Technical University of Applied Sciences Würzburg-Schweinfurt (THWS)

Faculty of Computer Science and Business Information Systems

Master Thesis

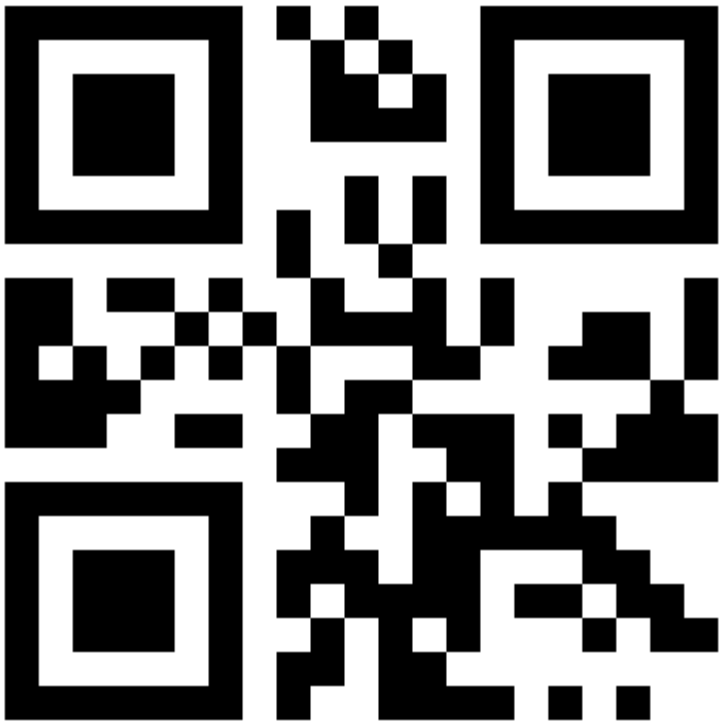
Electric Motor Modelling via Graph Neural Networks

Submitted to the Technical University of Applied Sciences Würzburg-Schweinfurt in the Faculty of Computer Science and Business Information Systems to complete a course of studies in Master of Artificial Intelligence

Lilly Abraham
K64889

To be Submitted on: 11.12.2024

Initial examiner: Prof. Dr. Magda Gregorova
Secondary examiner: Prof. Gracia Herranz Mercedes



Abstract

The thesis explores an approach to predict Key Performance Indicators(KPI)s of topology invariant Interior Permanent Magnet Synchronous Magnets(IPSM) Electric Motors by transforming its geometric, physical and simulation parameters into a graph representation.
The KPIs to be predicted are plots on Efficiency grid(3D) and Torque curve(2D).
We aim to first parameterize the EM design such that it is feasible to convert into a graph representation.
Next, we would create a Graph with relevant attributes and design a Graph Neural Network(GNN)with the graph as input and the plots in the format of vectors as target values.
Additionally we may also need to customize the loss function in a way that would smoothen out the plot curves of the prediction values.
Then, we would evaluate the predictions with the test target values by experimenting with various hyper-parameter tuning settings and as a baseline with an Multi Layer Perceptron(MLP) model of the parameters in tabular form.
Finally we will enable the KPI's plot visualisation in a manner presentable to the client Valeo(Automaker Company).
Not necessary remove i suppose.... The aim of the Master Thesis is to train a neural network to learn the parameters of Electric Motors and thus be able to predict its KPIs. The KPIs are 2D and 3D plots on Torque(Mgrenz) curve(Mgrenz) and Efficiency grid(ETA). Other KPIs can be calculated from these two KPIs. For instance the Vibration Costs are inversely proportional to the Efficieny values predicted.

Acknowledgement

I would like to thank my supervisor Prof. Dr. Magda Gregorova for her guidance and support throughout the course of this thesis. Her dedication and commitment to our work has been inspiring for me to the extend that I might have developed a new love for academia.
I would also like to express my sincere gratitude to Valeo for providing us with the dataset. Special thanks are in order to Daniel and Leo for sharing valuable insights of the data from an electromechanical standpoint and for giving me a detailed understanding of my task.
I deeply owe my Mother and my Siblings who have been very instrumental in making it possible for me to pursue a Master's degree outside my home country and for the endless support thoughout.
Finally, I humble myself before God Almighty for all his blessings and for giving me the strength to persevere and bring my dreams to fruition.

Contents

Abstract **2**

Acknowledgement **3**

Contents **4**

Abbreviations **6**

1 Introduction **7**
1.1 Objective . . . 8
1.2 Problem Statement . . . 8
1.3 Research Question . . . 9
1.4 Thesis Structure . . . 9

2 Literature Review **10**

3 Dataset **11**
3.1 Data Preprocessing for Multi Linear Perceptron (MLP) . . . 13
3.1.1 Deep Dive into the Input Parameters . . . 13
3.1.2 Deep Dive into the 2 Dimension (2D) KPI(Torque Curve) . . . 15
3.1.3 Deep Dive into the 3 Dimension (3D) KPI(ETA Curve) . . . 16
3.2 Scaling . . . 18
3.3 Dataset splitting . . . 18

4 Graph Modelling **19**
4.1 Data Preprocessing for Graph Neural Network (GNN) . . . 19
4.2 Heterogeneous GNN Model . . . 21

5 Modelling & Evaluation **23**
5.1 MLP Model . . . 23
5.2 Loss Functions . . . 24
5.2.1 Loss Regularization for 2D Key Performance Indicator (KPI)(Torque curve) . . . 24
5.2.2 Loss Customization for 3D KPI(Efficiency Grid) . . . 25
5.3 Optimizer . . . 26
5.4 Evaluation Metrics . . . 26
5.4.1 Evaluation Metrics for 2D KPI . . . 26
5.4.2 Evaluation Metrics for 3D KPI . . . 27

5.5 Post Processing 27

6 Experiments and Results **28**
6.1 Experiments with MLP 28
6.2 Results with MLP 30
6.2.1 2D Torque Curve Results with MLP 30
6.2.2 3D ETA Grid Results with MLP 32
6.3 Results with Baseline 35
6.3.1 2D Torque Curve Results with Baseline 36
6.3.2 3D ETA Grid Results with Baseline 37
6.4 Ablation Studies 38

7 Conclusion **40**
7.1 Future Improvements 40

List of Figures **41**

List of Tables **42**

Bibliography **43**

Declaration on oath **44**

Consent to Plagiarism Check **45**

Abbreviations

GNN	Graph Neural Network
MLP	Multi Linear Perceptron
KPI	Key Performance Indicator
EM	Electric Motor
FEM	Finite Element Method
CNN	Convolution Neural Network
2D	2 Dimension
3D	3 Dimension
Wandb	Weights & Biases
MSE	Mean Squared Error

Chapter 1

Introduction

In the design of electric motors, vast amounts of data are generated to determine which design of an Electric Motor (EM) fits best to KPIs.
KPIs of an Electric Motor are essential to judge the performance of the motor before it is manufactured.
Traditionally these KPIs are inferred from a description of an EM design via a Finite Element Method (FEM) approximating the solutions of the Maxwell's equations.
This process, though well established in the EM design, is very time consuming and does not allow for high-throughput engine design optimization.
The actual engine data of Valeo is used here as the dataset comprising of multiple variant designs of the Double-V topology.
The 3 motor topologies manufactured by Valeo are as below:

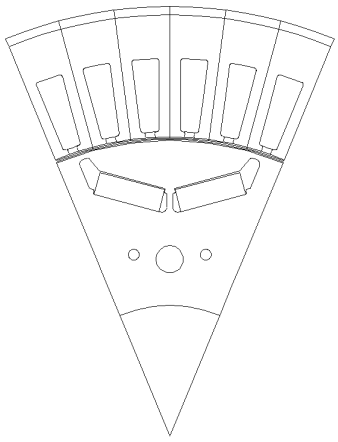


Figure 1.1: V1 Magnet (Source:Valeo)

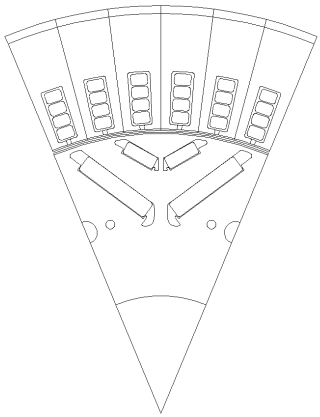


Figure 1.2: V2 Magnet (Source:Valeo)

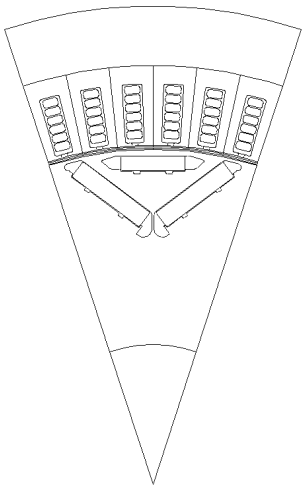


Figure 1.3: Nabla Magnet (Source:Valeo)

This master thesis explores a way to do surrogate modelling of the current process as is highlighted in Figure 1.4 by making use of GNN or MLP for the modelling of electrical engine designs described parameterically.

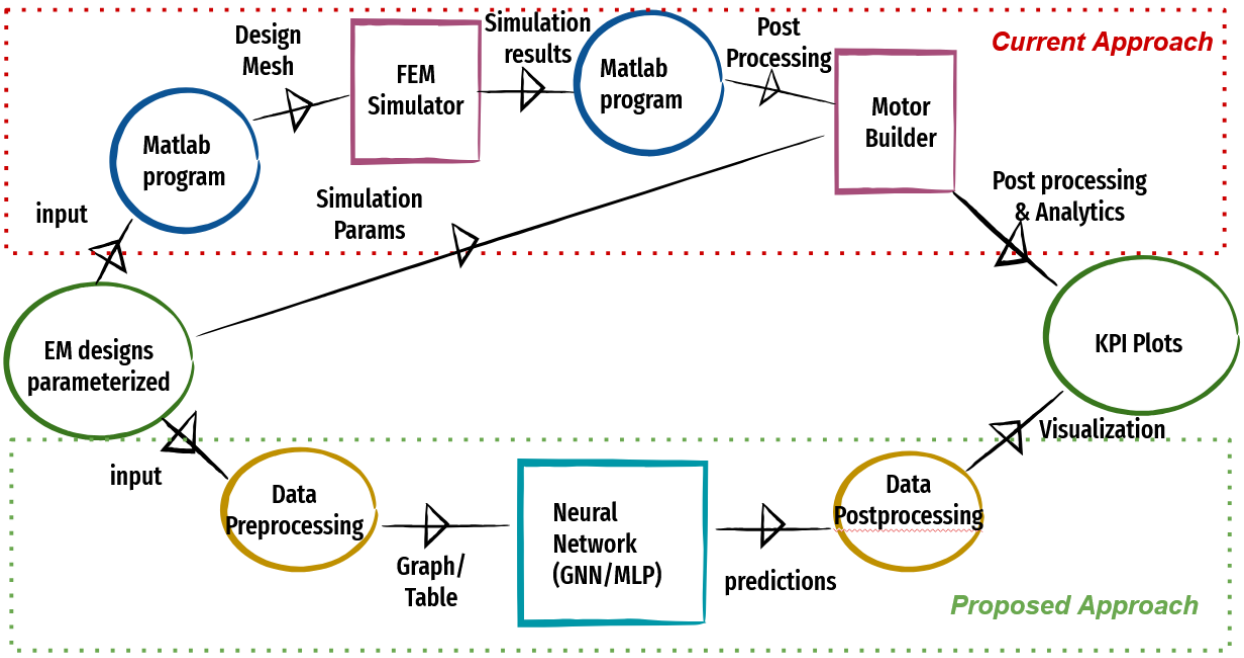


Figure 1.4: EM Design Flowchart

1.1 Objective

Our task is to predict 2 KPIs namely the Torque curve and the ETA grid from the parameteric description of topology invariant Electris Motors.
The Torque curve is a 2D plot ie, a vector across speed ranges and the ETA grid is a 3D plot ie, a matrix of dimensions torque ranges and its speed ranges.
The ETA grid is dependent on the shape of the torque curve and lies within the boundary of the torque curve.

1.2 Problem Statement

Although our training dataset was highly imbalanced to consists majorly of 1 topology ie, Double V Magnet Topology, we aim to build a system that caters to every topology.
With the tabular representation of the data, we assign distinct columns to include features from each topology.

