results (Fig 3C). We see that the posterior values from human spike recordings vary around the $A = 0.005$ and $\tau = 0.02$ learning rule, indicating that the learning rules in humans resembles the ones in animals.

To visualize how the connection between two neurons vary in time, and how the connections vary between neuron pairs in the same and different subjects, we display posterior weight trajectories (Fig 3D, left) and corresponding learning rules (Fig 3D, right) for four neuron pairs in four different subjects. We see that the uncertainties of the learning rules are only partially overlapping between the subjects. There is however no clear pattern between sign or magnitude of the weight trajectory and learning rule. We also display posterior weight trajectories (Fig 3E, left) and learning rules (Fig 3E, right) for four neuron pairs from the same subject and session. Even though the model is fitted independently to the spike data from the four neuron pairs, all four trajectories

have the same increase followed by a decrease at around 300 seconds. We also see that the uncertainties in the learning rules overlap to a larger degree when all neuron pairs are from the same subject and session (Fig 3E, left) than when they are from different subjects (Fig 3D, left). All weight trajectories are far from constant, which is supported by the corresponding learning rule parameters which are significantly larger than zero.

When ploting the synaptic modification relative to the patients age, we see an increasing trend in the maximum synaptic modification A with increasing age (Figs 3F and 3G). However, the number of neuron pairs analysed for each subject varies, and there are more connections for high ages than for low, so we cannot draw any conclusions without further analysis of more data. Note that some subjects have participated in two sessions. We do not indicate session numbers in the graphs. No trends similar to the age trend can be seen when grouping the learning rules by gender, brain area, or seizure onset zones, the differences are larger between subjects (S5 Fig).

Gender does not seem to impact the learning rule, neither the shape nor the magnitude (S5 FigA). Learning rules from neuron located in two different areas seems slightly larger in magnitude than two neurons in the same area (S5 FigB). Among the learning rules from pairs where both neurons are in the same area, the amygdala has the largest learning rules (S5 FigC). The trends in learning rule magnitude is not only due to the grouping, but also due to different individuals. When both neurons in a pair are located in a seizure onset zone, the learning rule tends to be smaller (S5 FigD), and the side of the brain (left/right) does not seem to affect the learning rule (S5 FigE). The most prominent pattern is with increasing age (Fig 3G and S5 FigF). Since this also corresponds to the different individuals, and the slight trends in gender and brain area are at least to some extent due to the sparsity of the data in terms of brain areas and subjects, we would need more data, both more subjects and more neuron pairs from all areas of the brain, to draw any conclusions. We have only sorted/grouped the posterior learning rules based on information not used in the inference. We do not look at the pathology, as all but two subjects have hippocampal sclerosis.

Discussion

We inferred the unknown, time-varying neural connections in simulated and real data using a Bayesian model. The simulated case study showed that the Bayesian approach with a binary data model can work, and the real case study with human brain electrophysiological recordings showed that the learning rules we found in the human brain resemble results found experimentally *in vitro* in rats and other animals.

Brain connectivity and STDP learning rules have long inspired the development of computer hardware and algorithms [34–36], in artificial neural networks [37–41], and in simulations of brain networks [42]. Extended knowledge on how these learning rules work in the human brain can improve the algorithms where learning rules are employed as well as our general understanding of plasticity in the brain. In particular, from this work, it seems that STDP in the human brain takes largely the same form as that of lower species, such as the rat.

Since the data from human single neurons were recorded during a short term memory task [29], changes in plasticity might also reflect different mnemonic items being retained in memory [43]. However, to link connectivity and plasticity to a short term mnemonic process would require orders of magnitude more data, since the item retention lasts only 3 seconds for every trial.

One of the several advantages with Bayesian inference is that we get uncertainties for our model estimates through the posterior distribution, opposed to just a point estimate from e.g. maximum likelihood approaches. We can also easily see if the inference has utilized the data, or if the posterior is similar to the prior, indicating that

the model did not learn from the data. Another advantage of the Bayesian framework is the possibility of including prior knowledge. Extensive research has been done on functional connectivity and STDP learning rules [11, 12, 15], which we can include in the model through the prior to stabilize the inference. The Bayesian model outperformed the maximum likelihood approach, reflecting that to assume a constant connection would be an oversimplification of the connections in the brain. This is further emphasized by the fact that the maximum likelihood approach is not always able to identify an inhibitory connection (labeled using the correlation approach) as inhibitory.

The model is computationally expensive to fit, partially due to the number of observations (500 observations each second), and we do not expect learning to happen at all times. Further work could help to identify specific moments of learning and only including moments of learning in the inference will decrease the run time of the model. Further work could also include investigating whether the learning rule is purely additive, purely multiplicative, or a combination of the two, or even something in between additive and multiplicative [44]. Triplet-STDP learning rules could be tested with this approach [44, 45], and it will be important to study if and how the learning rule parameters change over time.

Modeling the connections between more than two neurons at the same time, or even to unobserved neurons [46, 47], could improve the quality of the posterior distributions of the parameters, as we would have more data and thus more information to the model. This would however increase the model complexity. Parameter sharing across pairs of neurons could also increase the posterior distribution quality, but still with an increase in the complexity. To allow A_+ and A_-, and τ_+ and τ_-, to be inferred independently increases the model complexity even more, but allows additional flexibility in the model and would be interesting to investigate.

In the evaluation of the learning rules, we see that there appears be a relationship between age and magnitude of the learning rule (more specifically, the maximum synaptic modification A), but more data and evidence is needed in order to draw conclusions. It would be interesting to investigate this further, as this is in line with a change in memory function with age [30]. There are reported enhanced long term potention [48], reduction of learning [49], and more stable synapses with aging [50], which together appear contradictory. However, we measure functional connectivity, and one explanation for this trend is that older patients split less between different contexts such that the same neurons are active more often, in turn leading to bigger changes in the connectivity. Another explanation is that neurons in older brains lose firing precision and that their relative spike timing fluctuates more than in younger brains.

In conclusion we show that a Bayesian framework can be successfully used to model and gain insight into the connectivity and plasticity between neurons using single neuron spiking data with known parameters (synthetic data) and from intracranial human recordings.

Methods

Temporal correlation between spike trains

There are several thousand neuron pairs available in the human spike recordings. The Bayesian model is too computationally expensive to fit to all neuron pairs, so we pre-select connections where we are more likely to see learning. We only consider neuron pairs where both neurons have an average spike rate of at least 5 spikes/second to ensure the data contains enough information for the analysis. To identity temporal connections in the data, we used the method described in [32]. A summary of the method and the specific settings we chose follows.

For spike trains from two neurons, we made the continuous time series into binned 0/1 time series, where for each time bin one or more spikes in a time bin interval codes to 1 and else to 0. We use a bin size of 2 ms. We keep one of the neuron spike trains fixed, and compute the number of simultaneous spikes when we shift the other spike train from -100 to $+100$ time bin lags. To avoid a bias towards time lag 0 due to a lower number of time bins to compare for high lags, we cut the ends of the spike trains such that we use the same number of time bins in the comparison for all time lags. To ensure the assumptions of further tests will not be violated, we remove all spikes but the first when the interval between two spikes is less than 6 ms. When we have computed the number of simultaneous spikes for the 201 time lags (-100 to $+100$), we obtain a raw cross-correlation histogram (raw CCH).

The second step is to convolve the raw CCH with a partially hollow rectangle with a width of 11 time bins. The window hollow fraction is set to 0.42, as suggested in [32]. To prevent edge effects, bins two to six are duplicated, reversed and appended to the beginning of the raw CCH, and corresponding to the end of the CCH. This yields a predictor CCH.

Third, we compute the probability of observing the count n we observed or higher given the predictor CCH for each time lag a. We apply a continuity correction to obtain an unbiased test, and compute two tail probabilities:

$$p_a^{\text{ex}} \sim \text{Uniform}(\text{P}(X \geq n|\lambda(a)), \text{P}(X \geq n+1|\lambda(a))),$$

and

$$p_a^{\text{in}} \sim \text{Uniform}(\text{P}(X \leq n|\lambda(a)), \text{P}(X \leq n+1|\lambda(a)))$$

where X is Poisson distributed with rate $\lambda(a)$ for time lag a, and λ is obtained from the predictor CCH. The first tail probability, p_a^{ex}, is for excitatory connections (how unlikely it is that the neurons had this many simultaneous spikes, meaning spikes from the first neuron increases the probability for the second to spike), and the second, p_a^{in}, is for inhibitory (how unlikely it is that the neurons had this few simultaneous spikes, meaning spikes from the first neuron decreases the probability for the second to spike). For one neuron pair, we get these two tail probabilities for each time lag a. We are interested in the lags where we get small tail probabilities, and say that neurons and lags with tail probability less than 0.025 create a connection we want to analyze. Note that this method is not a formal test, and is merely used to reduce the amount of neuron pairs we study. Small tail probabilities indicate that the observed spike rate is unusual and thus caused by more than only random noise.

Since our goal is to estimate the learning rule parameters, we do not consider the zero-lag and we only consider lags up to ± 5, corresponding to 10 ms, to capture the desired learning effects. For each neuron pair, we use at most one connection with positive lag and one with negative lag. The lag with the smallest tail probability is chosen. A connection at positive lags means that the spike train we held still when computing the CCH belongs to neuron 1, and the moving spike train belongs to neuron 2. For connections at negative lags it is the opposite. For the data from [29] we considered the neurons recorded in different sessions with the same subject to be independent. We did not connect neurons from different sessions in any way.

Human spike data

The area each neuron is recorded from is given in the data, in addition to which (if any) areas that are seizure onset zones for each subject. The data is recorded during a visual memory task consisting of 192 trials with four different levels of difficulty. In addition to the spike trains, we also extract information on the difficulty of each trial and when each stage of the trial occurs, as well as the pathology, gender and age of the subjects.

For a given subject and session, we extract the spike times for each neuron, and create one time series with all the trials. Temporal correlation between the spike trains and requiring at least 5 spikes/second on average yielded 105 neuron pairs from 7 subjects (Table 1). Two subjects participated in two sessions.

Table 1. Summary of the human spike recordings.

Subject	Session	Excitatory (S, D)	Inhibitory (S, D)	Total
A (31)	S1	$(0, 1)$	$(0, 0)$	1
B (56)	S1	$(0, 0)$	$(1, 0)$	1
	S2	$(2, 1)$	$(0, 0)$	3
C (20)	S1	$(1, 0)$	$(3, 0)$	4
D (51)	S1	$(2, 5)$	$(3, 3)$	13
	S2	$(4, 10)$	$(3, 3)$	20
E (18)	S1	$(3, 0)$	$(3, 0)$	6
F (47)	S1	$(4, 15)$	$(11, 11)$	41
G (39)	S1	$(3, 6)$	$(1, 6)$	16
Total	9	$19 + 38 = 57$	$25 + 23 = 48$	105

Spike recordings from [29]. Age of subjects in parenthesis. S1 and S2 represent different sessions. (S, D) denotes the number of neuron pairs in the same (S) and in two different (D) brain areas. Subject A, F and G are male. The number of connections with satisfactory high spike rates found varies between subjects, from 1 to 41. Of the 105 neuron pairs, 61 are across different parts of the brain, and 44 are within the same brain area. We only consider neuron pairs recorded with two different electrodes. 57 neuron pair connections are excitatory and 48 are inhibitory. The 105 neuron pairs are from 7 different subjects, where two of those have participated in two sessions.

