Deep Variational Auto Encoder for Spike Sorting

Ori Noked, 203055744, orinoked@gmail.com

Daniel Cohen, 302787833, danielcohen629@gmail.com

Abstract

Objective: Introduce a new method for spike sorting. Spike sorting is key process in almost every electrophysiological research, it is the classification algorithm that determines which spike in extracellular recordings corresponds to which neuron. **Approach:** We developed an innovative spike sorting feature extractor variational autoencoder model. The encoder network is equipped with residual transformations to extract representative features from spikes. The latent space is used as a feature space for a Gaussian Mixture Model that separates the spikes to different clusters. The clustering accuracy and performance of the proposed feature space is compared to the traditionally used Principal Component space. **Main results**: Experimental results on in-vivo dataset show that the proposed approach consistently outperforms the conventional Principal Component approach. With the latent space compared to the Principal Component space, the Gaussian Mixture Model accuracy tested against ground truth was higher by 15%. Two common clustering quality indices, the Davies-Boulding index and the Calinski-Harabasz index, also showed that clustering at the latent space was superior to the clustering at the Principal Component space. **Significance:** Variational autoencoder as a feature extractor holds great potential for spike sorting. The superiority of this approach may lead to an improvement in accuracy of spike sorting and to a major reduction in the number of manual clustering corrections implemented by neuroscientists today.

https://github.com/orinoked1/SpikeSortingVAE

1 Introduction

1.1 Spike sorting

Neuronal activity in the brain gives rise to transmembrane currents that can be measured in the extracellular medium. Electric current contributions from all active cellular processes within a volume of brain tissue superimpose at a given location in the extracellular medium and generate an electrical potential with respect to a reference electrode. Electric fields can be monitored by extracellularly placed electrodes with sub millisecond time resolution and can be used to interpret many facets of neuronal communication and computation. A key component in cell to cell communication is the action potential (or spike) – a rapid depolarization and repolarization of a neuronal cell's membrane potential. To understand the functionality of a neuronal network one must be able extract timing of action potentials from an extracellular electrode.

Spike sorting is the classification of neuronal action potential into distinct (non-overlapping) clusters based on the difference between their shapes, (fig. 1). In principle, the spikes of different neurons are projected differently on a single extracellular electrode placed in the brain due to different distances, intermediate media, and proximity of the electrode to different neuronal compartments (soma, axon initial segment, axon hillock, axon, etc.). Moreover, some neurons tend to fire spikes of a shape than others. Thus, the different projections on a given recording electrode

