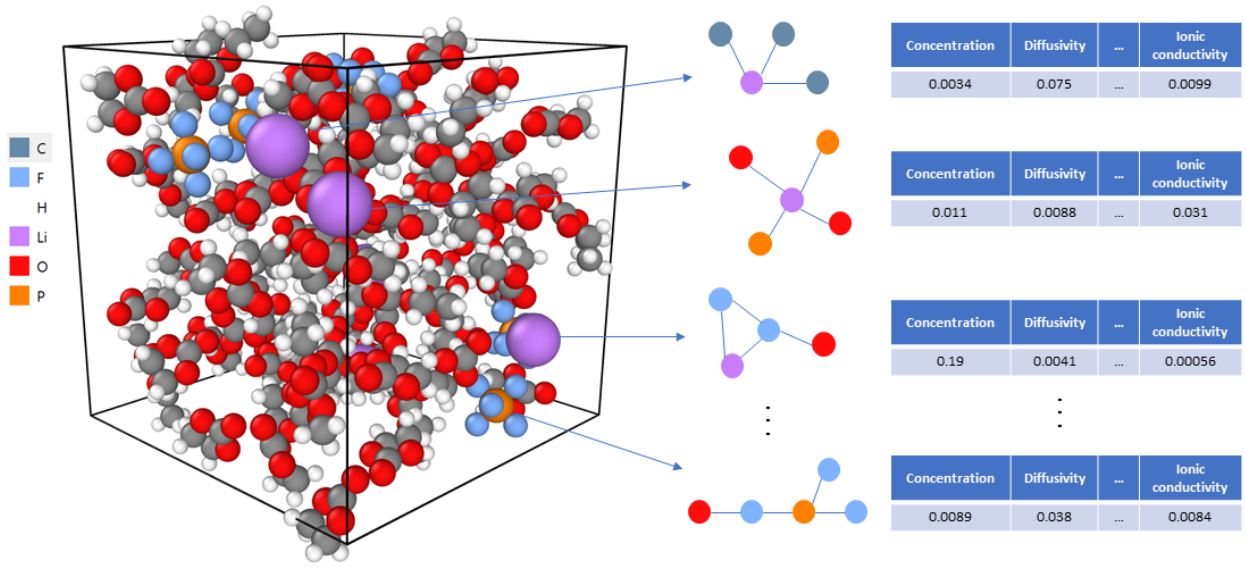
CHALMERS
UNIVERSITY OF TECHNOLOGY



Development of electrolyte descriptors for predicting cycling performance of electrochemical cells

Master's thesis in Complex Adaptive Systems + Systems, Control and Mechatronics

Victor Haugaard
Gustav Silverstam

DEPARTMENT OF PHYSICS
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2023
www.chalmers.se

Master's thesis 2023

Development of electrolyte descriptors for predicting cycling performance of electrochemical cells

Victor Haugaard
Gustav Silverstam

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Physics
Chalmers University of Technology
Gothenburg, Sweden 2023

Development of electrolyte descriptors for predicting cycling performance of electrochemical cells
VICTOR HAUGAARD
GUSTAV SILVERSTAM

© VICTOR HAUGAARD, GUSTAV SILVERSTAM 2023.

Supervisor: Rasmus Andersson, Compular
Examiner: Giovanni Volpe, Department of Physics, Gothenburg University

Master's Thesis 2023
Department of Physics
Chalmers University of Technology
SE-412 96 Gothenburg

Cover: Structural descriptor containing graph level features such as concentration, diffusivity and ionic conductivity for each molecular compound in a simulated representation of an electrolyte.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2023

Development of electrolyte descriptors for predicting cycling performance of electrochemical cells
VICTOR HAUGAARD, GUSTAV SILVERSTAM
Department of Physics
Chalmers University of Technology

Abstract

This master's thesis presents the development of structural electrolyte descriptors utilized for predicting the cycling performance of electrochemical cells such as lithium-ion batteries (LIBs). The research is conducted in collaboration with Compular, a startup company focusing on materials development and battery technology improvement. Through molecular dynamics (MD) simulations and trajectory analysis facilitated by Compular's software, electrolyte descriptors are developed, integrating structural and electrochemical molecular properties. The simulations applied are based on data from battery performance tests, collected and annotated in this project. We propose to use machine learning (ML) to model the relationship between the chemical structure of electrolytes and their performance characteristics. The developed descriptors function as input to a graph neural network (GNN) and thereby offer a novel and efficient method for evaluating electrolyte performance and optimizing electrochemical cells. The findings of this thesis confirm that the descriptors successfully extract necessary information from electrolytes using Compular's analysis software, CHAMPION, and demonstrate their compatibility with the GNN. Moreover, the discussion highlights the importance of annotated data, the complexity of electrolyte descriptors and their predictive abilities. Limitations, challenges and potential enhancements are also addressed, underscoring the need for a larger dataset and exploring possible actions to enhance the performance of the model. In conclusion, this research bridges the gap between empirical experiments and theoretical understanding of battery cycling performance while reducing the need for extensive manual testing. It provides a foundation for further investigations into electrolyte performance prediction and represents a significant step towards more efficient and sustainable battery technologies.

Keywords: lithium-ion batteries, electrolytes, descriptors, machine learning, graph neural network, molecular dynamics

Acknowledgements

We would like to express sincere gratitude to our supervisor at Compular, Rasmus Andersson, and co-supervisors Magnus Rahm and Fabian Årén, for their supportive and helpful guidance throughout this master's thesis. We would also like to thank our colleagues and friends at Compular for giving us this opportunity and providing us with their expertise, enthusiasm, constructive feedback and a welcoming atmosphere at the office. We also would like to thank Professor Giovanni Volpe for taking on this thesis as supervisor and examiner, we sincerely appreciate it.

Victor Haugaard and Gustav Silverstam, Gothenburg, May 2023

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

ANN	Artificial Neural Network
CHAMPION	Chalmers hierarchical atomic, molecular, polymeric, and ionic analysis toolkit
CNN	Convolutional Neural Network
CV	Cyclic Voltammetry
DoD	Depth of Discharge
EC	Ethylene Carbonate
EMC	Ethyl Methyl Carbonate
EoL	End-of-Life
ESW	Electrochemical Stability Window
GCN	Graph Convolutional Network
GNN	Graph Neural Network
HPC	High-Performance Computing
LIB	Lithium-Ion Battery
LiPF6	Lithium Hexafluorophosphate
LiFSI	Bis(fluorosulfonyl)imide
MD	Molecular Dynamics
ML	Machine Learning
MLP	Multi-Layer Perceptron
MSE	Mean Squared Error
ReLU	Rectified Linear Unit
SEI	Solid-Electrolyte Interphase

Contents

List of Acronyms **ix**

List of Figures **xiii**

1 Introduction **1**
1.1 Purpose and Scope . . . 1
1.2 Aim . . . 3
1.3 Limitations . . . 3

2 Theory **5**
2.1 Lithium-Ion Batteries . . . 5
2.1.1 Electrolytes . . . 6
2.1.2 Battery Performance Testing . . . 8
2.2 Molecular Dynamics . . . 10
2.3 Graph Theory . . . 11
2.4 Artificial Neural Networks . . . 13
2.4.1 Graph Neural Networks . . . 15
2.4.2 Descriptors . . . 17

3 Methodology **19**
3.1 Data Collection . . . 19
3.1.1 Target Value Estimation . . . 20
3.2 Simulation . . . 21
3.2.1 Simulation Pre-processing . . . 21
3.2.1.1 Electrolyte Components Annotation . . . 21
3.2.1.2 Molar Ratio Calculation . . . 22
3.2.1.3 Geometry Creation Using CHAMPION . . . 22
3.2.2 Molecular Dynamics Simulation . . . 23
3.2.3 Analysis Using CHAMPION . . . 24
3.3 Descriptor Development . . . 24
3.3.1 Node Embedding . . . 24
3.3.2 Graph Construction . . . 25
3.3.3 Graph Feature Extraction . . . 25
3.3.4 Graph Batching . . . 25
3.4 Model Development . . . 26
3.4.1 Model Design . . . 27

4 Results & Discussion **29**
4.1 Annotated Data 29
4.2 Descriptors 31
4.3 Model Evaluation 33

5 Conclusions & Future Work **37**

Bibliography **39**

List of Figures

2.1 Schematic of a Lithium-Ion battery. 6
2.2 Illustration of capacity fade compared to the number of cycles for three different electrolyte solvents. 10
2.3 MD simulation schematic. 11
2.4 Graph representation of nodes and their respective connections, showing the connection rules of a CHAMPION output. 12
2.5 The architecture of a simple four-layered fully connected ANN with three input neurons $\mathbf{x}$, weights $\boldsymbol{w}^{(n)}$, two hidden layers $\boldsymbol{l}^{(n)}$ of four neurons with corresponding biases $\boldsymbol{b}^{(n)}$ and two output neurons $\mathbf{y}$. . . 13
2.6 Schematic overview of the operations performed from input to output within a neuron in ANN. Input values x_n is multiplied by weights w_{nj}, summarized with an added bias b and then multiplied by an activation function φ. 14
2.7 Typical GNN architecture design. 16
2.8 Example of how graph convolutions propagate features to target node A. 16
2.9 An example of how a simple descriptor of four features describes the structure of a given input. In this example, each row of the descriptor is an atom in the molecule. All four features are numerically represented for each atom in this case. 17

3.1 Illustration of capacity fade compared to the number of cycles for three different electrolytes, including a line for battery failure (70% capacity). The initial drop in capacity is due to equilibration cycles, performed to create the SEI. 20
3.2 Representation of EC in a Python dictionary. 21
3.3 An illustration of the created starting geometry using CHAMPION. . 23
3.4 An illustration of the geometry after an electrolyte simulation using CP2K, final timestep. New molecular compounds have been shaped from molecules and atoms. 23
3.5 Illustration of what the electrolyte descriptors are representing. The features in this example are arbitrary. 26

3.6 Schematic overview of the GNN purpose of the network. Electrolyte descriptors as DataElectrolyte objects are sent into the network, where each molecular compound becomes a graph with each molecule as embedded nodes. The node-embedded graphs are sent into graph convolutions, that perform two layers of message passing within each graph, followed by pooling and summarizing each graph into a weighted value. 27
3.7 Illustration of the full network architecture from developed electrolyte descriptor input to predicted cycle performance output. The network is divided into graph convolutions, graph pooling, graph feature concatenation, sorting and padding, graph-wise fully connected layer, fully connected layers and prediction. Input vector and matrix sizes are shown in between all network operations. 28

4.1 Cycle performance of 104 annotated battery cycle tests, with *Cycles to 70%* capacity of each electrolyte test. The temperature used within the test is also represented. The relevant data from test 81 is presented in the bottom part of the figure. 29
4.2 Illustration of all attributes of a DataElectrolyte object. 31
4.3 Representation of an electrolyte including all molecular compounds in it. 32
4.4 Plot of graph number 12 in the electrolyte along with its corresponding graph features. In this example, the included graph features are charge, mass, concentration, molar conductivity, electrical mobility, ionic conductivity and diffusivity. 32
4.5 Training and validation loss of the GNN over 1000 epochs. 33

1
Introduction

A societal trend of sustainable solutions, green innovations, and environmentally friendly processes is evidently flourishing in the industrial market [1]. This change has been accompanied by a parallel surge in the development and implementation of artificial intelligence [2], which opens up a new universe of problem-solving methods. Of particular relevance to this thesis is battery development. However, a majority of methods currently employed to improve battery technology require a laborious process of manual testing and iteration, followed by an extended period of analysis to determine performance and lifespan-related properties [3].

Compular [4] is a startup company and university spinout from the battery research group of Professor Patrik Johansson at the Physics Department of Chalmers University of Technology, with the vision to digitalize materials development. Starting with liquid electrolytes for lithium-ion batteries (LIBs) and similar chemistries, Compular develops a software tool to automate high-quality molecular dynamics (MD) simulations and trajectory analysis to accelerate the work of industrial battery R&D departments, including their own unique patented analysis methods.

Compular's software can predict a much richer set of properties of liquid electrolytes than available from commonly available experimental or other computational techniques. A notable exception is the uniqueness of their technology, which discovers the emergent structures in electrolytes by detecting bonds between atoms based on which pairs of atoms move together. The analysis thus finds both the covalent bonds holding molecules together, as well as other cohesive interactions including ion-ion and ion-solvent coordination and hydrogen bonding. Based on these bonds, the tool creates a time-dependent global bond graph of the simulated system. Most of the subsequent analysis is based on partitioning this graph into smaller structures using either connected components or node embeddings and using statistical physics to characterize the properties of each topologically distinct structure. Overall system properties can be computed by aggregation of structure properties, which allows explaining how the distribution of structures gives rise to the system properties [5].

1.1 Purpose and Scope

A common problem for battery cell developers is predicting which electrolyte compositions will give rise to good battery performance. Good performance is measured

by building a coin cell and subjecting it to repeated charge and discharge cycles according to some predetermined schedule. This is called cyclic voltammetry (CV), herein referred to as cycling. During cycling, the battery capacity generally fades over time and for most possible combinations of electrolytes and electrodes, the cell operation abruptly stops functioning after a modest number of cycles, rendering the cell unusable. This capacity loss normally returns close to baseline when returning to lower charge/discharge rates. In addition, the faster the cell is charged and discharged, the less energy can be extracted per cycle. These types of cycling experiments are the main way cells are tested [6]. In electrolyte development, most R&D departments keep the other cell components constant while varying the electrolyte composition. It has been found empirically that small variations in electrolyte composition can affect battery performance significantly [7].

The physics determining cell cycling performance as a function of electrolyte composition is much too complicated to be accurately modeled, in part because electrolytes in LIBs usually operate outside of their electrochemical stability window (ESW) [8], so that cell operation is enabled only by the formation of a protective film on the electrode-electrolyte interface which permits lithium ions to pass through while avoiding the continuous breakdown of electrolyte components. This film is known as the solid-electrolyte interphase (SEI) and is a dynamic structure that is believed to be both heterogeneous and dynamic [9]. While the prediction of the structure and dynamics of the SEI is in very high demand, doing so reliably from first principles is an unsolved problem.