Bayesian model for inferring learning rules

To infer the parameters of the STDP learning rule in spike data, we first consider binned spike trains $\boldsymbol{s}_i$ of length T. Each time bin is of size 2 ms, and time bin t covers the events in the time interval $((t-1) \cdot 2\text{ms}, t \cdot 2\text{ms}]$ (except the first time bin which also includes 0). For a given neuron i and time bin t, $s_{i,t} = 1$ if the neuron spiked (at least once) in time bin t, and $s_{i,t} = 0$ else. We take the post-synaptic spikes as the response, and build a likelihood model with the pre-synaptic spikes as input. The natural choice of likelihood model for binary spike train data $\boldsymbol{s}_i$ is the Bernoulli model:

$$s_{i,t} \sim \text{Bern}(\mu_{i,t}) = \mu_{i,t}^{s_{i,t}} (1 - \mu_{i,t})^{1 - s_{i,t}},$$

where the linear predictor $\eta_{i,t} = \text{logit}(\mu_{i,t})$ is given by

$$\eta_{i,t} = \sum_{j=1, j \neq i}^{N} w_{ij,t} s_{j,t-a} + b_i + \sum_{k=1}^{K} \beta_k x_{k,t}.$$

i and j indicate different neurons, a is the time lag for this specific connection as estimated from the temporal correlation between the spike trains, b_i is an intercept representing background noise, and β_k is the parameter of the covariate $\boldsymbol{x}_k = (x_{k,1}, \ldots, x_{k,T})^{\mathrm{T}}$. $w_{ij,t}$ describes the connection from neuron j to neuron i, and $w_{ij,t} \neq w_{ji,t}$. We model the connection, or weight trajectory, $w_{ij,t}$ as a random walk of first order:

$$w_{ij,t+1} = w_{ij,t} + l(\boldsymbol{s}_{i,1:t}, \boldsymbol{s}_{j,1:t}, \boldsymbol{\theta}, w_{ij,t}) + \varepsilon_{t+1},$$

where $\varepsilon_{t+1} \sim \mathcal{N}(0, \sigma_w^2)$ is Gaussian noise and $\boldsymbol{\theta} = (A_+, A_-, \tau_+, \tau_-, w_{\min}, w_{\max})$ are the learning rule parameters. The initial value of the weight trajectory $w_{ij,t}$ is denoted $w_{ij,0}$. The weight trajectory is restricted to be non-negative ($w_{ij,t} \geq 0$) for an excitatory connection, and non-positive for an inhibitory connection ($w_{ij,t} \leq 0$). $l(\cdot)$ is a learning rule, and we considered a learning rule with additive dependence on the weight $w_{ij,t}$ is additive (additive STDP learning rule):

$$l(\boldsymbol{s}_{i,1:t}, \boldsymbol{s}_{j,1:t}, \boldsymbol{\theta}) = A_+ s_{i,t} \sum_{t'=1}^{t} s_{j,t'} e^{\frac{t'-t}{\tau_+}} - A_- s_{j,t} \sum_{t'=1}^{t} s_{i,t'} e^{\frac{t'-t}{\tau_-}},$$

and a learning rule where the dependence is multiplicative (multiplicative STDP learning rule):

$$l(\boldsymbol{s}_{i,1:t}, \boldsymbol{s}_{j,1:t}, \boldsymbol{\theta}) = (w_{\max} - w_{ij,t}) A_+ s_{i,t} \sum_{t'=1}^{t} s_{j,t'} e^{\frac{t'-t}{\tau_+}} - (w_{ij,t} - w_{\min}) A_- s_{j,t} \sum_{t'=1}^{t} s_{i,t'} e^{\frac{t'-t}{\tau_-}}.$$

A_+ and A_- represent maximum synaptic modification and τ_+ and τ_- represent time intervals where synaptic strengthening and weakening occur. [11] investigate which parameters for the STDP learning rules fit best with experimental data from rats and other animals, and find that $A = A+ = A_-/1.05 = 0.005$ and $\tau = \tau_+ = \tau_- = 0.02$ yields simulated results close to the experimental results in frogs and rats [9, 51–54].

The goal of our analysis was to infer the learning rules for pairs of neural connections, and investigate how the whole network of neurons changes in time was outside the scope. Due to this, we considered one neuron pair, and only the connection from the first neuron (neuron 1) to the second (neuron 2), such that the model simplifies to $s_{2,t} \sim \text{Bern}(\mu_{2,t})$ and $\eta_{2,t} = w_t s_{1,t-a} + b_2 + \sum_{k=1}^{K} \beta_k x_{k,t}$ where we simplify the notation by setting $w_{21,t} = w_t$. This means we assumed that the only neuron affecting neuron 2 is neuron 1, and all other influence is assumed to be background noise, included in b_2, and through the noise component of the weight trajectory. We used this assumption even though one neuron may have connections to multiple other neurons, and the same neuron can be neuron 1 in one pair, and neuron 2 in another. Other neurons are assumed to lead to a systematic bias, affecting both neurons equally, which affects the magnitude but not the shape of the weight trajectory.

Due to the complexity of the model, we used some simplifications to reduce the computational expense during inference: $A_+ = A_-/1.05 = A$, $\tau_+ = \tau_- = \tau$ [11], $w_{\min} = 0$ and $w_{\max} = 1$ for excitatory connections and $w_{\min} = -1$ and $w_{\max} = 0$ for inhibitory connections [44]. We also simplified the computation of the learning rule by only summing over the previous $10 \cdot \tau$ time bins, as the contribution further back in time is essentially zero, which is also supported by experimental results [11]. In addition, we fixed the noise parameter of the weight trajectory σ_w^2.

Inference method

The model is non-linear and not latent Gaussian due to the learning rule. This makes the model computationally expensive to fit, but also gives a flexible model. We used a Bayesian framework for the inference. Bayesian models require prior distributions for the model parameters A, τ, b_2, $w_{ij,0}$, and β_k, but also gives us the opportunity of including prior knowledge about these parameters. In a hierarchical model, such as the one we consider, the difficulty of inferring a model parameter depends on the distance to the data level of the model [33]. Intercepts (b_2) and fixed effects (β_k) are relatively easy to infer, while the learning rule parameters and weight trajectory are more difficult. The latter is especially true for τ, and we need large amounts of informative data to infer this parameter. This means that the intercept and fixed effects can have vague

priors, while the learning rule parameters require stronger priors. Too strong prior distributions can however yield posterior distributions highly similar to the priors, as the sampler is not able to investigate the full parameter space, so caution is needed. Based on this and the findings of [11], we chose the following prior distributions: $A \sim \text{Gamma}(2, 200)$, $\tau \sim \text{Gamma}(4, 100)$, $w_0 \sim \mathcal{N}_{\geq 0}(1, 5^2)$ (truncated Gaussian distribution) for excitatory connections and $w_0 \sim \mathcal{N}_{\leq 0}(-1, 5^2)$ for inhibitory connections, $b_2 \sim \mathcal{N}(0, 10^2)$, and $\beta_k \sim \mathcal{N}(0, 5^2)$. We fixed the noise of the weight trajectory to $\sigma_w^2 = \frac{1}{3}\mathrm{E}_{\pi(A)}(A) \approx 0.0033$ to ensure that the evolution of w_t rarely is no more than 50% noise at time steps with spiking.

We used a Metropolis-Hastings (M-H) algorithm [55] to fit the model (S1 Appendix). The background noise b_2 and the parameters of the covariates β_k are updated independently using Gibbs steps [56], while A, τ and w_0 are sampled together in a M-H step. In the same M-H step we also sampled the weight trajectory w_t using a particle filter [57]. To increase the efficiency of the algorithm, we used adaptive variance adjustment [58]. Approaches with adaptive sampling leads to correlated samples. To ensure that the algorithm has sufficient mixing of the Markov chains we use effective sample size to compute the number of independent samples from the posterior distribution that contain the same information as our correlated samples [59].

We compared the Bayesian model to a model assuming a static (constant) weight trajectory, i.e., the maximum likelihood, where $w_t = w$ for all t. We do not include restrictions or regulations for the maximum likelihood approach, meaning the maximum likelihood has to uncover whether the connection is excitatory or inhibitory by itself. For the comparison, we use the (in-sample) log-likelihood value of the posterior model. That is, we compute

$$\sum_{t=a+1}^{T} \log(p(s_{2,t}|\mu_{2,t})) = \sum_{t=a+1}^{T} s_{2,t} \log(\mu_{2,t}) + \sum_{t=a+1}^{T} (1 - s_{2,t}) \log(1 - \mu_{2,t}),$$

where

$$\text{logit}(\mu_{2,t}) = \eta_{2,t} = w_t s_{1,t-a} + b_2 + \sum_{k=1}^{K} \beta_k x_{k,t}$$

is the linear predictor with the posterior estimates. a is the time lag for this specific connection. For the maximum likelihood approach, we have point estimates and $w_t = w$ for all t, while with the Bayesian approach we compute the log-likelihood using each posterior sample of b_2, β_k and the weight trajectory w_t and obtain a distribution for the log-likelihood. We used the median of the log-likelihood value for the comparison.

We did not use the model for prediction, and did therefore not evaluate the prediction abilities of the model.

Code and reproducibility

We implemented the model in the statistical programming language `R` [60], and both simulated and real case studies were run on x86_64-pc-linux-gnu (64-bit) with `R` version 4.2.2. Necessary code for running the model is available here: `https://github.com/ingebogh/synaptic_plasticity`. This includes code to generate simulated data.

Visualizing the learning rules

To visualize the learning rules for a given value of A and τ, we used the following formula for the additive rule:

$$\begin{cases} A \exp\left(\Delta t/\tau\right), \Delta t < 0 \\ -1.05 \cdot A \exp\left(-\Delta t/\tau\right), \Delta t \geq 0 \end{cases}$$

while the multiplicative was computed with:

$$\begin{cases} (w_{\max} - w_t) A \exp\left(\Delta t/\tau\right), \Delta t < 0 \\ -(w_t - w_{\min}) \cdot 1.05 \cdot A \exp\left(-\Delta t/\tau\right), \Delta t \geq 0. \end{cases}$$

This gives slightly different interpretations of the two learning rules. We set $w_t = 0.5$ for excitatory connections and $w_t = -0.5$ for inhibitory. Note that since $w_{\min} = 0$ and $w_{\max} = 1$ for excitatory and $w_{\min} = -1$ and $w_{\max} = 0$ for inhibitory connections, the choice of $w_t = 0.5$ means that for equal A and τ the multiplicative learning rule has the same function values for excitatory and inhibitory connections, and is 0.5 times the corresponding additive learning rule.

