

Mathematical Neuroscience - Project

Adaptive exponential integrate-and-fire model for in-vivo two-photon calcium imaging data

Matuš Halák - 2724858 (VU Student ID)

May - June 2025

1 Introduction

In this project, I was assigned the seminal 2005 paper by Brette & Gerstner [2], that introduced the Adaptive exponential integrate-and-fire model (AdEx). The authors showed that this simple two-dimensional model of 9 parameters could predict the spike trains, as well as, subthreshold dynamics of a detailed five-dimensions Hodgkin-Huxley-like conductance-based model of 31 parameters with very high accuracy (96% of spikes $\pm 2\text{ms}$). They compared the detailed and AdEx models across a range of artificially applied currents, some resembling in-vivo synaptic activity by Ornstein-Uhlenbeck processes and others resembling typical in-vitro experiments with step currents. I will talk in more detail about the parameters and resulting firing patterns in the AdEx model and the work that followed its introduction in section 2. The authors also presented a systematic procedure for parameter fitting of AdEx models to data (in their case, data from the detailed model), in case one is able to collect additional data on the ground-truth neuron using additional electrophysiological paradigms. In section 3 I will discuss why this procedure and similar procedures described in later papers are only applicable to in-vitro or patch-clamp experiments. The performance of AdEx relative to the detailed model was quantified with percentage of missing spikes, and extra spikes, as well as, a coincidence factor Γ [6] frequently used to quantify model performance in many subsequent papers. In section 3, I will discuss the challenges of applying the Γ metric to in-vivo data sampled at a lower frequency. Overall, [2] showed that the AdEx model can faithfully reproduce realistic spike trains from complex models, if both models are stimulated by the same input current, and if the complex model can be additionally experimentally probed using electro-physiological paradigms for extraction of AdEx parameters.

Crucially, the AdEx model does not have too many parameters and at the same time, all of its parameters are directly interpretable. This makes it a very attractive model for neuroscientists working with in-vivo data for the following reasons. Provided we are able to accurately fit the AdEx model to in-vivo experimental data from an observed neuron, we now have an in-sillico model of that neuron that we can investigate under a range of scenarios prohibitively large for real-life experiments. Crucially, though, because of the high parameter interpretability, an accurate AdEx fit also directly gives us insightful information about what *kind* of neuron we are looking at, in terms of the fitted parameters. In this way, fitting AdEx models to neurons is not only a way to obtain in-sillico models, but also a way to characterize neurons in terms of meaningful parameters that define AdEx neurons and that might be otherwise impossible to learn in-vivo.

As a neuroscientist, these features of the AdEx model really intrigued me and made me want to see if I could devise a method to allow extraction of AdEx parameters and generation of AdEx models from in-vivo data. Because I work with two-photon calcium imaging as a method to record single-neuron activity from hundreds of neurons at the same time (briefly discussed in section 3), I was aware that developing an AdEx model fitting procedure for calcium imaging data would be extremely useful and could be a way to model the local connectivity of the recorded neurons, which is otherwise difficult to infer. At the same time, I was aware of the challenges related to the nature of calcium imaging data that would make it much more difficult than the relatively streamlined parameter fitting described in the original publication. Nevertheless, I found it all the more exciting to at least try to see if this was possible.

This work is focused on parameter fitting, as well as, on adapting the AdEx model to account for connectivity and a fundamentally different type of signal than it is designed to fit ¹. Previous work has fully characterized the AdEx model in terms of its parameters and resulting firing patterns [7], as well as, subthreshold and spiking dynamics and bifurcations [8]. Therefore, due to my interest in my described research question, I drew from this previous work in my research instead of focusing on re-deriving it.

2 Adaptive exponential integrate-and-fire model

The original AdEx neuron was defined by Brette and Gerstner [2] ² as:

$$C \frac{dV}{dt} = -g_L(V - E_L) + g_L \Delta_T \exp\left(\frac{V - V_T}{\Delta_T}\right) - w + I \quad (1)$$

$$\tau_w \frac{dw}{dt} = a(V - E_L) - w \quad (2)$$

$$\text{At spike time } t (V(t) > V_{peak}) : V \rightarrow V_r \quad \text{and} \quad w \rightarrow w + b \quad (3)$$

This simple formulation allows the model to accumulate the adaptation variable w during high-frequency activity and thus adapt its firing to intense stimulation.

2.1 Model parameters

Parameter	Value	Unit	Comment
C	281.0	pF	Membrane capacitance
g_L	30.0	nS	Leak conductance
E_L	-70.6	mV	Leak reversal potential
V_T	-50.4	mV	Spike-initiation threshold potential
Δ_T	2.0	mV	Spike threshold slope
V_{peak}	0.0	mV	Spike cutoff
τ_w	144.0	ms	Adaptation time-constant
a	4.0	nS	Sub-threshold adaptation conductance
V_r	-70.6	mV	Reset potential after a spike
b	80.5	pA	Spike-triggered adaptation increment

Table 1: (Default) AdEx model parameters from Brette and Gerstner [2].

The AdEx model is defined by 9 parameters $C, g_L, E_L, V_T, \Delta_T, \tau_w, a, V_r, b$ (not counting V_{peak} which only affects the timing of spikes by μs ' and is always set to $V_{peak} = 0$ because at larger values the exponential can blow-up and introduce errors into simulations). Default parameters from Brette and Gerstner [2] are shown in Table 1. The work of Naud et al. [7] and Touboul and Gerstner [8], shows that we can separate the parameters into five "scaling" parameters and four "bifurcation" parameters.

Scaling parameters include membrane capacitance C , leak conductance g_L , leak reversal potential E_L , spike threshold potential V_T and spike threshold slope Δ_T . Touboul and Gerstner [8] show that the AdEx

¹In this project, unless otherwise specified, all analyses and algorithms we custom-made in Python (or sometimes to increase efficiency, implemented in C via Cython). Creating this whole framework for modeling calcium imaging data in a scalable and efficient way was at times a significant challenge but I learned a lot through the process.

²In their paper, Brette & Gerstner also discuss a version with two synaptic currents instead of input current, one excitatory current for all excitatory synapses with an "excitatory" conductance g_e and "excitatory synaptic reversal potential" E_e ; and one inhibitory current for all inhibitory synapses with "inhibitory" conductance g_i and "inhibitory synaptic reversal" E_i , in which case they use a modified version of Equation 1: $C \frac{dV}{dt} = -g_L(V - E_L) + g_L \Delta_T \exp\left(\frac{V - V_T}{\Delta_T}\right) - g_e(t)(V - E_e) - g_i(t)(V - E_i) - w$. I do not discuss this formulation of the model further, but I drew inspiration from it when defining my model with synapses.

model can be reformulated in dimensionless units where all scaling parameters are absorbed and only four (bifurcation) parameters and the input current are retained. Namely, C and g_L can be absorbed into the membrane time constant $\tau_m = C/g_L$, which is used to re-scale the time units, and E_L, V_T and Δ_T can be used to re-scale and offset the other variables. For this reason, these variables do not determine the neuron's firing patterns, only the temporal / amplitude scaling of its responses.