Nevertheless, one can more straight-forwardly reason about what electrolyte properties should in principle determine the nature of the SEI, even if no solution can be found: the distribution of local structures, their formation, breakage and recombination dynamics, transport properties, and electrochemical stabilities, perhaps in addition to a few system-wide properties, e.g. density and viscosity. These properties coincide with the properties predicted well by Compular. This could enable the development of a set of electrolyte descriptors for predicting cycling performance based on supervised learning, where the molecular compounds and cycling data come from experiments, while the rest of the descriptors are based on Compular's simulation and analysis techniques.

Compular's analysis methods provide the means necessary to predict these properties, which could enable the development of electrolyte descriptors. These descriptors, combined with experimental data on molecular compounds and cycling performance, could enable the creation of predictive models for cycling performance. This approach would bridge the gap between empirical experiments and theoretical understanding, facilitating easier battery analysis and reducing the need for extensive manual testing. The situation begs the question: is it possible to develop efficient electrolyte descriptors, that capture relevant data from Compular's analysis software, and includes the decisive information that enables a machine learning (ML) model to produce accurate cycling predictions?

1.2 Aim

The aim of this thesis is to develop structural electrolyte descriptors that could be utilized to predict the cycling performance of electrochemical cells.

1.3 Limitations

The main objective is to focus exclusively on the development of structural electrolyte descriptors. As a primary limitation, an initial version of a neural network will be outlined to confirm the predictive capabilities of the descriptors, rather than predicting cycling performance accurately. Consequently, the scope of tuning and model evaluation will be confined to analyzing the utility of using the descriptors for prediction purposes. Moreover, the development of electrolyte descriptors is subject to limitations imposed by the availability of data collected from open-source datasets and the time allocated for data collection. It is important to note that this project is considered to be a preliminary study, which inherently restricts the extent of analysis and reliance on existing research findings for its outcomes.

2
Theory

In this section, the fundamentals of lithium-ion batteries are explained along with the properties of electrolytes in such batteries. Furthermore, the techniques used when analyzing battery performance are presented as well as the MD used when simulating electrolytes. Additionally, the theoretical background of artificial neural networks (ANNs) and graph neural networks (GNNs) is introduced.

2.1 Lithium-Ion Batteries

LIBs are extensively employed as power sources in a wide range of applications owing to their high energy densities, high Coulombic efficiencies, wide electrochemical stability, high ionic conductivity, low self-discharge features and a range of voltages accessible with diverse electrode designs [10, 11]. LIBs have higher energy density compared to other rechargeable batteries such as nickel metal hydride batteries and lead acid batteries, which is due to the higher operating voltages of LIB. While energy density is one of the most important factors for portable electronics, cycle life is just as relevant in some aspects, especially when the lifespan of the battery is considered [12].

The structure of a LIB is presented in Figure 2.1. The battery consists of two electrodes, a separator, two current collectors and an electrolyte. The negative and positive electrodes, also known as anode and cathode, are usually separated by a porous separator. The electrolyte is a liquid consisting of one or several carbonates like ethylene carbonate (EC) and ethyl methyl carbonate (EMC). During charge, lithium ions move from the positive electrode (cathode) to the negative electrode (anode) through the electrolyte. This creates free ions in the anode, which flows through an external cicuit to the cathode, creating an electric current. The opposite happens when discharging the battery, when the lithium ions move through the electrolyte from the anode to the cathode a flow of electrons through an outside current is generated [13] which creates a current to the electronic device that is plugged in. This procedure is possible due to the attributes of the electrolyte, which has an important role in ensuring the functionality of the battery, e.g. enabling ion transport while disabling electron transport between the electrodes [14].

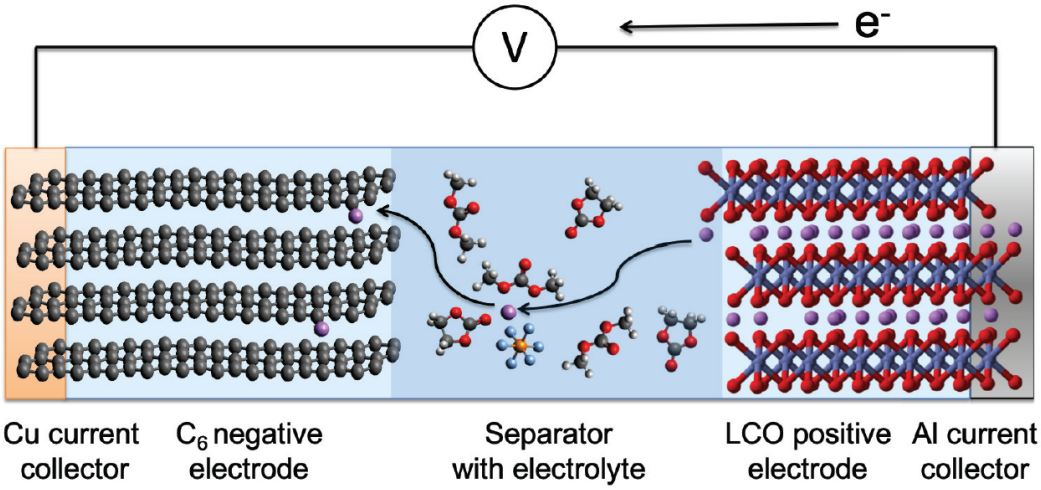


Figure 2.1: Schematic of a Lithium-Ion battery.

2.1.1 Electrolytes

Electrolytes play a crucial role in batteries, serving as the medium through which ions flow between the electrodes and facilitating the conversion of chemical energy into electrical energy. An electrolyte is a solution of a salt in a solvent that contains ions and conducts electricity when dissolved in a solvent, e.g. water. The electrolyte is typically a liquid or gel that contains ions, which can move between the two electrodes as the battery discharges and charges. During discharge, the electrolyte undergoes a chemical reaction that causes ions to flow from the anode to the cathode, generating an electric current as seen in Figure 2.1. In modern batteries, this reaction is reversible, meaning the ions flow back from the cathode to the anode, restoring the battery's state of charge during charging [15, 16].

The choice of electrolyte in a battery is critical due to its effect on electrochemical performance, safety and lifespan [11]. As mentioned above, common electrolytes for LIBs consist of a lithium salt such as lithium hexafluorophosphate (LiPF6) or lithium bis(fluorosulfonyl)imide (LiFSI) dissolved in an organic solvent [17, 18]. The characteristics and properties within a specific electrolyte play a relevant role in the determination of the potential cycle lifespan and behavior under certain conditions. From a cycle perspective, this thesis identifies the following properties that are essential to consider when creating electrolyte descriptors:

- *Ionic conductivity*: The ability of the electrolyte to conduct ions is crucial for the overall performance of the battery, as it affects the rate of charge and discharge. The rate at which electrons spontaneously flow from the more negative to the more positive potential of the electrodes. Herein, the partial conductivity of the active ion (Li^+ in LIB) is especially important, since this is the ion involved within the electrochemical reactions that drive the cell [19, 20].

- *Electrical mobility*: Refers to the speed at which the charged particles, like ions, move when interacting with the electrical field created by the electrodes [21].

- *Electrochemical Stability Window (ESW):* The range of potentials or voltages within which an electrolyte is stable against oxidation and reduction. At too high or too low voltages, the oxidation or reduction of the electroactive species may cause undesirable side reactions, such as the breakdown of the solvent or electrode materials, gas evolution, or the formation of unwanted byproducts. The ESW is determined by the redox potentials of the species involved in the electrochemical reaction, as well as by the chemical and physical properties of the solvent, electrode and the reaction environment [22, 23].

- *Solvent stability*: The stability of the solvent in the electrolyte is important for maintaining the homogeneity of the solution and avoiding degradation over time [24].

- *Redox stability*: The stability of the electrolyte against oxidation-reduction reactions is important for maintaining the stability and consistency of the battery voltage over time [25].

- *Viscosity*: The viscosity of the electrolyte affects the ion transport within the battery and therefore affects the overall rate performance of the battery [23].

- *Concentration*: The concentration of the inherent atoms, molecules and molecular compounds of an electrolyte determine the rate at which they appear.

- *Thermal stability*: The thermal stability of the electrolyte is important for ensuring that the electrolyte does not decompose at high temperatures, which can cause battery degradation [26].

- *Electrolyte/electrode interfacial resistance*: The resistance at the interface between the electrolyte and the electrode affects the rate and efficiency of ion transport and therefore, the overall performance of the battery [24, 26].

- *Electrolyte/electrode compatibility*: The compatibility between the electrolyte and the electrodes is important for reducing the formation of unwanted byproducts which can affect the performance and longevity of the battery, as well as enabling creation of the sufficient SEI layer, further explained in section 2.1.2. The compatibility depends on a combination of properties stated above [24].

These properties play a crucial role in determining the performance and longevity of an electrolyte and optimizing them is highly preferable and desired when developing electrolyte descriptors [19, 20, 22, 27, 28]. Moreover, to analyze the properties of an electrolyte, the battery behavior needs to be studied to understand what the electrolyte properties are influenced by. Therefore, battery performance testing is essential.

2.1.2 Battery Performance Testing

Traditional battery testing involves key parameters such as *C-rates*, *Depth of Discharge (DoD)*, *temperatures* and the use of electrochemical techniques such as *Cyclic Voltammetry (CV)* to study battery behavior within different cell chemistries. These parameters are important to understand the lifespan and characteristics of different batteries. When the performance of a battery is analyzed, cycle life is often an important aspect. The cycle life represents the number of times a battery can be charged and discharged over the lifetime of the battery. It should be noted that the cycle life of a battery is highly dependent on the DoD [29].

DoD is a term commonly used in battery testing and refers to the charge that has been removed from the battery electrode during discharge. More specifically, the capacity removed from a battery. It is typically expressed as a percentage of the battery's total capacity. For example, if a 100 mAh battery is discharged at a current of 50 mA for 20 minutes the DoD will be:

$$\text{DoD} = \frac{50\text{mA} \cdot 20\text{min}}{100\text{mA} \cdot 60\text{min}} \approx 16.7\%$$

DoD is an important parameter in battery testing because it directly affects the battery's overall lifespan and performance [30]. In general, deeper discharge cycles can reduce the battery's overall lifespan and performance, while shallower discharge cycles can help to extend the battery's lifespan and performance [30]. Therefore, understanding the DoD of a battery is critical when designing battery systems and performing battery testing. By measuring the DoD of a battery or electrolyte, one can evaluate its performance and optimize its use for specific applications.

Another important parameter when discussing battery performance testing is the *C-rate*. The C-rate is a measure of the rate at which a battery is charged or discharged relative to its capacity. Specifically, it refers to the rate at which a battery is charged or discharged relative to its rated capacity, which is typically measured in ampere-hours (Ah) [31]. For example, if the capacity is 100 Ah and it is charged or discharged at a C-rate of 1C, then it is being charged or discharged at a rate of 100 A. If it is charged or discharged at a C-rate of 2C, then it is being charged or discharged at a rate of 200 A. The C-rate has an impact on the cycle life of the battery cell and the lifespan typically decreases when using a higher C-rate [32]. If the battery will be used in a power tool or a vehicle motor, the C-rate will be high automatically since these are draining electricity at a higher rate. However, vehicles often have a power management system that limits the amount of received power in order to avoid rapid degradation of the battery [32].

Temperature is an additional parameter that can affect the performance of a battery. A direct consequence of higher temperature is the higher chemical reaction rate in the battery [33] which could lead to increased capacity loss. On the other hand, when the ambient temperature is low, the capacity retention of a battery decreases. This occurs since the ionic conductivity of the lithium salt electrolytes declines [34] and therefore the overall performance of the battery cell as well. Therefore, the optimal range for a LIB to operate with good performance is generally $15 - 35°C$ [35].

This is an important factor when measuring the performance of the battery and could lead to a lower lifespan of the LIB.

CV [36] is a useful technique for analyzing the performance of electrolytes in battery cells. In CV, the potential of an electrode is swept linearly and the resulting current is measured as a function of potential. CV can be used to study the ion transport properties of the battery cell, as well as its ESW. It can also be used to evaluate the electrochemical behavior of the electrolyte, such as its redox behavior and can provide information about the rate and mechanism of electrochemical reactions.

A further aspect that is considered when testing the cyclic performance of batteries is the cutoff voltages. The cutoff voltages are the specified lower and upper voltages of the battery during cycling [37]. The battery test is generally conducted with an upper cutoff voltage that avoids overcharging [38], which could have undesirable consequences for the battery cell. These cutoff voltages will be referred to as *voltage range* in this thesis.
When testing the performance of batteries and specifically electrolytes, the usual method is to cycle the batteries for a certain number of cycles or until failure. The point of failure could be described as its end-of-life (EoL), which is when the capacity has faded and the battery is no longer considered to perform acceptably for its area of use. EoL varies depending on battery, where for a LIB it is usually around 70% of its initial capacity [39]. During battery performance testing, equilibration with higher ESW is performed to form a protective SEI on the electrodes. The SEI is a thin, solid layer that acts as a protective film between the electrode and the electrolyte with the primary role to stabilize the interface between the electrode and the electrolyte. The SEI layer forms as a result of electrochemical reactions between the electrolyte and the electrode material. This layer is crucial since it helps prevent further reactions between the electrolyte and the electrode, reducing degradation of the battery's performance [40].

The parameters described in this section are vital and the conditions of these affect the lifespan of the battery when tested. There are many studies where the goal is to cycle the batteries for a set number of cycles using varying values for DoD, C-rate, temperature and with different electrolyte components [41–50]. An illustration of how results from a cycling test can look is presented in Figure 2.2. This is usually how the results are presented when testing different electrolytes and analyzing their capacity. In most studies, salt, solvent, electrodes, C-rates, DoD, temperature and voltage range are presented along with the results.

