Effective and Efficient Neural Networks for Spike Inference from *In Vivo* Calcium Imaging

Zhanhong Zhou [1], Chung Tin [1, *]

[1] Department of Biomedical Engineering, City University of Hong Kong, Hong Kong.
[*] Corresponding authors: chungtin@cityu.edu.hk

ABSTRACT

Calcium imaging technique provides irreplaceable advantages in monitoring large population of neuronal activities simultaneously. However, due to the generally low signal to noise ratio (SNR) of the calcium signal and variability in dye properties, it is still challenging to faithfully infer neuronal spikes from these calcium signals, especially from in vivo experiments. In this study, we tackled the problem of both spike-rate and spike-event predictions using a data-driven approach, based on a public pool of dataset with simultaneously recorded calcium and electrophysiological signals using different dyes and recorded from different brain regions. We proposed the ENS^2 (effective and efficient neural networks for spike inference from calcium signals) system using raw calcium inputs and it consistently outperforms state-of-the-arts algorithms in both spike-rate and spike-event predictions with reduced computational load. We have also demonstrated that factors such as sampling rates, smoothing window sizes and parametric evaluation metrics could readily bias the interpretation of inference performance. We concluded that optimizing our system for spike-event prediction could produce a more versatile inference system for real neuroscience studies.

Keywords: calcium imaging; spike inference; deep learning; neural networks

INTRODUCTION

One key to understanding the complex functions of the brain is to access to the activities of the countless neurons at the same time. Electrophysiological recordings, such as patch-clamp [1,2] and multielectrode extracellular recording [3], have long been the major method to record neuronal spiking events. These recordings are typically of high temporal resolution and high signal-to-noise ratio (SNR). However, it is technically challenging with these methods to acquire recordings from a large number of neurons stably *in vivo* [4].

In recent decades, the optical-based calcium imaging technique has grown much popularity for *in vivo* neuroscience research [5]. This imaging technique enables simultaneous monitoring of activities of thousands of neurons over a considerable period of time. Moreover, with more effective fluorescent calcium indicators becoming available [6-10] and imaging devices [11-13], 3D

localization of large number of neurons with their individual activities extracted becomes possible in various subcellular structures [14].

Nevertheless, calcium imaging is only an indirect measurement of neuronal activities. In brief, the non-linear changes in concentration of intracellular calcium evoked by neuronal firings are again non-linearly reflected by calcium indicators, whose fluorescent intensities could be imaged. Afterward, the locations of individual neurons or compartments (region of interest, ROI) are identified on images, and the fluctuations of fluorescence signals in ROIs through time are extracted as a surrogate of neuronal activities. Besides, calcium images are commonly noisy, especially for those recorded in deep brain regions *in vivo* or at low light conditions. Further, the high spatial resolution of calcium imaging trades off with indicators' slow temporal dynamics up to hundreds of milliseconds [7,15], resulting in low-pass filtered activities. These indicators come in different types, typically synthetic dyes or genetically encoded calcium indicators (GECIs), and their different dynamics further complicate the task to convert the calcium signal into neuronal signals.

In the past decade, researchers have developed various algorithms to recover multiunit neuronal spikes from the fluorescence signals based on paired data (simultaneously recorded calcium signals and electrophysiology ground truths). For example, [16] proposed a simple and generic deconvolution approach to filter calcium signals with a designated kernel. As one of the state-of-the-art algorithms, [17] proposed a physiologically constrained model with Maximize a Posteriori (MAP) algorithm (MLspike) to infer the most likely spike trains from noisy calcium signals. However, these model-based methods, together with several other works [18-21], require tuning of model parameters for each new recording, either manually or estimated by algorithms. This also holds true for several data-driven algorithms based on supervised learning [22-25], where the recordings used for training and testing have to share some similar properties. In this way, it is difficult to guarantee training data is sufficiently useful, and re-training of algorithms is compulsory prior to actual usage.

For real world applications, a calibration-free inference system that could generalize on un-seen recordings is always preferred. Recently, with the increased availability of paired data, a data-driven method CASCADE was proposed [26]. It was reported that top-ranking performance on un-seen testing datasets was achieved by the neural networks, which were deliberately trained with paired data whose noise-levels matched the testing dataset. In fact, neural networks have shown satisfactory performance in processing bio-signals with severe inter-record variability, including electrocardiogram [27-29] and electromyography [30]. Provided with sufficient amount of paired data, the generalization ability of neural networks makes it a promising approach for inferring spikes from calcium signals.

In this work, we performed thorough research on the impact of each component in the neural networks on the spike inference tasks. The preferred combinations of input types, network architectures, and cost functions were investigated. Here, we developed the ENS^2 (effective and efficient neural networks for spike inference from calcium signals) system with state-of-the-art performance but with lower computational complexity. We also conducted additional simulations to address factors that hindered the performance of deep learning based models for spike inference. During these processes, useful insights were provided which might enlighten future algorithm development.

RESULTS

Raw calcium input with MSE loss is sufficient for spike inferring tasks

For selecting the best method to infer spikes, we evaluated all possible combinations of models. First, we tested how differently pre-processed calcium inputs would benefit the inference task. When measured in Corr (Fig. 2A), raw inputs hold clear leads over either normalized inputs or de-noised inputs. However, its advantages reduce for the other evaluation metrices (Fig. 2B-D & Supp. Fig. 1A-B).

Next, we assessed if the choice of the loss functions would affect the performance of our models. Here, we compared the case with MSE, vRD and Corr as loss functions. Intuitively, we expected that using vRD loss and Corr loss would benefit the performance evaluation using vRD and Corr respectively. Nevertheless, our results show that this is only true for some cases. In fact, the benefit with Corr loss on Corr evaluation is not statistically significant (Fig. 2A). On the other hand, using Corr loss results in undesired vRD, VPD, and ER with either U-Net or FC-Net (Fig. 2B-D), and in fact, the performance (in terms of vRD, VPD, and ER) is always worse than using vRD loss or MSE loss. Similar observation is found when evaluated with Error and Bias (Supp. Fig. 1A-B). The major reason of this is that the Corr loss function fails to differentiate prediction of similar temporal pattern but different amplitude (Supp. Fig. 2D), in such a way that it tends to make a prediction with minimal spike-rate (in amplitude) as compared to other combinations (Supp. Fig. 2B-E). As a consequence, spike-event could not be reliably estimated from the predicted spike-rate due to low signal-to-noise ratio. On the other hand, using vRD loss does not show clear advantage over using MSE loss in terms of performance, even when evaluated with vRD (Fig. 2B). In fact, vRD loss is more costly to use than MSE loss since its computation requires ground truth spike-event in addition to ground truth spike-rate. As a result, we took raw calcium input with MSE loss function in our proposed ENS^2.

U-Net achieves the best overall performance in spike inferring tasks

We then compared three architectures of neural networks (U-Net, Le-Net, and FC-Net). As shown in Fig. 2, U-Net delivers the best overall performance, especially for Corr evaluation. When measured by vRD and ER, slight advantages with U-Net could also be observed over the other two networks, in particular with MSE loss or vRD loss. This also holds when we evaluated the models with Error and Bias (Supp. Fig. 1A-B).

To show that the difference in performance came from the network architecture rather than specific hyper-parameter settings, we repeated the simulation using the combination of U-Net, raw input, and MSE loss with various hyper-parameters (Supp. Fig. 3). The filled bars represent the default hyper-parameter combination used in this study as described in the Method section. Regardlessly, we showed that they all have little effect on the final performance. The Corr approaches 70% and vRD remains less than 3 in all cases. The VPD and ER are around 0.6 and 50%, respectively.

We also proved that our models were trained adequately with our early-stopping criteria (see Method). Supp. Fig. 3G illustrates the MSE training losses for all 20 datasets. The losses decrease with more iterations generally and are stabilized sufficiently as the training stops. Moreover, Supp. Fig. 3 demonstrates that the patience of iterations (see Method) before early-stopping has little influence on performance. Together, we proved that our models had been trained sufficiently with only thousands of iterations to avoid over-fitting.

Comparison to state-of-the-art studies

Based on our investigations above, we selected the combination of U-Net, raw input, and MSE loss as our proposed ENS^2 system. We took this further to compare it with the data-driven system, CASCDE and the model-based system, MLspike. Results are summarized in Fig. 3.

When measured by Corr, vRD, VPD, and ER, Fig. 3A-D shows that the data-driven systems (i.e. our proposed ENS^2 and CASCADE) generally perform better than the model-based system (i.e. MLspike). We showed here that data-driven models could perform better even in VPD and ER (up to 10%) than the model-based method, which is specially designed for spike-event prediction. This may originate from the generalization abilities of data-driven methods, while model-based method relies on auto-calibration of parameters. When compared to CASCADE, our systems also showed superior performance for both spike-rate prediction (Fig. 3A-B, Supp. Fig. 1C-D) and spike-event prediction (Fig. 3C-D). In particular, our ENS^2 shows 10% higher in Corr and 5% lower in ER than CASCADE.