Given ETA is dependent on Torque curve, we need to build a model that can predict both these KPIs. However, due to the challenge of predicting ETA grid of varying dimensions for each motor variant, we decided against building a model architecture wherein 2D KPI is dependent on the 2D KPI.
This in a way could be overcome if we build 2 models one for the 2D and 3D KPIs and thus feed in the corresponding 2D predictions when training the latter.
However, it would be computationally expensive and does not help in the scenario when we might need to generate EM parameteric descriptions. Additionally we deemed it unnecessary as the dimensions of the ETA grid vary with the torque curve and not necessarily the ETA values.

Furthermore, we presume graph representation of the data will be more logical than tabular representation due to its ability to grasp the connections within the EM structural properties better. This would be even more realistic with achieving topology invariance than the tabular representation and conserve memory and compute in the long run. However, we realize that our problem cannot be solved using Homogeneous

GNN which is relatively simpler and is built on a single node and edge type. In order to model our problem as a graph, we need to represent it as a Heterogeneous graph.

1.3 Research Question

GNN in general have not been to less explored even so more the heterogeneous GNN. Particularly in the scenario of Electric Motor Modelling, there has been no publications with GNNs. Hence the need to check its feasibility and its performance with our benchmarks on tabular data.

1.4 Thesis Structure

The thesis is structured to follow sections namely Literature Review, Dataset, Modelling, Experiments and Results, Conclusion, Appendix and Bibliography.
In Literature Review section will introduce the works that has already been carried out in this domain.
In the Dataset section a detailed insight to how our data is structured is elaborated.
In the Modelling section, we introduce the methodologies used to tackle the problem.
The Experiments and Results chapter gives an outlook on the outcomes of our work in addition to other findings we unearth.
Conclusion chapter summarizes the thesis briefly and would also give a small glimpse into areas of improvement.
Finally the Bibliography section lists out the articles cited for this thesis.

Chapter 2

Literature Review

There has been extensive research in modeling the Electric Motor with Convolution Neural Network (CNN) based on the images of the motor cross-section.
However our approach is progressive in the sense that once the KPIs are predicted we would like to be able to generate the inputs.
Reproducing images is not known to apt given the infamous known fact that AI generated images are faulty instead by generating the parameters of the motor we can be rest assured of more precise results.
Hence the need to focus on the inputs as they are with their parametric description.
Existing literature also covers works on modelling this work as tabular data using MLPs. Although this is fairly good forseeing the impact of generating the inverse process yet MLPs cannot necessarily learn all the intricacies within motor components.
Hence the need to better represent the data typically in the form of graphs and model Graph Neural Networks to achieve the desired results.
There has been close to no work of GNNs in this domain. However we see progress of GNNs in molecular chemistry and social networks usecases from which we draw inspiration.

Chapter 3

Dataset

Valeo an automotive company has supplied the dataset consisting of close to 1500 Double V Electric Motor parameters. There are close to 196 parameters which comprises of the geometric, physical and simulation properties of the motor.

The geometry of a whole Double V motor is as below

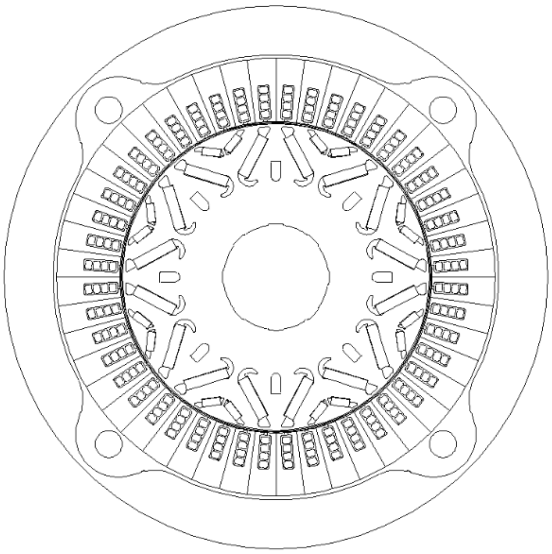


Figure 3.1: Complete EM Geometry(Source:Valeo)

Below is the geometry of 1/8 cross-section of the same motor.

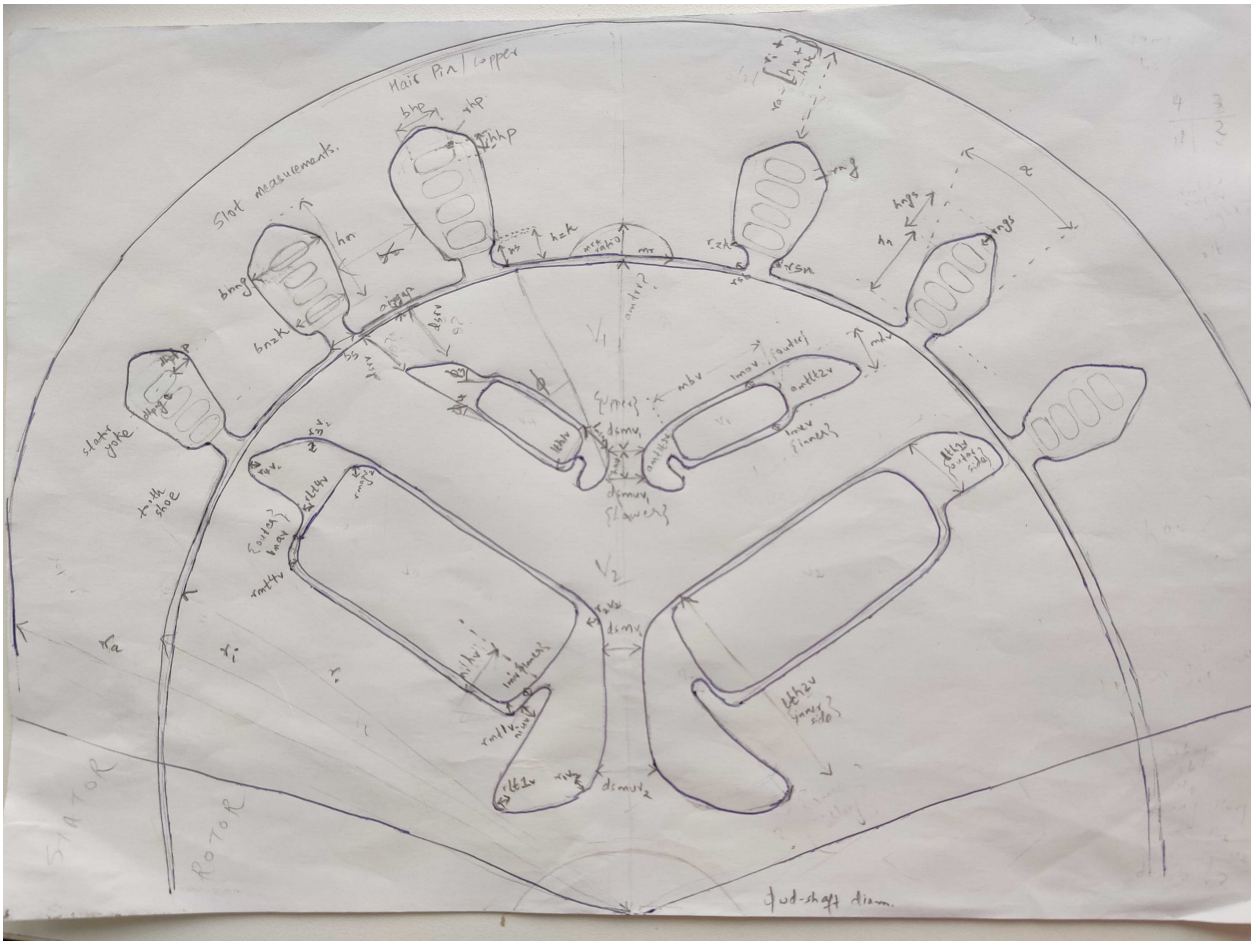


Figure 3.2: 1/8 Motor Crossection

Valeo has shared approximately 1500 Excel Workbook files for each motor variant. Each of the excel files contain multiple sheets. The sheets of interest to us are as below :

Sheet	**Sheet Name**	**Description**
Motor Parameters	input_data	• Contains the input parameters for the motor model. • Includes geometric, physical, and simulation properties. • Unit dimensions if applicable are generally mm or degrees
Speed Grid	NN	• Contains the input parameters for the motor model. • Includes geometric, physical, and simulation properties. • Unit dimensions if applicable are generally mm or degrees
Torque Grid	MM	• Contains the torque grid. • Used for plotting the ETA grid.
ETA Grid	ETA	• Contains the ETA grid. • Have the same dimensions as that of NN and MM sheets
Torque Curve	Mgrenz	• Contains the values corresponding to torque curve. • Have the same columns corresponding to the speed grid.

Table 3.1: Excel File Structure of an EM variant

3.1 Data Preprocessing for MLP

For modelling the MLP, we present the data in tabular form with the parameters corresponding to columns. In order to make the data compatible with our model, some level of data processing was carried out as elaborated below.

3.1.1 Deep Dive into the Input Parameters

All parameters including the additional ones in each topology is considered as a separate column and therefore if a particular column is topology dependent the data of the other topologies for that corresponding column is treated as 0 values.
The values are read and stored in their float equivalent to preserve data precision. Furthermore all degree columns are converted to their equivalent radian values to be fed to the model..EXPLAIN THE REASON WHY WITH CITATION PROBABLY::::

Parameter	Unit of Dimension	Mean	Standard Deviation	Value Range	Single V Magnet Topology	Double V Magnet Topology	Nabla Magnet Topology
General Parameters							
N	#	99	50–150	100	110	90	×
simQ	#	99	50–150	100	110	90	✓
r_a	mm	99	50–150	100	110	90	1,400
r_i	mm	99	50–150	100	110	90	1,400
Rotor Parameters							
lmsov2	mm	1,500	1,000–2,000	1,500	1,600	1,400	1,400
lth1v2	mm	1,325	1,000–1,500	1,300	1,350	1,325	1,400
lth2v2	mm	1,325	1,000–1,500	1,300	1,350	1,325	1,400
r1v2	mm	1,500	1,000–2,000	1,500	1,600	1,400	1,400
r11v2	mm	1,325	1,000–1,500	1,300	1,350	1,325	1,400
r2v2	mm	1,325	1,000–1,500	1,300	1,350	1,325	1,400
r3v2	mm	1,325	1,000–1,500	1,300	1,350	1,325	1,400
r4v2	mm	1,325	1,000–1,500	1,300	1,350	1,325	1,400
rmt1v2	mm	1,325	1,000–1,500	1,300	1,350	1,325	1,400
rmt4v2	mm	1,325	1,000–1,500	1,300	1,350	1,325	1,400
rlt1v2	mm	1,325	1,000–1,500	1,300	1,350	1,325	1,400
rlt4v2	mm	1,325	1,000–1,500	1,300	1,350	1,325	1,400
hav2	mm	1,325	1,000–1,500	1,300	1,350	1,325	1,400
mbv2	mm	1,325	1,000–1,500	1,300	1,350	1,325	1,400
mhv2	mm	1,325	1,000–1,500	1,300	1,350	1,325	1,400
rmagv2	mm	1,325	1,000–1,500	1,300	1,350	1,325	1,400
dsmv2	mm	1,325	1,000–1,500	1,300	1,350	1,325	1,400
dsmuv2	mm	1,325	1,000–1,500	1,300	1,350	1,325	1,400
amtrv2	mm	1,325	1,000–1,500	1,300	1,350	1,325	1,400
dsrv2	mm	1,325	1,000–1,500	1,300	1,350	1,325	1,400

Continued on next page

Table 3.2 – continued from previous page

Parameter	Unit of Dimension	Mean	Standard Deviation	Value Range	Single V Magnet Topology	Double V Magnet Topology	Nabla Magnet Topology
lmav2	mm	1,325	1,000–1,500	1,300	1,350	1,325	1,400
lmiv2	mm	1,325	1,000–1,500	1,300	1,350	1,325	1,400
lmov2	mm	1,325	1,000–1,500	1,300	1,350	1,325	1,400
lmuv2	mm	1,325	1,000–1,500	1,300	1,350	1,325	1,400
Stator Parameters							
Sekhar	M	922	800–1,100	900	950	915	1,400
Dipen	M	598	500–700	600	625	570	1,400

Table 3.2: EM Input Parameters

3.1.2 Deep Dive into the 2D KPI(Torque Curve)

Below is the standard deviation of the test dataset for the 2D KPI(Torque Curve).