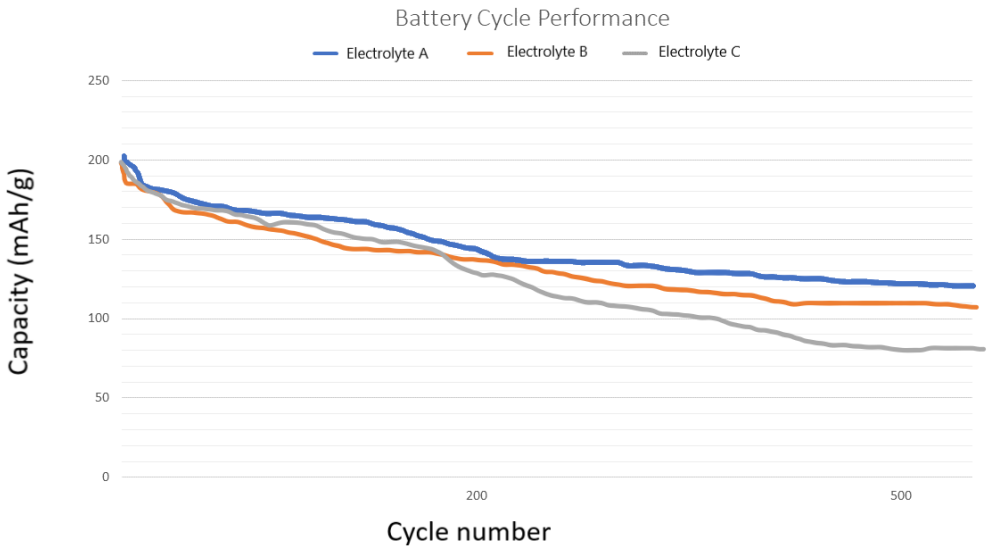


Figure 2.2: Illustration of capacity fade compared to the number of cycles for three different electrolyte solvents.

2.2 Molecular Dynamics

MD is a computational method used to study the behavior of molecules and their interactions over time. It simulates the motion of atoms and molecules using classical mechanics and statistical thermodynamics, providing insight into the physical and chemical properties of materials at the molecular level. MD simulations can be used to investigate a wide range of phenomena, from the behavior of simple gases and liquids to the folding of proteins and the dynamics of chemical reactions. They are widely used to predict the properties and behavior of molecules and materials under different conditions and to design new materials with specific properties [51–53]. The force calculations in MD are calculated for each atom based on Newton´s laws of motion as in (2.1).

$$\mathbf{a} = \frac{\mathbf{F}}{m} \tag{2.1}$$

There are many ways of simulating the steps and movement of the atoms, with different accuracies, cost functions and stabilities [22]. Here we perform MD using the force acting on each atom, one can then predict the spatial position of each atom as a function of time. Within a simulation, the acting force calculated from (2.1) is repeatedly calculated each timestep Δt and then used to update the position and velocity of each atom as in Figure 2.3.

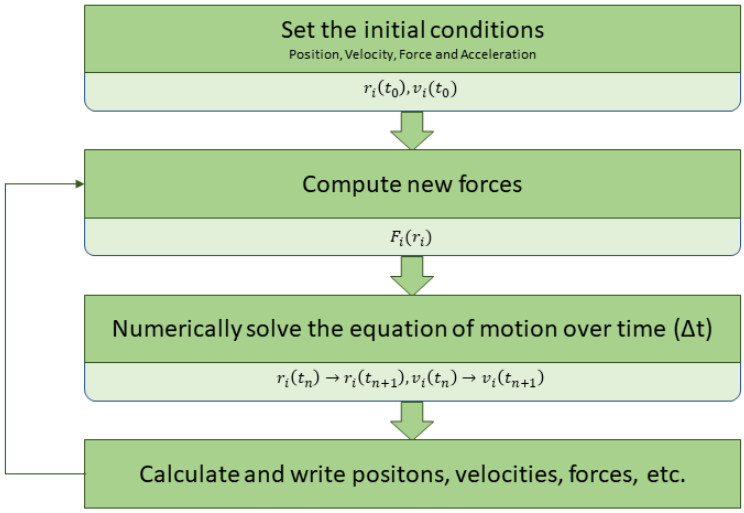


Figure 2.3: MD simulation schematic.

MD simulation possesses substantial capabilities from several aspects. Firstly, it computes the spatial position and motion of every atom at each moment in time, information that is difficult to obtain through experimental techniques. Secondly, MD is highly controllable and applicable to specific dynamics, which further enhances its utility and makes it a highly controllable and effective way of analyzing MD [54]. Within this thesis, MD is achievable through the utilization of the *CP2K* model *GFN-xTB* [55], a software that encompasses quantum chemistry and condensed matter physics. This program provides a general framework for various modeling approaches program and is capable of computing atomistic simulations of diverse systems such as solid-state, liquid, molecular, crystal, and biological systems [56].

Moreover, when applying MD for the analysis of electrolytes, the molecular structure needs to be structurally represented with all complex inherent information. Consequently, the data from MD would need to be represented using graphs.

2.3 Graph Theory

Graph theory is a branch of mathematics that deals with the study of graphs, which are structures consisting of a set of *vertices* (also called nodes) and *edges* (the links) connecting pairs of vertices. One of the key concepts in graph theory is the degree of a vertex, which is the number of edges that connect to it. Graphs can be classified based on their degree sequence, which is the list of degrees of the vertices in the graph. For example, a graph in which all vertices have the same degree is called a *regular graph*.

Another important concept in graph theory is connectivity, which refers to how easily information can flow between vertices in a graph. A graph is said to be con-

nected if there is a path between any two vertices. If a graph can be split into two or more disconnected components, it is called a *disconnected graph* [57, 58]. The connections within a graph could be obtained by creating a matrix with *one-hot encoding* consisting of 0's and 1's which represents positive or negative connection.

Moreover, when handling with *subgraphs*, for instance, a molecule in a molecular compound, this could be encoded into vector space with *graph embedding*, which is a way to represent the connections of all subgraphs when encoding it for network usage. Graph embedding is a technique used in graph analysis to transform graphs into a vector or a set of vectors. The goal of graph embedding is to capture the structural information of the graph [59]. One way of performing graph embedding of subgraphs practically is to use *PyTorch geometric* [60] which enables encoding of subgraphs into a network like a GNN which is explained later in section 2.4.1. Such graphs can be represented using *NetworkX* [61], which is a Python module that enables the creation, manipulation and study of the structure, dynamics and functions of complex networks. When handling graphs, NetworkX provides useful functions that add nodes and edges from the simulated input data, to create a structured graph representation.

Graph theory prevents molecular representation from losing information about the connections between molecules and atoms. The significance lies in the necessity to uphold and analyze the inherent associations existing between atoms, molecules and molecular compounds. Molecular graph representation can be accomplished from MD simulation data using Chalmers hierarchical atomic, molecular, polymeric, and ionic analysis toolkit (CHAMPION) [5, 22], as presented in Figure 2.4.

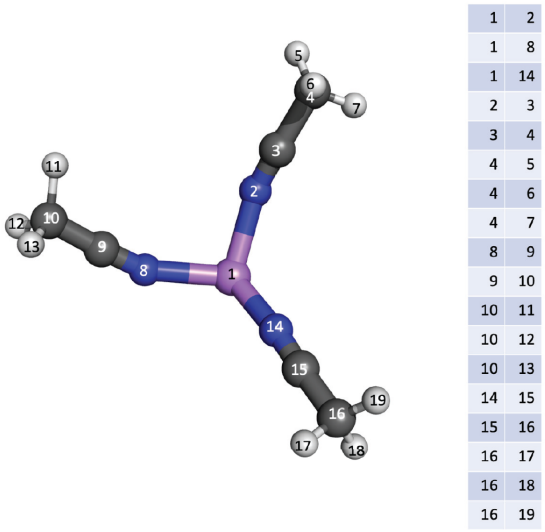


Figure 2.4: Graph representation of nodes and their respective connections, showing the connection rules of a CHAMPION output.

2.4 Artificial Neural Networks

An ANN is an ML method with the purpose of simulating complex perceptual and cognitive decisions of the human brain [62, 63]. Similar to a brain, the network consists of *neurons* that propagate, process and transmit information into a new *layer* of neurons. The neurons are organized into layers where each layer receives input from the previous layer and propagate output to the next layer. An ANN can consist of multiple layers of neurons, called *hidden layers*. All neurons within each layer receive input and transmit an output from the neurons in closeby layers, defined as a *fully connected layer* which could be seen in Figure 2.5. The input is multiplied with an individual numerical *weight* which determines the strength of influence the connections have on each other. These are updated and adjusted for optimization during a training process of the network. The weights are then summarized, with a *bias* term added. The bias is added to the weighted sum of the inputs before the activation function is applied. The bias term allows the neuron to have some influence on its output even if all of its inputs are zero [64]. Finally, the output is propagated through a non-linear activation function which tells the network whether each neuron should be activated or not, as illustrated in Figure 2.6.

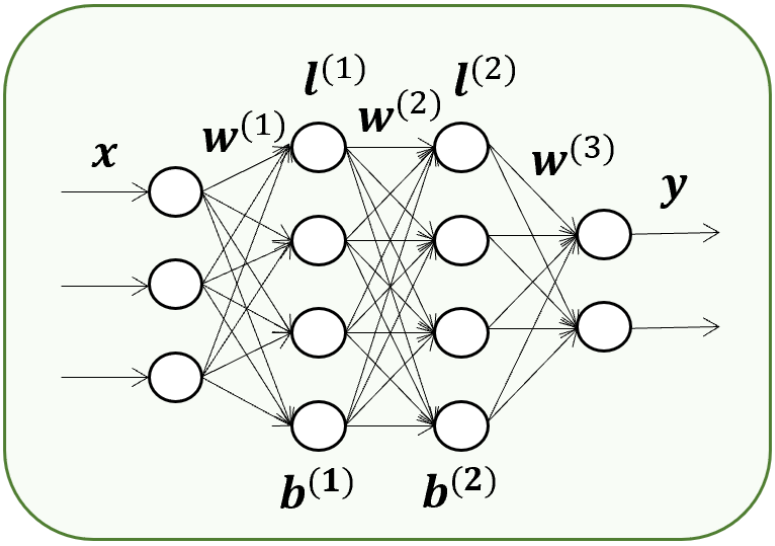


Figure 2.5: The architecture of a simple four-layered fully connected ANN with three input neurons $\mathbf{x}$, weights $\boldsymbol{w}^{(n)}$, two hidden layers $\boldsymbol{l}^{(n)}$ of four neurons with corresponding biases $\boldsymbol{b}^{(n)}$ and two output neurons $\mathbf{y}$.

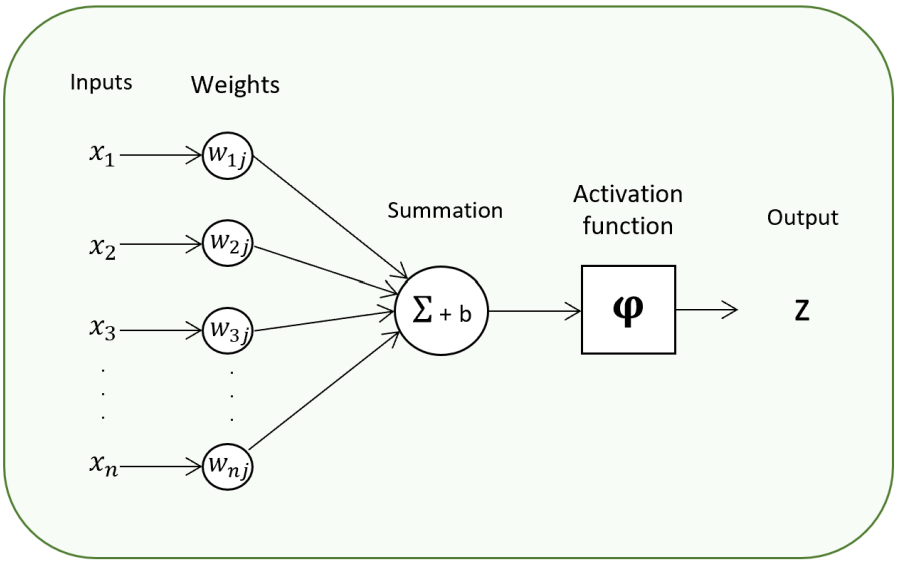


Figure 2.6: Schematic overview of the operations performed from input to output within a neuron in ANN. Input values x_n is multiplied by weights w_{nj}, summarized with an added bias b and then multiplied by an activation function φ.

The purpose of the activation function is to introduce non-linearity into the network, which allows it to model complex relationships between inputs and outputs. Without this, the ANN would be more in the range of a linear regression model. There are many different types of activation functions, such as the sigmoid function, the rectified linear unit (ReLU) function and the tanh function, each with its own strengths and weaknesses [64]. Within this thesis, ReLU will be considered.

Altogether, given an activation function φ, input values x, weights w and bias b, we can calculate the output value of a neuron z as seen in Figure 2.6 as follows:

$$z = \varphi\Big(\sum W_j x_j + b\Big) \tag{2.2}$$

When determining the prediction of the model, we calculate the correctness with *loss functions*. The loss function could be calculated differently, based on the desired way to quantify the difference between the ground truth $\hat{y}_j$ and the output of the network $\mathbf{y}$. A usual loss function L is Mean Squared Error (MSE):

$$L = \frac{1}{2}\sum(\hat{y}_i - y_i)^2 \tag{2.3}$$

For an ANN, the *weights* and *biases* of each neuron are randomly initialized and then updated for each iteration within a training process. This process is called *backpropagation*, with the purpose of converging toward a local minimum of the loss function and indicating how much the network should change its internal parameters. Backpropagation calculates the gradient (derivative) of the loss function associated with a given state with respect to the weights. A common way of performing backpropagation is using a method called *stochastic gradient descent*, which uses a small update rule of the weights expressed as follows [65]:

$$\delta W_{mn}^{(k)} = -\alpha \frac{\partial L}{\partial W_{mn}^{(k)}} \tag{2.4}$$

Where the constant $\alpha > 0$ is a predefined learning rate. The partial derivative of the loss function L with respect to the weight connecting the m^{th} neuron in the hidden layer k to the n^{th} neuron in the output layer, denoted as $W^{(3)}_{mn}$, we get the weight update chain rule for this output weight as:

$$\frac{\partial L}{\partial W^{(3)}_{mn}} = -\sum(\hat{y}_i - y_i)\frac{\partial y_i}{\partial W^{(k)}_{mn}} \tag{2.5}$$

Where the next layer is added to the chain rule:

$$\frac{\partial y_i}{\partial W^{(3)}_{mn}} = \frac{\partial}{\partial W^{(3)}_{mn}}\varphi\Big(\sum W^{(3)}_{ij} z^{(2)}_j + b_i\Big) \tag{2.6}$$

Which is equal to:

$$\frac{\partial y_i}{\partial W^{(3)}_{mn}} = \varphi'\Big(\sum W^{(3)}_{ij} z^{(2)}_j + b_i\Big)\delta_{im} z^{(2)}_n \tag{2.7}$$

Where δ_{im} is the Kronecker delta, which is 1 when i = m and zero otherwise. This process of updating the weight works the same way for the bias:

$$\delta b^{(k)}_m = -\alpha\frac{\partial L}{\partial b^{(k)}_m} \tag{2.8}$$

Which gives for b:

$$\frac{\partial L}{\partial b^{(2)}_m} = (\hat{y_m} - y_m)\varphi'\Big(\sum W^{(3)}_{ij} z^{(2)}_j + b_i\Big) \tag{2.9}$$

2.4.1 Graph Neural Networks

A GNN is a deep learning-based method used to operate on graphs described in section 2.3. A GNN can be used for many purposes and areas such as social networks [66], physical systems [67] and knowledge graphs [68]. Before creating a GNN, it is important to detect the graph structure for utilization. There are two different types of graph structures, non-structural and structural. When the graphs are implicit, it is a non-structural scenario and the task will be to build the graphs and then design the GNN to fit to this graph structure. When the scenario is structural, the graphs are explicit and already suited as input for the GNN [69].

