Spike Sorting using the Neural Clustering Process

Anonymous Author(s)
Affiliation
email

Abstract

We present a novel approach to spike sorting for high-density multielectrode probes using the Neural Clustering Process (NCP), a recently introduced neural architecture that performs scalable amortized approximate Bayesian inference for efficient probabilistic clustering. To optimally encode spike waveforms for clustering, we extended NCP by adding a convolutional spike encoder, which is learned end-to-end with the NCP network. Trained purely on labeled synthetic spikes from a simple generative model, the NCP spike sorting model shows promising performance for clustering multi-channel spike waveforms. The model provides higher clustering quality than an alternative Bayesian algorithm, finds more spike templates with clear receptive fields on real data and recovers more ground truth neurons on hybrid test data compared to a recent spike sorting algorithm. Furthermore, NCP is able to handle the clustering uncertainty of ambiguous small spikes by GPU-parallelized posterior sampling. The source code is publicly available [1].

1 Introduction

Large-scale neuronal population recordings using high-density multi-electrode arrays (MEA) are at the forefront of current progress in our understanding of neural circuit dynamics. In MEA recordings, each electrode channel reads extracellular signals from many neurons, and each neuron is recorded by multiple nearby electrodes. A key step in the analysis of MEA data is spike sorting, which converts the raw electrical signal into a set of neural spike trains belonging to individual neurons. As MEAs grow in scale and popularity, there is a new urgency in improving spike sorting performance [2–6].

A typical spike sorting pipeline consists of three steps. The spike detection step extracts putative spike events from noisy recordings. The clustering step groups similar spike waveforms into clusters, each representing a putative neuron. To resolve colliding waveforms, a deconvolution step is often performed. Spike clustering is at the core of the pipeline, as the clustering performance determines both the accuracy of spike assignment and the quality of spike templates used for deconvolution.

Spike clustering, however, poses significant challenges: (1) Spike waveforms form highly non-Gaussian clusters in spatial and temporal dimensions, and it is unclear what are the optimal features for clustering. (2) It is unknown *a priori* how many clusters there are. (3) Although existing methods perform well on spikes with high signal-to-noise ratios (SNR), there remain significant challenges in the lower-SNR regime with increased clustering uncertainty. Fully-Bayesian approaches proposed to handle this uncertainty [7, 8] do not scale to large datasets due to expensive Gibbs sampling.

To address these challenges, we propose a novel approach using the recently introduced Neural Clustering Process (NCP) [9, 10]. NCP is based on a neural architecture that performs scalable amortized approximate Bayesian clustering. (1) Rather than selecting arbitrary features for clustering, the spike waveforms are encoded with a convolutional neural network (ConvNet), which is learned end-to-end jointly with the NCP network. This ensures optimal feature encoding. (2) Using a variable-input softmax function, NCP is able to perform inference on cluster labels without assuming a fixed or maximum number of clusters. (3) NCP allows for efficient probablistic clustering by GPU-

Submitted to Workshop on Real Neurons & Hidden Units: Future directions at the intersection of neuroscience and artificial intelligence, 33rd Conference on Neural Information Processing Systems (NeurIPS 2019).

parallelized posterior sampling, which is particularly useful for handling the clustering uncertainty of ambiguous small spikes. (4) The computational cost of NCP training can be highly amortized, since neuroscientists often sort spikes form many statistically similar datasets.

We trained NCP on synthetic spikes from a simple yet effective generative model, and evaluated the performance on labeled synthetic data, unlabeled real data, and hybrid test data. We show that NCP provides high clustering quality, matches or outperforms a recent spike sorting algorithm [2], and handles clustering uncertainty by producing multiple plausible clustering configurations. These results show substantial promise for incorporating NCP into a production-scale spike sorting pipeline.

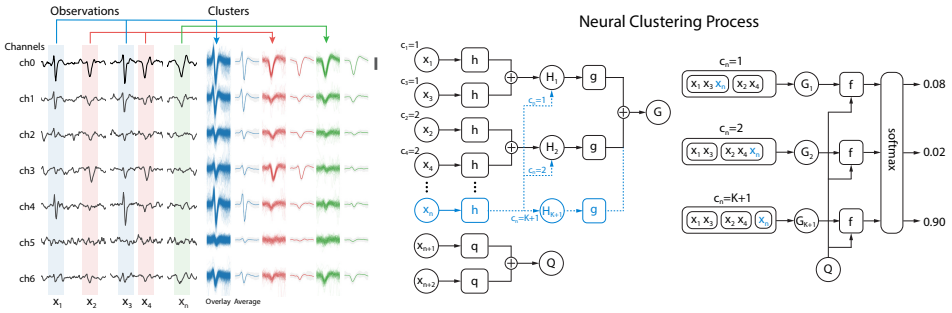

Figure 1: **Spike sorting with NCP.** *Left:* Multi-channel spike waveforms are grouped into clusters by NCP. Each row is an electrode channel. Spikes with the same color belong to the same cluster. (Scale bar: 5× standard deviation (SD)). *Right:* Diagram of the NCP architecture as in [10].