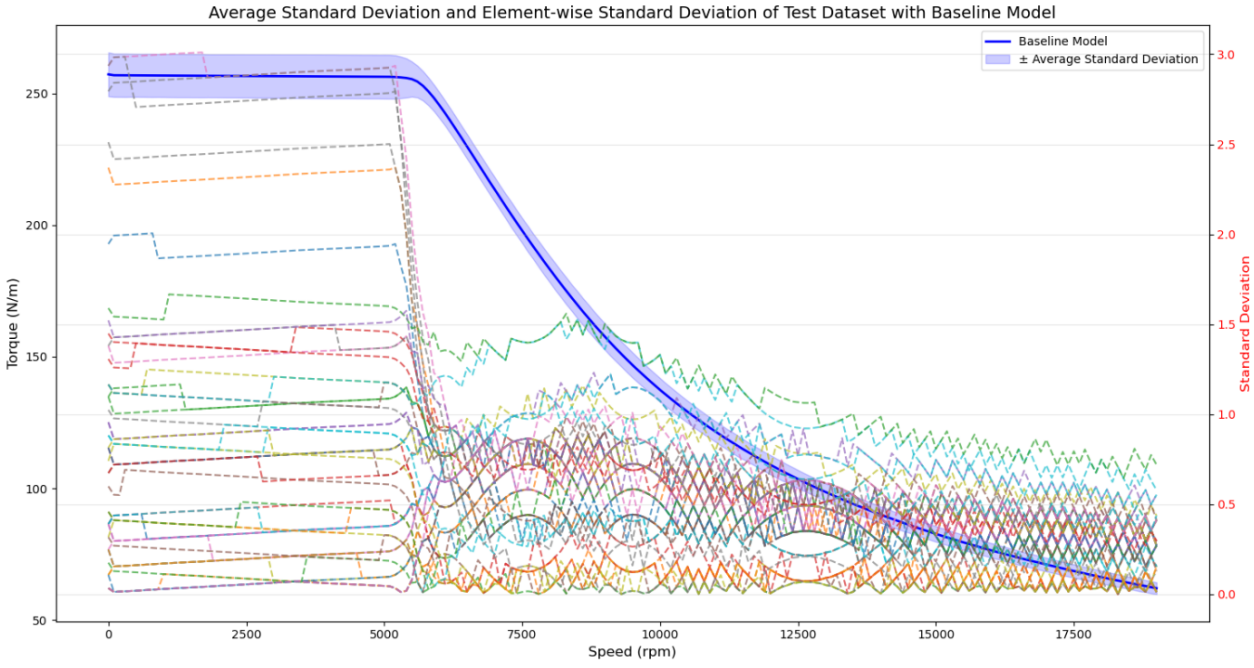


Figure 3.3: Standard Deviation of 2D KPI(ETA) Targets

From the above plot, we can conclude that the curve is relatively constant for 1/4 of the curve after which it is relatively monotonically decreasing. This finding is critical for how we modelled the loss regularization for the target. It will be further elaborated in the Section §5.2.1

3.1.3 Deep Dive into the 3D KPI(ETA Curve)

As the target values Mgrenz and ETA grids are not provided with the correct dimensions we have an additional step which takes the maximum torque value from the Mgrenz grid and create a similar grid ranging from -maximum torque to maximum torque.
We then choose only the rows corresponding to this range from the actual MM grid supplied and the same row indices is used to retrieve the ETA grid.
This step ensures that we grant the model the correct dimensions of the ETA grid based on Torque curve and predict likewise.

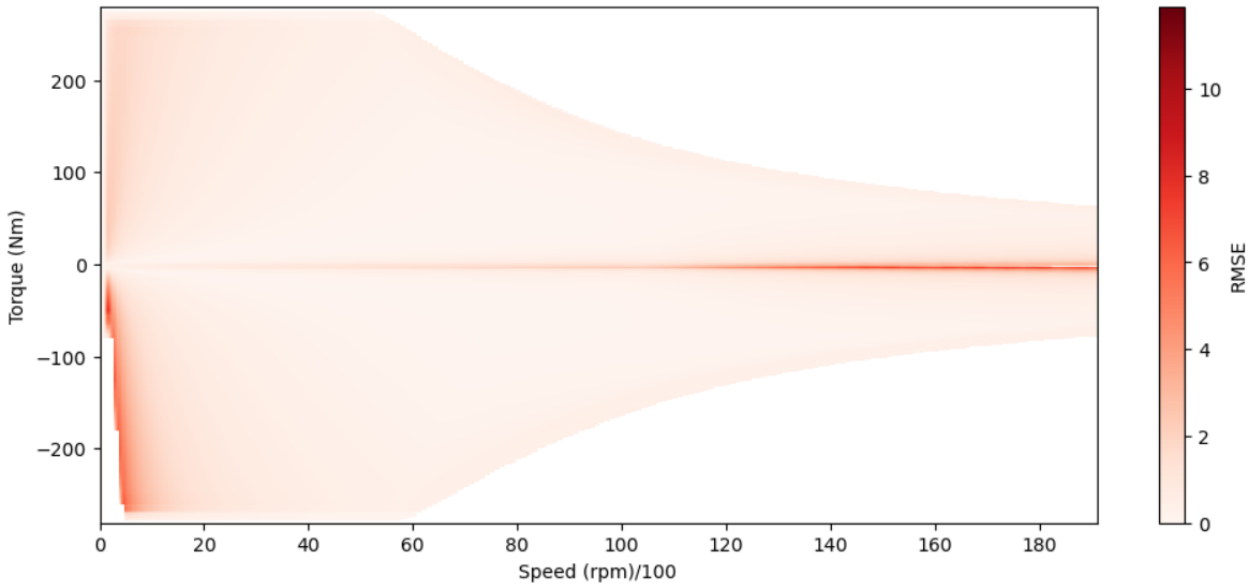


Figure 3.4: Standard Deviation of 3D KPI(ETA)

More apparent below :

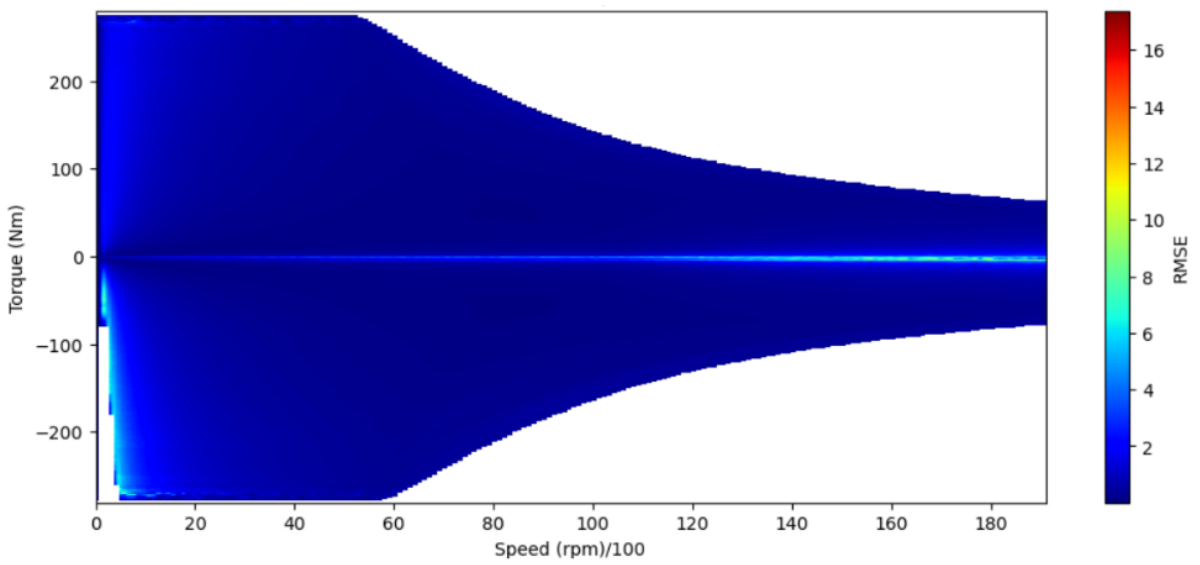


Figure 3.5: Standard Deviation of 3D KPI(ETA)

From our analysis of the above plots on the standard deviation of the ETA grids from the test dataset, we notice that the data shared was faulty in the ETA corresponding to negative torque values. As these are FEM simulations, it is probably an effect of a post processing step taken by the Motor builder. The ETA grid corresponding to negative torque values when motor is in generating mode is not the replica as how it is for positive torque values when motor is in monitoring mode. This is evident from low speed high torque distribution area where we can see NAN values across the test dataset

Additionally the ETA grid envelope is completely dependent on its equivalent torque curve. The area beneath the boundary of which is looked into by the EM manufacturers to determine the car's efficiency in the operating cycle.

From the plots received we find the ETA grid does not only cover the area within the torque curve but also a slightly exaggerated area typically in the portion where the 1/4 where the curve starts to monotonically decrease. To ensure that out model is not tricked into assuming predictions for this extra area we extact the grid to be of the same shape as that of its curve. This is a step also done as part of data post processing to be further overed in Section BLAH CITE.

Based on this finding, we have decided to only consider the ETA grid in monitoring mode for predictions and mirror the same to replicate the efficiency when it is in generating mode.

As reading the files take up a lot of time and compute, we read the files as a onetime job at the start and store them into pythonic objects for faster access in the future.
Both the input and target values for 1st KPI is stored locally as pandas dataframe .csv files whereas the 2nd KPI is stored as separate csv files per variant considering it is in the form of a 2D array.
The csv files are then concatenated and stored into a numpy array conserving dimensionality by padding nan values to match dimensionality of the grid corresponding to the torque curve with the largest torque value.
In our case the value is 561 but this is subject to change as we receive more data and can be overridden by the user.
The array is then saved locally as a .npy file for easy access and loading during training.

The input data consists of about 1500 examples mostly from the Double V Magnet Topology (and about 3 examples each for the other 2 topologies).

3.2 Scaling

We have used Standard Scaling for the input and both the outputs.

The Scaling is formulated as shown below :

$$z = \frac{x - \mu}{\sigma} \tag{3.1}$$

where

- x *: Input*
- μ*: : Mean*
- σ*: : Standard Deviation*

both Mean and Standard deviation are calculated across columns. This is attributed to the fact we have columns with different ranges for the input. Meanwhile the torque curve and the ETA grid are flattened to be 1D and thus have only 1 column.

We then removed scaling for the targets because these donot enter the model but is only used for loss calculation. With lower learning rates and higher epochs, we see the model's y1 performing better than with scaling. A plausible reason could be since the y1 targets are integers typically but we use torch tensor as it is a regression problem. This misleads the model with scaled targets to lose precision by generating within the full capacity of float tensors. However without scaling, we observe the model spits out integers itself even when the datatype is still torch tensor. This inturn fixes the y1 earlier problem of having fluctuations in the curve. Furthermore it now eliminates the need for regularizing the Y1 targets as predictions are as per target a smooth decreasing curve.

Additionally, the nan values in y2 are handled by masking before loss calculation as is defended in Section CITE Loss function Initially before masking y2's nans we try to set it as an incredibly high value - but this resulted in poor predictions as the model must have been confused and tried to increase its spread of predictions to cover this value and so all true values were also predicted to be close to this dummy value–with scaling

3.3 Dataset splitting

We have also split the dataset to have about 50 samples for test and the remaining is used for 5 fold cross validation with 80:20 split for training and validation.
The reason we have a separate test dataset from the validation is to ensure that there is no data leakage as we donot want to overfit the testdataset with the hyperparameters we choose during training.

Within 5 folds, we expect to cover most grounds on training and have good monitoring on the model's performance for each fold. We also choose the best hyperparameters that fits with our work on analysing the logs. The best performing model is then chosen from the folds based on the model which has a combination of least aggregated loss for both outputs and scores for each output which are closest to 0.

Chapter 4

Graph Modelling

We intended to model our problem as a Graph and solve using Graph Neural Networks. We presume this will be a more clever way of representing our usecase in comparision to tabular data.. The idea was to make use of the Graph dynamics in our usecase and aggregate features that are semantically similar. The main method of aggregating information within the nodes of a Graph is through Message Passing. Graph Neural Networks can be broadly classified as : Homogeneous GNN Here the nodes and edges are of the same type. Message passing is done across the neighbouring nodes and edges over hops until it learns a representation equivalent from its neighbours Heterogeneous GNN Wherein the node and edge are of different types. Here message passing is conditioned on the node and edge type thus allowing the flow of information to be more controlled.

Standard Message Passing GNNs (MP-GNNs) can not trivially be applied to heterogeneous graph data, as node and edge features from different types can not be processed by the same functions due to differences in feature type. A natural way to circumvent this is to implement message and update functions individually for each edge type. During runtime, the MP-GNN algorithm would need to iterate over edge type dictionaries during message computation and over node type dictionaries during node updates.

