MACHINE LEARNING-ENABLED ROBUST SPIKE DETECTION OF VOLTAGE IMAGING DATA

VISHNU MANGIPUDI, ETHAN SONG, AND CHARLES ZHANG

Abstract. Spike detection from neuronal activity measurements, whether through calcium imaging or voltage imaging, is a critical task in neuroscience research, enabling precise analysis of neuronal responses to stimuli. However, challenges such as class imbalance due to sparse spikes, signal baseline drift, and high background noise significantly complicate this task. Building on recent machine learning-driven classification approaches, we develop a U-Net-based fully convolutional neural network to accurately perform pointwise spike classification. In particular, we implement key modifications aimed at enhancing our model's spike inference performance: focal loss to address class imbalance, four derived feature channels (Gaussian-smoothed, central differences, median-filtered, and high-pass filtered) to improve signal clarity, and confidence-based shadowing to suppress duplicate detections. The tested models achieve strong performance across multiple evaluation metrics, with our best model attaining an F1 score of 0.9297. Our results demonstrate the effectiveness of tailored preprocessing, postprocessing, and innovative architectural adaptations in tackling the unique challenges of voltage signal analysis.

1. Introduction

Neuronal signaling data presents valuable information for researchers aiming to discover and interpret physiological responses to specific stimuli. This information is encoded in spikes, discrete action potentials observed in voltage imaging data of neurons. Identifying the locations of these action potentials during specified time windows elucidates the response to a stimulus of interest, enabling further analysis and interpretation of the effects of its exposure. However, many raw data collections only provide indirect measures of neuronal activity. Moreover, accurately and efficiently recovering spikes from such traces is still an open problem due to low signal-to-noise ratio, signal baseline drift, and general variability.

Numerous methodologies for signal spike inference have been developed for calcium fluorescence images–a related but distinct neuronal activity proxy–providing useful algorithmic precedents for our approach. Examples in the literature include template matching [5, 8, 9, 12, 14, 19], linear deconvolution [10, 20, 28], and full Bayesian inference [22, 27]. A notable example is CaImAn [7], which combines constrained nonnegative matrix factorization (CNMF) with the OASIS deconvolution algorithm [6] to isolate spike events and denoise calcium fluorescence signals in real time. More recently, neural network-based methods have shown strong performance on this signal type, including physiologically-informed models such as MLSpike [4], variational autoencoders like DeepSpike [26], and end-to-end signal-to-signal (S2S) networks [25]. In 2025, SpikeAgent [17] leveraged the power of LLMs to autonomously sort spikes and generate reports and justifications. However, these calcium fluorescence methods rely on signal-specific assumptions that do not transfer to voltage imaging, motivating the development of dedicated tools for this newer modality [1, 2, 13, 15, 21, 24]. The most well-known adaptation, VolPy, extends the CaImAn framework to voltage imaging data by creating an end-to-end voltage processing pipeline, integrating preprocessing steps such as motion correction, denoising, and segmentation, with OASIS-based deconvolution and spike detection [3]. Hong et al. [11] instead use Importance Weighted Variational Autoencoders to improve spike inference accuracy.

Inspired by these neural network-driven approaches, we extend current work in the application of deep learning techniques for pointwise classification in voltage imaging recordings. In particular, we propose our Voltage U-Net Evaluator (VUE), a U-Net convolutional neural network-based approach to binary spike identification of preprocessed manually labeled data. VUE differs from prior approaches in three key respects: it adopts a U-Net architecture for pointwise binary classification of voltage traces rather than treating spike detection as a segmentation or source-separation problem; it incorporates focal loss to handle the severe

Key words and phrases. machine learning, artificial intelligence, neural networks, neuroscience, voltage imaging, spike sorting, spike inference, convolutional neural network.

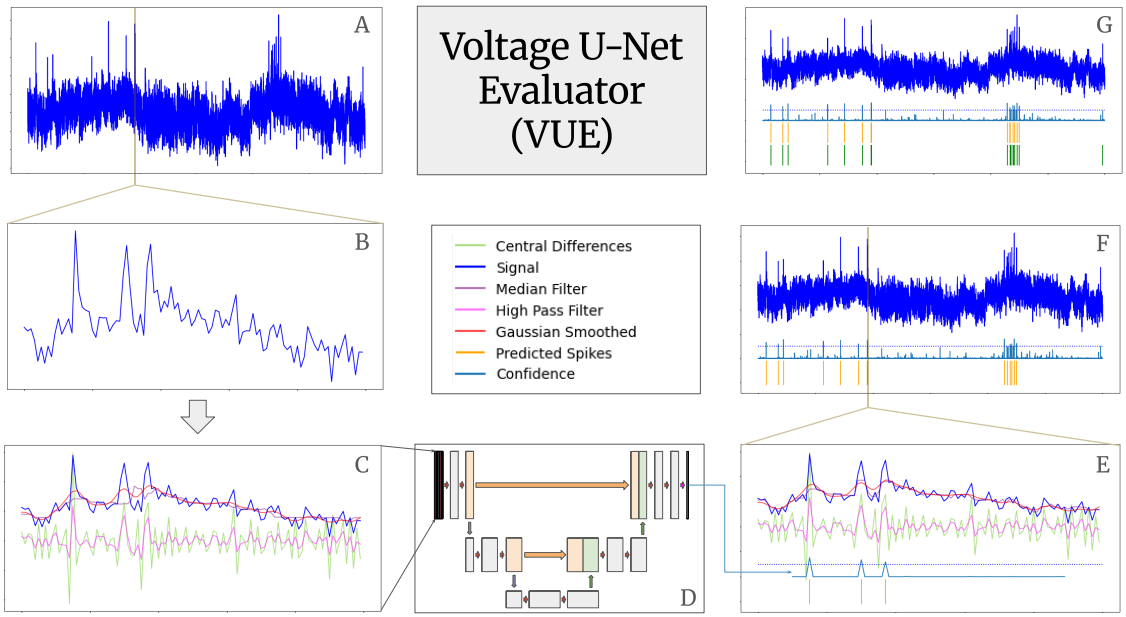

Figure 1. (A) The Voltage U-Net Evaluator (VUE) takes in as input a one-dimensional time-series neuronal voltage signal. (B) The signal is split into overlapping 100×1 windows. (C) The model input is augmented by calculating four modified feature channels using Gaussian smoothing, central differences, median filtering, and high-pass filtering. (D) The input channels are then fed into a tailored U-Net. (E) The U-Net generates spike confidences for the middle 80% of time points in the window, which represent predicted spikes if over the threshold of 0.5. (F) The windowed predictions are then recombined to generate the model's predictions over the entire file. (G) A comparison of the model predicted spikes (orange) and human-identified spikes (green) for the selected file.

class imbalance between spike and non-spike time points; and it augments the raw signal with four derived feature channels (Gaussian-smoothed, central differences, median filtered, and high-pass filtered) to give the model richer local context. Together these design choices allow VUE to achieve strong performance across both high- and low-SNR recordings. A high-level overview of VUE's pipeline is presented in Figure 1.

The remainder of this paper is organized as follows. Section 2 describes the model architecture, beginning with a progression from standard feed-forward networks to the U-Net design adopted for VUE, followed by the preprocessed feature channels and postprocessing steps used to enhance predictive performance. Section 3 details the dataset, normalization procedure, and windowing pipeline used to prepare inputs for the model. Lastly, Section 4 presents quantitative evaluation of the final model, an ablation study isolating the contribution of each pipeline component, and a discussion of model performance stratified by signal-to-noise ratio.

2. Methods

2.1. **Model Architecture.** A *feed-forward neural network* (FNN) is a function $\mathcal{N}$ on an *input vector* x with a set of parameters θ called *weights* and *biases*. Formally, consider L affine transformations of the form $T^\ell(x) = \mathbf{W}^\ell x + \mathbf{b}^\ell$ for $1 \leq \ell \leq L$ and some nonlinear *activation function* σ. A feed-forward neural network is defined as the function

$$\mathcal{N}(x;\theta) = T^L \circ (\sigma \circ T^{L-1}) \circ (\sigma \circ T^{L-2}) \circ \cdots \circ (\sigma \circ T^1)(x),$$

where $\theta = \{\mathbf{W}^\ell, \mathbf{b}^\ell\}_{\ell=1}^L$ are the network parameters. We call $\mathbf{W}^\ell$ and $\mathbf{b}^\ell$ the *weight matrix* and *bias vector* of the ℓ-th layer, respectively. We call a feed-forward neural network *fully connected* if each weight matrix $\mathbf{W}^\ell$ is structurally dense with no forced zero entries.

While FNNs are conventionally useful for a variety of problems, *convolutional neural networks* (CNNs) are particularly useful in feature detection and classification in data with a known Euclidean structure, including time-based neuronal voltage signal data. Conventional CNNs involve a series of layers consisting of convolutional and pooling steps applied to an initial input before flattening it and feeding it through a standard fully-connected neural network for final prediction via a softmax activation function. Instead of affine transformations, CNNs consist of L convolutional transformations of the form $C^\ell(x) = \mathbf{W}^\ell * x + \mathbf{b}^\ell$ for $1 \leq \ell \leq L$ where $*$ denotes the discrete convolution operator, $\mathbf{W}^\ell$ is a learnable kernel tensor, and $\mathbf{b}^\ell$ is a learnable bias vector. Given a nonlinear activation function σ and a set of learnable parameters $\theta = \{\mathbf{W}^\ell, \mathbf{b}^\ell\}_{\ell=1}^L$ and a fully-connected neural network $\mathcal{N}$, the corresponding convolutional neural network $\mathcal{C}$ is the function defined as

$$\mathcal{C}(x;\theta) = \mathcal{N} \circ C^L \circ \sigma \circ C^{L-1} \circ \sigma \circ \cdots \circ \sigma \circ C^1(x).$$

Since spike detection requires specific one-to-one pointwise classification of data points in the voltage signal, we opt to use a fully convolutional network (FCN). While conventional CNNs output at significantly lower resolution than their input, FCNs have been shown to be particularly effective at preserving such resolution for semantic segmentation problems [18]. FCNs are modified CNNs that exclude the fully-connected neural network at the output stage; hence a fully convolutional network is a function $\mathcal{C}$ defined as

$$\mathcal{C}(x;\theta) = C^L \circ \sigma \circ C^{L-1} \circ \sigma \circ \cdots \circ \sigma \circ C^1(x).$$

An overview of an FCN architecture can be seen in Figure 2.

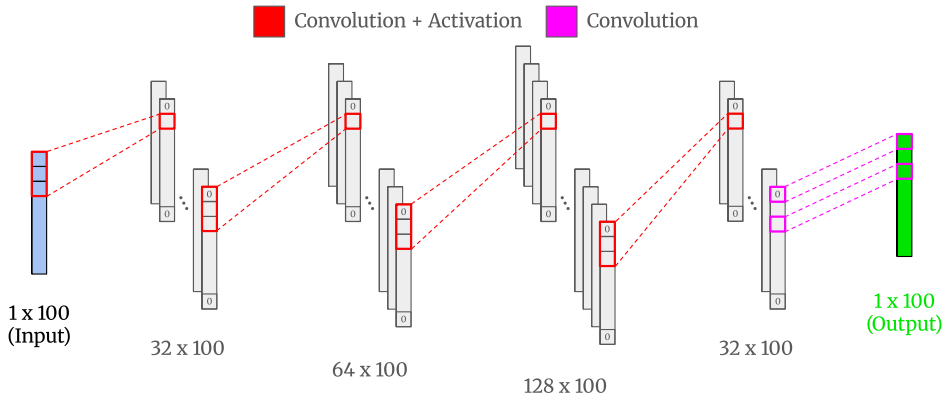

Figure 2. A diagram of an FCN model architecture. Note that each 100×1 segment of input data (left) is mapped to a final 100×1 segment of output predicted probabilities (right). The model runs with a batch size of 128 samples evaluated simultaneously.

To further improve our model accuracy, we use a U-Net, an extension to the traditional FCN architecture aimed at improving contextualization and localization of image segmentation models trained on small datasets [23]. Rather than directly following a set sequence of convolutions, U-Nets intermittently apply pooling to reduce dimensionality and improve feature extraction. However, as image segmentation requires the output to have the same dimensions as the input, U-Nets then apply a series of transposed convolutions to iteratively upsample the condensed input back to its original dimensionality. To better spatially contextualize information during this process, U-Nets concatenate upsampled tensors with earlier generated outputs of the same dimensions, mitigating lowered feature map precision brought about by pooling. Formally, U-Nets are defined similarly to FCNs, but introduce transposed convolutional transformations of the form $U^\ell(x) = \mathbf{W_U}^\ell *_F x + \mathbf{b_U}^\ell$ corresponding to the ℓ-indexed layer. Thus given L_D downsampling layers and L_U upsampling layers (including a final standard convolution to produce output of the correct dimension), along with a nonlinear activation function σ and sets of parameters $\theta_D = \{\mathbf{W_D}^\ell, \mathbf{b_D}^\ell\}_{\ell=1}^{L_D}$, $\theta_U = \{\mathbf{W_U}^\ell, \mathbf{b_U}^\ell\}_{\ell=1}^{L_U}$, we can formally express a U-Net (ignoring input augmentation through skip concatenation connections for clarity) as

$$\mathcal{C}_\mathcal{U}(x;\theta_D,\theta_U) = C^L \circ \sigma \circ U^{L_U-1} \circ \sigma \circ \cdots \circ \sigma \circ U^1 \circ \sigma \circ \mathcal{C}(x;\theta_D).$$

We adapt the traditional U-Net architecture to our problem to further enhance our model's predictive ability. As voltage signals are recorded as one-dimensional vectors, rather than the two-dimensional images that conventional U-Nets are typically designed for, we adjust all operations accordingly. Furthermore, since our individual data inputs are far smaller than the images typically trained on, we include zero padding for all convolutions besides the output layer to maintain input dimensionality (refer to Section 2.3). Finally, we integrate batch normalization in convolutional modules to speed up training and stabilize model predictions. To implement and train this model, we use the `Conv1d` function in the PyTorch library with `BatchNorm1d` selected for normalization, ReLU selected for the activation function for the hidden layers, and Dropout in each layer with $p = 0.3$. The Adam optimizer is used with an initial learning rate of 0.001 and a cosine annealing learning rate scheduler. Our architecture consists of two pooling steps and two transposed convolutions (implemented via `torch`'s `ConvTranspose1d` function), each spaced out by two traditional convolutional layers. The general design of our U-Net model is illustrated in Figure 3.