2 Spike Sorting using the Neural Clustering Process

Data preprocessing. Training and test data come from the 20 kHz retinal recordings in [11] using a 512-channel 2D hexagonal MEA. After spike detection [5], each spike is assigned to the channel where the waveform has the maximum peak-to-peak (PTP) amplitude (i.e. the center channel, ch0). We partitioned the recording data such that the data for each channel only contains spikes centered at that channel. For each spike, the extracted waveform is a 7×32 array containing the 32 time steps surrounding the peak from the center channel and its 6 immediate neighbor channels (Figure 1 left).

Neural architectures for NCP spike sorting. The NCP architecture contains four neural networks, h, q, g, f, as shown in Figure 1 (right). We refer to [10] for the detailed formulation and notations of NCP. To extract useful features from the spatial-temporal patterns of spike waveforms, we use a 1D ConvNet as the h and q encoder functions. The convolution is applied along the time axis, with each electrode channel treated as a feature dimension. The ConvNet uses a ResNet architecture with 4 residual blocks, each having 32, 64, 128, 256 feature maps (kernel size = 3, stride = [1, 2, 2, 2]). The last block is followed by an averaged pooling layer and a final linear layer. The outputs of the ResNet encoder are the h_i and q_i vectors of NCP, i.e. $h_i = \text{ResNetEncoder}(x_i)$. We used $d_h = d_q = 256$. The other two functions, g and f, are identical to those in the 2D Gaussian example in [10].

Training NCP using synthetic data. To train NCP for spike clustering, we created synthetic labeled training data using a mixture of finite mixtures (MFM) generative model of noisy spike waveforms that mimic the distribution of real spikes:

$$N \sim \text{Uniform}[N_{min}, N_{max}] \quad (1) \qquad c_1 \dots c_N \sim \text{Cat}(\pi_1, \dots, \pi_K) \quad (4)$$

$$K \sim 1 + \text{Poisson}(\lambda) \quad (2) \qquad \mu_k \sim p(\mu) \quad k = 1 \dots K \quad (5)$$

$$\pi_1 \dots \pi_K \sim \text{Dirichlet}(\alpha_1, \dots, \alpha_K) \quad (3) \qquad x_i \sim p(x_i | \mu_{c_i}, \Sigma_s \otimes \Sigma_t) \quad i = 1 \dots N \quad (6)$$

Here, N is the number of spikes between $[200, 500]$. The number of clusters K is sampled from a shifted Poisson distribution with $\lambda = 2$ so that each channel has on average 3 clusters. $\pi_{1:K}$ represents the proportion of each cluster and is sampled from a Dirichlet distribution with $\alpha_{1:K} = 1$. The training spike templates $\mu_k \in \mathbb{R}^{7 \times 32}$ are sampled from a reservoir of 957 ground-truth templates not present in any test data, with the temporal axis slightly jittered by random resampling. Finally, each waveform x_i is obtained by adding to μ_{c_i} Gaussian noise with covariance given by the Kronecker product of spatial and temporal correlation matrices estimated from the training data. This method creates spatially and temporally correlated noise patterns similar to real data (Figure 2 left). We trained NCP for 20000 iterations on a GPU with a batch size of 32 to optimize the NLL loss by the Adam optimizer. A learning rate of 0.0001 was used (reduced by half at 10k and 17k iterations).

Probabilistic spike clustering using NCP. At inference time, we fed the 7 x 32 arrays of spike waveforms to NCP, and performed GPU-parallelized posterior sampling of cluster labels (Figure 1). Using beam search with a beam size of 150, we were able to efficiently sample 150 high-likelihood clustering configurations for 2000 spikes in less than 10 seconds on a single GPU. After clustering, we obtained a spike template for each cluster as the average shape of the spike waveforms. The clustering configuration with the highest probability was used for most experiments in section 3.