When dealing with tasks on a graph level there are usually three types of tasks, node-level tasks, edge-level tasks and graph-level tasks. These tasks usually refer to classification where the goal is to categorize nodes, edges or graphs into different classes or regression where the aim is to predict a continuous value for a node, edge or graph [69]. The GNN is then designed to fit the specific need of the problem and there are three computational functions that can be used in a GNN, propagation, sampling and pooling. The purpose of propagation is to spread information between the nodes, usually by convolution or recurrence. This enables the network to capture relations between nodes and their features. Sampling is often used with the propagation when dealing with large graphs. The pooling operation is applied when high-level graph features need to be extracted using the node features [69].

A typical GNN architecture with the features described in this section is presented in Figure 2.7.

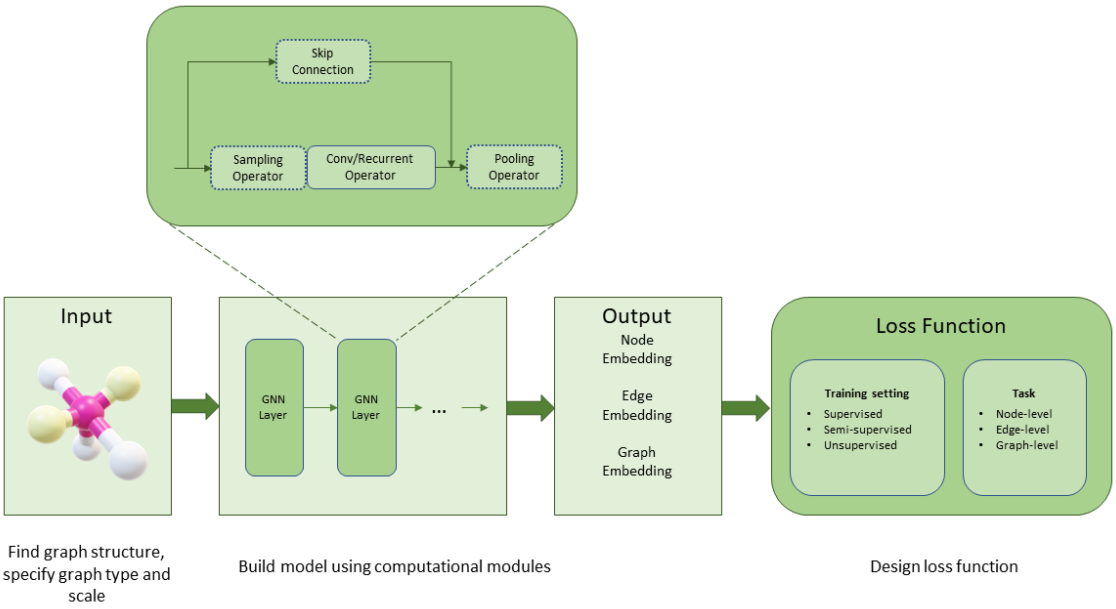


Figure 2.7: Typical GNN architecture design.

Graph convolutions are used in *graph convolutional networks (GCNs)* to propagate information between nodes similar to *convolutional neural networks* (CNNs) do with pixels in an image [70]. Convolutions are performed in order to learn feature information about nodes that are far away. By adding more convolutional layers, more information is propagated and the depth enables the receptive field of each node to grow [71]. In Figure 2.8, an illustration of how graph convolutions propagate features in a graph is presented. In this example, two convolutions are performed in order to propagate features from nodes in the neighborhood of node A.

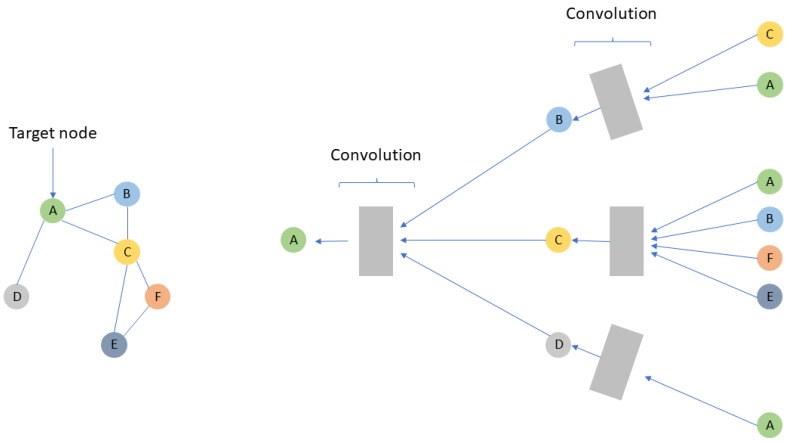


Figure 2.8: Example of how graph convolutions propagate features to target node A.

2.4.2 Descriptors

When evaluating the performance of an ML model, the data representation is essential. For example, when using compounds as data, the representations of these are called *descriptors* [72]. The main aim is to select good descriptors that fit the ML model properly. In essence, creating descriptors that include data properties that have a correlation with the target property [72]. This descriptor development coincides with the primary objective of this thesis, as mentioned in section 1.2.

When handling complex data such as databases that contain information about chemical compounds, the data may not fit directly as input in a neural network. Therefore, some kind of feature extraction needs to be performed in order to reduce the dimensionality and create representative descriptors [73]. In other words, the data needs to be processed and transformed in some way. The transformation of data features is often referred to as feature engineering, which is the method of transforming the input data to suitable descriptors for an ML model. The ability to fit the descriptors to different ML models could also be important. Therefore, flexible descriptors that can be decoupled from the model are preferred, since some descriptors need to be altered in order to fit the specific model [74].

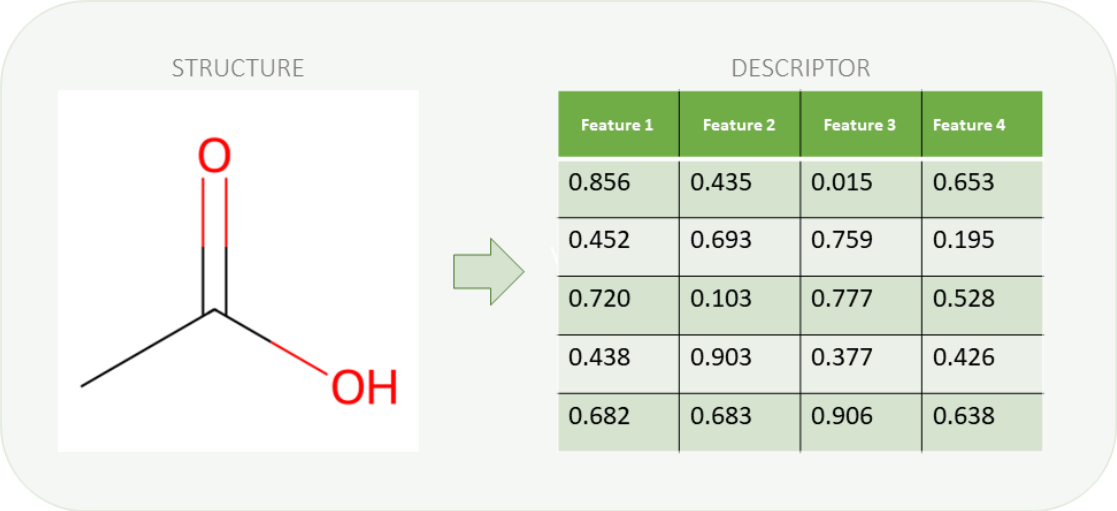


Figure 2.9: An example of how a simple descriptor of four features describes the structure of a given input. In this example, each row of the descriptor is an atom in the molecule. All four features are numerically represented for each atom in this case.

3
Methodology

Herein we aim to provide a comprehensive overview of the approach taken to conduct the research and achieve the objectives outlined in the introduction. This chapter describes in detail the procedures and techniques used in each of the four parts of the project, namely data collection, simulation, descriptor development and model development.

3.1 Data Collection

In order to develop electrolyte descriptors for predicting cycling performance, an investigation was conducted involving the examination of open-source work and journals for data collection and annotation [41–50]. Unfortunately, the open-source battery databases did not reveal the necessary data about cyclic performance testing. However, the journals selected for analysis contained comprehensive results of essential parameters and measurements obtained from classical battery performance testing as described in 2.1.2. The journals were numerically and graphically analyzed and annotations were made accordingly.

The annotation process involved conducting keyword searches for relevant terms such as long-term battery cycle testing, cycling performance, CV, ESW, cycle capacity, C-rates, DoD, voltage range and temperature in conjunction with battery cycling testing that shared similar testing criteria and setup. Once battery testing with appropriate parameters was identified, the primary focus was to locate how the specific discharge capacity declined over the number of life cycles, as illustrated in Figure 3.1. Whenever this data was given, combined with specific data of electrodes surrounding electrolytes, the ratios of the specific electrolyte salt and solvent used, the C-rates, voltage range, and long-term cycle number the electrolyte data was annotated in an Excel file system. In specific cases where the explicit capacity decline was not provided, a thorough analysis and interpretation of the graph was conducted.

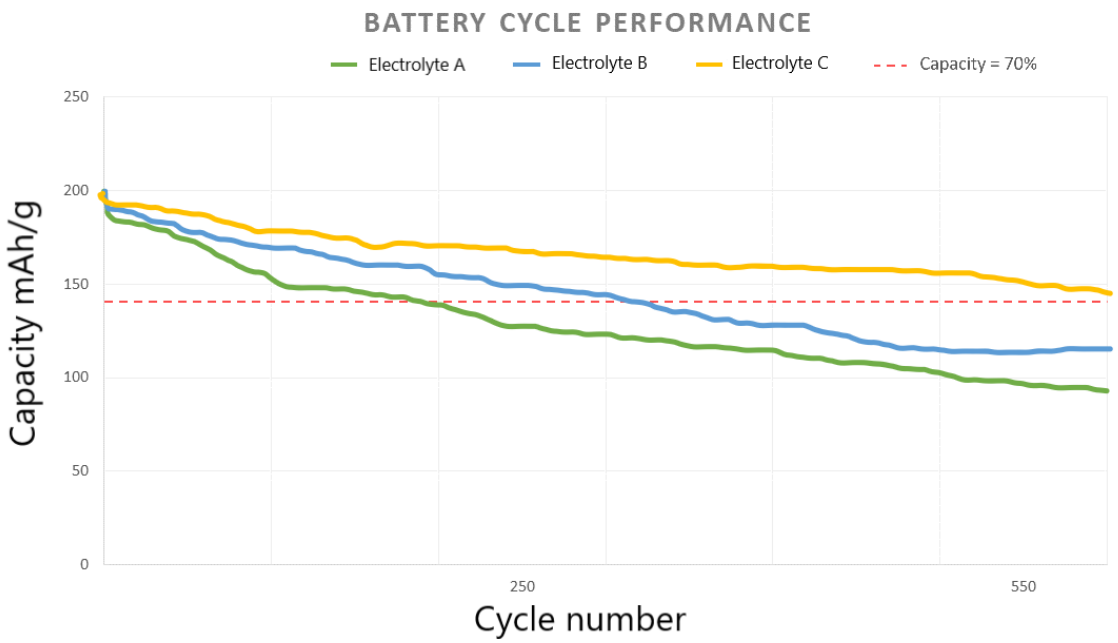


Figure 3.1: Illustration of capacity fade compared to the number of cycles for three different electrolytes, including a line for battery failure (70% capacity). The initial drop in capacity is due to equilibration cycles, performed to create the SEI.

Hence, for each unique electrolyte tested, the following measurements, ratios and essential parameters used were annotated:

- Salt
- Solvent
- Molar ratio of salt and solvent
- Electrodes
- Temperature
- C-rate
- Voltage range
- Cycles
- Capacity

All cycling tests annotated were performed with 100% DoD. After annotating a sufficient amount of data for initial testing, all annotated electrolyte data needed to be estimated on the same dimension for a reasonable comparison, since most electrolytes were cycled differently.

3.1.1 Target Value Estimation

The point where the batteries reach 70% of their initial capacity is the point where a battery generally has reached its EoL. Hence, the point of 70% capacity is estimated as a general target value for all annotated data, to achieve an equal dimension for the target value (label) in a neural network.

The target value *Cycles To 70%* is calculated from *Cycles* and *Capacity* from the obtained data. Based on common testing equilibration to form SEI at the beginning of the testing as described in 2.1.2, as seen from the quick drop in capacity at the start of the curves in Figure 3.1, the graphs needed to be interpreted from how the curves converge after the equilibration cycles. All curves were therefore graphically interpreted and annotated with an individual gradient factor. This gradient factor is the slope of the cycling graph at the end of the test and is used to calculate the number of cycles it takes for a test to fade down to 70% capacity. *Cycles to 70%* was then used as a label for cycling prediction.