4.1 Data Preprocessing for GNN

Node types

1. **General**

   - General parameters:
   $$r = \{r_i\} \quad \forall i \in \{a, r, o\}$$

   where:
   - r_a*: Outer Radius of the Stator*
   - r_r*: Outer Radius of the Rotor*
   - r_o*: Center of the EM*

2. **Stator**

   - Slot windings:
   $$sw = \{s_i w_j\} \quad \forall i \in \{1, \ldots, QSim\}, \quad \forall j \in \{1, \ldots, N\}$$
   - Slots:
   $$s = \{s_i\} \quad \forall i \in \{1, \ldots, QSim\}$$

   where
   - *Qsim : Count of slots in the Stator*
   - *N : Count of copper windings per slot*

3. **Rotor**

   - Magnet Flux Barriers:

$$v = \{v_{ij}\} \quad \forall i \in \{1, \ldots, T\}, \quad \forall j \in \{1, \ldots, V\}$$

   - Magnets:

$$vm = \{v_i m_j\} \quad \forall i \in \{1, \ldots, T\}, \quad \forall j \in \{1, \ldots, V\}$$

   where
   - *T : Topology type of the EM*
   - *V : Type of Magnet*

   As Valeo only manufactures Double V magnets we consider it to be 2

Edge types

1. **Angle**
   Relevant Paths

$$vm - -vm = \{v_{i_1} m_{j_1} - v_{i_2} m_{j_2}\} \forall i_1, i_2 \in \{1, \ldots, T\}, \quad \forall j_1, j_2 \in \{1, \ldots, V\} \mid i_1 = i_2, \quad j_1 \neq j_2$$

   angle=vm-vm

2. **Distance**
   Relevant Paths

$$vi - -vi = \{v_{ij_1} - v_{ij_2}\}, \forall i \in \{1, \ldots, T\}, \forall j_1, j_2 \in \{1, \ldots, V\} \mid j_1 \neq j_2$$

$$vi - -vj = \{v_{i_1 j} - v_{i_2 j}\}, \forall i_1, i_2 \in \{1, \ldots, T\}, \forall j \in \{1, \ldots, V\} \mid i_1 \neq i_2$$

$$v - -vm = \{v_{ij} - v_i m_j\} \forall i \in \{1, \ldots, T\}, \quad \forall j \in \{1, \ldots, V\}$$

$$v - -rr = \{v_{ij} - r_r\}, \forall i, j \in \{1, \ldots, T\}$$

$$o - -r = \{(o - r_r), (o - r_a)\}$$

$$rr - -s = \{r_r - s_i\}, \forall i \in \{1, \ldots, QSim\}$$

$$s - -sw = \{s_i - s_i w_j\}, \forall i \in \{1, \ldots, QSim\}, \forall j \in \{1, \ldots, N\}$$

$$s - -ra = \{s_i - r_a\}, \forall i \in \{1, \ldots, QSim\}$$

$$sw - -sw = \{s_i w_{j_1} - s_i w_{j_2}\}, \forall i \in \{1, \ldots, QSim\}, \forall j \in \{1, \ldots, N\} \mid (j_1 == j_2 - 1)$$

   distance = vi–vi + vi–vj + v–vm + v–rr + o–r + rr–s + s–sw + s–ra + sw–sw

Node Features

1. **v** = {lmsov, lth1v, lth2v, r1v, r11v, r2v, r3v, r4v, rmt1v, rmt4v, rlt1v, rlt4v, hav}
2. **vm** = {mbv, mhv, rmagv}
3. **r** = {r}
4. **s** = {b_nng, b_nzk, b_s, h_n, h_s, r_sn, r_zk, r_ng, h_zk}
5. **sw** = {bhp, hhp, rhp}

Path Features

1. **vm–vm** = {deg_phi}

2. **vi–vi** = {dsm, dsmu}
3. **vi–vj** = {amtrvj-amtrvi}
4. **v–vm** = {lmav, lmiv, lmov, lmuv}
5. **v–r** = {amtrv, dsrv}
6. **o–r** = {r}
7. **rr–s** = {airgap}
8. **s–sw** = {dhphp}
9. **sw–sw** = {dhpng}
10. **s–ra** = {r_a-(r_i + h_n + h_zk)}

The heterogeneous graph that was constructed earlier is as below:

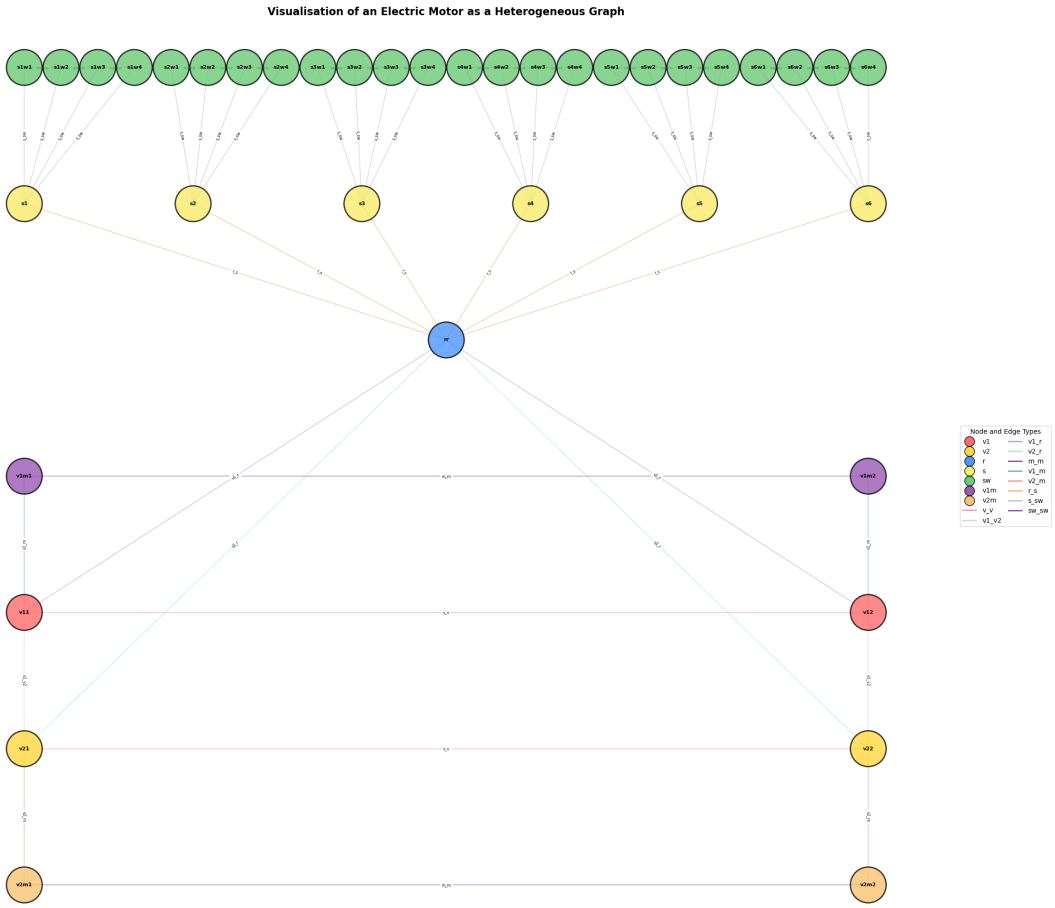


Figure 4.1: HetGraph

4.2 Heterogeneous GNN Model

We find the heterogeneous graph to be most apt for our use case with its different node and edge types as it preserves both the structural and semantics of our data.

This property is crucial in modelling our use case as we will then have similar node-edge types per topology. In addition the count of certain parameters with the motor such as stator poles with its corresponding slot and rotor magnets is made more comprehendable to the model having new nodes and edges whereas for the MLP architecture this information is represented only as a number in yet another column.
Heterogeneous GNN generally work by having separate non linear functions convolve over each edge type during message computation and over each node type when aggregating the learned information.

Chapter 5

Modelling & Evaluation

Since we aim to predict continuous vector values, we model this task into a regression problem As a baseline, we first train a MLP on the tabular representation of the data and work on it further to do the same with a heterogeneous GNN.

5.1 MLP Model

For the MLP model, we use a single model with input features corresponding to all the features in the tabular topology invariant representation of the data.
The model architecture is build to predict both the 2D and 3D KPIs by having 2 separate output layers for each of the KPIs.
Since the 2D KPI's targets are relatively learnable than that of the 3D KPI's targets we have experimented with fewer feed forward layers in the former than in the latter. RELU layers were also added in between to serve as the activation function and produce non-linearities in the model.
Dropout layers ensure that not all neurons in each layer are used up during training to prevent the model from memorizing the data and hence overfitting. Batch normalisation layers are used to normalize the input to the next layer and hence speed up the training process.

Th model architecture is designed to have 2 fully connected layers with RELU as an activation function and a 1D batch normalisation layer with a dropout layer for the 1st output.
We have also used a dropout percentage of 0.2 to ensure the model does not overfit the data by dropping out 20 percent of the neurons in the layer. Although the targets for the first output are an array of integer values, we use the float tensor and not integer tensor to represent the data else it would become a classification problem and not a regression problem as it should be.
For the second output we have 5 fully connected layers with RELU as an activation function and dropout and 2D batch normalisation for each layer.

The below figure gives an outline on how the MLP Model architecture is designed.

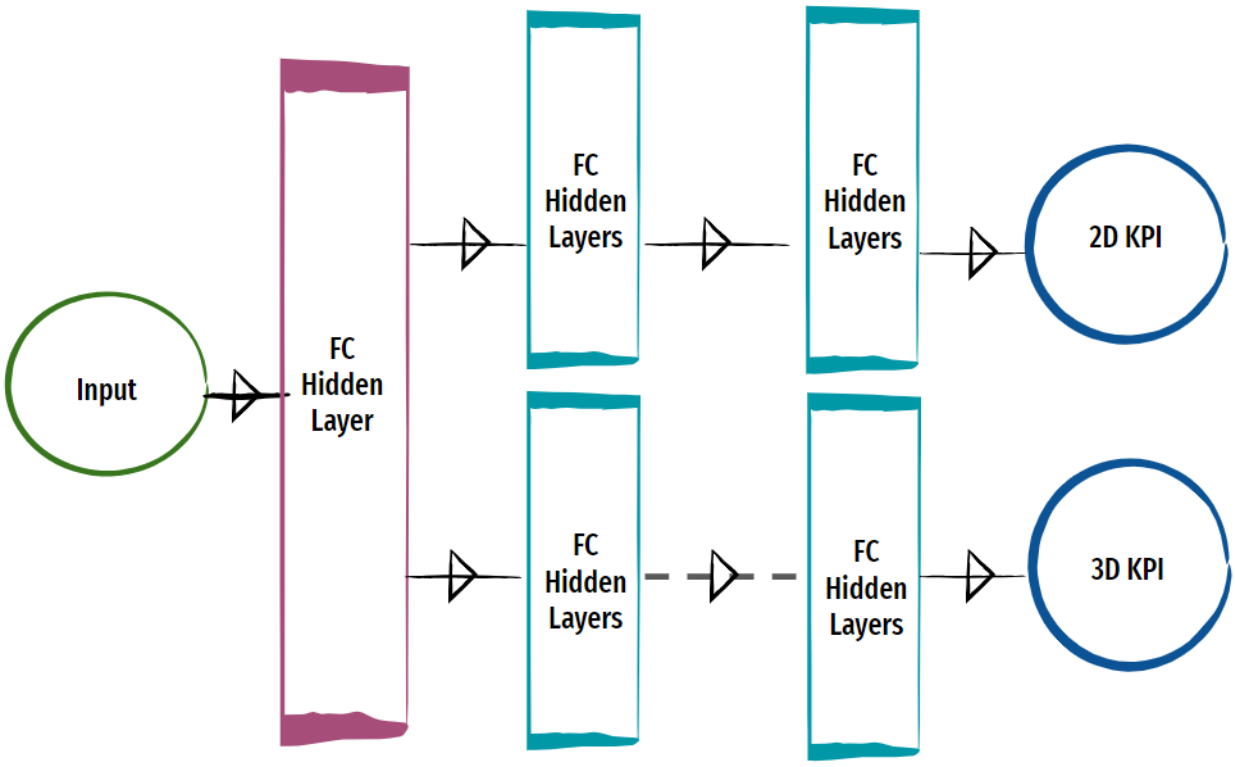


Figure 5.1: MLP Model Architecture

5.2 Loss Functions

The Mean Squared Error (MSE)loss is the loss function used for our problem with the intention that the losses are minimized. We have adopted 2 methodologies to regularize the loss one each for the 2D and 3D KPI.

5.2.1 Loss Regularization for 2D KPI(Torque curve)

The MSE loss for the 2D KPI is formulated as shown below :

$$\text{Y1 Loss} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h}\sum_{j=1}^{h}(y_{ij} - \hat{y}_{ij})^2 \tag{5.1}$$

where

- n *: EM samples*
- h *: dimensionality of 1D vector*

To smoothen out the curve for the 2D KPI(Torque curve) we apply a loss regularisation factor to take into account before backpropagating it the model during training.
We observed the curve inherently follows a decreasing pattern CITE AGAIN::COULD BE A KNOWN FACT:::and hence using this knowledge penalize the loss for non-decreasing values within each prediction.
CITE!!!!If the electromagnetic coil is enabled by the commutator for the time span t3, the (almost) maximal current is running through it's loops and the (almost) maximal magnetic field strength is generated. The (almost) maximal torque is acting on the rotor. If the time span is shortened to t2 by increasing rotational

speed, a slightly lower torque is acting, because the current through the coil is decreasing slightly. When reducing the time span to t1, the coil gets disconnected from the input voltage even though just half the maximum current is reached. Accordingly the torque decreases significantly:

$$\text{Y1 Loss Regularization} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h-1}\sum_{j=1}^{h-1}\left(\text{ReLU}(\hat{y}_{ij+1} - \hat{y}_{ij})\right)^2 \tag{5.2}$$

where

- *RELU : Activation Function*

Theoretically, we would expect the model to generate better predictions but on closer observation we notice the curve is still not smooth.
A reason to attribute this could be the model's incapability to infer that loss decrease depends on not just the prediction and target values but also within prediction values. Our deduction is that a single value calculated for the entire curve may not be sufficient to regularize this loss as we imagined.

A reason to attribute this could be the model's struggle to attempt to generate the next point in the curve to be smaller than the previous point yet also for the further point in addition to making sure the MSE is calculated accurately. Our hypothesis is this loss regularization creates a tug of war between the MSE loss and the regularization term when predicting the whole curve and hence the ever fluctuating curve being generated.

5.2.2 Loss Customization for 3D KPI(Efficiency Grid)

The MSE loss is calculated for the 3D KPI is formulated as shown below :

$$\text{Y2 Loss} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{w}\frac{1}{h}\sum_{j=1}^{w}\sum_{k=1}^{h}(M_{ijk}\cdot y_{ijk}) - (M_{ijk}\cdot \hat{y}_{ijk}))^2 \tag{5.3}$$

where

- M_{ijk} *: Mask matrix*

$$M_{ijk} = \begin{cases} 1 & \text{if } y_{ijk} \neq nan \\ 0 & \text{if } y_{ijk} = -1 \end{cases}$$

- w *: 1st dimension of 3D vector*
- h *: 2nd dimension of 3D vector*

The ETA Grid is a 3D plot of real numbers ranging between 0 and 100.NEED TO CITE...EVEN IF IT IS A KNOWN FACT We noticed in some portions of the grid, the plot not visible as it had nan values.
As ANN cannot be trained to predict NAN values we have a binary mask constructed such that values corresponding to -nan in the target have value 0 and all other values as 1. The mask is then multiplied with the prediction. Furthermore the mask is used to replace nan values in target with 0 as multiplying a nan value with 0 results in yet another nan value. This would cascade into nan loss calculation.
After this step, the MSE loss is calculated and backpropagated.
Mathematically, this process can be expressed as above
This formulation ensures that the -1 values (which replaced NaN values in the grid) are ignored in the loss calculation, as they are multiplied by 0 in the mask.

It is a known fact that at 0 Torque, the corresponding efficiency values for the motor is 0. To model this conception into our network, we have attempted to retrieve the center row of our ETA grid and the values it consists finally determine the violation count. We formulate is mathematically as below:

$$\text{Y2 Loss Regularization} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h}\sum_{k=1}^{h}|y_{i0k}| \tag{5.4}$$

We take the absolute values from the center row of the matrix else the loss would be negative and could negate with positive values. This would have a detrimental effect when calculating the loss

$$\begin{aligned}\text{Total Loss} = \text{Y1 Weight} \times (\text{Y1 Loss} + (\lambda_{\text{y1}} \times \text{Y1 Loss Regularization}))+ \\ \text{Y2 Weight} \times (\text{Y2 Loss} + (\lambda_{\text{y2}} \times \text{Y2 Loss Regularization}))\end{aligned} \tag{5.5}$$

where

$$\lambda_{y1} : \textit{Y1 Loss Regularization Weight}$$

$$\lambda_{y2} : \textit{Y2 Loss Regularization Weight}$$

- *Y1 Weight : Y1 Loss Weightage*
- *Y2 Weight : Y2 Loss Weightage*

5.3 Optimizer

Adam optimizer is used for optimization as it is known to be computationally efficient and requires little memoryBLAH BLAH
The optimizer acts once the loss is backpropagated across training each batch of the dataset.
We have also experimented with a learning rate scheduler which reduced the learning rate exponentially by a gamma parameter to decay learning as training progresses across epochs.

5.4 Evaluation Metrics

The evaluation metrics we have considered for our regression problem is the root mean squared error.
The model with the least combined loss and prediction scores closest to 0 is ideal for our application.

5.4.1 Evaluation Metrics for 2D KPI

The y1 loss for the 2D KPI is formulated as shown below :

$$\text{Y1 score} = \frac{1}{n}\sum_{i=1}^{n}\sqrt{\frac{1}{h}\sum_{j=1}^{h}(y_{ij} - \hat{y}_{ij})^2} \tag{5.6}$$

where

- *n : EM samples*
- *h : dimensionality of 1D vector*

5.4.2 Evaluation Metrics for 3D KPI

The y2 loss for the 3D KPI is formulated as shown below :

$$\text{Y2 score} = \frac{1}{n}\sum_{i=1}^{n}\sqrt{\frac{1}{w}\frac{1}{h}\sum_{j=1}^{w}\sum_{k=1}^{h}(y_{ijk} - \hat{y}_{ijk})^2} \tag{5.7}$$

where

- n *: EM samples*
- w *: 1st dimension of 3D vector*
- h *: 2nd dimension of 3D vector*

5.5 Post Processing

The mean and standard deviation from the train-validation datasets are applied to transform the test dataset to maintain uniformity on the predictions generated. In the case of new files we first convert it into the tabular representation our model consumes and then do the scaling.
This is the reason why we preserve the same scalers used during training as we not only evaluate our dedicated test dataset but also for clients to use on demand.

The inverse scaling is formulated as shown below :

$$\text{x} = \frac{z \times \sigma}{\mu}$$

where

- z *: Scaled Input*
- μ*: : Mean*
- σ*: : Standard Deviation*

Furthermore as part of post processing, we slice the 3D ETA grid to only contain the values within the predicted torque curve.

Chapter 6

Experiments and Results

6.1 Experiments with MLP

The hyperparameters were chosen via a random grid search and we finalized those that resulted in the ideal scores for both y1 ad y2 respectively. The hyperparmaters for the model was tuned over observations of the model's performance across 5 fold cross validation training.
The splits are saved locally and can be used to later to ensure reproducibility.

Hyperparameters	**Description**	**Value**
lr	Learning Rate	0.00075
hidden_sizes	Dimensionality of Hidden Layers	256
lr_gamma	Exponential Learning Rate Scheduler	0.9
batch_size	Batch Size	72
epochs	Number of Epochs	8
p	Dropout Probability	0.2
lambda_y1	Y1 Loss Regularizer	2.5
lambda_y2	Y2 Loss Regularizer	1.5
w_y1	Weightage of Y1 Loss	0.8
w_y2	Weightage of Y2 Loss	0.2

Table 6.1: Hyperparameter Tuning

Over the 5 folds, the model performance is observed to ensure its stability and the best performing model is saved to be loaded later for performance. The criteria for choosing the best performing model is roughly estimated based on the least combined loss and the combined score closest to 0.
We chose Weights & Biases (Wandb) to log metrics from the training run and to monitor model performance across folds.
Below is the visualisation of the training and validation metrics for both KPIs.

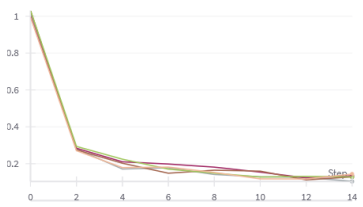


Figure 6.1: Aggregated Training Loss

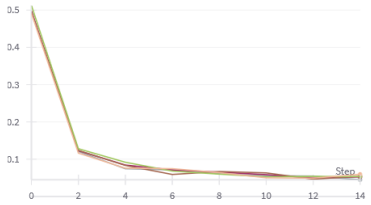


Figure 6.2: Training Loss for Torque Curve

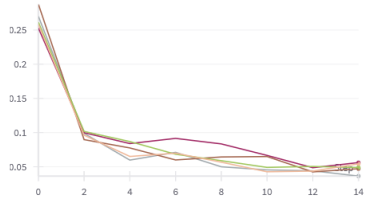


Figure 6.3: Training Loss for ETA grid

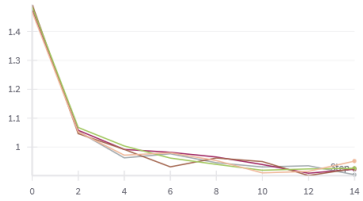


Figure 6.4: Aggregated Training Score

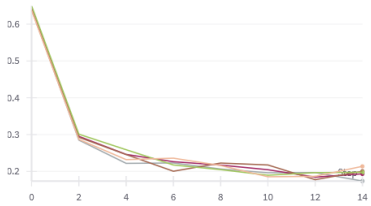


Figure 6.5: Training Score for Torque Curve

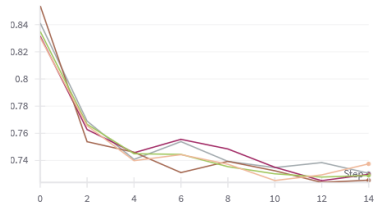


Figure 6.6: Training Score for ETA grid

From the training plots we see that the model has converged after having run for 10 epochs with a learning rate of 0.0075.

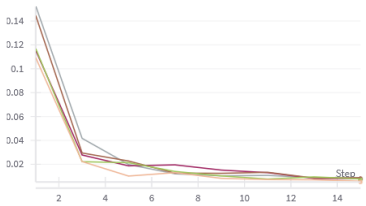


Figure 6.7: Aggregated Validation Loss

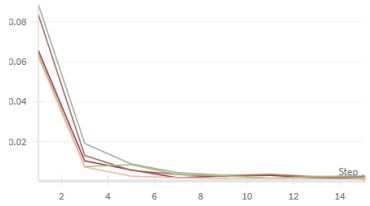


Figure 6.8: Validation Loss for Torque Curve

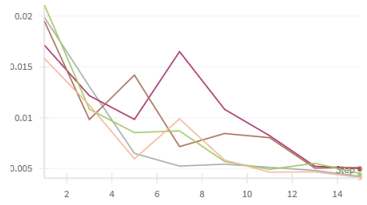


Figure 6.9: Validation Loss for ETA grid

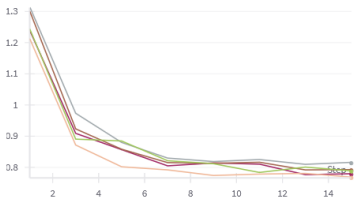


Figure 6.10: Aggregated Validation Score

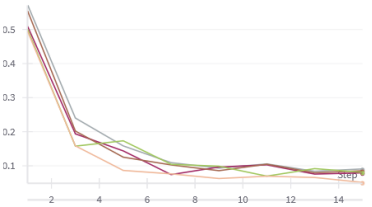


Figure 6.11: Validation Score for Torque Curve

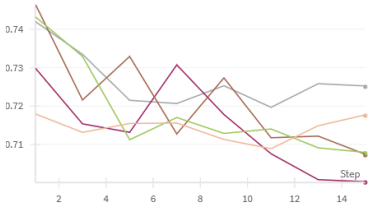


Figure 6.12: Validation Score for ETA grid

We see a good fit of the model to the data with corresponding y1 and y2 scores approaching close to 1. We have also enabled saving the trained model locally so it can be loaded on demand by the client to run

inference.

We have narrowed down scoring to follow the below criteria:

Scoring	Excellent	Very Good	Good	Average	Bad
Y1	0-1	1-2	2-4	4-6	>6
Y2	0-0.5	0.5-1	1-2	2-4	>4

Table 6.2: Scoring Criteria

6.2 Results with MLP

The results of the MLP model from inference is as below:

6.2.1 2D Torque Curve Results with MLP

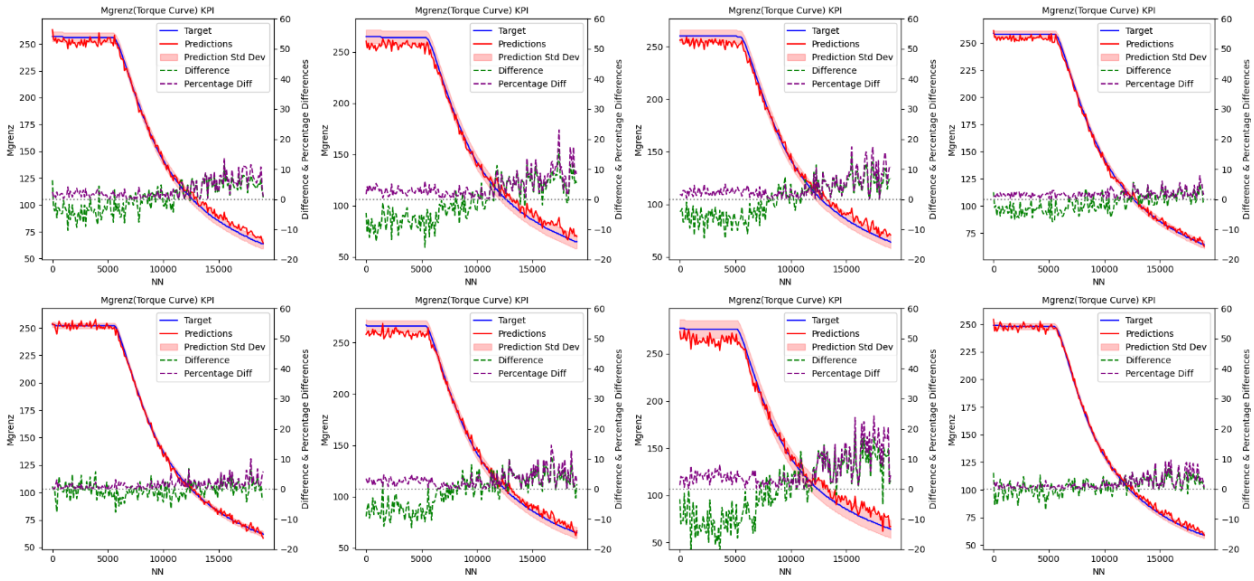


Figure 6.13: MLP Training Results for 2D KPI(Mgrenz)

The Average RMSE and element wise RMSE for the test dataset performance with the MLP is as below:

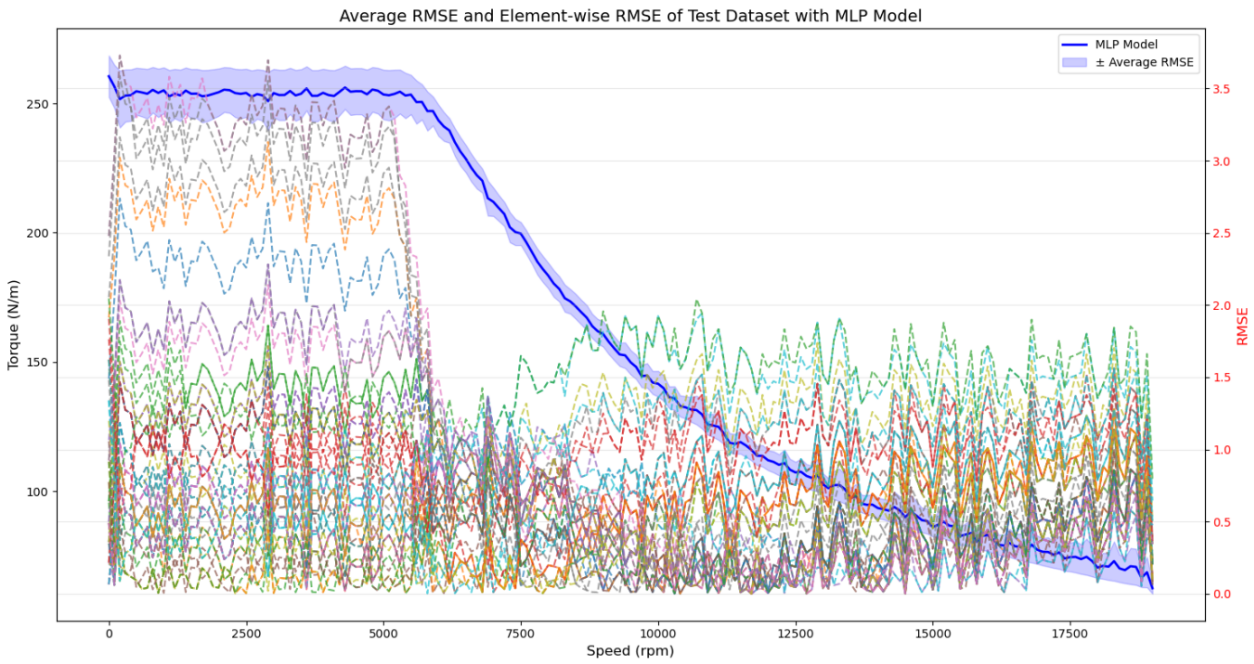


Figure 6.14: Test Dataset RMSE Evaluation for 2D KPI(Mgrenz)

We also have the score statistics of how the model performs over the test dataset.

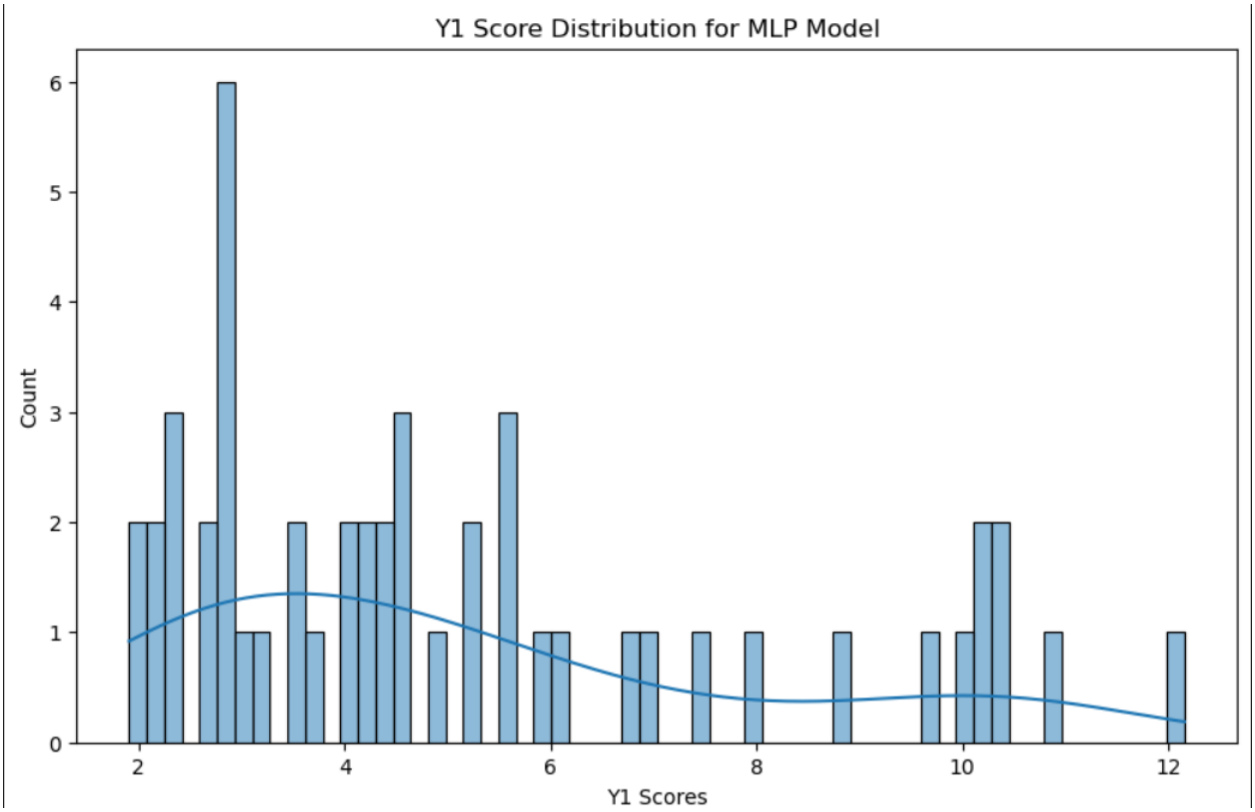


Figure 6.15: MLP Score statistics for 2D KPI(Mgrenz)

6.2.2 3D ETA Grid Results with MLP

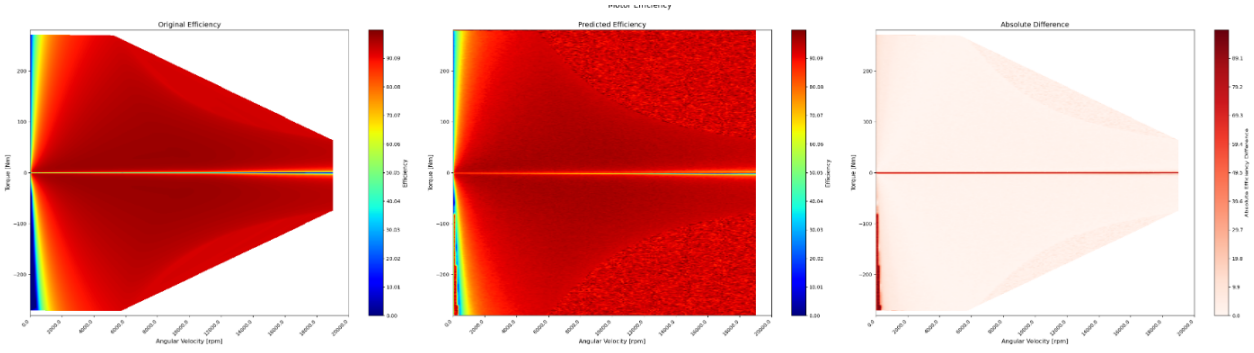


Figure 6.16: 1st MLP Training Results for 3D KPI(ETA)

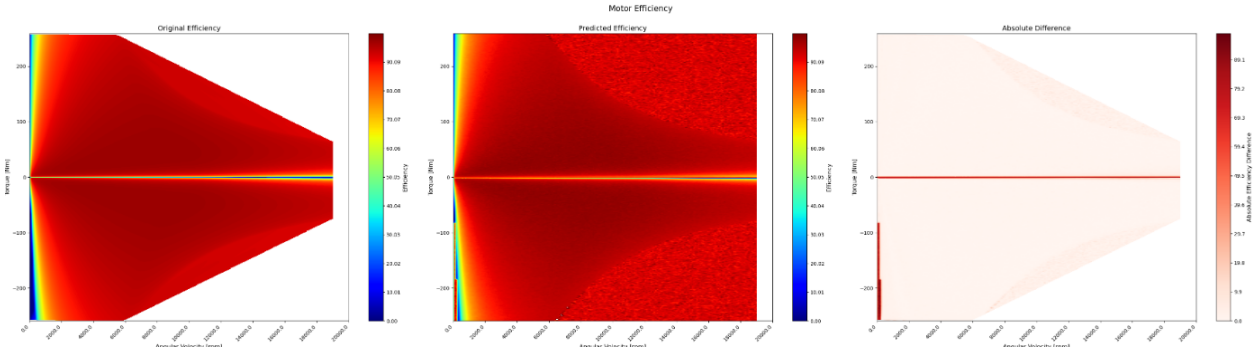


Figure 6.17: 2nd MLP Training Results for 3D KPI(ETA)

Since we are predicting a padded matrix to ensure dimensionality sync different ETA grids, the grid contains values even outside the boundard of the ETA envelope. Hence, we attempted to slice the shape of the Torque curve from the ETA grid by by counting the number of columns a row to have based on consecutive values in the curve. As you can see below, this has resulted in clipping some of the actual ETA envelope with the EM's operating drive cycle.
This brings us back to the point that it is imperative the prediction of the Torque curve is as good as possible to the actual curve as the envelope of the ETA grid is inherently dependent on it.

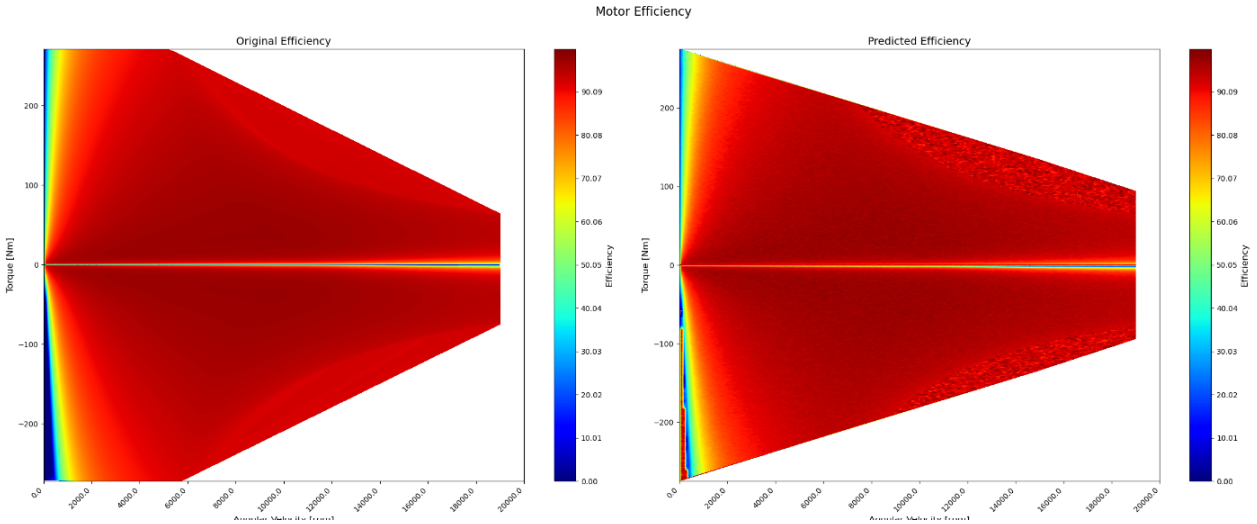


Figure 6.18: 3rd MLP Training Results for 3D KPI(ETA)

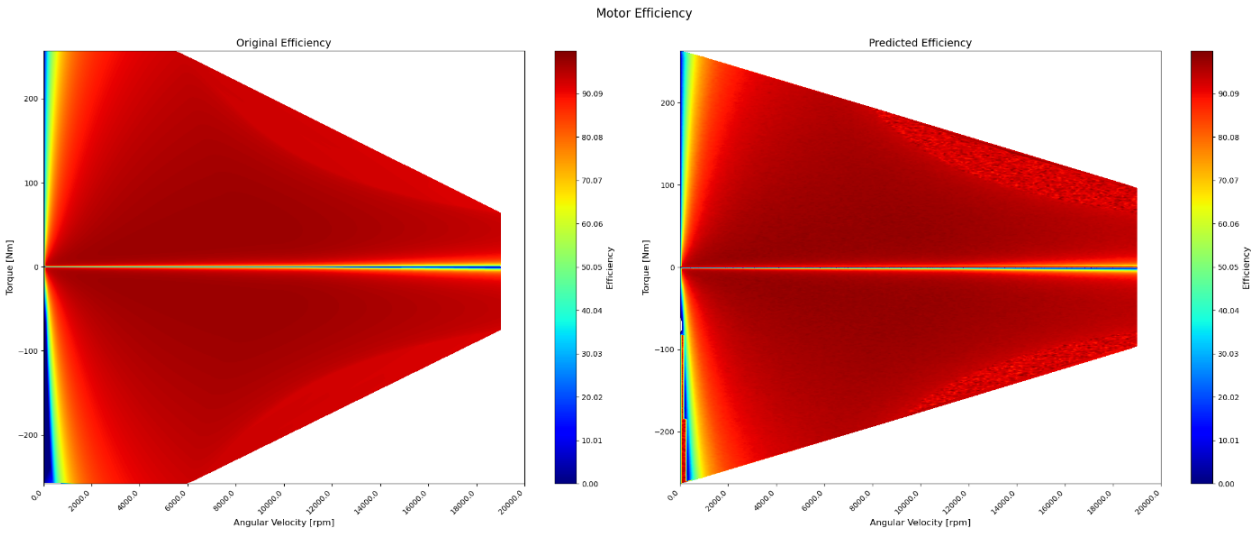


Figure 6.19: 4th MLP Training Results for 3D KPI(ETA)

Observations from the predictions helped to correct few discrepancies in our development for instance in the ETA grid we replaced 0 with NAN values which we later understood were both represented different in the grid.
As efficiency values can take up values only between 0 and 100, we consider the same as constant across plots and use it as a baseline for determining the levels in the contour plot.
We have also left the output predictions for the Torque curve to remain as float values even when the target values are integers to preserve data precision. We give the client the flexibility to turn this on/off demand.
We also have the score statistics of how the model performs over the test dataset.

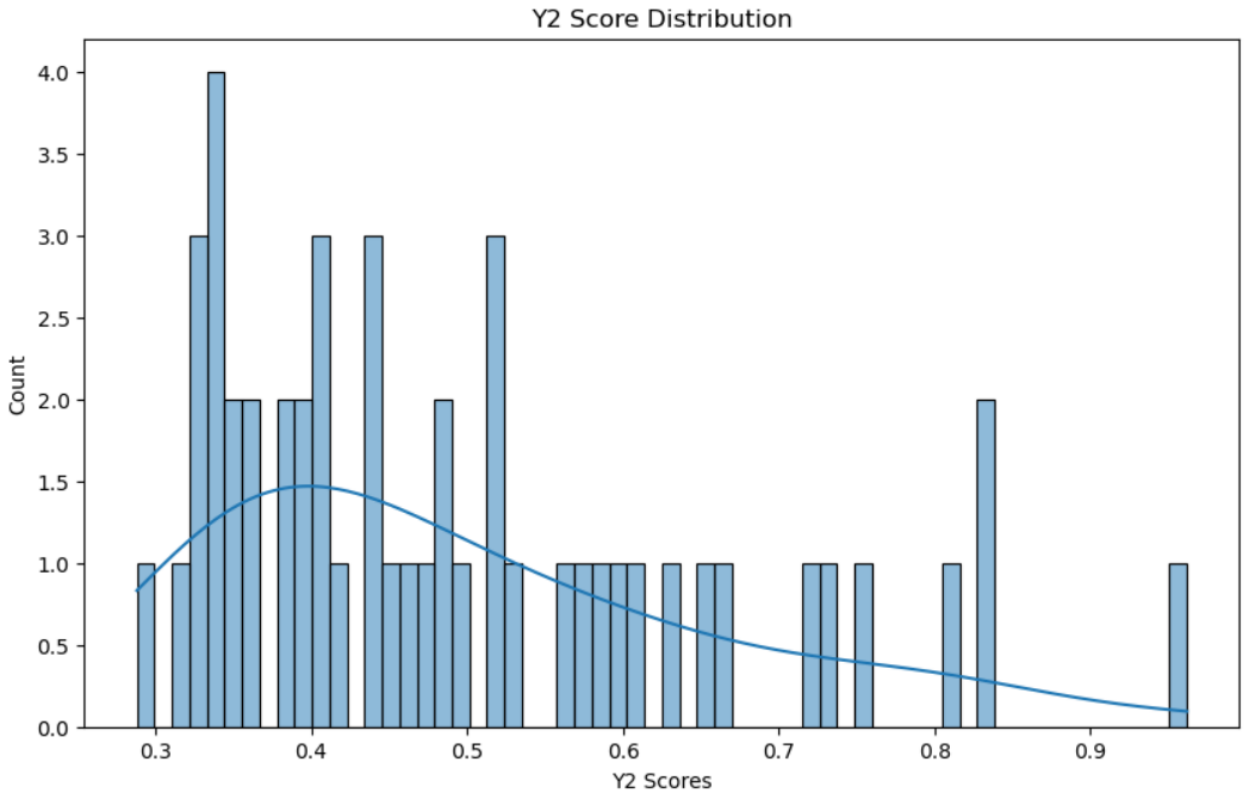


Figure 6.20: MLP Score statistics for 3D KPI(ETA)

We have used python 3.12.2 for our development and the pytorch library compatible with cuda. The model was trained on a NVIDIA V100 GPU with blah blah.

6.3 Results with Baseline

From our observations of how the predictions closely resembled that of the target values, we have proceeded with a baseline model which is essentially the average of the target in all ground truth. Based on the standard deviation of the target values with the baseline as is formulated in Equation CITE, the corresponding plots are generated.

$$\text{Y1 Element wise RMSE} = \sqrt{\frac{1}{n}(\bar{x} - x_i)^2} \quad \forall i \in \{0, \ldots, w-1\} \tag{6.1}$$

where

- n *: EM samples*
- w *: 1st dimension of 3D vector*
- $\bar{x}$ *: Baseline Average mean*

The results of the Baseline model from inference is as below:

6.3.1 2D Torque Curve Results with Baseline

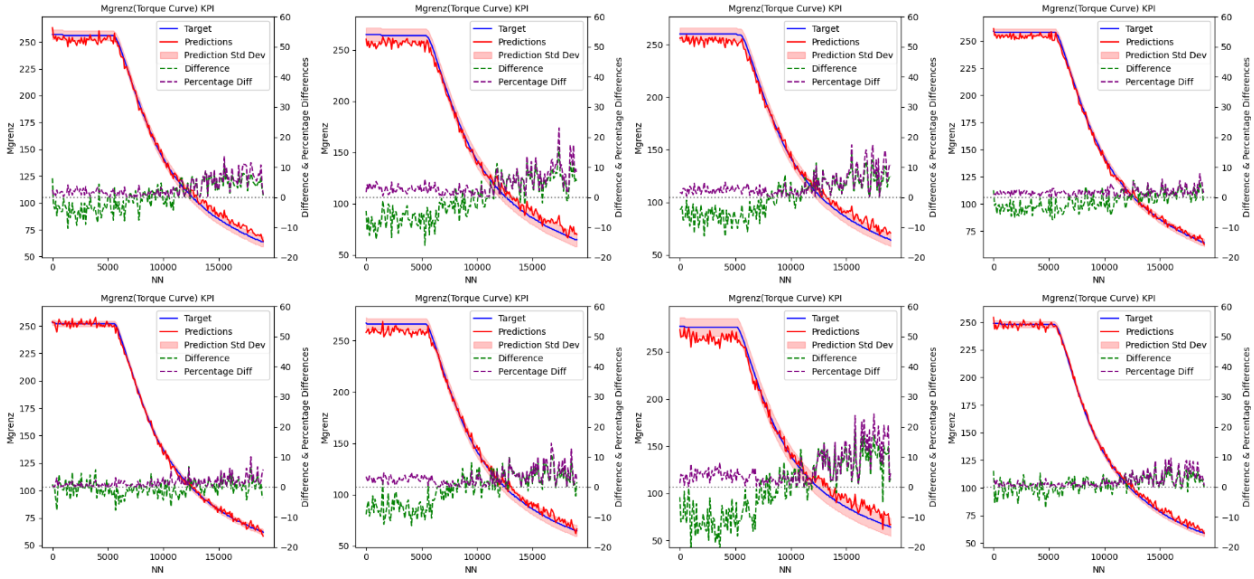


Figure 6.21: Baseline Results for 2D KPI(Mgrenz)

We also have the score statistics of how the Baseline model performs over the test dataset.

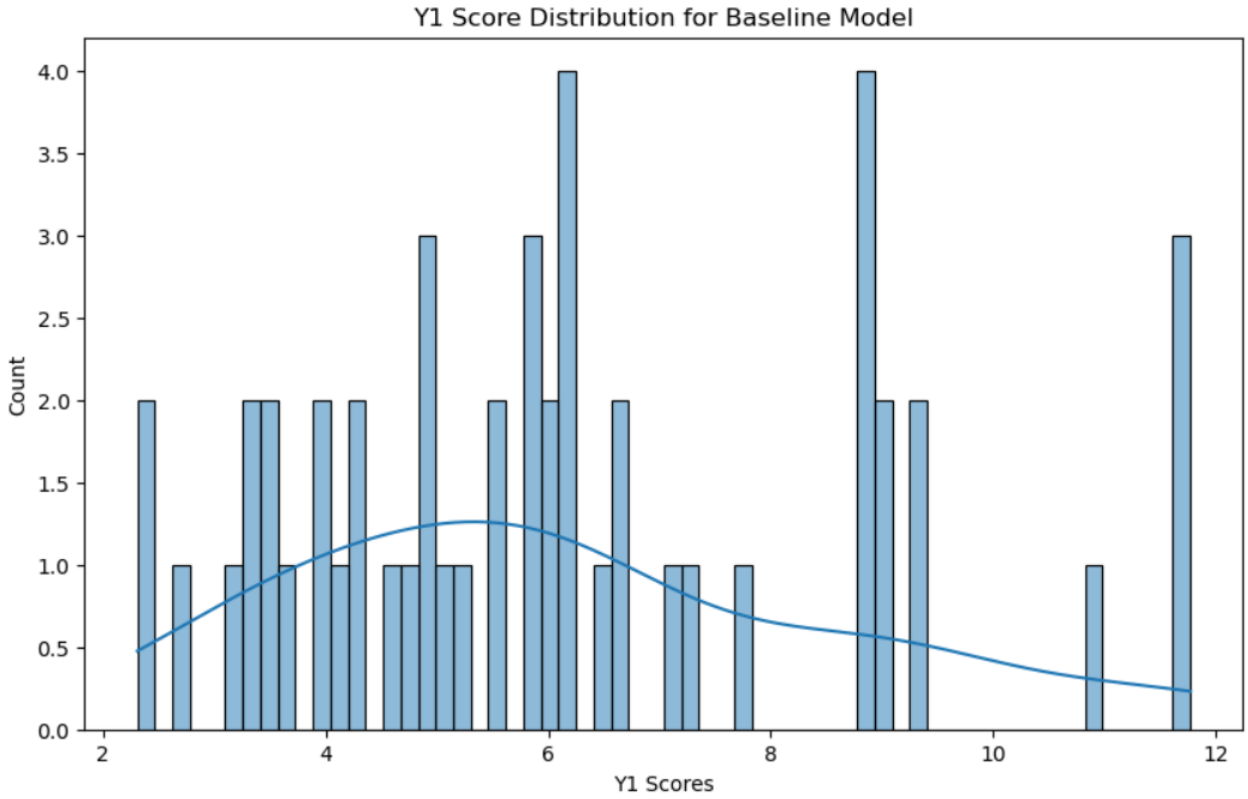


Figure 6.22: Baseline Score statistics for 2D KPI(Mgrenz)

6.3.2 3D ETA Grid Results with Baseline

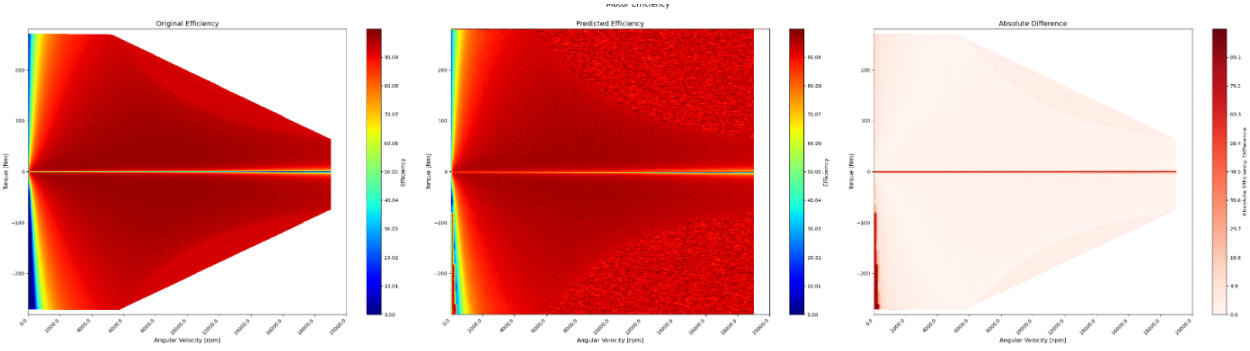


Figure 6.23: 1st Baseline Results for 3D KPI(ETA)

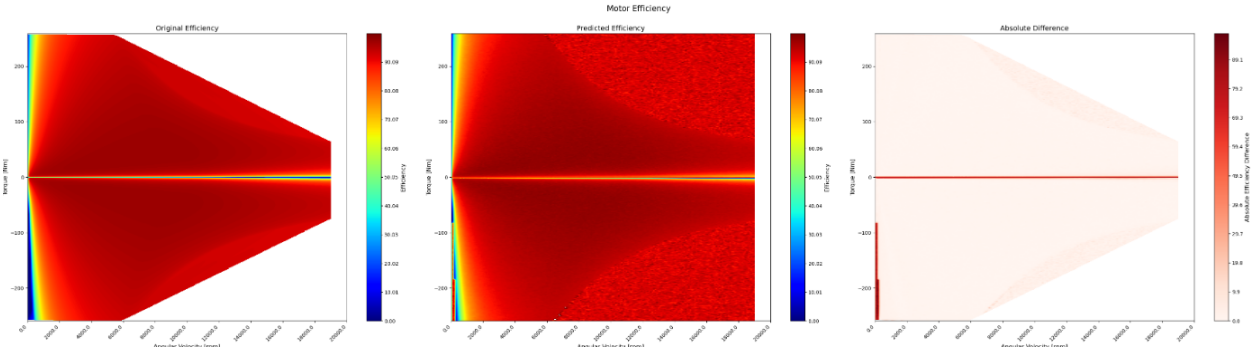


Figure 6.24: 2nd Baseline Results for 3D KPI(ETA)

We also have the score statistics of how the Baseline model performs over the test dataset.

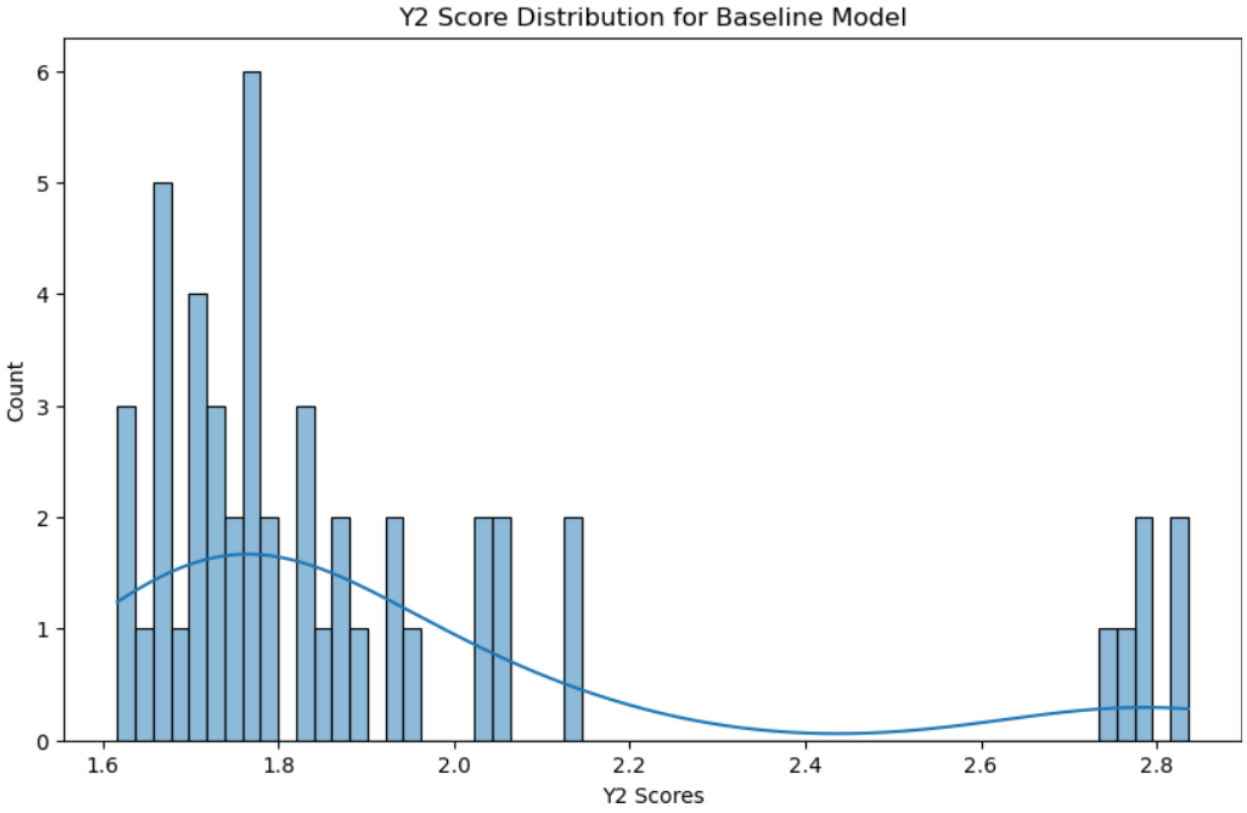


Figure 6.25: Baseline Score statistics for 3D KPI(ETA)

6.4 Ablation Studies

As part of ablation studies, we have compared our evaluations of both the MLP and the Baseline model on both targets respectively.

Model	Y1 Score	Y2 Score
Baseline	6.170	1.927
MLP	5.1994	0.4962

Table 6.3: Ablation Studies

We draw the inferences that our model has performed relatively better than the baseline for 2D KPI. The inferences shown are strictly speaking from the Double V Magnet Topology which assumed the bulk of the data we had received.

Chapter 7

Conclusion

This thesis offers a fresh outlook to the possibility of modelling the performance of an electric motor using GNNs. It also lays the foundation for future work on being able to generate electric motor design parameters conditioned on the 2 KPIs we predicted.

7.1 Future Improvements

I would suggest the following improvements to our study : [1989] 1. Build a model that uses the prediction from Y1 to predict its corresponding Y2. 2. Although we have designed the model to be topology invariant for 3 topologies, we only had sufficient data from Double V Topology to draw evaluations from. Further evaluations on the other topologies would be beneficial to critically assess our model's performance 3. Additionally, the motivation of building a topology invariant model was the reason we have considered building a heterogeneous graph to model the data. The machinery is elaborated in Section CIE. Such a model could serve as yet another ablation study to our problem.

List of Figures

1.1 V1 Magnet (Source:Valeo) . . . 7
1.2 V2 Magnet (Source:Valeo) . . . 7
1.3 Nabla Magnet (Source:Valeo) . . . 7
1.4 EM Design Flowchart . . . 8

3.1 Complete EM Geometry(Source:Valeo) . . . 11
3.2 1/8 Motor Crossection . . . 12
3.3 Standard Deviation of 2D KPI(ETA) Targets . . . 15
3.4 Standard Deviation of 3D KPI(ETA) . . . 16
3.5 Standard Deviation of 3D KPI(ETA) . . . 17

4.1 HetGraph . . . 21

5.1 MLP Model Architecture . . . 24

6.1 Aggregated Training Loss . . . 29
6.2 Training Loss for Torque Curve . . . 29
6.3 Training Loss for ETA grid . . . 29
6.4 Aggregated Training Score . . . 29
6.5 Training Score for Torque Curve . . . 29
6.6 Training Score for ETA grid . . . 29
6.7 Aggregated Validation Loss . . . 29
6.8 Validation Loss for Torque Curve . . . 29
6.9 Validation Loss for ETA grid . . . 29
6.10 Aggregated Validation Score . . . 29
6.11 Validation Score for Torque Curve . . . 29
6.12 Validation Score for ETA grid . . . 29
6.13 MLP Training Results for 2D KPI(Mgrenz) . . . 30
6.14 Test Dataset RMSE Evaluation for 2D KPI(Mgrenz) . . . 31
6.15 MLP Score statistics for 2D KPI(Mgrenz) . . . 32
6.16 1st MLP Training Results for 3D KPI(ETA) . . . 32
6.17 2nd MLP Training Results for 3D KPI(ETA) . . . 33
6.18 3rd MLP Training Results for 3D KPI(ETA) . . . 33
6.19 4th MLP Training Results for 3D KPI(ETA) . . . 34
6.20 MLP Score statistics for 3D KPI(ETA) . . . 35
6.21 Baseline Results for 2D KPI(Mgrenz) . . . 36
6.22 Baseline Score statistics for 2D KPI(Mgrenz) . . . 37
6.23 1st Baseline Results for 3D KPI(ETA) . . . 37
6.24 2nd Baseline Results for 3D KPI(ETA) . . . 38
6.25 Baseline Score statistics for 3D KPI(ETA) . . . 38

List of Tables

3.1 Excel File Structure of an EM variant 13
3.2 EM Input Parameters 15

6.1 Hyperparameter Tuning 28
6.2 Scoring Criteria 30
6.3 Ablation Studies 39

Bibliography

[1989] Arora J. S. *Introduction to Optimum Design.* McGraw-Hill, 1989.

[2001] Deb K. *Multi-Objective Optimization using Evolutionary Algorithms.* John Wiley & Sons, 2001.

Declaration on oath

I hereby certify that I have written my master thesis independently and have not yet submitted it for examination purposes elsewhere. All sources and aids used are listed, literal and meaningful quotations have been marked as such.

Lilly Abraham K64889, 11.12.2024

Consent to Plagiarism Check

I hereby agree that my submitted work may be sent to PlagScan (www.plagscan.com) in digital form for the purpose of checking for plagiarism and that it may be temporarily (max. 5 years) stored in the database maintained by PlagScan as well as personal data which are part of this work may be stored there.

Consent is voluntary. Without this consent, the plagiarism check cannot be prevented by removing all personal data and protecting the copyright requirements. Consent to the storage and use of personal data may be revoked at any time by notifying the faculty.

Lilly Abraham K64889, 11.12.2024