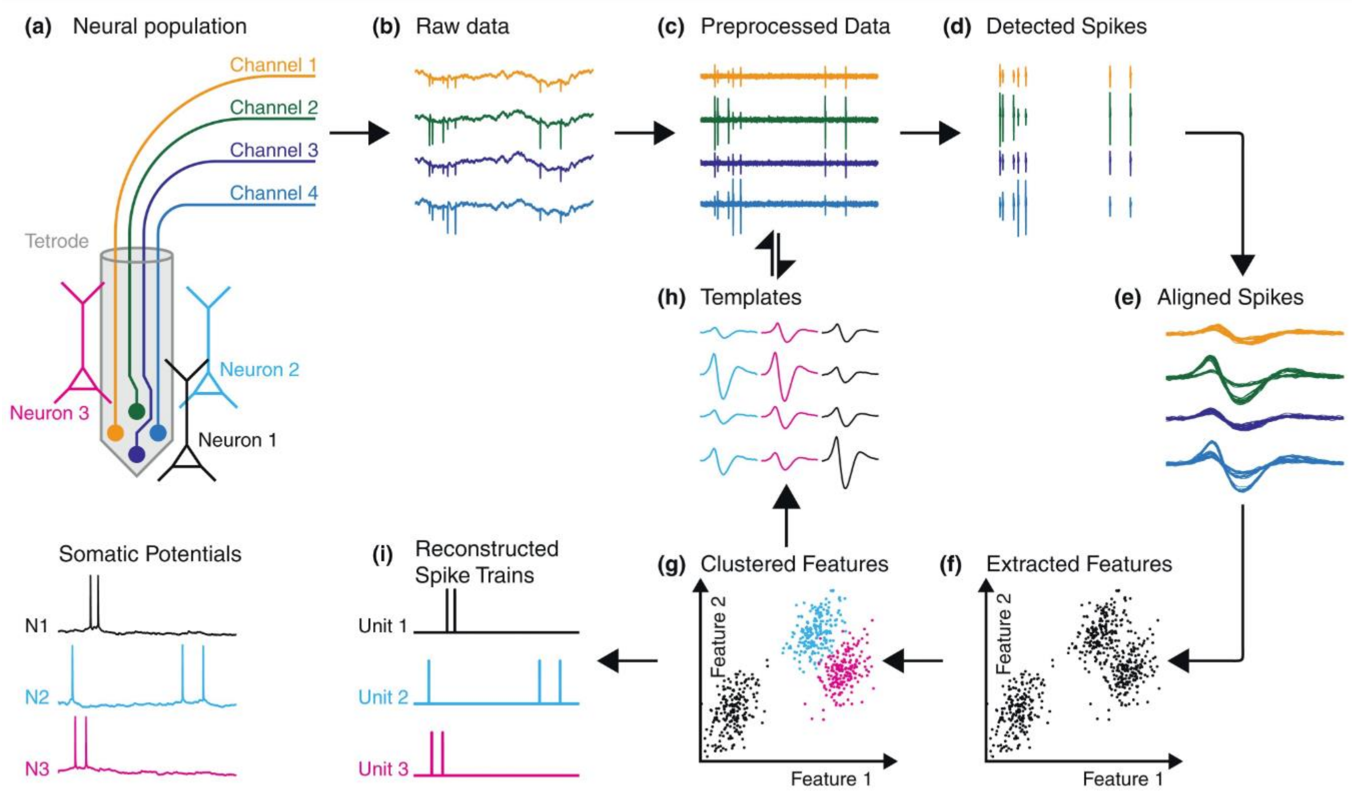

Figure 1 - Overview of spike-sorting process adopted from[1]. Electrodes record changes in the extracellular electrical potential (b) caused by action potentials of neurons in its vicinity (a) [intracellular potentials of three spiking example neurons (N1, N2, N3) are depicted in lower left panel]. For the detection and analysis a highpass or bandpass filter is usually used to remove the low-frequency part of the potential (c). The optimal procedure for detection of spikes, especially in multielectrode recordings, is still is an unsolved problem but simple voltage thresholding is commonly used (d). For most spike-sorting procedures the extracted spike waveforms need to be temporally aligned on a common feature like the position of the voltage peak (e) before features are extracted from every waveform (f). The feature extraction is crucial for decreasing the dimensionality of the data to the most informative dimensions. This can be done by, for example, using principal component analysis or wavelets. Clustering, that is, finding the number of clusters and their position in the feature space (g), is highly susceptible to the choice of those features. Individual clusters should contain all spikes of one putative neuron only and are commonly assumed to exhibit multivariate normal or t-distributions. The average waveform of all spikes belonging to one cluster is called the 'template' of that neuron (h).

("waveforms") may be sorted into different clusters. The resulting clusters correspond to typical waveforms of different putative neurons, and the individual spikes in each cluster thus correspond to the activity of those neurons. The result of spike sorting is the determination of which spike corresponds to which of these neurons [1].

In the past, extra cellular recordings of LFP (local field potential) were performed with a single channel electrode [2]. Triangulating a point in the three-dimensional space requires a set of four different reference points which are not located on a single plane. This fact makes it impossible to accurately pin-point the location of a firing neuron. Moreover, the use of a single electrode does not allow one to tell apart between two firing neurons located at the exact same distance from the recording electrode. Advancements in the field of electronics allow the use of multi-channel electrodes (initially, tetrodes, more recently, silicon probes) which could solve the issues arising from the use of a single channel electrode [3].

Several algorithms have been suggested to analyse the data recorded by a multi-electrode array. In the past, much work has been done using tetrodes[4], [5]. Today, the use of silicon probes has become more common. Recently, advanced methods have been suggested to cope with the issue of temporally overlaying spikes[6], [7].

1.2 autoencoders and variational autoencoders (VAE)

An autoencoder (vanilla as well as VAE) is a pair of two connected networks, an encoder and a decoder. An encoder network takes in an input, and converts it into a compact, dense representation, which the decoder network can use to convert it back to the original input. The entire network is usually trained as a whole. The loss function is usually either the mean-squared error or cross-entropy between the output and the input, known as the reconstruction loss, which penalizes the network for creating outputs different from the input.

As the encoding has far less units than the input, the encoder learns to preserve as much of the relevant information as possible in the limited encoding and discard irrelevant parts. The decoder learns to take the encoding and properly reconstruct it into a full input. Together, they form an autoencoder.

The fundamental problem with vanilla autoencoders for clustering problems, is that the latent space where their encoded vectors lie may not be continuous or allow easy interpolation. This may lead to poor clustering performances. In VAE the latent space is, *by design*, continuous, allowing easier clustering [8].

This is achieved by making the encoder output two vectors: a vector of means, and a vector of standard deviations. Encodings are generated at random from a normal distribution with the resulted means, and standard deviations. sampling from a distribution makes the decoder learn that not only is a single point in latent space referring to a sample of that class, but all nearby points refer to the same as well. In this method the decoder has to not just decode single, specific encodings in the latent space (leaving the decodable latent space discontinuous), but ones that slightly vary too, as the decoder is exposed to a range of variations of the encoding of the same input during training.

In order to force normal distribution on the latent space, VAEs also incorporate the Kullback–Leibler divergence (KL divergence) into the loss function. The KL divergence between two probability distributions measures how much they diverge from each other. Minimizing the KL divergence is equivalent to optimizing the probability distribution parameters (mean and standard deviation) to closely resemble that of the normal distribution.

2 Related work

Recently a deep autoencoder for data compression in large scale neural recordings was introduced [9]. The proposed method allows efficient transmission of data between a recording chip and a remote computer with compression ratio of 20-500X.
The compression model is built upon a deep compressive autoencoder (CAE) with discrete latent embeddings. The encoder network of CAE is equipped with residual transformations to extract representative features from spikes, which are mapped into the latent embedding space and updated via vector quantization (VQ). The indexes of VQ codebook are further entropy coded as the compressed signals. The decoder network reconstructs spike waveforms with high quality from the quantized latent embeddings through stacked deconvolution.

The proposed model consistently outperforms conventional methods that utilize hand-crafted features and/or signal-agnostic transformations and compressive sensing by achieving higher compression ratios with better or comparable reconstruction accuracies.

In this work, we propose an innovative variational autoencoder (VAE) for classification and spike sorting. The encoder and decoder architecture will be based on the compression network by Tong Wu et.al., while the latent embedding space will be replaced with the VAE scheme. We will show that clustering at a VAE latent space is superior to clustering at a PCA space (current standard).

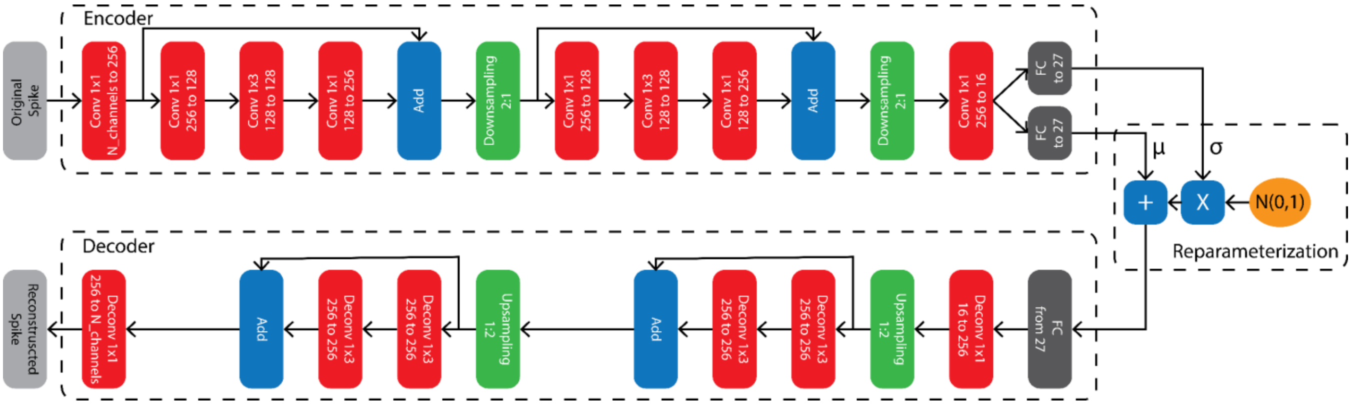

Figure 2 - feature extraction VAE architecture.

3 Methods

3.1 Data

The dataset used in this project is a 9 channel in-vivo data recorded from a single mouse. Spikes were detected from the raw signal by applying a threshold of 5 standard deviations from the mean. The spikes were then sorted, i.e. separated to distinct putative neurons. Sorting was performed by an expert electrophysiologist: first the spikes were transferred to the PCA space by taking 3 PCs from each recording channel (27-dimensional space), then an unsupervised gaussian mixture model (GMM) was used for finding preliminary clusters. The preliminary clusters were then reviewed by an expert and appropriate clusters were merged and\or split resulting the final labels. We refer to these labels as ground truth.

Each spike has 32 samples in time domain across 9 channels. The datasets used for training and testing were recorded using a different silicone probe located in the same neuronal tissue. Datasets size and number of neurons (ground truth clusters) is detailed in table 1.

Table 1 - Dataset details

	Training	Validation	Testing
Spikes [k]	1,500	100	200
# neurons	25	14	14

3.2 VAE for neural data classification

The diagram of the proposed VAE is presented in fig. 2. The layers layout is based on the compression CAE in previous work [9]. The encoder includes the input convolutional layer with kernel size 1*1 which maps input detected spikes organized in $\mathrm{M_{spk}}$ channels (9 for our data) to a 256-channel feature space. Then, a ResNeXt block [10] is implemented to enhance the feature extraction capability. The main pathway of the ResNeXt consists of a stack of 3 layers with 1*1, 1*3, and 1*1 convolutional kernels. The first 1*1 layer is responsible for reducing dimensions, the 1*3 layer extracts features with halved input/output dimensions and the final 1*1 layer is responsible for restoring dimensions.

The main feature of a ResNeXt refers to the grouped convolutions implemented in the encoder instead of vanilla convolutions. In group convolution, where groups number equals g, the operation becomes equivalent to having g conv layers side by side, each seeing 1/g the number of input channels, and producing 1/g the output channels, and all are subsequently concatenated. Following the ResNeXt block, down sampling in the time domain by a factor of 2 was implemented by max pooling. Than another ResNeXt block and a second down sampling is implemented. Eventually, a vanilla 1*1 convolutional layer that aggregates the features learned from previous layers and reduces the channel dimension from 256 16.

The encoder output is then transformed to two 27 dimensional vectors by two fully connected layers. The two vectors represent mean and standard deviations of distributions. Then, a sampling from the encoded distribution takes place. A 27 dimensions latent space was chosen in order to compare clustering at a space with the same number of dimensions as the current standard (3 PCs for each channel).

The decoder received the sampled latent vector as is its input. The decoder's structure is a reverse implementation of the encoder such that deconvolutions and up sampling are used instead of convolutions and down sampling respectively.

Unlike the encoder where group convolutions were used (ResNeXt blocks), in the decoder vanilla deconvolutions (ResNet blocks) are used. Each ResNet block includes two 1*3 deconvolutional layers that preserve the dimensions.

The main pathway of the decoder is: a fully connected layer from the latent space to the 16 channels representation, followed by a 1*1 deconvolution layer from 16 channels to 256. Then two blocks of up-sampling in time and a ResNet block. Finally, another 1*1 deconvolution layer from 256 channels to the original number of channels (in our data 9).

For reconstruction loss we used L2 norm and in order to impose a normal distribution we used the KL Divergence. The total loss was the sum of the two.

3.3 Methods for performance evaluation

Our main objective was to compare clustering at the VAE's latent space to the current standard i.e. clustering at a space comprised of 3 PCs from each recording channel (both are 27 dimensional spaces). For comparison we used 4 different metrics:

(1) Accuracy score of a gaussian mixture model (GMM): we fitted the test data with a GMM model with the 16 clusters (the ground truth number of clusters, table 1), we used 10 different initiations and chosen the best model found. We then matched between the ground truth clusters to the GMM clusters (by the highest union rates) and calculated the average accuracy for each cluster and the average accuracy for the entire test data. We repeated the process for the latent space as well as for the PCA space.

(2) Pairwise GMM accuracy: for each possible ground truth clusters pair we fitted the test data with a GMM model (with 2 clusters). We used 10 different initiations and chosen the best model found. We then matched between the ground truth clusters to the GMM clusters and calculated the number of correctly classified spikes for each pair at the latent space and at the PCA space.

(3) Common clustering quality indices: To test the clustering potential of each space, two common indices, The Davies-Boulding (DB) index and Calinski-Harabasz (CH) index were used. Both indices quantify the similarity within a cluster and separation between clusters. The definitions for similarity and separation differ between the two indices.

The DB index [10] signifies the average similarity between clusters, where the similarity is a measure that compares the distance between clusters with the size of the clusters themselves. A lower DB index relates to a model with better separation between the clusters. The index is defined as the average similarity between each cluster $\boldsymbol{C_i}$ for $\boldsymbol{i = 1, \ldots, k}$ and its most similar one $\boldsymbol{C_j}$. Similarity is defined as a measure $\boldsymbol{R_{ij}}$ that is composed of the following distances: (1) $\boldsymbol{s_i}$ - the average distance between each point of cluster $\boldsymbol{i}$ and the centroid of that cluster – also know as cluster diameter. (2) $\mathbf{d_{ij}}$ - the distance between cluster centroids $\mathbf{i}$ and $\mathbf{j}$. $\mathbf{R_{ij}}$ is defined as $\mathbf{R_{ij}} = \frac{\mathbf{s_i + s_j}}{\mathbf{d_{ij}}}$. Finally, the DB index is defined as $\mathbf{DB} = \frac{\mathbf{1}}{\mathbf{k}} \sum_{\mathbf{i=1}}^{\mathbf{k}} \max_{i \neq j} \boldsymbol{R_{ij}}$.

The CH index [11] is the ratio of the sum of between-clusters dispersion and of inter-cluster dispersion for all clusters. Dispersion is defined as the sum of distances squared. When clusters are dense and well separated, the CH index score is higher. For a set of data $\boldsymbol{E}$ of size $\boldsymbol{n_E}$ which is clustered to $\boldsymbol{k}$ clusters, the CH score s is defined as the ratio of the between-clusters dispersion mean and the within-cluster dispersion: $\boldsymbol{s} = \frac{\boldsymbol{tr(B_k)}}{\boldsymbol{tr(W_k)}} \times \frac{\boldsymbol{n_E - k}}{\boldsymbol{k-1}}$ where $\boldsymbol{tr(B_k)}$ is trace of the between group dispersion matrix and $\boldsymbol{tr(W_k)}$ is the trace of the within-cluster dispersion matrix defined by: $\boldsymbol{W_K} = \sum_{\mathbf{q=1}}^{\mathbf{k}} \sum_{\mathbf{x \in C_q}} \boldsymbol{(x - c_q)(x - c_q)^T}$ and $\boldsymbol{B_K} = \sum_{\mathbf{q=1}}^{\mathbf{k}} \boldsymbol{n_q (c_q - c_E)(c_q - c_E)^T}$ with $\boldsymbol{C_q}$ the set of points in cluster q, $\boldsymbol{c_q}$ the center of cluster $\boldsymbol{q}$, $\boldsymbol{c_E}$ the center of $\boldsymbol{E}$, and $\boldsymbol{n_q}$ the number of points in cluster $\boldsymbol{q}$.

3.4 Model training environment and Parameter configuration

The design and testing environment is AMD Ryzen 2700@3.20GHz, NVIDIA GeForce 1060 6GB, 16GB memory, 256GB SSD, and Windows 10.

The proposed VAE model is implemented using the deep learning framework PyTorch 1.4.0. We used the ADAM optimizer and evaluated the model performance after 15 epochs with batch-size 128 in all the experiments.

Several different models were trained with different parameters. In all models, the number of channels, $\boldsymbol{M_{spk}} = \mathbf{9}$ was determined by the dataset.

The following combinations of different parameters were examined. Learning rate was tested with values of $\mathbf{10^{-3}}$ and $\mathbf{10^{-4}}$. Weight decay tested with values $\mathbf{10^{-4}}$ and $\mathbf{10^{-5}}$. Drop rates that were tested were 0.2 and 0.5. Additionally, we tested performances with a channel shuffling augmentation (i.e. in training shuffle the different electrode channels) to try improving generalization ability.

4 Results

Eventually, the model that provided most accurate results in terms of accuracy and best clustering quality indices values learning rate of $\mathbf{10^{-3}}$, weight decay of $\mathbf{10^{-5}}$, drop rate of 0.2 and no shuffling augmentations.

4.1 Accuracy score of a GMM

Table 2 shows the GMM overall accuracy of the two techniques against the ground truth labels (16 different classes). A 15% improvement in GMM accuracy is achieved. It should be noted that in current standard method GMM in the PC space produces only the preliminary clusters (that are manually merged and\or split by the researcher) but an improvement in the preliminary clustering will be translates to less manual corrections.

Table 2 – Overall Accuracy

	PC space	Latent space
GMM accuracy [%]	51%	66%

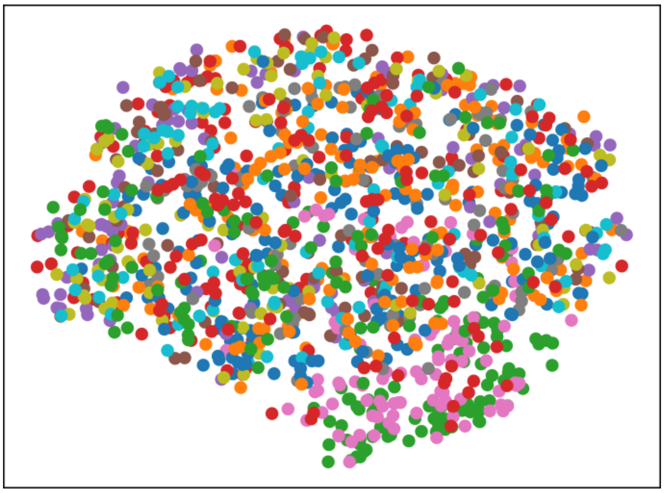

Figure 3 - 2-dimentinal t-SNE plot of the PC space. Each of the 14 colors represents a different ground truth cluster. For each cluster 100 random samples are displayed. No cluster forming can be seen.

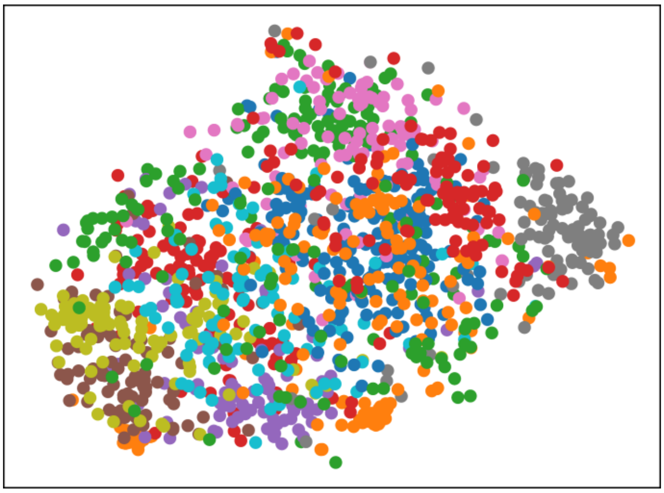

Figure 4 - 2-dimentinal t-SNE plot of the latent space. Each of the 14 colors represents a different ground truth cluster. For each cluster 100 random samples are displayed. Some separation between clusters can be seen.

In order to visualize the features distribution of the PC space and latent space, a 2-dimentinal t-SNE plots of both 27-dimensional feature space are shown in fig. 3 and 4 respectively. It is clearly shown that in 2-dimensional projection in latent space the clusters are much more separable.

4.2 pairwise GMM accuracy

Table 3 shows the GMM pairwise accuracy:

Table 3 – Pairwise Accuracy

	PC space	Latent space
GMM pairwise accuracy [%]	78%	84%
More separable pairs [#]	25	42

With the latent space features, a 6% improvement in GMM pairwise accuracy is achieved compared to the PC features.

Accordingly, the number of cluster pairs with better separation, i.e higher GMM pairwise accuracy, was 42 with the latent space compared to 25 cases with the PC space. That is approximately 2/3 of cluster pairs were better separated with the latent space features.

We also produced a t-SNE visualization for each pair of ground truth clusters. An example sample can be found in fig.5 (full comparison can be found in supplementary material figures s.1, s.2).

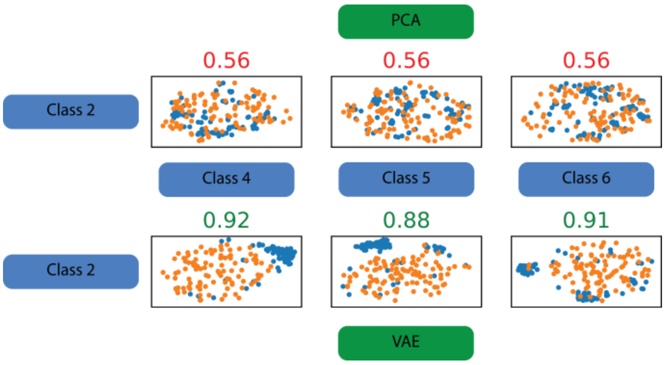

Figure 5 – t-SNE plot of 27-dimensional PC feature space for pairwise clustering for cluster pairs 2-4, 2-5, 2-6.the top row are the t-SNE plots for the PC space and the bottom row is for the latent space. Above each figure is the accuracy of a GMM classifier on that pair. In each cluster 100 random samples are presented.

4.3 Clustering Indices Results

Table 4 shows the values for the CH index and DB index that evaluate the clusters of both techniques in the 27-dimensional spaces:

Table 4 – Clustering Indices

	PC space	Latent space
CH - Index	970	5767
DB – Index	10.73	6.37

Both indices show superiority of the clustering implemented on the VAE latent space when compared to the PC space.

5 Discussion

We have shown that using a VAE a feature extractor holds great potential for spike sorting. While the final GMM accuracy achieved is still unacceptable (66%), we believe that the 15% accuracy improvement over traditionally used PCA features will lead to a major reduction in the number of manual clustering corrections by the researcher. Further evidence for better data representation at the VAE latent space can be found in the performances of the GMM classifier on each clusters pair as well as in the improvement in clustering indices (CH – Index and DB – Index).

A key factor in clustering performances is the number of dimensions in the feature space. In this work, in order to separate variables when comparing to the PC space, we have chosen to use a latent space of the same number of dimensions (27). We believe that even better clustering performance is achievable by finding the optimal latent space dimension.

A major factor that was not tested within the scope of this work is the number of clusters in the data. During all clustering experiments we used the known number of ground truth clusters in the data but in a real-world spike sorting problem the number of clusters is unknown. In practice the experimenter usually overestimates the number of clusters for the clustering algorithm, then by manual merging of clusters he attains an acceptable result. We believe that similar technique will work even better in the VAE latent space given the evidence for better initial separation we have shown.

Another factor that was not tested in this work is the ability of the VAE to generalize: first to different brain areas with different neuronal cell types, that may have different action potential waveform. Further testing is required in order to understand the VAE ability to generalize unseen waveforms.

A second generalization ability is to process data from recording probes with different geometry. Different recording probe geometry may lead to different relationship between recording channels and its effect on the VAE performance must be tested as well. Additionally, due to fully connected layer in our architecture the network is currently limited to a fixed number of recording channels. In the future we should consider replacing the fully connected layer with a convolution layer making the network agonistic to the number of channels.

6 Acknowledgment

All the data used throughout this work was supplied Eran Starks's lab at Sakler Faculty of Medicine, Tel Aviv University.

7 **References**

[1] G. T. Einevoll, F. Franke, E. Hagen, C. Pouzat, and K. D. Harris, "Towards reliable spike-train recordings from thousands of neurons with multielectrodes," *Curr. Opin. Neurobiol.*, vol. 22, no. 1, pp. 11–17, 2012.

[2] M. Abeles and M. H. Goldstein, "Multispike train analysis," *Proc. IEEE*, vol. 65, no. 5, pp. 762–773, 1977.

[3] B. L. McNaughton, J. O'Keefe, and C. A. Barnes, "The stereotrode: a new technique for simultaneous isolation of several single units in the central nervous system from multiple unit records," *J. Neurosci. Methods*, vol. 8, no. 4, pp. 391–397, 1983.

[4] K. D. Harris, D. A. Henze, J. Csicsvari, H. Hirase, and G. Buzsaki, "Accuracy of tetrode spike separation as determined by simultaneous intracellular and extracellular measurements," *J. Neurophysiol.*, vol. 84, no. 1, pp. 401–414, 2000.

[5] S. Takahashi, Y. Anzai, and Y. Sakurai, "Automatic sorting for multi-neuronal activity recorded with tetrodes in the presence of overlapping spikes," *J. Neurophysiol.*, vol. 89, no. 4, pp. 2245–2258, 2003.

[6] M. Pachitariu, N. Steinmetz, S. Kadir, M. Carandini, and K. D. Harris, "Kilosort: realtime spike-sorting for extracellular electrophysiology with hundreds of channels," *BioRxiv*, p. 061481, 2016.

[7] C. Rossant *et al.*, "Spike sorting for large, dense electrode arrays," *Nat. Neurosci.*, vol. 19, no. 4, p. 634, 2016.

[8] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *ArXiv Prepr. ArXiv13126114*, 2013.

[9] T. Wu, W. Zhao, E. Keefer, and Z. Yang, "Deep compressive autoencoder for action potential compression in large-scale neural recording," *J. Neural Eng.*, vol. 15, no. 6, p. 066019, 2018.

[10] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Trans. Pattern Anal. Mach. Intell.*, no. 2, pp. 224–227, 1979.

[11] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *Commun. Stat.-Theory Methods*, vol. 3, no. 1, pp. 1–27, 1974.

8 supplementary material

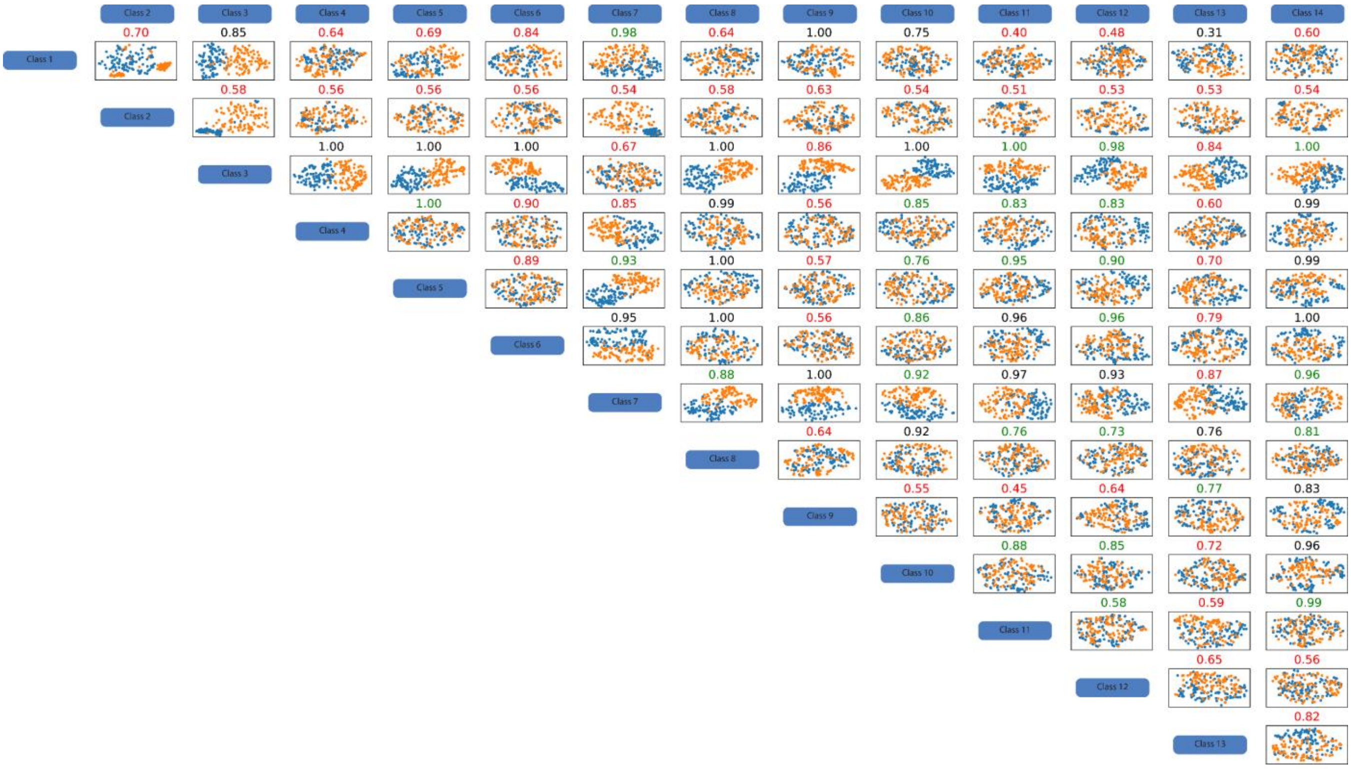

Figure S.1 - PC space pairwise GMM accuracy. t-SNE plot of 27 dimensional PC feature space for pairwise clustering for all clusters pairs. Above each figure is the accuracy of a GMM classifier on that pair. In each cluster 100 random samples are presented.

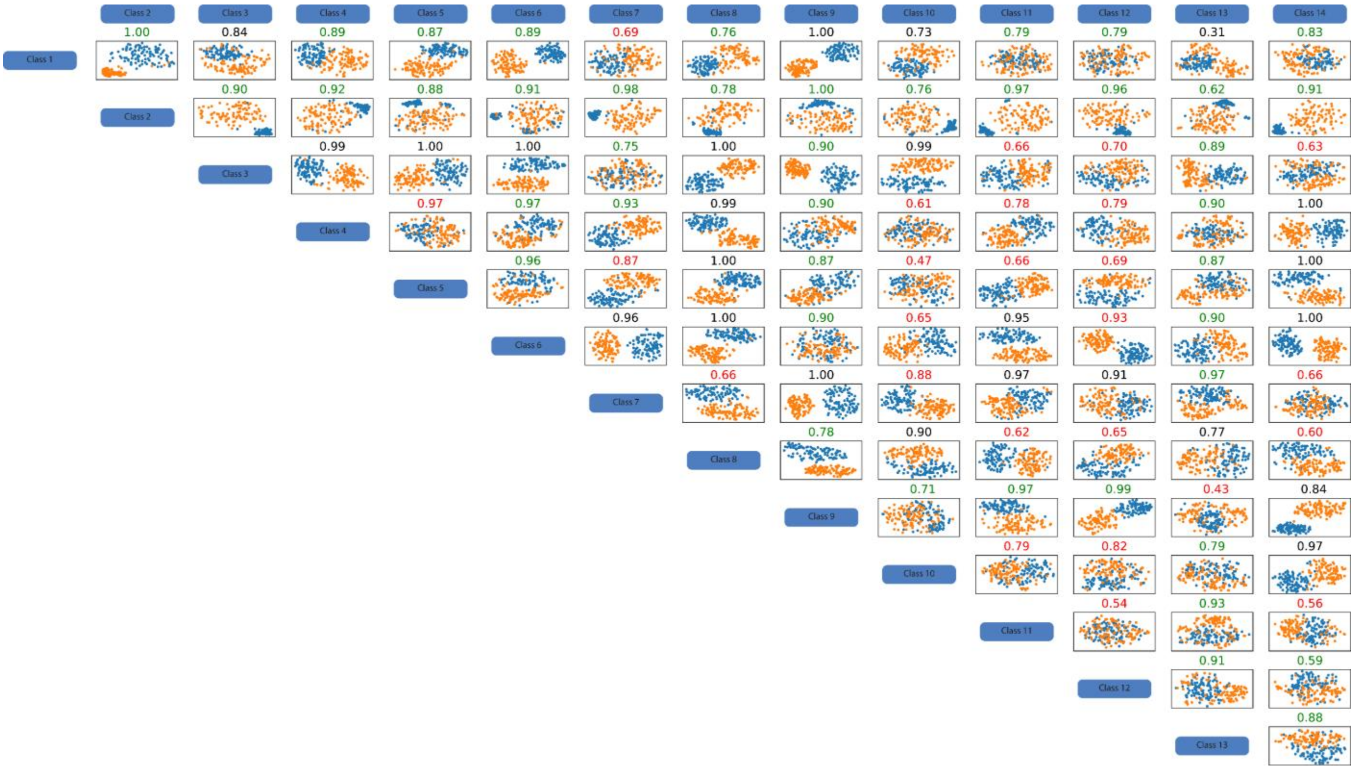

Figure S.2 - - latent space pairwise GMM accuracy. t-SNE plot of 27-dimensional latent feature space for pairwise clustering for all clusters pairs. Above each figure is the accuracy of a GMM classifier on that pair. In each cluster 100 random samples are presented.