3.2 Simulation

The simulation step was performed in order to gather further information about the electrolytes. This includes pre-processing of the electrolyte data collected, simulation in CP2K as well as data analysis using CHAMPION. The last step was necessary since the descriptors developed in this thesis are based on the simulated and analyzed electrolytes using Compular's technique.

3.2.1 Simulation Pre-processing

In order to simulate the electrolytes gathered from the performance tests, certain pre-processing steps were required. The components of the electrolyte had to be annotated, the molar ratio of the electrolyte needed to be calculated and a simulation geometry was necessary.

3.2.1.1 Electrolyte Components Annotation

The data collected in 3.1 were additionally annotated, since molecular information was necessary for the electrolyte simulations. First of all, the electrolyte that would be simulated included a set of molecules. These molecules were compiled in a Python file with all components being dictionaries with essential attributes. These attributes included name, atom type, number of atoms, density as well as molecular mass. In Figure 3.2, a Python dictionary representation of EC is presented.

```
EC = {
    "Name": "EC",
    "Elements": ["C", "H", "O"],
    "NAtoms": [3, 4, 3],
    "rho": 1.32,
    "mol_weight": 88.06
}
```

Figure 3.2: Representation of EC in a Python dictionary.

Using this Python file as a database for molecular components in the electrolytes chosen for simulation, the molar ratio for the specific electrolyte could be computed.

3.2.1.2 Molar Ratio Calculation

Using the electrolyte component information, the molar ratio of each electrolyte could be determined. Since the different solvents in the electrolytes were specified either by weight, mole per liter or weight percentage, the molar ratio was calculated depending on what was presented in the data collected. The goal of this calculation was to determine the number of molecules for each component of the electrolyte while the number of total atoms was between 800 and 1200. The reason for the constraint on the number of atoms was that the simulation of molecular dynamics is complex and computationally expensive. Therefore, simulating over 1200 atoms would take more than 3 days which was not possible in this project.

Since the calculations of the molar ratio were different for each electrolyte, a number of methods needed to be implemented in Python. The main method calculates the number of atoms which should be around 1000, based on this the number of molecules of each component is calculated. Another calculates the solvent ratios based on the molecular weight of the salt and the solvent which is needed if the ratio of the electrolyte solvent is defined but the salt is defined in moles/liter. The two last functions handle electrolyte solvents that are defined in weights or weight percentages. The first converts weight ratios for the solvent into molar ratios to fit the ratio to the salt while the second adds solvent components based on weight percentage.

These methods were used to compute the amount of substance for each component in the selected electrolyte. Subsequently, this information was used to create the geometry in CHAMPION, which was the last pre-processing step before the molecular dynamics simulation.

3.2.1.3 Geometry Creation Using CHAMPION

In order to simulate the molecular dynamics, the starting geometry of the electrolytes was necessary. This is essential when simulating in CP2K since the initial geometry coordinates need to be defined when using it. Using the amount of substance of each component of the electrolyte, calculated as described in section 3.2.1.2, the geometry was created using CHAMPION. The created geometry defines the position and rotation of all molecular species at the initial step which is used as input to CP2K. An example of how the geometry looks before a simulation is presented in Figure 3.3.

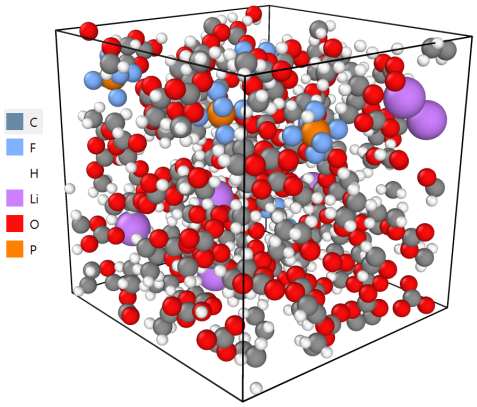


Figure 3.3: An illustration of the created starting geometry using CHAMPION.

3.2.2 Molecular Dynamics Simulation

With a prepared initial geometry setup of correct molar ratios and the number of atoms, the MD simulations of high-performance computing (HPC) could be started. The simulation was set up with approximately five electrolytes on parallel computing nodes using Amazon Web Services (AWS).

The simulation was set up with 60,000 timesteps Δt where one step simulates 1 femtosecond for each electrolyte. The forces acting on the atoms were simulated as described in section 2.2 using CP2K. The final electrolyte structure output of the molecular dynamics from approximately 1000 atoms can be seen in Figure 3.4. After simulation, the structure becomes more ordered and reflects the interactions between different species in the system, resulting in a clearer arrangement of closest neighbors. In this thesis, structures that consist of more than one molecule are referred to as molecular compounds.

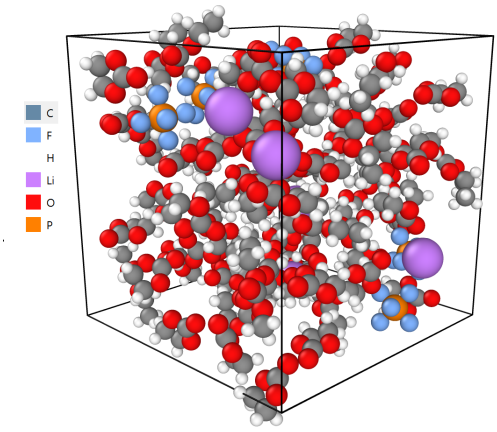


Figure 3.4: An illustration of the geometry after an electrolyte simulation using CP2K, final timestep. New molecular compounds have been shaped from molecules and atoms.

3.2.3 Analysis Using CHAMPION

From the simulated electrolyte structure, a configuration file for CHAMPION analysis was set up. The *CHAMPION* simulation was run on each electrolyte to obtain essential properties and characteristics of electrolytes in batteries, as well as structural graph data of connections with nodes and vertices. The output of the analysis was given as a database to be analyzed in *SQLite* for descriptor development. Each molecular compound in the produced database contains a graph number. The database also includes a list of edges between all molecules within the specific molecular compound and the corresponding parent graph. In the database, a list of nodes corresponding to a specific graph is present. These are the nodes of the graph that corresponds to the specific molecular compound. With this information, the development of structural descriptors for each graph within the electrolyte could be initiated.

3.3 Descriptor Development

The development of the descriptors was the most important process of this thesis. This includes extracting descriptive information about the simulated and analyzed electrolytes that could be used in an ML model. To produce detailed descriptors for the electrolytes, some processing steps needed to be done. These steps include creating node embeddings, adding nodes, edges, graph measurements and graph attributes.

3.3.1 Node Embedding

As described in section 3.2.3, the molecular compounds in the produced database represent graphs with molecules as nodes. Since all molecular species of each graph are specified by name in the database, they needed to be embedded in some way, in this case by using one-hot encoding. The first step to perform one-hot encoding was to list all molecular species present in all electrolytes that were simulated. An example of a list of molecules is presented below.

$$[\text{"}Li^{+}\text{"}, \quad \text{"}EMC\text{"}, \quad \text{"}EC\text{"}, \quad \text{"}PC\text{"}, \quad \text{"}PF6^{-}\text{"}, \quad \text{"}FMES\text{"}, \quad \text{"}VC\text{"}]$$

When the embedding of each node of a graph was performed, all nodes were compared to a molecular list similar to the one above and one-hot encodings were created. The one-hot encoding for a node that is EC would therefore look like this:

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

This way, each molecule (node) of each graph gets a vector representation instead of a name, which is useful since neural networks handle numerical values rather than categorical text strings.

3.3.2 Graph Construction

When the node embedding was completed, the graphs consisting of these nodes were constructed. In order to construct the graphs, the databases generated from the analysis were used. As described in section 3.2.3, every molecular compound in the electrolyte has a corresponding graph number which consists of all edge numbers and node numbers representing the graph. The interrelated nodes and edges could therefore be extracted from the database accordingly.

The first step in the process of constructing the graphs was to create empty graphs, this was done using the *NetworkX* module in Python which is described in section 2.3. The new class that was created in this project is called *MolGraph* and it inherits all features from a NetworkX graph. It also contains additional methods that were added. The first additional method is the ability to add the nodes and edges to the graph from the corresponding database. This method is essential in order to structure the molecular compound graph with their molecules as nodes and the connections between them as edges. Additionally, this method generates corresponding one-hot encoding for each molecule in the graph, this is performed as described in section 3.3.1.

3.3.3 Graph Feature Extraction

Following graph construction, the features of the graph were extracted from the electrolyte analysis database. This is necessary since all molecular compounds in the electrolyte have features that affect the behavior and performance of the electrolyte. The class MolGraph possesses supplementary methods that add features to the graph. These methods extract attributes such as *charge*, *mass*, *diffusivity*, *concentration*, *molar conductivity*, *electrical mobility* and *ionic conductivity* for each molecular compound in the electrolyte and append these as graph features. To prevent irrelevant information from the database, graphs that are not present in the final simulation step are excluded from the electrolyte representation. In essence, the concentration of these graphs is zero in the database.

This procedure was performed on each graph in the electrolyte and the resulting graph objects acquired the necessary features to describe a molecular compound. Subsequently, the graphs needed to be assembled to represent the whole electrolyte, which was done using graph batching.

3.3.4 Graph Batching

Since the graphs constructed were not assembled, they needed to be aggregated into a suiting object that represents an electrolyte and is applicable as input to an ML model. Therefore, the *PyTorch Geometric* batch object was used. This object type aggregates graphs into a batch that functions as a large graph. Therefore, a class called *Electrolyte* was created that inherits the features of a batch object. This electrolyte object represents all graphs in the electrolyte and includes the corresponding graph features for each graph within the electrolyte. This way, *graph embedding*

such as *GNN* could be used with the electrolyte as input. Furthermore, additional methods were added to the electrolyte class. These include a method for plotting the electrolyte, a method for plotting a specific molecular compound from an electrolyte and a print method that produces graph feature values from all molecular compounds in an electrolyte. The methods are flexible in the aspect of what molecular compound should be plotted or printed as well as what graph features to visualize.

The resulting electrolyte objects, therefore, function as descriptors for the electrolyte representations from the database. This allows the information about the simulated and analyzed electrolytes to be concretized and the descriptors to function as input to an ML model. Additionally, the electrolytes can be analyzed further after processing and specific aspects of its features can be inspected in detail. An illustration of what the electrolyte descriptor is representing is presented in Figure 3.5.

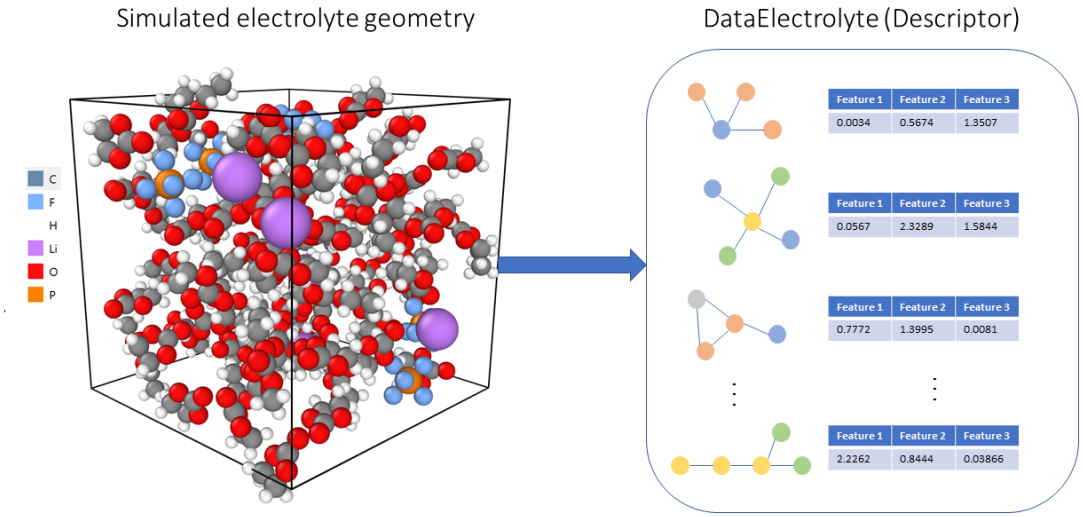


Figure 3.5: Illustration of what the electrolyte descriptors are representing. The features in this example are arbitrary.

3.4 Model Development

The ML model was developed as a limited initial version outlined to confirm the predictive capabilities of the descriptors, rather than predicting cycling performance accurately. In this section, the model development is described step by step including its parameters, operations, architecture and lastly the training with the evaluation procedure.

3.4.1 Model Design

The model receives electrolyte descriptors input as *PyTorch Geometric* data batch objects, *DataElectrolyte* objects. As described in Section 3.3.4, these descriptors consist of molecular compounds presented as batched graphs with connected graph features. The model's objective is to find patterns between the node connections within the graphs, followed by learning from the graph-level features. To obtain this the network could be divided into two parts, consisting of a GNN with *message-passing* convolutions and a *multi-layer perceptron* (MLP) neural network with prediction.

The first part of the model consists of a GNN with message passing. The electrolyte descriptor inputs were transformed into *node-embedded graphs*, where each node embedding represents a molecule and its connections. Message passing on between the nodes was implemented with two layers of graph convolutions, followed by pooling all nodes into weighted graph values. The pooled graph values were then concatenated with each respective graph feature for the MLP and prediction. This part of the model and its schematic overview is illustrated in Figure 3.6.