Bifurcation parameters of the AdEx model are the reset potential V_r , the adaptation time-constant τ_w , subthreshold adaptation "conductance" a and spike-triggered adaptation b . These are the "interesting" AdEx parameters that determine the neuron's firing pattern and that one would like to extract. Figure 1 shows the result of a 1000 millisecond simulation I performed to investigate the effects of the bifurcation parameters on the neuron's firing pattern to identical pulsed input. Pulsed current of 1000 pA was applied every 100 milliseconds for 60 milliseconds ON and 40 milliseconds OFF. Each parameter is varied in isolation with all other parameters kept at default values from Table 1. As such, each situation should be compared with Figure 1a, which shows the response of the "default" neuron to the pulsed current.

- V_r . The reset potential V_r clearly has drastic influence on the neuron's firing pattern. At low V_r , after each spike the voltage drops far from below the spike initiation zone, and the model needs to "climb" back up in voltage before another spike can be initiated (Figure 1a). At high V_r , namely, at $V_r > V_T$, the situation is very different, because after emitting a spike the neuron is put right back into the spiking zone, where the growth in voltage is exponential - eliciting another spike. As long the current pulse is on, this keeps repeating, despite ever-increasing adaptation drive w , resulting in high-frequency bursting (Figure 1b). Interesting to note, is that due to the rapid accumulation of w during the quick burst, once the current pulse is switched off, the voltage plummets and because the adaptation variable is relatively slow in the default configuration, the response to the next pulse is reduced to subthreshold level. However, because no spike is emitted in the pulse following the burst pulse, w just keeps decreasing and by the next pulse, it is low enough that the pulse takes the voltage back into the bursting regime.
- τ_w . The adaptation time-constant has clearly noticeable effects on the evolution of the adaptation variable w , and as a result on the generated spike train. The closer τ_w is to 1, the more the evolution of the w variable starts to resemble the voltage, because the w can now respond to voltage faster and the separation of timescales is reduced. As a consequence, w starts responding very quickly to the post-spike voltage reset, quickly decreasing itself, and opening up the possibility for more spikes (Figure 1c). On the other hand at very high τ_w the situation is reversed. Now the adaptation is so slow, that it keeps building up and accumulating over the subsequent current pulses, leading to lower and lower spiking, until w is so high, that there is a current pulse when no spike occurs, which gives it enough time to decrease just enough to allow one spike (Figure 1c).
- a. The sub-threshold adaptation conductance can make neurons both more or less excitable based on its magnitude and sign. If a is large and negative, the effect of this is that w now acts as positive feedback (instead of adaptation) and increases voltage, leading to high frequency spiking during the current pulse (since every spike leads to an increase in w and if $a < 0$) high w makes spikes more likely) (Figure 1e). On the contrary if a is large and positive, it enhances the magnitude of the adaptation / negative feedback and reduces the neuron's excitability (Figure 1h). Important thing to note that is not clearly visible in this stimulation paradigm, is that high magnitude of a leads to the adaptation variable responding more strongly to subthreshold fluctuations in voltage, whereas values closer to 0 lead to linear decay after a spike.
- b. Finally, the spike-triggered adaptation controls how much w is incremented after each spike. Values of b closer to zero barely increment the adaptation variable after a spike and actually most of the increase in adaptation comes from the subthreshold adaptation a during the rising phase after each post-spike reset (Figure 1g). High values of b lead to massive adaptation increments that take a long time to overcome, even at the moderate default τ_w (Figure 1h).

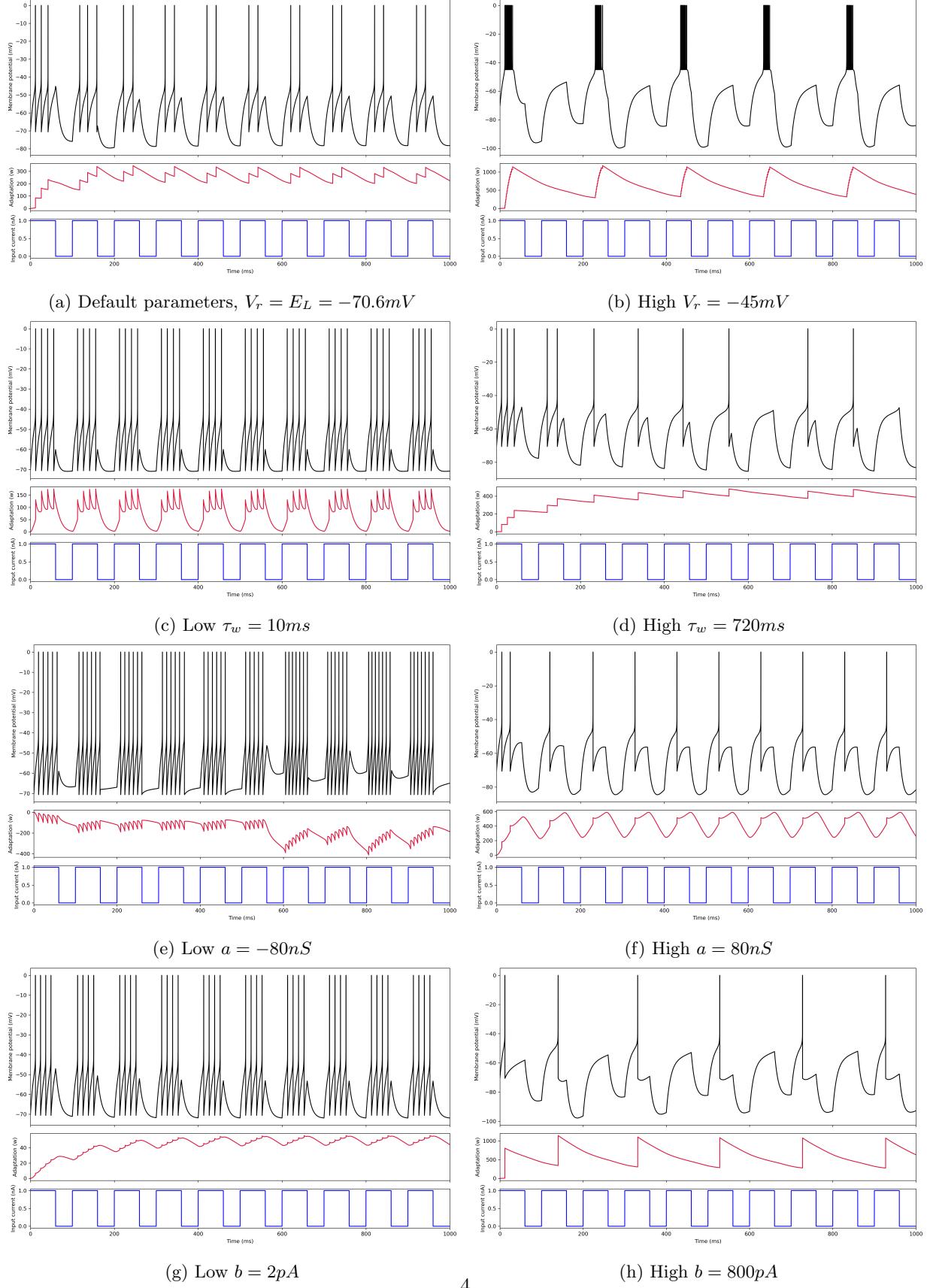


Figure 1: Bifurcation parameters that determine firing pattern of AdEx model. Black - Voltage (mV), Red - Adaptation variable w , Blue - Applied current (nA). Low (Left) and high (Right) parameter values for V_r, τ_w, a, b respectively top-bottom.

2.2 AdEx Phase plane

To get further insight on the firing patterns and to find a systematic way to classify the firing patterns of AdEx neurons, we can use phase-plane analysis, as has been extensively done after the conceptualization of the AdEx neuron [7, 8]. Setting $dV/dt = dw/dt = 0$, we obtain the nullclines of the AdEx model:

$$w = -g_L(V - E_L) + g_L \Delta_T \exp\left(\frac{V - V_T}{\Delta_T}\right) + I \quad \text{V-nullcline} \quad (4)$$

$$w = a(V - E_L) \quad \text{w-nullcline} \quad (5)$$

We can immediately notice that the w-nullcline is just a straight line, the slope of which equals the subthreshold adaptation conductance a , and which is shifted offset on the w axis based on the leak reversal potential E_L . Similarly, we see that the shape of the V-nullcline for a given set of parameters will be constant and will only shift along the w-axis based on the applied current. This is insightful because there are other models where the entire nullcline changes shape with changes in applied current. We can now display the model behaviors from Figure 1 in the phase plane according to the same AdEx phase plane visualization convention as Naud et al. [7]:

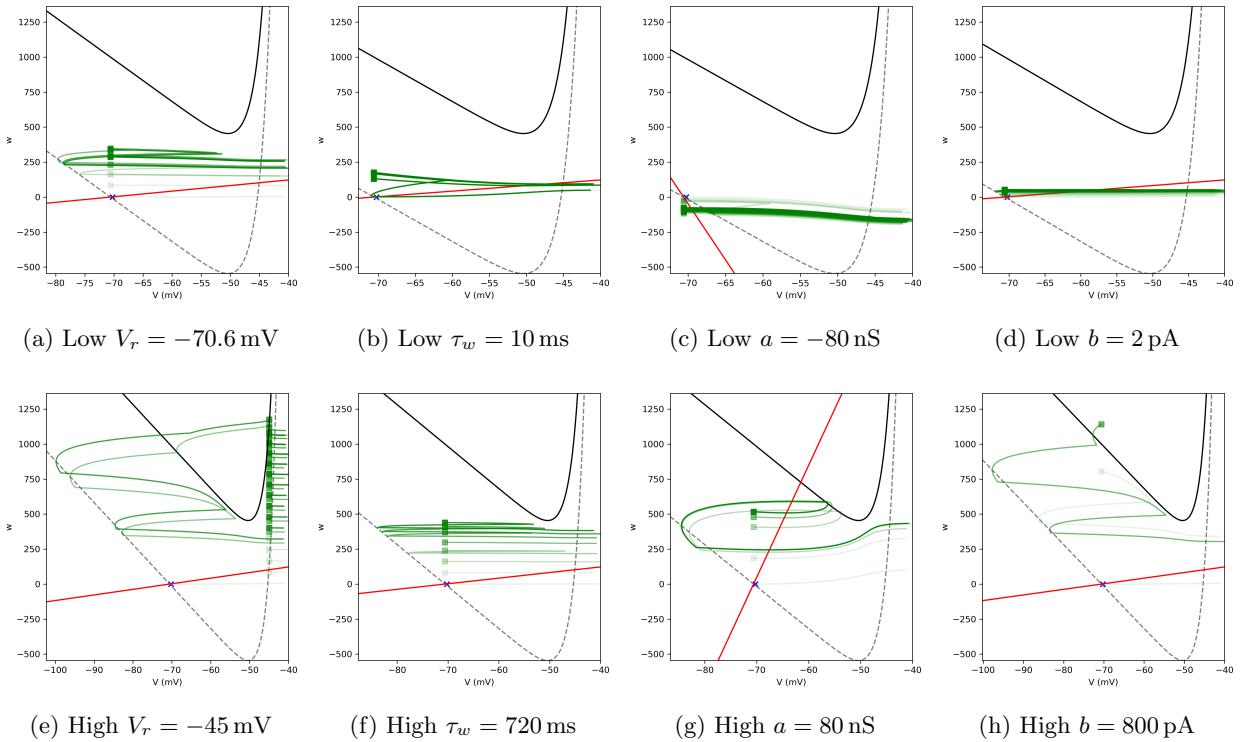


Figure 2: Phase plane representation of Figure 1. Black (solid) : V-nullcline during current pulse, Grey (dashed) : V-nullcline at rest, Red : w-nullcline. Blue "x" represents (resting equilibrium) starting point. Squares represent post-spike resets. Shade of squares and lines indicate time since beginning of experiment (light to dark). Only data from first half of the experiment is shown. Trajectories between each spike are shown until one datapoint before reaching V_{peak} which ≈ 40 mV.

The phase-plane analysis confirms the logical assumption that the simulation starting point $(V, w) = (E_L, 0)$ is an equilibrium of the system, because the resting V-nullcline and the w-nullcline intersect in all conditions exactly at the starting point $(V, w) = (E_L, 0)$ (Figure 2). Additionally, bursting solutions and other high frequency spiking solutions show a completely different phase-plane signature. Namely, bursting

solutions keep accumulating w , until the current is switched off, at which point the solution travels below the V nullcline and even after turning the current back on, the reset of the first spike in the next burst is "below" some reset points in the previous burst (Figure 2e). The other "high frequency" spiking regimes (Figure 2b, 2c, 2d) either alternate between the global equilibrium during the current OFF periods and a slightly elevated resets during current ON periods if the adaptation can recover fast enough (Figure 2b) or, between a state close to the global equilibrium but offset by the growing adaptation that persists due to the frequency of the current pulses (Figure 2c, 2d). I could keep going and focus the whole project on analysis of these firing patterns and bifurcations in the AdEx model, but as stated in the Introduction, I spent most of my time on a different part of the project. However, Naud et al. [7] and Touboul and Gerstner [8] provide exhaustive investigations of the topic with principled criteria to classify different firing patterns in AdEx neurons.

2.3 Basic AdEx behaviors from Brette and Gerstner 2005

Here I exactly reproduce some basic behaviors Brette and Gerstner reported in the initial AdEx paper [2] and add the corresponding phase-portraits which the authors omitted in the original publication.

2.3.1 Adaptation to step current

In this experiment, the default model parameters from Table 1 are used. First a subthreshold 0.5 nA current is applied for 200 milliseconds, to observe the sub-threshold response. It can be seen than square pulse results in a square response proportional to stimulus amplitude (Figure 3a) and after stimulus offset, returns back to the stable equilibrium baseline (3b). At 500 milliseconds, a suprathreshold 0.8 nA step-current is turned on and maintained. The advantage of this experimental approach and the reason why step-current is the most common way to characterize neuronal responses is that it allows one to study the new "steady state" to which a neuron might settle after a while at the stepped-up current (in contrast, this is more challenging with my previous approach where the current is applied periodically and the solutions do not have time to settle). We can see that in with the default parameters, the AdEx neuron displays adaptation to the step current. That is, initially high frequency of spiking, followed by an increase in inter-spike interval, until a the inter-spike interval settles at the frequency determined by the adaptation-time constant. This can be seen both in the voltage and w traces which settle to the same shape after the fourth spike (Figure 3a), as well as in the phase plane (Figure 3b), where we see that after the fourth spike, all subsequent spikes reset to the same dark-green spike-rest coordinate (V, w). These results exactly reproduce the AdEx behavior shown in Figure 2C of [2].

2.3.2 Bursting to step current

Keeping the same step current protocol, simply changing $V_r > V_t$, namely $V_r = -47 \text{ mV}$, while keeping the default $V_T = -50.4 \text{ mV}$, maintains the same subthreshold behavior, but induces the onset of a sustained bursting regime (Figure 3c). Here again, the initial burst is higher-frequency, emitting six spikes, but afterwards settles into spiking three times per burst at regular timing, always following the same trajectory, as can be seen in the phase plane (Figure 3d). One can appreciate the similarity between these trajectories, and the bursting we saw previously with the pulsed current approach in Figures 1b and 2e, as well as the fact that the observation from before still holds that with sustained bursting, the reset of the first spike in the next burst is "below" some reset from the previous burst. These results exactly reproduce the AdEx behavior shown in Figure 3C of [2].

2.3.3 Post-inhibitory rebound

Finally, we study the response of the AdEx neuron to switching ON and OFF a 400 millisecond hyperpolarizing current pulse of -0.8 nA . After changing the following parameters $E_L = -60, V_r = -60, a = 80$ and $\tau_w = 720$, we notice that during the inhibitory pulse, the voltage initially goes down, but then, because of

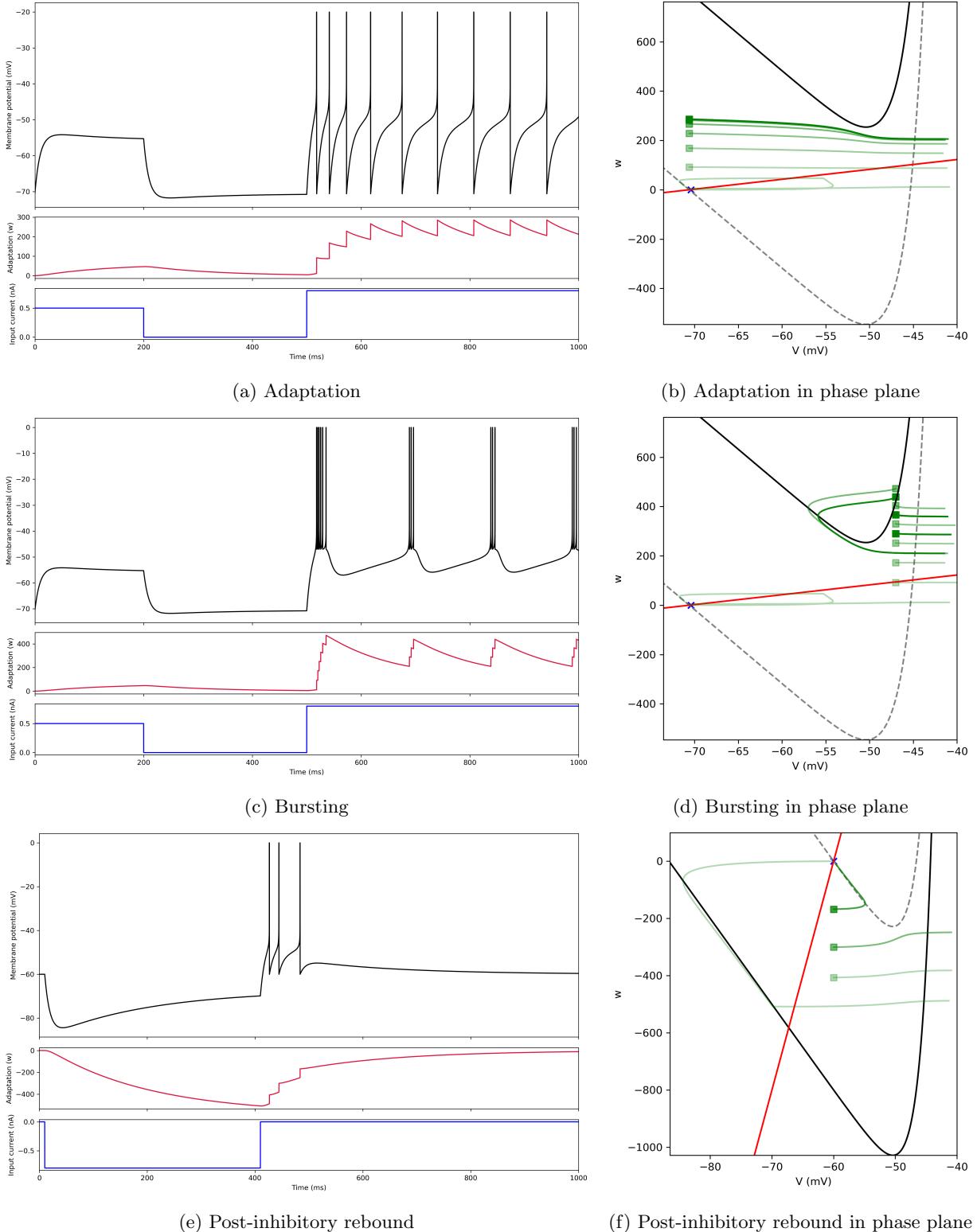


Figure 3: Replication of basic firing patterns reported in the original AdEx paper [2]. Conventions for time-series sub-figures (Left) same as in Figure 1, and for phase-plane sub-figures (Right) same as in Figure 2.

subthreshold adaptation and negative w , the voltage increases starts approaching -70 mV (which is close to a fixed-point of the system when $I = -0.8$ nA and the V nullcline is shifted down, as can be seen in Figure 3f). Because there is no excitation at all, w is at an all-time low when the hyperpolarizing current is turned off and thus, together with the voltage change stimulates the neuron to fire as the applied current returns to 0 (Figure 3e). However, after three spikes, this quickly stabilizes and w returns to 0 and V returns to $V = E_L$ mV, the steady state equilibrium at $I = 0$ (Figure 3f). This reproduces Figure 3D of [2].

3 Spiking model for in-vivo two-photon calcium imaging data

The initial analysis of the AdEx model, helped me understand how the different model parameters control the neuron's behavior, and verify that my implementation was correct. The rest of my work and the rest of this report focuses on my attempt to develop a method to fit these models to in-vivo two-photon calcium imaging data. The sketch of my idea is outlined in Figure 4, and I describe each step of the process below.

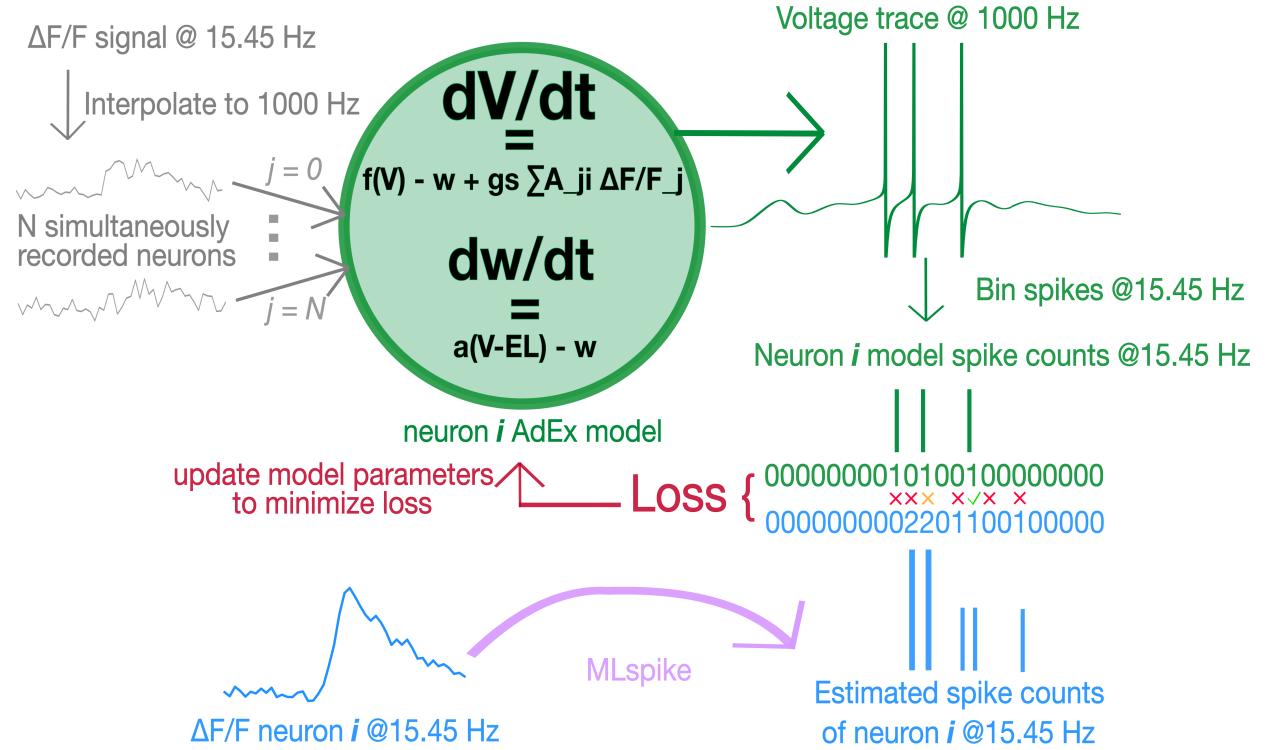


Figure 4: Schematic of proposed AdEx model-fitting framework for in-vivo two-photon calcium imaging.

3.1 Nature of two-photon calcium imaging data and modeling trade-offs

To understand the decisions I made in the model fitting process, it is important to have at least a basic idea about the nature of two-photon calcium imaging data (further referred to as calcium imaging). Calcium imaging takes advantage of genetic engineering tools which allow one to express calcium-sensing fluorescent proteins in mouse neurons. These genetically-encoded-calcium-indicators (GECIs) emit green light when they bind calcium, and this light fluorescence can be detected in-vivo in behaving awake animals using a two-photon microscope. Because intracellular calcium increase is a proxy for neuronal activity, this approach is often used in neuroscience to simultaneously record the location and activity of hundreds to thousands

of individual neurons in-vivo. Therefore, the type of signal one obtains from a calcium imaging experiment is a "live" video of a small part of the brain captured in the imaging frame of the microscope across time, and localized changes in brightness in this video can be traced to activity fluctuations in individual neurons across time. The final processed version of this signal is called $\Delta F/F_0$ and is a measure of the particular neuron's instantaneous change in calcium-related fluorescence (ΔF) relative to baseline fluorescence of that neuron (F_0). Despite being a revolutionary and incredibly useful technique, in-vivo calcium imaging presents a number of challenges for model-fitting.

The first problem is that calcium concentration is an indirect readout of neuronal activity and develops on a much slower time-scale than voltage. To make matters worse, two-photon microscope samples the full field of view at only 15.45 Hz. This means that we only have an indirect readout of the neuron's activity every 65 milliseconds, while voltage fluctuations and spikes evolve at sub-millisecond level (both in reality and computational models). This is one major issue I had to work around.

The second problem relates to parameter fitting approaches. As I alluded to before, Brette and Gerstner [2] outline a procedure for systematically estimating AdEx parameters, and similar procedures are described in subsequent work by Clopath [3], who fit AdEx models to in-vivo data. However, a problem with these approaches is that they assume one has direct access to the membrane voltage of the real neuron to be fitted (through patch-clamp etc.) at the same time-resolution as the fitted model. A more fundamental issue is that these parameter estimation strategies rely on being able to physically stimulate individual in-vivo or in-vitro neurons with the exact same currents that are then applied to the in-vivo models. These are extremely constraining and artificial circumstances that are mostly impossible in practice. At least to my knowledge, the authors and their successors have yet to outline a systematic method to fit AdEx models to real in-vivo data obtained during sensory stimulus presentation or behavioral tasks, in absence of highly controlled applied currents. For this reason, I had to resort to a black-box optimization approach, which not only made parameter fitting less systematic but also revealed other issues which I discuss later.

Another hurdle was the duration of recordings typically being fitted. Across various publications about AdEx models, I noticed these models were mainly being fitted to few seconds of data. On the contrary, in-vivo mouse experiments take tens of minutes, sometimes up to an hour. Therefore, trying to fit an AdEx model sampled at 10000 Hz resolution (as was usually done in most papers, $dt = 0.1$ ms) to such an in-vivo recording quickly becomes prohibitively computationally expensive. I addressed this issue in multiple ways.

- I simulated the model at 1000 Hz, which is much coarser time-resolution than most researchers use, but especially for my evaluation metric which involves matching spike counts in 66 millisecond bins, this was not too much of an issue. What is more, all the results obtained at 10000 Hz that I presented in Section 2 could also be reproduced at 1000 Hz ($dt = 1$ ms) (not shown), which means the model can still produce all the different firing patterns and behaviors that make it interesting, at a much lower computational cost when using $dt = 1$ ms.
- I used the simple forward euler integration method and implemented it in C, which resulted in orders of magnitude performance gains over more sophisticated integration methods with adaptive step sizes etc. implemented in `scipy`, especially for long recordings. Again, this did not change anything about the results and in fact all the results presented in this report were obtained using this fast and simple integration method.
- Finally, to accelerate prototyping of this approach even further, I mostly used only the first five minutes of my recordings, although my pipeline scales well for longer durations.

3.2 Adding Synaptic inputs

The main idea for my model, was to try modeling the in-vivo activity of individual neurons as a function of responses of other neurons recorded in the same local area at the same time. My hope was that if this was possible, it could reveal insights into local neuronal connectivity, which are otherwise difficult to infer from extracellular in-vivo data. I also used this approach to avoid the problem of modeling the moving auditory and visual stimuli that were being shown to the mice during the experiment, as translating those stimuli to

meaningful input / synaptic currents could be a whole project in itself. Nevertheless, the absence of a general stimulus-dependent or even just noisy input current apart from the synaptic currents from other neurons is certainly a weakness of my model. Drawing inspiration from Brette and Gerstner's synaptic formulation of the AdEx model [2], and mainly from Borges et al. [1] who constructed a randomly-connected AdEx network, I formulated my synaptic AdEx model as:

$$i, j \in (0, \dots, N) \quad \text{and} \quad N = \text{number of recorded neurons} \quad (6)$$

$$C_i \frac{dV_i}{dt} = -g_{L,i}(V_i - E_{L,i}) + g_{L,i}\Delta T_i \exp\left(\frac{V_i - V_{T,i}}{\Delta T_i}\right) - w_i + g_s \sum_{j=0}^N A_{ji}F_j \quad \text{where } F_j = \Delta F_j/F_{0,j} \quad (7)$$

$$\tau_{w,i} \frac{dw_i}{dt} = a_i(V_i - E_{L,i}) - w_i \quad (8)$$

$$\text{At spike time } t \text{ } (V_i(t) > 0) : V_i \rightarrow V_{r,i} \quad \text{and} \quad w_i \rightarrow w_i + b_i \quad (9)$$

where N is the number of neurons recorded in one calcium imaging session. I added g_s as the overall synaptic conductance (ranging between 0-1), to allow each neuron to tune its overall responsiveness to synaptic inputs. Admittedly, this parameter would be more meaningful if there was also another source of input current apart from the synapses from other neurons. Another consideration behind the g_s parameter is that once model parameters for an in-vivo neuron are extracted, in-principle any current in the appropriate amplitude range can be plugged in instead of the weighed sum of synaptic inputs and it will be rescaled by g_s to match the input strength the neuron responds to.

The main addition to the equation, and the only driving current is the weighed sum of synaptic inputs. A is the adjacency matrix, containing synaptic weights of each neuron to each other neuron. A is not necessarily symmetric because neuron i might receive a strong synapse from neuron j , but neuron j might only receive a weak synapse from neuron i . The values of A range from $(-1, 1)$, which was an easy way to work both excitatory and inhibitory synapses into the model. Namely, synapses with negative weights are considered inhibitory and synapses with positive weights are considered excitatory.

F_j is the synaptic input from neuron j at time t . In the current version, I used the $\Delta F/F_0$ signal (discussed in previous section) of neuron j as the j^{th} synaptic input for neuron i . To be able to feed this signal to the AdEx model (which I ran at 1000 Hz as discussed), I had to upsample F of each neuron by a factor of 65, since F was collected at 65 times lower sampling resolution. I did this by linear interpolation, followed by convolution with a gaussian kernel ($SD = 20$ ms) to smooth out artifacts from the linear interpolation. I also shifted all the interpolated synaptic inputs by 10 ms relative to the target signal of the modeled neuron to encourage the model to learn from the surrounding activity from just the moment before. Additionally, such offset also partially helped account for the delayed nature of the $\Delta F/F_0$ signal relative to spikes. An important consideration when working with F is that it must be in the scaled to correct "units", otherwise the model might become either epileptic, or never spike. Since we can substitute any synaptic signal for F , it is up to the us to determine the mapping between "F units" and "Input current units". In the previous section, we saw that the model usually starts spiking once the input current crosses ≈ 700 pA. Therefore, I tuned the F scaling factor in such a way that when run with the default parameters, the model emits spikes when it is receiving strong inputs but also is not just constantly spiking. For $\Delta F/F_0$ synaptic signals, 10 seemed to be a good scaling factor, but it somewhat differed neuron-by-neuron, and it would be better to find a systematic way to scale this signal.

Figure 5 shows example of a neuron simulated with 10 minutes of "synaptic input data" from the other neurons in the recording. The neuron is simulated with the default parameters from Table 1, $g_s = 0.99$ and $A_{:,i}$ is initialized to the correlations between the $\Delta F/F_0$ signal of the example neuron i and all other neurons in the recording.

A choice I made early on in the project, was to keep the weights of all the auto-synapses at 0. This was because I wanted the model to learn local connectivity patterns and find combinations of inputs that could reconstruct its signal. However, it is possible that this was both an unrealistic choice (auto-synapses do exist) and made the parameter fitting problem too difficult in cases where the neuron's activity was not clearly predictable only from the recorded local population.

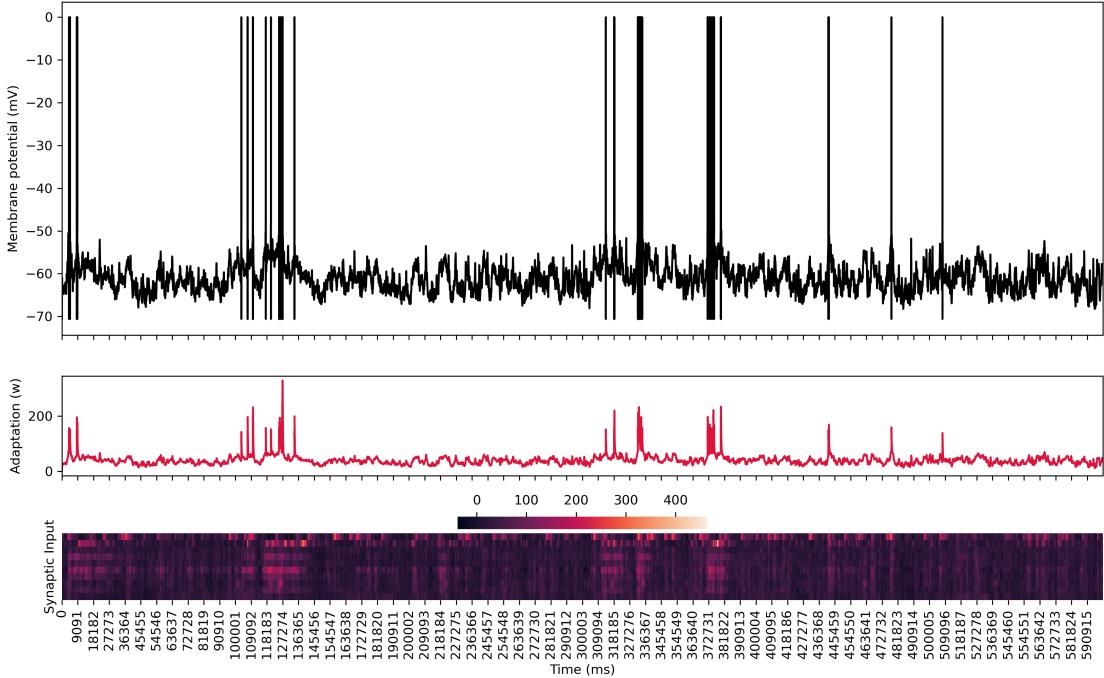


Figure 5: Example simulation of a default parameter synaptic AdEx neuron with 10 minutes of synaptic input data. Heatmap on the bottom shows input from 10 strongest synapses.

In principle, Equation 7 is a network formulation and ultimately, it would be very exciting to work on fitting the entire network to the recorded population of neurons. However, given the time constraints, I chose to focus on fitting each neuron as a single neuron i , completely independently of other neurons' model fits.

3.3 Spike count estimation

An obvious challenge I had to address was how to translate the calcium fluorescence signal into a spike train that I could compare with the spike train produced by the AdEx model. This is a widely studied problem in neuroscience and I was familiar with the pros and cons of the state-of-the art algorithms applied to this problem. Still, I tried multiple algorithms, but ultimately settled on the popular open-source MLspike algorithm [4], which I had already run on all my data in the past. It is important to stress that even with this advanced algorithm, it is impossible to get around the fundamental issue of low-temporal resolution in calcium imaging. As such, MLspike output is sampled at exactly the same rate as the original calcium signal. The difference is, that MLspike returns the most likely "spike train" that underlies the observed calcium response. Because the sampling resolution is so coarse, however, the entries of the MLspike "spike train" are not just binary, but instead are estimated counts of how many spikes occurred in a given ≈ 65 ms time-bin. The counts are restricted to the range $(0,1,2,3)$ because most in-vivo physiological neurons do not spike more than three times per calcium imaging frame and if there are bursts of more spikes, the calcium indicator saturates and it is hard to reliably determine the exact number. Thus, the "pseudo-count" 3 actually indicates $3 \leq$ spikes underlie the signal observed in the imaging frame. To keep the data directly comparable, I also clipped the counts of the spikes produced by the model between $(0,3)$.