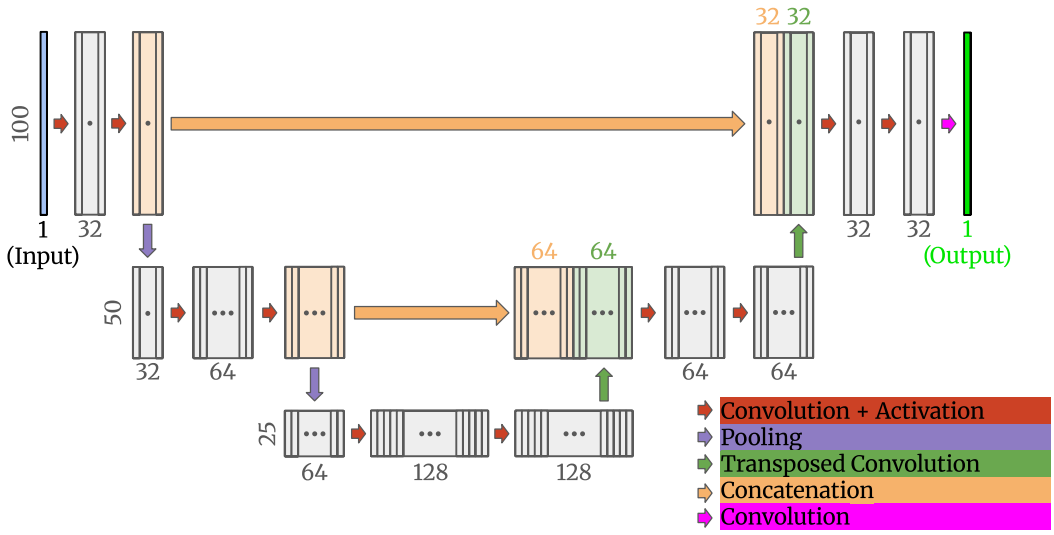

Figure 3. A representative diagram of our U-Net model architecture for spike inference. The height and width of each rectangle represent the length and number of channels of their respective tensors. Arrow colors correspond to specific PyTorch functions.

Since spikes are a rare occurrence in voltage signals, we improve our model by implementing focal loss to address object detection in problems with extreme class imbalances between foreground and background classes [16]. For a predictive model, we denote by p_t the model's predicted probability of the ground-truth class $y \in \{0, 1\}$ for each data point; in other words,

$$p_t = \begin{cases} p & \text{if } y = 1, \\ 1 - p & \text{if } y = 0. \end{cases}$$

Focal loss is then defined by the focal parameter $\gamma \geq 0$ and balancing parameter $\alpha \in [0, 1]$. Corresponding α_t are defined similarly to the p_t as

$$\alpha_t = \begin{cases} \alpha & \text{if } y = 1, \\ 1 - \alpha & \text{if } y = 0. \end{cases}$$

The standard binary cross entropy loss $\text{CE}(p_t) = -\log(p_t)$ is multiplied by α_t to scale loss with class size and a modulating factor of $(1-p_t)^\gamma$ to downweight the effect of well-classified classes, preventing the vast number of "easy" non-spike predictions from overwhelming the loss during training. Thus, focal loss is defined as

$$\text{FL}(p_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t).$$

The optimal values of α and γ are determined empirically in Section 4.1.

2.2. **Preprocessed Data Channels.** Signal drift and variable background noise change the shapes, positions, and heights of spikes over the course of recordings. To account for non-stationary data, we feed into the model several added feature channels in parallel with the normalized recording, outlined in Figure 4.

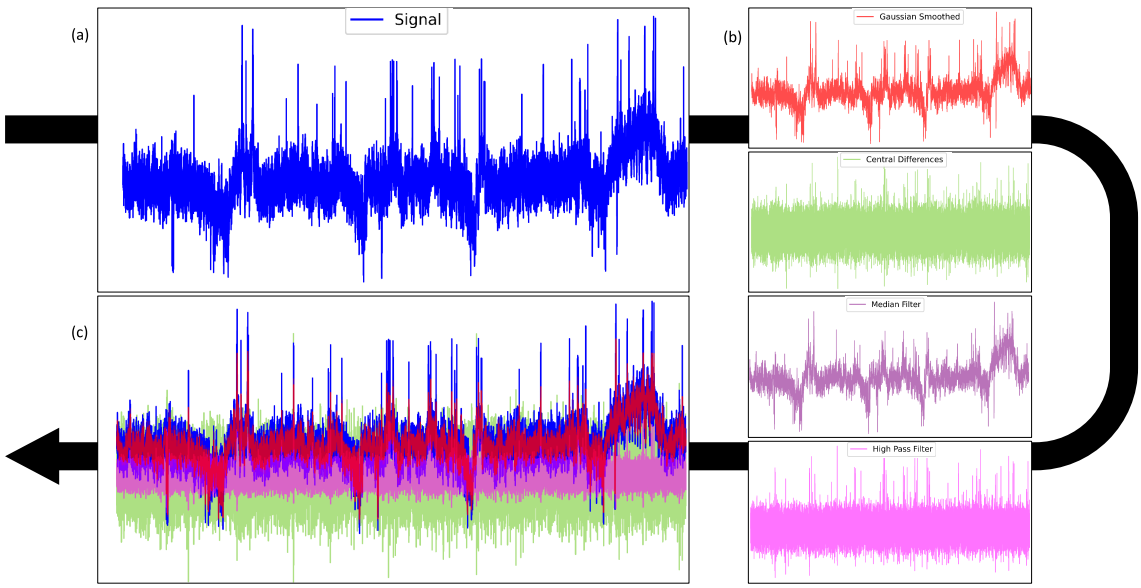

Figure 4. The preprocessed data channels fed into the model. (a) The initial voltage signal of a certain data file. (b) The computed Gaussian-smoothed curve and median filter to contextualize drift and background noise, along with the central differences curve and high-pass filter to contextualize large outliers and spikes in the data. (c) All five computed data channels eventually fed into the model overlaid.

We provide a Gaussian-smoothed curve of each voltage signal as an additional feature to the model to directly contextualize drift and background noise in the data, to be processed in parallel with the normalized signal. We do so by applying a Gaussian filter to smooth the voltage imaging data, feeding it into our U-Net during training and testing as an extra channel. For a given standard deviation $\sigma > 0$, we define the discrete one-dimensional Gaussian kernel as

$$g(u, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{u^2}{2\sigma^2}\right),$$

where u is an integer. Let $f : \mathbb{Z} \to \mathbb{R}$ be a discrete-time voltage signal. The discrete Gaussian filtering of f at time t is given by

$$h(t) = \sum_{u=-\infty}^{\infty} g(u, \sigma) f(t - u).$$

As h is mathematically a convolution product, it applies a filter by convolving f against the more regular Gaussian function g, resulting in a smoothing transform. Typically u ranges from $-\lceil 4\sigma \rceil$ to $\lceil 4\sigma \rceil$ due to it being impractical to implement a complete continuous convolution of f and $g(u, \sigma)$ on $(-\infty, \infty)$. In practice, we apply the `gaussian_filter1d` function of the `scipy.ndimage` package to compute this channel, which normalizes the truncated discrete kernel to have weights summing to 1.

Due to spikes being inherently local events, we also feed into the model an extra feature channel consisting of the negative second-order central differences of each point in the input window, defined as

$$\Delta_t = \begin{cases} x_0 - x_1, & \text{when } t = 0, \\ x_{n-1} - x_{n-2}, & \text{when } t = n - 1, \\ (x_t - x_{t-1}) + (x_t - x_{t+1}), & \text{when } 1 \leq t \leq n - 2, \end{cases}$$

where n is the length of the one-dimensional input signal. This computation is similar to observing the first and second derivatives to identify local extrema, aiding accurate spike inference. We feed these values

into our model as a third input channel, in addition to the normalized neuronal voltage signal data and its Gaussian-smoothed image to allow the model to more easily identify points with signal value significantly higher than their immediate neighbors.

To further account for short- and long-period signals encoded in neuronal voltages, we generate additional channels by applying median and high-pass filters to signal data. Median filters attempt to mitigate noise by replacing signal values with the median of a window centered at their position. For a signal $x = (x_1, \ldots, x_N)$ and an odd kernel width κ, the median-filtered value at index i is

$$M(x_i) = \text{med}(x_{i-\frac{\kappa-1}{2}}, \ldots, x_{i+\frac{\kappa-1}{2}}),$$

with zero padding $x_i = 0$ when $i < 1$ or $i > N$.