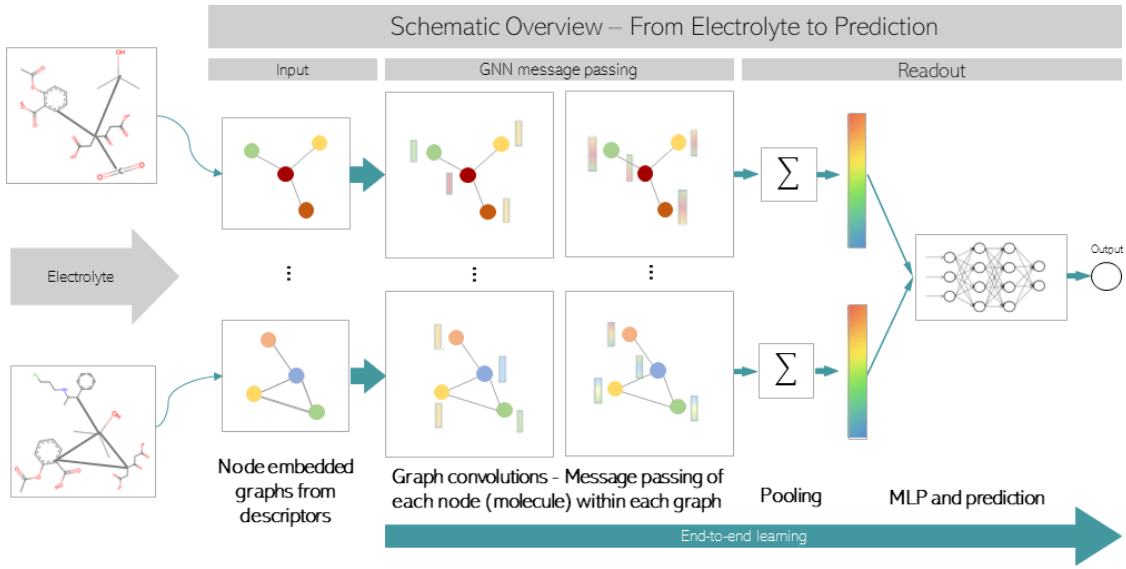


Figure 3.6: Schematic overview of the GNN purpose of the network. Electrolyte descriptors as DataElectrolyte objects are sent into the network, where each molecular compound becomes a graph with each molecule as embedded nodes. The node-embedded graphs are sent into graph convolutions, that perform two layers of message passing within each graph, followed by pooling and summarizing each graph into a weighted value.

The second part of the model consists of a MLP, with linear *fully connected layers*. Graph features were concatenated with pooled graph values, followed by sorting the graphs by the molecular compounds of the highest concentration. Out of these sorted values, P graphs were *padded* to the same size to be used in the MLP. The following layer is a *graph-wise fully connected layer* that learns from the graph feature and outputs fully connected values. These are sent into several fully connected layers for

deeper learning with *ReLU* activation function utilized after each layer. Lastly, the model predicts the cycle performance by comparing the predicted value against the true target value *Cycles To 70%*, which is normalized around zero. The prediction is evaluated with *MSE* loss. This choice of the loss function is usual in regression tasks such as true value prediction since the goal is to minimize the error between the target value and the predicted value. The complete network architecture from input to prediction, including the network operations and dimensions, is described in figure 3.7.

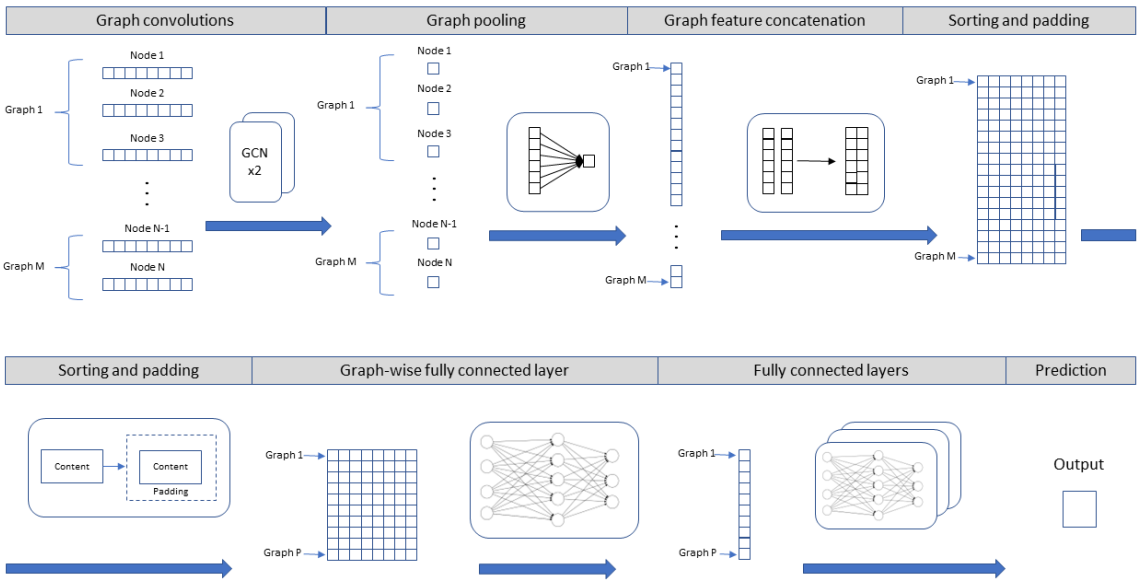


Figure 3.7: Illustration of the full network architecture from developed electrolyte descriptor input to predicted cycle performance output. The network is divided into graph convolutions, graph pooling, graph feature concatenation, sorting and padding, graph-wise fully connected layer, fully connected layers and prediction. Input vector and matrix sizes are shown in between all network operations.

To obtain some indication of predictable descriptors from this initial version, the data were divided into training and validation sets. Additionally, the Adam optimizer, MSE loss and a low learning rate were utilized considering that the amount of data is limited. The training was performed until the network converged or found a minimum, and compared to the validation loss. Hyperparameter tuning, evaluation metrics and visual interpretation were done to evaluate the results and analyze if the descriptors were sufficient as input for the model.

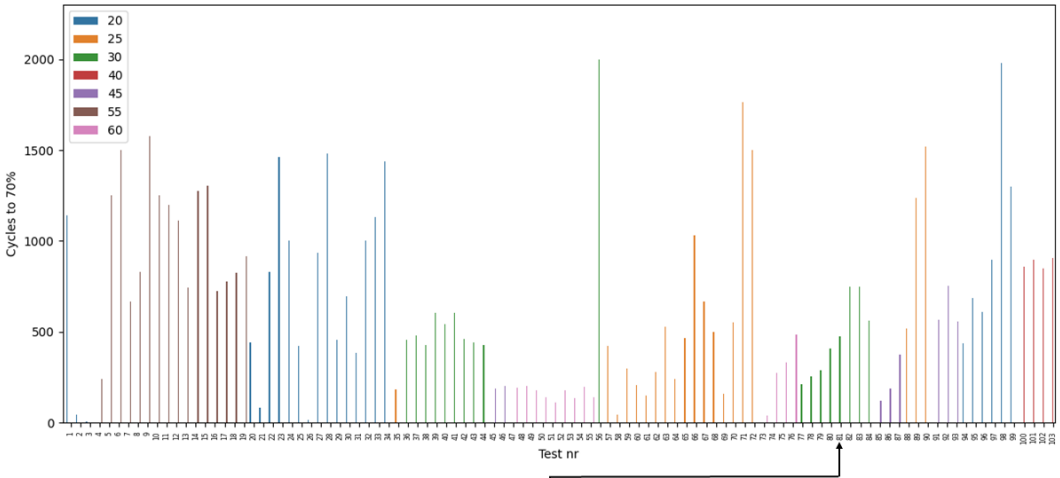
4

Results & Discussion

This section offers a comprehensive analysis of the research efforts and presents an in-depth interpretation of the obtained results. The annotated data, the developed electrolyte descriptors and the results from the GNN are presented, emphasizing the complexity of the electrolyte descriptors and their predictive abilities. The methodologies employed are evaluated, limitations are acknowledged and the broader implications of electrolyte descriptors for cycle prediction are discussed extensively.

4.1 Annotated Data

The collected data consisted of 104 battery cell cycle performance tests. These tests involved various electrolyte compositions, electrodes, temperature, C-rates and voltage ranges. The cycle performance of each test was annotated and the target value was estimated with *Cycles To 70%* as in section 3.1.1. The resulting target values are presented in Figure 4.1.



Test nr	Electrolyte	Solvent (%=wt)	Electrodes	Temperature [°C]	Cycles [n]	Capacity [%]	Cycles to 70%	C-Rate [C]	Voltage range[V]
81	LiFSI	DMC/OTE 0.51:0.84:0.84 (mol), 5% FEC, 1%VC	Graphite/Si | (90% NMC532,5% C45, 5% 5130 PVDF)	30	300	78	409	1/3	3-4.1

Figure 4.1: Cycle performance of 104 annotated battery cycle tests, with *Cycles to 70%* capacity of each electrolyte test. The temperature used within the test is also represented. The relevant data from test 81 is presented in the bottom part of the figure.

The annotated data is considered profoundly important in developing electrolyte descriptors, as it determines the fundamental properties of electrolytes and ultimately influences the characteristics, complexity and dimensions of the descriptors, which serve as the network input. Consequently, the annotated data exceedingly influence the predictive capabilities of the descriptors. The optimal approach based on this would be to obtain a great amount of structural data from the same reliable source. This was not an option since no such publicly available dataset exists, which limited the data collection to manual annotation and collection. However, while the amount of data collected in this project may not be sufficient for extensive model training purposes, it does provide enough data for developing functional electrolyte descriptors and testing a functional prediction model.

One of the main challenges associated with collecting battery data from open sources is the complexity and variability of cycling test characteristics. Not only is it a challenging task to find similar research papers, but also difficult to find research papers that provide all the comprehensive information used within their battery testing. Additionally, there is a significant variation of testing metrics like C-rate and voltage range among these papers. This variability adds complexity and increases the dimensionality, hence weakening the comparability between the different battery tests. The number of parameters that profoundly affect the results includes temperature, C-rate, voltage range and electrodes used. For instance, a battery performance test conducted over a wider voltage range may face more challenging conditions and consequently yield worse cycle results compared to a similar battery test performed over a narrower voltage range. Furthermore, open-source battery tests could easily fool the masses by tuning parameters and presenting results from certain viewpoints [75]. This makes it challenging to differentiate and compare the performance between the different battery tests, posing a complexity for an ML model to identify patterns.

This issue of complexity could be resolved by reducing the number of parameters and performance metrics, which could be done by simply performing each battery test with the same metrics, i.e. same dimensions. Ideally, the only varying metric would be the electrolyte composition. Nevertheless, as previously mentioned in the introduction, this objective lies beyond the scope of this project and is constrained by the availability of open-source battery testing resources.

Furthermore, the estimated target value *Cycles to 70%* serves as a convenient method for comparing battery performance, as it assigns standardized labels to battery tests within the same dimension. The 70% mark is rooted in the recognition that a battery has reached its EoL, which is commonly used as an indicator of a finished test. This target value provides a consistent reference across all tests, making it a suitable label for ML. However, this approach is a simplified way of determining its performance and may lose accuracy, since the provided data varies. Various parameters can affect the behavior of battery tests including the equilibration cycles in the beginning, as explained in 3.1.1, which is performed differently among research papers and solved by multiplying an individual gradient factor. Although

efforts were conducted to determine this factor through interpretation and analysis, it should be considered to be arbitrarily chosen. In conclusion, the labeling is only as accurate as it could be from visual interpretation, making prediction harder.

4.2 Descriptors

The resulting electrolyte descriptors are *DataElectrolyte* objects that are flexible for the intended user. In Figure 4.2, its attributes are presented. All attributes are vital for the future neural network to process it correctly, although the molecule list is only necessary for plotting the electrolyte. As can be seen in Figure 4.2, all attributes of the electrolyte are explained and each has significance for the descriptor to be viable.

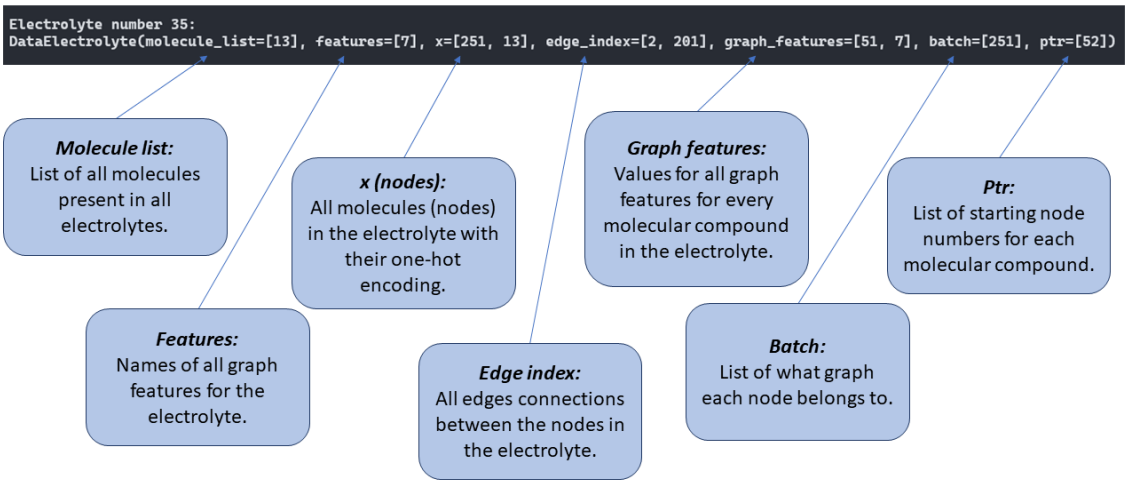


Figure 4.2: Illustration of all attributes of a DataElectrolyte object.

The graph attributes of the electrolyte descriptor are essential to ensure that the neural network can accept it as input, these include *x*, *edge index* and *batch*. The nodes along with their one-hot encoding are stored in x and the edge index ensures that each connection between nodes is accounted for. Batch is the attribute that keeps track of what subgraph each node belongs to, which is essential when pooling the molecular compounds in the neural network. However, the most important traits of the descriptor are the graph features, which are the values produced by the analysis from CHAMPION. This is what separates this project from other studies on electrolyte performance prediction since these descriptors include additional information about the simulated behavior of the electrolytes.

The resulting electrolyte descriptor contains the ability to plot the representation of the electrolyte or one of the molecular compounds in it. In Figure 4.3, a plot of an electrolyte is presented. The plot enables the user to inspect the electrolyte further and detect specific molecules in it.