Model verification

We performed a simulation study to test the properties of the model and algorithm. We used the datasets displayed in the results from simulated data (Fig 2A) unless otherwise stated, that is, $A = 0.005$ and $\tau = 0.02$, $w_0 = 0.6$, $b_1 = b_2 = -3.476$, $\sigma_w = 0.001$, 180 seconds of data, and an additive learning rule. These data have been divided into three groups based on the difference of the first and last value of the weight trajectory $(w_T - w_1)$. The groups are Δw_{low} (difference smaller than -0.35), Δw_{mid} (difference between -0.35 and 0.35) and Δw_{high} (difference larger than 0.35). We use time lag 1 in both the simulated data and the model. We fit the model with both the additive learning rule and the multiplicative learning rule.

Performance assessment

To assess the performance of the model, we evaluated the following: Continuous rank probability score (CRPS) [31] of the model parameters. CRPS of the weight trajectory, where we take the average of the CRPS over the time steps. Low values are desired. We used the `R`-package `scoringRules` [61] to compute the CRPS. Effective sample size (ESS). Samples obtained from a Markov chain sampler are correlated. The ESS is a number reflecting how many independent samples the correlated samples we have correspond to. This is lower than the number of samples, and gives an indication on how well the chains have mixed during inference. High values are desired. We computed the ESS using the `R`-package `coda` [59]. We include all model fits that have not crashed during inference, also those with a low ESS, and report the average CRPS and ESS. A few model fits crashed during the inference, that is, the algorithm diverged into a part of the parameter space where the parameters are so close to zero that it experiences numerical errors. The number of crashed fits are indicated in the relevant figure.

Number of particles P

We tested the difference in performance when we use $P = 1, 10, 50, 100, 200, 500$ and 1000 particles in the particle filter in the algorithm. We assessed the CRPS for the model parameters (compared to the simulated truth) and the weight trajectory w_t, and

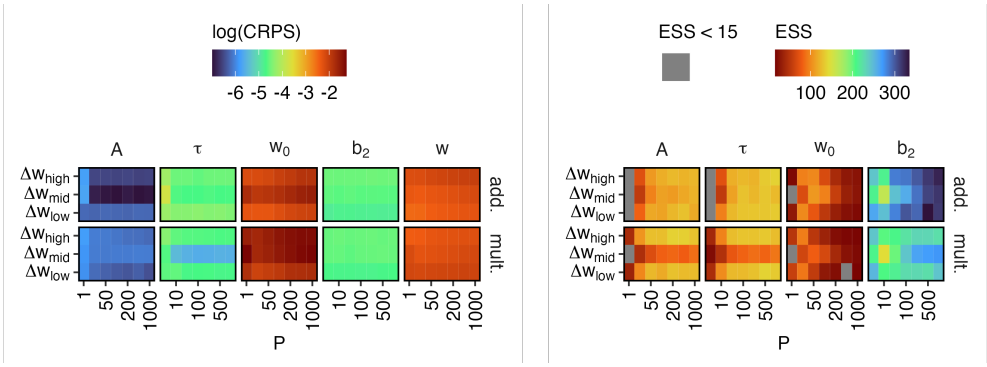


Fig 4. Performance for different number of particles P. log(CRPS) for the model parameters and weight trajectory (left) and ESS for the model parameters (right), as the mean over the 15, 53 and 32 model fits for each weight trajectory group (Δw_{low}, Δw_{mid} and Δw_{high}) and value of P. w represents the whole weight trajectory. We compute the average of the CRPS, and use the logarithm when visualizing the results. 1 multiplicative model for $P = 10$ crashed during the inference.

the ESS for the model parameters (Fig 4). We display results for the different values of P, and the different dataset groups (Δw_{low}, Δw_{mid} and Δw_{high}).

We see that the CRPS of A and τ plateaus at around 100 particles, while the ESS stops increasing at 200 particles. The CRPS for b_2 is almost constant, while the ESS for b_2 increases up to 500 particles. The model with additive learning rule is in general outperforming the multiplicative (this is expected since the data is generated using the additive rule). The results for the weight trajectory w_t and the initial value and w_0 are best for $P = 50$ and becomes worse for more particles. Our focus is on the learning rule parameters A and τ, but we also need to make sure the other model parameters and effects obtain sufficient mixing in the algorithm, so we chose $P = 200$ particles in the particle filter where all parameters and effects result in good inference. The results differ slightly between the dataset groups.

Prior distributions for A and τ

We tested four different priors for A, and four for τ (Fig 5). The prior choice for A mainly affects the posterior diagnostics for A (Fig 6), but the other parameters are to some degree influenced as well. The Gamma$(2, 400)$-prior yields the best results, but has a mean equal to the true value and is deemed too strong and not optimal for new data where we do not know the truth. The Gamma$(4, 50)$-prior is too wide and does not lead to accurate parameter estimates, and Gamma$(1, 50)$ shrinks the parameter to 0 and leads to poor mixing (low ESS). We therefore chose the Gamma$(2, 200)$-prior, as it is not too wide and not too narrow. The multiplicative learning rule model is outperformed by the additive model. The prior choice for τ mainly affects τ (Fig 7). The Gamma$(1, 100)$-prior lead to lower ESS, while Gamma$(2, 100)$ has mean equal to the true (simulated) value and is too strong. Gamma$(4, 100)$ leads to a slightly lower and thus better CRPS for τ than Gamma$(5, 100)$, so we chose $(4, 100)$ as the prior parameters for τ. Again, the multiplicative learning rule model is outperformed by the additive. The weight trajectory in the most realistic dataset group (Δw_{mid}) is more difficult to estimate than the others, based on the CRPS value of the weight trajectory (Figs 6 and 7).

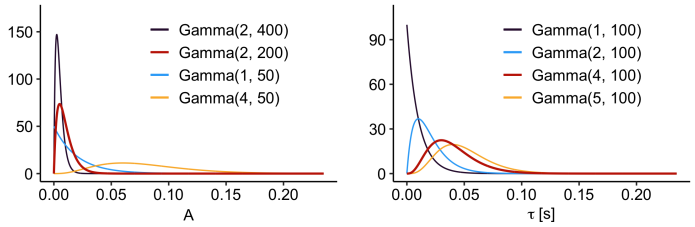


Fig 5. Prior distributions for learning rule parameters. The prior distributions for A (left) and τ (right) investigated in the simulation study. The bold priors are the chosen ones ($A \sim \text{Gamma}(2, 200)$ and $\tau \sim \text{Gamma}(4, 100)$).

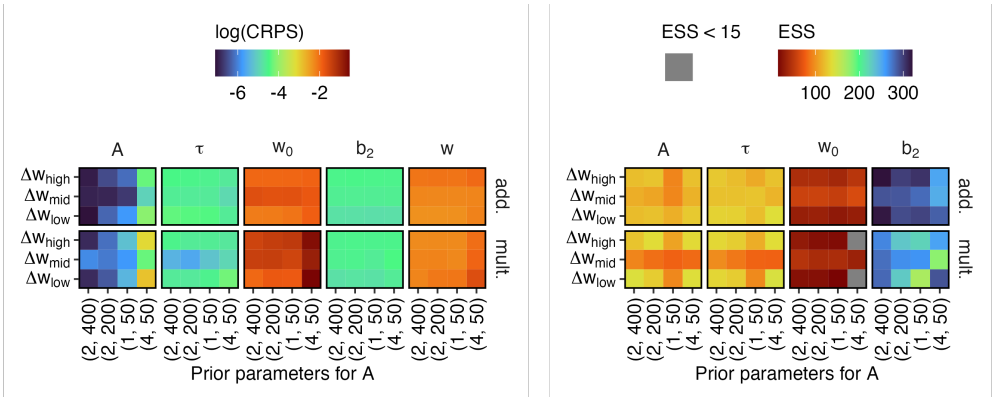


Fig 6. Performance when different prior parameters for the Gamma distribution on A are used. Sorted by increasing prior mean (x-axis) and for the different groups of weight trajectories (y-axis). log(CRPS) for the model parameters and weight trajectory (left) and ESS for the model parameters (right), as the mean over the 15, 53 and 32 model fits for each weight trajectory group (Δw_{low}, Δw_{mid} and Δw_{high}) and prior. w represents the whole weight trajectory. We compute the average of the CRPS, and use the logarithm when visualizing the results. 5 multiplicative models with a Gamma$(1, 50)$-prior for A crashed during the inference.

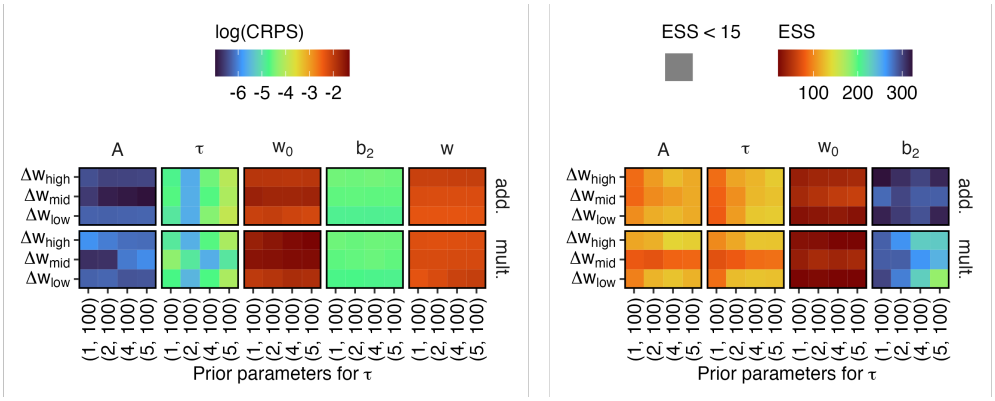


Fig 7. Performance when different prior parameters for the Gamma distribution on τ are used. Sorted by increasing prior mean (x-axis) and for the different groups of weight trajectories (y-axis). log(CRPS) for the model parameters and weight trajectory (left) and ESS for the model parameters (right), as the mean over the 15, 53 and 32 model fits for each weight trajectory group (Δw_{low}, Δw_{mid} and Δw_{high}) and prior. w represents the whole weight trajectory. We compute the average of the CRPS, and use the logarithm when visualizing the results. 2 multiplicative models with a Gamma$(1, 100)$-prior for τ and 1 multiplicative model with a Gamma$(5, 100)$-prior for τ crashed during the inference.