On the other hand, an n-th order Butterworth high-pass filter is characterized by the magnitude response function

$$|H(\omega)| = \frac{1}{\sqrt{1 + \left(\frac{\omega_c}{\omega}\right)^{2n}}}$$

for a given angular frequency ω and a specified cutoff frequency ω_c. Such a filter attenuates signals below the cutoff frequency while passing signals at or above it, and this suppression scales strongly with n. We apply a 2nd order ($n = 2$) filter with $\omega_c = 100$ Hz and a sampling rate of 1000 Hz (matching the 1 ms measurement intervals) to strongly select for local signal intensity variations while maintaining some level of general contextual information. This is implemented via `scipy`'s `signal.butter` function to generate the filter, and is then applied with zero-phase distortion using `signal.sosfiltfilt`.

By generating these additional feature channels and feeding them as input to our U-Net, our model architecture is now shown in Figure 5.

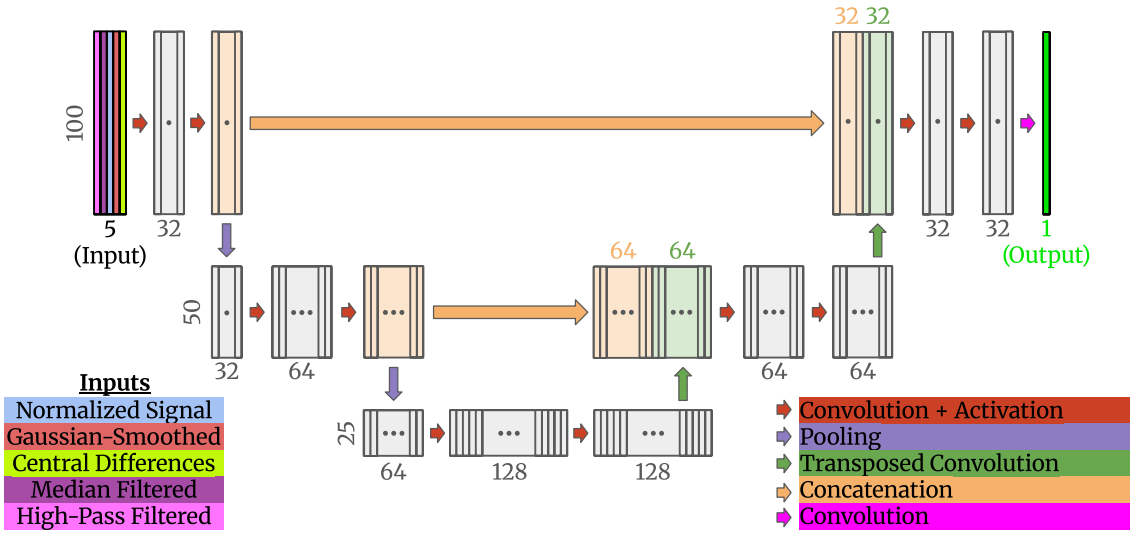

Figure 5. A representative diagram of our modified U-Net model architecture. The input (left) consists of five parallel data windows of 100 time-series data points each: the normalized signal data, its Gaussian-smoothed image, its central differences, and its median/high-pass filtered versions. Each 100×5 input corresponds to a single 100×1 output window of predicted probabilities for each data point in the input signal window.

2.3. **Postprocessing.** As all of the convolutional layers used in our model apart from the output layer use zero padding to maintain input dimensionality, outer tensor elements have less context on original signal information than ones nearer the center. These factors motivate us to ignore these less accurate predictions entirely, instead pruning the ends of our model's generated prediction windows before synthesizing them by averaging confidences to generate overall predictions (further described in Section 3.3).

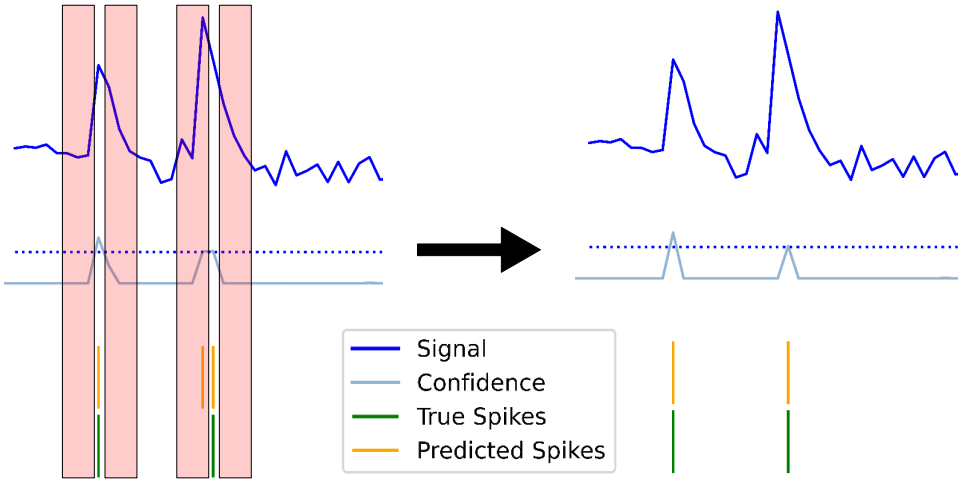

Figure 6. Confidence-based spike shadowing eliminates a duplicate detection. *Left:* The model predicts two spikes in rapid succession, only one of which is a true spike. The rightmost prediction has slightly higher confidence, shadowing the highlighted red intervals and suppressing the nearby erroneous detection. *Right:* After shadowing, the duplicate is removed and its confidence set to zero, leaving only the correct rightmost spike.

Another problem with our initial FCN was the prediction of duplicate spikes, inferring two point spikes in quick succession associated with a single neuron activation. To address this issue, we introduce biologically-informed confidence-based shadowing of detected peaks. After the model generates its predictions as a vector of sigmoid outputs, they are sorted in descending order. Then, for each identified spike, the confidence levels for the time points from 2 ms behind it to 2 ms ahead (excluding itself) are set to 0, such that no two spikes can be inferred within 2 ms of another. This threshold aligns with observed refractory periods between neuronal signals *in vivo*. To conserve resources, this shadowing step was performed after models were fully trained, and consistently demonstrated improvements across other modifications tested. A schematic for the shadowing methodology is shown in Figure 6.

3. Problem Setup and Dataset

3.1. **Raw Data.** Our data were provided as a MATLAB table containing the results of 461 voltage measurement trials. The channels relevant to our analysis are `IntensOrig`, containing the raw voltage signal values, and `BinSpikeT`, containing human-labeled spike locations that represent ground truth for the model. Each trial encapsulates measurements of neuronal signal voltage over 29990 ms with 1 ms intervals, which our model maps to a length-29990 vector of spike probabilities (confidence that a given millisecond contains a spike). Files containing the relevant data channels for each trial were then generated and randomly split into 320 training files, 80 validation files, and 61 testing files using a fixed seed to ensure consistency when model inputs were varied for optimization.

3.2. **Data Normalization.** To ensure consistency of input to our model, we normalize each raw data file before feeding it into the Voltage U-Net Evaluator. To preserve spike behaviors, we normalize the data such that 90% of the signal lies in the range $[0, 1]$ and 10% lies outside of it. An example of this data normalization method is shown in Figure 7.

3.3. **Data Windowing.** Directly mapping each data file to a respective array of predicted spike times is not ideal for a number of reasons. Voltage signal measurements are prone to extremely high noise and signal baseline drift. Moreover, spikes are local events, changing depending on each spike's immediate context, thus not requiring global data of entire data files. To address these issues, we train our model to infer spikes within 100 ms windows of the entire 29990 ms measurement period. In practice, the original files are each split into 599 smaller 100 ms windows with 50 ms overlap (with the last window having a 60 ms overlap with the prior one to include the final time points) during preprocessing, and are then treated as separate entries in the data points provided to the model. Several such windows are shown in Figure 8.

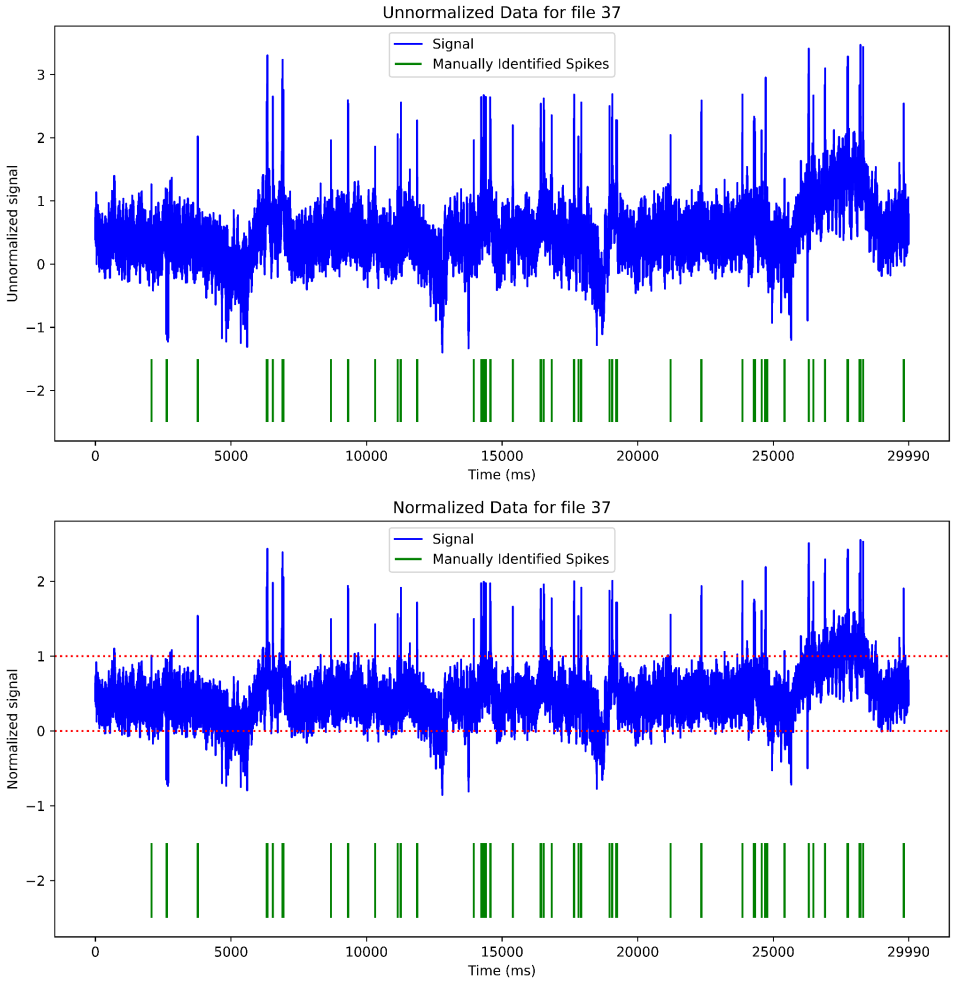

Figure 7. *Top*: The raw voltage signal data for data file 37. The raw signal data (`IntensOrig`) is shown in blue above the manually labeled spike locations (`BinSpikeT`) shown in green. *Bottom*: The normalized voltage signal data for file 37. 90% of the signal data now lies in the range $[0, 1]$ shown between the dotted red lines. Note that the manually labeled spike locations do not change.

Full spike predictions over entire files are generated in a similar way to how windows were initially created. The file is split into 100 ms windows with 50 ms overlap from left to right, then right to left, giving a total of 1198 windows where spikes are inferred. The confidences generated at each point are then averaged across all windows containing that point, which would entail four overlapping predictions in most cases with two or three at the edges. A point is classified as a spike if its averaged confidence exceeds 0.5, and as a non-spike otherwise. This stabilizes the model's predictions by providing it with more context at each point. Overall model metrics were evaluated on these synthesized predictions rather than individual windows to better reflect model performance during practical use. The model's primary goal is to produce predictions aligning with the manually provided labels in the provided dataset so that it can then be reliably and rapidly applied to unlabeled readings, while also being able to flag potential labeling omissions.

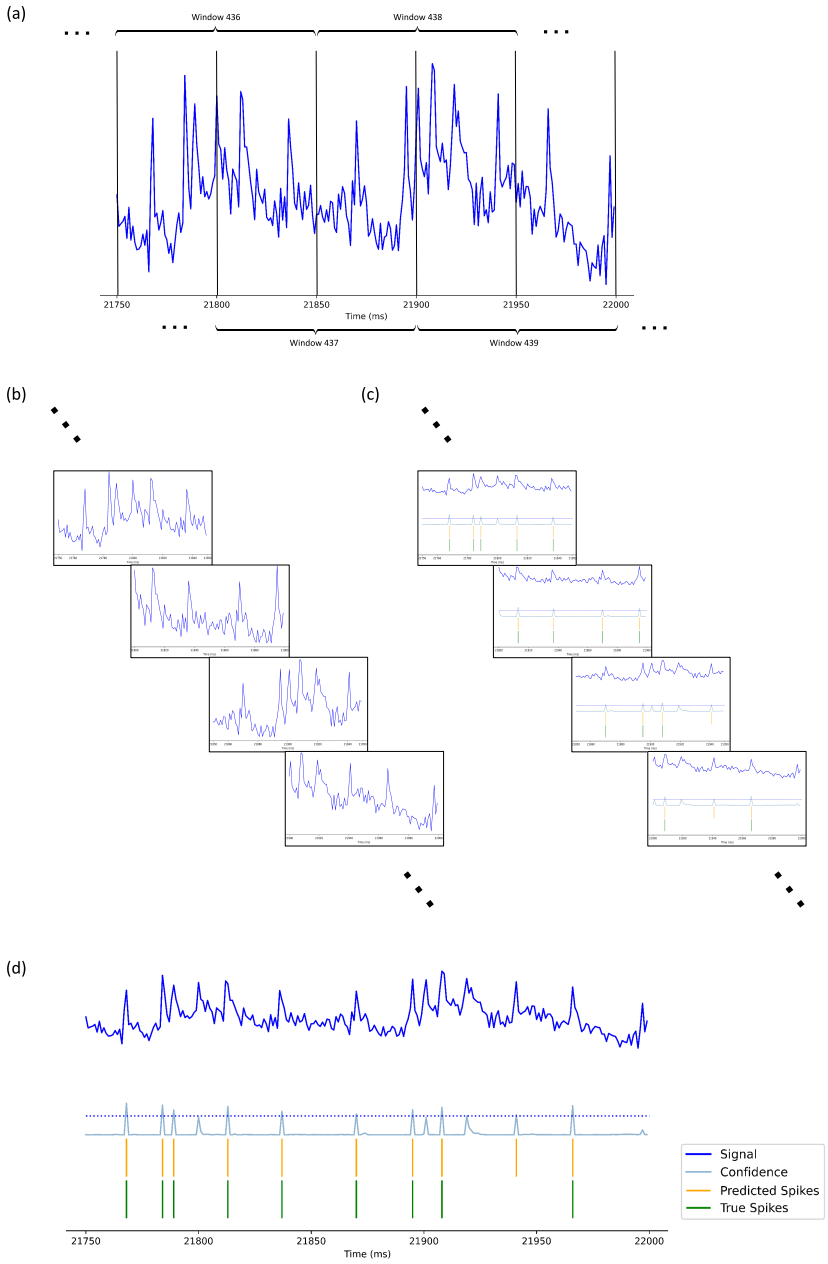

Figure 8. The windowing pipeline for the voltage signal analysis. (a) The windowing off of an initial voltage signal. The windows each have length 100 ms, and each is taken every 50 ms. (b) The windows of the signal stacked on top of one another, offset by 50 ms each. (c) The windows of the signal after passing through the model, generating confidence values and predicted spike locations in each of the windows. (d) The final predicted spike probability of the voltage signal at each data point is the average of that across all windows containing the data point.