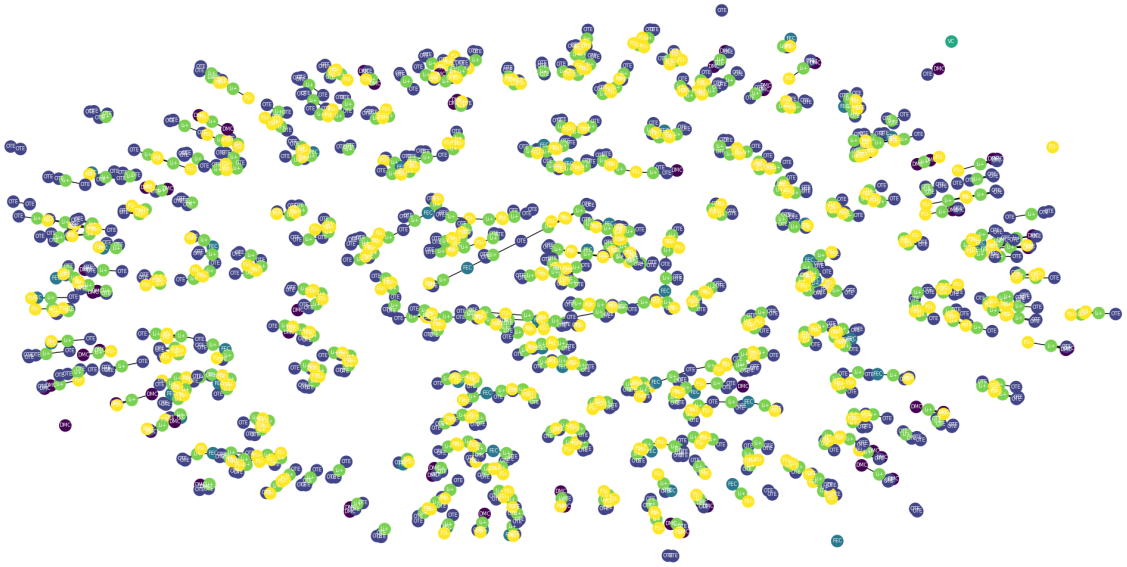
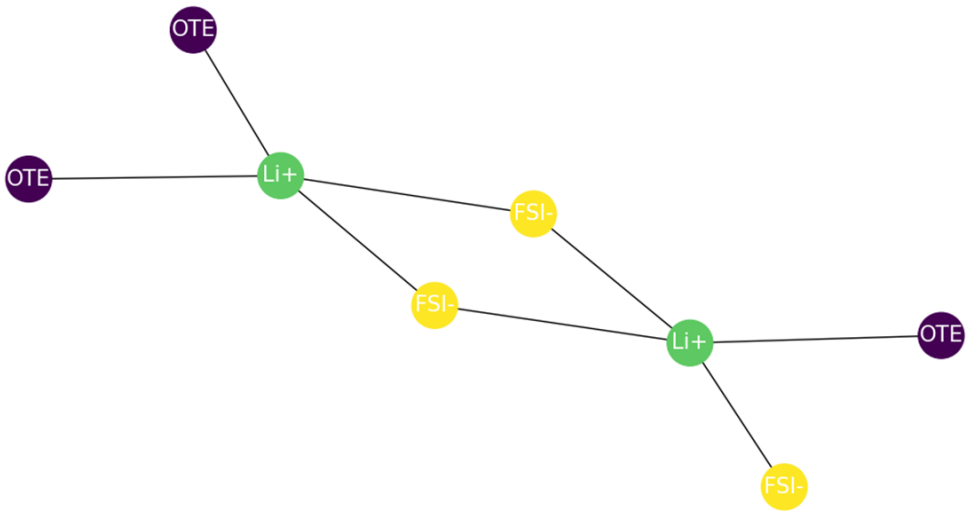


Figure 4.3: Representation of an electrolyte including all molecular compounds in it.

In Figure 4.4, a plot of specific molecular compounds is presented along with its graph features. Since the descriptor has this method attached, the user is enabled to analyze specific molecular compounds within the electrolyte and investigate its features more accurately.



GraphNumber	FCharge	FMass	TBodyTypeConcentrationTrajectory	TMolarConductivity	TElectricalMobility	TIonicConductivity	TDiffusivity
12	-1.0	1550.489	0.001042	0.000091	0.000091	9.509502e-08	0.000224

Figure 4.4: Plot of graph number 12 in the electrolyte along with its corresponding graph features. In this example, the included graph features are charge, mass, concentration, molar conductivity, electrical mobility, ionic conductivity and diffusivity.

As discussed, the produced electrolyte descriptors are flexible and incorporate the necessary attributes that are required for a GNN to accept it as input. The flexibil-

ity is especially demonstrated when choosing graph features from the analysis using CHAMPION. There are several features that were not included in this project which can efficiently be added, using the *MolGraph* class discussed in section 3.3.2. Since the main aim was to add features that affect the performance of the battery, relevant electrolyte attributes in this aspect could be added subsequently. Furthermore, the ability to include multiple molecular compounds as graphs is favorable in this project. This is valuable since all electrolytes contain several molecules and molecular compounds that usually are represented as graphs. Therefore, the descriptors are suitable for a GNN and can include the necessary graph features to predict the cycle performance. However, what features are relevant for the prediction are yet to be determined, but the user could easily switch between graph attributes in order to find the most suitable for the task.

The user has, as mentioned, the ability to add further features to the descriptors. This is valuable in the aspect of battery level features such as electrodes, C-rates for charge and discharge as well as cycling temperature. These electrolyte attributes were annotated in the collected data as described in section 4.1. Since these attributes can have relevance to the performance of the electrolyte, they could be valuable as input to the GNN as well. In this project, these features were not added to the electrolytes but could easily be introduced as electrolyte level features in the *DataElectrolyte* objects.

4.3 Model Evaluation

The developed GNN accepts an electrolyte descriptor as input in order to predict the cycle number of the electrolyte. The predictive capabilities of the GNN are presented as the loss over epochs in Figure 4.5.

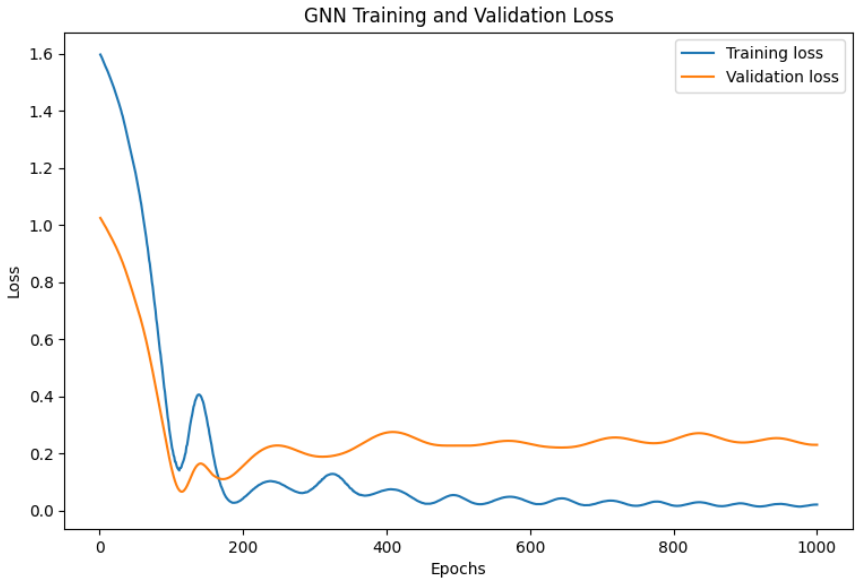


Figure 4.5: Training and validation loss of the GNN over 1000 epochs.

As can be seen in the resulting plot, the training loss is decreasing and converges after 1000 epochs which indicates that it learns from the training set. However, the validation loss is high compared to the training loss which demonstrates the absence of accurate prediction on unseen data. Since the amount of training data is confined, it does not represent the entirety of the dataset and subsequently the model overfits the training data. Overfitting could also occur when the dataset possesses noisy data, which would indicate that some of the features within the electrolyte data points are not relevant for predicting purposes. In the context of this thesis, it is impossible to deduce if this is the case since the amount of data is limited. Furthermore, due to complex and time-consuming simulations, only 20 out of the 104 battery tests were simulated and could be generated as electrolyte descriptors. Out of these 20 electrolytes, 6 had inherent faulty data and had to be removed. The remaining 14 electrolytes provided a sufficient amount of molecular compound data which enabled pattern recognition on the training set but as expected it does not accurately represent all possible input data points.

One way to expand the dataset for an ML model is to use data augmentation [76], which is a procedure where the data is modified to appear different for the model. This way, the amount of data could increase and the diversity of the dataset could grow. Applying data augmentation on this dataset could be an option but would probably be difficult since the data is produced from analysis using CHAMPION and the target values are based on battery testing. Another way to achieve better performance with limited data is to use transfer learning [77]. In transfer learning, the model is pre-trained, typically trained on a large and general dataset. By transmitting learned weights and representations, the model can benefit from the knowledge acquired during the pre-training phase, leading to improved performance as well as reduced data requirements for the target task. However, this may be difficult to apply on this GNN since there are no similar networks that are pre-trained for the same purpose as this project. Nonetheless, it may be applicable in the aspect of generating more descriptor data, by using descriptors of molecules from publically available databases if they are similar enough.

Another critical aspect of the model evaluation is the importance of node embedding. Proper node embeddings are essential in a GNN considering that node features are propagated through graphs and the model learns from this. If the node embeddings are not diverse enough, the necessary information about each node will not matter for the model. In this project, one-hot encodings were used as node representations to specify what molecular species each node is. However, molecular fingerprints [78] are another option for node embedding. Molecular fingerprints encode the structure of a certain molecule in order to extract relevant features. The most common fingerprint is a series of binary digits (bits) that represent the presence or absence of particular substructures within a molecular species. These fingerprints could enrich the feature representation of molecules in the electrolyte and possibly work better as node embedding. Such fingerprints are obtainable using structural text representation as input in open-source toolkits for cheminformatics such as RDKit [79]. The molecular text representation could be accessed through the database following

the CHAMPION analysis step. With that in mind, the lack of time in this project prevented further investigation of fingerprints as node embedding.

Although there are several approaches to increase the performance of the model, these options could only be examined further if the dataset was more comprehensive. Without a sufficient amount of data, it is difficult to detect if an alteration improves the model performance or coincidentally improves it. With an extended dataset, the hyperparameters such as model complexity and learning rate could also be examined to a greater extent. By adjusting the number of layers, learning rate and optimizer, the prediction capability could increase and consequently enhance model performance. However, the hyperparameter modification currently ensures the model's ability to accept electrolyte descriptors as input and generates a prediction based on the training set. Since this was the primary scope of the thesis, the results are satisfactory.

5

Conclusions & Future Work

This thesis challenges to bridge the gap between empirical experiments and theoretical understanding of battery cycling performance as well as reducing the need for extensive manual testing by aiming to determine cell cycling performance as a function of electrolyte compositions. In detail, this project investigated the development of structural electrolyte descriptors for the purpose of cycling prediction. The developed descriptors were developed based on electrolyte MD simulations and CHAMPION analysis, successfully possessing structural information about molecular compounds and their inherent chemical and physical features. The electrolyte descriptors are structured as *DataElectrolyte* objects that easily could be modified based on the desired features. The developed electrolyte descriptors effectively serve as cycle predictors by serving as input to a neural network, namely a GNN, where they demonstrate successful functionality. In correlation with the project limitations, only an initial version of the network was developed. Although the model robustly exhibited some indications of predictive capabilities, it is challenging to draw definitive conclusions about its performance.

For the matter of improving model performance, further analysis necessitates a larger dataset to obtain more conclusive results. The significance of data volume can be attributed to several factors. Firstly, a larger dataset provides a broader representation of the diverse range of electrolyte compositions and properties, allowing the models to capture the intricacies and nuances of electrolyte behavior more effectively. Additionally, a substantial dataset helps overcome issues related to data sparsity and improves the generalizability of the developed models. By training on a larger and more diverse dataset, the GNN can better account for variations in electrolyte systems, leading to enhanced predictive capabilities.

Building upon our findings, there are several promising avenues for future research in the field of predictive electrolyte descriptors. Primarily, efforts should focus on expanding and diversifying the existing electrolyte databases. By collaborating with researchers and industry partners, data-sharing initiatives can be established to consolidate and standardize electrolyte information from various sources. This ensures the avoidance of manual data collection and annotation, which is highly time-consuming. Furthermore, the development of experimental techniques and automated data acquisition methods, such as those employed by Compular, holds the potential to facilitate the rapid generation of large-scale electrolyte datasets.

Moreover, future work involves exploring advanced ML and data analysis techniques.

Deep learning architectures, such as recurrent neural networks or transformers, can be employed to leverage vast amounts of data and extract intricate patterns from electrolyte descriptors. Additionally, incorporating domain knowledge and physical principles into the predictive models can further enhance their accuracy and interoperability. Moreover, it is crucial to validate and refine the developed predictive models through experimental validation. Conducting targeted experiments using selected electrolyte compositions predicted by the models can provide valuable feedback and enable iterative improvements.

By addressing these areas of future work, we can continue to harness the power of data and predictive models to propel advancements in electrolyte research, leading to more efficient, safer and sustainable energy storage technologies.

Bibliography

[1] Arora, N. K., Fatima, T., Mishra, I., Verma, M., Mishra, J., & Mishra, V. (2018). Environmental sustainability: challenges and viable solutions. Environmental Sustainability, 1, 309-340.

[2] Verdecchia, R., Sallou, J., & Cruz, L. (2023). A Systematic Review of Green AI. arXiv preprint arXiv:2301.11047.

[3] Borah, R., Hughson, F. R., Johnston, J., & Nann, T. (2020). On battery materials and methods. Materials Today Advances, 6, 100046.

[4] Compular. (n.d.). Compular Tech. Retrieved May 26, 2023, from https://compulartech.com/

[5] Andersson, R., Årén, F., Franco, A. A., & Johansson, P. (2021). CHAMPION: Chalmers hierarchical atomic, molecular, polymeric and ionic analysis toolkit. Journal of Computational Chemistry, 42(23), 1632-1642.

[6] Mabbott, G. A. (1983). An introduction to cyclic voltammetry. Journal of Chemical education, 60(9), 697.

[7] Huang, J., Dong, X., Wang, N., & Wang, Y. (2022). Building low-temperature batteries: Non-aqueous or aqueous electrolyte?. Current Opinion in Electrochemistry, 100949.

