Spike Sorting using the Neural Clustering Process

Yueqi Wang, Ari Pakman, Catalin Mitelut, JinHyung Lee, Liam Paninski

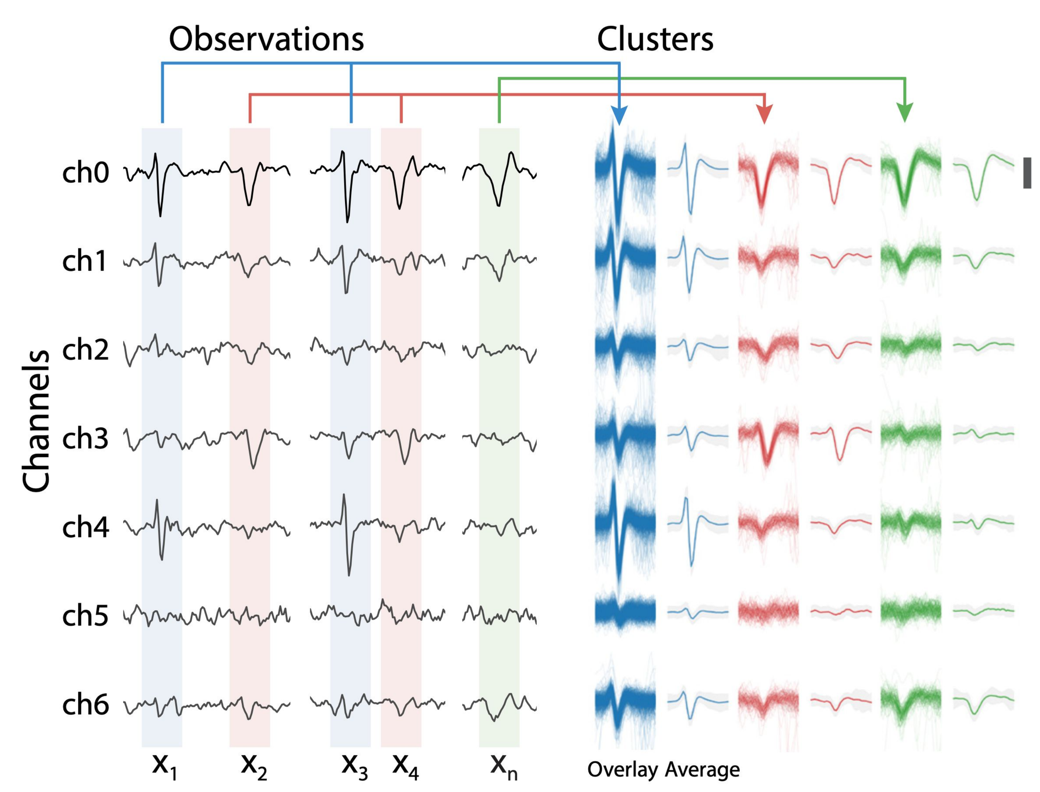
Columbia University

Paper Code

Summary

We present a novel approach to spike sorting for high-density multielectrode probes using the Neural Clustering Process (NCP), a neural architecture that performs scalable amortized approximate Bayesian inference for efficient probabilistic clustering.

Spike Sorting



Goal: Group similar multi-channel spike waveforms into clusters, each representing a putative neuron.

Challenges:

1. It is unclear what are the optimal features for clustering.
2. The number of clusters is unknown *a priori*.
3. Uncertainty of cluster assignments for low-SNR ambiguous spikes.

Neural Clustering Process (NCP)[1]

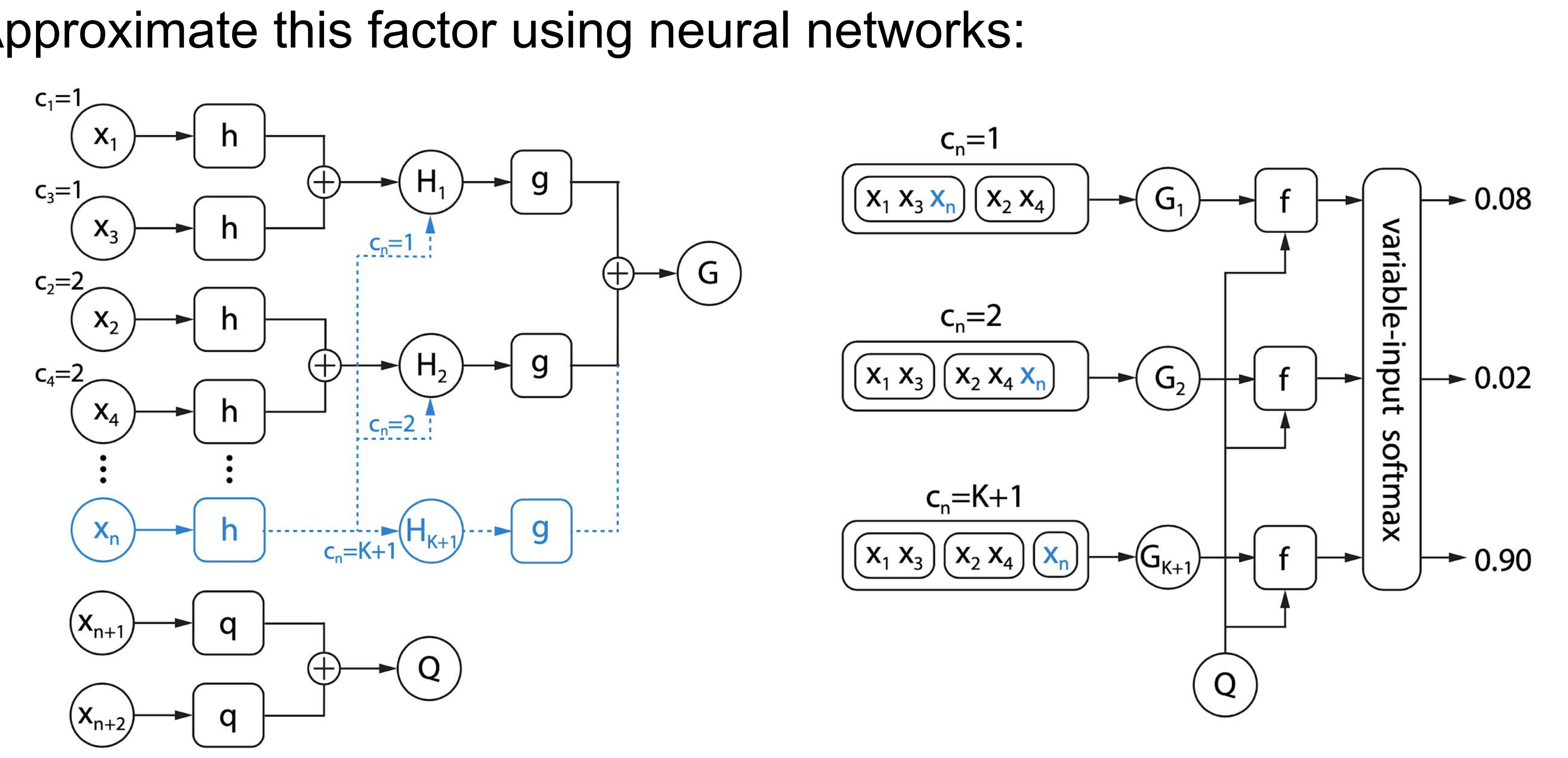
A novel architecture for efficient approximation to full Bayesian nonparametric inference over high-dimensional mixture models.

Given N data points $\mathbf{x} = \{x_i\}$, we would like to sample the cluster labels $c_{1:N}$ from the posterior:

$$p(c_{1:N}|\mathbf{x}) = p(c_1|\mathbf{x})p(c_2|c_1,\mathbf{x})\dots p(c_N|c_{1:N-1},\mathbf{x})$$

Each factor: $$p(c_n|c_{1:n-1},\mathbf{x}) = \frac{p(c_1\dots c_n,\mathbf{x})}{\sum_{c'_n=1}^{K+1} p(c_1\dots c'_n,\mathbf{x})}$$

Approximate this factor using neural networks:



Permutation-invariant representations:

$$H_k = \sum_{i:c_i=k} h(x_i) \quad k=1\dots K \qquad G = \sum_{k=1}^{K} g(H_k) \qquad Q = \sum_{i=n+1}^{N} q(x_i)$$

Variable-input softmax function:

$$p_\theta(c_n = k|c_{1:n-1},\mathbf{x}) = \frac{e^{f(G_k,Q)}}{\sum_{k'=1}^{K+1} e^{f(G_{k'},Q)}} \qquad k=1\dots K+1$$

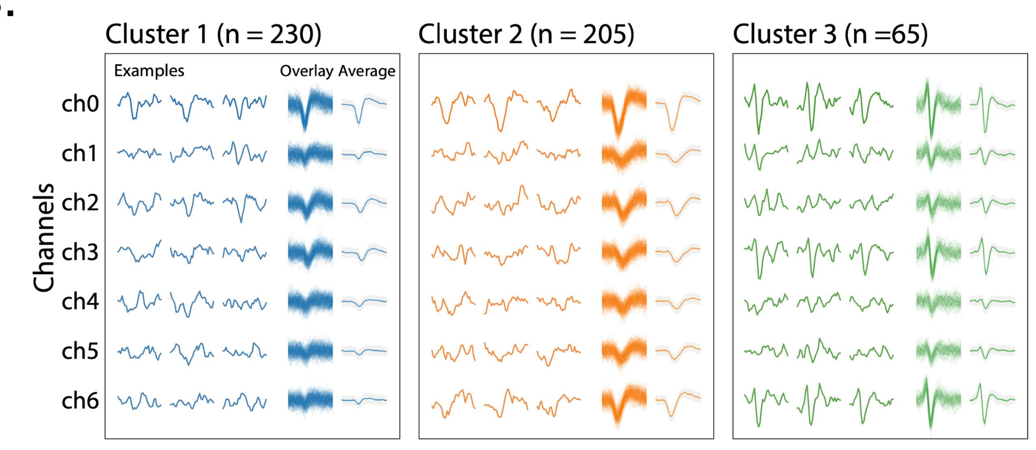
Spike Sorting using NCP

- The spike waveforms are encoded with a convolutional neural network learned end-to-end jointly with the NCP network.
- NCP computes the full posterior on cluster labels and the number of clusters, without assuming a fixed or maximum number of clusters.
- NCP handles the clustering uncertainty by efficient probabilistic clustering implemented as GPU-parallelized posterior sampling.
- The computational cost of NCP training can be highly amortized for statistically similar datasets.

Training NCP using synthetic data

We created synthetic labeled training data using a mixture of finite mixtures (MFM) generative model of noisy spike waveforms that mimics the distribution of real spikes.

$$\begin{aligned} N &\sim \text{Uniform}[N_{min}, N_{max}] \\ K &\sim 1 + \text{Poisson}(\lambda) \\ \pi_1\dots\pi_K &\sim \text{Dirichlet}(\alpha_1,\dots,\alpha_K) \\ c_1\dots c_N &\sim \text{Cat}(\pi_1,\dots,\pi_K) \\ \mu_k &\sim p(\mu) \quad k=1\dots K \\ x_i &\sim p(x_i|\mu_{c_i}, \Sigma_s \otimes \Sigma_t) \end{aligned}$$



A related generative model was introduced in [2] to train a spike detector network.

Spike Sorting using NCP

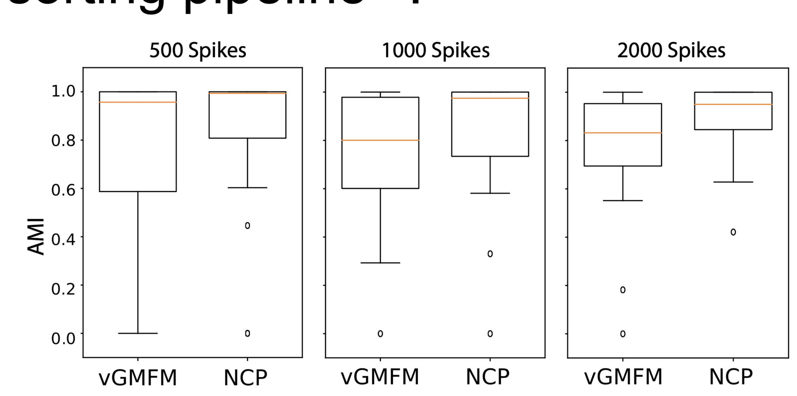
At inference time, we perform GPU-parallelized posterior sampling of cluster labels to find high-likelihood clustering configurations.

NCP spike sorting vs. two other methods:

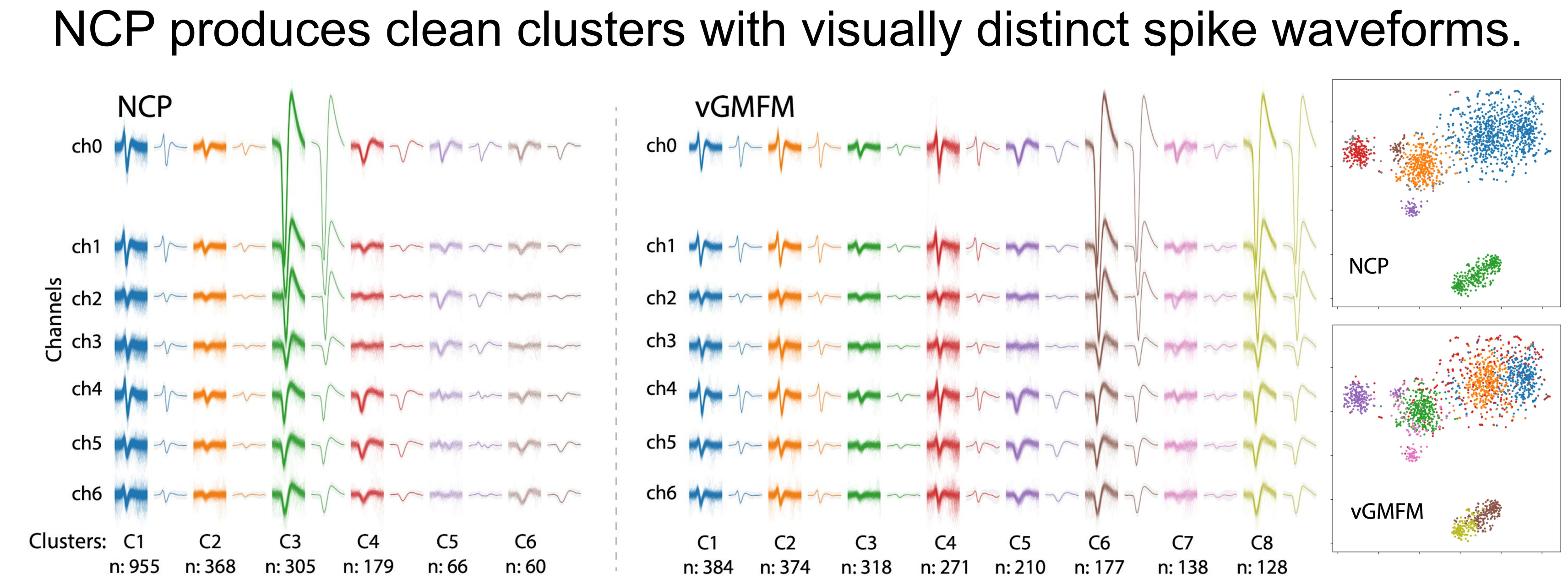
vGMFM: variational inference on a Gaussian MFM.

Kilosort2: a state-of-the-art spike sorting pipeline[3].

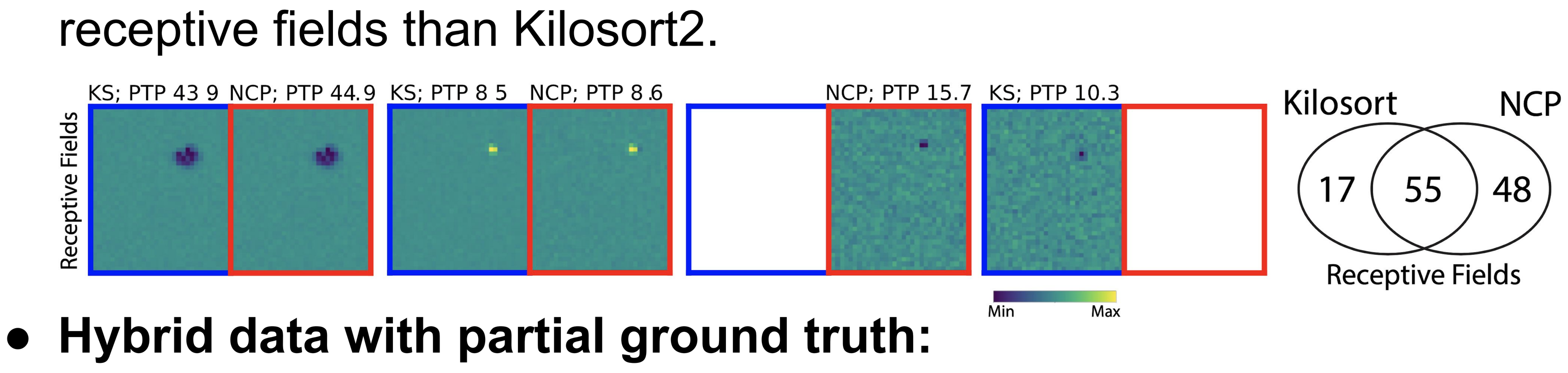
- **Labeled synthetic data:** Higher clustering quality than vGMFM measured by adjusted mutual information (AMI).



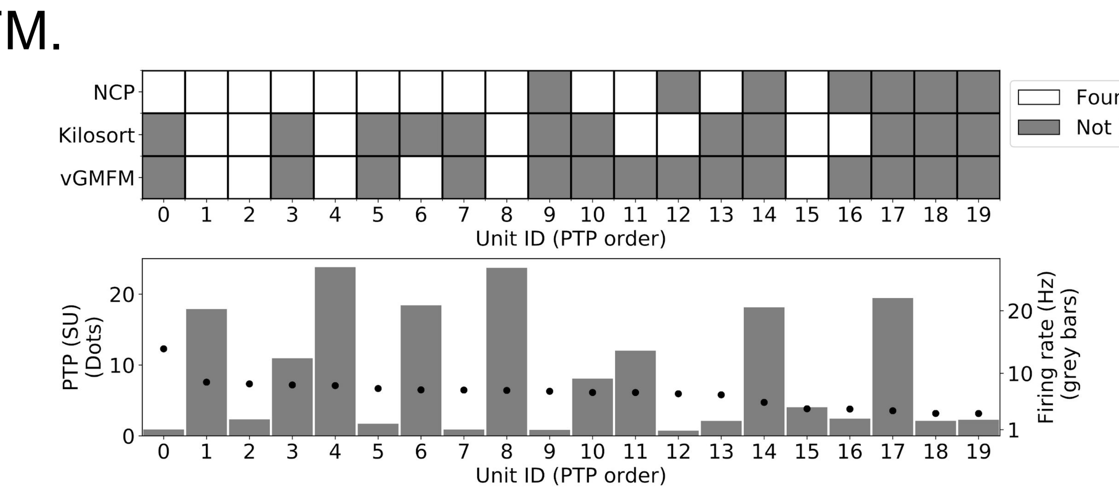
- **Unlabeled real data:** (49-channel, 20-min retina recording) NCP produces clean clusters with visually distinct spike waveforms.



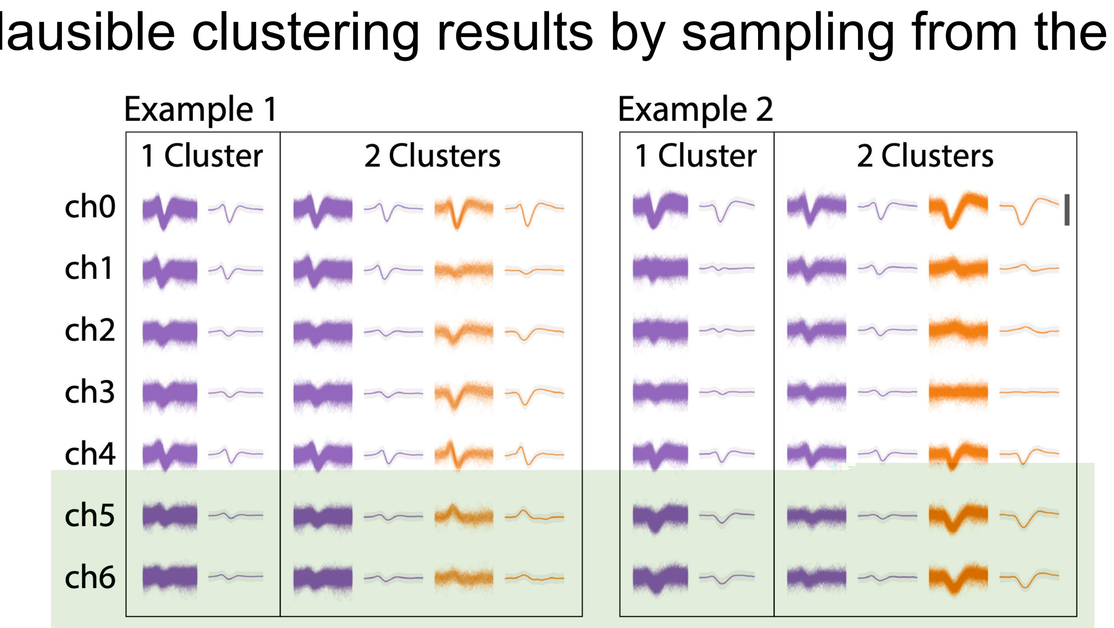
NCP finds more spike templates (putative neurons) with clear receptive fields than Kilosort2.



- **Hybrid data with partial ground truth:** NCP recovers more injected ground-truth templates than Kilosort2 and vGMFM.



- **Probabilistic clustering of ambiguous small spikes:** Multiple plausible clustering results by sampling from the posterior.



References

[1] A Pakman, et al. (2019). Discrete neural processes, arXiv:1901.00409

[2] JH Lee, et al. (2017). Yass: Yet another spike sorter, NIPS

[3] M Pachitariu, et al. (2016). Fast and accurate spike sorting of high-channel count probes with KiloSort, NIPS