3.4 Parameter optimization

As discussed, because of the problem constraints imposed by the nature of calcium imaging, as well as, collecting the data in-vivo in awake behaving mice, traditional AdEx parameter estimation techniques were not an option and instead I resorted to black-box optimization using a Differential Evolution algorithm implemented in the `nevergrad` Python package. An evolutionary algorithm was chosen because previous AdEx model fitting studies have used evolutionary algorithms in-conjunction with more systematic approaches [3, 5], and because of its ability to navigate large search spaces in absence of gradient information. The evolutionary algorithm was initialized with a population size of 100 and ran for 20 generations, with initial solutions generated using Latin Hypercube Sampling. Each candidate parameter solution was integrated over the full specified duration (for testing I mostly used five minutes - 300000 samples at 1000 Hz), and the resulting spike train from the model was binned into 65 millisecond bins, and evaluated against MLspike spike counts at the same sampling resolution using the custom evaluation metric discussed below. The algorithm tried to generate candidate solutions that would minimize the evaluation metric and in the end, returned the candidate solution with the lowest evalutation score as the putative estimated model parameters. The search was carried out over the following parameter space:

Parameter	Parameter Range
C	(30, 300)
g_L	(1.5, 31)
E_L	(-71, -57)
V_T	(-60, -41.9)
Δ_T	(0.6, 6)
τ_w	(15, 500)
a	(-30, 80)
V_r	(-80, -45)
b	(0.00001, 400)
g_s	(0, 1)
A_{ji}	(-1, 1)

Table 2: Parameter ranges for black-box parameter optimization using Differential Evolution.

The parameter ranges were hand-picked to cover all the possible firing patterns discussed in Section 2, as well as others comprehensively studied in [7]. To aid the model in traversing this vast search space and to help guide it towards firing patterns of interest, every two generations, a set of parameter configurations, known to elicit spiking and lead to various firing patterns (taken from [7]) was introduced. Additionally, the synaptic weights matrix A for each candidate solution was "hot started" with the correlations between the $\Delta F/F_0$ signals of all the individual neurons recorded in a given session.

3.4.1 Evaluation metric (Loss function)

To evaluate the fit of the my Synaptic AdEx model to the data (MLspike output), I was again forced to look for a different metric than had traditionally been used for most AdEx fitting studies. The traditional metric used in the field is the coincidence factor Γ [6]. This metric penalizes mismatches in both the total number, and timing of spikes (by counting spike coincidence within a given temporal precision cutoff, usually 2 ms) between the model spike train and the ground-truth spike train. The issue was, that because I did not have access to a ground truth spike train at millisecond resolution; and because I had to compare spike "counts" in 65 ms bins ranging from (0-3), not binary spike events, I could not use this metric in its standard form. In hindsight I realize I probably could have modified it and express the precision cutoff in terms of number of the coarser 15.47 Hz time bins, while keeping the total number of spikes as the sum over all the bins. This could be one way to improve performance of the model.

Instead, I chose another method meant for randomly-distributed count data, namely the Negative Poisson-log-likelihood. Treating binned spike-counts as a Poisson-distributed random variable is a common approach

in neuroscience modeling, so I considered this to be an appropriate place to start. If at each 65 ms time-bin, we consider k to be the observed Poisson-distributed spike count and λ to be the model-estimated spike count, we obtain the likelihood of parameter λ given that we observed k as $\mathcal{L}(\lambda|k) = \frac{\lambda^k e^{-\lambda}}{k!}$. To simplify the likelihood estimation, we take the log likelihood as:

$$\begin{aligned} LL(\lambda|k) &= \log(\mathcal{L}(\lambda|k)) = \log\left(\frac{\lambda^k e^{-\lambda}}{k!}\right) \quad \text{using } \log(a/b) = \log(a) - \log(b) \quad \text{and } \log(ab) = \log(a) + \log(b) \\ &= \log(\lambda^k) + \log(e^{-\lambda}) - \log(k!) \quad \text{using } \log(a^b) = b \log(a) \\ &= k \log(\lambda) - \lambda \log(e) - \log(k!) \quad \text{because } \ln(e) = 1 \\ LL(\lambda|k) &= k \log(\lambda) - \lambda - \log(k!) \end{aligned}$$

Because at each of the T time-bins i , we have a different k_i and λ_i , the total likelihood is the product of the individual likelihoods $\mathcal{L}_{total}(\lambda|k) = \prod_{i=0}^T \frac{\lambda_i^{k_i} e^{-\lambda_i}}{k_i!}$. Because we are using the log-likelihood, the product turns to a sum and we obtain the formula for the Total Log-likelihood between the observed MLspike counts k and the model output spike counts λ . Because we are working with a minimization algorithm, it is useful to minimize the negative log-likelihood (NLL):

$$LL_{total}(\lambda|k) = \sum_{i=0}^T k_i \log(\lambda_i) - \lambda_i - \log(k_i!) \quad (10)$$

$$NLL_{total}(\lambda|k) = -LL_{total}(\lambda|k) \quad (11)$$

Because most of the spike counts are 0 and $\ln(0)$ is undefined, the 0 counts were incremented with $\epsilon = 1e-7$.

Running the algorithm with NLL_{total} as the loss function revealed an important challenge with this modeling approach. In particular, there is very broad local minimum at $\lambda = 0$, when the model emits no spikes. This is the case, because many regions in the parameter search space lead to non-spiking models, and because most of the ground-truth counts in k are 0, it is extremely hard to find a way out of this local minimum. To deal with this, I introduced verified spiking solutions every 2 generations as discussed, but it was also necessary to change the loss function. It was imperative to provide strong incentive against non-spiking solutions, and also to reward models with the same firing rate as the ground-truth. Namely, I introduced a large no-spike penalty D , and penalized the squared error between the total ground truth spike counts K and total model spike counts Λ . I also normalized the likelihood relative to optimal that would be found, if exactly the ground truth spike train was returned by the model ($LL_{total}(\lambda = k|k)$), so that the optimal $NLL_{normalized}(\lambda = k|k) = 0$. This gives the final loss function used in this project:

$$NLL_{normalized}(\lambda|k) = -(LL_{total}(\lambda|k) - LL_{total}(\lambda = k|k)) \quad (12)$$

$$LOSS = \begin{cases} D & \text{if } \Lambda = 0 \\ NLL_{normalized}(\lambda|k) + \alpha(\Lambda - K)^2 & \text{otherwise} \end{cases} \quad (13)$$

In practice, setting $D = 1e8$ and setting $\alpha = 1/2$, managed to quickly steer the optimizer away from no-spiking solutions.