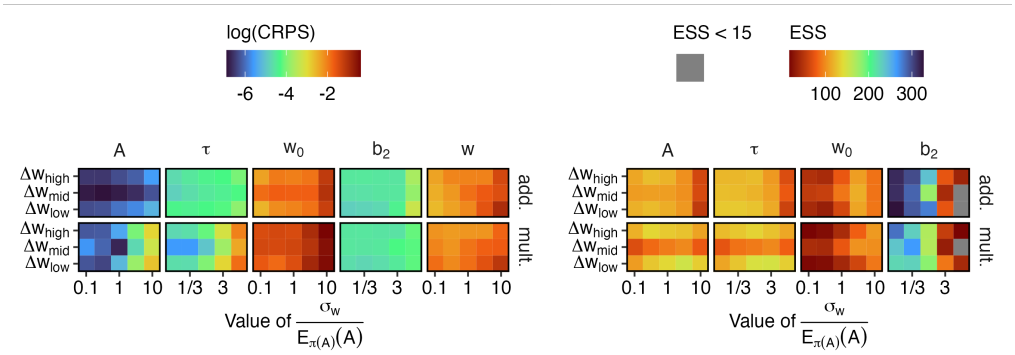


Fig 8. Performance when different values of the fixed noise parameter σ_w. Sorted by increasing value (x-axis) and for the different groups of weight trajectories (y-axis). The prior mean $E_{\pi(A)}(A)$ is $2/200$. log(CRPS) for the model parameters and weight trajectory (left) and ESS for the model parameters (right), as the mean over the 15, 53 and 32 model fits for each weight trajectory group (Δw_{low}, Δw_{mid} and Δw_{high}) and value of the noise. w represents the whole weight trajectory. We compute the average of the CRPS, and use the logarithm when visualizing the results. 4 multiplicative models with σ_w fixed to $0.1 \cdot E_{\pi(A)}(A)$ crashed during the inference.

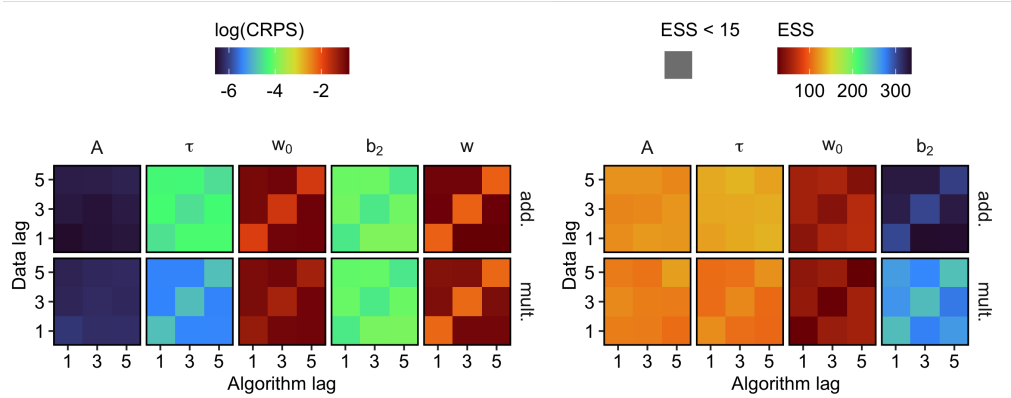


Fig 9. Performance for different lag values. Different lags used in the algorithm (x-axis) and simulated data (y-axis) (all combinations of lag 1, 3 and 5). log(CRPS) for the model parameters and weight trajectory (left) and ESS for the model parameters (right), as the mean over 100 simulated datasets for each scenario (lag value in data) and algorithm setting (lag value used by the algorithm). w represents the whole weight trajectory. We compute the average of the CRPS, and use the logarithm when visualizing the results. No model fits crashed during the inference.

Value of weight trajectory noise σ_w^2

We studied how influential the fixed value of σ_w is on the inference (Fig 8). A too high value leads to poor performance, especially with the multiplicative learning rule, while too low values will remove the randomness of the weight trajectory. Lower values perform better. We chose $\sigma_w = E_{\pi(A)}(A) \cdot 1/3$, since the random walk is assumed to be Gaussian distributed, and 3 standard deviations covers more than 99.7% of the distribution. This means that in the prior model, the noise is almost never larger than the contribution of A, and the trend is mainly driven by the learning rule.

Different lag values

The correct algorithm lag setting (equal to the lag in the simulated data) performs better than using the wrong setting (different than the lag in the data) (Fig 9). This is as expected and intended. Note that we have simulated 100 new datasets with each lag (1, 3, 5), in total 300 datasets here, with the same values for A, τ, w_0, b_1, b_2, σ_w, time series length and learning rule as before.

Model settings and model priors

We chose algorithm settings (Table 2) and model priors (Fig 10) based on the simulated case study for algorithm verification. Every $U = 100$ iteration, we adapt the variance of the proposal distributions according to the procedure of [58], using the newest $H = 100$ samples. The proposal standard deviation of w_0 is restricted to be minimum 0.1 to ensure satisfactory mixing.

Model performance in various scenarios

We investigated how the model with the chosen settings (Table 2) and priors (Fig 10) with both additive and multiplicative learning rule performs in various scenarios.

We varied the values of the true (simulated) A, τ, w_0, σ_w, length of time series, b_1 and b_2, covariate inclusion and which learning rule is used, when generating new

Table 2. Algorithm settings and prior distributions.

Parameter	Value	Prior distribution	Proposal distribution
P	200	-	-
A	-	$A \sim \text{Gamma}(2, 200)$	$A^{(i)} \sim \text{Gamma}(\alpha_A^{(i)}, \alpha_A^{(i)}/A^{(i-1)})$
τ	-	$\tau \sim \text{Gamma}(4, 100)$	$\tau^{(i)} \sim \text{Gamma}(\alpha_\tau^{(i)}, \alpha_\tau^{(i)}/\tau^{(i-1)})$
$w_0,\ w_t > 0$	-	$w_0 \sim \mathcal{N}_{\geq 0}(1, 5^2)$	$w_0^{(i)} \sim \mathcal{N}_{\geq 0}(w_0^{(i-1)}, {\sigma_{w_0}^2}^{(i)})$
$w_0,\ w_t < 0$	-	$w_0 \sim \mathcal{N}_{\leq 0}(-1, 5^2)$	$w_0^{(i)} \sim \mathcal{N}_{\leq 0}(w_0^{(i-1)}, {\sigma_{w_0}^2}^{(i)})$
b_2	-	$b_2 \sim \mathcal{N}(0, 10^2)$	$b_2^{(i)} \sim \mathcal{N}(b_2^{(i-1)}, {\sigma_{b_2}^2}^{(i)})$
β_k	-	$\beta_k \sim \mathcal{N}(0, 5^2)$	$\beta_k^{(i)} \sim \mathcal{N}(\beta_k^{(i-1)}, {\sigma_{\beta_k}^2}^{(i)})$
σ_w	$\frac{1}{3}\text{E}_{\pi(A)}(A) \approx 0.0033$	-	-

${\sigma_{w_0}^2}^{(i)}$ is restricted to be ≥ 0.1 to ensure that the Markov chains don't get stuck. $\mathcal{N}(\mu, \sigma^2)$ is the Gaussian distribution with expected value μ and variance σ^2. $\mathcal{N}_{\geq 0}()$ is the truncated Gaussian distribution, truncated at zero to only positive numbers.

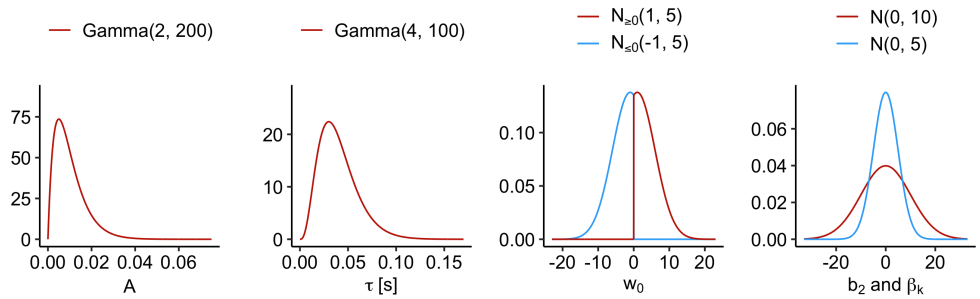


Fig 10. The prior distributions used for the model parameters in the inference. $b_2 \sim \mathcal{N}(0, 10^2)$ and $\beta_k \sim \mathcal{N}(0, 5^2)$, and are displayed in the same graph for comparison (right).

datasets. We simulated 10 datasets for each setting. The covariate was simulated as a sine function with period 6.28 seconds and repeated for the length of the time series. We evaluated the performance using CRPS and ESS of the model parameters (Fig 11 and Fig 12).

A good performance means low CRPS and high ESS. Bad performance means high CRPS and/or low ESS. The additive model is in general performing equally good or better than the multiplicative model, but the differences are bigger between normal and more extreme scenarios than between the different learning rules.

As long as the value of A is reasonable (less than 0.1), the model is performing steady and well (Fig 11A). The model is performing slightly better for intermediate values of τ than for very high or very low values (Fig 11B). The additive model is better at estimating A for large initial values of weight trajectory, w_0, and the ESS for w_0 is higher for higher values of true w_0, but otherwise not much changes (Fig 11C). For small values of σ_w, the model is performing well, and for $\sigma_w = 0.01$ the model performs worse with both additive and multiplicative learning rules (Fig 11D). The model is performing better for longer time series (Fig 11E). Higher spike rates of both s_1 and s_2 leads to better model performance, but high spike rates of only one of them yields worse performance (Fig 12A). Inclusion of covariates improves estimate accuracy of the learning rule parameters slightly, but leads to lower ESS (Fig 12B). The additive and multiplicative models are both able to recover the true (simulated) values for data simulated with both learning rules (Fig 12C).