4. Results

4.1. **Model Metrics.** We now report the performance of our final model on the held-out test set, along with a comparison to a minimal baseline. In characterizing performance, raw accuracy is not suitable for identifying the "best model" because the large class imbalance in our dataset limits the information it provides. Spike events are exceedingly rare compared to non-spikes: out of 29990 total time points, files

have an average of 101 spikes, corresponding to an occurrence rate of 0.34%. Explicitly, the distribution of total spike occurrences in files is shown in Figure 9.

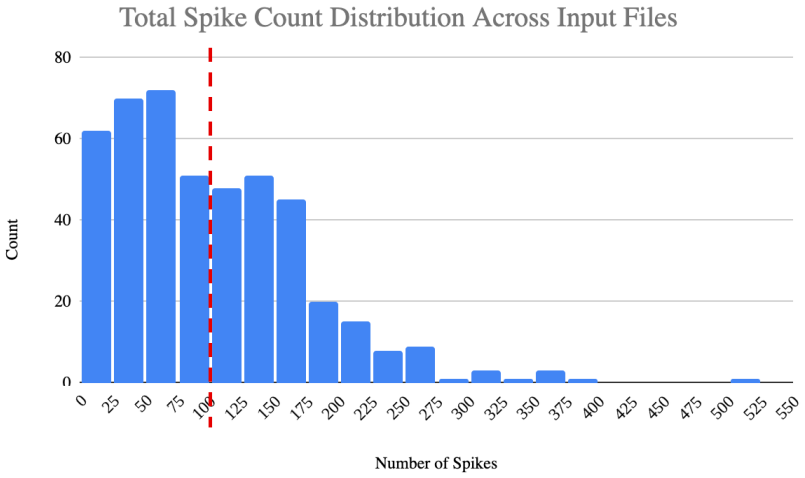

Figure 9. Histogram of total number of manually identified spikes in each data file, out of 29990 total recorded data points.

As such, a given model could attain near-perfect accuracy simply by never predicting spikes. Thus, to assess the performance of our approach on the provided dataset, we train a number of models on data from the training set, select the model with the highest F1 score–a metric that better accounts for predictive ability over all classes–on the validation set, and evaluate reported metrics on the disjoint test set.

As described in Section 3.3, model predictions on the entire file length are generated by averaging confidences on overlapping windows–the final metrics evaluated were on these whole-file predictions and not on individual windows to better reflect the model's practical performance. Thirty total training runs were conducted, varying both the weighting α (used to lower false negatives by increasing sensitivity towards potential spikes) and focal parameter γ (quantifying suppression of overrepresented classes, in this case non-spikes). Our best model, trained with $\alpha = 0.47$ and $\gamma = 2.85$, predicted 6068 true positives, 548 false positives, and 369 false negatives in the pruned test set of 1,826,950 time points. This yielded an accuracy of 0.9995, a precision of 0.9172, a recall of 0.9427, and an F1 score of 0.9297. The confusion matrix for its predictions is shown in Figure 10.

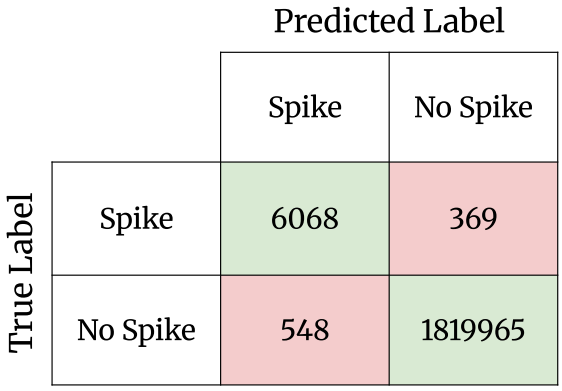

Figure 10. The confusion matrix of our final model.

The final model demonstrates strong overall predictive ability. Its precision of 0.9172 indicates the large majority of inferred spikes corresponded to manually labeled spikes $\left(\frac{6068}{6616}\right)$, while its recall of 0.9427 shows

that the model successfully identifies most of the assigned spike events $\left(\frac{6068}{6437}\right)$. Since the recall is higher than the precision, we find that the model has been tuned to be more sensitive to labeled spikes, desirable in this instance because it is far easier to manually eliminate false positives downstream than to re-identify any lost spikes from false negatives.

We also compare the performance of our final model to a minimal baseline FCN model that only uses windowed predictions. In particular, the baseline uses binary cross-entropy loss instead of focal loss and does not incorporate any augmented feature channels or postprocessing steps. This yields a precision of 0.8939, a recall of 0.8401, and an F1 score of 0.8661, all substantially lower than the final model.

4.2. **Ablation Study.** While comparison with the baseline gives some idea of the utility of the modifications as a whole, to more specifically determine the impact of each modification we introduced to the model pipeline, we generate an *ablation table* that enumerates model performance when exactly one method is suppressed. For instance, one of the augmented feature channels may be removed from the input, or binary cross-entropy loss may be used instead of focal loss (but not both at the same time). Each ablated model was retrained from scratch using the same train/validation/test split and training schedule as the full model, and the model with the highest F1 score out of 10 separate runs was chosen as the representative for comparison. We include the metrics of the minimal baseline and final models to produce Table 1:

Configuration	**Precision**	**Recall**	**F1**
Minimal Baseline	0.8939	0.8401	0.8661
w/o Focal Loss	0.9219	0.9334	0.9276
w/o U-Net	0.9046	0.8529	0.8780
w/o Gaussian Smooth.	*0.9241*	0.9301	0.9271
w/o Central Diff.	0.9215	0.9340	0.9277
w/o High-Pass Filter	**0.9261**	0.9245	0.9253
w/o Median Filter	0.9192	0.9349	0.9270
w/o End Pruning	0.9179	0.9395	0.9285
w/o Conf. Shadowing	0.9178	*0.9416*	*0.9295*
Full model	0.9172	**0.9427**	**0.9297**

Table 1. Ablation study: each variant removes one component from the full model. Bolded values are the largest in their respective column, while italicized are second largest. Note that the full model has the highest F1 score and recall of the ablated models, but not the highest precision; this is likely due to applying an ensemble of methods that each trade off an increased false positive rate for a decreased false negative rate. In particular, the high-pass filter contributes most prominently to this effect, as it has the highest precision but second-lowest recall (only above the vanilla FCN model) when ablated.

Several conclusions can be drawn from these data. The largest decrease in F1 score occurred when the U-Net architecture was replaced with the vanilla FCN, indicating that the architectural change was the most important contributor to enhanced model performance. Many of the other modifications served to enhance the model's recall while keeping precision as high as possible. The final model has the highest recall of all models tested in the ablation study, but its precision is only higher than those of the baseline and FCN models. As justified earlier, we prefer high recall to high precision for the model's practical implementation, and the channels that most contribute to this are the high-pass and Gaussian-smoothed channels (though all improve it marginally). The postprocessing steps had comparatively modest yet still measurable effects. Ablating either end pruning or confidence-based shadowing only slightly decreases F1 score, confirming that these modifications refine predicted spike trains by eliminating less reliable or faulty predictions. As a whole, the ablation study shows that the U-Net architecture is the dominant contributor to the performance gain, while the remaining preprocessing and postprocessing steps provide smaller but still positive refinements.

4.3. **Discussion.** Our results show that a machine learning-driven pipeline for neuronal voltage signal interpretation can yield strong results for spike inference on manually labeled datasets. The final model's F1 score of 0.9297 is quite high, indicating strong agreement with the manual labels. VUE is particularly applicable as a first-pass detector in a pipeline that includes downstream manual review. Recall is the most important metric to track for such a pipeline, and the final model's recall of 0.9427 means that fewer than 6% of true spikes are missed, which the Fan lab has indicated is sufficient in significantly speeding up their spike analysis procedure.

Moreover, the interpretation of these metrics is limited by the fidelity of human-identified spikes to actual neuronal signals. While manually labeled spikes almost certainly represent true neuronal activations, it is likely that a number of true spikes were missed across the tens of thousands of possible spikes in the dataset.

A proxy to determine the accuracy of the generated labels is the signal-to-noise ratio (SNR), a numerical value quantifying the strength of a particular spike measurement relative to background noise. SNR values were computed as the quotient of averaged spike amplitude and characteristic noise determined from a large gap between spikes. The distribution of these values is shown in Figure 11.

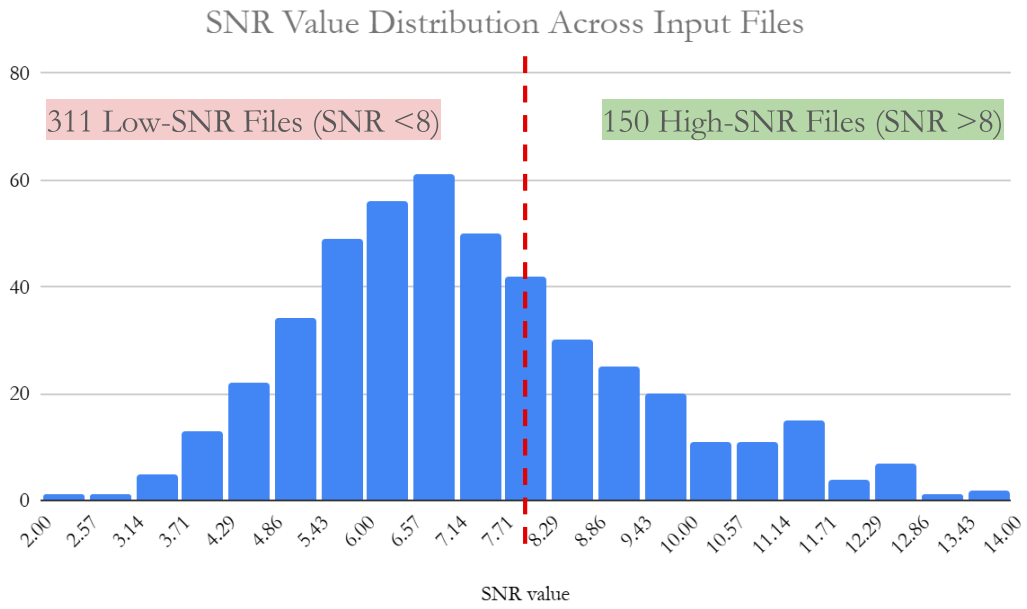

Figure 11. A histogram showcasing the distribution of computed SNR values across all files. Out of 461 total files, 150 (32.5%) were determined to have a high SNR and therefore reliable manual labels, while the remaining 311 (67.5%) were classified as low SNR and thus less reliable.

The empirically determined threshold of 8 delineating low-SNR and high-SNR files was provided by Dr. Fan's lab after careful inspection. As a concrete example, file 434–with a computed SNR of 4.35 (in the bottom 5% of files in the dataset)–displays significant errors in manual labeling. In particular, in the rightmost section of the file, a series of unlabeled local maxima were identified upon manual review as likely true spikes. This is explicitly pictured in Figure 12.

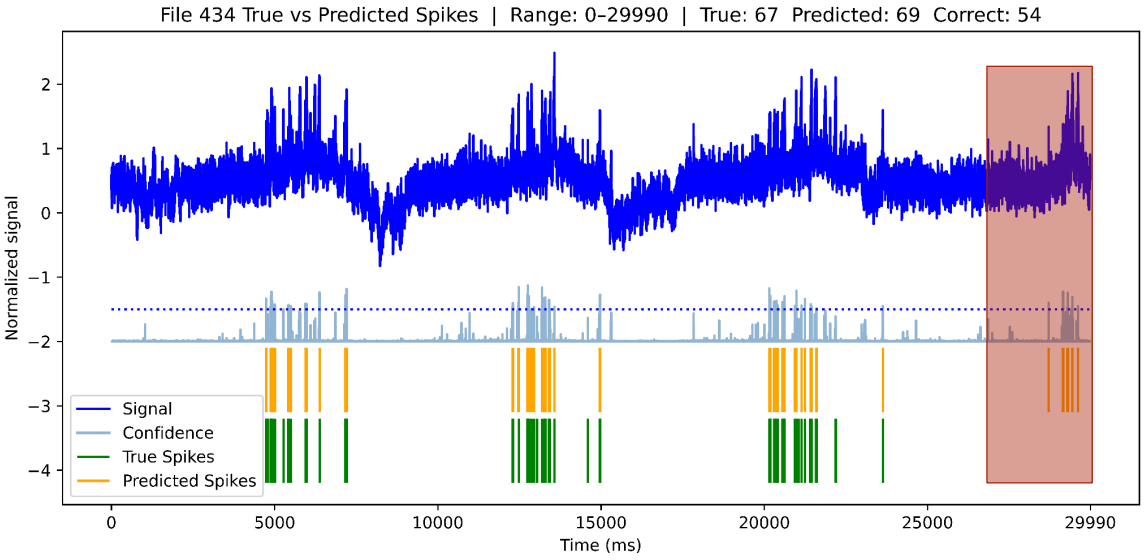

Figure 12. The predicted and actual spike locations for file 434. The red rectangle enclosing the last 3000 ms of the file represents a region where the VUE identified spikes that initial threshold-assisted manual classification apparently failed to find. Upon review, this region was noted to likely contain identifiable spikes, showcasing an error in training labeling that could have hindered observed predictive accuracy.

The effect illustrated by file 434 is a systematic one. Evaluating the final model on a stratified random sample of the test set by SNR class, we obtain an F1 score of 0.9688 on high-SNR files compared to 0.9130 on low-SNR files. Because high-SNR recordings correlate with more reliable manual labels, this gap indicates that much of the model's apparent error on low-SNR files may arise from imperfect ground-truth labels rather than a misclassification by the model.

Direct numerical comparison of VUE to prior voltage imaging tools such as VolPy is complicated by differences in evaluation datasets and reported metrics; VolPy reports performance primarily via correlation coefficients rather than F1 scores on binary spike labels. Nevertheless, the strong performance of VUE on both high- and low-SNR recordings, and its ongoing integration into the Fan lab's processing pipeline, suggest it presents a viable, practical tool for voltage imaging spike detection as a first-pass detector. In practice, VUE identifies a strong majority of candidate spike locations for later confirmation, substantially reducing the manual inspection burden for larger datasets.

Acknowledgements

The authors gratefully acknowledge their mentor Dr. Lu Lu for his support and guidance throughout this project. The authors would also like to thank Dr. Mitchell Daneker for his feedback on this paper, Dr. Linlin Fan and her group for providing the datasets and background for this project, and the MIT PRIMES-USA program for making this research opportunity possible.