A more detailed comparison is illustrated with record-wise examples for four datasets of different frame rates, calcium indicators and species (Fig. 4). For dataset 9 with the GCaMP6f indicator and a high sampling rate up to 160Hz (Fig. 4A), CASCADE tended to output broad spike-rate prediction, and thus resulting in over-estimation with shifted spike-events (shaded in orange). In contrast, MLspike tends to under-estimate such that quite a number of missing spike events were observed. On the other hand, our system (ENS^2) showed better predictions than both these two methods for all evaluation metrices (bold values on the right in Fig. 4A). For dataset 12 under a similar high frame rate but using the GCaMP6s indicator (Fig. 4B), the calcium signals possess diverse dynamics upon spiking. The prediction task becomes challenging because some spike-events could hardly be identified even by visual inspection. In this case, we find our ENS^2 system predicts spike-rates and spike-events that are closest to the ground truth. Similarly in dataset 13 (Fig. 4C), we found excessive spike-event predictions by both CASCADE and MLspike, probably due to the additive noise in noise matching input or shot noise in raw input. For dataset 6 recorded from zebrafish (Fig. 4D), the frame rate is lower (30Hz). The slow calcium dynamics caused over-estimation in all three methods. However, we found that the ENS^2 system best recovered the temporal firing pattern, predicting three more precise spike-rate windows (shaded in orange) with fewer over-estimated spike-events. Similar advantages of our systems could also be observed in Supp. Fig. 2A-B, where CASCADE and MLspike also tend to over-estimate and under-estimate in dataset 11, respectively. Here, we show that our proposed systems could maintain robust inference capability under varying conditions.

Fig. 2-4, and Supp. Fig. 2 also show that pre-processing input data did not improve performance of our models apparently. For instance, when predicted on dataset 11, explicitly canceling noise (e.g. de-noised input, Supp. Fig. 2E) or matching noise-levels of input data (e.g. CASCADE, Supp. Fig. 2B) did not gain any advantage in the benchmark. They consistently showed worse performance in both spike-rate and spike-event predictions than the ENS^2 system with raw input (Supp. Fig. 2B). Instead, we found that using raw inputs for our proposed model achieves the best inferring results for most cases. We suggest that our data-driven neural networks could handle a considerable range of noise turbulence implicitly by itself, since they perform equally well or even better with raw inputs. This reduces the workload of input pre-processing and feature engineering that were compulsory in traditional machine learning tasks.

We would also like to point out that although our proposed ENS^2 is data-driven, it is less computationally demanding than the previous method (e.g., CASCADE, Table 2 & Fig. 5). For a specific sampling rate (e.g. 60Hz), series of noise matching models were trained in CASCADE to meet the need of different noise levels. Each of their noise matching model consisted of 5 identical networks for ensemble learning to boost performance. On the other hand, only a single network is required in our ENS^2 to predict

data under each sampling rate for various conditions. As a result, the ENS^2 with U-Net requires 20k fewer trainable parameters, and only a maximum of 5.12 million data segments are fed for training, which is fewer than that of CASCADE by two orders (Table 2). In particular, even using an entry-level GPU (Nvidia GTX1650), training of ENS^2 on millions of samples from 19 datasets completed in less than 2.5 minutes on average, and it took just seconds for inference on a testing dataset. These features are not only beneficial to off-the-shelf usage for spike inference, but also enable cost-effective re-training or fine-tuning of our model when more paired datasets are available. Our spike-event estimation algorithm is also more computationally efficient than that used in CASCADE (Fig. 5). When applied on the same spike-rates predictions of all 20 datasets from the ENS^2 system, our algorithm performed faster by one order of magnitude (Fig. 5A), while maintaining similar accuracy (Fig. 5B-C). As such, we show that our ENS^2 demonstrated high performance and good computational efficiency for inferring spikes from calcium data than the state-of-art methods. In fact, while CASCADE took 26sec for spike-event estimation alone, this time is sufficient for the ENS^2 system to complete both spike-rate prediction and spike-event estimation.

DISCUSSION

In this work, we have developed a high performance inference system (ENS^2) through extensive research, and showed its usefulness in inferring both spike-rates and spike-events. To understand further what contributes to good spike inference performance, we investigate how the dataset itself affects the performance of these models.

Fig. 6A-D show the performance with individual dataset achieved by different combinations of models (the combinations are numbered in the same order as in Fig. 2). The results show that performance indeed depends strongly on the dataset. For instance, regardless of the networks used, dataset 19 and 20 achieved notably poor vRD than other datasets (Fig. 6B), and some datasets (e.g., 2-3, 14-15, and 19-20) showed considerably worse ER than the others (Fig. 6D). We extracted various quantities from each dataset, such as noise level, frame rate, firing rate and AP amplitude, and examined how they may affect the inference performance (Fig. 6E-H, Supp. 1M-N). It is shown that AP amplitude of the dataset is the key predictor for the inference performance (including Corr, ER, Error, and Bias). We also noticed that the raw AP amplitudes are of critical importance for inference, which relate the number of spike events for a certain calcium indicator. For example, normalization of calcium inputs might be beneficial when calcium signals fluctuate significantly (e.g. Supp. Fig. 2A), but would also cause under-estimation or over-estimation since the original AP amplitudes are altered. On the other hand, de-noising the inputs may improve the signal-to-noise (SNR) ratio. But it will also smooth out the calcium traces, causing broader spike-rate outputs and less precise spike-event predictions. Matching the inputs' noise-level with the testing dataset is another potentially useful approach to improve inference performance, but it is not always straight-forward to distinguish the AP amplitude and noise amplitude (e.g. Fig. 4B-C & Supp.

Fig. 2A). A previous model-based study also testified that an accurate estimate of AP amplitude improved performance [31]. The AP amplitude indeed strongly depends on the calcium indicators' sensitivity. It is clear that one major bottleneck of inference algorithm is in the calcium indicators.

While Fig. 6 suggests that the sensitivity of calcium indicator is a strong predictor for inference performance, we examined how mixing datasets with different types of calcium indicators, like the one used here, may impact the inference performance. We partitioned the training data according to their calcium indicator type with respect to the testing data. Here, "*All*" is the same as the leave-one-dataset-out setting that we have been using in the other part of this paper. "*Same*" includes only those training datasets with the same calcium indicator as the testing data. "*Different*" includes only those training datasets using distinct calcium indicators from the testing data. Fig. 7A-E and Supp. Fig. 1I-J show that apparently there is no consistent advantage in using dataset with the same calcium indicators for training. Previous study [26] also reported that clustering the same calcium indicators for training showed no advantage. This may suggest that the neural network could calibrate itself over different calcium dynamics in order to generalize to un-seen testing data to certain extent.

Sine our ENS^2 is a data-driven model, we wonder how much training data is needed for achieving good inference performance. We randomly sampled different numbers of segments from the total of over 20 hours of available paired data. Here, we ignored the overlapping in these segments and just assumed that around 68,000 64-sized segments under 60Hz were equivalent to 20 hours of recordings. When supplying all available paired data to the model, the total duration is approximately 20 × 64 hours since the paired data is segmented with a step of 1 (see Method). Not surprisingly, the performance of inference increased with the amount of training data but it converged at a maximal level with roughly 5 hours of paired data (Fig. 7F-I, Supp. Fig. 1G-H).

We have also found that networks with convolutional layers (e.g. U-Net and Le-Net) typically out-performed the other (e.g. FC-Net). This may be partly due to the regularization capability of the convolutional layers. On the other hand, it is quite intuitive for humans to examine the calcium segments fraction by fraction to identify spike-events, just as sliding a kernel for convolution by the artificial neural networks. In fact, a recent data-driven model (CASCADE [26]) with state-of-art performance also used a network with convolutional layers. Surprisingly, although inferring spikes from calcium signals is a typical sequence-to-sequence translation task, recurrent neural networks (e.g., LSTM) did not show any advantages [23]. However, in this work, we used deep convolutional networks with sequence-to-sequence translation ability (e.g., 1D U-Net), and showed that it excelled the other state-of-art systems. The advantages of U-Net may come from its skip-connecting architecture (see Method), which is beneficial to temporal prediction. Recently, a 3D U-Net based model has also been proposed to improve SNR in calcium images

and facilitate calcium signal extraction [32].

Several other factors may also have significant impact on the inference performance, such as sampling rate (resolution of prediction), size of smoothing window (for spike-rate prediction), and hyper-parameter of evaluation metric (e.g. ER window size). The comparisons are summarized in Fig. 8A-D. Fig. 8A shows that the Corr increased consistently with larger smoothing windows. Similar observations can also be found in several recent studies [22,23,33]. This is because the GT spike-rates convolved from the GT spike-events with larger smoothing windows have smoother and broader patterns, which favors the measure of Corr. Fig. 8F shows the GT spike-rates obtained by convolving the GT spike-events in Fig. 8E with varying smoothing windows (25ms to 200ms) and their corresponding predictions. Apparently, the smoother and broader waveform of GT spike-rate (with larger smoothing windows) simplifies the prediction task, and it is easier to obtain a high Corr with such simpler and smoother PD spike-rate waveform. This also holds for the spike-rate evaluation of vRD and Error (Fig. 8B & Supp. Fig. 1E). However, we argue that such resultant "better" performance (e.g. high Corr) would not guarantee meaningful predictions as reflected in the PD spike-events, since multiple GT spike-events could be merged into a single peak of spike-rate (Fig. 8E-F). Instead, the temporal firing patterns could be better predicted with narrower smoothing windows. On the other hand, spike-event predictions (VPD & ER, Fig. 8C-D & Fig. 8G) generally improve with higher sampling rates. This is quite reasonable as smaller bin sizes allow more precise estimation of spike-events from spike-rate predictions. Moreover, when high sampling rates are used (e.g. 30 or 60Hz), VPD and ER would also reduce along with smoothing window sizes, indicating improved spike-event predictions. Here, the spike-event inference performance would possibly be restricted by the overly smoothed spike-rates (e.g. Fig. 8F). We also analyzed the effect of ER window sizes on ER evaluation (Fig. 8G). As expected, reducing ER window sizes strictens the evaluation and produces higher ER, similar to the trend reported in a previous study [17].