3.5 Model fitting results

There are many real scientific questions that I wanted to investigate using this model. Unfortunately, because I developing this model fitting paradigm involved overcoming many foreseen and unforeseen challenges at every step, I only managed to get the full optimization framework working two days before the report deadline, with significant improvements necessary to yield interpretable and scientifically useful results. Therefore, due to a lack of time to fit all the models, and because the results would be meaningless until the framework is optimized further, I did not proceed with fitting the model to an entire population of neurons and analyzing the results. Instead, I present here two model fits that illustrate where the challenges the method is still facing.

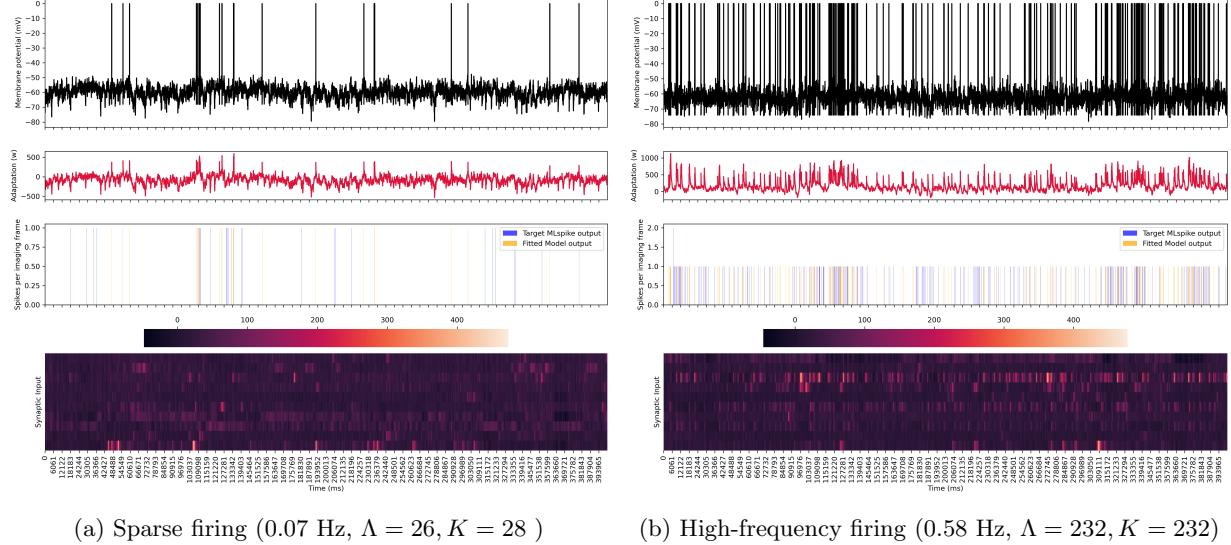


Figure 6: Example model fit for two neurons for a 400 second portion of a calcium imaging recording. Third row shows spike counts at 15.45 Hz resolution from MLspike (blue) and the fitted AdEx model (orange). Perfectly overlapping spikes are brown.

Figure 6 shows the results of two independent fits on two different neurons from the same recording. Neuron a is the same "neuron" as in Figure 5, only now it is not integrated with the default parameters but with parameters fitted according to the described procedure. The first noticeable thing is that the model is very good at fitting the firing rate of the real neuron for both sparsely(6a) and densely (6b) firing neurons. Specifically, for neuron a, the model predicted 26 spikes when the real neron fired 28 times. For neuron b, the model predicted exactly the same number of spikes as the neuron fired. This is thanks to the squared error penalty I introduced in Equation 13, which very effectively steers the model towards highly accurate firing rates. However, this is still relatively easy, because there are great many possible parameter configurations that would yield the same firing rate. The second observation that jumps out is that the timing of spikes is still highly inaccurate. While the model does generally spike in the same temporal "neighborhood" during bouts of high activity (eg. around 100 s for neuron a or around 120 and 340 s for neuron b), on the whole, the model does not perform well at matching the spike-times of the ground-truth spike train. This is highly problematic because accurate spike timing should emerge from the fact that the neuron is receiving temporally accurate synaptic inputs. Because of this issue, it is as of now hard to claim that the model is able to accurately fit to the in-vivo neuron purely based on the synaptic inputs from other neurons at the same recording site.

4 Discussion

Note that in principle, my framework is now able to extract all the AdEx parameters for a given neuron, and it is possible to then study the phase-plane behavior of this isolated AdEx neuron with any electrophysiological applied current protocol. Therefore the method and implementation is there to mathematically isolate and study in-vivo neurons recorded with calcium imaging. The only issue is, that because the optimizer still gets stuck in local minima, the output of the framework is not interpretable and reliable. Improving the loss function by for instance convolving the spike trains with a continuous function and minimizing the Mean Squared Error could be one way to force the model to care more about the "shape" and therefore timing of the model spike train. Another idea would be to replace the $\Delta F/F_0$ synaptic inputs with the MLspike spike trains from each neuron, and convolve those spike trains with a synaptic kernel. Through these approaches

(and perhaps other tweaks to the loss function and synapse mechanics), I believe the outlined approach could become a valid research framework.

References

- [1] BORGES, F., PROTACHEVICZ, P., LAMEU, E., BONETTI, R., IAROSZ, K., CALDAS, I., BAPTISTA, M., AND BATISTA, A. Synchronised firing patterns in a random network of adaptive exponential integrate-and-fire neuron model. *Neural Networks* 90 (June 2017), 1–7.
- [2] BRETTE, R., AND GERSTNER, W. Adaptive Exponential Integrate-and-Fire Model as an Effective Description of Neuronal Activity. *Journal of Neurophysiology* 94, 5 (Nov. 2005), 3637–3642.
- [3] CLOPATH, C., JOLIVET, R., RAUCH, A., LÜSCHER, H.-R., AND GERSTNER, W. Predicting neuronal activity with simple models of the threshold type: Adaptive Exponential Integrate-and-Fire model with two compartments. *Neurocomputing* 70, 10-12 (June 2007), 1668–1673.
- [4] DENEUX, T., KASZAS, A., SZALAY, G., KATONA, G., LAKNER, T., GRINVALD, A., RÓZSA, B., AND VANZETTA, I. Accurate spike estimation from noisy calcium signals for ultrafast three-dimensional imaging of large neuronal populations in vivo. *Nature Communications* 7, 1 (July 2016), 12190.
- [5] JOLIVET, R., KOBAYASHI, R., RAUCH, A., NAUD, R., SHINOMOTO, S., AND GERSTNER, W. A benchmark test for a quantitative assessment of simple neuron models. *Journal of Neuroscience Methods* 169, 2 (Apr. 2008), 417–424.
- [6] KISTLER, W. M., GERSTNER, W., AND HEMMEN, J. L. V. Reduction of the Hodgkin-Huxley Equations to a Single-Variable Threshold Model. *Neural Computation* 9, 5 (July 1997), 1015–1045.
- [7] NAUD, R., MARCILLE, N., CLOPATH, C., AND GERSTNER, W. Firing patterns in the adaptive exponential integrate-and-fire model. *Biological Cybernetics* 99, 4-5 (Nov. 2008), 335–347.
- [8] TOUBOUL, J., AND BRETTE, R. Dynamics and bifurcations of the adaptive exponential integrate-and-fire model. *Biological Cybernetics* 99, 4-5 (Nov. 2008), 319–334.