3 Experimental Results

We compared NCP spike sorting against two other methods: (1) **vGMFM**, variational inference on a Gaussian Mixture of Finite Mixtures, using the first 5 PCA components of the multi-channel spike waveforms. (2) **Kilosort**, a state-of-the-art spike sorting pipeline described in [2].

Synthetic Data. We run NCP and vGMFM on 20 sets of synthetic test data each with 500, 1000, and 2000 spikes. As the ground-truth cluster labels are known, we compared the clustering quality using Adjusted Mutual Information (AMI). As shown in Figure 2 (right), The AMI of NCP is on average 11% higher than vGMFM, showing better performance of NCP on synthetic data.

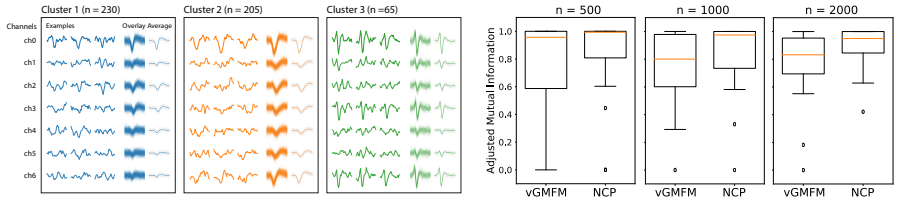

Figure 2: **Clustering on synthetic data.** *Left:* A synthetic data example containing 3 clusters and 500 spikes. *Right:* The AMI scores for 20 sets of 500, 1000, and 2000 unseen synthetic spikes.

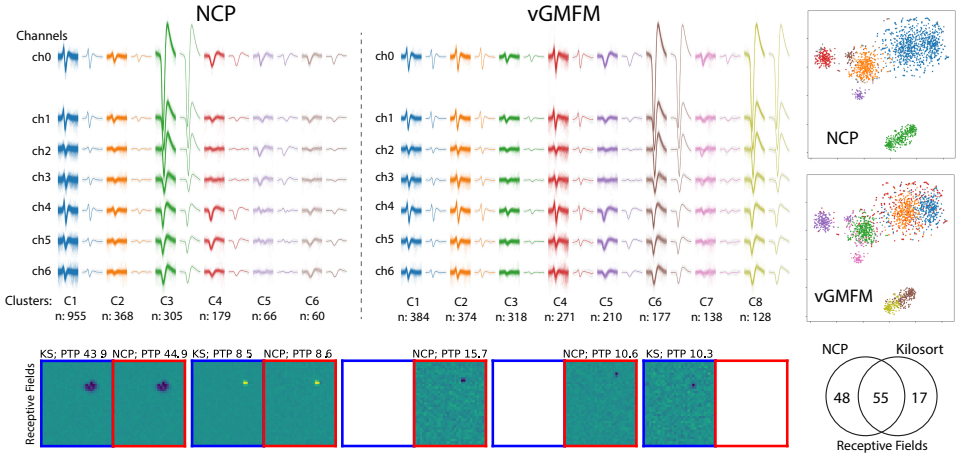

Figure 3: **Spike sorting on real data.** 2000 spikes from real data were clustered by NCP (*top-left*) and vGMFM (*top-mid*). Each column shows the spikes assigned to one cluster (overlaying traces and their average). Each row is one electrode channel. *Top-right:* t-SNE visualization of the spike clusters. *Bottom-left:* RFs of 5 example units recovered by NCP (red boxes) and Kilosort (blue boxes). *Bottom-right:* Venn diagram of RFs recovered by NCP and Kilosort.

Real Data. We run NCP, vGMFM and Kilosort on a 49-channel, 20-minute retina recording with white noise stimulus. For NCP and vGMFM, we performed clustering on 2000 randomly sampled spikes for each channel, and assigned all remaining spikes to a cluster based on the L2 distance to the cluster centers. Then, a final set of unique spike templates were computed, and each detected spike was assigned to one of the templates. For Kilosort, we run the entire pipeline using the Kilosort2 package. Figure 3 (top) shows example clustering results by NCP and vGMFM. NCP produces clean clusters with visually more distinct spike waveforms compared to vGMFM.

As real data do not come with ground-truth cluster labels, we compared the spike templates extracted from the NCP pipeline and Kilosort using retinal receptive field (RF), which is computed for each cluster (i.e. putative neuron) as the mean of the white-noise stimulus present at each spike. A clearly demarcated RF provides encouraging evidence that the spike template corresponds to a real

neuron. Pairs of templates were matched by L-infinity distance and pairs of RFs by cosine distance. Side-by-side comparisons of 5 example pairs are shown in Figure 3 (bottom-left). Overall, the NCP pipeline found 103 templates with clear RFs, among which 48 were not found in Kilosort. Kilosort found 72 and 17 of them were not found by NCP (Figure 3 bottom-right). This shows that NCP performs at least as well as Kilosort, and finds many additional spike templates with clear RFs.

Hybrid Data. We compared NCP against vGMFM and Kilosort on a hybrid recording with partial ground truth as in [2]. 20 ground-truth spike templates were manually selected from a 49-channel recording and inserted into another dataset. This approach tests the clustering performance on realistic recordings with complex background noise and colliding spikes. The methods were evaluated by the number of injected ground-truth templates recovered. As shown in Figure 4, NCP recovered 13 of the 20 templates, outperforming both Kilosort and vGMFM, which recovered 8 and 6, respectively.

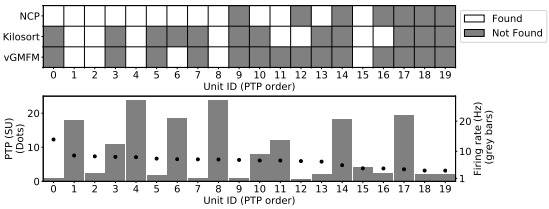

Figure 4: **Spike sorting on hybrid data.** 20 ground-truth units were inserted into real data. *Top:* NCP, Kilosort, vGMFM recovered 13, 8, and 6 units, respectively. *Bottom:* PTP size and firing rate of each injected unit. (Smaller units with lower firing rates are more challenging.)

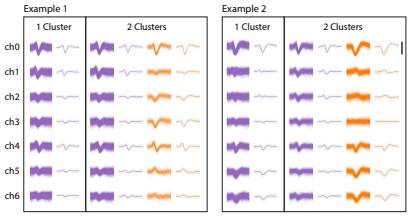

Figure 5: **Clustering ambiguous small spikes.** In both examples, multiple plausible clustering results of small spikes were produced by sampling from the NCP posterior. (scale bar = 5× SD)

Probabilistic clustering of ambiguous small spikes. Spike sorting of small-amplitude waveforms has been challenging due to the low SNR and increased uncertainty of cluster assignment. Traditional methods could not handle the uncertainty and fully-Bayesian approaches do not scale. By efficient GPU-parallelized sampling of cluster labels from the posterior, NCP is able to handle the clustering uncertainty for ambiguous small spikes by producing multiple plausible clustering results. Figure 5 shows examples where NCP separates spike clusters with amplitude as low as 3-4× SD into plausible units that are not mere scaled version of each other but have distinct shapes on different channels.

Conclusions. Our results show that NCP spike sorting matches or outperforms current state-of-the-art methods, and is able to handle clustering uncertainty by efficient posterior sampling. Future directions include more realistic generative models, better spike encoders that utilize information from distant channels, and integrating NCP into a standard spike sorting pipeline.

References

[1] Anonymous Authors. ncp-sort. `https://bit.ly/2lJtPJT`, 2019.

[2] Marius Pachitariu, et al. Kilosort: realtime spike-sorting for extracellular electrophysiology with hundreds of channels. *bioRxiv*, 2016.

[3] Jason E Chung, et al. A fully automated approach to spike sorting. *Neuron*, 2017.

[4] James J. Jun, et al. Real-time spike sorting platform for high-density extracellular probes with ground-truth validation and drift correction. *bioRxiv*, 2017.

[5] Jin Hyung Lee, et al. Yass: Yet another spike sorter. *NIPS*, 2017.

[6] David Carlson and Lawrence Carin. Continuing progress of spike sorting in the era of big data. *Current opinion in neurobiology*, 2019.

[7] Frank Wood and Michael J Black. A nonparametric bayesian alternative to spike sorting. *Journal of neuroscience methods*, 2008.

[8] David E Carlson, et al. Multichannel electrophysiological spike sorting via joint dictionary learning and mixture modeling. *IEEE Transactions on Biomedical Engineering*, 2013.

[9] Ari Pakman and Liam Paninski. Amortized bayesian inference for clustering models. *arXiv:1811.09747, BNP@NeurIPS 2018 Workshop*, 2018.

[10] Ari Pakman, et al. Discrete neural processes. *arXiv:1901.00409*, 2019.

[11] E. J. Chichilnisky and Rachel S. Kalmar. Functional asymmetries in on and off ganglion cells of primate retina. *Journal of Neuroscience*, 2002.