Given these analyses, we suggest that our ENS^2 system should be trained with a sampling rate of 60Hz with 25ms smoothing windows for practical use (labeled with white dashed boxes in Fig. 8A-D & Supp. Fig. 1E-F). On one hand, preparing calcium inputs under 60Hz could reduce information loss (see Method), and adopting the 25ms smoothing window could achieve near optimal spike-event inference (Fig. 8C-D). We decided to optimize our system for spike-event prediction rather than spike-rate for the reasons discussed above. On the other hand, the performance of spike-event prediction starts to saturate with this scheme. Further increase in the sampling rate would cause computational over-head while to reduce further the smoothing window size might be harmful to training neural networks with gradient descent. It is worth noting that the CASCADE algorithm (Rupprecht et al., 2021) was indeed benchmarked under 7.5Hz with 200ms smoothing windows. We also trained our ENS^2 under such conditions (labeled in red dashed boxes in Fig. 8A-D & Supp. Fig. 1E-F), and it shows that our ENS^2 consistently outperformed

the CASCADE algorithm in this way for both spike-rate and spike-event predictions (Supp. Fig. 4). These results support that our ENS^2 is a versatile and highly effective algorithm for spike inference from calcium signals.

METHODS

Database

In this study, we used the publicly available datasets containing both calcium imaging signals and simultaneously recorded electrophysiological signals from excitatory neurons [6-10,22,26,34-36]. For benchmarking and algorithm development purposes, they were recently compiled by [26] into an extensive database with 21 datasets. Specifically, we adopted dataset #2 to #21 following [26] for a fair comparison, and they are labeled as dataset 1 to 20 in this study as shown in Table 1. These 20 datasets cover eight different kinds of calcium indicators, a wide range of frame rates (7.7Hz to 500Hz), and various firing rates (0.2Hz to 5.8Hz on average). Over 20 hours of paired ground truth data (calcium signals and spike-events) were recorded from a total of 230 neurons of either mouse or zebrafish brains.

In each dataset, raw calcium signals are provided as the percentage changes of fluorescence amplitude against baseline ($\Delta F/F_0$), while individual timestamps label spike-events. We also computed the noise-levels as defined in [26] and listed them in Table 1. Furthermore, we presented the increase in $\Delta F/F_0$ induced by one action potential (AP amplitude) for each dataset. The AP amplitude is computed using the averaged calcium kernel, which was extracted from paired ground truth data using the deconvolution function with regularized filter in MATLAB.

Data preparation

1) Re-sampling data

To develop and validate the spike inference algorithms, we first re-sampled the input data (both training set and testing set) of different frame rates to the same sampling rates. In this work, we referred to the original frequencies where calcium signals were captured as *frame rates*, and the re-sampled frequencies as *sampling rates*. Given that most of the datasets were captured with frame rates not higher than 60Hz (Table 1), we re-sampled all calcium signals to 60Hz. All the inference systems were then benchmarked under this same sampling rate. We also tested our system under 7.5Hz as suggested by CASCADE in [26]. The impact of sampling rates on inference results is discussed in this work.

2) Pre-processing of calcium signals

Aside from the raw calcium inputs (where only re-sampling is performed), we also considered several pre-processed inputs. First, since normalization is beneficial in back-propagation [37], we prepared the "normalized inputs" by rescaling the amplitude of the raw inputs to $[0,1]$ on a record-by-record basis. Second, considering that calcium signals are intrinsically noisy [14], we computed the "de-noised inputs" by down-sampling the raw inputs to 6Hz and then up-sampling them to the required sampling rates. This should be sufficient to preserve the spiking properties as we noted that the firing rates of all datasets are well below 6Hz (Table 1). Moreover, the "noise matching inputs" were obtained from [26] to reproduce results with the CASCADE algorithm, where the algorithm is designed to have the noise-levels of training calcium data matching with those of the testing data.

3) Pre-processing of spike-rate

For a pre-defined sampling rate (e.g. 60Hz), raw timestamps of ground truth spike (spike-events) are re-allocated into their corresponding time bins. We can then compute the sequence of spike counts by counting the total firing events in each time bin. Note that the different pre-processed calcium inputs (raw/normalized/de-noised/noise matching) considered here share identical spike count sequences. The sequences are then smoothed with Gaussian filters to facilitate gradient descent. The smoothing window size τ for the Gaussian kernels was set to 25ms, which produces the optimal spike-event predictions with high temporal resolution in general. The selection of smoothing window size for deep learning based systems is also carefully studied. The convolved spike counts are denoted as "spike-rate" in this work.

4) Data segmentation

To train the neural networks properly, paired sequences of calcium signals and spike-rates were segmented with a moving step of 1. The length of each segment was set to 64 samples, ensuring that each contains at least 1 sec of information (e.g. for a sampling rate of 60Hz, 1sec of data consists of 60 data points). In the case of a sampling rate of 60Hz, a total of >4 million segments of paired data were obtained for training.

Network architectures

We tested three different architectures of neural networks to evaluate their effectiveness in the spike inference task. All of them (U-Net [38], Le-Net [39], and FC-Net (fully-connected network)) are typical representations of their own categories (deep neural network (DNN), convolutional neural network (CNN), and typical neural network (NN)), respectively. We adopted these existing models with minor modifications for 1-D calcium signal inputs. The network architectures are summarized in Supp. Table 1.

The U-Net used in this study contains three contracting blocks and expanding blocks. On one hand, the input information from contracting blocks passes through the bottleneck block to the expanding blocks. On the other hand, skip-connections from contracting blocks to the corresponding expanding blocks allow direct and localized information flows [38]. Within each contracting/expanding block and the bottleneck block, two convolution layers with 3-sized kernels are deployed. Instead of batch normalization, we used instance normalization [40] for regularization, since calcium signals with various dynamics may co-exist in a same batch of data. The Le-Net consists of three convolution layers with kernel sizes of 3, 3, and 6, respectively. Average pooling layers with 2-sized kernels are applied between the three convolution layers. A dropout layer [41] is included before the output layer for regularization. A typical fully-connected network with four hidden layers and two dropout layers is adopted as FC-Net. All three networks are designed to take 64-sized calcium signal inputs, and output 64-sized spike-rate vectors in a sequence-to-sequence translation manner. During prediction, given that the input data are segmented with steps of 1, each time point is indeed predicted for up to 64 times separately from its adjacent segments. We, thus, were able to average these predictions for a robust final spike-rate output of each time point. Spike-event output could then be estimated from this final spike-rate sequence as introduced below.

We kept all three networks to have similar numbers (under 150k) of trainable parameters for comparison (Table 2). They are all randomly initiated to have zero means and standard deviations of 0.02. Leaky ReLU (rectified linear units) with slopes of 0.2 is used as activation functions for all layers except for the output layers, where ReLU is used for non-negative spike-rate prediction.

Loss functions and optimization

For each type of network, we optimized them with three different loss functions, respectively, for comparison. First, mean square error (MSE) loss is used, which is one of the most commonly used loss functions applicable to a wide variety of machine learning tasks. The models are expected to minimize the MSE between predicted spike-rates and ground truth spike-rates, penalizing the prediction both in time and amplitudes. In addition, we used two of the evaluation metrics (see below), correlation (Corr) and van Rossum distance (vRD) [42], as loss functions. Through gradient descents, the models are expected to maximize the performance measured by these two metrics.

The Adam optimizer [43] with a default learning rate of 1e-3 is used for all models. Each model is allowed to update for a maximum of 5000 iterations. In each iteration, a batch of 1024 paired segments is drawn randomly and fed to the model for

training. The training losses are noted, and early-stopping is introduced when the losses do not improve in the past 500 interactions ($patience = 500$). Under these criteria, we observed that most models completed the trainings within 2500 iterations. The resultant models are then ready for prediction. Other details of hyper-parameters and operational environment are summarized in Supp. Table 2.

Estimation of spike-events from spike-rate predictions

To reliably convert the spike-rates output by the neural networks to spike-event predictions, we propose an un-supervised algorithm that is simple and straight-forward (Fig. 1). The workflow is briefly introduced here.

Step 1: Fragments of spike-rate predictions (*pd_rates*) with non-zero spike rate are identified by thresholding the spike-rate sequence output with an epsilon value. We do not use zero threshold to avoid including any fragment with overly low peak amplitude (i.e. those showing extremely small spiking probabilities or background noise), where no spike should be estimated.

Step 2: For each *pd_rate* of length L (in terms of number of data points), we initialize a zero-filled vector (*est_spike*) with L bins.

Step 3: One spike is assigned to any one bin in *est_spike* at one time and convolve it into a spike-rate vector (*est_rate*) in a same way as we have described above. Then the MSE between the resultant *est_rate* and *pd_rate* is calculated. This step is implemented parallelly for all L bins to determine the most suitable bin (i.e. with the smallest MSE) for assigning the spike.

We then repeat step 3 to assign another spike each time to the most suitable bin, until the MSE would no longer be reduced by adding a spike to any location in *est_spike*. Then the updated *est_spike* is regarded as the final estimation of spike events for the concerned *pd_rate* fragments. The timestamp of a spike is defined as the center time of the corresponding bin within *est_spike*. If multiple spikes are predicted in the same bin, the same timestamp is repeated accordingly.

For a spike-rate sequence output with N fragments of *pd_rates*, this algorithm executes in $O(N \times L \times k)$ time, where k is the maximum number of spikes in any one bin. In practice, considering the typically slow dynamics of calcium signals and relatively low firing rates of neurons imaged, this estimation method operates in linear time in proportional to the duration of recordings. We have validated this spike-events estimation algorithm with the Monte-Carlo importance sampling based algorithm proposed in CASCADE [26].

Evaluation metrics

How to reliably assess the performance of the spike inference tasks remains an open topic, where a single evaluation metric could be biased in certain aspects [22,23,25,26]. In this regard, recent studies proposed to employ multiple metrics to supplement each other [17,22-26]. In this work, we used four metrics to examine spike-rates prediction and two others for spike-events prediction.

Firstly, Pearson correlation coefficient (Corr) is used as the primary metric for comparing similarities of spike rates as follow:

$$Corr_{GT,PD} = \frac{E[(GT-\mu_{GT})(PD-\mu_{PD})]}{\sigma_{GT}\sigma_{PD}} \tag{1}$$

where GT and PD stand for ground truth and prediction, respectively. Secondly, we use the van Rossum distance (vRD) [42] for the evaluation of spike rates prediction:

$$vRD_{GT,PD} = \sqrt{\frac{1}{\tau}\int[GT(t) - PD(t)]^2\, dt} \tag{2}$$

where the time constant τ is the normalizing factor (smoothing window size) for smoothing spike-events into spike-rates (e.g., $\tau = 0.025s$ for our proposed system). Moreover, Error and Bias proposed in [26] are also used to evaluate spike-rates:

$$Error = \frac{\int|PD-GT|dt}{\int GTdt} \tag{3}$$

$$Bias = \frac{\int|PD-GT|\, dt\,|_{PD(t)>GT(t)} - \int|PD-GT|\, dt\,|_{PD(t)<GT(t)}}{\int GTdt} \tag{4}$$

On the other hand, for measuring spike-event prediction, we adopt the Victor-Purpura distance (VPD) [44]. It is defined as the minimal cost to transform the PD spike-events to the GT spike-events. The cost for either inserting or deleting a spike equals 1, while shifting a spike by Δt costs $q|\Delta t|$. We use the default value $q = 1$ in this work. To make comparison across different datasets, we present the VPD as the minimal total cost divided by the total number of GT spikes.

Lastly, we compute the error rate (ER) as below [17,24,31], which measures the F_1 score of the predicted spike-events:

$$ER_{GT,PD} = 1 - F_1 = 1 - 2\frac{sensitivity \times precision}{sensitivity + precision} \tag{5}$$

$$sensitivity = \frac{true\ positive}{true\ positive + false\ negative} \tag{6}$$

$$precision = \frac{true\ positive}{true\ positive + false\ positive} \tag{7}$$

The GT spike-events and PD spike-events are matched based on their VPD. Here, a spike is said to be correctly predicted if it co-exists with its real counterpart within a time window of 50ms (defined as the ER window size). This time window is one order smaller than that used in previous study [17], suggesting a much more stringent assessment of model performance in this study. We also examined the effect of ER window sizes in this work.

Experiments and implementation

As described above, this study involves combinations from three types of calcium inputs, three neural network models, and three loss functions, resulting in a total of 27 combinations of models. The best performing system as evaluated by the six metrics described in the Method section was then benchmarked against the state-of-the-art algorithms, including the data-driven method, CASCADE [26] and the model-based method, MLspike [17].

Our simulations followed the leave-one-dataset-out protocol. In brief, the model is first trained on 19 datasets and tested on the remaining one. This is repeated 20 times such that all the 20 datasets are tested respectively.

For the CASCADE algorithm, we follow the protocol as described in [26]. For each dataset, noise matching inputs are constructed (artificial noise is added to the 19 training datasets to match the noise-level of the remaining one tested on), and five identical models are trained separately for 10 epochs. The averaged outputs of these five models are regarded as the final spike-rate predictions. Spike-event predictions are estimated using a Monte-Carlo importance sampling based algorithm in CASCADE [26].

For the MLspike algorithm [17], raw calcium inputs are used. The drift ($\eta = 0.01$) and non-linearity parameters (saturation $\gamma = 0.1$ for synthetic dyes, polynomial coefficient $[p_2, p_3] = [1.0, 0.0]$ for the rest indicators) are set manually with prior knowledge. The values of the model parameters of A (action potential (AP) amplitude), τ (calcium decay time constant), and σ (noise amplitude) are obtained using its built-in auto-calibration algorithm, since manual calibration is impossible in actual usage without ground truth paired data. For a fair comparison, in addition to the direct spike-event outputs from MLspike, we also convolve them into spike-rate outputs in the same way as described in the Method section.

REFERENCE

1 Neher, E. & Sakmann, B. Single-channel currents recorded from membrane of denervated frog muscle fibres. *Nature* **260**, 799-802, doi:10.1038/260799a0 (1976).

2 Hamill, O. P., Marty, A., Neher, E., Sakmann, B. & Sigworth, F. J. Improved patch-clamp techniques for high-resolution current recording from cells and cell-free membrane patches. *Pflügers Archiv* **391**, 85-100, doi:10.1007/BF00656997 (1981).

3 Spira, M. E. & Hai, A. Multi-electrode array technologies for neuroscience and cardiology. *Nat. Nanotechnol.* **8**, 83-94, doi:10.1038/nnano.2012.265 (2013).

4 Buzsáki, G. Large-scale recording of neuronal ensembles. *Nat. Neurosci.* **7**, 446-451, doi:10.1038/nn1233 (2004).

5 Kerr, J. N. D. & Denk, W. Imaging in vivo: watching the brain in action. *Nat. Rev. Neurosci.* **9**, 195-205, doi:10.1038/nrn2338 (2008).

6 Akerboom, J. *et al.* Optimization of a GCaMP Calcium Indicator for Neural Activity Imaging. *The Journal of Neuroscience* **32**, 13819, doi:10.1523/JNEUROSCI.2601-12.2012 (2012).

7 Chen, T.-W. *et al.* Ultrasensitive fluorescent proteins for imaging neuronal activity. *Nature* **499**, 295-300, doi:10.1038/nature12354 (2013).

8 Tada, M., Takeuchi, A., Hashizume, M., Kitamura, K. & Kano, M. A highly sensitive fluorescent indicator dye for calcium imaging of neural activity in vitro and in vivo. *European Journal of Neuroscience* **39**, 1720-1728, doi:10.1111/ejn.12476 (2014).

9 Dana, H. *et al.* Sensitive red protein calcium indicators for imaging neural activity. *eLife* **5**, e12727, doi:10.7554/eLife.12727 (2016).

10 Bethge, P. *et al.* An R-CaMP1.07 reporter mouse for cell-type-specific expression of a sensitive red fluorescent calcium indicator. *PLoS One* **12**, e0179460, doi:10.1371/journal.pone.0179460 (2017).

11 Denk, W., Strickler, J. H. & Webb, W. W. Two-photon laser scanning fluorescence microscopy. *Science* **248**, 73, doi:10.1126/science.2321027 (1990).

12 Stosiek, C., Garaschuk, O., Holthoff, K. & Konnerth, A. In vivo two-photon calcium imaging of neuronal networks. *Proceedings of the National Academy of Sciences* **100**, 7319, doi:10.1073/pnas.1232232100 (2003).

13 Sofroniew, N. J., Flickinger, D., King, J. & Svoboda, K. A large field of view two-photon mesoscope with subcellular resolution for in vivo imaging. *eLife* **5**, e14472, doi:10.7554/eLife.14472 (2016).

14 Grienberger, C. & Konnerth, A. Imaging Calcium in Neurons. *Neuron* **73**, 862-885, doi:10.1016/j.neuron.2012.02.011 (2012).

15 Kerr, J. N. D., Greenberg, D. & Helmchen, F. Imaging input and output of neocortical networks in vivo. *Proceedings of the National Academy of Sciences of the United States of America* **102**, 14063, doi:10.1073/pnas.0506029102 (2005).

16 Yaksi, E. & Friedrich, R. W. Reconstruction of firing rate changes across neuronal populations by temporally deconvolved Ca2+ imaging. *Nat. Methods* **3**, 377-383, doi:10.1038/nmeth874 (2006).

17 Deneux, T. *et al.* Accurate spike estimation from noisy calcium signals for ultrafast three-dimensional imaging of large neuronal populations in vivo. *Nat. Commun.* **7**, 12190, doi:10.1038/ncomms12190 (2016).

18 Vogelstein, J. T. *et al.* Fast Nonnegative Deconvolution for Spike Train Inference From Population Calcium Imaging. *Journal of Neurophysiology* **104**, 3691-3704, doi:10.1152/jn.01073.2009 (2010).

19 Friedrich, J., Zhou, P. & Paninski, L. Fast online deconvolution of calcium imaging data. *PLoS Comput. Biol.* **13**, e1005423, doi:10.1371/journal.pcbi.1005423 (2017).

20 Pachitariu, M. *et al.* Suite2p: beyond 10,000 neurons with standard two-photon microscopy. *bioRxiv*, 061507, doi:10.1101/061507 (2017).

21 Grewe, B. F., Langer, D., Kasper, H., Kampa, B. M. & Helmchen, F. High-speed in vivo calcium imaging reveals neuronal network activity with near-millisecond precision. *Nat. Methods* **7**, 399-405, doi:10.1038/nmeth.1453 (2010).

22 Theis, L. *et al.* Benchmarking Spike Rate Inference in Population Calcium Imaging. *Neuron* **90**, 471-482, doi:10.1016/j.neuron.2016.04.014 (2016).

23 Berens, P. *et al.* Community-based benchmarking improves spike rate inference from two-photon calcium imaging data. *PLoS Comput. Biol.* **14**, e1006157, doi:10.1371/journal.pcbi.1006157 (2018).

24 Hoang, H. *et al.* Improved hyperacuity estimation of spike timing from calcium imaging. *Sci. Rep.* **10**, 17844, doi:10.1038/s41598-020-74672-y (2020).

25 Sebastian, J., Sur, M., Murthy, H. A. & Magimai-Doss, M. Signal-to-signal neural networks for improved spike estimation from calcium imaging data. *PLoS Comput. Biol.* **17**, e1007921, doi:10.1371/journal.pcbi.1007921 (2021).

26 Rupprecht, P. *et al.* A database and deep learning toolbox for noise-optimized, generalized spike inference from calcium imaging. *Nat. Neurosci.*, doi:10.1038/s41593-021-00895-5 (2021).

27 Zhai, X. & Tin, C. Automated ECG classification using dual heartbeat coupling based on convolutional neural network. *IEEE Access* **6**, 27465-27472, doi:10.1109/access.2018.2833841 (2018).

28 Zhai, X., Zhou, Z. & Tin, C. Semi-supervised learning for ECG classification without patient-specific labeled data. *Expert Syst. Appl.* **158**, 113411, doi:10.1016/j.eswa.2020.113411 (2020).

29 Zhou, Z., Zhai, X. & Tin, C. Fully automatic electrocardiogram classification system based on generative adversarial network with auxiliary classifier. *Expert Syst. Appl.* **174**, 114809, doi:10.1016/j.eswa.2021.114809 (2021).

30 Zhai, X., Jelfs, B., Chan, R. H. M. & Tin, C. Self-recalibrating surface EMG pattern recognition for neuroprosthesis control based on convolutional neural network. *Front. Neurosci.* **11**, doi:10.3389/fnins.2017.00379 (2017).

31 Éltes, T., Szoboszlay, M., Kerti-Szigeti, K. & Nusser, Z. Improved spike inference accuracy by estimating the peak amplitude of unitary [Ca2+] transients in weakly GCaMP6f-expressing hippocampal pyramidal cells. *The Journal of Physiology* **597**, 2925-2947, doi:10.1113/jp277681 (2019).

32 Li, X. *et al.* Reinforcing neuron extraction and spike inference in calcium imaging using deep self-supervised learning. *bioRxiv*, 2020.2011.2016.383984, doi:10.1101/2020.11.16.383984 (2020).

33 Stringer, C. & Pachitariu, M. Computational processing of neural recordings from calcium imaging data. *Curr. Opin. Neurobiol.* **55**, 22-31, doi:10.1016/j.conb.2018.11.005 (2019).

34 Huang, L. *et al.* Relationship between simultaneously recorded spiking activity and fluorescence signal in GCaMP6 transgenic mice. *bioRxiv*, 788802, doi:10.1101/788802 (2020).

35 Khan, A. G. *et al.* Distinct learning-induced changes in stimulus selectivity and interactions of GABAergic interneuron classes in visual cortex. *Nat. Neurosci.* **21**, 851-859, doi:10.1038/s41593-018-0143-z (2018).

36 Schoenfeld, G., Carta, S., Rupprecht, P., Ayaz, A. & Helmchen, F. In vivo calcium imaging of CA3 pyramidal neuron populations in adult mouse hippocampus. *bioRxiv*, 2021.2001.2021.427642, doi:10.1101/2021.01.21.427642 (2021).

37 Lecun, Y., Bottou, L., Orr, G. B. & MÜLler, K. R. in *NIPS workshop (1996) Lecture notes in computer science* 9-50 (Springer, Berlin, 1998).

38 Ronneberger, O., Fischer, P. & Brox, T. in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015.* (eds Nassir Navab, Joachim Hornegger, William M. Wells, & Alejandro F. Frangi) 234-241 (Springer International Publishing).

39 Lecun, Y., Bottou, L., Bengio, Y. & Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **86**, 2278-2324, doi:10.1109/5.726791 (1998).

40 Ulyanov, D., Vedaldi, A. & Lempitsky, V. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022* (2016).

41 Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* **15**, 1929-1958 (2014).

42 van Rossum, M. C. A novel spike distance. *Neural Comput* **13**, 751-763, doi:10.1162/089976601300014321 (2001).

43 Kingma, D. P. & Ba, J. in *3rd International Conference on Learning Representations (ICLR)* (San Diego, CA, USA, 2015).

44 Victor, J. D. & Purpura, K. P. Nature and precision of temporal coding in visual cortex: a metric-space analysis. *Journal of Neurophysiology* **76**, 1310-1326, doi:10.1152/jn.1996.76.2.1310 (1996).

ACKNOWLEDGEMENT

This work was supported by Research Grants Council of Hong Kong SAR (Project CityU 11104220) and City University of Hong Kong (Project 7005645). Data and source codes to replicate the primary results will be shared upon reasonable request.

AUTHOR CONTRIBUTIONS

C.T. conceived the study. Z.Z. designed the algorithm and performed analyses. C.T. and Z.Z. interpreted the results and wrote the manuscript.

COMPETING INTERESTS

The authors declare no competing interests.

FIGURES AND TABLES

Fig. 1. Schematic workflow of the un-supervised algoithm for estimating spike-events from spike-rate predictions.

Fig. 2. Performance comparison of models with various model architectures, loss functions, and pre-processed inputs.

Fig. 3. Performance comparison of the proposed method against state-of-the-art algorithms.

Fig. 4. Examples of spike-rates and spike-events prediction among the proposed methods and state-of-the-art methods.

Fig. 5. Comparison between the spike-event estimation algorithms in ENS^2 and CASCADE.

Fig. 6. Effect of dataset properties on spike inference performance.

Fig. 7. Effect of calcium indicators and data quantity on the spike inference performance.

Fig. 8. Effect of the strictness of evaluation metrics on spike inference performance.

Table 1. Summary of datasets used in this study (adapted from Rupprecht et al., 2021).

Table 2. Comparison of the proposed models and previous study.

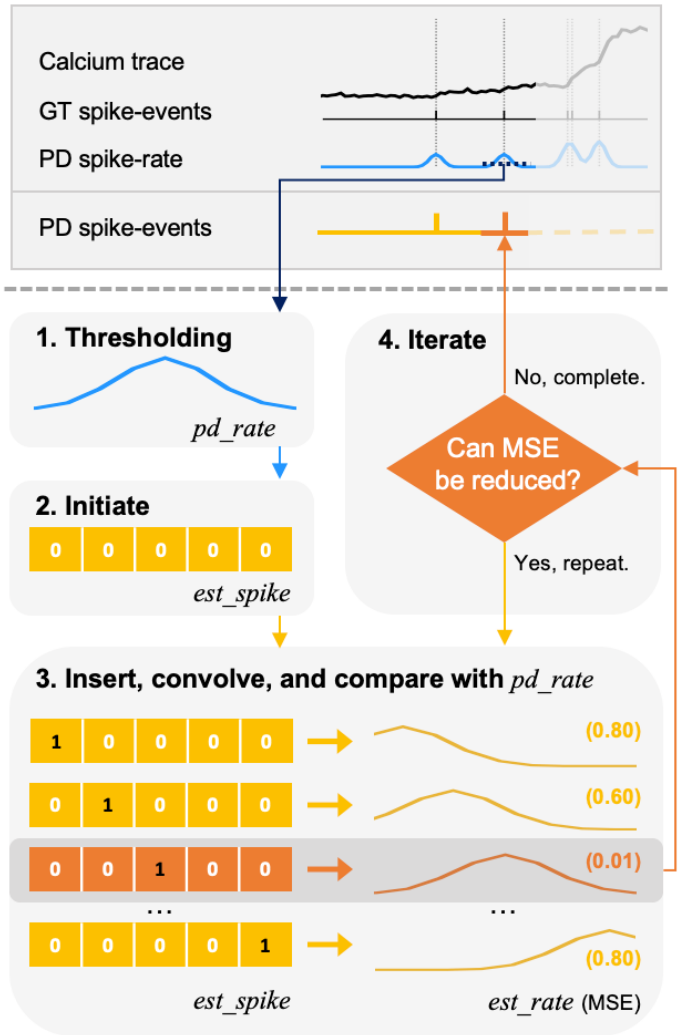

Fig. 1. Schematic workflow of the un-supervised algorithm for estimating spike-events from spike-rate predictions. For a given segment of calcium trace input, our ENS^2 system will first predict the corresponding spike-rate. Afterward, spike-events are estimated by the illustrated four-step algorithm. In brief, valid fragments of spike-rate prediction are extracted by thresholding. Then, spike-event is inserted tentatively to approximate the extracted spike-rate fragment. The resultant spike-events sequence that achieves the minimal MSE is regarded as the final spike-events prediction. Details are explained in the Method section.

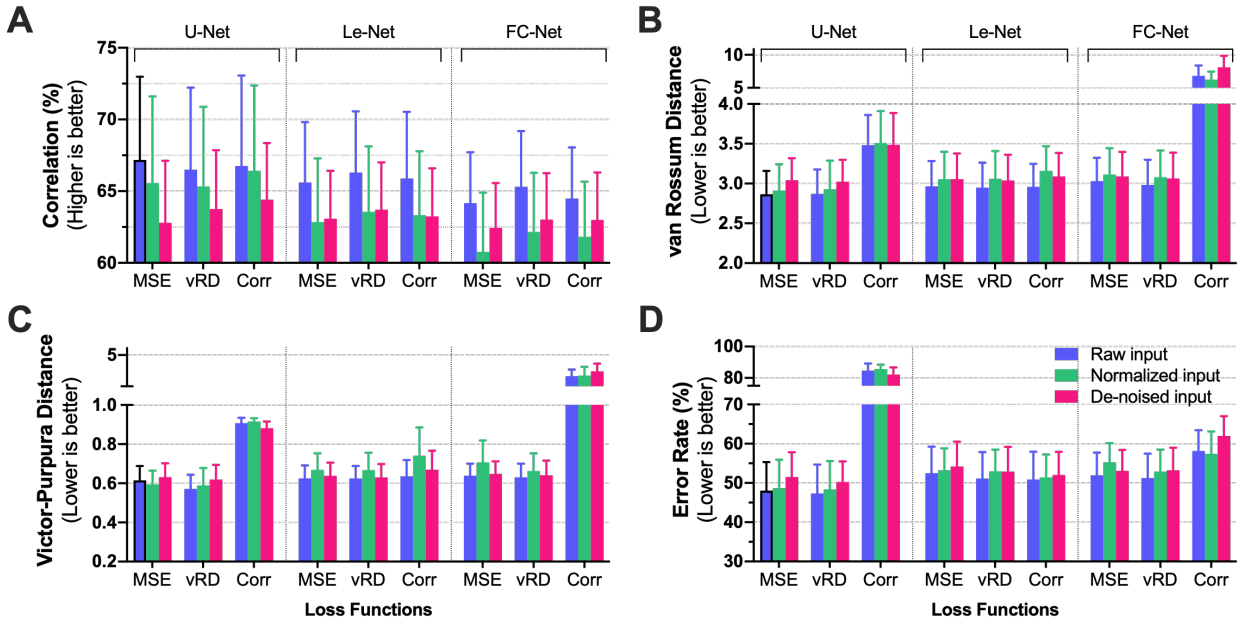

Fig. 2. Performance comparison of models with various model architectures, loss functions, and pre-processed inputs. Performances are measured in (**A**) correlation, (**B**) van Rossum distance, (**C**) Victor-Purpura distance, and **(D**) error rate, respectively. Generally, U-Net could achieve the best inference performance for both spike-rate prediction and spike-event prediction. MSE loss delivers the most cost-effective results, while raw input shows advantages in correlation evaluation. Details are discussed in the Result section. Bar plots present means with 95% confidence intervals. Black squares denote the best performing combination in general.

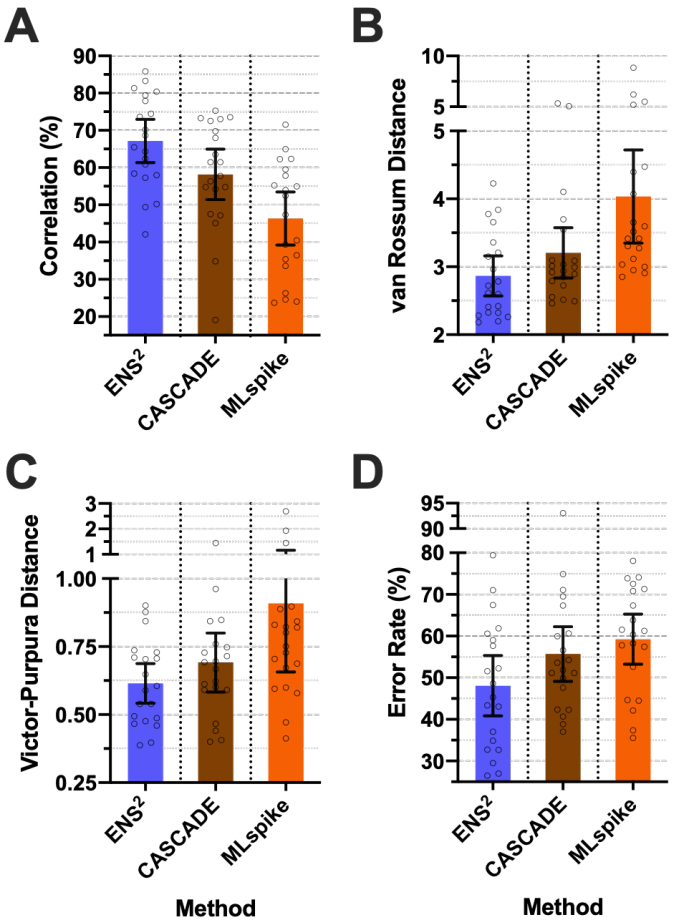

Fig. 3. Performance comparison of the proposed method against state-of-the-art algorithms. Performance is measured in (**A**) correlation, (**B**) van Rossum distance, (**C**) Victor-Purpura distance, and (**D**) error rate, respectively. Our proposed ENS^2 system shows clear advantages over the data-driven method CASCADE, and the model-based method MLspike in all evaluation metrics. Empty dots mark the performance on individual datasets.

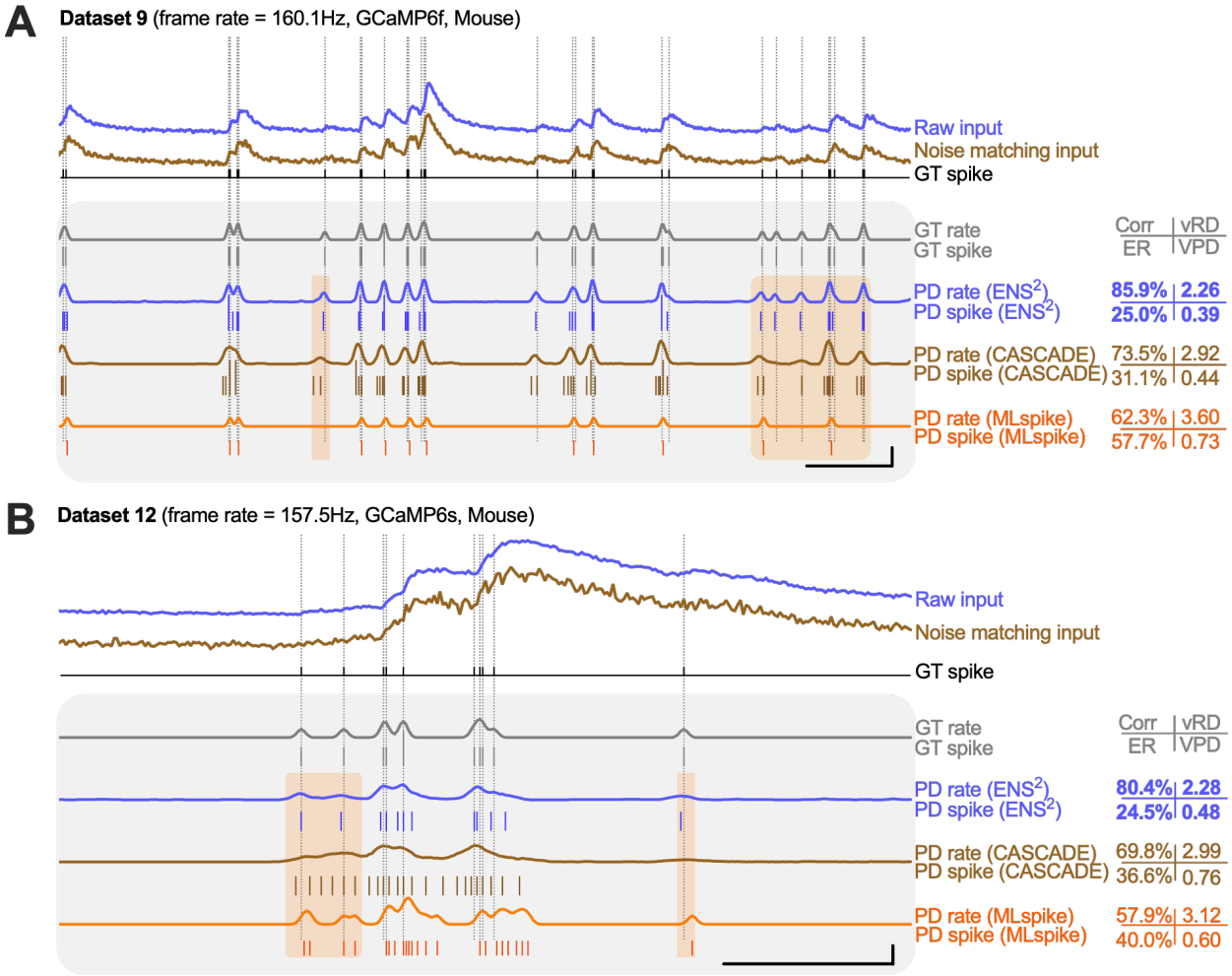

Fig. 4. Examples of spike-rates and spike-events prediction among the proposed methods and state-of-the-art methods. In each subfigure, raw calcium inputs (blue) and noise matching inputs (brown) are shown with the ground truth spike-events (black) on top. The predicted spike-rates and corresponding spike-events (colored) by various methods are aligned below. Metrics on the right measure the performance on the whole dataset. Orange shaded areas represent regions of interest where discrepancies in predictions are observed among different methods.

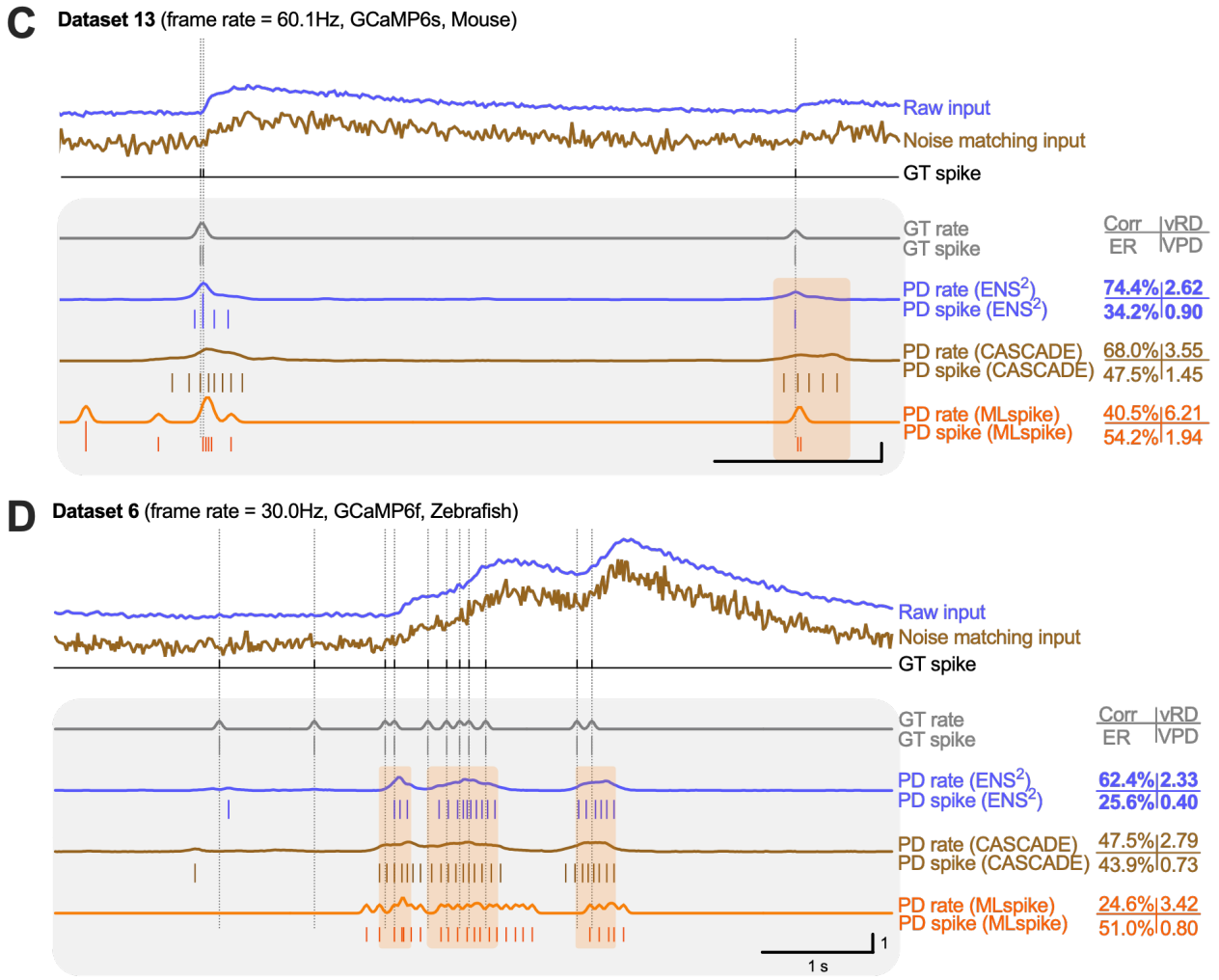

Fig. 4. Examples of spike-rates and spike-events prediction among the proposed methods and state-of-the-art methods (continued).

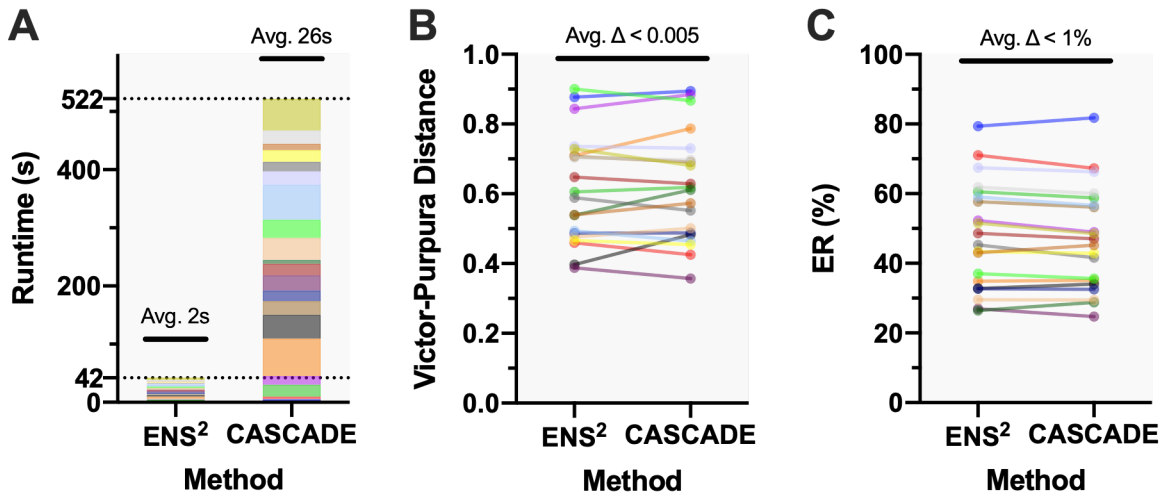

Fig. 5. Comparison between the spike-event estimation algorithms in ENS^2 and CASCADE. (**A**) Runtime spent on estimating spike-events for all 20 datasets under 60Hz. The estimation algorithm in ENS^2 is over one order faster than that in CASCADE. (**B**) VPD measured from spike-events estimated by the two different algorithms, showing comparable estimation performance. The same spike-rate predictions from ENS^2 were supplied as inputs for fair comparison. (**C**) Same as (B), but for ER measurement. Different colors denote 20 individual datasets. The runtime was measured on a PC with an Intel Core i7-4770 CPU.

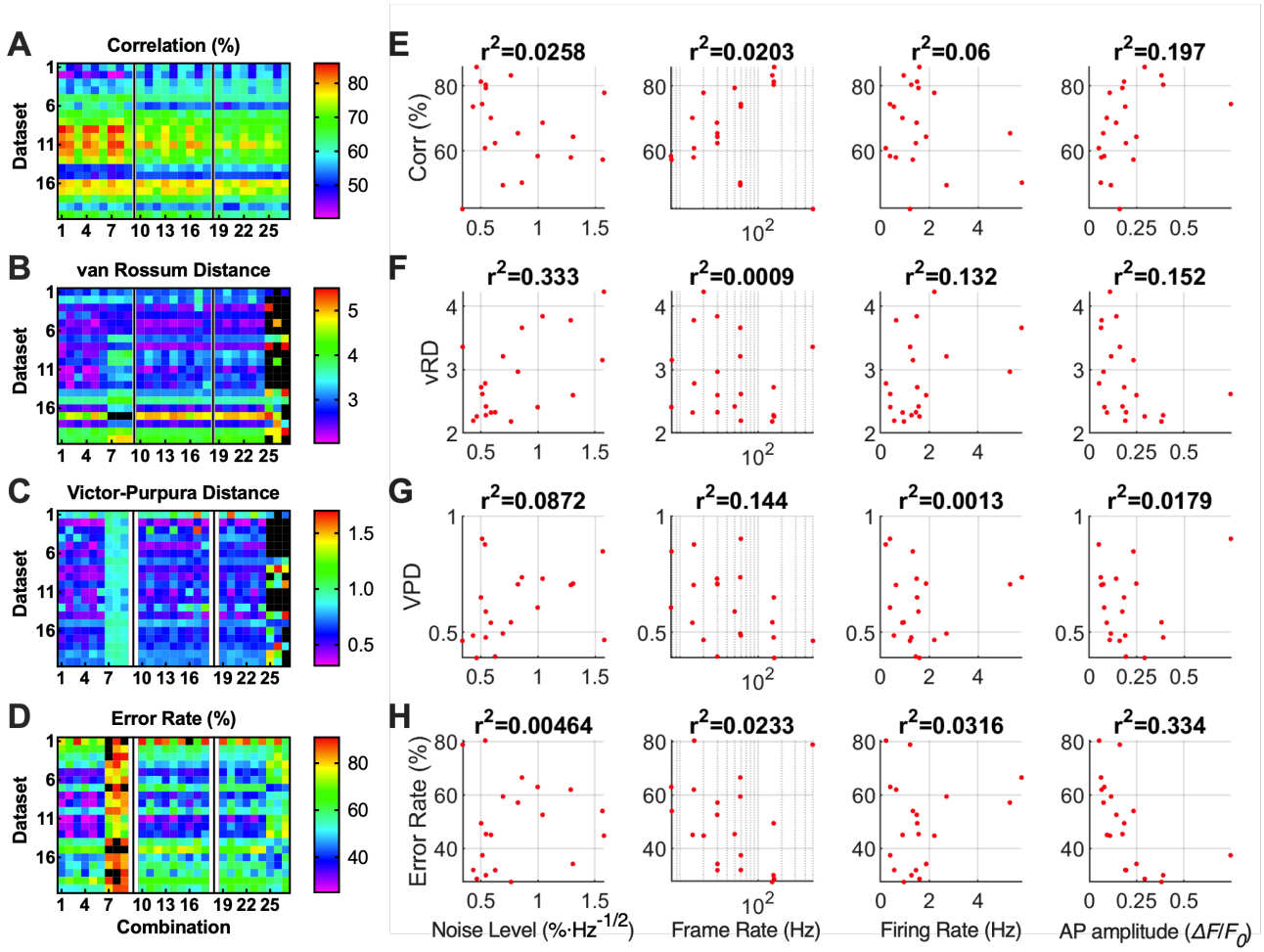

Fig. 6. Effect of dataset properties on spike inference performance. (**A-D**) Overview of different combinations of methods performing on each dataset. Different combinations are arranged horizontally in a same order as in Fig. 2. The inference performance is affected by both the method combinations and the dataset's properties. (**E-H**) Pearson correlation coefficients between the properties of each dataset and their achieved inferring performance. AP amplitude is the relatively major property that affects both spike-rate and spike-event predictions.

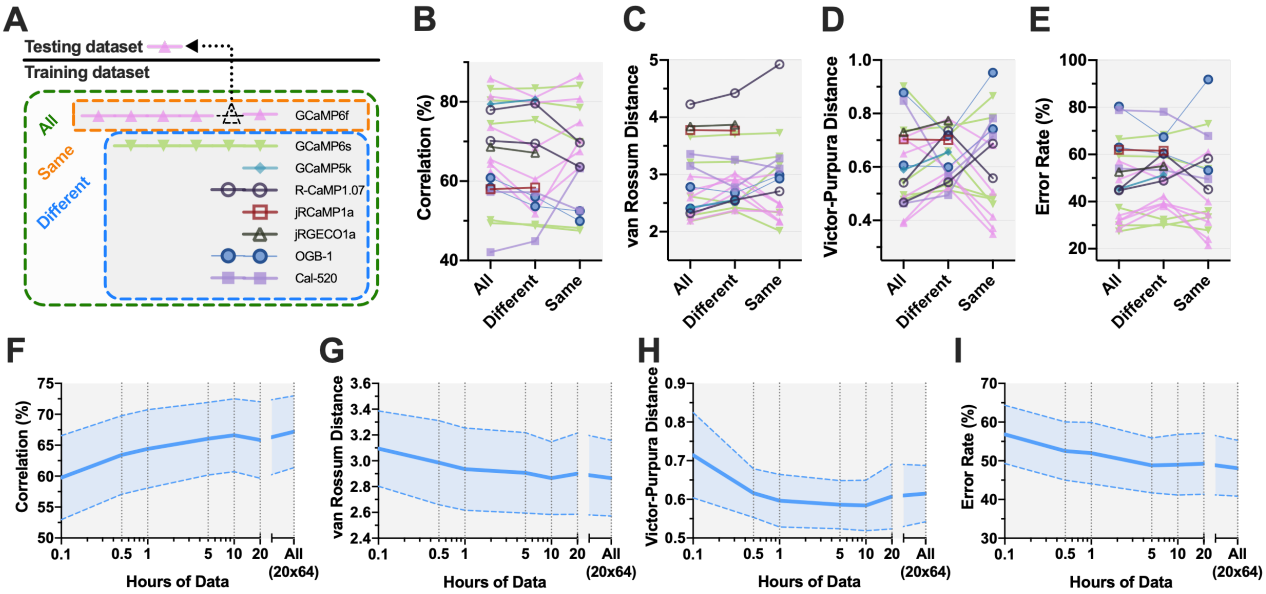

Fig. 7. Effect of calcium indicators and data quantity on the spike inference performance. (**A**) Example showing the training datasets included by *All*, *Same*, or *Different* method when a GCaMP6f dataset was regarded as testing dataset. See the Discussion section for detailed definitions. (**B-E**) Performance of correlation, van Rossum distance, Victor-Purpura distance, and error rate when only selected training dataset based on calcium indicators were provided. No consistent trend could be observed by selecting training datasets based on calcium indicators. (**F-I**) Performance of correlation, van Rossum distance, Victor-Purpura distance, and error rate when only portions of data were provided. The inference performance generally improved with more paired data but saturated at around five hours of data used. Shaded areas denote means with 95% confidence intervals.

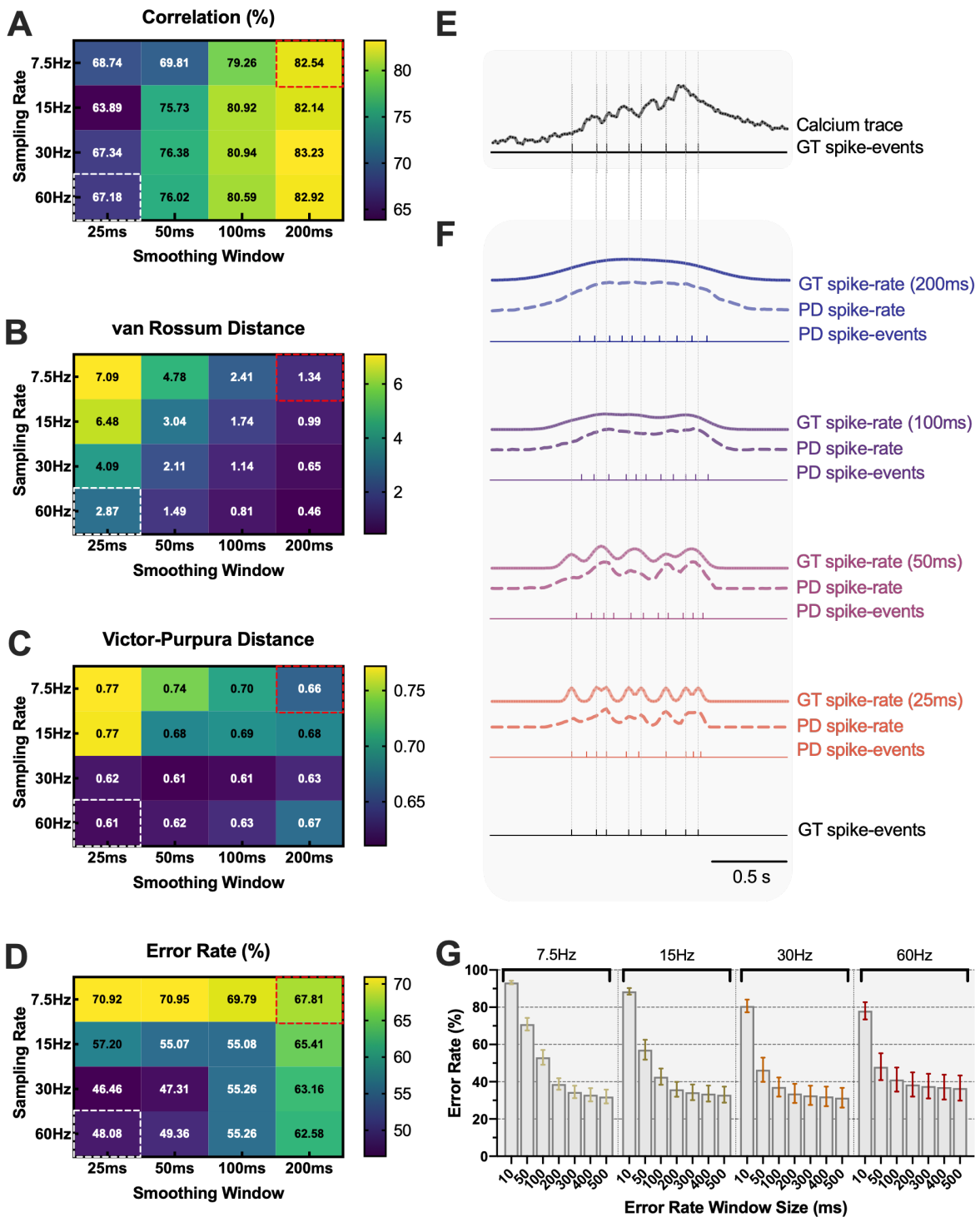

Fig. 8. Effect of the strictness of evaluation metrics on spike inference performance. (**A-D**) Performance of correlation, van Rossum distance, Victor-Purpura distance, and error rate, respectively, when measured under different sampling rates and smoothing window sizes. White and red squares denote the evaluation schemes adopted by ENS[2] and CASCADE, respectively. (**E**) Example of calcium signals under 60Hz with paired spike-events. (**F**) Examples of ground truth spike-rates convolved with different smoothing window sizes (from 200ms to 25ms). The resultant spike-rate and spike-event predictions are also shown. (**G**) Performance of error rate when measured under different sampling rates and error rate window sizes. Details are in the Discussion section.

Table 1. Summary of datasets used in this study (adapted from Rupprecht et al., 2021)

Data set	Calcium indicator	Animal	Frame rate (Hz)	Firing rate (Hz)	Noise level (%Hz$^{-1/2}$)	AP amplitude ($\Delta F/F_0$)	Neuron #	Trial #	Duration (min.)
1	OGB-1	Mouse	15.1±1.9	0.2±0.2	0.5±0.1	0.052	16	47	116
2	Cal-520		500.0±0.0	1.2±0.8	0.3±0.1	0.161	8	161	22
3	OGB-1	Zebrafish	7.7±0.1	0.4±0.5	1.0±0.2	0.080	15	45	81
4	Cal-520		7.8±0.0	1.3±1.9	1.6±0.5	0.233	5	16	31
5	GCaMP6f		30.0±0.0	1.9±0.6	1.3±0.7	0.249	8	23	45
6			30.0±0.0	1.5±0.5	0.6±0.1	0.194	10	35	69
7			30.0±0.0	5.3±3.1	0.8±0.2	0.075	9	23	45
8		Mouse	60.1±0.0	0.6±0.2	0.4±0.1	0.190	11	33	129
9			160.1±4.8	1.6±1.4	0.5±0.2	0.292	23	23	71
10			158.3±0.0	1.5±1.4	0.5±0.2	0.184	25	25	75
11	GCaMP6s		151.6±15.0	1.0±0.4	0.8±0.1	0.381	6	6	13
12			157.5±3.3	1.3±0.7	0.5±0.2	0.390	26	26	62
13			60.1±0.0	0.4±0.3	0.5±0.2	0.745	7	18	69
14			59.1±0.0	2.7±0.9	0.7±0.2	0.115	9	9	77
15			59.1±0.0	5.8±3.1	0.9±0.2	0.062	9	9	25
16	GCaMP5k		50.0±0.0	1.6±0.8	0.5±0.2	0.175	9	9	29
17	R-CaMP1.07		20.0±0.0	2.2±0.7	1.6±0.3	0.109	4	67	33
18			14.4±1.6	0.9±0.9	0.6±0.2	0.093	9	100	49
19	jRCaMP1a		15.0±0.1	0.6±0.6	1.3±0.4	0.065	10	18	87
20	jRGECO1a		29.8±0.2	1.5±1.9	1.0±0.3	0.143	11	27	117

Table 2. Comparison of the proposed models and previous study

Model		# of trainable parameters	# of samples to compute in training	# of noise-levels applicable to	Prediction format
This work	U-Net	145,468	Up to 5000 iterations×1024 per iteration = 5.12M	All	Sequence of spike-rates
	Le-Net	148,160			
	FC-Net	148,288			
CASCADE		35,411×5 = 177,055	10 epochs×5M per epoch×5 = 250M	One	Individual timestep of spike-rate