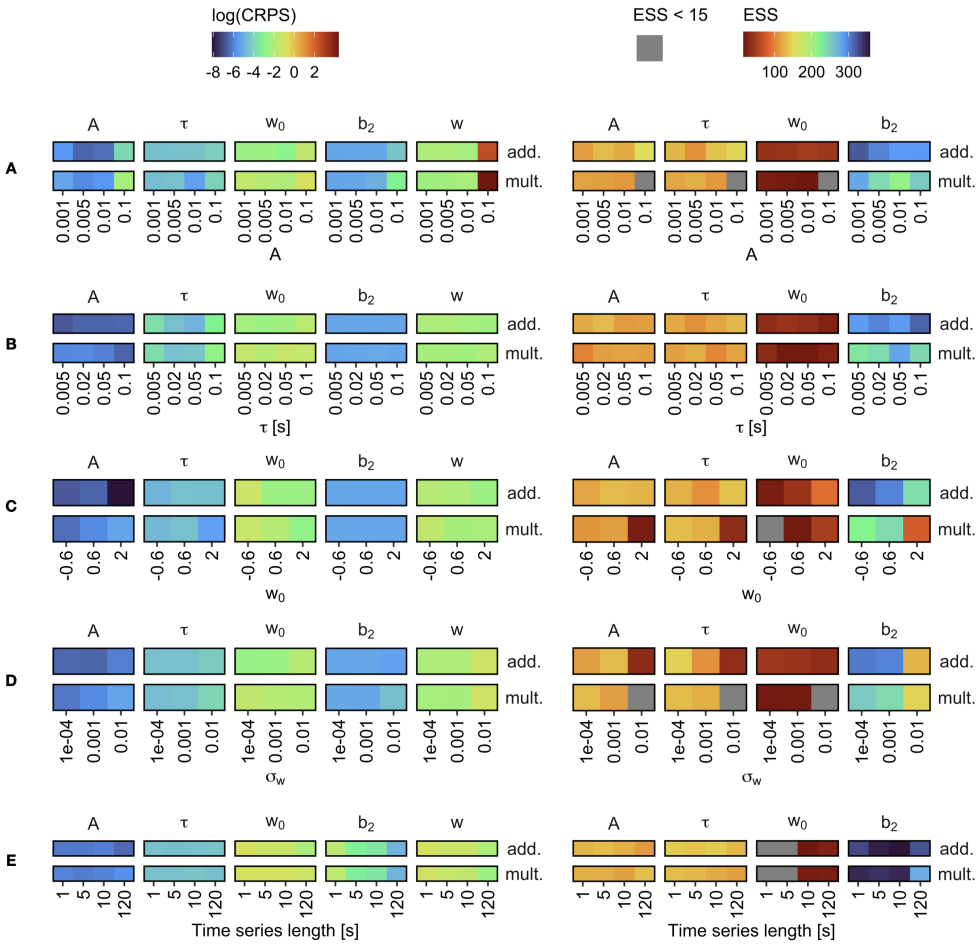


Fig 11. Algorithm performance in different scenarios. log(CRPS) for the model parameters and weight trajectory (left) and ESS for the model parameters (right), as the median over 10 simulated datasets. Different parameters/effects of true (simulated) dataset. We use the chosen settings (Table 2) and fit models with both additive and multiplicative learning rules (corresponding to rows). w represents the whole weight trajectory. We use the logarithm of the CRPS when visualizing the results. **A)** Different values of A (learning rule parameter). 1 multiplicative model with $A = 0.1$ crashed during the inference. **B)** Different values of τ (learning rule parameter). **C)** Different values of w_0 (initial value of weight trajectory). **D)** Different values of σ_w (standard deviation of weight trajectory noise). **E)** Different length of the time series.

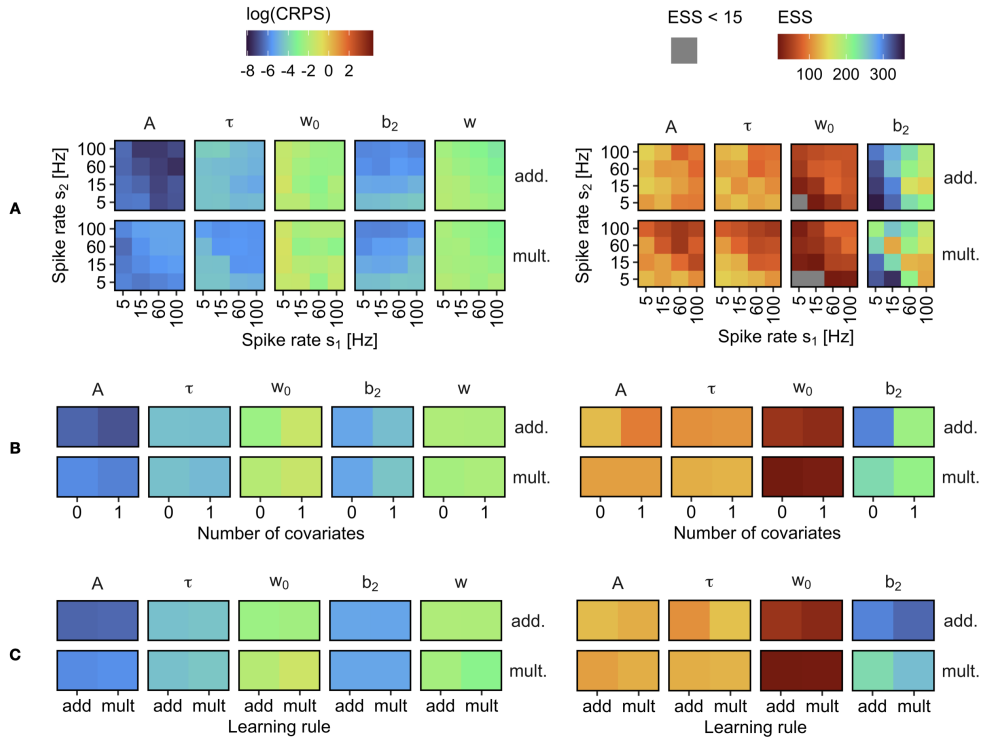


Fig 12. Algorithm performance in different scenarios. log(CRPS) for the model parameters and weight trajectory (left) and ESS for the model parameters (right), as the median over 10 simulated datasets. Different parameters/effects of true (simulated) dataset. We use the chosen settings (Table 2) and fit models with both additive and multiplicative learning rules (corresponding to rows). w represents the whole weight trajectory. We use the logarithm of the CRPS when visualizing the results. **A)** Different values of spike rate for both neurons 1 and 2 (i.e., different values of b_1 and b_2). 1 multiplicative model with spike rate of s_1 equal to 100 and spike rate of s_2 equal to 60 crashed during the inference. **B)** Without (0) and with (1) one covariate. The covariate is simulated as a sine function with period 6.28 seconds and repeated. We use $\beta_1 = 0.5$. **C)** Additive and multiplicative learning rules for the simulated data.

Supporting information

The supporting information can be found at the end of the manuscript.

S1 Appendix. Algorithms and mathematical details.

S1 Fig. Tail probabilities from temporal correlation investigation on simulated data.

S2 Fig. CRPS and ESS for all parameters from scenarios with increasing values of $b_1 = b_2$ and time series length T.

S3 Fig. Posterior weight trajectories and learning rules for both learning rule rypes and all covariate sets.

S4 Fig. Posterior learning rules for all model fits.

S5 Fig. Sorted learning rules, for the model with additive learning rule and no covariates.

References

1. Yankner BA. Mechanisms of neuronal degeneration in Alzheimer's disease. Neuron. 1996;16(5):921–932.
2. Moser EI, Krobert KA, Moser MB, Morris RG. Impaired spatial learning after saturation of long-term potentiation. Science. 1998;281(5385):2038–2042.
3. Whitlock JR, Heynen AJ, Shuler MG, Bear MF. Learning Induces Long-Term Potentiation in the Hippocampus. Science. 2006;313(5790):1093–1097. doi:10.1126/science.1128134.
4. Zott B, Simon MM, Hong W, Unger F, Chen-Engerer HJ, Frosch MP, et al. A vicious cycle of β amyloid–dependent neuronal hyperactivation. Science. 2019;365(6453):559–565.
5. Wong M. Modulation of dendritic spines in epilepsy: Cellular mechanisms and functional implications. Epilepsy & Behavior. 2005;7(4):569–577. doi:https://doi.org/10.1016/j.yebeh.2005.08.007.
6. Bell B, Lin JJ, Seidenberg M, Hermann B. The neurobiology of cognitive disorders in temporal lobe epilepsy. Nature Reviews Neurology. 2011;7(3):154–164. doi:10.1038/nrneurol.2011.3.
7. Moscovitch M, Cabeza R, Winocur G, Nadel L. Episodic Memory and Beyond: The Hippocampus and Neocortex in Transformation. Annual Review of Psychology. 2016;67(1):105–134. doi:10.1146/annurev-psych-113011-143733.
8. Bassett DS, Bullmore ET. Human brain networks in health and disease. Current opinion in neurology. 2009;22(4):340.
9. Markram H, Lübke J, Frotscher M, Sakmann B. Regulation of Synaptic Efficacy by Coincidence of Postsynaptic APs and EPSPs. Science. 1997;275(5297):213–215. doi:10.1126/science.275.5297.213.

10. Abbott LF, Nelson SB. Synaptic plasticity: taming the beast. Nature Neuroscience. 2000;3(11):1178–1183. doi:10.1038/81453.

11. Song S, Miller KD, Abbott LF. Competitive Hebbian learning through spike-timing-dependent synaptic plasticity. Nature neuroscience. 2000;3(9):919–926.

12. Verhoog MB, Obermayer J, Kortleven CA, Wilbers R, Wester J, Baayen JC, et al. Layer-specific cholinergic control of human and mouse cortical synaptic plasticity. Nature Communications. 2016;7(1):12826. doi:10.1038/ncomms12826.

13. Yao H, Dan Y. Stimulus Timing-Dependent Plasticity in Cortical Processing of Orientation. Neuron. 2001;32(2):315–323. doi:https://doi.org/10.1016/S0896-6273(01)00460-3.

14. Jacob V, Brasier DJ, Erchova I, Feldman D, Shulz DE. Spike timing-dependent synaptic depression in the in vivo barrel cortex of the rat. Journal of Neuroscience. 2007;27(6):1271–1284.

15. Brzosko Z, Mierau SB, Paulsen O. Neuromodulation of Spike-Timing-Dependent Plasticity: Past, Present, and Future. Neuron. 2019;103(4):563–581. doi:https://doi.org/10.1016/j.neuron.2019.05.041.

16. Kobayashi R, Kurita S, Kurth A, Kitano K, Mizuseki K, Diesmann M, et al. Reconstructing neuronal circuitry from parallel spike trains. Nature Communications. 2019;10(1):4468. doi:10.1038/s41467-019-12225-2.

17. Truccolo W, Eden UT, Fellows MR, Donoghue JP, Brown EN. A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects. Journal of neurophysiology. 2005;93(2):1074–1089.

18. Pillow JW, Shlens J, Paninski L, Sher A, Litke AM, Chichilnisky E, et al. Spatio-temporal correlations and visual signalling in a complete neuronal population. Nature. 2008;454(7207):995–999.

19. Roudi Y, Dunn B, Hertz J. Multi-neuronal activity and functional connectivity in cell assemblies. Current opinion in neurobiology. 2015;32:38–44.

20. Zaytsev YV, Morrison A, Deger M. Reconstruction of recurrent synaptic connectivity of thousands of neurons from simulated spiking activity. Journal of Computational Neuroscience. 2015;39(1):77–103. doi:10.1007/s10827-015-0565-5.

21. Stevenson IH, Rebesco JM, Hatsopoulos NG, Haga Z, Miller LE, Kording KP. Bayesian Inference of Functional Connectivity and Network Structure From Spikes. IEEE Transactions on Neural Systems and Rehabilitation Engineering. 2009;17(3):203–213. doi:10.1109/TNSRE.2008.2010471.

22. Gerwinn S, Macke J, Bethge M. Bayesian inference for generalized linear models for spiking neurons. Frontiers in Computational Neuroscience. 2010;4. doi:10.3389/fncom.2010.00012.

23. Chen Z, Putrino DF, Ghosh S, Barbieri R, Brown EN. Statistical Inference for Assessing Functional Connectivity of Neuronal Ensembles With Sparse Spiking Data. IEEE Transactions on Neural Systems and Rehabilitation Engineering. 2011;19(2):121–135. doi:10.1109/TNSRE.2010.2086079.

24. Mishchenko Y, Vogelstein JT, Paninski L. A Bayesian approach for inferring neuronal connectivity from calcium fluorescent imaging data. The Annals of Applied Statistics. 2011;5(2B):1229 – 1261. doi:10.1214/09-AOAS303.

25. Linderman S, Stock CH, Adams RP. A framework for studying synaptic plasticity with neural spike train data. Advances in neural information processing systems. 2014;27.

26. Chen S, Yang Q, Lim S. Efficient inference of synaptic plasticity rule with Gaussian process regression. Iscience. 2023;.

27. Wu Y, Moon J, Zhu X, Lu WD. Neural Functional Connectivity Reconstruction with Second-Order Memristor Network. Advanced Intelligent Systems. 2021;3(8):2000276. doi:https://doi.org/10.1002/aisy.202000276.

28. Jun JJ, Steinmetz NA, Siegle JH, Denman DJ, Bauza M, Barbarits B, et al. Fully integrated silicon probes for high-density recording of neural activity. Nature. 2017;551(7679):232–236. doi:10.1038/nature24636.

29. Boran E, Hilfiker P, Stieglitz L, Sarnthein J, Klaver P. Persistent neuronal firing in the medial temporal lobe supports performance and workload of visual working memory in humans. NeuroImage. 2022;254:119123.

30. Hedden T, Gabrieli JDE. Insights into the ageing mind: a view from cognitive neuroscience. Nature Reviews Neuroscience. 2004;5(2):87–96. doi:10.1038/nrn1323.

31. Gneiting T, Raftery AE. Strictly Proper Scoring Rules, Prediction, and Estimation. Journal of the American Statistical Association. 2007;102(477):359–378. doi:10.1198/016214506000001437.

32. Stark E, Abeles M. Unbiased estimation of precise temporal correlations between spike trains. Journal of neuroscience methods. 2009;179(1):90–100.

33. Goel PK, Degroot MH. Information About Hyperparameters in Hierarchical Models. Journal of the American Statistical Association. 1981;76(373):140–147. doi:10.1080/01621459.1981.10477620.

34. Davies M, Srinivasa N, Lin TH, Chinya G, Cao Y, Choday SH, et al. Loihi: A Neuromorphic Manycore Processor with On-Chip Learning. IEEE Micro. 2018;38(1):82–99. doi:10.1109/MM.2018.112130359.

35. Patel D, Hazan H, Saunders DJ, Siegelmann HT, Kozma R. Improved robustness of reinforcement learning policies upon conversion to spiking neuronal network platforms applied to Atari Breakout game. Neural Networks. 2019;120:108–115. doi:https://doi.org/10.1016/j.neunet.2019.08.009.

36. Vigneron A, Martinet J. A critical survey of STDP in Spiking Neural Networks for Pattern Recognition. In: 2020 International Joint Conference on Neural Networks (IJCNN); 2020. p. 1–9.

37. Yusoff N, Grüning A. Biologically Inspired Temporal Sequence Learning. Procedia Engineering. 2012;41:319–325. doi:https://doi.org/10.1016/j.proeng.2012.07.179.

38. Diehl P, Cook M. Unsupervised learning of digit recognition using spike-timing-dependent plasticity. Frontiers in Computational Neuroscience. 2015;9. doi:10.3389/fncom.2015.00099.

39. Ferré P, Mamalet F, Thorpe SJ. Unsupervised Feature Learning With Winner-Takes-All Based STDP. Frontiers in Computational Neuroscience. 2018;12. doi:10.3389/fncom.2018.00024.

40. Gilra A, Gerstner W. Non-linear motor control by local learning in spiking neural networks. In: Dy J, Krause A, editors. Proceedings of the 35th International Conference on Machine Learning. vol. 80 of Proceedings of Machine Learning Research. PMLR; 2018. p. 1773–1782. Available from: `https://proceedings.mlr.press/v80/gilra18a.html`.

41. Xiang S, Zhang Y, Gong J, Guo X, Lin L, Hao Y. STDP-Based Unsupervised Spike Pattern Learning in a Photonic Spiking Neural Network With VCSELs and VCSOAs. IEEE Journal of Selected Topics in Quantum Electronics. 2019;25(6):1–9. doi:10.1109/JSTQE.2019.2911565.

42. Chen K, Johnson A, Scott EO, Zou X, De Jong KA, Nitz DA, et al. Differential Spatial Representations in Hippocampal CA1 and Subiculum Emerge in Evolved Spiking Neural Networks. In: 2021 International Joint Conference on Neural Networks (IJCNN); 2021. p. 1–8.

43. Ghanbari A, Malyshev A, Volgushev M, Stevenson IH. Estimating short-term synaptic plasticity from pre-and postsynaptic spiking. PLoS computational biology. 2017;13(9):e1005738.

44. Morrison A, Diesmann M, Gerstner W. Phenomenological models of synaptic plasticity based on spike timing. Biological cybernetics. 2008;98(6):459–478.

45. Yang R, Huang HM, Hong QH, Yin XB, Tan ZH, Shi T, et al. Synaptic Suppression Triplet-STDP Learning Rule Realized in Second-Order Memristors. Advanced Functional Materials. 2018;28(5):1704455. doi:https://doi.org/10.1002/adfm.201704455.

46. Dunn B, Roudi Y. Learning and inference in a nonequilibrium Ising model with hidden nodes. Physical Review E. 2013;87(2):022127.

47. Battistin C, Dunn B, Roudi Y. Learning with unknowns: analyzing biological data in the presence of hidden variables. Current Opinion in Systems Biology. 2017;1:122–128.

48. Kumar A, Foster TC. Enhanced Long-Term Potentiation During Aging Is Masked by Processes Involving Intracellular Calcium Stores. Journal of Neurophysiology. 2004;91(6):2437–2444. doi:10.1152/jn.01148.2003.

49. Orock A, Logan S, Deak F. Age-related cognitive impairment: role of reduced synaptobrevin-2 levels in deficits of memory and synaptic plasticity. The Journals of Gerontology: Series A. 2020;75(9):1624–1632.

50. Mostany R, Anstey JE, Crump KL, Maco B, Knott G, Portera-Cailliau C. Altered Synaptic Dynamics during Normal Brain Aging. Journal of Neuroscience. 2013;33(9):4094–4104. doi:10.1523/JNEUROSCI.4825-12.2013.

51. Bi Gq, Poo Mm. Synaptic Modifications in Cultured Hippocampal Neurons: Dependence on Spike Timing, Synaptic Strength, and Postsynaptic Cell Type. Journal of Neuroscience. 1998;18(24):10464–10472. doi:10.1523/JNEUROSCI.18-24-10464.1998.

52. Zhang LI, Tao HW, Holt CE, Harris WA, Poo Mm. A critical window for cooperation and competition among developing retinotectal synapses. Nature. 1998;395(6697):37–44. doi:10.1038/25665.

53. Debanne D, Gähwiler BH, Thompson SM. Long-term synaptic plasticity between pairs of individual CA3 pyramidal cells in rat hippocampal slice cultures. The Journal of physiology. 1998;507(Pt 1):237.

54. Feldman DE. Timing-Based LTP and LTD at Vertical Inputs to Layer II/III Pyramidal Cells in Rat Barrel Cortex. Neuron. 2000;27(1):45–56. doi:https://doi.org/10.1016/S0896-6273(00)00008-8.

55. Chib S, Greenberg E. Understanding the Metropolis-Hastings Algorithm. The American Statistician. 1995;49(4):327–335. doi:10.1080/00031305.1995.10476177.

56. Casella G, George EI. Explaining the Gibbs Sampler. The American Statistician. 1992;46(3):167–174.

57. N J Gordon AFMS D J Salmond. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. IEE Proceedings F (Radar and Signal Processing). 1993;140:107–113(6).

58. Haario H, Saksman E, Tamminen J. Adaptive proposal distribution for random walk Metropolis algorithm. Computational statistics. 1999;14(3):375–395.

59. Plummer M, Best N, Cowles K, Vines K. `CODA`: Convergence Diagnosis and Output Analysis for MCMC. `R` News. 2006;6(1):7–11.

60. `R` Core Team. `R`: A Language and Environment for Statistical Computing; 2022. Available from: `https://www.R-project.org/`.

61. Jordan A, Krüger F, Lerch S. Evaluating Probabilistic Forecasts with `scoringRules`. Journal of Statistical Software. 2019;90(12):1–37. doi:10.18637/jss.v090.i12.

62. Langsrud A. Inferring the learning rule from spike train data with particle Metropolis-Hastings; 2020. Available from: `https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2778344`.

63. Myhre EA. Bayesian optimal experimental design for studying synaptic plasticity; 2021. Available from: `https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2827066`.

64. Cappé O, Moulines E, Rydén T. Inference in Hidden Markov Models. Springer Science & Business Media Inc.; 2005.

65. Martino L, Elvira V, Louzada F. Effective sample size for importance sampling based on discrepancy measures. Signal Processing. 2017;131:386–401.

66. Fuglstad GA, Hem IG, Knight A, Rue H, Riebler A. Intuitive joint priors for variance parameters. Bayesian Analysis. 2020;15(4):1109–1137.

67. Gelman A, Vehtari A, Simpson D, Margossian CC, Carpenter B, Yao Y, et al.. Bayesian Workflow; 2020. Available from: `https://arxiv.org/abs/2011.01808`.

68. Lambert PC, Sutton AJ, Burton PR, Abrams KR, Jones DR. How vague is vague? A simulation study of the impact of the use of vague prior distributions in MCMC using WinBUGS. Statistics in Medicine. 2005;24(15):2401–2428.

69. Gelman A, Simpson D, Betancourt M. The prior can often only be understood in the context of the likelihood. Entropy. 2017;19(10):555.

Appendix S1 - Algorithms and mathematical details

In the following, we explain the details of our Metropolis-Hastings algorithm. See e.g. [55] for a general introduction to Metropolis-Hastings. This algorithm is an extended version of the algorithm presented in [62] and [63], which again are based on the model and inference methods proposed by [25].

The overall procedure follows Algorithm S1.1, and consists of several sub-algorithms. With β_k, we refer to the parameter(s) of the covariate(s), if any. For models without covariates, this parameter is not included. Ga() is a short-hand for Gamma(). $\mathcal{N}()$ is the Gaussian distribution.

Algorithm S1.1 Main algorithm for Bayesian inference of learning rule parameters

Initialize $(b_2^{(0)}, \beta_k^{(0)}, A^{(0)}, \tau^{(0)}, w_0^{(0)})$
for $i = 1, 2, \ldots$ **do**
  Sample $b_2^{(i)}$ [S1.2]
  Sample $\beta_k^{(i)}$ for all k [S1.2]
  Sample $\boldsymbol{\theta}^{(i)} = (A^{(i)}, \tau^{(i)}, w_t^{(i)})$ [S1.3]
end for

Algorithm S1.2 Sample B[a]

Adjust proposal variance $({\sigma_B^2}^{(i)})$ [S1.4]
Propose $B^* \sim \mathcal{N}(B^{(i-1)}, {\sigma_B^2}^{(i)})$
Compute $r = r_B(B^*, B^{(i-1)})$
Draw $u \sim \text{Uniform}[0, 1]$
if $u < r$ **then**
  Set $B^{(i)} = B^*$
else
  Set $B^{(i)} = B^{(i-1)}$
end if

[a] B represents b_2 and β_k for all k.

The acceptance ratio in a Metropolis-Hastings algorithm $r_B(B^*, B^{(i-1)})$ for a parameter B is given by:

$$r_B(B^*, B^{(i-1)}) = \frac{\text{likelihood}(B^*)}{\text{likelihood}(B^{(i-1)})} \times \frac{\text{prior}(B^*)}{\text{prior}(B^{(i-1)})} \times \frac{\text{proposal}(B^{(i-1)}|B^*)}{\text{proposal}(B^*|B^{(i-1)})}$$

Note that the observed data $\boldsymbol{s}$, prior and proposal hyperparameters, and the model parameters not involved in that particular part of the algorithm also appear in the expression through conditioning, but are omitted here for simplicity. Also note that when the proposal distribution is symmetric around the mean, the last term is equal to 1.

Algorithm S1.3 Sample A, τ and w_t

Adjust proposal variance and shape parameters ($\alpha_A^{(i)}$, $\alpha_\tau^{(i)}$, ${\sigma^2_{w_0}}^{(i)}$) [S1.4, S1.5]
Propose $A^* \sim \text{Ga}(\alpha_A^{(i)}, \alpha_A^{(i-1)}/A^{(i)})$
Propose $\tau^* \sim \text{Ga}(\alpha_\tau^{(i)}, \alpha_\tau^{(i-1)}/\tau^{(i)})$
Run particle filter [S1.6]
Compute $r = r_{\boldsymbol{\theta}}(\boldsymbol{\theta}^*, \boldsymbol{\theta}^{(i-1)})$

Draw $u \sim \text{Uniform}[0, 1]$

if $u < r$ **then**
  Set $\boldsymbol{\theta}^{(i)} = \boldsymbol{\theta}^*$ and $\boldsymbol{w}^{(i)} = \boldsymbol{w}^*$

else
  Set $\boldsymbol{\theta}^{(i)} = \boldsymbol{\theta}^{(i-1)}$ and $\boldsymbol{w}^{(i)} = \boldsymbol{w}^{(i-1)}$
end if

Algorithm S1.4 Adjust proposal variance for parameter ϕ, Gaussian distribution

if $(i \bmod U) = 0$ **then**
  Set ${\sigma^2_\phi}^{(i)} = \text{Var}(\phi^{(i-H):(i-1)}) \cdot 2.4^2$
else
  Set ${\sigma^2_\phi}^{(i)} = {\sigma^2_\phi}^{(i-1)}$
end if

Algorithm S1.5 Adjust proposal variance for parameter ϕ, gamma distribution

if $(i \bmod U) = 0$ **then**
  Set $\alpha_\phi^{(i)} = \frac{\text{Mean}(\phi^{(i-H):(i-1)})^2}{\text{Var}(\phi^{(i-H):(i-1)}) \cdot 2.4^2}$
else
  Set $\alpha_\phi^{(i)} = \alpha_\phi^{(i-1)}$
end if

The acceptance ratio for $\boldsymbol{\theta}$ is computed in a similar matter as for the parameter B, but since the weight trajectory w_t is unknown, we use a particle filter to estimate the trajectory and evaluate the likelihood (Algorithm S1.6).

Algorithm S1.6 Particle filter

Sample particles $(w_1^*)_{p=1}^P \sim \max(0, \mathcal{N}_P(w_0^{(i-1)}, \sigma_{w_0}^{2\ (i)}))^a$
Initialize particle weights $(v)_{p=1}^P = \frac{1}{P}$
Set likelihood$^* = 1$
for $t = a+1, \ldots, T$ **do**
  Set $(v_{\text{normalized}})_{p=1}^P = (v)_{p=1}^P / \sum (v)_{p=1}^P$
  Resample particles [S1.7]
  Compute learning rule $\text{lr} = l(s_{1,1:t}, s_{2,1:t}, \boldsymbol{\theta}^*)$
  Sample $\varepsilon_t \sim \mathcal{N}_P(0, \sigma_w^2)$
  Set $(w_t^*)_{p=1}^P = \max(0, (w_{t-1})_{p=1}^P + \text{lr} + \varepsilon_t)^b$
  Compute likelihood $\text{lik}_t = \text{Bern}(s_{2,t} | \text{expit}((w_t)_{p=1}^P s_{1,t-a} + b_2^{(i)}))$
  Set $(v)_{p=1}^P = (v_{\text{normalized}})_{p=1}^P \cdot \text{lik}_t$
  Set likelihood* = likelihood$^* \cdot \frac{1}{P} \sum v_{p=1}^P$
end for

[a] Or min, depending on whether the connection is excitatory (max) or inhibitory (min).
[b] See footnote *a*.

For each time step, w_t is given by

$$\sum_{p=1}^{P} \left(w_{t,p} \cdot \frac{v_p}{\sum_{p=1}^{P} v_p} \right),$$

which is the weighted mean of the particles for each time step. v_p are the particle weights after the last time step. This is the weight trajectory used to compute the likelihood when sampling b_2 and β_k for all k.

Some of the particles will have a low probability due to the random component (ε_t). To concentrate the particles around the most probable values and avoid degeneration of the particle filter [64], we resampled the particles whenever the variation in the particle weights become too big (Algorithm S1.7). We measured this variation using the perplexity of the particle weights, such that we resampled whenever

$$\frac{\exp\left\{ -\sum_{p=1}^{P} v_p \log(v_p) \right\}}{P}$$

became smaller than some threshold. [65] suggests to set this to about 0.66. In practice, we used $(w_t)_{p=1}^P$ and v_p to compute the average w_t for each timestep and store that, due to memory issues for long spike trains.

Algorithm S1.7 Resampling particles

Compute perplexity of $(v_{\text{normalized}})_{p=1}^P$
if Perplexity < 0.66 **then**
  Resample $(w_{t'})_{p=1}^P$ with probability $(v_{\text{normalized}})_{p=1}^P$ $\forall t' < t$
  Set $(v_{\text{normalized}})_{p=1}^P = \frac{1}{P}\mathbf{1}$
end if

After obtaining the likelihood from the particle filter the ratio $r_{\boldsymbol{\theta}}(\boldsymbol{\theta}^*, \boldsymbol{\theta}^{(i-1)})$ is used to determine whether the sample of $\boldsymbol{\theta}$ is accepted or not:

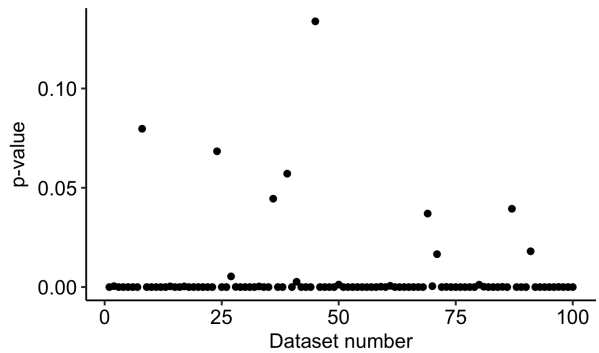
$$
\begin{aligned}
r_{\boldsymbol{\theta}}(\boldsymbol{\theta}^*, \boldsymbol{\theta}^{(i-1)}) = & \frac{\text{likelihood}^*}{\text{likelihood}^{(i-1)}} \\
& \times \frac{\text{Ga}(A^*|\alpha_A, \beta_A)}{\text{Ga}(A^{(i-1)}|\alpha_A, \beta_A)} \cdot \frac{\text{Ga}(\tau^*|\alpha_\tau, \beta_\tau)}{\text{Ga}(\tau^{(i-1)}|\alpha_\tau, \beta_\tau)} \cdot \frac{\mathcal{N}(w_0^*|1, 10^2)}{\mathcal{N}(w_0^{(i-1)}|1, 10^2)} \\
& \times \frac{\text{Ga}(A^{(i-1)}|\alpha_A^{(i)}, \alpha_A^{(i)}/A^*)}{\text{Ga}(A^*|\alpha_A^{(i-1)}, \alpha_A^{(i-1)}/A^{(i-1)})} \cdot \frac{\text{Ga}(\tau^{(i-1)}|\alpha_\tau^{(i)}, \alpha_\tau^{(i)}/\tau^*)}{\text{Ga}(\tau^*|\alpha_\tau^{(i-1)}, \alpha_\tau^{(i-1)}/\tau^{(i-1)})} \\
& \cdot \frac{\mathcal{N}(w_0^{(i-1)}|w_0^*, \sigma_{w_0}^{2\ (i)})}{\mathcal{N}(w_0^*|w_0^{(i-1)}, \sigma_{w_0}^{2\ (i-1)})}.
\end{aligned}
$$

Since a neuron is either excitatory or inhibitory, we restricted the weight trajectory to have the same sign for the whole time series. To ensure this, we used a truncated Gaussian distribution for w_0. To increase the speed of the algorithm, we did not do this for the weight trajectory, and instead set all negative values of w_t to 0, for excitatory connections, or positive values to 0, for inhibitory connections.

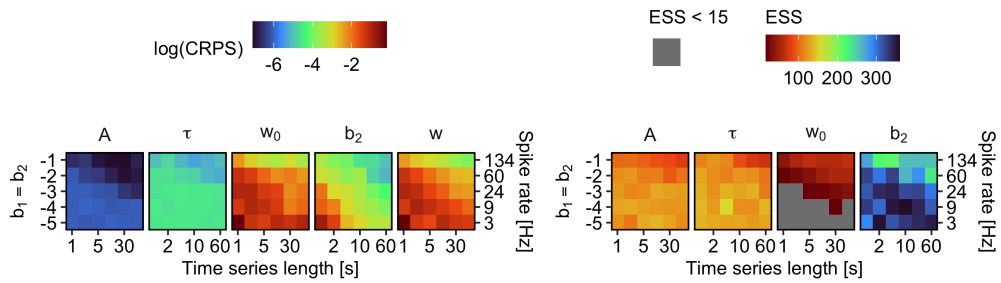
The prior distribution is a central component in a Bayesian framework. A good prior can yield more computationally efficient inference, and can contain prior knowledge about the problem at hand [66, 67]. It is however not always straight-forward to select prior distributions [68, 69].

The proposal distribution is also crucial. We used the prior distribution families as the proposals, but with different hyperparameters to increase the acceptance rate of the inference (Table 2). For all proposal distributions, we used the previous sample as the mean value, and we adjusted the variance according to the procedure in [58] to increase the number of samples that are accepted.

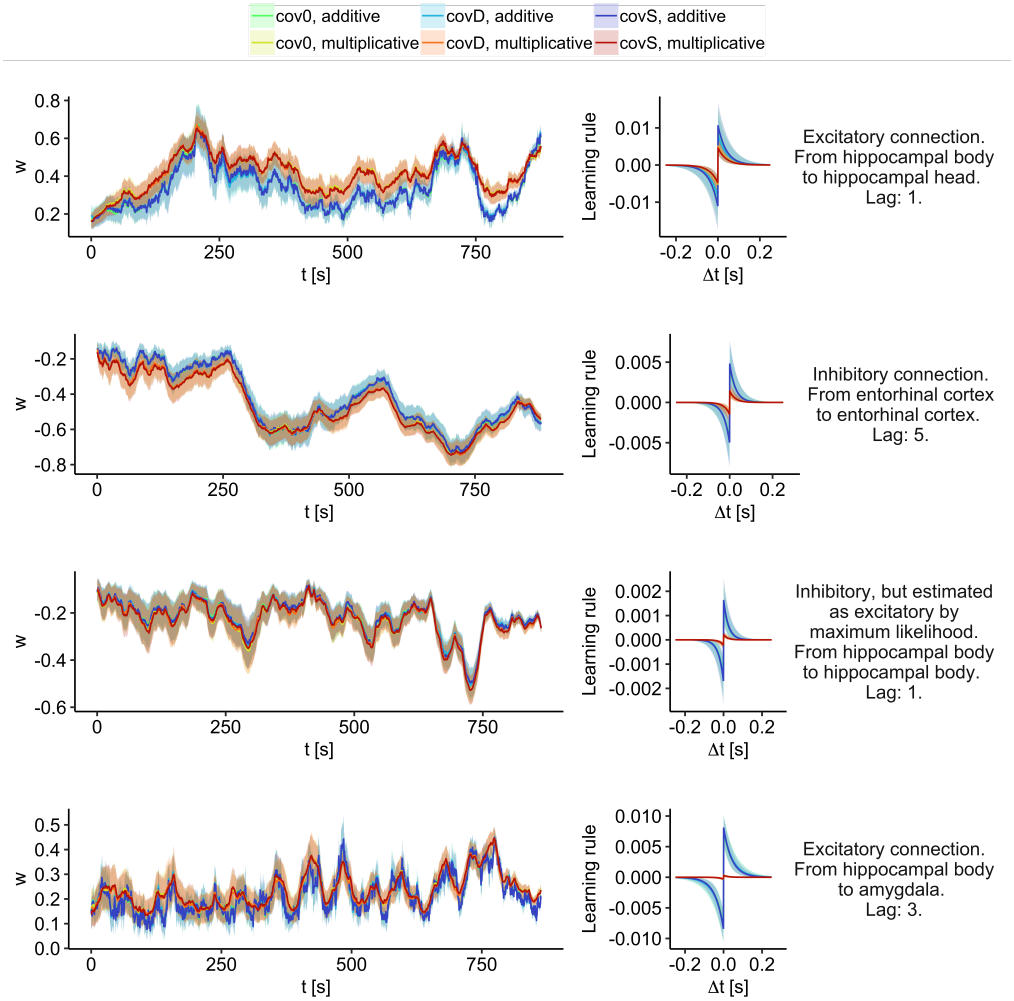
Supporting figures



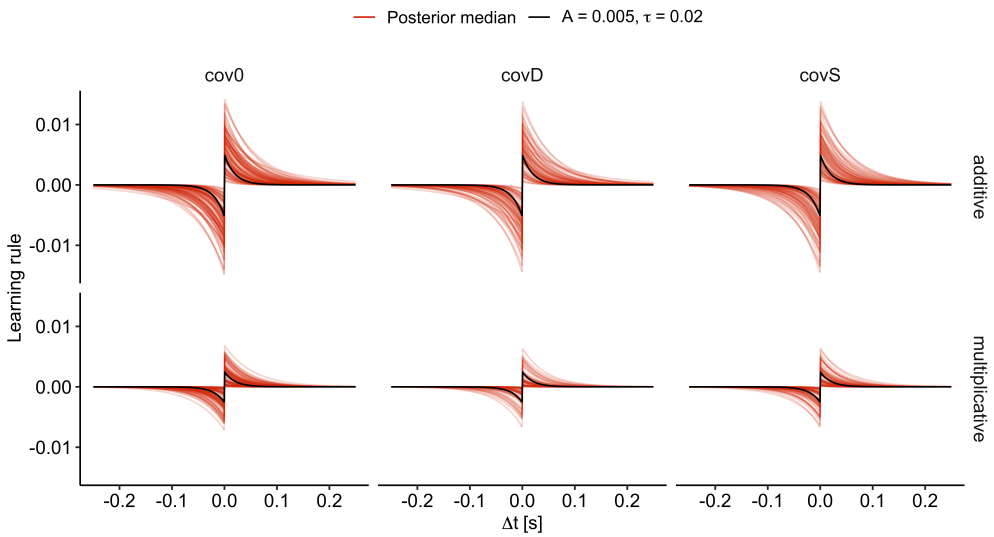
S1 Fig. Tail probabilities from temporal correlation investigation on simulated data. Tail probabilities for excitatory connections at time lag 1 from the temporal correlation between spike trains from the 100 simulated neuron pairs. The true time lag is equal to 1 time bin. The tail probability for excitatory connections is smaller than 0.025 at the correct lag in 93 of the 100 datasets.



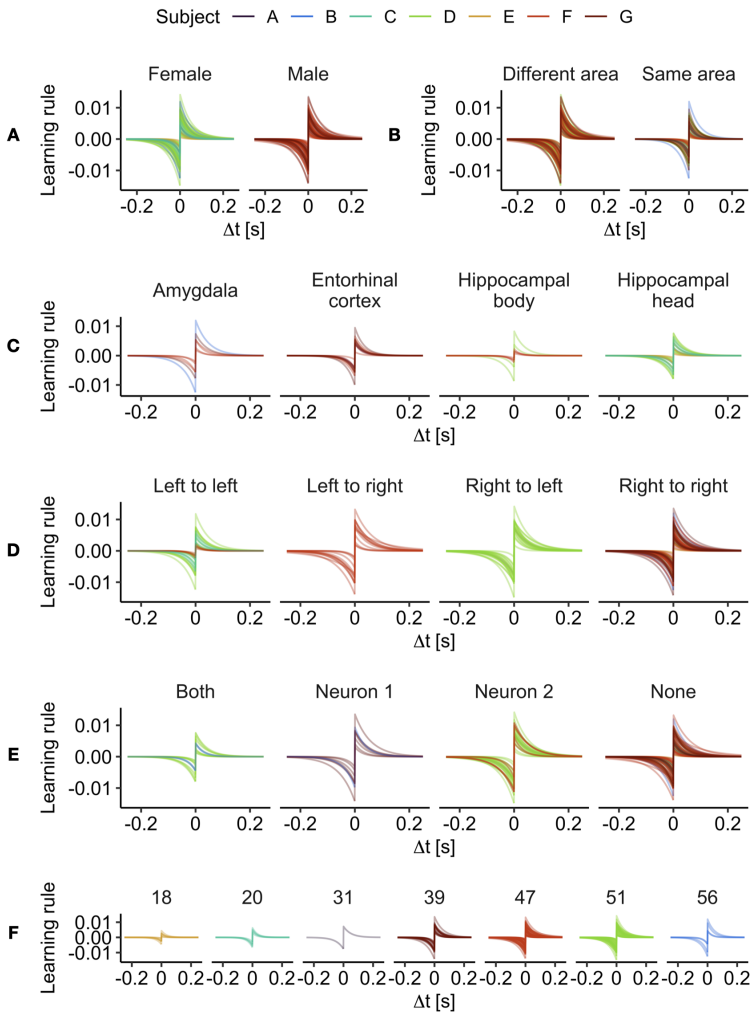
S2 Fig. CRPS and ESS for all parameters from scenarios with increasing values of $b_1 = b_2$ and time series length T. log(CRPS) for the model parameters and weight trajectory (left) and ESS for the model parameters (right), as the median over 10 simulated datasets for each combination of $b_1 = b_2$ and T. Additive learning rule only. w represents the whole weight trajectory. We use the logarithm of the CRPS when visualizing the results. For w_0, we get median effective samples less than 15 for nearly half of the scenarios (combination of low T and low $b_1 = b_2$). In longer time series with spike rates above 24 Hz, we get enough effective samples. No model fits crashed (stopped due to numerical errors) during the inference, and no model fits were removed even when the ESS was lower than 15. Longer datasets and higher spike rates give a lower (better) CRPS.



S3 Fig. Posterior weight trajectories and learning rules for both learning rule types and all covariate sets. Posterior weight trajectories (left) and corresponding learning rules (right) from four neuron pairs (one in each row, different subjects), for the two learning rules and three sets of covariates. Posterior weight trajectories and learning rules inferred using different covariates are almost perfectly overlaid, for the additive and multiplicative rule respectively. Connection types, brain areas and lags are indicated in the figure. The weight trajectories are displayed with posterior mean ± 1 standard deviation. The learning rules are based on posterior medians and corresponding uncertainties on lower and upper quartiles of A and τ. Note that the interpretation of the additive and multiplicative learning rules are slightly different.



S4 Fig. Posterior learning rules for all model fits. Posterior median of A and τ are used to visualize the learning rule for the model with additive (top) and multiplicative (bottom) learning rules, without covariates (cov0, left), with covariates describing the difficulty level of each trial (covD, middle) and covariates describing the stage of the trial (covS, right). We set $w_t = 0.5$ for all t for the multiplicative learning rule. This is plotted with an additive learning rule with parameters $A = 0.005$ and $\tau = 0.02$ (top) and multiplicative learning rule with $A = 0.005$, $\tau = 0.02$ and $w_t = 0.5$. Only learning rules from converged model fits are included. Note that the y-axis vary between panels.



S5 Fig. Sorted learning rules, for the model with additive learning rule and no covariates. Posterior median of A and τ are used to visualize the learning rule for the additive model without covariates (cov0). We sort them using different information about the data, that is not included in the inference. **A)** Gender of subject (male/female). **B)** Both neurons in a pair in the same or in two different areas. **C)** Of the neurons in the same area, which area are they located in. **D)** Which side of the brain neurons are located in (neuron 1 to neuron 2). **E)** Neurons located in seizure onset zones (both, neuron 1 ($\boldsymbol{s}_1$), neuron 2 ($\boldsymbol{s}_2$), or none). **F)** Age of subject.