[8] Dou, Q., Wang, Y., Wang, A., Ye, M., Hou, R., Lu, Y., ... & Yan, X. (2020). "Water in salt/ionic liquid" electrolyte for 2.8 V aqueous lithium-ion capacitor. Science Bulletin, 65(21), 1812-1822.

[9] Quintans De Souza, G. (2021). A comparison between aqueous and organic electrolytes for lithium ion batteries.

[10] Kim, T., Song, W., Son, D. Y., Ono, L. K., & Qi, Y. (2019). Lithium-ion batteries: outlook on present, future, and hybridized technologies. Journal of materials chemistry A, 7(7), 2942-2964.

[11] Angulakshmi, N., & Stephan, A. M. (2015). Efficient electrolytes for lithium–sulfur batteries. Frontiers in Energy Research, 3, 17.

[12] Manthiram, A. (2017). An outlook on lithium ion battery technology. ACS central science, 3(10), 1063-1069.

[13] Xie, J., & Lu, Y. C. (2020). A retrospective on lithium-ion batteries. Nature communications, 11(1), 2499.

[14] Boz, B., Dev, T., Salvadori, A., & Schaefer, J. L. (2021). Electrolyte and electrode designs for enhanced ion transport properties to enable high performance lithium batteries. Journal of The Electrochemical Society, 168(9), 090501.

[15] Li, J., Mazzola, M. S., Gafford, J., Jia, B., & Xin, M. (2012). Bandwidth based electrical-analogue battery modeling for battery modules. Journal of Power Sources, 218, 331-340. https://doi.org/10.1016/j.jpowsour.2012.07.006

[16] Meng, Y. S., Srinivasan, V., & Xu, K. (2022). Designing better electrolytes. Science, 378(6624), eabq3750.

[17] Xu, K. (2004). Nonaqueous liquid electrolytes for lithium-based rechargeable batteries. Chemical reviews, 104(10), 4303-4418.

[18] Kang, S. J., Park, K., Park, S. H., & Lee, H. (2018). Unraveling the role of LiFSI electrolyte in the superior performance of graphite anodes for Li-ion batteries. Electrochimica Acta, 259, 949-954.

[19] Chen, K., & Xue, D. (2016). Materials chemistry toward electrochemical energy storage. Journal of Materials Chemistry A, 4(20), 7522-7537.

[20] Armand, M., & Tarascon, J. M. (2008). Building better batteries. nature, 451(7179), 652-657.

[21] Ashrafizadeh, S. N., Seifollahi, Z., Ganjizade, A., & Sadeghi, A. (2020). Electrophoresis of spherical soft particles in electrolyte solutions: A review. Electrophoresis, 41(1-2), 81-103.

[22] Andersson, R. (2020). Dynamic Structure Discovery and Ion Transport in Liquid Battery Electrolytes. Chalmers Tekniska Hogskola (Sweden).

[23] Lewandowski, A., & Świderska-Mocek, A. (2009). Ionic liquids as electrolytes for Li-ion batteries—An overview of electrochemical studies. Journal of Power sources, 194(2), 601-609.

[24] Chen, X., & Zhang, Q. (2020). Atomic insights into the fundamental interactions in lithium battery electrolytes. Accounts of Chemical Research, 53(9), 1992-2002.

[25] Schwietert, T. K., Arszelewska, V. A., Wang, C., Yu, C., Vasileiadis, A., de Klerk, N. J., ... & Wagemaker, M. (2020). Clarifying the relationship between redox activity and electrochemical stability in solid electrolytes. Nature materials, 19(4), 428-435.

[26] Ping, P., Wang, Q., Sun, J., Xiang, H., & Chen, C. (2010). Thermal stabilities of some lithium salts and their electrolyte solutions with and without contact to a LiFePO4 electrode. Journal of the Electrochemical Society, 157(11), A1170.

[27] Li, Q., Chen, J., Fan, L., Kong, X., & Lu, Y. (2016). Progress in electrolytes for rechargeable Li-based batteries and beyond. Green Energy & Environment, 1(1), 18-42.

[28] Newman, J., & Balsara, N. P. (2021). Electrochemical systems. John Wiley & Sons.

[29] Borah, R., Hughson, F. R., Johnston, J., & Nann, T. (2020). On battery materials and methods. Materials Today Advances, 6, 100046.

[30] Guena, T., & Leblanc, P. (2006, September). How depth of discharge affects the cycle life of lithium-metal-polymer batteries. In INTELEC 06-Twenty-Eighth International Telecommunications Energy Conference (pp. 1-8). IEEE.

[31] Ahn, D., & Raj, R. (2011). Cyclic stability and C-rate performance of amorphous silicon and carbon based anodes for electrochemical storage of lithium. Journal of Power Sources, 196(4), 2179-2186.

[32] Mothilal Bhagavathy, S., Budnitz, H., Schwanen, T., & McCulloch, M. (2021). Impact of charging rates on electric vehicle battery life. Findings, 2021(March).

[33] Laidler, K. J. (1984). The development of the Arrhenius equation. Journal of chemical Education, 61(6), 494.

[34] Zhang, S., Xu, K., & Jow, T. (2003). Low-temperature performance of Li-ion cells with a LiBF 4-based electrolyte. Journal of Solid State Electrochemistry, 7, 147-151.

[35] Ma, S., Jiang, M., Tao, P., Song, C., Wu, J., Wang, J., ... & Shang, W. (2018). Temperature effect and thermal impact in lithium-ion batteries: A review. Progress in Natural Science: Materials International, 28(6), 653-666.

[36] Kissinger, P. T., & Heineman, W. R. (1983). Cyclic voltammetry. Journal of chemical education, 60(9), 702.

[37] Togasaki, N., Yokoshima, T., Oguma, Y., & Osaka, T. (2020). Prediction of overcharge-induced serious capacity fading in nickel cobalt aluminum oxide lithium-ion batteries using electrochemical impedance spectroscopy. Journal of Power Sources, 461, 228168.

[38] Belov, D., & Yang, M. H. (2008). Failure mechanism of Li-ion battery at overcharge conditions. Journal of Solid State Electrochemistry, 12, 885-894.

[39] Meegoda, J. N., Malladi, S., & Zayas, I. C. (2022). End-of-Life Management of Electric Vehicle Lithium-Ion Batteries in the United States. Clean Technologies, 4(4), 1162-1174.

[40] Wang, A., Kadam, S., Li, H., Shi, S., & Qi, Y. (2018). Review on modeling of the anode solid electrolyte interphase (SEI) for lithium-ion batteries. npj Computational Materials, 4(1), 15.

[41] Yu, Z., Wang, H., Kong, X., Huang, W., Tsao, Y., Mackanic, D. G., ... & Bao, Z. (2020). Molecular design for electrolyte solvents enabling energy-dense and long-cycling lithium metal batteries. Nature Energy, 5(7), 526-533.

[42] Xia, J., Ma, L., & Dahn, J. R. (2015). Improving the long-term cycling performance of lithium-ion batteries at elevated temperature with electrolyte additives. Journal of Power Sources, 287, 377-385.

[43] Su, C. C., He, M., Shi, J., Amine, R., Zhang, J., Guo, J., & Amine, K. (2021). Superior long-term cycling of high-voltage lithium-ion batteries enabled by single-solvent electrolyte. Nano Energy, 89, 106299.

[44] Nagpure, S. C., Tanim, T. R., Dufek, E. J., Viswanathan, V. V., Crawford, A. J., Wood, S. M., ... & Liaw, B. (2018). Impacts of lean electrolyte on cycle life for rechargeable Li metal batteries. Journal of Power Sources, 407, 53-62.

[45] Li, Q., Jiao, S., Luo, L., Ding, M. S., Zheng, J., Cartmell, S. S., ... & Xu, W. (2017). Wide-temperature electrolytes for lithium-ion batteries. ACS applied materials & interfaces, 9(22), 18826-18835.

[46] Cao, X., Jia, H., Xu, W., & Zhang, J. G. (2021). Localized high-concentration electrolytes for lithium batteries. Journal of The Electrochemical Society, 168(1), 010522.

[47] Chae, S., Kwak, W. J., Han, K. S., Li, S., Engelhard, M. H., Hu, J., ... & Zhang, J. G. (2021). Rational design of electrolytes for long-term cycling of Si anodes over a wide temperature range. ACS Energy Letters, 6(2), 387-394.

[48] Li, W., Dolocan, A., Li, J., Xie, Q., & Manthiram, A. (2019). Ethylene carbonate-free electrolytes for high-nickel layered oxide cathodes in lithium-ion batteries. Advanced Energy Materials, 9(29), 1901152.

[49] Brox, S., Röser, S., Husch, T., Hildebrand, S., Fromm, O., Korth, M., ... & Cekic-Laskovic, I. (2016). Alternative Single-Solvent Electrolytes Based on Cyanoesters for Safer Lithium-Ion Batteries. ChemSusChem, 9(13), 1704-1711.

[50] Sharova, V., Moretti, A., Diemant, T., Varzi, A., Behm, R. J., & Passerini, S. (2018). Comparative study of imide-based Li salts as electrolyte additives for Li-ion batteries. Journal of Power Sources, 375, 43-52.

[51] Hansson, T., Oostenbrink, C., & van Gunsteren, W. (2002). Molecular dynamics simulations. Current opinion in structural biology, 12(2), 190-196.

[52] Frenkel, D., Smit, B., & Ratner, M. A. (1996). Understanding molecular simulation: from algorithms to applications (Vol. 2). San Diego: Academic Press.

[53] Leach, A. R. (2001). Molecular modelling: principles and applications (2nd ed.). Prentice Hall.

[54] Hollingsworth, S. A., & Dror, R. O. (2018). Molecular Dynamics Simulation for All. Neuron, 99(6), 1129-1143. https://doi.org/10.1016/j.neuron.2018.08.011

[55] Grimme, S., Bannwarth, C., & Shushkov, P. (2017). A robust and accurate tight-binding quantum chemical method for structures, vibrational frequencies, and noncovalent interactions of large molecular systems parametrized for all spd-block elements (Z= 1–86). Journal of chemical theory and computation, 13(5), 1989-2009.

[56] Kühne, T. D., Iannuzzi, M., Del Ben, M., Rybkin, V. V., Seewald, P., Stein, F., ... & Hutter, J. (2020). CP2K: An electronic structure and molecular dynamics software package-Quickstep: Efficient and accurate electronic structure calculations. The Journal of Chemical Physics, 152(19), 194103.

[57] Beezer, R. A. (2008). Review of: Graph Theory by JA Bondy and USR Murty.

[58] Chartrand, G., Lesniak, L., & Zhang, P. (2010). Graphs & digraphs (Vol. 39). CRC press.

[59] Goyal, P., & Ferrara, E. (2018). Graph embedding techniques, applications, and performance: A survey. Knowledge-Based Systems, 151, 78-94.

[60] Fey, M., & Lenssen, J. E. (2019). Fast graph representation learning with PyTorch Geometric. arXiv preprint arXiv:1903.02428.

[61] Hagberg, A., Swart, P., & S Chult, D. (2008). Exploring network structure, dynamics, and function using NetworkX (No. LA-UR-08-05495; LA-UR-08-5495). Los Alamos National Lab.(LANL), Los Alamos, NM (United States).

[62] Hopfield, John J. "Artificial neural networks." IEEE Circuits and Devices Magazine 4.5 (1988): 3-10.

[63] Abraham, A. (2005). Artificial neural networks. Handbook of measuring system design.

[64] LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. Nature 521, 436–444 (2015). https://doi.org/10.1038/nature14539

[65] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. nature, 521(7553), 436-444.

[66] Wu, Y., Lian, D., Xu, Y., Wu, L., & Chen, E. (2020, April). Graph convolutional networks with markov random field reasoning for social spammer detection. In Proceedings of the AAAI conference on artificial intelligence (Vol. 34, No. 01, pp. 1054-1061).

[67] Sanchez-Gonzalez, A., Heess, N., Springenberg, J. T., Merel, J., Riedmiller, M., Hadsell, R., & Battaglia, P. (2018, July). Graph networks as learnable physics engines for inference and control. In International Conference on Machine Learning (pp. 4470-4479). PMLR.

[68] Hamaguchi, T., Oiwa, H., Shimbo, M., & Matsumoto, Y. (2017). Knowledge transfer for out-of-knowledge-base entities: A graph neural network approach. arXiv preprint arXiv:1706.05674.

[69] Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., ... & Sun, M. (2020). Graph neural networks: A review of methods and applications. AI open, 1, 57-81.

[70] Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., ... & Chen, T. (2018). Recent advances in convolutional neural networks. Pattern recognition, 77, 354-377.

[71] Wu, F., Souza, A., Zhang, T., Fifty, C., Yu, T., & Weinberger, K. (2019, May). Simplifying graph convolutional networks. In International conference on machine learning (pp. 6861-6871). PMLR.

[72] Seko, A., Togo, A., & Tanaka, I. (2018). Descriptors for machine learning of materials data. Nanoinformatics, 3-23.

[73] Amar, Y., Schweidtmann, A. M., Deutsch, P., Cao, L., & Lapkin, A. (2019). Machine learning and molecular descriptors enable rational solvent selection in asymmetric catalysis. Chemical science, 10(27), 6697-6706.

[74] Himanen, L., Jäger, M. O., Morooka, E. V., Canova, F. F., Ranawat, Y. S., Gao, D. Z., ... & Foster, A. S. (2020). DScribe: Library of descriptors for machine learning in materials science. Computer Physics Communications, 247, 106949.

[75] Johansson, P., Alvi, S., Ghorbanzade, P., Karlsmo, M., Loaiza, L., Thangavel, V., ... & Årén, F. (2021). Ten ways to fool the masses when presenting battery research. Batteries & Supercaps, 4(12), 1785-1788.

[76] Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. Journal of big data, 6(1), 1-48.

[77] Weiss, K., Khoshgoftaar, T. M., & Wang, D. (2016). A survey of transfer learning. Journal of Big data, 3(1), 1-40.

[78] Duvenaud, D. K., Maclaurin, D., Iparraguirre, J., Bombarell, R., Hirzel, T., Aspuru-Guzik, A., & Adams, R. P. (2015). Convolutional networks on graphs for learning molecular fingerprints. Advances in neural information processing systems, 28.

[79] RDKit: Open-source cheminformatics. https://www.rdkit.org

DEPARTMENT OF PHYSICS
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY