Abstract

asdfasdf

Primary reader and thesis advisor:

Secondary readers:

Table of Contents

List of Tables

List of Figures

Chapter 1

Background and Motivation

1.1 Electrochemistry

Electrochemistry is the branch of chemistry studying electron mobility, which leads to the phenomenon known as electricity. This flow of electrons occurs through the transfer from one chemical species to another in what is called an oxidation-reduction reaction. When a substance loses an electron, its oxidation state increases, indicating oxidation. When a substance acquires an electron, its oxidation state decreases, indicating reduction. For example, consider the following redox reaction which has oxidation and reduction components:

$$H_2 + F_2 \rightarrow 2HF \tag{1.1}$$

Oxidation:

$$H_2 \rightarrow 2H^+ + 2e^- \tag{1.2}$$

Reduction:

$$F_2 + 2e^{-} \rightarrow 2F^{-} \tag{1.3}$$

An electrode serves as a stable electrical conductor, facilitating the flow of electrical current within non-metallic solids, liquids, gases, plasmas, or even vacuums. While electrodes often exhibit high electrical conductivity, they are not limited to metals [1]. An electrochemical cell is a device capable of either producing electrical energy through internal chemical reactions or utilizing supplied electrical energy to drive chemical processes within it. This device effectively transforms chemical energy into electrical energy or vice-versa. In an electrochemical cell, reduction and oxidation reactions take place at the electrodes. The electrode where reduction occurs is termed the cathode, while oxidation occurs at the anode.

Electrode potential is the voltage of an electrochemical cell composed of a reference electrode and another electrode to be characterized [2]

1.2 Cyclic Voltammetry

Cyclic voltammetry is a common electrochemical technique that generates important reduction and oxidation information about different molecules [3]. Typically, the working electrode potential increases linearly with time. After a set potential is reached, the potential decreases to return to the initial potential. Theses cycles can be repeated as many times as needed. The rate of voltage change over time is known as the experiment's scan rate (V/s) [4]. Cyclic voltammetry serves as a valuable tool for studying qualitative information about electrochemical processes across diverse conditions. It enables the examination of intermediates in oxidation-reduction reactions and the assessment of reaction reversibility. Moreover, CV facilitates the determination of electron stoichiometry, analyte diffusion coefficients, and formal reduction potentials, aiding in identification processes. Additionally, in reversible, Nernstian systems, the proportional relationship between concentration and current allows for the determination of unknown solution concentrations via the construction of calibration curves correlating current and concentration [5]. In cyclic voltammetry, peaks represent electrochemical processes occurring at the electrode surface. The anodic peak ($E_{p,a}$) is observed during the scan where oxidation of the electroactive species occurs at the electrode and corresponds to the potential at which oxidation is most favourable. The current increases as the potential applied to the electrode becomes more positive, reaching a maximum at the peak potential. The cathodic peak is observed during the reverse scan where reduction of the electroactive species occurs at the working electrode and corresponds to the potential at which reduction is most favorable. The current increases as the potential becomes more negative, reaching a maximum at the peak potential [6]. Typically, researchers are especially interested in these peaks.

1.3 Differential Pulse Voltammetry

Differential Pulse Voltammetry (DPV) is an electrochemical measurement technique from linear sweep voltammetry [7]. The current is measured right before each potential alteration, and the difference in current is plotted against the potential. This method helps reduce the impact of charging current by sampling the current just before the potential change. DPV is well suited for measurements with extremely low concentrations of chemicals. This is because the effect of the charging current can be minimized to achieve high sensitivity, and only the faradaic current, the electric current generated by the redox of a chemical at an electrode, is extracted, so electrode reactions can be measured precisely.

1.4 Potentiostat

A potentiostat is an electronic device used to control the working electrode's potential in a multiple electrode electrochemical cell [8]. Most labs use potentiostats provided by commercial vendors, which are typically governed by proprietary software, employ graphical user interfaces (GUI), and produce processed data. These potentiostats lack the capability for comprehensive control using an application programming interface (API) and direct access to unprocessed measurements. While convenient for manual tasks, these characteristics present difficulties when integrating into automated systems, highlighting the need for potentiostats that are thoroughly digitized to facilitate data-rich experiments and electrochemical process analysis in modern self-driving laboratories. The Matter Lab has developed an open-source potentiostat along with open-source firmware and interface [9]. Notably, the instrument is affordable and compact, making it particularly advantageous for groups with budget constraints or those establishing their initial self-driving laboratory.

When working with a potentiostat, the working electrodes should be immediately polished after use to ensure there are no surface contaminants that inhibit electron transfer. Even a few hours of air exposure will degrade the electrode surface.

1.5 The Matter Lab

The Matter Lab is a research group at the University of Toronto. One of the main research areas is materials discovery with self-driving synthetic laboratory. Developing a fully autonomous self-driving laboratory is a complex endeavor that combines various research disciplines. Machine learning and modeling techniques are utilized to forecast materials properties and propose new experiments. Concurrently, robotics, computer vision, and automated characterization methods are employed to conduct experiments and analyze outcomes. Central to the design of autonomous labs is the integration of these disparate technologies into a cohesive platform, facilitating seamless interaction between experiments and computational modeling [10]. Within the self-driving laboratory (SDL) subgroup, exploration spans multiple domains, encompassing artificial intelligence and optimization methods for experiment control and design, robotics systems for execution, and automatic characterization methods for result analysis. A novel research avenue involves leveraging computer vision to develop visually-aware robotic systems capable of executing chemical and materials science experiments. By automating high-throughput experimentation and streamlining experiment planning and execution, SDLs possess the potential to substantially accelerate research in chemistry and materials discovery. SDLs have played a pivotal role and made noteworthy advancements in various fields including drug discovery, genomics, chemistry, and materials science. SDLs typically use Bayesian optimization to guide its decision-making algorithm. Atlas, a brain for SDLs used software to identify the voltage peak in CV experiments to optimize the oxidation potential of a set of metal complexes.

Chapter 2

Clustering

2.1 Introduction

One of the many goals of the Matter Lab is the development of a self-driving laboratory. Cyclic voltammetry is an important part of the lab as it generates valuable mechanical information for redox-active chemical systems.

2.2 K-Means

K-Means clustering is an unsupervised machine learning algorithm aimed to divide a set of data points into clusters such that the data points within each cluster are similar and different from the data points in other clusters [11]. In this context, K represents the desired number of clusters.

1. Initially, K points are selected randomly as the cluster centroids

2. Each data point is assigned to the closest mean, quantified by the Euclidean distance.

3. Each cluster centroid is updated to reflect the average of data points currently assigned to that cluster

4. This process is repeated for a specified number of iterations

One of the questions that needs to be answered is the choice of K. This means finding a balance between the number of clusters represented by K and the average variance of the clusters while minimizing both. There is no approach that works better than all others in all cases. For this case, the elbow method is used by plotting the within-cluster sum of squares (WCSS) for a range of k and choosing the value k where adding more clusters does not significantly decrease the WCSS.

2.3 DBSCAN

2.4 t-Distributed Stochastic Neighbor Embedding

t-Distributed Stochastic Neighbor Embedding (t-SNE) is a dimensionality reduction technique used for visualizing high-dimensional data in a low-dimensional space [12]. The similarity between two data points is represented by its euclidean distance. The first step of the algorithm is to create a probability distribution that represents the similarity between neighbors. For each data point, it is placed in the middle of the Gaussian curve and the rest of the data is placed along the curve. This is represented by the following equation where $j \neq i$ and $p_{i|i} = 1$:

$$p_{j|i} = \frac{exp(-||x_i - x_j||^2/2\sigma_i^2)}{\sum_{k\neq i} exp(-||x_i - x_k||^2/2\sigma_i^2)} \tag{2.1}$$

"The similarity of datapoint x_j to datapoint x_i is the conditional probablity, $p_{j|i}$, that x_i would pick x_j if neighbors were picked in proportion to their probability density under a Gaussian centered at at x_i" [12]. The last variable that has not been discussed yet is sigma. This variable is not chosen directly, but rather by choosing a value for perplexity. Perplexity is defined as:

$$Perp(p) := 2^{-\sum_x p(x) log_2(p(x))} \tag{2.2}$$

Perplexity represents the density of data and how many neighbors the central point should have with higher values relating to higher variance. After choosing the perplexity value, the corresponding sigma values are found using binary search. Next, the similarities between datapoints for low-dimensional representational will also need to be found to ensure that similar data are close together after projection.

2.5 UMAP

2.6 Curse of Dimensionality

The curse of dimensionality refers to the phenomena that causes various challenges and complications when analyzing data in high-dimensional spaces. As the number of features in a dataset increases, the amount of data needed to generalize accurately grows exponentially. As the number of dimensions increases, the data becomes increasingly sparse. This makes tasks like clustering and classification more challenging. In higher dimensions, the difference between distances between data points start to become negligible, making measurements like Euclidean distance negligible. As such, algorithms that rely on distances measurements will experience a drop in performance. Furthermore, more dimensions will require more computational resources and time to process the data.

2.7 Ramer-Douglas-Peucker Algorithm

The Ramer-Douglas–Peucker (RDP) algorithm is employed to reduce the number of points in a curve approximated by a series of points. It operates by conceptualizing a line between the initial and terminal points within a point set defining the curve. Subsequently, it identifies the point furthest from this line among the intermediary points. If this point, termed the "outlier point", and consequently all intervening points, lie within a specified distance 'epsilon' from the line, they are removed. Conversely, if the outlier point surpasses the epsilon threshold, the curve is segmented into two parts: from the initial point to the outlier point, inclusive and the outlier point and the remaining points. The algorithm is then recursively applied to both resulting segments, and the reduced forms of the curve are reassembled.

2.8 Data Collection

The data used was gathered through an automated electrochemistry experimentation that operates through through an iterative workflow [9]. The proposed workflow was used to synthesize and characterize 10 distinct metals and 10 distinct ligands resulting in 100 unique complexes. Each complex was synthesized using a metal/ligand concentration ratio of 1:7 to ensure complete complexation. The synthesis process employed 1.0 M NaCl in water as the electrolyte/solvent, and a buffer solution consisting of a 1:1 ratio of HOAc/NaOAc. Following synthesis, comprehensive characterizations were conducted using cyclic voltammetry (CV) and differential pulse voltammetry (DPV) techniques. This thorough investigation yielded a substantial database comprising 400 voltammetry datasets. Importantly, our workflow is adaptable, with the potential to encompass a broader range of parameters, including additional ligands, varying metal/ligand ratios, mixed ligands, different buffer pH levels, and reaction times. The accumulation of data points is ongoing, contributing to the continuous expansion and refinement of our understanding.

2.9 Data Preparation

With cyclic voltammetry data, there are many different variables that are unique to each experiment. Particularly, the experiment's scan rate affects the sampling frequency and resolution of data points collected over a given time interval. The length of data obtained changes with the scan rate. However, many data investigation techniques require the data to be the same length. Similarly, the potential limit at which the potential begins to return to its initial point will affect the overall shape of the cyclic voltammogram. To handle this, the following steps are used to prepare the data:

1. Split experiment cycles into separate data points
2. Normalize values to fit between [0, 1]
3. Reduce points using the Ramer-Douglas-Peucker algorithm
4. Duplicate datapoints until total length reaches the longest cycle's length
5. Order datapoints based on angular position relative to the center

Due to the curse of dimensionality, the RDP algorithm is used to reduce the number of dimensions. Since the RDP algorithm takes only a variable ϵ, the final length after reduction will be different for each set of data. To ensure the data has the same length as the longest data after RDP reduction, datapoints are randomly selected and duplicated. Similarly, by ordering the data based on its angular position, the overall shape of the cyclic voltammogram is maintained.

2.10 Results and Discussion

To cluster the data using K-Means, a value of K will need to be selected. This is done using the elbow method. As seen in Figure 2.1, there are many valid values for K, and it is difficult to definitely say which value of K is best. To aid the decision-making process, the Silouhette method is used to analyze promising values. A cluster with a value of 1 means points are perfectly assigned in a cluster and clusters are easily distinguishable, 0 means clusters are overlapping, and -1 means points are assigned to the wrong cluster [13]. The K value should be chosen based on which value

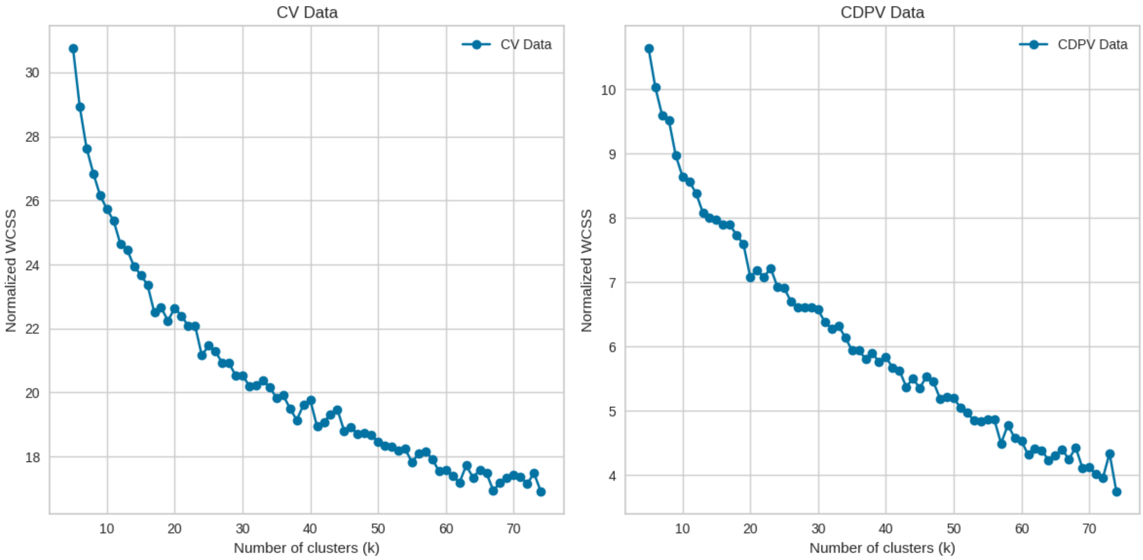


Figure 2.1: K-Means Elbow Method

produces the most clusters with Silhouette scores greater than the average score of the dataset, represented by the red-dotted line. Furthermore, there should not be wide fluctuations in the size of the clusters. The width of the clusters represents the number of data points belonging to the cluster. In Figure 2.2 showcasing the Silhouette method for CV, K = 38 results in the most clusters with a score above the dataset mean while also minimizing the number of clusters with a score below zero and minimizing the variance in the size of clusters. Similarly, in Figure 2.3 showcasing the Silhouette method for DPV, K = 42 results in the best quality of clusters. A subset of the cluster results is available in the appendix. Despite having 100 different combinations of metals and ligands, using a relatively small K value still shows promising results, as the datapoints within each cluster have similar overall shape. To further demonstrate the efficacy of the encoding and classification, t-SNE and UMAP projections are created to visualize the data in 2-D and show how the

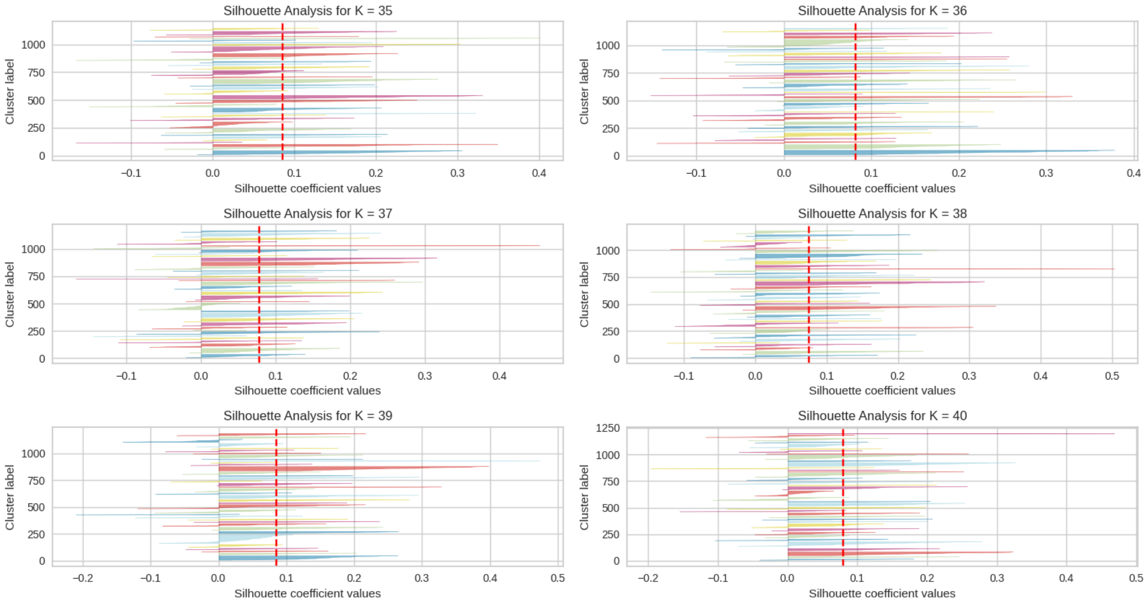


Figure 2.2: CV Silhouette Method

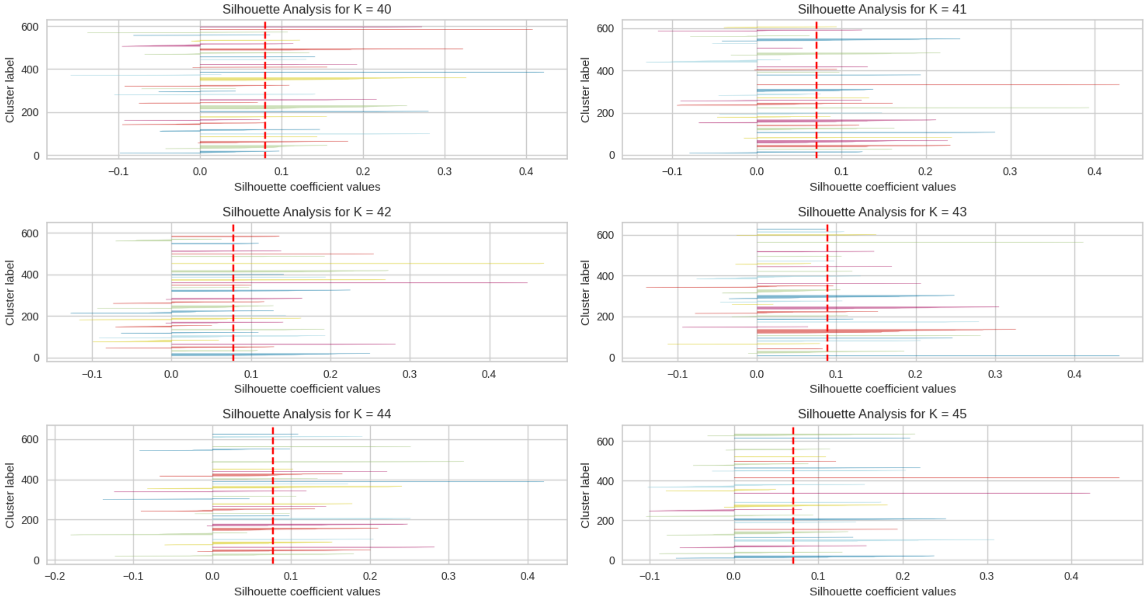


Figure 2.3: DPV Silhouette Method

shapes, metals, and ligands are distributed. Interactive plots made with Bokeh are available. As seen in the resulting figures fig. 2.4 and **??????**, t-SNE emphasizes local structure and tends to agglomerate similar data points into tight clusters. As a result, t-SNE plots often show clearer separation between clusters but may not preserve the global structure as effectively. t-SNE primarily preserves local neighborhoods, which leads to tighter clusters of similar points. However, it may not always capture the global structure accurately, especially for complex datasets. t-SNE embeddings can vary significantly with different random initializations and parameter choices, making it less stable and potentially more sensitive to noise in the data. UMAP tends to focus more on preserving global structure and maintaining relative distance between clusters. Therefore, clusters in the UMAP plot are usually well separated and evenly distributed. UMAP tries to preserve local and global neighborhoods, resulting in more evenly spaced clusters and better representation of both local and global structures. UMAP embeddings are generally more stable across different runs and parameters settings compared to t-SNE. Using machine learning techniques to classify voltammetry data according to the overall shape offers several advantages over simply using a script to find the number of peaks. Machine learning models can be trained to recognize patterns and variations regarding the overall shape and number of peaks. They can adapt to experimental conditions, electrode materials, and analytes without needing manual adjustment of parameters. Voltammetry data can often be noisy, especially at low concentrations. ML models can be trained to distinguish true peaks from noise more effectively than simple peak-finding algorithms. Voltammograms can vary in characteristics due to factors such as electrode deterioration, surface roughness, and solution composition. ML models can learn to handle this variability and provide more reliable peak classification across different experimental conditions. Additionally, ML models can learn when the electrode deteriorates and automatically polish it. ML models can automatically extract relevant features from voltammogram data such as peak heights, peak widths, peak potential, and overall shape. This allows for more comprehensive analysis beyond locating peaks. Once trained, ML models can be integrated into larger data analysis pipelines to classify cyclic voltammetry data rapidly and efficiently, potentially saving time and effort compared to manual analysis or parameter tuning for peak-finding algorithms.

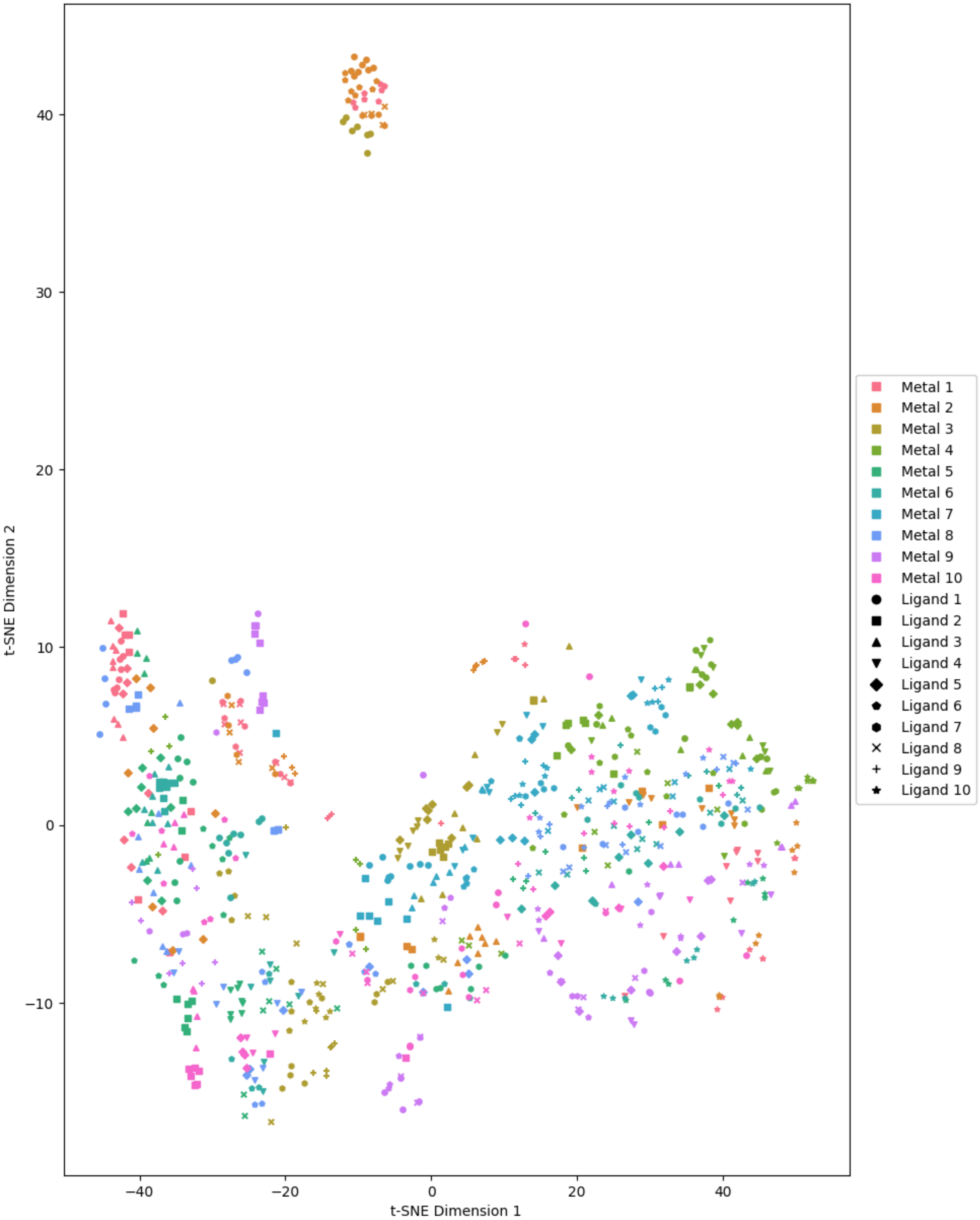


Figure 2.4: Cyclic Voltammetry t-SNE Projection

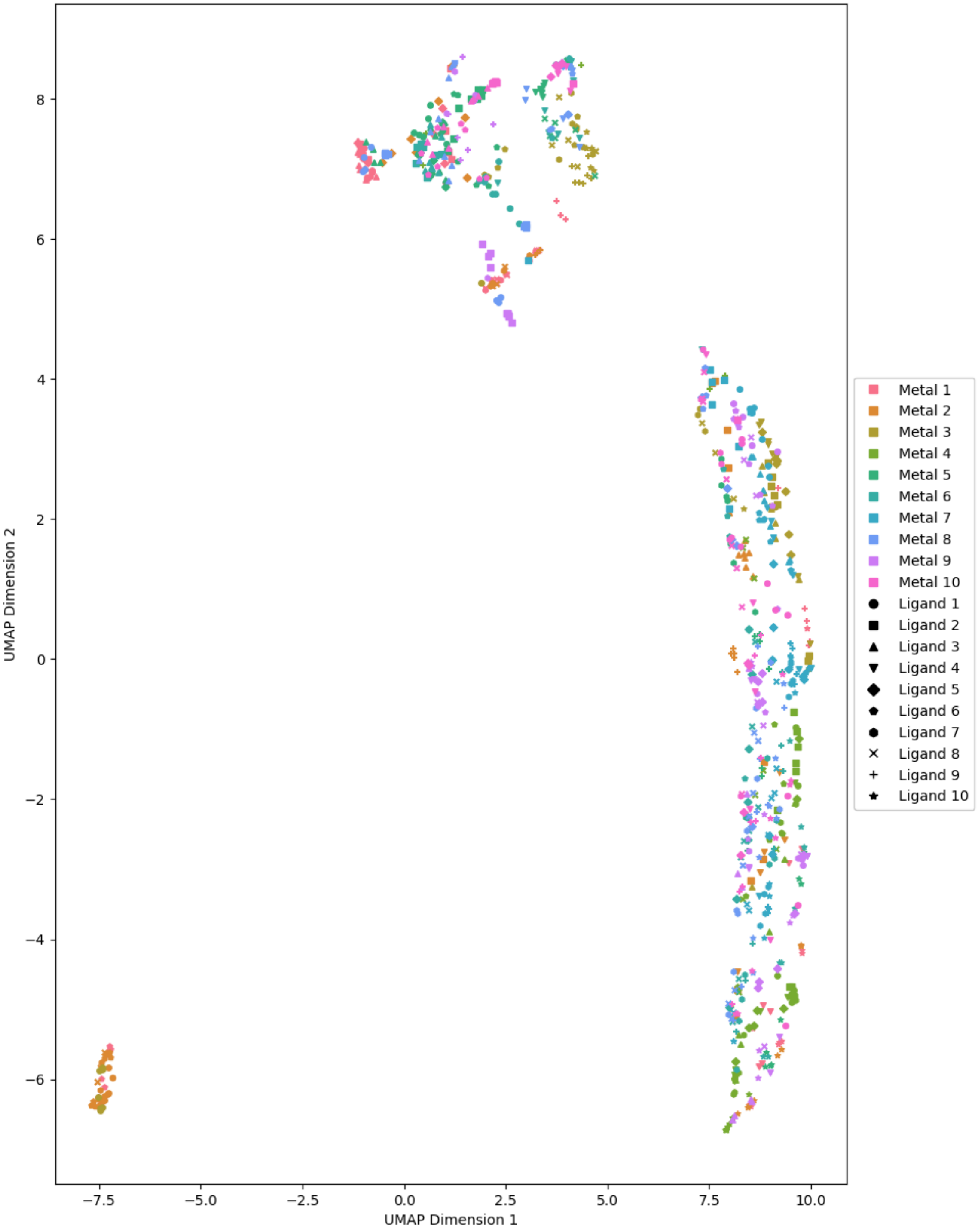


Figure 2.5: Cyclic Voltammetry UMAP Projection

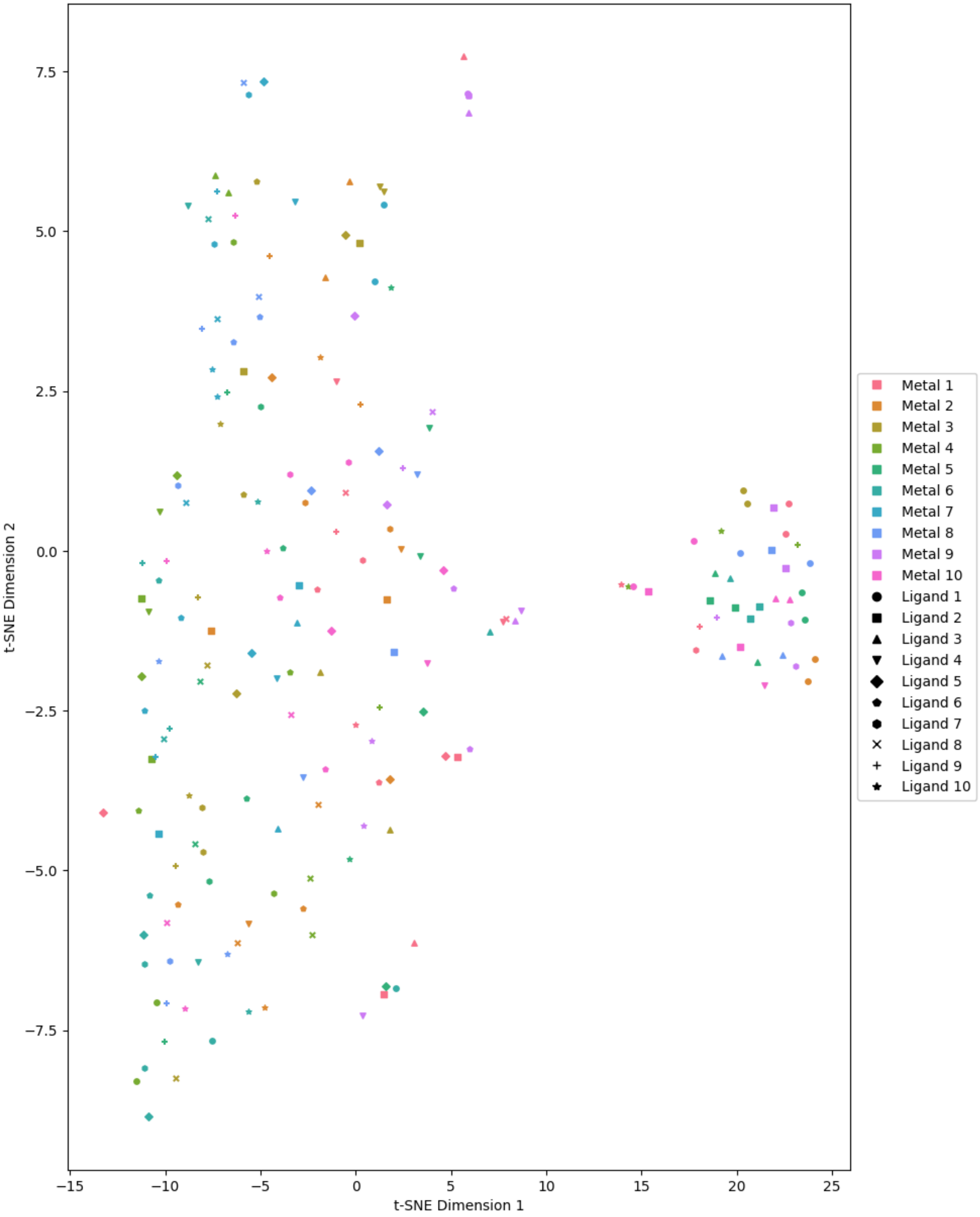


Figure 2.6: Differential Pulse Voltammetry t-SNE Projection

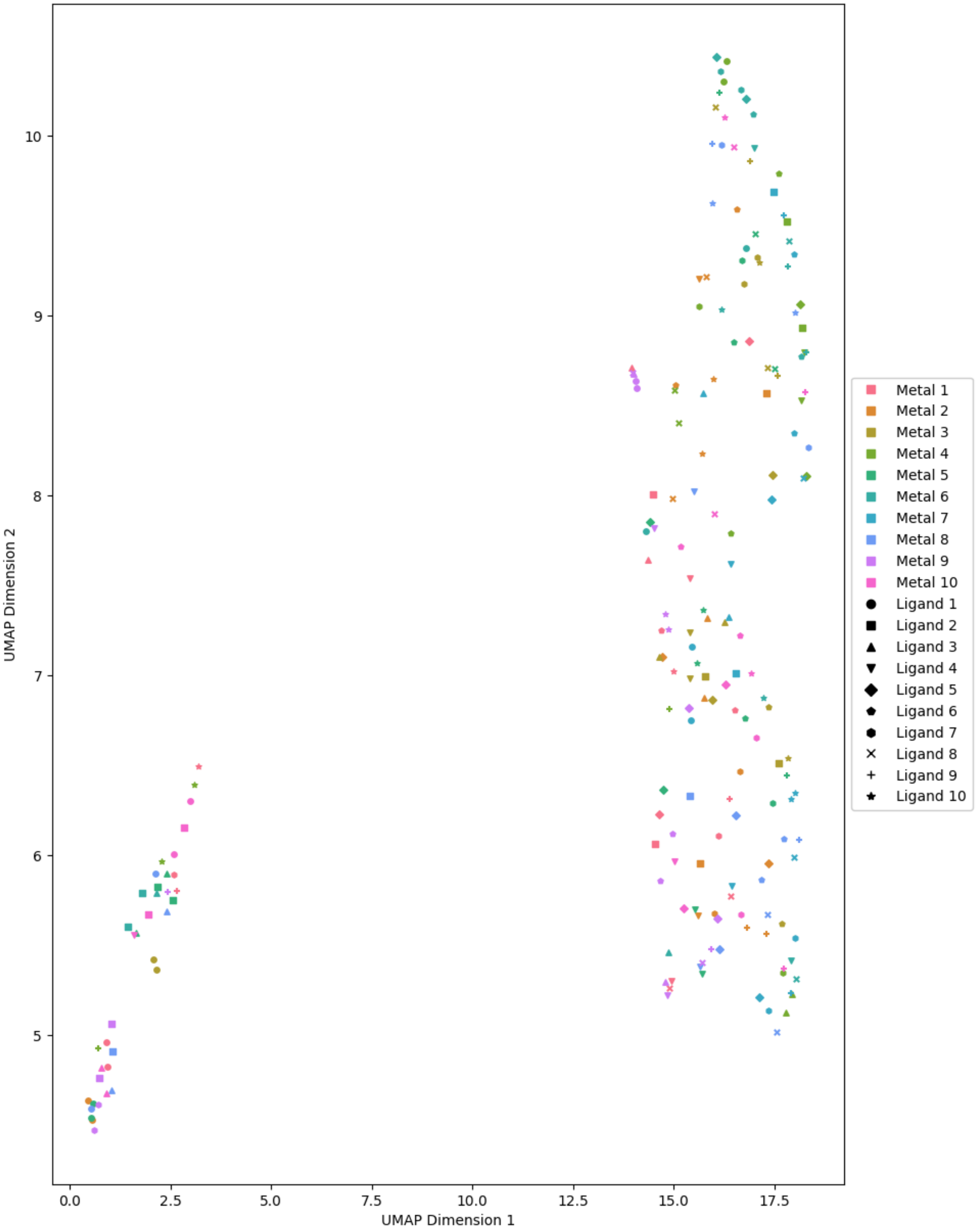


Figure 2.7: Differential Pulse Voltammetry UMAP Projection

Chapter 3

Classification

3.1 Introduction

Classifying the experiments based on the metals and ligands used is explored to further demonstrate the feasibility of using this encoding technique for various machine learning tasks. An important insight to consider is the similarity between voltammetry data and images. After all, each point has a potential and current value, which is similar to an image's RGB values. The main difference is that an image is 2-dimensional while voltammetry data is 1-dimensional. Many previous works have used convolutional neural networks for classification tasks [**SHARMA2018377**]. Using this as inspiration, the proposed model architecture for voltammetry data classification uses 1-dimensional convolutional layers. It is important to note that the dataset used only contains 800 CV datapoints and 200 DPV datapoints for a total of 1000. For training, the dataset was split with 80% for training, 10% for validation, and 10% for testing.

3.2 Variational Autoencoders

Since one of the large challenges faced is the dataset size, one method to address this is to create synthetic data. A variational autoencoder (VAE) is similar to an autoencoder neural network architecture with the main difference being that VAEs connects the encoder to its decoder through a probabilistic latent space that corresponds to the parameters of a variational distribution [**PinheiroCinelli2021**]. The encoder maps each point from the dataset into a distribution within the latent space rather than a single point in that space. The distribution is typically Gaussian with a mean and a variance. Once the VAE is trained, different points can be sampled from the learned latent space distribution. These samples represent different configurations of the input data in the latent space. The sampled points from the latent space are then fed into the decoder network, which generates reconstructions of the input data corresponding to those points. By sampling multiple points from the latent space and decoding them, a diverse set of synthetic data samples that resemble the original data distribution is generated. The variability in the latent space allows for the generation of novel and diverse data samples that capture the underlying characteristics of the training data.

3.3 Conditional Variational Autoencoders

While traditional VAEs learn a latent space for the dataset, conditional variational autoencoders (CVAEs) expand this concept by introducing conditional dependencies between the input data and the latent variables. In the context of generating synthetic data, CVAEs offer a more controlled approach by allowing the generation process to be conditioned on additional information, such as class labels or other attributes associated with the data. By conditioning the generation process on known attributes or labels, CVAEs can generate synthetic data samples that not only capture the underlying data distribution but also adhere to specific conditions or constraints defined by the conditioning variables. This enables targeted generation of synthetic data for different classes or categories, even in the absence of labeled data. In this case, the metal and ligand are encoded using one-hot encoding and passed to the decoder to generate data belonging to the same class.

3.4 Classifier Model Architecture

The model consists of several convolutional layers followed by max-poolling layers to encode the data and reduce dimensions. All layers except for the output layer use the ReLU activation function. The output layer is a dense layer with 10 units and softmax activation function. The Adam optimizer and categorical cross-entropy loss are used to train the model. Additionally, the model uses L2 regularization and early stopping to prevent overfitting and ensure smooth convergence. The Glorot uniform initializer is used for weight initialization to facilitate better gradient flow and prevent exploding gradients.

3.5 Results and Discussion

The accuracy of the classifiers were much better for the CV data compared to the DPV data. This difference can likely be attributed to the size of the datasets. Similarly after incorporating synthetic data generated with the CVAE into the training process,

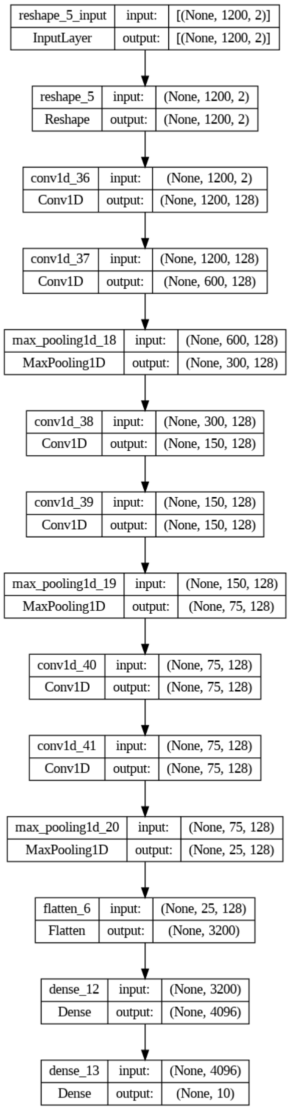

reshape_5_input | input: | [(None, 1200, 2)]
InputLayer | output: | [(None, 1200, 2)]
reshape_5 | input: | (None, 1200, 2)
Reshape | output: | (None, 1200, 2)
conv1d_36 | input: | (None, 1200, 2)
Conv1D | output: | (None, 1200, 128)
conv1d_37 | input: | (None, 1200, 128)
Conv1D | output: | (None, 600, 128)
max_pooling1d_18 | input: | (None, 600, 128)
MaxPooling1D | output: | (None, 300, 128)
conv1d_38 | input: | (None, 300, 128)
Conv1D | output: | (None, 150, 128)
conv1d_39 | input: | (None, 150, 128)
Conv1D | output: | (None, 150, 128)
max_pooling1d_19 | input: | (None, 150, 128)
MaxPooling1D | output: | (None, 75, 128)
conv1d_40 | input: | (None, 75, 128)
Conv1D | output: | (None, 75, 128)
conv1d_41 | input: | (None, 75, 128)
Conv1D | output: | (None, 75, 128)
max_pooling1d_20 | input: | (None, 75, 128)
MaxPooling1D | output: | (None, 25, 128)
flatten_6 | input: | (None, 25, 128)
Flatten | output: | (None, 3200)
dense_12 | input: | (None, 3200)
Dense | output: | (None, 4096)
dense_13 | input: | (None, 4096)
Dense | output: | (None, 10)

Figure 3.1: Classification Model Architecture

there was an significant improvement in accuracy for classifying CV data. However, the DPV classifiers actually saw a decrease in performance. Again, this is likely due to the size of the dataset. In utilizing Variational Autoencoders (VAEs) to generate synthetic data, several key considerations impact classifier performance, especially when dealing with small datasets. Firstly, the quality and diversity of the original

Model	Accuracy (%)
CV Ligands	70.13%
CV Metals	79.24%
DPV Ligands	30.00%
DPV Metals	15.87%

Table 3.1: Classification Results

Model	Accuracy (%)
CV Ligands	81.96%
CV Metals	84.82%
DPV Ligands	29.34%
DPV Metals	13.85%

Table 3.2: Classification Results with Synthetic Data

data influence the effectiveness of the synthetic data produced by VAEs. With limited variation or complexity in a small dataset, the VAE might struggle to capture the true underlying data distribution accurately, potentially resulting in synthetic data that fails to fully represent the characteristics of the real data. This mismatch can detrimentally affect classifier performance. Additionally, the risk of overfitting is heightened in small datasets, where the classifier may excessively specialize on training data patterns that do not generalize well. Introducing synthetic data from a VAE can compound this issue if the VAE itself overfits to the small dataset, producing synthetic data overly similar to the training data, which provides minimal additional information for the classifier and can lead to decreased performance on unseen data. VAEs implicitly learn the probability distribution of the input data. However, if the distribution of the real data is significantly different from the distribution learned by the VAE due to the small dataset size, the synthetic data generated by the VAE may not accurately represent the true data distribution. This distribution mismatch can confuse the classifier, as it may encounter data points in the synthetic dataset that deviate from the real data distribution, leading to suboptimal performance. The area under receiving operating characteristic (ROC) curve shows good results for both metals and ligands. The area under the ROC curve (AUC) calculation summarized the ROC curve analysis into a scalar value, which ranges between 0 and 1. The closer the AUC score to value 1, the better the application's overall performance. From the confusion matrix for metals 3.4, metal 1 was difficult to recognize with many metals being misclassified as metal 1. From the confusion matrix for ligands 3.5, metal 7 was difficult to recognize and was often misclassified as metal 6.

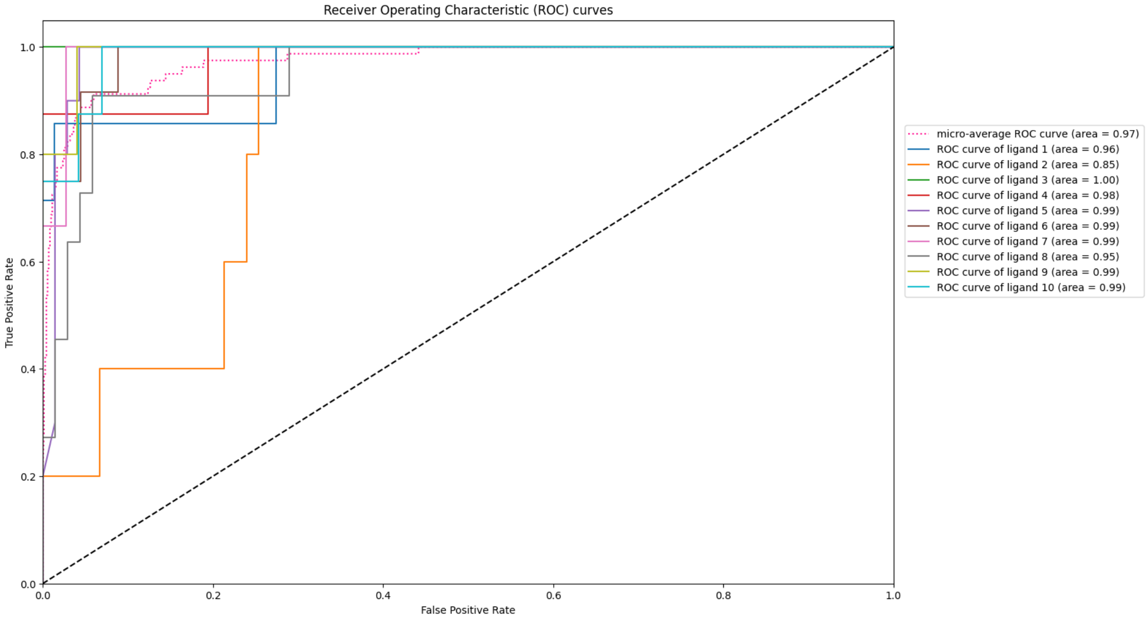


Figure 3.2: CV Ligand ROC Curves

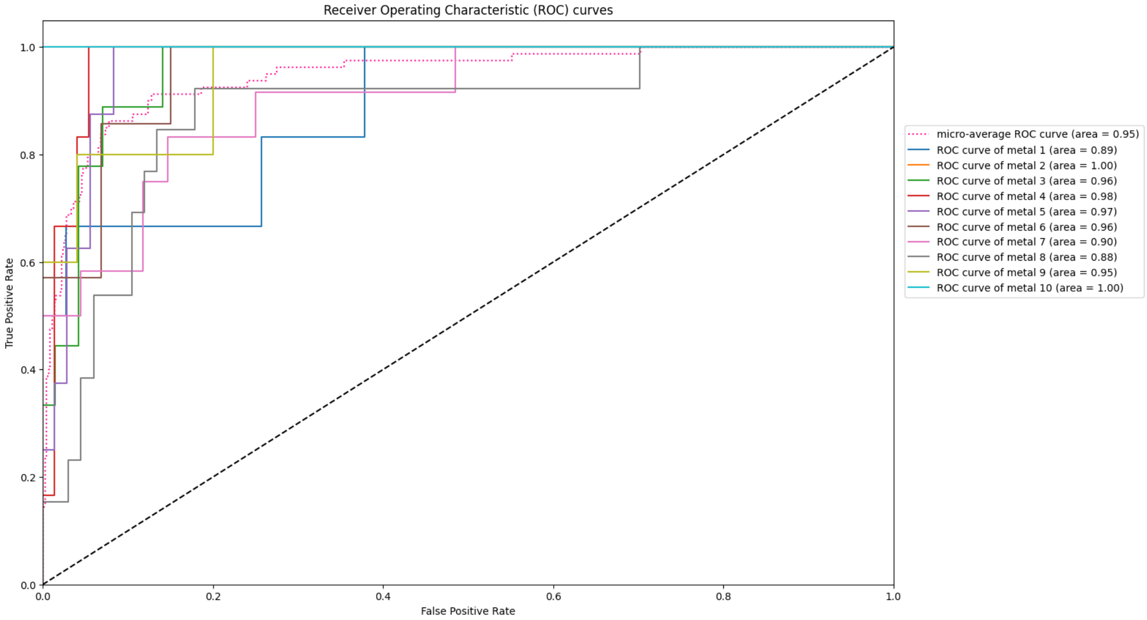
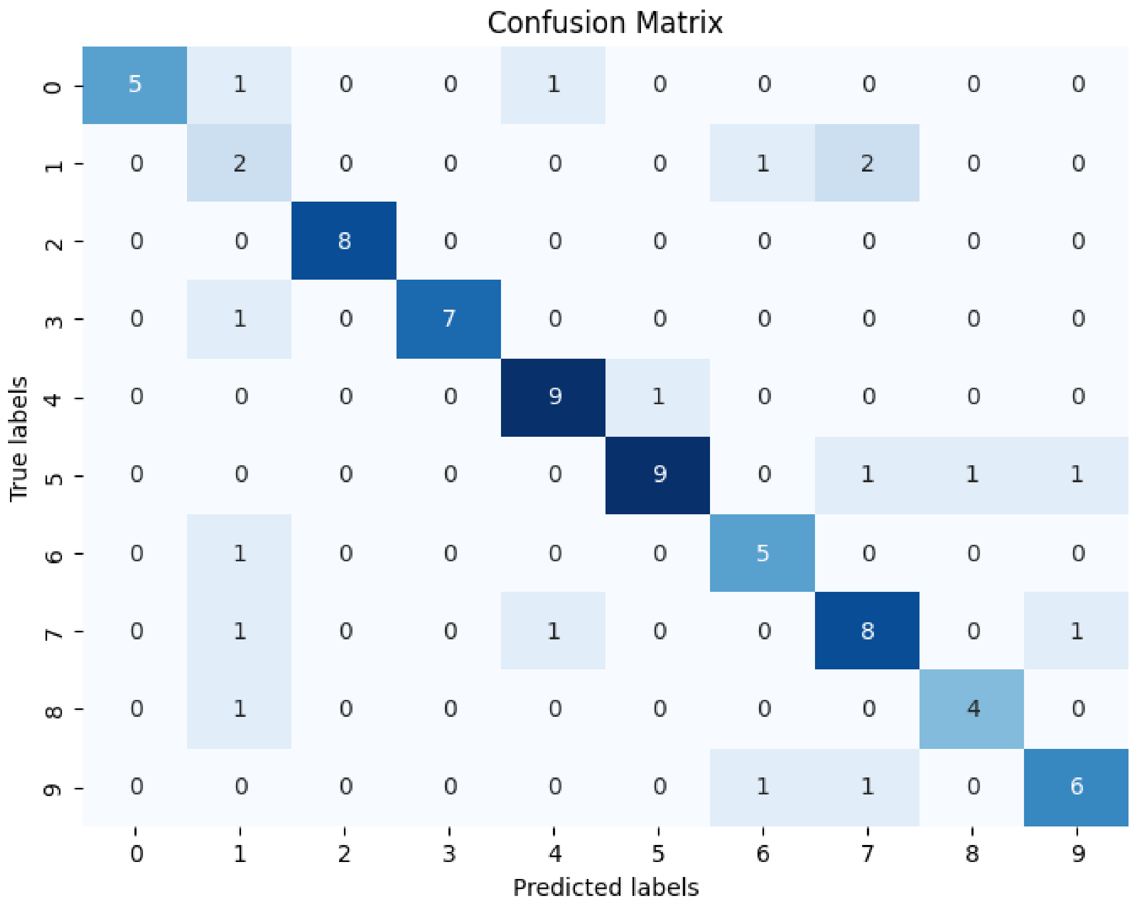


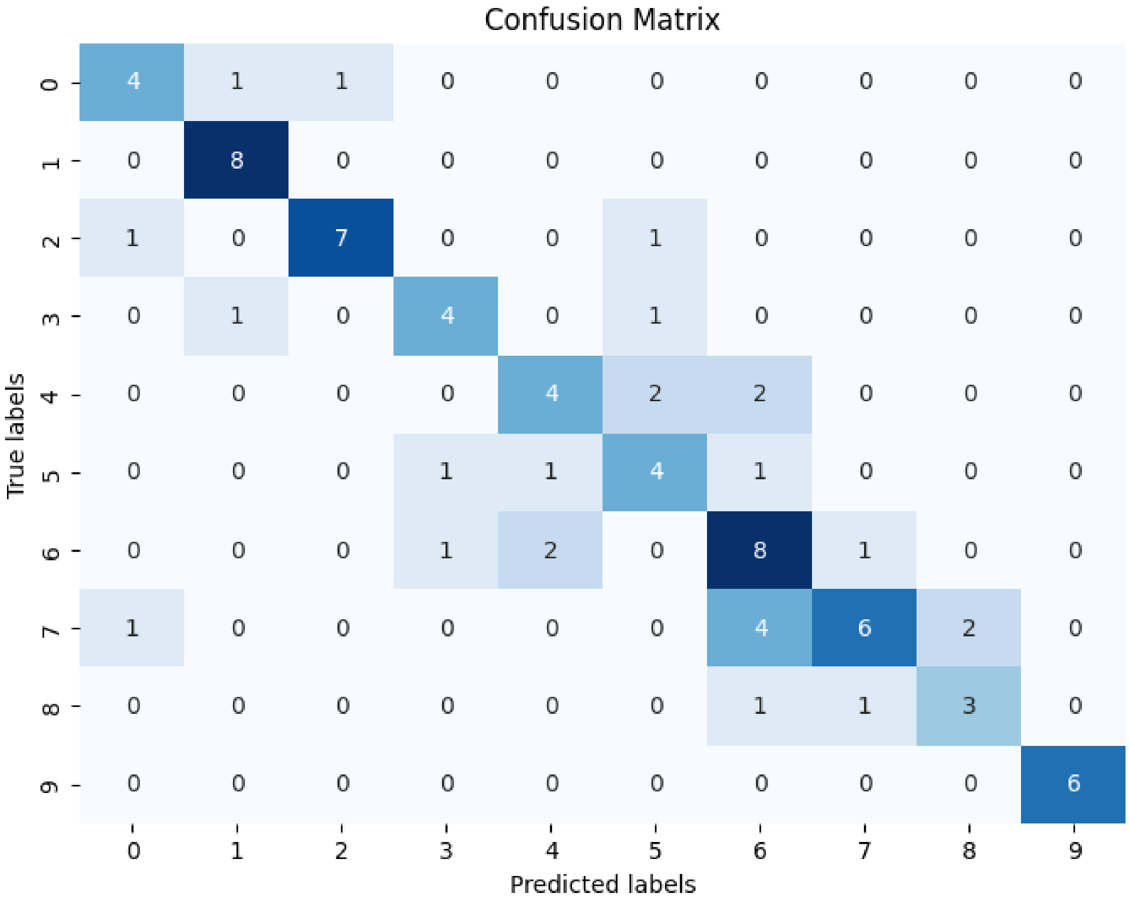
Figure 3.3: CV Metal ROC Curves

A major challenge in supervised learning is providing good examples during training. However, despite using a small dataset, these results are promising.



	0	1	2	3	4	5	6	7	8	9
0	5	1	0	0	1	0	0	0	0	0
1	0	2	0	0	0	0	1	2	0	0
2	0	0	8	0	0	0	0	0	0	0
3	0	1	0	7	0	0	0	0	0	0
4	0	0	0	0	9	1	0	0	0	0
5	0	0	0	0	0	9	0	1	1	1
6	0	1	0	0	0	0	5	0	0	0
7	0	1	0	0	1	0	0	8	0	1
8	0	1	0	0	0	0	0	0	4	0
9	0	0	0	0	0	0	1	1	0	6

Figure 3.4: CV Metal Confusion Matrix



	0	1	2	3	4	5	6	7	8	9
0	4	1	1	0	0	0	0	0	0	0
1	0	8	0	0	0	0	0	0	0	0
2	1	0	7	0	0	1	0	0	0	0
3	0	1	0	4	0	1	0	0	0	0
4	0	0	0	0	4	2	2	0	0	0
5	0	0	0	1	1	4	1	0	0	0
6	0	0	0	1	2	0	8	1	0	0
7	1	0	0	0	0	0	4	6	2	0
8	0	0	0	0	0	0	1	1	3	0
9	0	0	0	0	0	0	0	0	0	6

Figure 3.5: CV Ligand Confusion Matrix

Chapter 4

Denoising

4.1 Introduction

Previous works have developed low-cost potentiostat alternatives that return competitive results compared to the traditional platform based on commercial options [9]. However, these results could still be improved. To try and improve these results, data from commercial options is used for comparison.

4.2 AutoEncoder

An autoencoder is a neural network used to learn an efficient low-dimensional encoding of data. An autoencoder consists of an encoder and decoder. The encoder transforms the input data into an encoded representation, and the decoder attempts to recreate the data from the encoded representation. Since the goal is to try and improve the data quality, the commercial potentiostat data is used for the decoder instead. This way, the low-cost potentiostat data is used to create an encoded representation, and an equivalent commercial potentiostat data is decoded. The main problem to solve is how to pair results from the two potentiostats. The metal and ligand used for each experiment are recorded. However, there are many other variables that can impact the data. As such, the clustering technique described previously to pair similar experimental results that use the same metal and ligand.

4.3 Results and Discussion

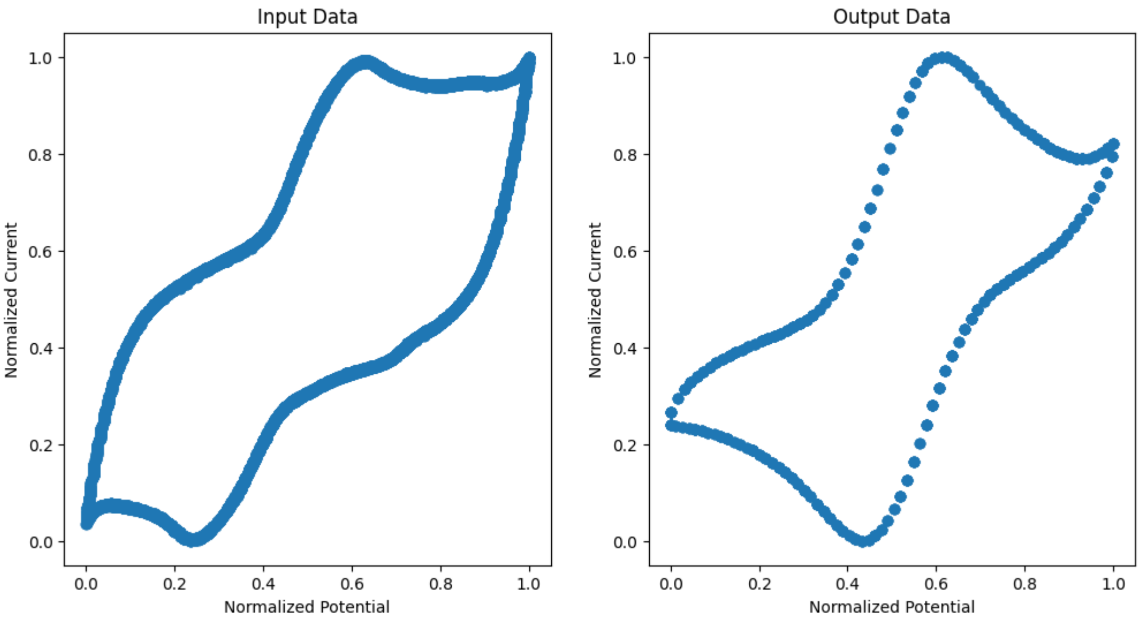


Figure 4.1: AutoEncoder Results

In Figure 4.1, both the As seen in Figure 4.1, both the input and output are similar in overall shape. However, the output contains a much more defined duck-shaped voltammogram, which is typically expected. The results show promising outcomes and indicate that an autoencoder can be effectively transform data from the low-cost potentiostat to resemble data from the commercial potentiostat. By leveraging the capacity of deep neural networks to learn complex patterns and relationships within the data, it becomes feasible to enhance the quality of measurements obtained from low-cost instruments, thereby expanding their utility in research and industrial applications. However, despite the promising results, several drawbacks and considerations must be acknowledged. Firstly, the effectiveness of the transformation heavily relies on the quality and diversity of the training data. Insufficient or biased training samples may lead to suboptimal performance and generalization issues, especially when dealing with complex electrochemical processes or diverse experimental conditions. While the autoencoder can effectively capture and replicate the dominant features present in the data, it may struggle with preserving subtle nuances or domain-specific characteristics inherent to the commercial potentiostat. Variations in hardware specifics, measurement protocols, or environmental factors could introduce discrepancies between the transformed and reference datasets. In conclusion, while autoencoders offer a promising avenue for enhancing the capabilities of low-cost potentiostats, their deployment must be accompanied by rigorous validation and consideration of the aforementioned limitations. Future research could focus on optimizing the autoencoder architecture, exploring alternative deep learning techniques, and investigating strategies for addressing data heterogeneity to further improve the robustness and versatility of the proposed approach.

Chapter 5

Conclusion

Using this novel technique to encode and classify cyclic voltammetry data according to the overall shape and peaks offers significant advantages.

Bibliographic references

1. Faraday, M. S. (B. *On electro-chemical decomposition* 1970.

2. Electrode potential. doi:doi:10.1351/goldbook.E01956 (2019).

3. Nicholson, R. S. & Shain, I. Theory of stationary electrode polarography. single scan and cyclic methods applied to reversible, irreversible, and kinetic systems. *Analytical chemistry* **36,** 706–723. doi:10.1021/ac60210a007. eprint: https://doi.org/10.1021/ac60210a007 (1964).

4. Heinze, J. Cyclic voltammetry—"electrochemical spectroscopy". new analytical methods (25). *Angewandte chemie international edition in english* **23,** 831–847. doi:https://doi.org/10.1002/anie.198408313. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/anie.198408313 (1984).

5. Libretexts. *Cyclic voltammetry* 2023.

6. Grimshaw, J. in *Electrochemical reactions and mechanisms in organic chemistry* (ed Grimshaw, J.) 1–26 (Elsevier Science B.V., Amsterdam, 2000). doi:https://doi.org/10.1016/B978-044472007-8/50001-X.

7. *Electroanalytical methods* 1st ed. en (ed Scholz, F.) (Springer, Berlin, Germany, 2005).

8. J., A. & Faulkner, L. R. *Student solutions manual to accompany electrochemical methods: fundamentals and applicaitons, 2e* en (John Wiley & Sons, Nashville, TN, 2002).

9. Pablo-García, S. *et al.* An affordable platform for automated synthesis and electrochemical characterization. doi:10.26434/chemrxiv-2024-cwnwc (2024).

10. Strieth-Kalthoff, F. *et al.* Delocalized, asynchronous, closed-loop discovery of organic laser emitters. doi:10.26434/chemrxiv-2023-wqp0d (2023).

11. MacQueen, J. B. *Some methods for classification and analysis of multivariate observations* in *Proc. of the fifth berkeley symposium on mathematical statistics*

and probability (eds Cam, L. M. L. & Neyman, J.) **1** (University of California Press, 1967), 281–297.

12. Van der Maaten, L. & Hinton, G. Visualizing data using t-SNE. *Journal of machine learning research* **9,** 2579–2605 (2008).

13. Rousseeuw, P. J. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics* **20,** 53–65. doi:https://doi.org/10.1016/0377-0427(87)90125-7 (1987).

14. Dirac, P. A. M. *The principles of quantum mechanics* (Clarendon Press, 1981).

15. Knuth, D. *Knuth: computers and typesetting* https://www-cs-faculty.stanford.edu/~knuth/abcde.html.

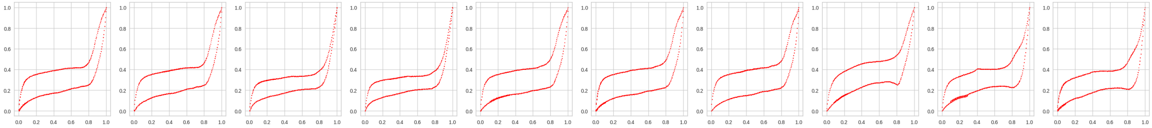
16. Knuth, D. E. in. Chap. 1.2 (Addison-Wesley, 1973).

17. Yoshikawa, N., Akkoc, G. D., Pablo-García, S., Cao, Y., Hao, H. & Aspuru-Guzik, A. Does one need to polish electrodes in an eight pattern? automation provides the answer. doi:10.26434/chemrxiv-2024-ttxnr (2024).

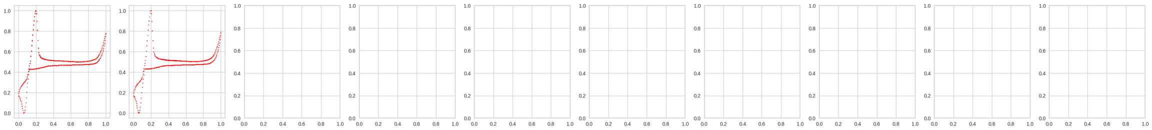
Appendix A

CV K-Means Cluster Results

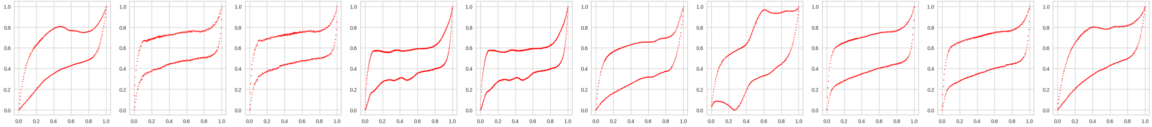
Cluster 1:



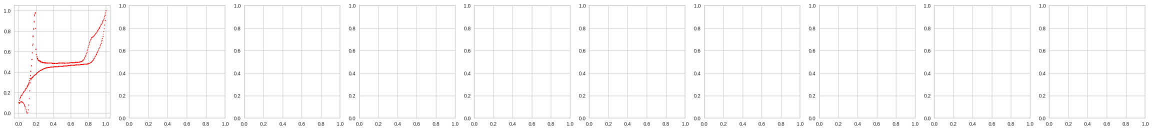
Cluster 2:



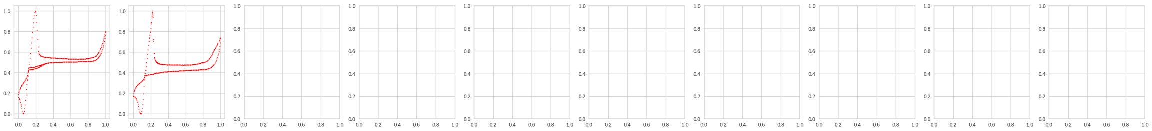
Cluster 3:



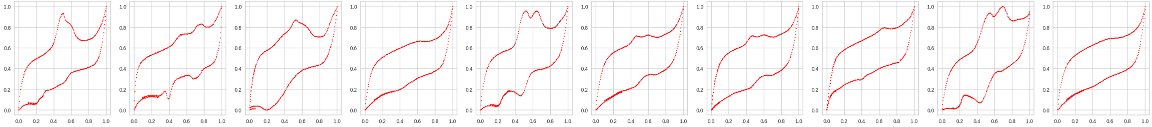
Cluster 4:



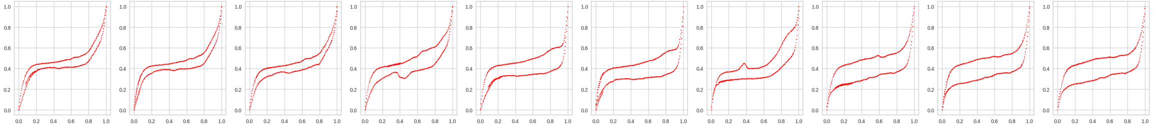
Cluster 5:



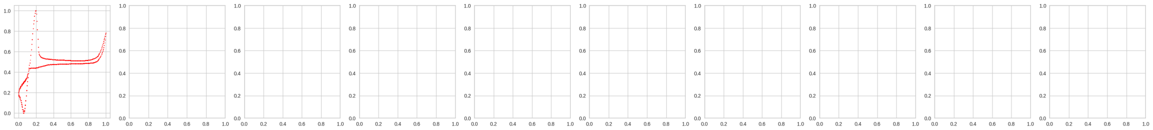
Cluster 6:



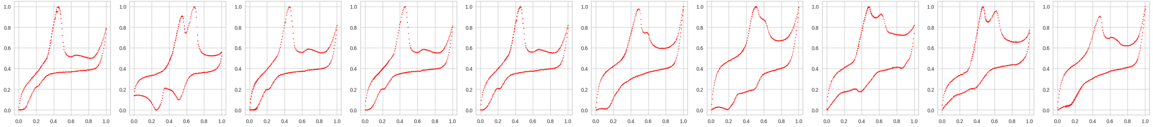
Cluster 7:



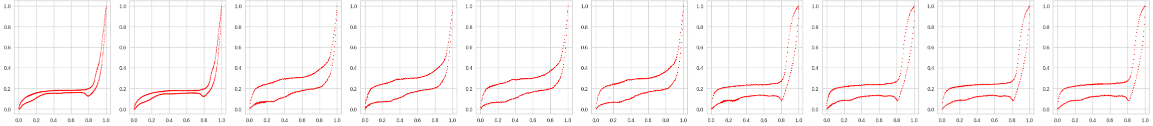
Cluster 8:



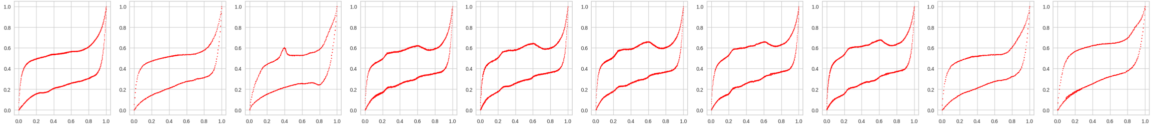
Cluster 9:



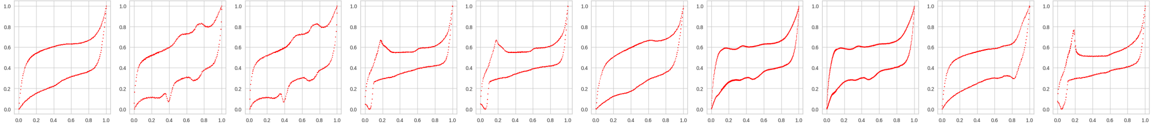
Cluster 10:



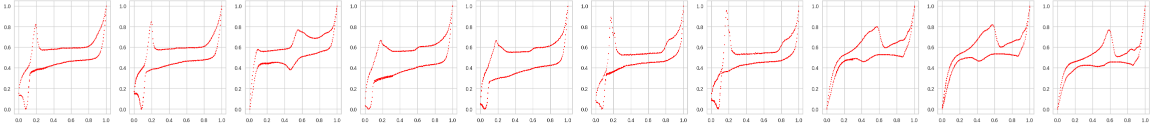
Cluster 11:



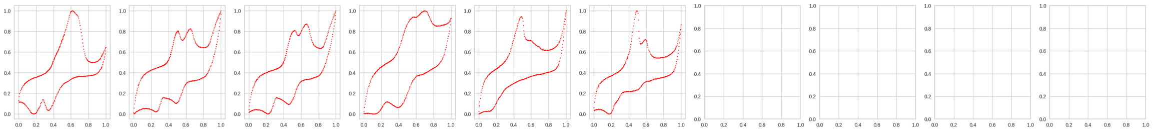
Cluster 12:



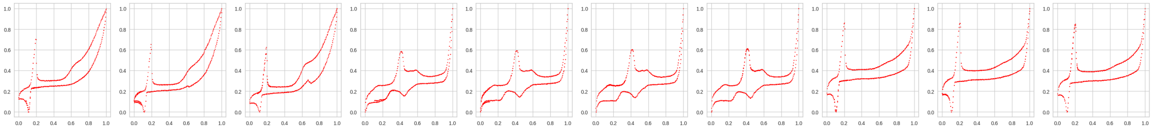
Cluster 13:



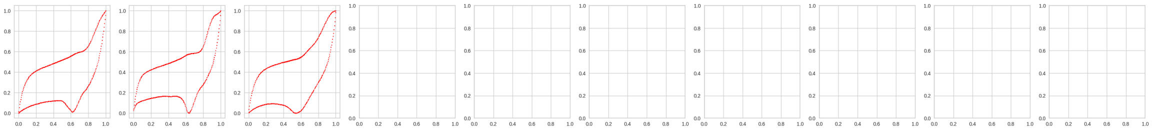
Cluster 14:



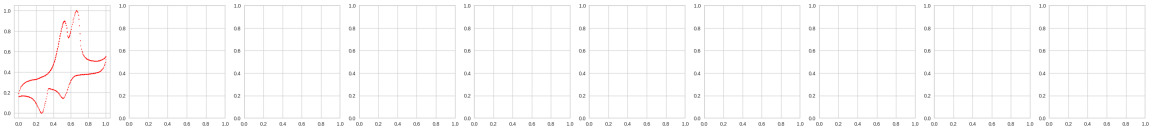
Cluster 15:



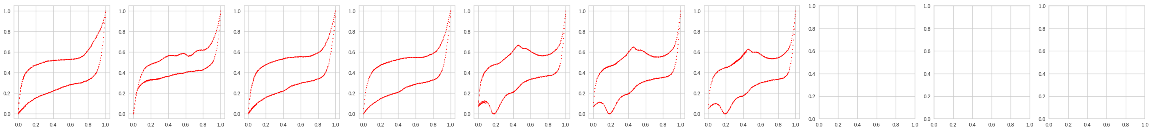
Cluster 16:



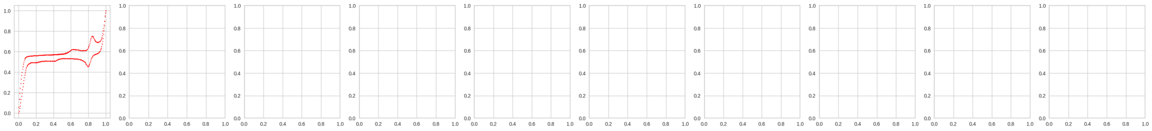
Cluster 17:



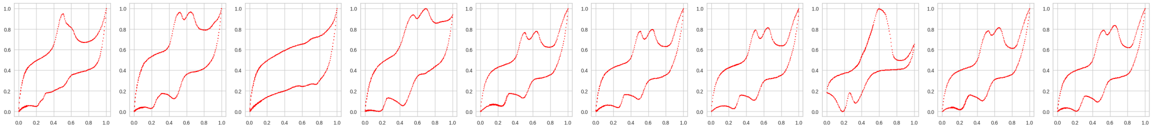
Cluster 18:



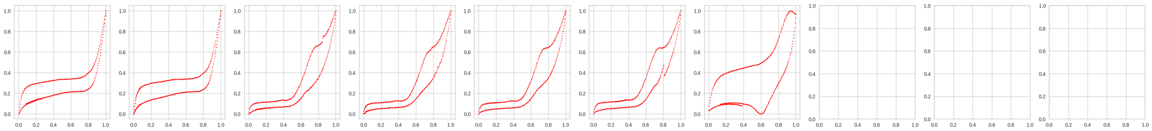
Cluster 19:



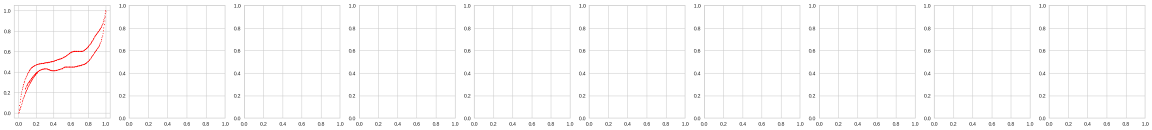
Cluster 20:



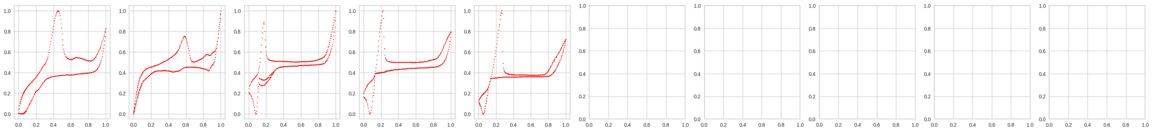
Cluster 21:



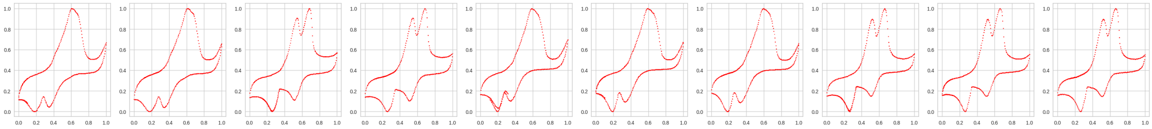
Cluster 22:



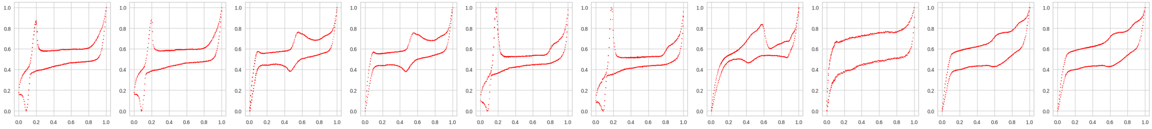
Cluster 23:



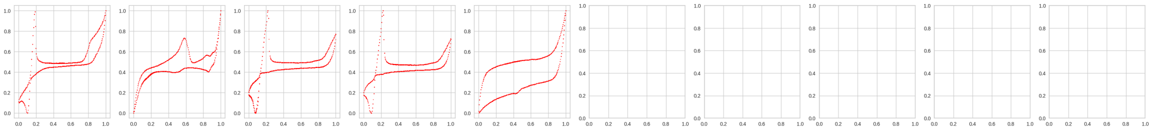
Cluster 24:



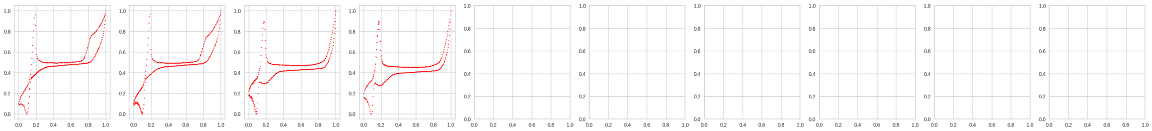
Cluster 25:



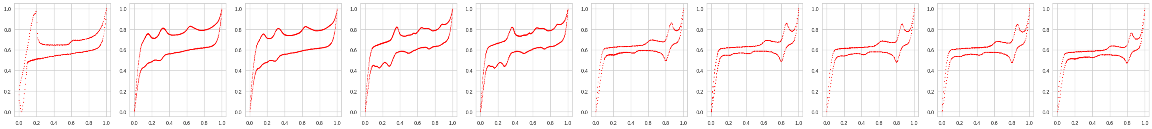
Cluster 26:



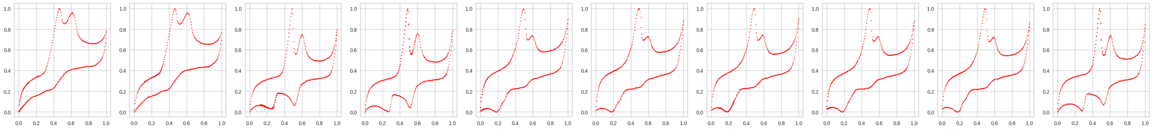
Cluster 27:



Cluster 28:

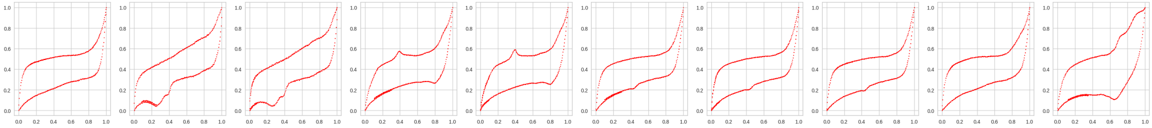


Cluster 29:

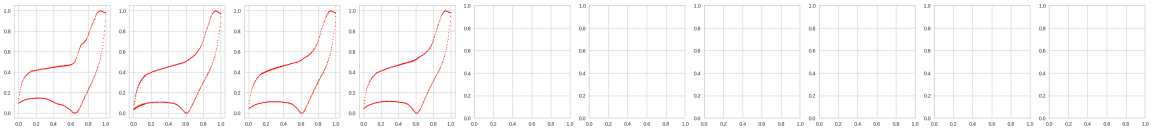


Appendix A. CV K-Means Cluster Results

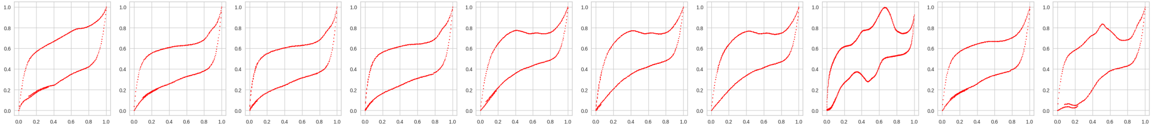
Cluster 30:



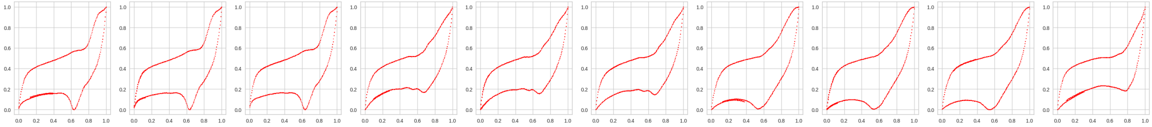
Cluster 31:



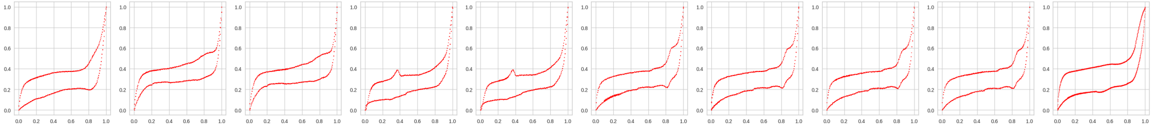
Cluster 32:



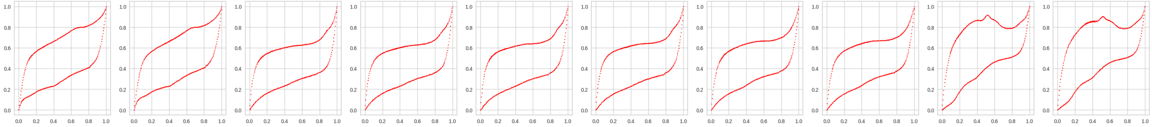
Cluster 33:



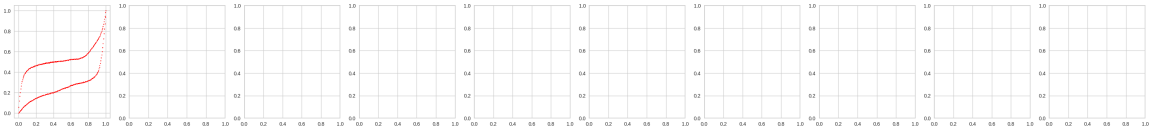
Cluster 34:



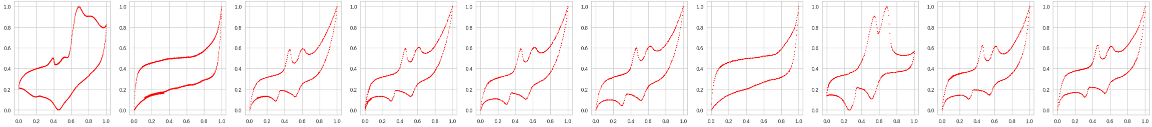
Cluster 35:



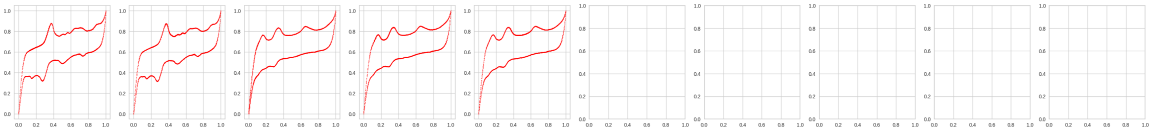
Cluster 36:



Cluster 37:



Cluster 38:


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0