References

[1] Abdelfattah, A. S., Kawashima, T., Singh, A., Novak, O., Liu, H., Shuai, Y., Huang, Y.-C., Campagnola, L., Seeman, S. C., Yu, J., Zheng, J., Grimm, J. B., Patel, R., Friedrich, J., Mensh, B. D., Paninski, L., Macklin, J. J., Murphy, G. J., Podgorski, K., Lin, B.-J., Chen, T.-W., Turner, G. C., Liu, Z., Koyama, M., Svoboda, K., Ahrens, M. B., Lavis, L. D., and Schreiter, E. R. Bright and photostable chemigenetic indicators for extended in vivo voltage imaging. *Science 365*, 6454 (2019), 699–704.

[2] Adam, Y., Kim, J. J., Lou, S., Zhao, Y., Xie, M. E., Brinks, D., Wu, H., Mostajo-Radji, M. A., Kheifets, S., Parot, V., Chettih, S., Williams, K. J., Gmeiner, B., Farhi, S. L., Madisen, L., Buchanan, E. K., Kinsella, I., Zhou, D., Paninski, L., Harvey, C. D., Zeng, H., Arlotta, P., Campbell, R. E., and Cohen, A. E. Voltage imaging and optogenetics reveal behaviour-dependent changes in hippocampal dynamics. *Nature 569*, 7756 (2019), 413–417.

[3] Cai, C., Friedrich, J., Singh, A., Eybposh, M. H., Pnevmatikakis, E. A., Podgorski, K., and Giovannucci, A. Volpy: Automated and scalable analysis pipelines for voltage imaging datasets. *PLOS Computational Biology 17*, 4 (2021), e1008806. https://journals.plos.org/ploscompbiol/article/file?id=10.1371/journal.pcbi.1008806.

[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 (2016), 12190.

[5] Franke, F., Quian Quiroga, R., Hierlemann, A., and Obermayer, K. Bayes optimal template matching for spike sorting –combining fisher discriminant analysis with optimal filtering. *Journal of Computational Neuroscience 38*, 3 (2015), 439–459.

[6] Friedrich, J., Zhou, P., and Paninski, L. Fast online deconvolution of calcium imaging data. *PLOS Computational Biology 13*, 3 (03 2017), 1–26.

[7] Giovannucci, A., Friedrich, J., Gunn, P., Kalfon, J., Brown, B. L., Koay, S. A., Taxidis, J., Najafi, F., Gauthier, J. L., Zhou, P., Khakh, B. S., Tank, D. W., Chklovskii, D. B., and Pnevmatikakis, E. A. CaImAn an open source tool for scalable calcium imaging data analysis. *Elife 8* (Jan. 2019).

[8] Greenberg, D. S., Houweling, A. R., and Kerr, J. N. D. Population imaging of ongoing neuronal activity in the visual cortex of awake rats. *Nature Neuroscience 11*, 7 (2008), 749–751.

[9] Grewe, B. F., Langer, D., Kasper, H., Kampa, B. M., and Helmchen, F. High-speed in vivo calcium imaging reveals neuronal network activity with near-millisecond precision. *Nature Methods 7*, 5 (2010), 399–405.

[10] Holekamp, T. F., Turaga, D., and Holy, T. E. Fast three-dimensional fluorescence imaging of activity in neural populations by objective-coupled planar illumination microscopy. *Neuron 57*, 5 (2008), 661–672.

[11] Hong, N., Kim, B., Lee, J., Choe, H. K., Jin, K. H., and Kang, H. Machine learning-based high-frequency neuronal spike reconstruction from low-frequency and low-sampling-rate recordings. *Nature Communications 15*, 1 (2024), 635.

[12] Hwang, W.-J., Wang, S.-H., and Hsu, Y.-T. Spike detection based on normalized correlation with automatic template generation. *Sensors 14*, 6 (2014), 11049–11069.

[13] Kannan, M., Vasan, G., Huang, C., Haziza, S., Li, J. Z., Inan, H., Schnitzer, M. J., and Pieribone, V. A. Fast, in vivo voltage imaging using a red fluorescent indicator. *Nature Methods 15*, 12 (2018), 1108–1116.

[14] Kim, S., and McNames, J. Automatic spike detection based on adaptive template matching for extracellular neural recordings. *Journal of Neuroscience Methods 165*, 2 (2007), 165–174.

[15] Knöpfel, T., and Song, C. Optical voltage imaging in neurons: moving from technology development to practical tool. *Nature Reviews Neuroscience 20*, 12 (2019), 719–727.

[16] Lin, T., Goyal, P., Girshick, R. B., He, K., and Dollár, P. Focal loss for dense object detection. *CoRR abs/1708.02002* (2017).

[17] Lin, Z., Marin-Llobet, A., Baek, J., He, Y., Lee, J., Wang, W., Zhang, X., Lee, A. J., Liang, N., Du, J., Ding, J., Li, N., and Liu, J. Spike sorting AI agent. *bioRxiv* (2025).

[18] Long, J., Shelhamer, E., and Darrell, T. Fully convolutional networks for semantic segmentation. *CoRR abs/1411.4038* (2014).

[19] Oñativia, J., Schultz, S. R., and Dragotti, P. L. A finite rate of innovation algorithm for fast and accurate spike detection from two-photon calcium imaging. *Journal of Neural Engineering 10*, 4 (2013), 046017.

[20] Park, I. J., Bobkov, Y. V., Ache, B. W., and Principe, J. C. Quantifying bursting neuron activity from calcium signals using blind deconvolution. *Journal of Neuroscience Methods 218*, 2 (2013), 196–205.

[21] Piatkevich, K. D., Bensussen, S., Tseng, H.-a., Shroff, S. N., Lopez-Huerta, V. G., Park, D., Jung, E. E., Shemesh, O. A., Straub, C., Gritton, H. J., Romano, M. F., Costa, E., Sabatini, B. L., Fu, Z., Boyden, E. S., and Han, X. Population imaging of neural activity in awake behaving mice. *Nature 574*, 7778 (2019), 413–417.

[22] Pnevmatikakis, E. A., Soudry, D., Gao, Y., Machado, T. A., Merel, J., Pfau, D., Reardon, T., Mu, Y., Lacefield, C., Yang, W., Ahrens, M., Bruno, R., Jessell, T. M., Peterka, D. S., Yuste, R., and Paninski, L. Simultaneous denoising, deconvolution, and demixing of calcium imaging data. *Neuron 89*, 2 (2016), 285–299.

[23] Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation, 2015.

[24] Roome, C. J., and Kuhn, B. Simultaneous dendritic voltage and calcium imaging and somatic recording from purkinje neurons in awake mice. *Nature Communications 9*, 1 (2018), 3388.

[25] Sebastian, J., Sur, M., Murthy, H. A., and Magimai-Doss, M. Signal-to-signal neural networks for improved spike estimation from calcium imaging data. *PLoS Comput. Biol. 17*, 3 (Mar. 2021), e1007921.

[26] Speiser, A., Yan, J., Archer, E. W., Buesing, L., Turaga, S. C., and Macke, J. H. Fast amortized inference of neural activity from calcium imaging data with variational autoencoders. In *Advances in Neural Information Processing Systems*

(2017), I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30, Curran Associates, Inc.

[27] Vogelstein, J. T., Packer, A. M., Machado, T. A., Sippy, T., Babadi, B., Yuste, R., and Paninski, L. Fast nonnegative deconvolution for spike train inference from population calcium imaging. *Journal of Neurophysiology 104*, 6 (2010), 3691–3704. PMID: 20554834.

[28] Yaksi, E., and Friedrich, R. Reconstruction of firing rate changes across neuronal populations by temporally deconvolved ca2+ imaging. *Nature methods 3* (06 2006), 377–83.